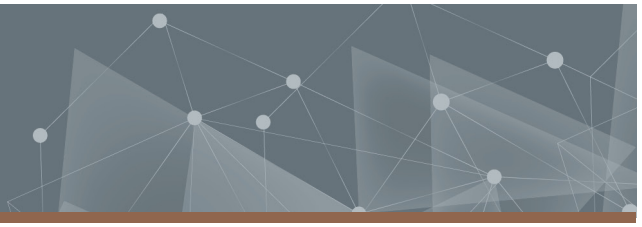


CHALMERS
UNIVERSITY OF TECHNOLOGY



Retail AI Roadmapping
A Data-Driven Portfolio Decision Engine

Master's thesis in Data Science and AI

TED GRANKVIST

DEPARTMENT OF MATHEMATICAL SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis in Data Science and AI 2026

Retail AI Roadmapping: A Data-Driven Portfolio Decision Engine

Ted Grankvist



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Chalmers University of Technology
Gothenburg, Sweden 2026

Retail AI Roadmapping
A Data-Driven Portfolio Decision Engine
Ted Grankvist

© Ted Grankvist, 2026.

Supervisor and examiner: Marina Axelson-Fisk, Chalmers University of Technology
Industry supervisors: Gustav Nirvin, Triathlon AB; Philip Wissinger, Triathlon AB

Master's thesis in Data Science and AI 2026
Department of Mathematical Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Retail AI Roadmapping
A Data-Driven Portfolio Decision Engine
Ted Grankvist
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

Retail companies face a growing number of possible AI initiatives, but the harder problem is often not idea generation. It is deciding which use cases should be prioritised, which ones fit the company context, and which ones are realistic to move beyond experimentation. This thesis addresses that problem by developing a structured decision-support framework for AI use-case prioritisation in retail, based on an organisation $\times$ use-case representation that combines company-side attributes, use-case-side attributes, and computed fit logic.

The empirical material comes from interviews with 33 Nordic retail companies across Sweden, Norway, Denmark, and Finland. The sample was weighted towards larger retailers, with 16 companies in the SEK 1–10bn annual revenue range and 12 above SEK 10bn, and included senior respondents such as CIOs, CTOs, CDOs, and heads of data, technology, or AI. Across these interviews, respondents listed 151 AI use cases, of which 55 were coded in greater depth with structured attributes and manager-rated impact assessment. The study combines descriptive portfolio analysis, baseline impact modelling, model interpretability analysis, cluster-strength analysis, and qualitative interpretation. The baseline impact models showed some predictive signal in the structured interview attributes, especially for focused non-fit feature sets with tree-based ensemble models. However, explanatory performance remained modest, and predictions were compressed towards the middle of the impact scale. The models should therefore be read as support for prioritisation rather than as precise impact predictors.

The results suggest that AI use-case prioritisation in retail should not be reduced either to GenAI adoption or to one universal ranking of use cases. GenAI adoption was widespread, but traditional ML use cases often showed clearer operational impact in the observed material. Data availability was generally stronger than data quality and accessibility, while process-change burden appeared more prominent than technical integration burden. In the observed-only cluster-strength results, internal delivery, high data readiness, high change capacity, and forecasting or inventory-planning use cases appeared as recurring strong patterns, although these should be interpreted as signals rather than general rules.

Overall, the thesis supports a context-sensitive playbook logic. Retailers should prioritise AI use cases by asking where the operational pain is, whether the relevant data is reliable and accessible, what portfolio role the use case should have, what delivery route fits the context, and what operating model is needed after launch.

Keywords: retail AI, AI use-case prioritisation, decision support, portfolio selection, machine learning.

Acknowledgements

I would like to express my sincere gratitude to my Chalmers supervisor and examiner, Professor Marina Axelson-Fisk, for her guidance, feedback, and support throughout the thesis process. Her perspective helped me refine the research direction, strengthen the methodological framing, and develop the work into a more coherent thesis.

I would also like to thank Triathlon Group, where this thesis was carried out, for the opportunity to work closely with the firm and to be based at the office throughout the thesis process. I am especially grateful to Sofia Eklund, Partner at Triathlon, for making this collaboration possible. I would also like to thank my supervisors at the firm, Gustav Nirvin and Philip Wissinger, for their engagement, practical perspective, and continuous support. The weekly supervision meetings with Gustav and Philip, together with the recurring joint meetings with Marina, were highly valuable for shaping both the empirical work and the analytical direction of the thesis.

Finally, I would like to thank all interview respondents who generously contributed their time, experience, and reflections. Their willingness to discuss AI use cases, organisational conditions, and practical implementation challenges made the empirical foundation of this thesis possible.

List of Acronyms

AI	Artificial Intelligence
API	Application Programming Interface
CDO	Chief Digital Officer
CIO	Chief Information Officer
CSV	Comma-Separated Values
CTO	Chief Technology Officer
ETL	Extract, Transform, Load
FTE	Full-Time Equivalent
GenAI	Generative Artificial Intelligence
IT	Information Technology
KPI	Key Performance Indicator
LLM	Large Language Model
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multilayer Perceptron
MSEK	Million Swedish Crowns
OOF	Out-of-Fold
PLS	Partial Least Squares
PoC	Proof of Concept
RAG	Retrieval-Augmented Generation
RMSE	Root Mean Squared Error
ROI	Return on Investment
SEO	Search Engine Optimization

SHAP SHapley Additive exPlanations

SVR Support Vector Regression

Contents

List of Acronyms **x**

List of Figures **xix**

List of Tables **xxv**

1 Introduction **1**
1.1 Background and motivation 1
1.2 Research objective 2
1.3 Research questions 2
1.4 Scope 2
1.5 AI declaration 3

2 Theory **5**
2.1 AI in retail and supply chains 5
2.2 Prioritisation and portfolio selection 5
2.3 Tabular machine learning and decision support 6
2.3.1 Structured prediction with mixed-type data 6
2.3.2 Interpretability in decision support 6
2.3.3 Prediction vs. explanation 7
2.4 Similarity-based inference 7
2.4.1 Similarity as a basis for inference 7
2.4.2 Case-based and prototype-oriented reasoning 7
2.4.3 Limits of similarity-based inference 8
2.5 Segmentation and archetypes 8
2.5.1 Segmentation and heterogeneity reduction 8
2.5.2 Taxonomies and archetypes for analytical comparison 8
2.6 Sparse-data decision support 9
2.7 Text representation and qualitative support 9

3 Methodology **11**
3.1 Research design 11
3.2 Study workflow 11
3.3 Early exploratory phase and framework refinement 12

3.4 Data collection . . . 13
3.4.1 Interviews . . . 13
3.4.2 Sampling logic . . . 13
3.4.3 Contact and recruitment process . . . 14
3.5 Interview instrument . . . 14
3.5.1 General design principles . . . 14
3.5.2 Attribute structure . . . 15
3.5.3 Company attributes . . . 15
3.5.4 Use-case attributes . . . 19
3.5.5 Fit attributes . . . 23
3.5.6 Open-ended questions . . . 27
3.6 Interview procedure . . . 28
3.6.1 Interview flow . . . 28
3.6.2 Structured scales . . . 29
3.6.3 Note capture . . . 29
3.7 Data representation . . . 30
3.7.1 Unit of analysis . . . 30
3.7.2 Structured representation . . . 30
3.7.3 Label logic . . . 30
3.7.4 Open-ended responses . . . 31
3.8 Ethical and methodological considerations . . . 31
3.8.1 Confidentiality and respondent burden . . . 31
3.8.2 Consistency, comparability, and subjectivity . . . 32
3.8.3 Sampling limitations . . . 32
3.9 Technical implementation . . . 32
3.9.1 Purpose of the technical module . . . 32
3.9.2 Dataset-pipeline relation . . . 33
3.10 Analytical dataset construction . . . 33
3.10.1 Workbook parsing and normalisation . . . 33
3.10.2 Normalisation and coding . . . 34
3.10.3 Organisation–use-case rows . . . 34
3.10.4 Sparse-row handling . . . 35
3.11 Feature construction . . . 35
3.11.1 Encoding of structured attributes . . . 35
3.11.2 Derivation of fit attributes . . . 36
3.11.3 Alternative feature-set variants . . . 37
3.12 Baseline impact modelling . . . 38
3.12.1 Prediction objective and label definition . . . 38
3.12.2 Candidate model families and experimental design . . . 38
3.12.3 Preprocessing inside the modelling pipeline . . . 39
3.12.4 Grouped nested cross-validation and hyperparameter selection . . . 39
3.12.5 Evaluation metrics and output artefacts . . . 40
3.13 Impact model interpretability . . . 40
3.13.1 Motivation for interpretability analysis . . . 40

3.13.2 Coefficient-based interpretation of linear models 41
3.13.3 SHAP-based interpretation of nonlinear models 41
3.13.4 Permutation-based robustness check 42
3.13.5 Interpretive role and limitations 43
3.14 Cluster-strength analysis for playbook support 43
3.14.1 Motivation for moving from row-level impact to cluster-level guidance 43
3.14.2 Grouping strategies and pair-universe logic 44
3.14.2.1 Company grouping strategies 44
3.14.2.2 Use-case grouping strategies 46
3.14.3 Observed-only edge-strength computation 50
3.14.4 Prototype-based completion of missing combinations 51
3.14.5 Manual adjustments and smoothing 52
3.14.6 Evaluation of grouping strategies and edge structure 53
3.14.7 Batch experimentation and configuration choices 53
3.15 Reproducibility and experiment management 54
3.15.1 Configuration as the primary control surface 54
3.15.2 Stored manifests, audits, and run outputs 54
3.16 Technical limitations and boundary conditions 55
3.17 Method summary 55

4 Results **57**
4.1 Study sample 57
4.1.1 Respondent role distribution 58
4.1.2 Distributions of interviewed companies 59
4.1.3 AI solution pattern distributions 60
4.2 Portfolio patterns 61
4.2.1 GenAI adoption 61
4.2.2 Company readiness 63
4.2.3 Implementation burden 66
4.2.4 Impact and delivery patterns 70
4.3 Qualitative results 73
4.3.1 AI execution 73
4.3.1.1 AI's bottleneck is execution depth, not idea generation 73
4.3.1.2 Business pull is what turns AI into traction 74
4.3.1.3 Process change often outweighs the technical challenge 74
4.3.1.4 Trust and oversight must be designed per use case . . 74
4.3.1.5 Frontline adoption decides whether AI becomes real . 75
4.3.1.6 Product and supplier data are retail AI bottlenecks . 75
4.3.2 Readiness 76
4.3.2.1 Data readiness means connected, governed context . 76
4.3.2.2 Start where repetitive operational pain is already visible 76
4.3.2.3 Readiness varies by workflow more than by company 77
4.3.2.4 The right AI pattern is not always GenAI 77

4.3.2.5 Different AI cases need different measurement logic . 77
4.3.3 Use-case value . . . 78
4.3.3.1 AI can make previously uneconomical work scalable . 78
4.3.3.2 The workflow wrapper often determines AI value . . 78
4.3.3.3 AI policy should enable safe use, not only restrict it . 78
4.3.3.4 AI reveals undocumented process knowledge 79
4.3.3.5 Fast vendor change favors modular AI architecture . 79
4.3.3.6 Some AI cases build capability before direct ROI . . 79
4.3.3.7 AI leadership mandate is not the same as use-case ownership . . . 79
4.3.4 Governance and operating model . . . 80
4.3.4.1 Data ownership and model-provider risk shape AI adoption . . . 80
4.3.4.2 Agentic AI is an operating-model question 80
4.3.4.3 Explainability is needed where AI affects consequential decisions . . . 81
4.3.4.4 AI capability must spread without becoming ungoverned . . . 81
4.3.4.5 Exploration only matters if it leads to scale discipline 81
4.3.4.6 Strong AI opportunities come from several discovery routes . . . 82
4.3.4.7 Vendor hype makes AI sourcing a strategic choice . . 82
4.3.4.8 Bandwidth scarcity can block willing organisations . 82
4.3.4.9 Customer acceptance constrains customer-facing AI . 82
4.4 Baseline impact modelling . . . 83
4.4.1 Model framing . . . 83
4.4.2 Consolidated non-fit experiment set . . . 83
4.4.3 Feature-set labels . . . 84
4.4.4 Top-25 ranking . . . 85
4.4.5 Top-25 metric summary . . . 86
4.4.6 Model-family comparison . . . 91
4.4.7 Selected winner diagnostics . . . 96
4.4.8 Baseline model interpretation . . . 104
4.5 Impact model interpretability . . . 104
4.5.1 Coefficient-based interpretation . . . 105
4.5.2 SHAP-based interpretation . . . 113
4.5.3 Permutation-importance robustness check . . . 117
4.5.4 Candidate manual-adjustment logic . . . 119
4.5.5 Interpretive caution . . . 121
4.6 Cluster-strength results . . . 121
4.6.1 Configuration-enabled analysis . . . 122
4.6.1.1 Experiment setup . . . 122
4.6.1.2 Variant sensitivity . . . 123
4.6.1.3 Top and stable supported edges . . . 125

4.6.1.4 Selected matrix heatmaps . . . 128
4.6.1.5 Cross-variant stability heatmap . . . 137
4.6.1.6 Technical takeaway . . . 139
4.6.2 Observed-only analysis . . . 140
4.6.2.1 Observed-only takeaway . . . 141

5 Discussion **143**
5.1 Technical interpretation . . . 143
5.1.1 Manager-rated expected impact . . . 143
5.1.2 Common-rater and subjectivity risks . . . 144
5.1.3 Predictive signal in a small dataset . . . 145
5.1.4 Prioritisation over exact prediction . . . 145
5.1.5 Model complexity helped, but only partly . . . 146
5.1.6 Interpretability is diagnostic, not causal . . . 147
5.1.7 Manual adjustments as sensitivity settings . . . 148
5.1.8 Cluster strength as pattern discovery . . . 148
5.2 Retail AI roadmapping . . . 149
5.2.1 Roadmaps should start from readiness and retail context . . . 149
5.2.2 AI portfolios should separate quick wins, strategic bets, and learning cases . . . 150
5.2.3 Delivery model should depend on readiness, ownership, and strategic importance . . . 151
5.2.4 The central bottleneck is moving from use case to operating practice . . . 152
5.2.5 Scaling AI requires explicit operating-model choices . . . 153
5.2.6 The playbook logic is a sequence of fit questions . . . 154
5.3 Research questions . . . 155
5.4 Limitations and boundary conditions . . . 157
5.5 Future work . . . 157

6 Conclusion **159**

Bibliography **161**

A Appendix **I**
A.1 Baseline impact modelling supplement . . . I
A.1.1 Feature-set inventory . . . I
A.1.2 Top-72 non-fit model ranking . . . IV
A.1.3 Top-72 summary plots . . . V
A.2 Interpretability supplement . . . X
A.2.1 Coefficient plots . . . X
A.2.2 SHAP plots . . . XIII
A.2.3 Permutation plots . . . XIX

List of Figures

4.1 Respondent role distribution. 58
4.2 Distributions of interviewed companies by revenue class, channel profile, and retail industry type. 59
4.3 AI solution pattern distributions for all listed use cases and the in-depth impact-rated subset. 60
4.4 GenAI adoption by size class and retail industry type. The retail-industry view should be read together with the denominator labels, since several of the apparent top categories are based on only one to three companies. 62
4.5 Company-readiness distributions across low, medium, and high coded categories. The chart separates overall data readiness from data availability, data quality, and data accessibility in order to show where the data foundation is strong or weaker. 63
4.6 Average data-foundation attributes by size class. Scores are on the coded 1–3 scale, where 3 represents the strongest coded level. 64
4.7 Average data readiness by retail industry type. Scores are on the coded 1–3 scale. The figure should be interpreted cautiously because several industry categories contain few companies. 66
4.8 Implementation-burden distributions across the studied use cases. Low, medium, and high refer to the coded burden tiers for each attribute. 67
4.9 Average process-change burden, integration burden, and ROI-metric clarity by AI solution pattern. Scores are on the coded 1–3 scale, and row labels include the number of use cases in each solution-pattern category. 68
4.10 Average observed impact by AI solution pattern. The dashed line marks the overall observed-impact average of 6.73. Row labels include the number of use cases in each solution-pattern category. 70
4.11 Quick-win matrix by AI solution pattern. The dashed lines mark the overall average impact and average time-to-value values for the plotted categories. Circle size represents the number of use cases in each category. 71
4.12 Average observed impact by use-case family. The dashed line marks the overall observed-impact average of 6.73. 73

4.13 Top-25 consolidated non-fit impact modelling experiments ranked by mean absolute error. The plot shows that the strongest results are concentrated around `opf`, especially when paired with gradient-boosting-style ensemble models. 87

4.14 Global out-of-fold R^2 for the top-25 consolidated non-fit impact modelling experiments. Even the strongest values are modest, which indicates useful but limited explanatory performance in the small interview-derived dataset. 88

4.15 Global Spearman rank correlation for the top-25 consolidated non-fit impact modelling experiments. The figure reports how well each experiment preserves the relative ordering of use-case impact across the pooled out-of-fold predictions. 89

4.16 Ranking-oriented metrics for the top-25 consolidated non-fit impact modelling experiments. The figure supplements error-based metrics with prioritisation-oriented measures such as mean company Spearman and top-1 hit rate. 90

4.17 True-versus-predicted impact diagnostics for the best non-fit result from four model families. The panels show that several simple or weakly regularised baselines compress predictions towards the middle of the impact scale. 93

4.18 True-versus-predicted impact diagnostics for the strongest gradient-boosting results and two contrasting model families. The boosting models with `opf` provide the strongest overall results, while the lasso and MLP examples illustrate the limitations of overly simple or data-hungry approaches in this small-sample setting. 94

4.19 True-versus-predicted impact diagnostics for the remaining best-by-model-family results. The comparison shows that the strongest simple and kernel-based models can be competitive on MAE, but that the strongest tree ensembles recover more of the broad variation in observed impact. 95

4.20 Diagnostic triplet for selected winner w01, `opf` with histogram gradient boosting. This experiment has the lowest observed MAE among the consolidated non-fit models and shows useful signal, while still compressing predictions towards the middle of the impact scale. . . . 97

4.21 Diagnostic triplet for selected winner w02, `opf` with gradient boosting. This experiment has nearly the same MAE as w01, but stronger global Spearman correlation and out-of-fold R^2, making it particularly relevant from a prioritisation perspective. 98

4.22 Diagnostic triplet for selected winner w03, `opf` with random forest. The model remains competitive in absolute error, but its weaker rank-ordering metrics show that similar MAE values can still imply different prioritisation usefulness. 99

4.23 Diagnostic triplet for selected winner w04, `uhlc` with gradient boosting. This experiment shows that a more use-case-heavy feature set can remain competitive, although its out-of-fold R^2 is weaker than the strongest `opf` models. 100

4.24 Diagnostic triplet for selected winner w05, `tbc_dq_pgm` with histogram gradient boosting. This compact balanced-core variant performs close to the strongest models and has the highest top-1 hit rate among the selected winners. 101

4.25 Diagnostic triplet for selected winner w06, `uhlc` with SVR. The experiment is close to the top models in MAE, but its weak global rank-ordering signal makes it less convincing as a prioritisation model. 102

4.26 Diagnostic triplet for selected winner w07, `tiny_balanced_core` without time to value, with SVR. The result reinforces that a competitive MAE alone is not enough to treat a model as strong for prioritisation, since the ranking metrics remain limited. 103

4.27 Attribute-level coefficient magnitude heatmap for selected linear impact models. The heatmap aggregates encoded feature coefficients back to conceptual attributes and therefore shows which attributes were used most strongly by each linear model, without showing whether the direction was positive or negative. 106

4.28 Attribute-level signed coefficient heatmap for selected linear impact models. Red indicates a positive net coefficient direction and blue indicates a negative net coefficient direction. For categorical attributes, the aggregate can hide category-level differences, so selected feature-level coefficient plots are also inspected. 108

4.29 Feature-level coefficient plot for the `opf` ridge model. The plot shows why categorical attributes should be inspected at encoded-feature level: for example, realtime data recency is positive, while static or no-data recency categories are negative. 110

4.30 Feature-level coefficient plot for the `chlu` PLS regression model. This plot is mainly used as a diagnostic complement to the ridge view, especially for the negative direction of `time_to_value`. 112

4.31 Attribute-level mean absolute SHAP values for selected nonlinear impact models. Higher values indicate that the attribute contributed more strongly to fitted predictions on average, but not whether the direction was positive or negative. 114

4.32 Attribute-level SHAP direction diagnostic for selected nonlinear impact models. The direction is based on the relationship between encoded feature values and SHAP values. It is most directly interpretable for ordinal attributes and should be read more cautiously for categorical attributes. 116

4.33 Attribute-level permutation importance for selected nonlinear impact models. Values represent the mean increase in MAE after permuting the attribute group. Higher values indicate that the fitted model relied more strongly on that attribute. 118
4.34 Configuration-enabled cluster-strength heatmap for channel profile and use-case family. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison. 129
4.35 Configuration-enabled cluster-strength heatmap for data readiness tier and AI solution pattern. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison. 130
4.36 Configuration-enabled cluster-strength heatmap for data readiness tier and delivery source. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison. 131
4.37 Configuration-enabled cluster-strength heatmap for IT competence and delivery source. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison. 132
4.38 Configuration-enabled cluster-strength heatmap for change capacity and delivery source. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison. 133
4.39 Configuration-enabled cluster-strength heatmap for retail industry type and delivery source. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison. 134
4.40 Configuration-enabled cluster-strength heatmap for execution-readiness tier and AI solution pattern. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison. . . 135
4.41 Configuration-enabled cluster-strength heatmap for data readiness tier and use-case family. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison. 136

4.42 Cross-variant stability heatmap for selected supported cluster-strength edges. Cells show edge strength by configuration variant. Row labels include observed row count n. The colour scale is fixed at 5.5–8.0 for comparison. 138

A.1 Extended top-72 MAE ranking for consolidated non-fit baseline impact modelling experiments. VI
A.2 Extended top-72 global out-of-fold R^2 summary for consolidated non-fit baseline impact modelling experiments. VII
A.3 Extended top-72 global Spearman summary for consolidated non-fit baseline impact modelling experiments. VIII
A.4 Extended top-72 ranking-metrics summary for consolidated non-fit baseline impact modelling experiments. IX
A.5 Feature-level coefficient plot for `opf` with ridge regression. X
A.6 Feature-level coefficient plot for `opf` with Bayesian ridge regression. . XI
A.7 Feature-level coefficient plot for `chlu` with PLS regression. XII
A.8 Feature-level SHAP plot for `opf` with histogram gradient boosting. . XIII
A.9 Feature-level SHAP plot for `opf` with gradient boosting. XIV
A.10 Feature-level SHAP plot for `opf` with random forest. XV
A.11 Feature-level SHAP plot for `opf` with extra trees. XVI
A.12 Feature-level SHAP plot for `uhlc` with gradient boosting. XVII
A.13 Feature-level SHAP plot for `tbc_dq_pgm` with histogram gradient boosting. XVIII
A.14 Feature-level permutation-importance plot for `opf` with histogram gradient boosting. XIX
A.15 Feature-level permutation-importance plot for `opf` with gradient boosting. XX
A.16 Feature-level permutation-importance plot for `opf` with random forest. XXI
A.17 Feature-level permutation-importance plot for `opf` with extra trees. . XXII
A.18 Feature-level permutation-importance plot for `uhlc` with gradient boosting. XXIII
A.19 Feature-level permutation-importance plot for `tbc_dq_pgm` with histogram gradient boosting. XXIV
A.20 Feature-level permutation-importance plot for `uhlc` with SVR. XXV
A.21 Feature-level permutation-importance plot for `tiny_balanced_core` without time to value, with SVR. XXVI

List of Tables

4.1 Study-sample overview. 58
4.2 Average data readiness, IT competence, and change capacity by size class. Scores are on the coded 1–3 scale, where 3 represents the strongest coded level. 65
4.3 Average integration burden by solution type. Scores are on the coded 1–3 scale, where higher values indicate heavier integration burden. . . 67
4.4 Delivery-source distribution among observed-impact use cases. 72
4.5 Delivery-model profile by delivery source. Impact is measured on the 0–10 observed-impact scale. Time to value and maintenance cost are measured on coded 1–3 scales, where higher values indicate slower time to value or heavier maintenance. 72
4.6 Feature-set labels used in the reported baseline impact modelling results. 84
4.7 Top-25 consolidated non-fit baseline impact modelling experiments ranked by MAE. 85
4.8 Best consolidated non-fit impact modelling result for each model family. 91
4.9 Selected final non-fit impact modelling experiments used for detailed diagnostic inspection. 96
4.10 Coefficient-derived candidate manual-adjustment settings for later cluster-strength experimentation. 120
4.11 SHAP- and permutation-derived candidate manual-adjustment settings for later cluster-strength experimentation. 120
4.12 Configuration variants used in the configuration-enabled cluster-strength analysis. 123
4.13 Variant-level sensitivity comparison for the configuration-enabled cluster-strength experiments. 124
4.14 Strongest supported edge per configuration variant, requiring at least five observed rows. 125
4.15 Strongest de-duplicated edges after requiring at least ten observed rows. 126
4.16 Selected supported edges used for cross-variant stability inspection. . 127
4.17 Observed-only delivery-source and internal-adaptation edges. Coverage is the observed share of the relevant company-use-case pair universe. 140

4.18 Observed-only AI solution pattern and use-case archetype edges. Coverage is the observed share of the relevant company-use-case pair universe. . . . 141
4.19 Observed-only readiness and implementation-complexity edges. Coverage is the observed share of the relevant company-use-case pair universe. . . . 141

A.1 Full feature-set inventory for report-mentioned non-fit baseline impact modelling feature sets. . . . I
A.2 Extended top-72 consolidated non-fit baseline impact modelling experiments ranked by MAE. . . . IV

1

Introduction

1.1 Background and motivation

In recent years, AI has taken a more visible role in retail and supply chain related decision-making. This includes applications such as forecasting, inventory planning, anomaly detection, customer service support, content generation, and different forms of operational decision support. From a business perspective, the interest is understandable. Many of these use cases are linked to practical problems such as stockouts, tied-up working capital, manual work, poor service levels, slow response times, or weak process visibility.

However, in practice, the main difficulty is often not to identify possible AI ideas. A possible reason is that most medium-sized and large retail companies already have a relatively long list of candidate initiatives. The harder question is instead how to move from that long list to a smaller and more realistic set of priorities. In other words, the problem is often one of prioritisation and sequencing rather than idea generation alone.

This challenge becomes more difficult because AI initiatives do not exist in a vacuum. The likely value of a use case depends on the company's goals, data situation, technical capabilities, governance maturity, privacy posture, change capacity, and operating model. At the same time, the use case itself has its own profile. Some use cases require strong data foundations and heavy systems integration, while others can create value more quickly with lower technical complexity. As a result, the same use case may be attractive in one company and poorly timed in another.

This thesis is motivated by that practical problem. The work aims to support more structured prioritisation of retail AI use cases by representing both sides of the decision problem. More specifically, the framework separates the company profile from the use-case profile and later introduces fit logic between them. The broader intention is to support retail AI use-case prioritisation in a way that can later be translated into playbook-oriented guidance.

1.2 Research objective

The overall objective of the thesis is to develop a structured and reusable framework for prioritising AI use cases in retail. The framework should help evaluate candidate use cases based on three types of information: the company context, the characteristics of the use case itself, and the fit between the two. In practice, the ambition is therefore not to build a generic "AI score," but to support a more context-sensitive decision logic.

A second objective is that the framework should remain practically usable. This means that it should not only produce technical outputs, but also be interpretable enough to support explanation, discussion, and later use in advisory or workshop settings. In that sense, the thesis combines a data-driven ambition with a decision-support ambition.

1.3 Research questions

Based on the project framing, the thesis is guided by the following research questions:

RQ1. How can a structured interview-based dataset, where each observation represents an organisation × use-case pair, be used to model the expected impact of AI use cases in retail?

RQ2. How can company attributes, use-case attributes, and computed fit attributes be organized into a representation that supports more context-sensitive AI use-case prioritisation?

RQ3. How can the outputs of the structured analytical framework be translated into practically usable guidance for retail AI use-case prioritisation and playbook-oriented decision support?

1.4 Scope

The thesis concerns decision support for prioritisation of AI initiatives in retail. It does not include full production implementation of specific AI systems inside the participating companies. Similarly, the thesis does not attempt to measure realised operational KPI effects from implemented solutions. Instead, the focus is on structured assessment, comparison, and decision support.

The empirical scope is also limited to retail. This means that the interviews, use cases, and practical interpretation are framed around retail organisations and retail-relevant business problems. While some of the underlying logic may be transferable to other sectors, the study does not claim general applicability across industries without further testing.

On the technical side, the thesis is limited to structured data preparation, experimental modelling, interpretability analysis, and cluster-level decision-support logic.

It does not include deployment of production systems, live operational integration, or direct measurement of realised post-deployment KPI effects. The focus is therefore on analytical evaluation and prioritisation support rather than on production implementation of AI solutions.

1.5 AI declaration

Generative AI tools were used as supporting tools during the thesis process. ChatGPT was used to help refine language, improve clarity, and suggest possible structures for selected sections. These suggestions were reviewed and used selectively by the author.

Programming-oriented AI tools, including Codex and Claude Code, were also used to support the technical implementation of the modelling pipeline, including code development, debugging, and refinement of analysis scripts.

All content, analysis, interpretations, and conclusions presented in this thesis are the author's own.

2

Theory

2.1 AI in retail and supply chains

The use of AI in supply chain management and retail has been studied in a growing body of literature. A recurring observation is that value often emerges in areas such as forecasting, replenishment, inventory control, operational monitoring, and customer-facing support. At the same time, the literature also suggests that value creation depends on more than the choice of model family alone. In practice, organisational readiness, data quality, process integration, and managerial anchoring are often decisive factors (Toorajipour et al., 2021; Culot et al., 2024; Helo & Hao, 2022).

This is important for the present thesis because it supports the idea that AI use-case prioritisation cannot be reduced to a purely technical question. A use case may be attractive in principle, but still be a poor near-term candidate if the company lacks the required data foundations, integration capability, or organisational readiness. In that sense, the literature supports a broader framing where technical attractiveness and organisational feasibility need to be considered together.

2.2 Prioritisation and portfolio selection

The prioritisation of initiatives under constraints is closely related to the literature on project portfolio selection and multi-criteria decision-making. In that literature, organisations are often assumed to face a set of candidate initiatives, limited resources, and several competing objectives. A common methodological idea is therefore to combine different evaluation dimensions and to select projects under effort, cost, or risk constraints (Kandakoglu et al., 2024; Nemhauser & Wolsey, 1988; Kellerer et al., 2004).

This literature is relevant here because retail AI prioritisation has a similar structure. In practice, a company rarely decides whether AI is good or bad in general. Instead, it decides which use cases to start with, what to delay, and what to avoid for now. That makes sequencing and constrained prioritisation central. The present thesis builds on this general decision logic, but adapts it to a retail AI setting and to an interview-based empirical design.

In this type of setting, prioritisation is not only a question of expected upside. It

also requires comparing that upside with organisational and operational constraints. This means that a useful framework should not only estimate which use cases appear promising, but should also support structured comparison between different kinds of use cases and different kinds of company contexts. That is one of the main reasons why the later technical framework combines row-level impact estimation with broader group-level interpretation.

2.3 Tabular machine learning and decision support

2.3.1 Structured prediction with mixed-type data

When decision objects can be represented as structured rows with numeric, ordinal, and categorical attributes, tabular machine learning methods are natural practical candidates. The literature shows that methods such as regularised linear models, random forests, and gradient boosting can perform well on this kind of mixed-type data, especially when the dataset is not extremely large (Breiman, 2001; Chen & Guestrin, 2016; Ke et al., 2017; Prokhorenkova et al., 2018). This is relevant because the structured part of the project consists mainly of company attributes, use-case attributes, and later derived fit attributes.

For this thesis, the point is fairly direct. The study does not start from images, free text, or relational clickstream data as its main analytical object. Instead, it starts from a structured interview design that is later translated into a row-wise dataset. Therefore, the theoretical question is not which AI method is strongest in general, but which modelling logic is suitable when the evidence is a relatively small structured table with mixed attribute types and a strong need for careful interpretation.

2.3.2 Interpretability in decision support

However, prediction quality is not the only relevant criterion in a decision-support setting. If a model is later meant to support prioritisation discussions, managerial interpretation, or playbook-oriented recommendations, then it also matters whether the drivers behind the model output can be inspected in a meaningful way. This is one of the central reasons why interpretability is relevant in the present thesis. The goal is not only to produce a score, but also to understand which structured conditions seem to matter and in what direction they seem to matter.

In the wider literature, this issue is often discussed as a trade-off between predictive power and interpretability, although in practice the balance is more nuanced than that. Rudin argues that when decisions are important and explanations matter, there are strong reasons to prefer inherently interpretable models where possible rather than relying entirely on black-box models with post-hoc explanations (Rudin, 2019). At the same time, explainability methods such as SHAP remain relevant because they provide one route for inspecting less transparent models as well (Lundberg &

Lee, 2017). For this thesis, the main theoretical implication is that interpretability should be treated as part of model usefulness, not only as an optional extra after prediction has already been optimized.

2.3.3 Prediction vs. explanation

Another relevant theoretical distinction is the difference between predictive modelling and explanatory modelling. Shmueli argues that these are related but not identical aims, and that a model can be useful for prediction without necessarily providing a strong explanatory account of the underlying causal structure (Shmueli, 2010). This distinction is highly relevant here. The impact-model part of the thesis is not meant to establish causal effects of company or use-case attributes. Rather, it is meant to estimate patterns that may support prioritisation and to inspect whether these patterns appear stable and conceptually plausible.

This also shapes how later interpretability outputs should be read. If one attribute repeatedly appears as influential in a predictive setting, that can still be useful. It may suggest that the attribute captures an important condition in the structured dataset. However, it does not by itself prove that the attribute is causally responsible for impact. In practice, this means that coefficient-based or feature-attribution-based signals are theoretically valuable as structured effect indications, but they should be treated cautiously and interpreted together with the broader design of the study rather than as isolated proof.

2.4 Similarity-based inference

2.4.1 Similarity as a basis for inference

When direct observations are incomplete, one common analytical idea is to infer unknown cases from nearby known cases. In nearest-neighbour style methods, an unlabelled observation is approximated from one or more labelled observations that are close to it in the chosen feature space (Cover & Hart, 1967). The theoretical appeal of this logic is straightforward. If two cases are similar along the attributes believed to matter, then it can be reasonable to expect that their outcomes may also be similar, at least as a first approximation.

This logic is relevant for the thesis because not every company–use-case combination is observed directly in the interview material. The analytical problem is therefore partly sparse by construction. It is not only a question of predicting one observed row better than another, but also a question of how to reason carefully about structured combinations that were not explicitly labelled in the original data collection.

2.4.2 Case-based and prototype-oriented reasoning

The same general idea also appears in the literature on case-based reasoning, where previous cases are reused as analogies for new decision situations (Aamodt & Plaza,

1994). In that tradition, the point is not that one has discovered a universal rule in the strong scientific sense. Rather, the point is that a new case may be handled by retrieving similar earlier cases and adapting their lessons in a systematic way. This is conceptually close to what the present thesis later does in its more sparse analytical layers.

For the present project, this theoretical perspective matters because it gives a principled basis for similarity-based completion. If an organisation–use-case pair has not been observed directly, the next-best source of evidence may be a set of observed pairs that are structurally similar in terms of encoded attributes. This does not remove uncertainty, and it does not make the synthetic case fully equivalent to a true observed case. Still, it provides a more disciplined basis for inference than arbitrary guessing, especially when the goal is decision support under incomplete coverage.

2.4.3 Limits of similarity-based inference

At the same time, similarity-based reasoning is only as strong as its representation. What counts as a "near" case depends entirely on how cases are encoded, scaled, and compared. Therefore, similarity is not a neutral property of the data. It is partly a modelling choice. In practice, this means that the usefulness of prototype-style reasoning depends strongly on whether the chosen feature space captures the practically relevant dimensions of similarity. This limitation is important to state already at the theory level, because it later becomes one of the key boundary conditions of the technical implementation.

2.5 Segmentation and archetypes

2.5.1 Segmentation and heterogeneity reduction

Another central idea in the thesis is that heterogeneous cases often become more interpretable when they are compared through broader segments rather than only as isolated rows. In the segmentation literature, grouping is used to reduce complexity and to create categories that can support comparison, communication, and action (Punj & Stewart, 1983). This matters because a manager usually does not reason naturally in terms of one anonymous row in a dataset. Instead, the more natural reasoning pattern is often closer to "this type of retailer" and "this type of use case."

This is relevant for the present study because the later playbook-oriented analysis is not mainly interested in one-off case descriptions. It is interested in whether broader company contexts and broader use-case types appear to fit well together. In other words, segmentation in this thesis is not mainly a descriptive convenience. It is part of the logic for producing interpretable decision-support signals.

2.5.2 Taxonomies and archetypes for analytical comparison

In addition, the grouping logic in the thesis is not purely unsupervised. It also relies on explicit taxonomies. Nickerson et al. describe taxonomy development as

a structured way of organizing objects into dimensions and characteristics so that comparison becomes more systematic (Nickerson et al., 2013). This is an important theoretical point for the project. Some of the later groupings are not meant to discover hidden latent clusters automatically. They are meant to create stable and interpretable archetypes that reflect meaningful business or technical distinctions.

This distinction is worth emphasizing. A taxonomy-based grouping strategy and a data-driven cluster analysis do not answer exactly the same question. A learned cluster may optimize internal statistical similarity, while a manually defined taxonomy may optimize interpretability and practical relevance. In a managerial decision-support setting, the second goal can be very important. Therefore, the theoretical role of grouping in this thesis is not only pattern compression, but also interpretive scaffolding. It provides a structured way to compare otherwise diverse company and use-case observations at a level that later can be communicated in a playbook.

2.6 Sparse-data decision support

Taken together, the previous sections suggest that sparse applied decision support often cannot rely on one analytical logic alone. If one insists on using only fully observed cases, the evidence may remain clean but too thin to support broader comparison. If one relies only on predictive modelling, one may gain coverage but lose some transparency. If one relies only on broad taxonomies, one gains interpretability but may lose too much row-level nuance. Therefore, a practical framework in a sparse setting often becomes hybrid almost by necessity.

For the present thesis, this means that the final analytical logic is better understood as a structured combination of partial evidence sources than as one single definitive model. Predictive modelling contributes row-level pattern estimation. Interpretability analysis contributes structured effect signals. Similarity-based reasoning contributes one possible way to handle incomplete pair spaces. Grouping strategies contribute interpretable comparison units. Overall, the theoretical justification for this combined logic is not that it eliminates uncertainty, but that it may organize uncertainty in a disciplined and practically usable way. This is also why the framework should be judged not only by predictive accuracy in isolation, but by whether it supports careful prioritisation reasoning under constrained evidence (Shmueli, 2010; Rudin, 2019; Aamodt & Plaza, 1994; Nickerson et al., 2013).

2.7 Text representation and qualitative support

Open-ended organisational responses can also be represented numerically and grouped into themes or archetypes. The literature on transformer-based embeddings and clustering is therefore relevant for the more qualitative side of the project (Devlin et al., 2018; Reimers & Gurevych, 2019; MacQueen, 1967; Rousseeuw, 1987). In

practical terms, this literature matters because organisations often describe their AI needs, barriers, and goals in text rather than in neat structured variables.

However, it is important to state that open-ended text is not the primary structured input to the first version of the analytical framework. Rather, text is theoretically relevant because it can later support segmentation, richer interpretation, and playbook development. In other words, the theory justifies why text may matter, even if the structured core of the project comes first.

There is also a broader methodological reason to keep the text dimension in view. Structured attributes are good for explicit comparison, but they do not capture every nuance of why a retailer sees one use case as difficult, attractive, politically sensitive, or strategically urgent. Therefore, even if the first quantitative layer of the thesis is tabular, the later qualitative and text-oriented layer remains important as a complement. It can help contextualize structured results and can also reveal themes that are not visible in the more rigid attribute representation.

3

Methodology

This chapter describes both how the empirical basis of the study was created and how that empirical material was translated into the technical analysis pipeline. It first explains the exploratory phase, the interview-based data collection process, the development of the interview instrument, and the representation of the resulting organisation × use-case observations. It then describes the technical implementation used to transform the collected material into analytical datasets, encoded feature representations, baseline impact-model experiments, interpretability analyses, and cluster-strength analyses for playbook-oriented decision support.

3.1 Research design

The study followed an applied and design-oriented research approach. The main objective was not only to describe how retail companies currently work with AI, but also to develop a practical decision-support framework that can support more structured prioritisation of AI use cases in retail. In practice, this means that the project combined empirical data collection with framework development. The empirical side was needed in order to ground the work in real organisational contexts, while the design-oriented side was needed because the final goal was to construct something reusable, that is, a framework that can support evaluation and prioritisation of AI initiatives and later contribute to playbook-oriented guidance.

A central implication of this design was that the work could not start directly with modelling. Instead, the first phase had to focus on understanding how retail companies actually talk about AI initiatives, what kinds of use cases are relevant in practice, which organisational conditions matter most, and how these factors can be translated into structured attributes that are comparable across firms. For that reason, the project began with exploratory expert interviews and later moved into a more standardised interview format.

3.2 Study workflow

Overall, the study was carried out in two connected phases. The first phase was empirical and consisted of four main steps. First, an early exploratory phase was used to clarify the practical decision problem and refine the initial framework. Secondly,

potential respondents were identified and recruited, mainly among retail companies with some form of AI, analytics, data, or digitalization activity. Thirdly, semi-structured interviews were conducted using a structured interview guide designed specifically for this study. Fourthly, the collected material was organized into a dataset structure aligned with the conceptual framework of the project.

The second phase was technical. In that phase, the collected material was transformed through ETL and normalisation, translated into machine-readable feature representations, used in baseline impact-model experiments, analysed through coefficient-based interpretability methods, and later extended into cluster-strength analysis intended to support playbook-oriented guidance.

This sequence is important to state clearly, because the interview instrument did not emerge fully formed from the beginning. Rather, it was gradually refined. In practice, the project moved from a broader and somewhat flatter initial framing towards a more explicit representation in which the company, the use case, and the compatibility between them were treated as distinct conceptual components. That refinement shaped both the interview guide, the structure of the collected material, and the later technical representation logic.

3.3 Early exploratory phase and framework refinement

The starting point of the project was an initial proposal with a predefined candidate list of retail AI use cases and an early common attribute template. As the early expert interviews progressed, it became increasingly clear that the framework needed a more explicit internal structure.

More specifically, the early phase suggested that it was not analytically clean to treat all structured inputs as if they belonged to one undifferentiated block. Some variables were really about relatively stable organisational conditions, such as data readiness, change capacity, or governance posture. Others were more naturally interpreted as intrinsic properties of the use case itself, such as time-to-value, data need, integration burden, or operational risk. In addition, a third type of factor emerged, namely the fit between a company and a use case. For example, a use case may in itself be attractive, but still be a weak near-term candidate for a specific company if the firm lacks relevant data, integration capability, or willingness to adapt ways of working.

Because of this, the framework was gradually clarified into three conceptual parts. The first part consisted of company attributes, intended to describe the organisational context and readiness of the firm. The second part consisted of use-case attributes, intended to describe the requirements, value profile, and implementation characteristics of the specific use case. The third part consisted of fit attributes, which were not asked directly in interviews but instead defined as later derived measures capturing the compatibility between the company-side profile and the use-case-side profile. This clarification was important methodologically, because it affected how

the interview guide was written, how the data collection trackers were structured, and how the later dataset was intended to be represented.

3.4 Data collection

3.4.1 Interviews

The main empirical data collection method was semi-structured interviews with representatives from retail companies. A purely written survey was considered too limited for the purpose of the study. The main reason was that many of the variables of interest, such as data accessibility, change capacity, process ownership, operational risk, or privacy posture, are difficult to interpret in a comparable way without some degree of explanation and contextualization. In practice, respondents often needed short clarifications, examples, or boundary-setting in order to place their organisation or a specific use case on a meaningful scale.

For that reason, the interview format was chosen as a compromise between structure and flexibility. On the one hand, the model-oriented parts of the interview were deliberately standardised. The interviewer moved through a predefined sequence of questions and asked most of the attribute questions using simple categorical or ordinal response formats. On the other hand, the interview format still allowed the respondent to explain context when necessary, to clarify what a given use case meant in the specific company setting, and to bring up relevant details that would have been difficult to capture in a rigid form-only process.

Another practical reason for the interview-based design was that many of the relevant respondents were senior decision-makers. In such cases, a fully self-administered survey could easily have resulted in low response rates, partial completion, or inconsistent interpretation of the questions. A guided interview reduced that risk. At the same time, the design still allowed some degree of asynchronous preparation. In some cases, respondents could fill in large parts of the structured material in advance, after which the follow-up conversation focused more on clarification and open-ended reflection.

3.4.2 Sampling logic

The study used purposive sampling rather than random sampling. The target respondents were individuals with sufficiently broad insight into both the business and the technical side of AI initiatives within retail companies. In practice, this typically meant roles such as CTO, CIO, CDO, Head of AI, Head of Digital, Managing Director of IT, or similar senior functions. The exact title was not treated as decisive on its own. What mattered more was whether the respondent had enough visibility into which AI initiatives had been tested, how they had been prioritised, what had been difficult in practice, and how organisational conditions affected what was realistic to pursue.

The selection logic was therefore guided by relevance rather than representativeness in a statistical sense. The aim was not to estimate prevalence in the retail population, but rather to collect comparable and practically meaningful observations across companies. In several cases, identifying the right respondent required some degree of snowballing within the firm. If the initially contacted person was not the most suitable one, the conversation was instead used to identify another person closer to the company's AI work. This was treated as a normal part of the sampling process rather than as a deviation from it.

3.4.3 Contact and recruitment process

The recruitment process combined cold outreach, email contact, referrals, and follow-up contact. In practice, both email and phone were used. A brief project description was used to explain the purpose of the study and why the recipient's perspective was relevant. The communication was intentionally short and concrete, with emphasis on the practical nature of the project, the relevance for retail, and the limited time burden for the respondent. If the contacted person was not the right one, the respondent was asked whether another person in the organisation would be better suited for the interview. At the end of interviews, respondents were also commonly asked whether they knew other relevant people in the same sector who might be willing to participate.

This recruitment logic had two practical advantages. Firstly, it improved the chance of reaching people with real insight into how AI initiatives were handled in practice. Secondly, it reduced the friction of the process, since it allowed the project to use existing conversations to find better contacts instead of treating each company as a one-shot outreach attempt.

3.5 Interview instrument

3.5.1 General design principles

The interview instrument was designed to satisfy two requirements at the same time. Firstly, it needed to be structured enough to generate data that could later be compared across firms and use cases. Secondly, it needed to be practical enough to use in real conversations with senior respondents. This meant that most questions were kept short, concrete, and easy to answer approximately. In other words, the purpose was not to elicit exact measurements in the interview itself, but to collect sufficiently comparable judgements for later structuring and analysis.

A related design principle was that not all relevant information should be forced into one format. The instrument therefore combined three different types of input. The first type was structured company attributes. The second type was structured use-case attributes. The third type was open-ended qualitative responses used to capture pain points, practical barriers, and more general reflections. This allowed the study to preserve both comparability and contextual richness.

3.5.2 Attribute structure

The final attribute structure was developed iteratively from three sources: an original heuristic proposal by the thesis author, the early expert interviews, and later framework clarification work. The proposal already suggested a common attribute template and a set of organisational context variables. However, during the exploratory phase, this structure was expanded and reorganized. The result was a richer and more explicit attribute design in which the structured data was divided into company attributes, use-case attributes, and fit attributes.

This restructuring was important because it improved conceptual clarity. A company attribute was intended to capture a relatively stable organisational condition or preference. A use-case attribute was intended to capture something about the use case itself. A fit attribute, in contrast, was not directly collected in the interview at all, but defined conceptually as a derived measure of compatibility between the first two attribute groups. This distinction made the later representation logic easier to explain, and it also made the interview guide easier to use, since respondents were not asked to reason directly about abstract compatibility scores.

3.5.3 Company attributes

The company attributes were designed to describe relatively stable organisational conditions, priorities, and constraints that shape which AI use cases are realistic for a given firm. These attributes were collected once per company rather than once per use case. The underlying idea was that the same company context should later be combined with several use-case profiles, which means that company-side information needed to be defined in a way that was sufficiently stable across the rows belonging to the same organisation.

In practice, the company-side attributes covered strategic orientation, structural context, readiness and capability, and more general organisational posture. Most of them were collected directly in the interview, typically using short categorical or ordinal answer formats. One exception was size class, which could be filled in manually.

The complete company-side attribute set used in the study was the following.

1. **Primary AI goal.** This attribute captured what the company was mainly trying to achieve with AI at the current stage. The available alternatives were *Cost / productivity*, *Service / availability*, *Growth / customer experience*, and *Working capital*. *Cost / productivity* meant that AI was mainly seen as a way to save time, reduce manual work, or lower operating cost. *Service / availability* meant that AI was mainly seen as a way to improve service level, availability, responsiveness, or reliability. *Growth / customer experience* meant that AI was mainly seen as a way to improve revenue, conversion, personalization, or customer interaction. *Working capital* meant that AI was mainly seen as a

way to optimize inventory, reduce tied-up capital, or strengthen control. This attribute was collected in the interview and treated as a categorical variable.

2. **Size class.** This attribute was used as a simple size indicator for the company. The available alternatives were *Small*, *Mid*, *Large*, and *Enterprise*. In order to make this categorisation more concrete and more consistent across companies, the classes were tied to annual revenue intervals. *Small* referred to companies with annual revenue below SEK 250 million. *Mid* referred to companies with annual revenue between SEK 250 million and SEK 1 billion. *Large* referred to companies with annual revenue between SEK 1 billion and SEK 10 billion. *Enterprise* referred to companies with annual revenue exceeding SEK 10 billion. In practice, the purpose of this variable was not to produce a precise financial classification, but rather to capture a rough difference in organisational scale, structural complexity, and likely resource base. This attribute was handled manually rather than asked directly in the interview and was treated as an ordinal variable.

3. **Channel profile.** This attribute described the company's commercial operating model. The alternatives were *Store-led*, *Omnichannel*, and *E-commerce-led*. *Store-led* meant that physical stores dominated the business and operations from a revenue perspective. *Omnichannel* meant that both physical and digital channels mattered and required integration, as opposed to one channel accounting for a large majority of the revenue. *E-commerce-led* meant that the digital channel was dominant revenue-wise. This attribute was collected in the interview and treated as a categorical variable.

4. **Supply chain complexity.** This attribute described how complex the operational environment appeared to be. The alternatives were *Low*, *Medium*, and *High*. *Low* meant relatively straightforward flows, limited variety, and limited operational volatility. *Medium* meant some complexity in assortment, flows, geographies, or planning, but still at a manageable level. *High* meant many nodes, high assortment complexity, volatile demand, or complex network and process dependencies. This attribute was collected in the interview and treated as an ordinal variable.

5. **Data availability.** This attribute captured whether the company usually had the relevant data at all when it wanted to work on an AI use case. The alternatives were *Low*, *Medium*, and *High*. *Low* meant that important data was often missing or not systematically captured. *Medium* meant that much of the needed data existed, but not consistently across all areas. *High* meant that most relevant data typically existed and was available for analysis and use-case work. This attribute was collected in the interview and treated as an ordinal variable.

6. **Data quality.** This attribute described how trustworthy and usable the company's data usually was. The alternatives were *Low*, *Medium*, and *High*. *Low* meant frequent issues with completeness, consistency, correctness, or standardisation. *Medium* meant that the data was usable in many cases, but still required cleaning or caveats. *High* meant that the data was generally reliable, consistent, and strong enough to support operational decision-making. This attribute was collected in the interview and treated as an ordinal variable.

7. **Data accessibility.** This attribute captured how easy it was to actually access and combine the relevant data. The alternatives were *Low / fragmented*, *Medium*, and *High / integrated*. *High / integrated* meant that data was relatively easy to access and combine across relevant systems. *Medium* meant that data could be accessed, but that it often required manual work, coordination, or restructuring. *Low / fragmented* meant that data was scattered, siloed, difficult to extract, or slow to access. This attribute was collected in the interview and treated as an ordinal variable.

8. **IT competence.** This attribute described the company's ability to connect new solutions into existing systems and workflows. The alternatives were *Low*, *Medium*, and *High*. *Low* meant that integration work was usually slow, difficult, or heavily constrained by legacy systems. *Medium* meant that some integration capability existed, but was not consistently strong. *High* meant that the company was generally capable of integrating new tools and workflows into its stack without extreme friction. This attribute was collected in the interview and treated as an ordinal variable.

9. **Change capacity.** This attribute described the organisation's overall ability and willingness to absorb and implement change. In other words, it was intended to capture both whether the company had sufficient organisational bandwidth to drive new initiatives and whether the broader organisation was generally open to changing existing ways of working when needed. The alternatives were *Low*, *Medium*, and *High*. *Low* meant that the organisation had limited room for new initiatives and that resistance to changing established workflows was relatively high. *Medium* meant that some change could be handled and that parts of the organisation were open to adapting ways of working, but not without meaningful friction. *High* meant that the organisation had both strong capacity to drive new initiatives and a broad willingness across the organisation to adapt workflows where the value case was clear. This attribute was collected in the interview and treated as an ordinal variable.

10. **Process & governance maturity.** This attribute described how mature the organisation was in clearly owning important business processes and in making, reviewing, and following up on AI or data-related projects in a structured way. The alternatives were *Low*, *Medium*, and *High*. *Low* meant that ownership and

governance were unclear and inconsistent. *Medium* meant that some ownership and governance existed, but not consistently. *High* meant that clear ownership and structured governance were in place. This attribute was collected in the interview and treated as an ordinal variable.

11. **Privacy mindedness.** This attribute described how constrained the company was by personal data, sensitive data, and responsible usage concerns. The alternatives were *Low*, *Medium*, and *High*. *Low* meant that privacy concerns existed, but were not a major practical constraint in most use cases. *Medium* meant that privacy was important and often needed active consideration and approval. *High* meant that privacy was a strong constraint and materially shaped what kinds of use cases were realistic. This attribute was collected in the interview and treated as an ordinal variable.

12. **Compliance constraint level.** This attribute described how strongly regulatory, policy, legal, or audit-related requirements constrained the company. The alternatives were *Low*, *Medium*, and *High*. *Low* meant that security and compliance requirements rarely blocked or reshaped use-case choices significantly. *Medium* meant that such requirements mattered and often required active management. *High* meant that security and compliance constraints were central and could strongly limit feasible approaches. This attribute was collected in the interview and treated as an ordinal variable.

13. **Sourcing preference.** This attribute described what kind of solution approach the company was most comfortable with. The alternatives were *Off-the-shelf*, *Tailored external*, and *Internal*. *Off-the-shelf* meant that the company preferred standard tools and products whenever possible. *Tailored external* meant that the company preferred solutions adapted or tailored by external partners or vendors to their needs. *Internal* meant that the company was most comfortable with fully internally developed solutions. This attribute was collected in the interview and treated as a categorical variable.

14. **AI-dependence tolerance.** This attribute described how comfortable the company was with using AI in processes where operations would become materially dependent on it. The alternatives were *Low*, *Medium*, and *High*. *Low* meant that the company preferred AI mainly as support and did not want operations to depend heavily on it. *Medium* meant that it was comfortable with partial dependence if safeguards existed. *High* meant that it was comfortable making AI a central operational dependency where the value justified it. This attribute was collected in the interview and treated as an ordinal variable.

3.5.4 Use-case attributes

The use-case attributes were designed to describe the properties, requirements, and practical implications of each specific AI use case discussed in the interviews. These attributes were collected separately for each use case, which means that one company could contribute several use-case profiles. The intention was to capture not only what business problem a use case addressed, but also how demanding it was to implement, what kind of data or integration it required, and what kinds of risks or constraints were associated with it.

The complete use-case-side attribute set used in the study was the following.

1. **Use case family.** This attribute was used as a manual business-oriented grouping of the use case. The alternatives were *Planning & inventory*, *Logistics & operations*, *Commercial & customer*, and *Risk & control*. *Planning & inventory* included use cases such as forecasting, replenishment, allocation, inventory optimisation, and workforce planning. *Logistics & operations* included ETA, delay prevention, control tower logic, and operational exception handling. *Commercial & customer* included customer service, pricing, content support, store support, and customer-facing productivity. *Risk & control* included fraud, supplier risk, anomaly detection, monitoring, and guardrails. This attribute was assigned manually and treated as a categorical variable.

2. **Solution type.** This attribute grouped the use case based on the nature of the solution rather than the business function. The alternatives were *GenAI*, *Predictive / ML*, *Optimisation / recommendation*, and *Detection / anomaly / control*. *GenAI* referred to language-driven support such as copilots, summarisation, and content assistance. *Predictive / ML* referred to forecasting, risk scoring, or other prediction-heavy use cases using company data. *Optimisation / recommendation* referred to recommending actions, quantities, allocations, priorities, or decisions. *Detection / anomaly / control* referred to flagging unusual patterns, risks, exceptions, or deviations. This attribute was assigned manually and treated as a categorical variable.

3. **Delivery source.** This attribute described how the use case was sourced. The alternatives were *Off-the-shelf*, *Tailored external*, and *Internal*. *Off-the-shelf* meant a procured standard tool or product. *Tailored external* meant a solution tailor-made by external partners or vendors. *Internal* meant a fully internally developed solution. This attribute was collected in the interview and treated as a categorical variable.

4. **KPI family.** This attribute described the main business outcome the use case was intended to improve. The alternatives were *Cost / productivity*, *Service / availability*, *Growth / customer experience*, and *Working capital*. *Cost / productivity* referred to saving time, reducing manual work, or lowering cost.

Service / availability referred to improving service levels, availability, response quality, or reliability. *Growth / customer experience* referred to improving revenue, conversion, customer experience, or commercial performance. *Working capital* referred to improving inventory efficiency, lowering tied-up capital, or reducing losses and exposure. This attribute was collected in the interview and treated as a categorical variable.

5. **Time-to-value.** This attribute described how quickly the use case could reasonably start creating visible value. The alternatives were *Short*, *Medium* and *Long*. *Short* meant that meaningful value appeared relatively quickly. *Medium* meant that some setup time was needed before value became visible. *Long* meant that the use case required substantial preparation, build, or adoption before value would appear. This attribute was collected in the interview and treated as an ordinal variable.

6. **ROI metric clarity.** This attribute described how clearly the value of the use case could be measured. The alternatives were *Low*, *Medium*, and *High*. *Low* meant that value was hard to define or quantify cleanly. *Medium* meant that there were plausible metrics, but that value measurement still contained ambiguity. *High* meant that the use case had clear and concrete metrics that could reasonably be tracked. This attribute was collected in the interview and treated as an ordinal variable.

7. **Scope clarity.** This attribute described how concrete and bounded the use case was. The alternatives were *Very concrete*, *Moderately defined*, and *Broad / diffuse*. *Very concrete* meant that the use case was narrow, well-defined, and operationally specific. *Moderately defined* meant that the core use case was understood, but that some boundaries remained unclear. *Broad / diffuse* meant that the use case was still quite generic, wide, or loosely defined. This attribute was collected in the interview and treated as an ordinal variable.

8. **Organisational breadth relevance.** This attribute described how broad the relevance of the use case was across the organisation. The alternatives were *Narrow*, *Moderate*, and *Broad*. *Narrow* meant that the use case was mainly relevant to one team, function, or process. *Moderate* meant that it was relevant across several adjacent teams or functions. *Broad* meant that it was relevant to many parts of the organisation or clearly strategic across functions. This attribute was collected in the interview and treated as an ordinal variable.

9. **Data need.** This attribute described how demanding the use case was in terms of data prerequisites. The alternatives were *Low*, *Medium*, and *High*. *Low* meant that the use case could work with limited internal data or relatively simple data requirements. *Medium* meant that it needed meaningful data, but not at extreme scale or complexity. *High* meant that it was heavily dependent

on strong, well-structured data foundations. This attribute was collected in the interview and treated as an ordinal variable.

10. **Data recency requirement.** This attribute described what kind of data dependency the use case had. The alternatives were *Static*, *Historical*, *Real-time*, and *Multi-source*. *Static* referred to use cases that could work mainly from existing documents, policies, manuals, or stable content. *Historical* referred to use cases that mainly depended on historical transactional or master data. *Real-time* referred to use cases that needed relatively fresh operational data in order to be useful. *Multi-source* referred to use cases that required several data types or sources to work well. This attribute was collected in the interview and treated as a categorical variable.

11. **Integration burden.** This attribute described how much system and process integration the use case required. The alternatives were *Low*, *Medium*, and *High*. *Low* meant limited integration need, meaning that the use case could operate fairly standalone. *Medium* meant that some meaningful integration was needed, but at a manageable level. *High* meant that the use case required substantial integration into systems, workflows, or data pipelines. This attribute was collected in the interview and treated as an ordinal variable.

12. **Process change burden.** This attribute described how much change in the way people worked was required for the use case to create value. The alternatives were *Low*, *Medium*, and *High*. *Low* meant that the use case fitted relatively well into current workflows. *Medium* meant that it required some behaviour or process changes. *High* meant that it required substantial redesign of roles, routines, or decision flows. This attribute was collected in the interview and treated as an ordinal variable.

13. **Build effort.** This attribute described how much work it would take to get the use case to a usable pilot or implementation. The alternatives were *Low*, *Medium* and *High*. *Low* meant limited effort and a relatively quick path to standing the solution up. *Medium* meant meaningful effort, but still realistic within a normal initiative. *High* meant a large effort involving several teams and significant setup. This attribute was collected in the interview and treated as an ordinal variable.

14. **Maintenance cost.** This attribute described how heavy the ongoing cost and maintenance burden would be after launch. The alternatives were *Low*, *Medium*, and *High*. *Low* meant limited ongoing maintenance or support effort. *Medium* meant noticeable but manageable ongoing effort or cost. *High* meant significant ongoing support, monitoring, retraining, or service cost. This attribute was collected in the interview and treated as an ordinal variable.

15. **Human-in-the-loop need.** This attribute described how much human review or intervention was needed for the use case to be effective. The alternatives were *Low*, *Medium*, and *High*. *Low* meant that the use case could largely run with limited human review. *Medium* meant that human review was important in some cases or stages. *High* meant that strong human oversight was needed for the most important decisions or outputs. This attribute was collected in the interview and treated as an ordinal variable.

16. **Operational risk.** This attribute described the practical business disruption if the use case gave a bad output or recommendation. The alternatives were *Low*, *Medium*, and *High*. *Low* meant that wrong output would cause limited disruption and be relatively easy to absorb. *Medium* meant that wrong output would have meaningful operational consequences. *High* meant that wrong output could create serious disruption, cost, or service failure. This attribute was collected in the interview and treated as an ordinal variable.

17. **Privacy friendliness.** This attribute described how naturally compatible the use case was with privacy-sensitive environments. The alternatives were *Low*, *Medium*, and *High*. *Low* meant that the use case was clearly privacy-sensitive and harder to deploy in strict environments. *Medium* meant that some privacy sensitivity existed and needed active handling. *High* meant limited privacy concern, meaning that the use case was relatively privacy-friendly. This attribute was collected in the interview and treated as an ordinal variable.

18. **Compliance sensitivity.** This attribute described how much legal, regulatory, audit, or policy concerns mattered for the use case. The alternatives were *Low*, *Medium*, and *High*. *Low* meant limited compliance sensitivity. *Medium* meant some meaningful policy or legal sensitivity. *High* meant that compliance requirements were central and could strongly affect feasibility or design. This attribute was collected in the interview and treated as an ordinal variable.

19. **Required interpretability.** This attribute described how important it was that the logic behind outputs could be understood and explained. The alternatives were *Low*, *Medium*, and *High*. *Low* meant that a relatively black-box solution could be acceptable if it worked well enough. *Medium* meant that some interpretability was desirable, especially for stakeholder trust or review. *High* meant that strong explainability was required, such that users or governance stakeholders needed to understand why outputs were produced. This attribute was collected in the interview and treated as an ordinal variable.

20. **Reversibility of wrong decisions.** This attribute described how easy it would be to undo or correct a bad recommendation, output, or decision if the use case produced the wrong result. The alternatives were *Low*, *Medium*, and *High*. *High* meant that wrong decisions were relatively easy to reverse and

could usually be corrected with limited damage. *Medium* meant that some recovery was possible, but that reversing the decision would typically involve meaningful cost, delay, or operational friction. *Low* meant that wrong decisions were hard to reverse and could create consequences that were costly, slow, or difficult to undo. This attribute was collected in the interview and treated as an ordinal variable.

3.5.5 Fit attributes

The fit attributes were not asked directly in the interviews. Instead, they were defined as derived variables based on the combination of company-side and use-case-side attributes. The purpose of these variables was to represent the compatibility between a specific company context and a specific use case more explicitly. This was methodologically important because the study did not only aim to describe companies and use cases separately, but also to represent how well the two sides matched each other in practice.

At the conceptual level, this means that each row in the dataset could later be understood as the interaction between a company profile and a use-case profile, with the fit attributes expressing selected types of compatibility between them. In this section, the fit attributes are described conceptually, that is, in terms of what they were based on and what type of compatibility they were intended to capture. The exact operational computation logic used to derive these variables is described later in the technical implementation part of the methodology chapter.

In the current framework, the fit layer consisted of the following variables.

1. **Strategic fit.** This fit attribute described how well the company's primary AI goal matched the KPI family served by the use case. It was derived from *Primary AI goal* and *KPI family*. Conceptually, the attribute was intended to capture how directly the use case supported the company's current AI priorities. A high strategic fit meant that the use case was closely aligned with what the company was mainly trying to achieve with AI, whereas a low strategic fit meant that the use case was less clearly aligned with those priorities. The intuition was that, for example, a company focused mainly on working capital or risk would generally fit better with inventory optimisation or fraud-related use cases than with a customer-facing copilot.

2. **Data availability fit.** This fit attribute described how well the company's internal data availability matched the use case's data need. It was derived from *Data availability* and *Data need*. Conceptually, the attribute was intended to capture whether the company had enough relevant data for the use case to be realistic at all. A high data availability fit meant that the company's available data foundation matched the use case's level of data dependency relatively well, whereas a low data availability fit meant that the use case was likely

demanding more data than the company could reliably provide. The intuition was that a use case with high data demand would generally be a weaker fit in an organisation where relevant data was often missing or not systematically captured.

3. **Data quality fit.** This fit attribute described how well the company's internal data quality supported the use case's need for trustworthy inputs. It was derived from *Data quality* and *Data need*. Conceptually, the attribute was intended to capture whether the company's data was reliable enough for the use case to function in a meaningful way. A high data quality fit meant that the company's data quality was broadly sufficient relative to the demands of the use case, whereas a low data quality fit meant that the use case relied on a level of data trustworthiness that the company was less likely to have in place. The intuition was that even when data existed, poor completeness, consistency, or correctness could still make the overall fit weak.

4. **Data accessibility fit.** This fit attribute described how well the company's internal data accessibility matched the use case's data recency requirement. It was derived from *Data accessibility* and *Data recency requirement*. Conceptually, the attribute was intended to capture whether the company could actually access, combine, and operationalise the data in the way the use case required. A high data accessibility fit meant that the company's data environment was reasonably compatible with the timing and integration demands of the use case, whereas a low data accessibility fit meant that the relevant data was likely too fragmented, too slow to access, or too difficult to combine relative to the use case's needs. The intuition was that a use case needing near-real-time or multi-source data would fit worse if the company's data environment was highly siloed or operationally hard to work with. Because this compatibility relationship was somewhat more indirect than several of the other fit definitions, it was treated more cautiously in later technical feature-set experimentation.

5. **Capability fit.** This fit attribute described how well the company's internal technical capability matched the type of solution implied by the use case. It was derived from *IT competence* and *Solution type*. Conceptually, the attribute was intended to capture whether the organisation had the internal sophistication needed for the kind of AI solution under consideration. A high capability fit meant that the company's internal technical capability was broadly well matched to the nature of the use case, whereas a low capability fit meant that the use case likely required a level or type of capability that the company was less prepared for. The intuition was that predictive or optimisation-heavy use cases would often require stronger internal capability than simpler and more contained support-oriented solutions.

6. **Integration fit.** This fit attribute described how well the company's internal

IT competence matched the use case's integration burden. It was derived from *IT competence* and *Integration burden*. Conceptually, the attribute was intended to capture whether the company was able to connect the use case into existing systems and workflows without excessive friction. A high integration fit meant that the company's internal technical environment and integration capability were broadly sufficient for the use case, whereas a low integration fit meant that the use case required more systems integration than the company was likely to handle easily. The intuition was that a high-integration use case would usually be a much better fit where internal integration capability was strong and the systems landscape was manageable.

7. **Change fit.** This fit attribute described how well the company's ability and willingness to absorb change matched the use case's process change burden. It was derived from *Change capacity* and *Process change burden*. Conceptually, the attribute was intended to capture whether the organisation could realistically absorb the operational and behavioural change required by the use case. A high change fit meant that the company was broadly capable of implementing the required changes and that the use case was not asking for more adaptation than the organisation could absorb, whereas a low change fit meant that the use case likely required a degree of process or workflow change that the organisation would struggle with. The intuition was that even a technically promising use case could still be a poor candidate if the organisation lacked the broader readiness to adopt new ways of working.

8. **Governance fit.** This fit attribute described how well the company's governance maturity matched the use case's need for cross-functional ownership, coordination, and follow-up. It was derived from *Process & governance maturity*, *Organisational breadth relevance*, and *Operational risk*. Conceptually, the attribute was intended to capture whether the organisation had a sufficiently clear and structured governance environment for the use case to be managed responsibly. A high governance fit meant that the company's ownership structures and governance practices were broadly suitable relative to the breadth and risk profile of the use case, whereas a low governance fit meant that the use case likely required more structured ownership and follow-up than the company currently had in place. The intuition was that use cases touching many functions or carrying higher operational risk often required stronger governance fit in order to work well in practice.

9. **Operational dependence fit.** This fit attribute described how well the company's tolerance for depending on AI matched the operational criticality of the use case. It was derived from *AI-dependence tolerance*, *Operational risk*, and *Human-in-the-loop need*. Conceptually, the attribute was intended to capture whether the company was comfortable letting the use case play a meaningful role in operational decision-making. A high operational dependence

fit meant that the organisation's posture towards AI dependence was broadly compatible with the use case's operational role and safeguard needs, whereas a low operational dependence fit meant that the use case implied a stronger operational dependency on AI than the company was likely to accept comfortably. The intuition was that some higher-risk use cases would be a weaker fit if the company preferred AI mainly as support rather than as a central operational dependency.

10. **Sourcing fit.** This fit attribute described how well the company's sourcing preference matched the delivery mode of the use case. It was derived from *Sourcing preference* and *Delivery source*. Conceptually, the attribute was intended to capture whether the company's preferred way of acquiring or building solutions was well aligned with the practical sourcing logic of the use case. A high sourcing fit meant that the use case could plausibly be pursued in a way that matched the company's sourcing posture, whereas a low sourcing fit meant that the use case was more likely to require a sourcing approach that the company was less comfortable with. The intuition was that a company strongly preferring off-the-shelf solutions could be a poorer fit for use cases that in practice required internal build or heavily tailored external delivery.

11. **Privacy fit.** This fit attribute described how well the company's privacy posture matched the privacy sensitivity of the use case. It was derived from *Privacy mindedness* and *Privacy friendliness*. Conceptually, the attribute was intended to capture whether the company's level of caution around personal or sensitive data was well aligned with the privacy profile of the use case. A high privacy fit meant that the use case's privacy demands were broadly compatible with the organisation's privacy posture, whereas a low privacy fit meant that the use case was more privacy-sensitive than the company was likely to be comfortable handling easily. The intuition was that privacy-sensitive use cases would fit worse in contexts where privacy considerations strongly shaped what was seen as realistic.

12. **Compliance fit.** This fit attribute described how well the company's compliance constraints matched the compliance sensitivity of the use case. It was derived from *Compliance constraint level* and *Compliance sensitivity*. Conceptually, the attribute was intended to capture whether the company's legal, policy, audit, or regulatory environment was well aligned with the demands implied by the use case. A high compliance fit meant that the use case's compliance demands were broadly manageable within the company's governance and control environment, whereas a low compliance fit meant that the use case was likely to be more constrained by legal or compliance-related concerns. The intuition was that highly compliance-sensitive use cases were generally easier to pursue in environments where governance and risk handling were sufficiently mature.

Overall, these fit attributes were intended to represent where the actual matching logic happened. The company attributes and use-case attributes described the two sides separately, while the fit attributes described selected ways in which those two sides could align or misalign in practice.

A practical distinction was also made between attributes directly scored with the respondent and classification variables later standardised by the researcher for comparability. In particular, use-case family and solution type were treated as structured classification attributes rather than as free respondent labels. This reduced variation in naming and made cross-company comparison more analytically consistent.

3.5.6 Open-ended questions

The interview guide also contained an open-ended section. This section was placed near the end of the interview and used for broader reflection rather than scoring. The main purpose of these questions was to capture issues that are important in practice but difficult to fully represent through short categorical or ordinal answers alone.

These open-ended responses were considered methodologically important, but they were not treated as the main structured model inputs. Rather, their role was primarily to support contextual understanding, later qualitative synthesis, possible archetype development, and eventually playbook-oriented interpretation.

More concretely, the open-ended part of the interview drew from the following question set:

1. **What are the main barriers and difficulties you experience when it comes to implementing AI and making use of AI opportunities more broadly?**

2. **What types of problems or workflows within your retail business do you think are best suited for AI, and which are less suited for AI?**

3. **What would you need to have more firmly in place internally in order to succeed better with additional AI use cases going forward?**

4. **At the current stage, do you think companies should focus on testing many different AI use cases at the proof-of-concept level, or should they focus on a smaller number of use cases and make sure to take them from proof of concept to deeper implementation? In other words, is it better to go broad or deep?**

5. **If you look at your organisation today, what types of AI initiatives feel most realistic for you to succeed with over the next 12–24 months?**

6. **What are the three most important business problems or bottlenecks where AI could help in your organisation?**

7. **When you identify strong AI opportunities, where do they usually come from – strategy, operational pain points, or somewhere else?**

8. **What are the main operational or business problems that the organisation is dealing with right now?**

9. **Given where the company stands today, what types of AI initiatives feel realistic to start with, and what should most likely wait?**

It should be noted, however, that not all of these questions were asked during every interview. In practice, the open-ended section was used somewhat selectively, partly because interview time varied and partly because some questions overlapped in the type of reflection they elicited. As a result, only a subset of the questions was usually used in each interview, most often questions 1–3 together with one or two additional questions depending on how the discussion developed and which themes appeared most relevant for the specific respondent.

Taken together, these questions were intended to complement the structured attribute-based sections by eliciting broader managerial judgement about practical AI adoption. In practice, they helped capture not only what respondents believed about specific use cases, but also how they reasoned about organisational readiness, prioritisation logic, and the broader conditions for successful AI work in retail.

3.6 Interview procedure

3.6.1 Interview flow

Although the interviews were semi-structured, they followed a fairly consistent sequence. First, the interview began with a short introduction to the study, including an outline of the interview agenda. Second, the company block was completed (see company attributes), meaning that the respondent was asked about the broader organisational context and the firm's current AI-related goals, capabilities, and constraints. Third, the respondent was asked to identify relevant AI use cases that the company had tested, piloted, implemented, or otherwise worked with. Fourth, one or more selected use cases were then discussed in a repeated structured use-case module (see use case attributes). Fifth, the respondent was asked to give a direct impact assessment for each discussed use case. Finally, the interview moved into a more open-ended reflection block (see open ended questions section) and ended with a short closing discussion, often including possible referrals to other relevant respondents.

This structure was important for two reasons. Firstly, it made the interviews easier to

manage in practice, since the respondent was guided through one clear sequence rather than jumping between different levels of abstraction. Secondly, it supported later data structuring, because it mirrored the intended distinction between company-level information, row-level use-case information, labels, and open-ended context.

3.6.2 Structured scales

The model-oriented parts of the interview relied heavily on short structured questions with limited response alternatives. In practice, many questions used low/medium/high-style scales, while some used four-step ordinal scales and others used categorical alternatives - as covered in the attribute section. This was a deliberate design choice. A more fine-grained instrument could in principle have produced richer variation, but it would also have increased respondent burden and made the interviews less robust. Since many respondents were senior professionals with limited time, the practical goal was instead to collect judgements that were coarse enough to answer quickly but still informative enough to be compared across firms.

A related interviewer principle was that the respondent should not be forced to over-justify every rating in the structured blocks. Instead, the structured parts were framed as short assessment questions, often intended to be answerable in roughly 15–30 seconds each, while the open-ended section later in the interview created space for broader reasoning. In practice, this separation between short structured scoring and later reflection made the interviews easier to control and helped preserve comparability across cases.

For most low/medium/high-style structured attributes, the coding logic was centred on the ordinal values 1, 2, and 3. In addition, half-step values were allowed when the respondent's assessment was judged to fall between two neighbouring levels. More specifically, 1.5 was used to represent low-to-medium and 2.5 to represent medium-to-high. No other decimal values were intended for this type of attribute.

A further distinction was made between single-select and multi-select categorical attributes. Primary AI goal, KPI family, sourcing preference, and solution type were allowed to contain several selected categories when appropriate and were therefore treated as multi-select attributes. In contrast, channel profile, delivery source, and use-case family were treated as single-select attributes.

3.6.3 Note capture

The structured parts of the interview were documented using dedicated Excel trackers designed specifically for the project. Separate trackers were maintained for company attributes, use-case attributes, and open-ended questions. The practical logic of the trackers was intentionally human-friendly rather than model-ready. In the company tracker, one column represented one company. In the use-case tracker, one column represented one use-case datapoint. In the open-ended tracker, one column represented one interview. Each structured attribute was typically represented by

a value row and an associated comment row. Metadata rows were also included at the top in order to preserve basic contextual information about the interview, respondent, and company.

This design had two advantages. Firstly, it made note-taking during live interviews manageable. Secondly, it preserved explanatory comments alongside the structured values, which was useful for quality assurance and later interpretation. At the same time, the design also meant that the raw interview material could not be used directly as analysis input without later restructuring. This was accepted as a practical trade-off, because the priority at the data-collection stage was reliable documentation rather than technical elegance.

3.7 Data representation

3.7.1 Unit of analysis

The basic observation unit of the study was one organisation × one use-case pair. This means that a single company could contribute multiple rows if several use cases were discussed. Company-level attributes were then repeated across the rows belonging to that company, while use-case attributes varied across rows depending on the characteristics of the specific initiative being discussed.

This unit of analysis was chosen because it reflects the actual decision problem the thesis is trying to address. The practical question is not whether a company is generally "good at AI," but rather whether a specific use case appears suitable, valuable, and realistic for a specific company context. In other words, the row was defined around the interaction between a company and a use case, not around the company alone and not around the use case alone.

3.7.2 Structured representation

At the structured data level, each organisation–use-case observation consisted of three main parts. The first part was the company profile, that is, the set of company attributes describing the organisational setting. The second part was the use-case profile, that is, the set of attributes describing the selected use case. The third part consisted of labels, collected separately from the descriptive attributes.

In addition to these directly collected components, a third conceptual attribute family was defined for later use, namely fit attributes. These were not asked directly in the interview. Instead, they were defined as measures of how well the company-side profile matched the use-case-side profile. This was one of the main conceptual refinements of the project compared with the earlier flatter framing.

3.7.3 Label logic

For each selected use case, an impact-oriented label was collected separately from the descriptive attributes. In the interview guide, this was asked as the respondent's

assessment of how large the business impact of the use case was to the company. The respondent was asked to respond on a scale of 0 - 10 where 0 meant that the use case had no impact, 5 that it had meaningful yet moderate impact, and 10 meant it was completely transforming for the business. The important methodological decision was to keep this label separate from the input attributes. In other words, the structured attributes describe the company and the use case, while the impact assessment describes the expected outcome of that combination.

This separation was important because it reduced the risk of baking outcome information directly into the descriptive feature set. It also kept the data structure conceptually cleaner. A use case can be demanding, risky, or strategically well aligned without that automatically determining how large the expected impact will be. The label therefore remained a distinct part of the row structure.

3.7.4 Open-ended responses

The open-ended responses were attached to the interview material as a separate qualitative layer. They were not intended to replace the structured attributes, and they were not required in order to define the core organisation $\times$ use-case rows. Instead, they served as complementary material. In practice, this means that the structured attributes formed the main comparable core of the dataset, while the open-ended text added richer information about needs, barriers, readiness, and practical interpretation.

This distinction matters because it clarifies the logic of the dataset before the technical implementation starts. The dataset was not conceived as a single flat survey table with one homogeneous type of information. Rather, it was conceived as a structured core with an attached qualitative layer. That qualitative layer was expected to be especially relevant later for archetype interpretation and playbook-oriented outputs.

3.8 Ethical and methodological considerations

3.8.1 Confidentiality and respondent burden

The interviews were designed to avoid requesting confidential internal company data. The focus was instead on structured managerial judgements, descriptions of use cases, and broader reflections on AI-related priorities and constraints. This reduced the practical and ethical burden of participation, since respondents could contribute useful information without having to disclose sensitive operational details, proprietary datasets, or internal financial information at a granular level.

Respondent burden was also considered in the structure of the interview itself. The use of short structured questions, limited answer scales, and a clear sequence was intended to make participation feasible even for senior respondents with limited time. In addition, the possibility of filling in parts of the structured material in advance made the process more flexible when needed.

3.8.2 Consistency, comparability, and subjectivity

A clear methodological challenge in this kind of study is that many of the collected judgements are subjective. Different respondents may interpret scales somewhat differently, and even within one interview, there is a risk that some ratings are influenced by uncertainty, recency effects, or local organisational politics. This does not make the data unusable, but it does mean that comparability must be treated as something to actively support rather than simply assume.

Several design choices were therefore used to improve consistency. The interview guide relied on clear definitions, limited scales, repeated structure, and examples when necessary. The interviewer also had the possibility to clarify the intended meaning of a question if the respondent seemed uncertain. In addition, comment fields were preserved alongside the structured values so that later ambiguities could be revisited rather than silently ignored.

3.8.3 Sampling limitations

The sampling approach also implies clear limitations. Since the respondents were selected purposively and mainly through targeted outreach and referrals, the resulting material cannot be interpreted as a statistically representative sample of the retail sector. The strength of the design lies elsewhere. More specifically, the study aims to build a practically grounded comparative dataset rather than estimate population-level frequencies.

This means that the empirical material should be interpreted primarily as structured expert-oriented evidence from relevant cases. In practice, that is sufficient for the design purpose of the study, namely to construct and later evaluate a practical decision framework. However, it also means that strong claims about the retail population as a whole should be made carefully.

3.9 Technical implementation

3.9.1 Purpose of the technical module

The empirical design described above created the raw material for the technical part of the study, but it did not in itself create a machine-readable analytical system. Therefore, a separate technical module was implemented in order to transform the structured interview material into a reproducible analytical pipeline. In practice, this meant that the technical implementation had to do several things in sequence. Firstly, it had to read and normalise the raw workbook structure. Secondly, it had to construct row-level organisation $\times$ use-case observations that could be used consistently across experiments. Thirdly, it had to generate machine-readable feature representations for supervised modelling. Fourthly, it had to support multiple forms of quantitative analysis, namely row-level impact prediction, coefficient-based interpretability analysis, and cluster-level playbook-oriented strength estimation.

The technical module was therefore not built as a single model script, but rather as a layered framework. This is important methodologically, because different analytical questions in the project required different technical representations. For example, the row-level impact model required encoded features and grouped validation logic, while the later cluster-strength module required a cross-product pair universe, configurable grouping strategies, and explicit handling of missing company–use-case combinations. In other words, the technical implementation was designed as a reusable analytical backbone rather than as one narrow prediction routine.

3.9.2 Dataset-pipeline relation

The technical implementation starts from the same observation logic that motivated the data collection design, namely that the relevant unit is one organisation × one use-case pair. However, the technical pipeline does not work directly on the raw interview workbook. Instead, it introduces a sequence of transformations that gradually make the data more standardised, more explicit, and more suitable for comparison. Concretely, the pipeline can be described as:

raw interview workbook → normalised tables → merged row dataset → encoded feature sets → impact-model experiments → interpretability analysis → cluster-strength estimation

This ordering matters. The later modelling and clustering results depend strongly on the assumptions made in the earlier stages, for example how missing values are represented, how multi-select categorical attributes are normalised, how fit attributes are computed, and which feature subsets are included in a given experiment. Therefore, the technical implementation section is not only about code. It is also about formalizing the analytical assumptions that connect the interview material to the later quantitative outputs.

3.10 Analytical dataset construction

3.10.1 Workbook parsing and normalisation

The raw interview material was maintained in a multi-sheet Excel workbook. Since the active environment was not built around a heavy spreadsheet stack, a lightweight workbook reader was implemented that reads the `.xlsx` file directly through its underlying XML structure. In practice, this means that the workbook is treated as a zipped XML container rather than being loaded through a full-featured spreadsheet package. This choice was mainly practical, but it also had one methodological advantage, namely that the pipeline became explicit about the workbook structure it assumes instead of relying on hidden spreadsheet-library defaults.

The workbook logic itself was governed by the central schema definition. More specifically, the file `config/schema.json` defined sheet names, layout assumptions, attribute labels, allowed codes, scale definitions, and which labels or attributes were

considered active. The ETL layer then used this schema as the canonical source of truth when normalising values. This meant that the same structured definitions could later be reused by the feature encoder and the downstream analytical modules. In practice, this reduced the risk that one stage would silently reinterpret an attribute differently from another stage.

The normalised workbook stage wrote four main tables: `companies.csv`, `usecases.csv`, `labels.csv`, and `open_ended.csv`. In addition, an audit file was produced to preserve information about parsing issues, unmatched company names, invalid categorical values, values outside allowed scales, and similar data-quality problems. This is worth noting, because the ETL layer was designed not only to produce cleaned outputs, but also to preserve a trace of where ambiguity or inconsistency entered the data.

3.10.2 Normalisation and coding

During normalisation, different attribute types were handled differently. Numeric and ordinal attributes were parsed as numbers whenever possible. Categorical single-select variables were normalised to one internal code. Multi-select categorical variables were normalised to a pipe-separated internal representation such as `code_a|code_b`. In other words, the raw interview workbook remained human-friendly, while the normalised tables became explicit about the internal coding that later stages needed.

The half-step logic introduced earlier in the methodology became active here as well. For low/medium/high-style numeric-ordinal variables, the ETL layer accepted 1, 1.5, 2, 2.5, and 3 as valid values. This means that the methodological decision to permit low-to-medium and medium-to-high assessments was not merely descriptive in the interview design; it was also implemented formally in the data-processing pipeline.

Another important normalisation step concerned company identity matching between sheets. The company sheet and the use-case sheet were not structurally linked in the raw workbook. Therefore, the ETL layer constructed normalised company identifiers and then matched use-case rows back to company rows through cleaned company-name matching, with a constrained fuzzy fallback when exact standardised matching was not possible. This is not a perfect solution, but in practice it was a necessary step in order to transform separate human-maintained sheets into one coherent analytical dataset.

3.10.3 Organisation–use-case rows

After normalisation, the company, use-case, and label tables were merged into the row-level analytical dataset. Let c_i denote company-level information for company i, let u_j denote use-case-level information for use case j, and let y_{ij} denote the impact label attached to the original discussed pair when such a label exists. The merged row dataset was then built as a collection of records of the form

$$x_{ij} = (c_i, u_j, y_{ij}),$$

where the company attributes and use-case attributes are kept as separate structured blocks and the impact label remains a distinct outcome field.

In implementation terms, this merge stage used the normalised `companies.csv`, `usecases.csv`, and `labels.csv` tables. Each use-case row was linked to its originating company and to its label row. If the company identifier was missing, unmatched, or if the label row could not be found, the issue was logged explicitly. This means that the merged row dataset was not treated as a trivial flattening step. It was treated as another stage where structural inconsistencies could occur and therefore had to be made visible.

3.10.4 Sparse-row handling

An important methodological choice was made regarding sparse use-case rows. The ETL layer preserved sparse use cases rather than dropping them immediately. In other words, the normalised and merged datasets were designed to remain as complete as possible. However, a general candidate flag was also computed in ETL in order to support later experiment filtering. Concretely, a use case was marked as a general modelling candidate if it had a valid active impact label and at least seven filled structured use-case attributes. This threshold was implemented as a broad practical rule rather than as a theoretically optimal cutoff.

The reason for separating preservation from experiment filtering is that different later modules require different tolerances for sparsity. The baseline impact model benefits from somewhat stricter row quality, because row-level supervised learning is highly sensitive to missingness and sample size. The cluster-strength module, on the other hand, can still make use of sparse rows in certain contexts, especially when the analytical goal is to preserve the existence of a use case within a grouping strategy rather than to fit a direct row-level predictor. Therefore, exclusion logic was intentionally staged rather than hard-coded once at the start.

3.11 Feature construction

3.11.1 Encoding of structured attributes

The next stage of the pipeline transformed the merged row dataset into encoded feature matrices. Let the merged row dataset contain n observations and let one feature-set definition select p encoded predictors. The encoded dataset for one experiment can then be written as

$$X \in \mathbb{R}^{n \times p}, \qquad y \in \mathbb{R}^n,$$

where each row of X corresponds to one organisation $\times$ use-case observation and y contains the impact labels.

The encoding logic was intentionally explicit. Numeric and ordinal attributes were represented as one feature column containing the numeric value. Categorical attributes were expanded into one-hot or multi-hot indicator columns, meaning that each allowed categorical code received its own binary feature column. In addition, each selected attribute also received a separate missingness indicator column of the form `feature__attribute__missing`. This means that missingness was not hidden. Instead, the pipeline allowed later models to distinguish between a genuinely low coded value and a value that was absent entirely.

This representation has some clear tradeoffs. On the positive side, it is simple, transparent, and compatible with both linear models and distance-based procedures. On the negative side, it can create fairly wide matrices, especially when many categorical attributes are expanded. However, for this project, that tradeoff was acceptable, because interpretability and explicitness were valued more than highly compressed learned representations.

3.11.2 Derivation of fit attributes

In addition to the directly collected company and use-case attributes, the implementation computed a separate layer of deterministic fit attributes. These attributes were not asked directly in the interview. Instead, they were derived from company-side readiness variables and use-case-side demand variables. The purpose was to encode selected forms of compatibility that could be analytically useful but would be cumbersome to collect directly as one more respondent judgement for every pair.

Several fit attributes used a common gap-based logic. In simplified form, if r denotes a company-side readiness score and d denotes a use-case-side demand score, a fit score $f(r, d)$ was computed from the difference $r - d$ and then mapped onto the same 1 to 3 style scale. The practical mapping was:

$$f(r,d) = \begin{cases} 3.0 & \text{if } r - d \geq 1.0 \\ 2.5 & \text{if } 0.5 \leq r - d < 1.0 \\ 2.0 & \text{if } 0.0 \leq r - d < 0.5 \\ 1.5 & \text{if } -0.5 \leq r - d < 0.0 \\ 1.0 & \text{if } r - d < -0.5 \end{cases}$$

This gap-based structure was used for several fit dimensions such as data availability fit, data quality fit, capability fit, integration fit, change fit, governance fit, and operational dependence fit. In these cases, a higher fit score means that the company's relevant readiness is at or above the practical demand implied by the use case, whereas a lower fit score means that the use case appears to demand more than the company is likely to support comfortably.

For some fit attributes, either the readiness side or the demand side was itself a derived quantity rather than a directly collected single attribute. For example, data accessibility fit compared company-side data accessibility to a demand implied by the use case's data recency requirement. Capability fit compared company IT competence to a demand implied by solution type. More concretely, generative AI and detection-oriented solution types were assigned demand score 2.0, predictive solution types were assigned 2.5, and optimisation-oriented solution types were assigned 3.0. Therefore, use cases involving prediction or optimisation were treated as requiring somewhat stronger internal technical capability than more contained generative or detection-oriented solution forms. Change fit first averaged change capacity and workflow change willingness into a composite readiness score and then compared that to process change burden.

Two fit attributes used a different rule structure. Strategic fit was derived from the multi-select attributes *Primary AI goal* and *KPI family*. Let G denote the set of normalised company-side AI-goal categories and K the set of normalised use-case KPI categories. If either set was missing, no fit value was assigned. If $G \cap K = \varnothing$, strategic fit was set to 1.0. If there was overlap and one set was fully contained in the other, strategic fit was set to 3.0. Otherwise, that is, in cases of partial but not full overlap, strategic fit was set to 2.5. Thus, this attribute did not measure a numeric gap, but rather categorical alignment between company priorities and use-case outcome orientation.

Sourcing fit also used a categorical alignment rule rather than a numeric gap. Let P denote the set of normalised company-side sourcing-preference categories and let s denote the single normalised delivery-source category of the use case. If either side was missing, no fit value was assigned. If $s \in P$, sourcing fit was set to 3.0; otherwise it was set to 1.0. In practice, this means that the use case received a high sourcing fit if its delivery mode was compatible with at least one of the company's accepted sourcing postures, and a low sourcing fit otherwise.

This fit layer should therefore be understood as a deterministic engineering layer built on top of the collected interview data. The fit attributes were not learned from the data. They were defined by explicit operational rules. The advantage is that the logic remains explainable and reproducible. The drawback is that the layer necessarily reflects analytical assumptions about which compatibility relations matter and how they should be discretized.

3.11.3 Alternative feature-set variants

The implementation did not assume that one single feature representation would be optimal for all experiments. Instead, named feature-set presets were defined in configuration and resolved into concrete attribute lists at runtime. This made it possible to compare, for example, fuller raw feature sets, reduced raw feature sets, raw-plus-fit variants, and later manually pruned or excluded variants without rewriting the encoding logic each time. The purpose of this subsection is to describe

that generation logic at a methodological level. The concrete feature-set variants taken forward in the reported experiments are presented later in the results chapter.

Methodologically, this matters because feature selection was treated as part of the experiment design rather than as one fixed hidden preprocessing choice. Each encoded output directory stored a feature manifest describing the selected conceptual attributes, excluded attributes, encoded feature columns, active label, and other audit information. Therefore, later modelling results could always be tied back to the exact effective feature space that produced them.

3.12 Baseline impact modelling

3.12.1 Prediction objective and label definition

The supervised prediction target was the respondent-assessed impact score for a given organisation × use-case pair. This target was treated as a continuous regression label. In other words, the baseline impact model was not formulated as a pure classification task. The main motivation for this was that the original data collection design used a graded 0–10 impact scale, and reducing that directly to a small number of classes would have thrown away potentially useful ordering information.

At the same time, the study did not only care about numeric regression error. A central practical question was whether the model could preserve useful prioritisation orderings between use cases within a company. Therefore, the modelling evaluation included both ordinary regression metrics and within-company ranking metrics.

3.12.2 Candidate model families and experimental design

The baseline modelling layer was implemented as a configurable experiment framework rather than as one fixed estimator. It included a mean dummy baseline, regularised linear models such as ridge, lasso, elastic net, and Bayesian ridge, a partial least squares regression model, and more flexible nonlinear families such as random forest, extra trees, gradient boosting, histogram-based gradient boosting, support vector regression, and multilayer perceptron regression.

The reason for including this range of models was not that all of them were equally likely to become the final preferred approach. Rather, the idea was to test whether the impact signal looked primarily linear and additive, or whether there were signs that interactions, threshold effects, or nonlinear structure might matter. This is especially relevant in a setting where the feature space contains a mix of ordinal assessments, one-hot categorical expansions, multi-select indicators, and derived fit attributes. A purely linear model is transparent and often quite defensible in a small-data setting, but it can also miss interaction structure. Therefore, the more advanced model families were included as a methodological comparison layer, even though they are less transparent and, in a small-sample setting, also more fragile.

The practical baseline philosophy of the project still remained conservative. That

is, the baseline layer was intended to be serious and useful, but also transparent and explainable enough to serve as a reference point. Therefore, strong emphasis was placed on simple preprocessing, grouped validation, and models that can be interpreted reasonably, especially in the earlier comparison rounds.

3.12.3 Preprocessing inside the modelling pipeline

Each model was wrapped in a pipeline. Missing encoded feature values were imputed with a constant value of 0.0, and this was combined with the explicit missingness flag columns described earlier. For models that are scale-sensitive, such as ridge, lasso, elastic net, Bayesian ridge, support vector regression, multilayer perceptron regression, and partial least squares regression, standardisation was applied after imputation. For tree-based models, standardisation was not required.

This preprocessing choice was intentionally simple. A more advanced imputation strategy could in principle be used, but here the goal was to avoid introducing another opaque modelling layer before it had become clear that such complexity was justified. Since each attribute also had a missingness indicator, the model was not forced to treat an imputed zero as if it were a real substantive zero in the same way a simpler pipeline without missingness flags would have done.

3.12.4 Grouped nested cross-validation and hyperparameter selection

The primary evaluation logic used grouped nested cross-validation at company level. Let $g(i)$ denote the company identity associated with row i. The outer split was then constructed so that all rows with the same company identity remained in the same fold. This means that the evaluation was designed to estimate generalisation to unseen companies rather than to new rows from companies already seen during training.

More concretely, the outer loop used grouped K-fold splitting with the number of folds constrained by the number of distinct companies. Inside each outer-training split, an inner grouped cross-validation loop was used for hyperparameter selection whenever the model family had a non-empty hyperparameter grid. Therefore, the pipeline did not choose hyperparameters on the same held-out companies that were later used for honest performance reporting.

This distinction is important. The main reported row-level performance metrics were based on pooled out-of-fold predictions from the outer loop. In other words, the summary metrics were not training scores, and they were not simply taken from one final refitted model. They were based on predictions made for companies that were held out from the corresponding training folds.

3.12.5 Evaluation metrics and output artefacts

For the row-level impact model, three main groups of evaluation information were produced. Firstly, standard regression metrics such as mean absolute error and root mean squared error were computed on pooled out-of-fold predictions. Secondly, a supplementary global Spearman correlation was computed across all pooled row-level predictions in order to measure monotonic agreement at a broader level. Thirdly, within-company prioritisation metrics were computed, including mean company Spearman, median company Spearman, and a top-1 hit-rate style metric indicating how often the predicted top use case within a company overlapped with the true top use case.

This metric mix was chosen because exact numeric prediction and within-company prioritisation are not the same thing. A model can have moderate regression error but still preserve useful within-company ordering, and the reverse can also happen. Since the intended business use later concerns prioritisation support, this broader metric view was considered more relevant than regression error alone.

In addition, a final full-data refit was produced after evaluation for each experiment so that a final model artefact could be stored. This final refit is methodologically distinct from the honest evaluation results. It exists in order to preserve a trained model after comparison, not in order to replace the outer-fold evaluation as the reported estimate of generalisation.

3.13 Impact model interpretability

3.13.1 Motivation for interpretability analysis

After the baseline impact-model comparison, a separate interpretability layer was used for selected model and feature-set combinations. The purpose of this layer was not to decide predictive performance by itself, but rather to inspect what kinds of structured attributes the models appeared to rely on. This was considered useful for two reasons. Firstly, it made it easier to assess whether the predictive signal looked conceptually plausible. Secondly, it provided one possible input for later rule-based adjustments and playbook interpretation.

The interpretability analysis was divided into three complementary parts. Linear models were inspected through coefficient extraction, because their fitted coefficients are directly available after training. Nonlinear tree and boosting models were inspected through SHAP-based explanations, because their behaviour cannot be summarised by one ordinary coefficient vector. Finally, grouped permutation importance was used as a robustness-oriented check for both the nonlinear models and selected support-vector regression models. This means that interpretability was not treated as one single method, but as a small toolkit chosen according to the model family being inspected.

3.13.2 Coefficient-based interpretation of linear models

For linear models exposing a coefficient vector, the saved fitted model was loaded and the coefficient vector was mapped back to the encoded feature columns stored in the feature manifest. At the most detailed level, this gives one coefficient per encoded feature column, for example separate encoded columns for internal delivery and off-the-shelf delivery inside the broader delivery-source attribute. However, the implementation also rolled these detailed encoded coefficients back up to the conceptual attribute level.

If attribute a expands into encoded columns $k \in K(a)$ with coefficients β_k, then two aggregate summaries were computed:

$$\text{attribute_abs_sum}(a) = \sum_{k \in K(a)} |\beta_k|,$$

$$\text{attribute_signed_sum}(a) = \sum_{k \in K(a)} \beta_k.$$

The first quantity measures total coefficient mass regardless of sign, meaning that it indicates how strongly the model seems to lean on the attribute overall. The second quantity keeps net direction, meaning that it indicates whether the attribute's encoded columns lean positive or negative on balance. In addition, the dominant encoded feature within each conceptual attribute was identified as the encoded column with the largest absolute coefficient.

3.13.3 SHAP-based interpretation of nonlinear models

For selected nonlinear tree and boosting models, SHAP-based interpretation was used. The purpose of this step was to obtain a local additive explanation of model predictions, where each encoded input feature receives a contribution value for each row. In simplified terms, for one row i with encoded features x_i, the model prediction can be written as an additive decomposition

$$f(x_i) \approx \phi_0 + \sum_{k=1}^{p} \phi_{ik},$$

where ϕ_0 is a baseline value and ϕ_{ik} is the SHAP contribution assigned to encoded feature k for row i. A positive ϕ_{ik} means that the feature pushes the prediction upward relative to the baseline for that row, while a negative value means that it pushes the prediction downward. In practice, the exact interpretation depends on the fitted model and the background distribution used by the SHAP explainer, so the values should be read as model explanations rather than causal effects.

As with the coefficient analysis, the SHAP outputs were aggregated from encoded feature level to conceptual attribute level. If attribute a expands into encoded

columns $k \in K(a)$, the main magnitude summary was computed as

$$\text{attribute_mean_abs_SHAP}(a) = \sum_{k \in K(a)} \frac{1}{n} \sum_{i=1}^{n} |\phi_{ik}|.$$

This quantity measures how much the model uses that conceptual attribute in absolute terms, regardless of whether the attribute pushes predictions upward or downward for individual rows. In addition, the implementation stored the mean signed SHAP contribution for each attribute and identified the dominant encoded feature within the attribute. Because ordinary mean signed SHAP values can average towards zero when an attribute has mixed effects across rows, an additional directional diagnostic was also computed. This diagnostic correlated each encoded feature value with its SHAP contribution and then averaged those correlations within each conceptual attribute, weighted by the encoded feature's mean absolute SHAP magnitude. This was used as a practical indication of whether higher encoded feature values tended to increase or decrease the model prediction.

3.13.4 Permutation-based robustness check

Grouped permutation importance was used as a second interpretability check. This method asks a slightly different question than SHAP. Instead of decomposing each prediction into additive feature contributions, it measures how much worse the fitted model becomes when the information in one conceptual attribute is disrupted. This gives a more direct indication of model reliance: if shuffling an attribute makes the model's predictions much worse, then the model was probably using that attribute in an important way.

The permutation was performed at conceptual attribute level rather than at individual encoded column level. For an attribute a with encoded columns $K(a)$, all columns in $K(a)$ were shuffled together across rows. The model was then used to predict on this permuted matrix, and the resulting mean absolute error was compared with the baseline full-data prediction error of the same fitted model. For one permutation repetition r, the importance value can be written as

$$\Delta_{a,r} = \text{MAE}\left(y, f(X^{\pi_r(a)})\right) - \text{MAE}\left(y, f(X)\right),$$

where $X^{\pi_r(a)}$ denotes the feature matrix where the encoded columns belonging to attribute a have been jointly permuted. The final reported importance for the attribute was the mean of $\Delta_{a,r}$ across repeated permutations. A larger positive value means that disrupting the attribute increased prediction error more strongly, and therefore that the fitted model relied more on that attribute.

It is worth noting that this permutation check was not used as a new model-evaluation result. The baseline error in this calculation is computed using the saved fitted model and the full encoded data used for inspection, so it should not be compared directly

with the out-of-fold evaluation metrics from the baseline model comparison. Instead, the permutation values were used internally within each fitted model as relative evidence of which conceptual attributes mattered most for that model's predictions.

3.13.5 Interpretive role and limitations

This interpretability layer should be treated carefully. It is most meaningful as an aid for understanding fitted model behaviour, not as a causal analysis of what truly drives AI impact in retail organisations. Even for linear models, coefficient interpretation is not the same thing as causal interpretation. Moreover, for one-hot or multi-hot categorical attributes, positive and negative encoded columns can partially cancel each other out in the signed aggregate. For nonlinear models, SHAP and permutation importance provide useful additional views, but they are still conditional on the fitted model, the encoded feature representation, and the relatively small empirical dataset.

Therefore, the interpretability layer was used mainly as a structured diagnostic aid. In practice, it helped identify attributes that appeared to matter repeatedly, attributes whose direction looked unstable, and attributes that might later be relevant when constructing or evaluating more explicit cluster-strength adjustment rules. This is important because the later adjustment logic should not be understood as learned end-to-end from the interpretability outputs. Rather, the interpretability outputs helped suggest candidate adjustment levers that could then be tested transparently in the cluster-strength module.

3.14 Cluster-strength analysis for playbook support

3.14.1 Motivation for moving from row-level impact to cluster-level guidance

The row-level impact model answers one question, namely how the impact of one specific organisation × use-case pair may be predicted from structured attributes. However, the playbook-oriented goal of the thesis required another type of analysis as well. More specifically, it required a way to reason about whether certain types of companies and certain types of use cases appear to fit well together at a more aggregated and interpretable level. This motivated the separate cluster-strength module.

The core idea was to aggregate organisation–use-case observations into edges between company groups and use-case groups. A cluster-strength edge should therefore be interpreted as an estimated mean impact for one company-group × use-case-group combination. This is a different object than the row-level prediction problem. It is closer to a structured compatibility or prioritisation signal for interpretable archetypes.

3.14.2 Grouping strategies and pair-universe logic

The cluster-strength pipeline starts by reading one company grouping CSV and one use-case grouping CSV. These grouping files assign each included company and each included use case to one named group. Methodologically, these groupings should not be understood as unsupervised clusters learned from the data. Instead, they were explicit taxonomies, either directly derived from standardised structured attributes or manually defined in order to support interpretable comparison between broader company contexts and broader use-case types.

3.14.2.1 Company grouping strategies

The company-side grouping strategies were intended to capture sector context, organisational profile, readiness level, and strategic posture:

1. **Retail industry type.** This strategy reflected broad differences in assortment structure, operating logic, customer interaction, and regulatory environment. The categories were:
   - grocery, food, and convenience retail
   - health, pharmacy, and beauty retail
   - fashion, sports, and apparel retail
   - home, DIY, and interior retail
   - consumer electronics and technology retail
   - marketplace and recommerce platforms
   - regulated beverage retail
   - automotive services
   - diversified or adjacent retail contexts

2. **Size class.** This strategy was intended to capture broad differences in organisational scale, resource base, and likely implementation capacity. It reused the same revenue-based classification as the underlying company attribute *Size class*. The categories were:
   - small: annual revenue below SEK 250 million
   - mid: annual revenue between SEK 250 million and SEK 1 billion
   - large: annual revenue between SEK 1 billion and SEK 10 billion
   - enterprise: annual revenue above SEK 10 billion

3. **Channel profile.** This strategy distinguished between predominantly physical retail, predominantly e-commerce, deliberately integrated omnichannel models, and cases where a reliable assignment could not be made from the available structured information. The categories were:
   - store
   - e-commerce
   - omnichannel

- missing

4. **Supply-chain complexity.** This strategy summarised how operationally demanding the firm's assortment and logistics environment appeared to be. The categories were:
   - low
   - medium
   - high
   - missing

5. **IT competence.** This strategy represented the firm's internal technical capability for developing, integrating, and governing AI-related solutions. The categories were:
   - low
   - medium
   - high
   - missing

6. **Change capacity.** This strategy represented how well the organisation appeared able to absorb and implement organisational change. The categories were:
   - low
   - medium
   - high
   - missing

7. **Primary AI goal.** This strategy retained the strategic direction firms were primarily trying to pursue with AI rather than collapsing all firms into only one generic objective. The base categories were:
   - cost
   - growth
   - service
   - working capital
   - observed multi-goal combinations of these priorities when firms simultaneously emphasised more than one strategic objective
   - missing

8. **Sourcing preference primary.** This strategy summarised the firm's dominant sourcing posture when acquiring or building AI-related solutions. The categories were:
   - internal
   - external
   - off-the-shelf

- unknown

9. **Data-readiness tier.** This derived strategy summarised the firm's data-side readiness by combining data availability, data quality, and data accessibility into one broader readiness tier. The categories were:
   - low
   - medium
   - high
   - unknown

10. **Execution-readiness tier.** This derived strategy summarised broader implementation readiness by combining IT competence, change capacity, and workflow-change willingness into one composite tier. The categories were:
    - low
    - medium
    - high
    - unknown

11. **Goal × data readiness.** This composite strategy represented strategic intent together with data-side maturity. The resulting categories were constructed by combining the firm's primary AI-goal profile with its data-readiness tier, for example:
    - cost with high data readiness
    - growth with high data readiness
    - cost and service with medium data readiness
    - other observed goal-profile and data-readiness combinations
    - combinations with missing strategic or readiness information

12. **Goal × execution readiness.** This composite strategy represented strategic intent together with broader implementation readiness. The resulting categories were constructed by combining the firm's primary AI-goal profile with its execution-readiness tier, for example:
    - cost with high execution readiness
    - growth with medium execution readiness
    - service-oriented goal combinations with high or medium execution readiness
    - other observed goal-profile and execution-readiness combinations
    - combinations with missing strategic or readiness information

3.14.2.2 Use-case grouping strategies

The use-case-side grouping strategies were intended to capture solution pattern, organisational ownership, implementation form, and business outcome orientation:

1. **AI solution pattern grouping strategy.** This taxonomy supported comparison between broader AI solution patterns rather than between individual use-case labels. The categories were:
    - Customer Support Agent
    - Internal Knowledge Assistant
    - Recommendation / Personalization
    - Forecasting / Inventory Planning
    - Pricing / Markdown Optimisation
    - Content / Translation / SEO Generation
    - Creative / Image Generation
    - Vision / Detection / Attribute Extraction
    - Coding / Productivity Tools
    - Risk / Compliance Screening
    - Research / Trend / Insight Agent
    - Finance Process Automation
    - Operations Assistance / Matching
    - Product Design / Fit Simulation
    - General / Unclear

2. **Business department owner.** This taxonomy grouped use cases by the business function most naturally owning or using them. The categories were:
    - Customer Support
    - Finance / Administration
    - General Cross-Functional
    - IT / Internal Enablement
    - Merchandising / Pricing
    - Product / Content / Creative
    - Risk / Legal / Compliance / Sustainability
    - Sales / Marketing / Loyalty / E-commerce
    - Store / Field / Service Operations
    - Strategy / Analytics / Business Development
    - Supply Chain / Logistics / Planning
    - Unclear

3. **Organisational support area.** This taxonomy grouped use cases by the broad organisational area they primarily supported. The categories were:
    - Customer Support
    - Sales / Marketing / Commerce
    - Supply Chain / Operations
    - Pricing / Merchandising
    - Product Content / Localization

- Creative / Design / Product Data
- Internal Enablement / IT
- Finance / Back Office
- Risk / Compliance / Sustainability
- Strategy / Research / Analytics
- General / Unclear

4. **Use-case family.** This grouping provided a compact high-level family taxonomy and was intentionally lower-cardinality than the broader semantic taxonomies. The categories were:
   - planning and inventory
   - commercial or customer-oriented use cases
   - logistics and operations
   - risk-oriented use cases
   - missing or unclear cases

5. **Solution type.** This strategy distinguished generative, predictive, optimisation-oriented, and hybrid solution forms. The categories were:
   - generative AI
   - predictive AI
   - optimisation
   - predictive + optimisation hybrids
   - generative + optimisation hybrids
   - generative + detection hybrids
   - missing

6. **Delivery source.** This strategy described the dominant delivery mode through which the use case was or would be implemented. The categories were:
   - internal
   - external
   - off-the-shelf
   - missing

7. **KPI family.** This strategy captured the type of business value the use case was intended to create. The base categories were:
   - cost
   - growth
   - service
   - working capital
   - observed combinations when one use case was described as targeting several types of business value at the same time

- missing

8. **Human-in-the-loop need.** This strategy reflected how dependent the use case was on continued human oversight or intervention. The categories were:
   - low
   - medium
   - high
   - missing

9. **Implementation-complexity tier.** This derived strategy summarised broader implementation difficulty in a compact way. It was computed as the mean of five use-case attributes: build effort, process change burden, integration burden, maintenance cost, and operational risk. The resulting score was then mapped to *High* if it was at least 2.5, *Medium* if it was at least 1.75 but below 2.5, and *Low* otherwise. Missing values led to an *Unknown* group. The categories were:
   - low
   - medium
   - high
   - unknown

10. **Time-to-value × build effort.** This composite strategy summarised expected payback speed together with implementation burden. The categories were:
    - fast time-to-value with light build effort
    - fast time-to-value with heavy build effort
    - slower time-to-value with light build effort
    - slower time-to-value with heavy build effort
    - missing

11. **Data need × recency requirement.** This composite strategy summarised how demanding a use case was from a data perspective in terms of both volume or richness and timeliness requirements. The categories were:
    - light data need with static data
    - light data need with historical data
    - light data need with real-time data
    - medium data need with historical data
    - medium data need with real-time data
    - medium data need with static data
    - high data need with historical data
    - high data need with real-time data
    - high data need with mixed recency requirements
    - high data need with uncertain recency requirements
    - missing

Taken together, these grouping strategies made it possible to compare company contexts and use-case archetypes at several levels of abstraction, ranging from direct attribute-level categories to broader semantic and composite taxonomies. This was methodologically important because the cluster-strength analysis did not only aim to estimate isolated row-level outcomes, but also to identify interpretable cross-case patterns that could later support managerial prioritisation and playbook-oriented guidance.

Given a chosen company grouping and a chosen use-case grouping, the module constructs a full pair universe. Let C be the set of selected companies and let U be the set of selected use cases that pass the chosen row filter. The pair universe is then the full cross-product

$$\mathcal{P} = C \times U.$$

This is an important design choice. Only a subset of these pairs correspond to originally observed interview rows with true impact labels. The rest are synthetic pairs in the sense that the company and use case exist in the dataset, but that particular combination was not observed as a labelled interview row.

For each synthetic pair (c, u), the company-side attributes are taken from company c, while the use-case-side attributes are taken from the original source use case u. Therefore, synthetic pairs preserve the target company context while reusing the structured description of an existing use case. This makes it possible to ask a question of the form: what if this type of use case were considered in this type of company context?

3.14.3 Observed-only edge-strength computation

In the simplest and rawest configuration, only true observed impacts are used. Suppose that one edge e corresponds to company group g_c and use-case group g_u. Let $\mathcal{O}_e$ be the set of observed company–use-case rows inside that edge with non-missing true impact. The observed-only edge strength is then

$$s_e^{\text{obs}} = \frac{1}{|\mathcal{O}_e|} \sum_{(c,u)\in\mathcal{O}_e} y_{cu},$$

provided that $|\mathcal{O}_e| > 0$.

In addition, the module calculates observed coverage:

$$\text{coverage}_e = \frac{|\mathcal{O}_e|}{|\mathcal{P}_e|},$$

where $\mathcal{P}_e$ is the full set of possible company–use-case pairs inside that edge. This coverage measure is important because two edges with similar mean impact can be

very different in how much direct observed support they actually have.

The observed-only mode is methodologically attractive because it is the cleanest version. It does not introduce any synthetic predicted impacts. At the same time, it is often sparse. Many interesting company-group × use-case-group combinations have low observed coverage or even zero observed rows. Therefore, the implementation also supports a more augmented mode.

3.14.4 Prototype-based completion of missing combinations

When prototype completion is enabled, the module fits a separate nearest-neighbour predictor on the observed row-level pairs in the chosen prototype feature space. The training set for this predictor contains only rows with real observed impacts. Each row is represented by the encoded feature columns of the chosen feature set. If a separate prototype feature-set directory is supplied, the neighbour search can be performed in a broader or different feature space than the main cluster-strength run itself uses for pair-universe reconstruction. This was implemented explicitly so that clustering and neighbour similarity need not always be tied to the exact same feature subset.

Let $z_i \in \mathbb{R}^p$ denote the encoded feature vector for observed row i, and let y_i denote its true impact. For one missing pair with feature vector z^*, the prototype module finds the k nearest observed rows under the configured distance metric, typically Euclidean distance after optional standardisation. Standardisation, when enabled, is fit on the observed training matrix and then applied both to the training rows and to the query row before neighbour search.

Two weighting modes are supported. In uniform weighting,

$$\hat{y}(z^*) = \frac{1}{k} \sum_{j \in N_k(z^*)} y_j.$$

In inverse-distance weighting,

$$\hat{y}(z^*) = \frac{\sum_{j \in N_k(z^*)} w_j y_j}{\sum_{j \in N_k(z^*)} w_j}, \qquad w_j = \frac{1}{\max(d(z^*, z_j), \varepsilon)},$$

unless one or more neighbours have effectively zero distance, in which case the prediction is taken as the mean of those exact or near-exact matches. In practice, this avoids numerical instability and gives exact matches special treatment.

An optional shrinkage step can then be applied:

$$\hat{y}_{\text{shrunk}} = (1 - \lambda)\hat{y} + \lambda \bar{y},$$

where $\bar{y}$ is the global mean observed impact in the prototype training data and λ is

a configured shrink weight. The reason for this option is practical. In sparse settings, a nearest-neighbour prediction can become quite sensitive to a small local set of rows. Shrinking slightly towards the global observed mean can reduce some of this instability, although it also makes the predictions less locally sharp.

It is important to note that the neighbour search is not limited to the current edge. Instead, the prototype predictor searches across the full observed training set in the chosen prototype feature space. This means that similarity is defined by encoded attribute proximity, not by prior membership in the same company-group $\times$ use-case-group cell.

3.14.5 Manual adjustments and smoothing

After prototype completion, optional manual adjustments can be applied. These adjustments are rule-based additive deltas on predicted missing values. A rule may refer to one contextual attribute and apply when that attribute equals a value, belongs to a set, contains a token, or satisfies a numeric inequality. If a missing pair prediction $\hat{y}$ matches one or more rules with deltas δ_r, then the adjusted prediction becomes

$$\hat{y}_{\text{adj}} = \hat{y} + \sum_{r \in R(z^*)} \delta_r,$$

after which an optional clipping interval can be applied. Methodologically, these manual adjustments should be understood as an explicit expert-rule layer rather than as part of the learned predictor. In the intended experimental logic, they were not meant to be arbitrary. Rather, they were intended to operationalise signals from the impact-model interpretability analysis. More specifically, attributes that repeatedly appeared important across coefficient, SHAP, or permutation-based diagnostics could be translated into candidate positive or negative adjustment rules, after which alternative adjustment settings could be tested across cluster-strength runs. The manual-adjustment layer therefore acted as a transparent way of transferring impact-model insight into cluster-strength experimentation, although this transfer remained heuristic rather than learned end-to-end. At the same time, manual adjustments are one of the more assumption-heavy parts of the framework and should therefore be interpreted cautiously.

After pair-level completion, the module decides per edge whether to use observed-only values or observed-plus-predicted values. This is controlled by an observed-coverage threshold. If coverage is above the threshold, the edge can remain observed-only. If it falls below the threshold, the edge can switch to a mode where completed missing pairs are allowed to contribute to the final edge mean. This makes the augmentation policy selective rather than all-or-nothing.

Finally, optional global smoothing can be applied at edge level. If an edge has raw final mean s_e based on n_e contributing rows and the chosen global target is μ, then

the smoothed edge strength becomes

$$\tilde{s}_e = \frac{n_e s_e + \alpha \mu}{n_e + \alpha},$$

where α is a configured pseudo-count. This is essentially a simple empirical-Bayes-style shrinkage towards a chosen global mean. The motivation is again practical: very small edges can otherwise look overly strong or weak due to a handful of rows.

3.14.6 Evaluation of grouping strategies and edge structure

The cluster-strength module also includes an evaluation layer. One part of this evaluation is a held-out full-pipeline check in which one observed pair at a time is masked, reconstructed through the same completion pipeline, and then compared to the true impact. This produces regression-style diagnostics both at completed-pair level and at edge-strength proxy level. The point is not that edge-strength analysis becomes a standard prediction benchmark, but rather that the completion logic can still be checked against observed rows in a disciplined way.

Another part of the evaluation summarises sparsity and separation structure. This includes, for example, counts of zero-observed edges, the share of edges with at least one, two, or three observed rows, and summary statistics of edge-strength distributions. This was included because a grouping strategy cannot be judged from one number alone. It also matters whether it produces interpretable edges, whether those edges are supported by actual observed rows, and whether there is enough spread between stronger and weaker edges to make the grouping useful for playbook reasoning.

3.14.7 Batch experimentation and configuration choices

The cluster-strength framework was deliberately built as a config-driven experimentation system. The main configurable dimensions included the chosen normalised input directory, the main encoded feature-set directory, the optional separate prototype feature-set directory, the company grouping file, the use-case grouping file, the row filter mode, the prototype settings, the manual-adjustment settings, the augmentation policy, the smoothing settings, and the evaluation toggle.

More specifically, the prototype predictor can be varied by turning it on or off, changing the number of neighbours, changing the distance metric, switching feature standardisation on or off, choosing uniform or inverse-distance weighting, and enabling or disabling shrinkage to the global mean. The manual-adjustment layer can be turned on or off and can also vary in which rules are active, which rows they are applied to, and how clipping is handled. The augmentation policy can change the observed-coverage threshold and can decide which edge mode is used above and below that threshold. The smoothing layer can be disabled entirely or configured through both a target mean and a pseudo-count strength.

In addition, a separate batch runner was implemented. The batch framework takes one base run configuration, one list of company grouping strategies, one list of use-case grouping strategies, and one list of run variants. It then generates concrete run configurations by combining these elements, applies shared and per-variant overrides, executes the runs, and writes batch-level summary tables and comparison plots. This made it possible to study many combinations of grouping strategies and configuration settings systematically without manually editing each configuration one by one.

3.15 Reproducibility and experiment management

3.15.1 Configuration as the primary control surface

Reproducibility in the project was handled primarily through explicit configuration files and stored run artefacts. The most important control files were the central schema definition, the feature-set definition file, individual cluster-strength run configurations, and batch configurations. This means that many analytical assumptions that might otherwise live implicitly inside scripts were instead exposed as structured configuration choices.

Methodologically, this serves two purposes. Firstly, it makes it easier to rerun an analysis on updated data without changing the core implementation logic. Secondly, it makes it easier to compare two runs and understand whether a changed result came from the data, the grouping strategy, the feature space, the model family, or one of the cluster-strength augmentation settings.

3.15.2 Stored manifests, audits, and run outputs

At each major technical stage, the pipeline writes both primary outputs and audit-oriented side artefacts. The ETL layer writes normalised tables and workbook-audit information. The fit layer writes a fit dataset and a fit-audit file. The encoder writes the encoded dataset together with a feature manifest and an encoding-audit file. The impact-model layer writes summary CSV files, detailed JSON files, pooled out-of-fold prediction tables, calibration summaries, and final refitted model artefacts. The interpretability layer writes detailed coefficient tables, SHAP summaries, permutation-importance tables, and aggregate heatmaps. The cluster-strength layer writes edge tables, pair-level completion tables, run summaries, resolved configurations, evaluation summaries, and edge-distribution plots. The batch layer writes generated run configurations, batch summaries, comparison plots, and a batch report.

The reason this matters is that the implementation was designed not only to produce final numbers, but also to preserve the intermediate context needed for later interpretation and report writing. In practice, this made it easier to reconstruct why a given experiment behaved the way it did and which assumptions were active in that run.

3.16 Technical limitations and boundary conditions

Several limitations should be stated clearly. Firstly, the row-level supervised dataset is small, and the labels are based on structured managerial judgement rather than on direct measured operational outcomes. Therefore, all row-level predictive results must be interpreted carefully. Secondly, the fit-attribute layer is deterministic and assumption-driven. It is useful as an explicit compatibility representation, but it is not a discovered truth from the data itself.

Thirdly, the encoded feature space is simple and transparent, but it also imposes a particular geometry on the data. For example, nearest-neighbour similarity in the prototype predictor is computed in the encoded numeric space, which means that one-hot or multi-hot categorical differences contribute through ordinary geometric distance rather than through a specially learned similarity function. This is defensible and easy to explain, but it is still an assumption.

Fourthly, grouping strategies in the cluster-strength module are partly researcher-defined. This is unavoidable if the aim is to create interpretable company and use-case archetypes, but it also means that the resulting edge structure depends in part on how the taxonomies are defined. Fifthly, manual adjustments and smoothing can improve practical usability in sparse settings, but they also move the framework further away from purely observed evidence. Therefore, the most playbook-facing conclusions should be grounded most strongly in the cleaner and more directly supported parts of the analysis.

3.17 Method summary

Overall, the methodological design of the study can be summarised as a combined empirical and technical pipeline. The first part of the methodology established the structured interview design, the attribute logic, the organisation $\times$ use-case observation unit, and the separation between structured data and open-ended qualitative material. The second part then transformed that empirical structure into a reproducible analytical framework through schema-driven ETL, deterministic fit computation, encoded feature generation, grouped impact-model experimentation, coefficient-based interpretability analysis, and cluster-level strength estimation for playbook support.

This means that the methodology of the thesis is not only about how the data was collected, but also about how the analytical object was constructed. In practice, that construction work is a large part of the contribution, because the later quantitative results depend directly on how the structured interview material was normalised, represented, grouped, and evaluated. Therefore, the technical implementation should be understood as a methodological layer in its own right rather than as a purely operational coding step added after the fact.

4

Results

This chapter presents the empirical and technical results of the study. It starts by describing the study sample, since the interpretation of all later results depends on what kind of companies, respondents, and AI use cases the analysis is based on. After that, the chapter presents descriptive portfolio patterns and qualitative results before moving into the more technical analytical outputs, including baseline impact modelling, impact model interpretability, and cluster-strength analysis. Unless otherwise stated, the term use case in this chapter refers specifically to an AI use case.

The overall intention is to move from more direct study evidence towards more model-based results. This distinction is important because the different parts of the chapter do not support the same type of conclusion. The sample statistics, descriptive portfolio patterns, qualitative results, and observed-only cluster-strength results are closer to direct empirical description, while the baseline impact model and configuration-enabled cluster-strength analysis should be read more as exploratory technical evidence about modelling signal, ranking usefulness, and analytical sensitivity.

4.1 Study sample

Before presenting the modelling and cluster-strength results, the empirical base of the study should be made explicit. The study is based on interviews with 33 Nordic retail companies, covering companies from Sweden, Norway, Denmark, and Finland. The interviews generated two related use-case datasets. Firstly, respondents listed a broader set of 151 AI use cases. Secondly, a smaller subset of 55 use cases was coded in greater depth with structured attributes and observed impact ratings. The latter subset is the basis for the later impact-oriented quantitative results, including the baseline impact modelling and the observed-only cluster-strength analysis.

This distinction between all listed use cases and the in-depth impact-rated subset is important. The larger set gives a view of the broader AI use-case landscape that appeared in the interviews. The smaller set gives the more structured basis for modelling, because those rows include the company and use-case attributes needed for the technical pipeline. Table 4.1 summarises this empirical material.

Table 4.1: Study-sample overview.

Empirical material	Scope
Interviewed companies	33 Nordic retail companies
Countries represented	Sweden, Norway, Denmark, and Finland
All listed use cases	151 AI use cases listed by the interviewed companies
In-depth impact-rated use cases	55 use cases with structured attributes and observed impact ratings
Main respondent base	Senior technology, data, AI, digital, and transformation roles

4.1.1 Respondent role distribution

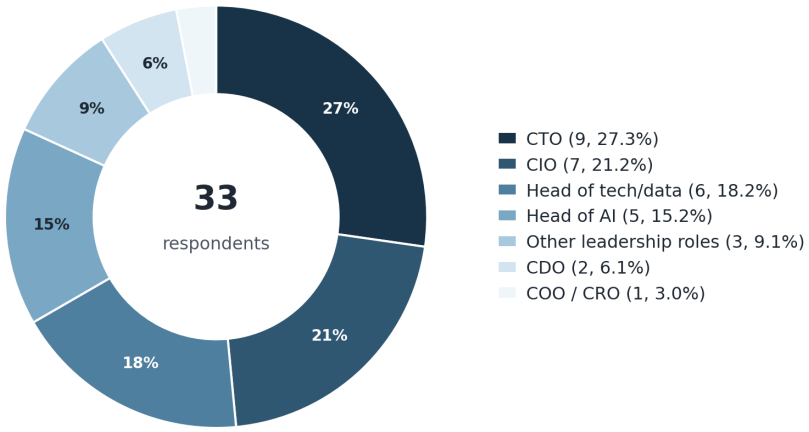


Figure 4.1: Respondent role distribution.

The respondent base was concentrated in senior technology, data, AI, and digital leadership roles, as shown in Figure 4.1. CTOs were the largest group, with 9 respondents, followed by 7 CIOs, 6 respondents in head of technology or data roles, and 5 respondents in head of AI roles. The remaining respondents were CDOs, COO/CRO-level leaders, or other leadership roles. This gives the study a strong basis for discussing AI prioritisation and implementation from a decision-making perspective. At the same time, the results should be read as senior-managerial and technology-oriented views of retail AI adoption, rather than as a broad measure of how all employees in these companies experience AI.

4.1.2 Distributions of interviewed companies

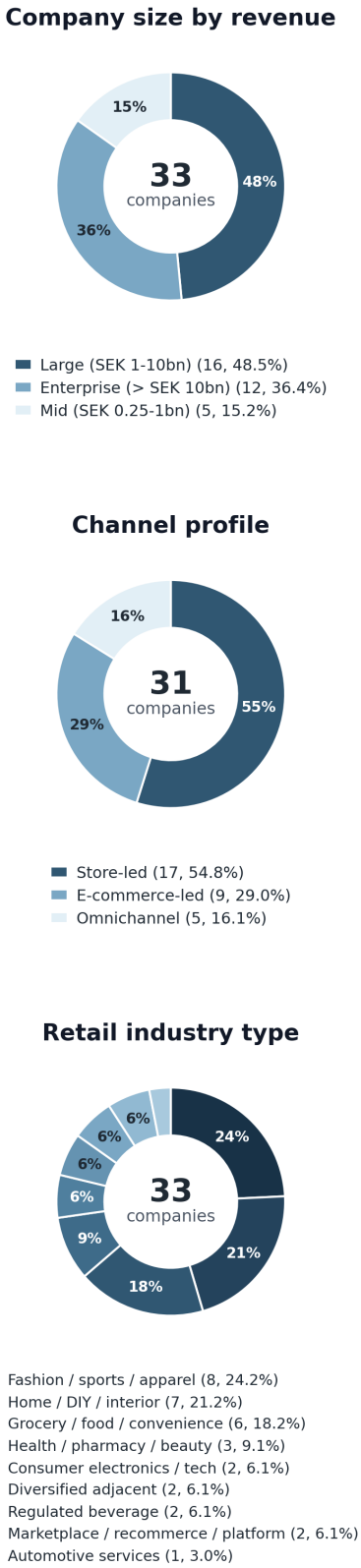


Figure 4.2: Distributions of interviewed companies by revenue class, channel profile, and retail industry type.

Figure 4.2 shows that the company sample is weighted towards larger retail companies. Large companies, defined as SEK 1–10bn in annual revenue, account for 16 of the 33 companies. Enterprise companies, defined as more than SEK 10bn in annual revenue, account for 12 companies, while mid-sized companies account for 5 companies. In other words, the empirical base is mainly a medium-to-large and enterprise retail sample.

The channel profile distribution uses 31 coded companies and shows that store-led retailers are the largest group, followed by e-commerce-led and omnichannel companies. The industry distribution is also relatively broad. Fashion / sports / apparel, home / DIY / interior, and grocery / food / convenience are the largest categories, but the sample also includes health / pharmacy / beauty, consumer electronics / tech, regulated beverage, marketplace / recommerce / platform, diversified adjacent retail, and automotive services.

This composition matters because the results should not be interpreted as representing every kind of retailer equally. Instead, the study mainly captures how AI is discussed and implemented among Nordic retailers with meaningful scale and with several different sector contexts represented.

4.1.3 AI solution pattern distributions

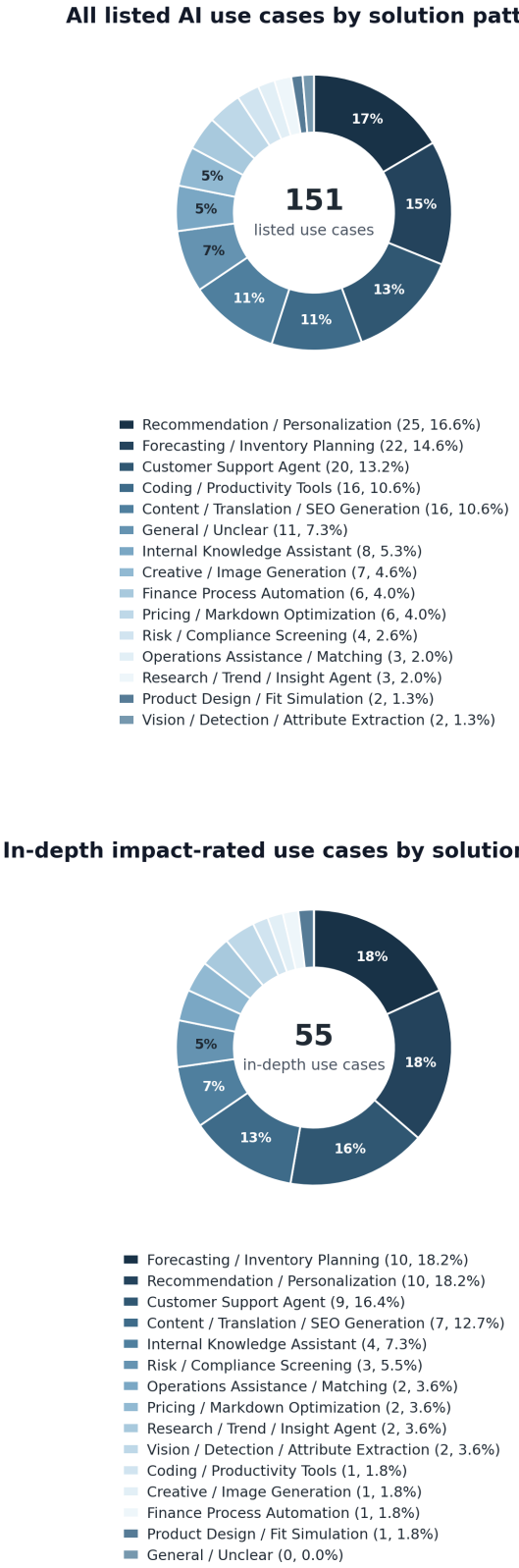


Figure 4.3: AI solution pattern distributions for all listed use cases and the in-depth impact-rated subset.

Figure 4.3 compares the broader list of 151 AI use cases with the 55 use cases that were coded in depth for impact-oriented analysis. In the full listed-use-case base, recommendation / personalization, forecasting / inventory planning, customer support agents, coding / productivity tools, and content / translation / SEO generation were among the largest categories. This shows which AI applications were most visible in the interviews.

The in-depth subset is not identical to the full list, but it still covers many of the same major solution patterns. Forecasting / inventory planning and recommendation / personalization each account for 10 of the 55 in-depth use cases, customer support agents account for 9, and content / translation / SEO generation accounts for 7. The remaining categories are smaller, but include internal knowledge assistants, risk / compliance screening, pricing / markdown optimisation, research / trend agents, and vision-based use cases.

The later impact-oriented results should therefore not be read as a random sample of all possible AI use cases. They are based on the use cases that were concrete enough in the interviews to be coded with structured attributes and an impact rating. This gives the analysis more depth, but it also makes the evidence a selected view of more developed or more clearly described AI use cases.

4.2 Portfolio patterns

The study sample also gives a set of simpler quantitative patterns that are useful before moving into the modelling sections. These results are not based on the baseline impact model or on the cluster-strength framework. Instead, they describe the coded company attributes, the coded use-case attributes, and selected averages across the 55 in-depth use cases. This makes the section closer to ordinary descriptive analysis than to prediction or model interpretation.

The section is included because it gives a more direct view of the AI use-case portfolio. Some findings describe adoption and readiness, while others describe which types of AI use cases appear faster, heavier, clearer to measure, or more impactful in the observed data. These results should still be read carefully, because several groups have small denominators. However, they provide an important bridge between the sample description and the more technical results that follow.

4.2.1 GenAI adoption

GenAI adoption was measured as a company-level indicator. A company was counted as having GenAI adoption if it had at least one specifically identified GenAI use case in the structured use-case material, or if the interview notes gave clear evidence of a GenAI application used in practice. General availability of tools such as ChatGPT, Claude, or Copilot was not counted on its own.

Using this definition, 25 of 30 companies with sufficient evidence, or 83.3 percent, had at least one identified GenAI use case. The studied in-depth use-case subset was also GenAI-heavy. Among use cases with non-missing solution-type coding, 35 of 53 use cases, or 66.0 percent, were coded as GenAI, while 18 use cases, or 34.0 percent, were coded as ML.

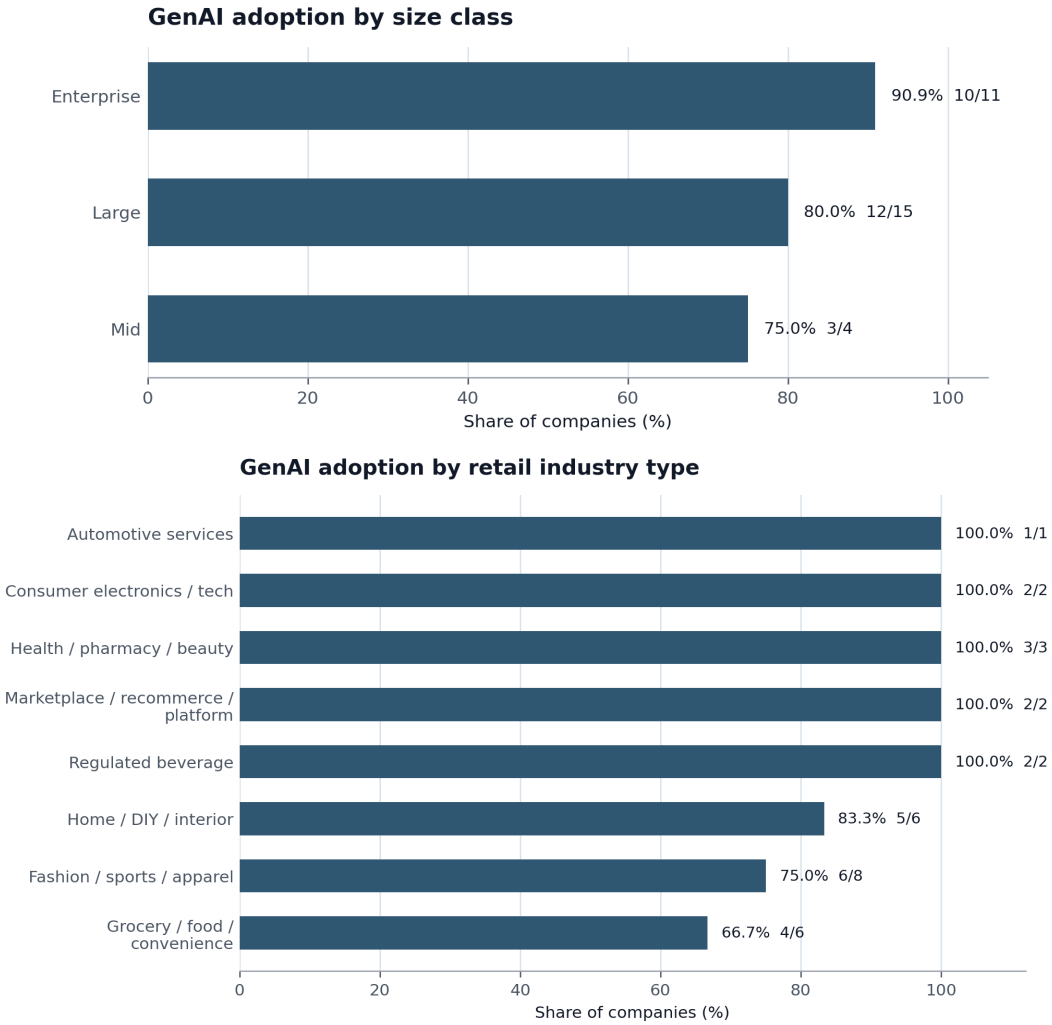


Figure 4.4: GenAI adoption by size class and retail industry type. The retail-industry view should be read together with the denominator labels, since several of the apparent top categories are based on only one to three companies.

Figure 4.4 shows that GenAI adoption was high across all size classes in the coded data. Enterprise companies had the highest share, at 90.9 percent, while large companies had 80.0 percent and mid-sized companies had 75.0 percent. The size pattern therefore suggests some difference, but not a complete separation between large and smaller companies. Even in the mid-sized group, three out of four coded companies had at least one identified GenAI use case.

The industry view is more difficult to interpret. Several retail industry types show 100 percent GenAI adoption, but some of these categories contain only one, two, or three companies. The result should therefore not be read as strong evidence that specific retail industries are always ahead in GenAI adoption. A more careful interpretation is that GenAI use-case activity was visible across several retail contexts, while the small denominators make the industry-level ranking uncertain.

It is also worth separating adoption from impact. Although the studied use-case portfolio was GenAI-heavy, ML use cases had a higher average observed impact in the coded subset, at 7.19 compared with 6.57 for GenAI use cases on the 0–10 impact scale. This does not mean that ML is generally better than GenAI. A more likely explanation is that many ML use cases in the sample were tied to forecasting, optimisation, or operational decision problems where impact was easier to observe and quantify. GenAI was widespread, but the impact results suggest that the AI portfolio should not be reduced to GenAI alone.

4.2.2 Company readiness

The company-side attributes suggest that many companies had some of the basic conditions for AI work in place, but that readiness was uneven across different dimensions. Figure 4.5 reports the low, medium, and high distribution for data readiness, its three underlying data-foundation attributes, IT competence, and change capacity. The data-foundation attributes are useful here because the overall data-readiness tier can otherwise hide whether the issue is availability, quality, or accessibility.

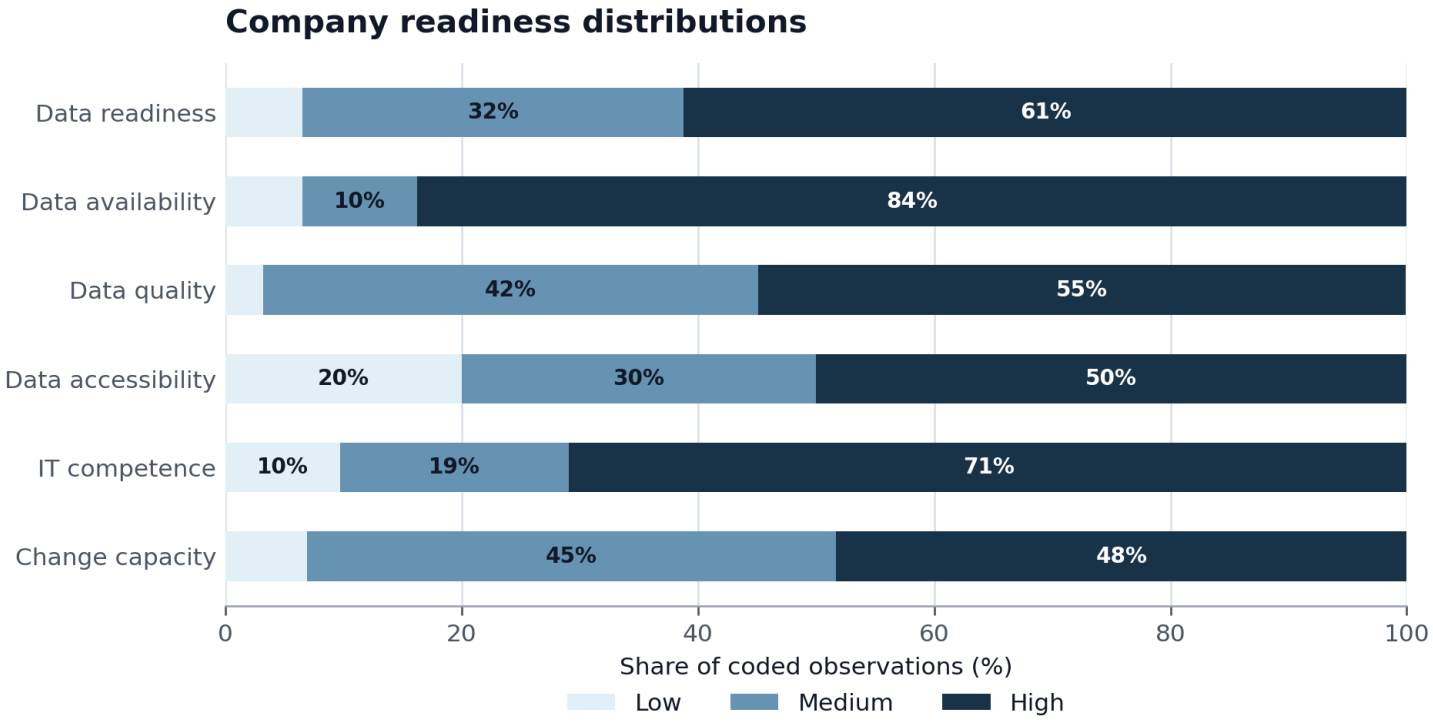


Figure 4.5: Company-readiness distributions across low, medium, and high coded categories. The chart separates overall data readiness from data availability, data quality, and data accessibility in order to show where the data foundation is strong or weaker.

The clearest pattern is that data availability was stronger than the other two data-foundation dimensions. Among companies with usable coding, 83.9 percent had high

data availability. Data quality was weaker, with 54.8 percent high and 41.9 percent medium. Data accessibility was the most mixed of the three, with 50.0 percent high, 30.0 percent medium, and 20.0 percent low. This suggests that many retailers have relevant data somewhere in the organisation, but that making the data reliable and usable across systems is more difficult.

The same figure also shows that IT competence was relatively strong in the sample, with 71.0 percent of companies coded as high. Change capacity was more evenly distributed, with 48.3 percent high and 44.8 percent medium. This difference is important for the interpretation of later results. A company may have technical competence and still struggle to absorb AI into workflows, governance routines, and operating processes. In practice, this means that readiness is not only a question of whether a retailer can technically build or integrate AI.

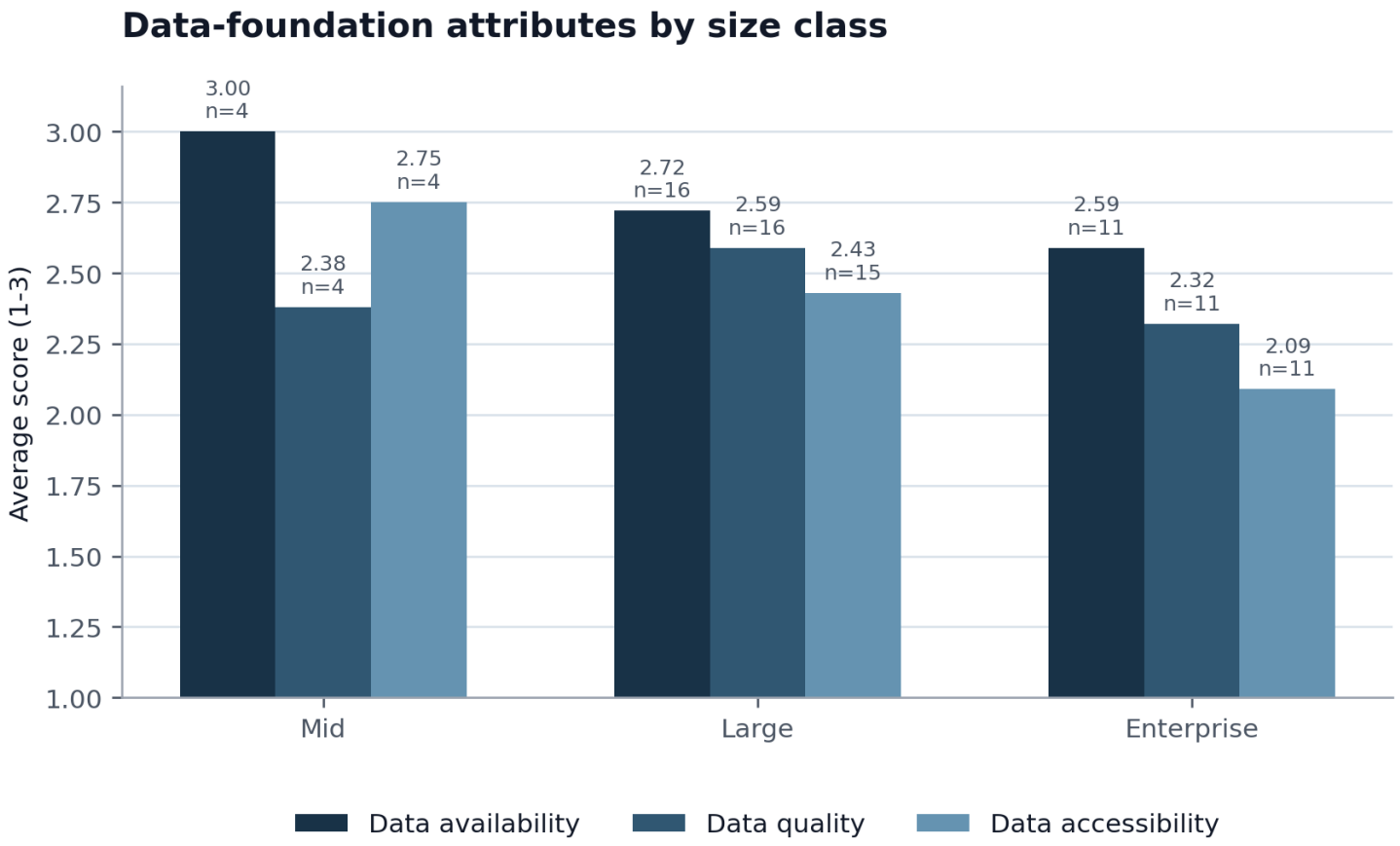


Figure 4.6: Average data-foundation attributes by size class. Scores are on the coded 1–3 scale, where 3 represents the strongest coded level.

Table 4.2: Average data readiness, IT competence, and change capacity by size class. Scores are on the coded 1–3 scale, where 3 represents the strongest coded level.

Size class	Companies	Data readiness	IT competence	Change capacity
Mid	4	2.71	3.00	2.75
Large	16	2.59	2.75	2.30
Enterprise	11	2.33	2.32	2.25

Figure 4.6 and Table 4.2 give a more detailed size-class view. The result is somewhat counterintuitive, since mid-sized and large companies score at least as high as the enterprise group on several readiness dimensions. Mid-sized companies have the highest average data readiness at 2.71, the highest average IT competence at 3.00, and the highest average change capacity at 2.75, although this is based on only four companies. Large companies follow with average data readiness of 2.59, IT competence of 2.75, and change capacity of 2.30. Enterprise companies have lower averages, at 2.33 for data readiness, 2.32 for IT competence, and 2.25 for change capacity.

The component-level view suggests that data accessibility is one reason for this pattern. Enterprise companies still have fairly high data availability at 2.59, but their average data accessibility is 2.09. A possible explanation is that larger retailers may have more data, but also more systems, business units, legacy architectures, and governance layers. This makes the result plausible even if it should not be overstated. The sample is too small to conclude that enterprise retailers are generally less data ready. What the data does show is that scale does not automatically translate into easier access to usable AI data.

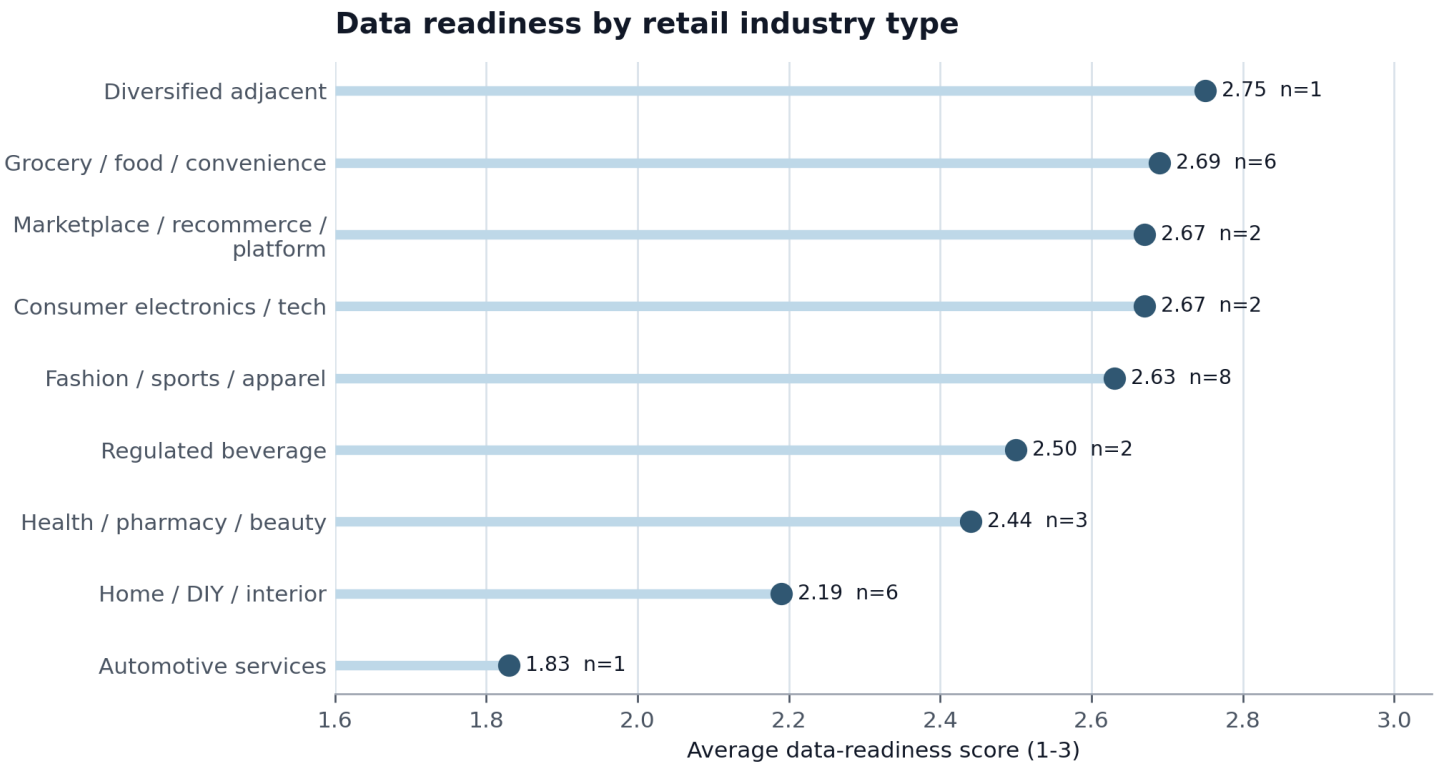


Figure 4.7: Average data readiness by retail industry type. Scores are on the coded 1–3 scale. The figure should be interpreted cautiously because several industry categories contain few companies.

Figure 4.7 gives an additional retail-industry view of data readiness. The plot works mainly as a descriptive overview and should not carry too much analytical weight. Some categories have only one or two companies, which means that the averages can move substantially if a single company is coded differently. The main value of the figure is therefore not to rank retail industries precisely, but to show that data readiness varied meaningfully across the interviewed companies and that this variation was not only a size-class issue.

4.2.3 Implementation burden

The use-case attributes show that implementation burden is not one single thing. Some use cases are difficult because they require process change. Others are difficult because they require technical integration, repeated maintenance, high build effort, or slow time to value. Figure 4.8 summarises these distributions across the studied use cases.

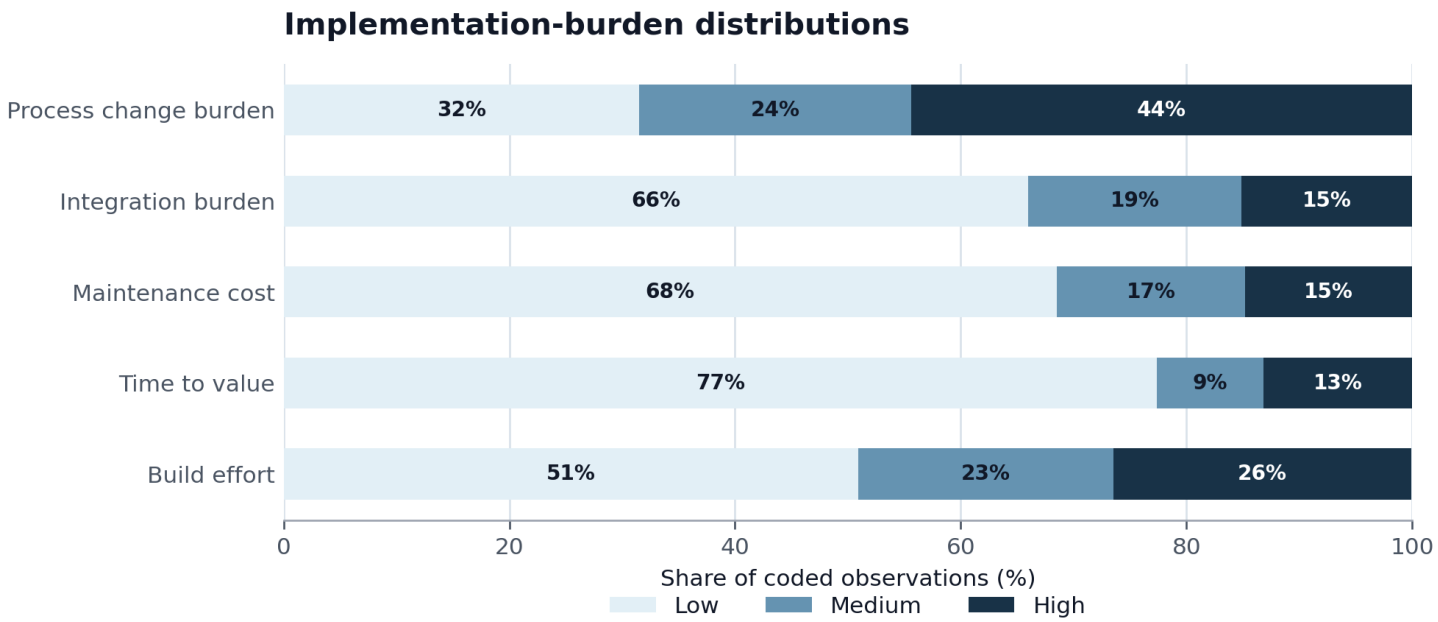


Figure 4.8: Implementation-burden distributions across the studied use cases. Low, medium, and high refer to the coded burden tiers for each attribute.

The most visible result is that process-change burden was higher than the other implementation-burden dimensions. In the coded use-case subset, 44.4 percent of use cases had high process-change burden and 24.1 percent had medium process-change burden. By contrast, integration burden was low for 66.0 percent of use cases, maintenance cost was low for 68.5 percent, and time to value was low for 77.4 percent. Build effort was more mixed, with 50.9 percent low, 22.6 percent medium, and 26.4 percent high.

This suggests that the difficult part of implementation often lies outside the model itself. In several cases, the technical build or integration may be manageable, while the heavier issue is how the use case changes work routines, responsibilities, decision rights, and adoption behaviour. This is only a descriptive result, but it is consistent with the qualitative material discussed later in the report, where several respondents described AI scaling as a workflow and ownership challenge rather than only a technology challenge.

Table 4.3: Average integration burden by solution type. Scores are on the coded 1–3 scale, where higher values indicate heavier integration burden.

Solution type	Use cases	Avg. integration burden
ML	15	1.70
GenAI	37	1.41

Table 4.3 gives one simple comparison by solution type. ML use cases had a somewhat

higher average integration burden than GenAI use cases, at 1.70 compared with 1.41. This is a rather moderate difference rather than a sharp separation. Still, it fits the idea that prediction, optimisation, and machine-learning-heavy use cases often need closer connection to historical, operational, or transactional systems, while many GenAI use cases in the sample could be applied around language, content, or knowledge workflows with lighter system integration.

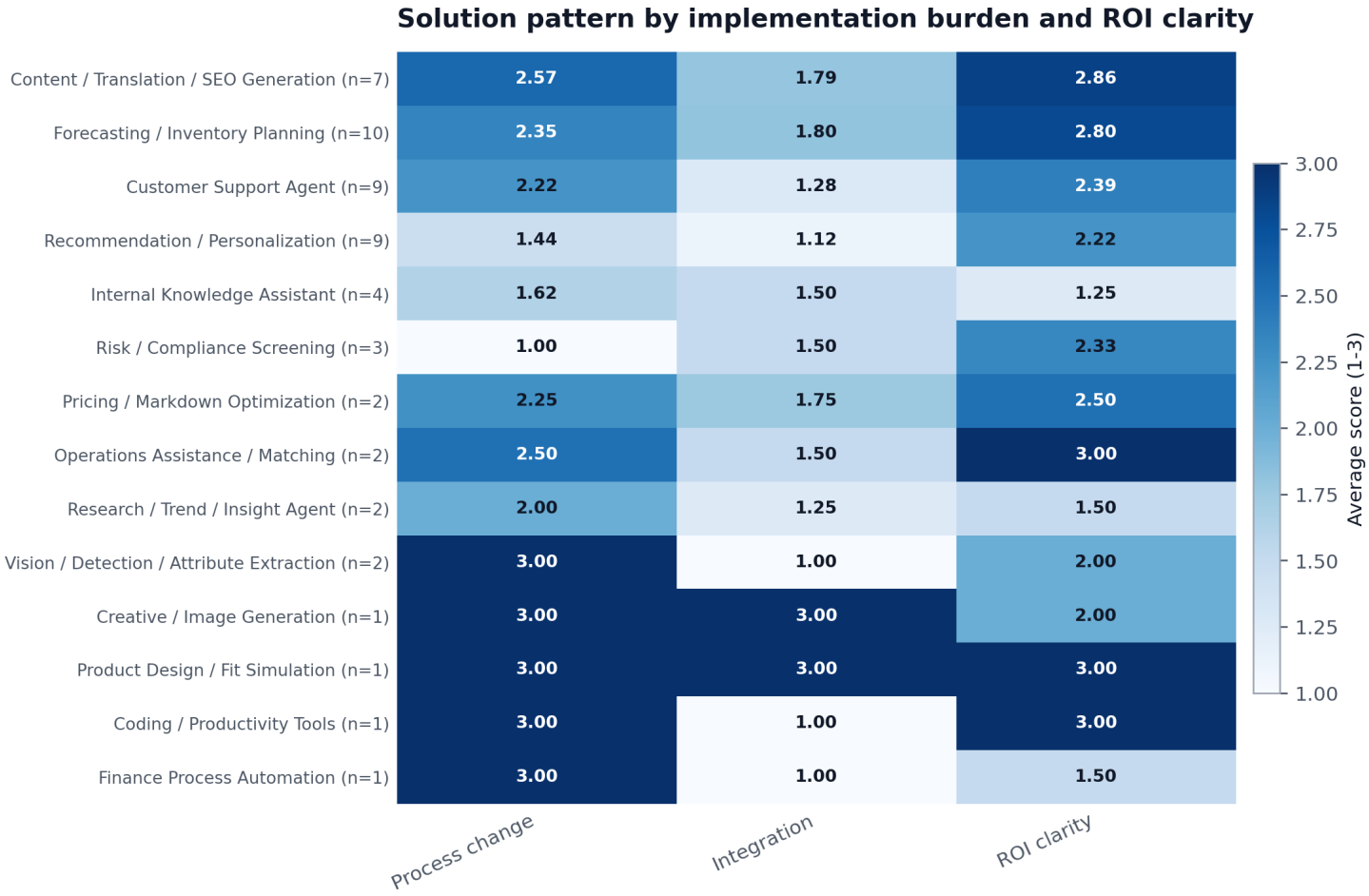


Figure 4.9: Average process-change burden, integration burden, and ROI-metric clarity by AI solution pattern. Scores are on the coded 1–3 scale, and row labels include the number of use cases in each solution-pattern category.

Figure 4.9 shows that the implementation profile also differs by AI solution pattern. Content / translation / SEO generation has high average process-change burden at 2.57 and high ROI-metric clarity at 2.86. Forecasting / inventory planning also has a high process-change burden at 2.35, a relatively high integration burden at 1.80, and high ROI-metric clarity at 2.80. Here, ROI-metric clarity should be read as the degree to which the use case has a clear way of measuring value, not as a direct statement that the ROI itself is high. These two patterns therefore appear different from very light experimental use cases. They may be attractive partly because the value pathway is clearer to measure, but they also require real implementation work.

Customer support agents form a slightly different pattern. They have a relatively high average process-change burden at 2.22, but a lower average integration burden at 1.28. A possible interpretation is that the main difficulty is not always technical integration into many systems, but rather response quality, escalation rules, customer acceptance, and how service work is reorganized around the tool. Recommendation / personalization has lower integration burden at 1.12 and lower ROI-metric clarity at 2.22, which may reflect that the measurement logic for personalization can be harder to isolate from the surrounding commercial context. Some rows in the heatmap have very low support, so the strongest interpretation should focus on the better-supported categories.

4.2.4 Impact and delivery patterns

The final descriptive portfolio view combines impact, time to value, and delivery source. These results are useful because an AI roadmap is not only about choosing use cases with high average impact. It also needs to separate faster use cases from slower ones, and lighter delivery routes from more internally shaped or integrated routes.

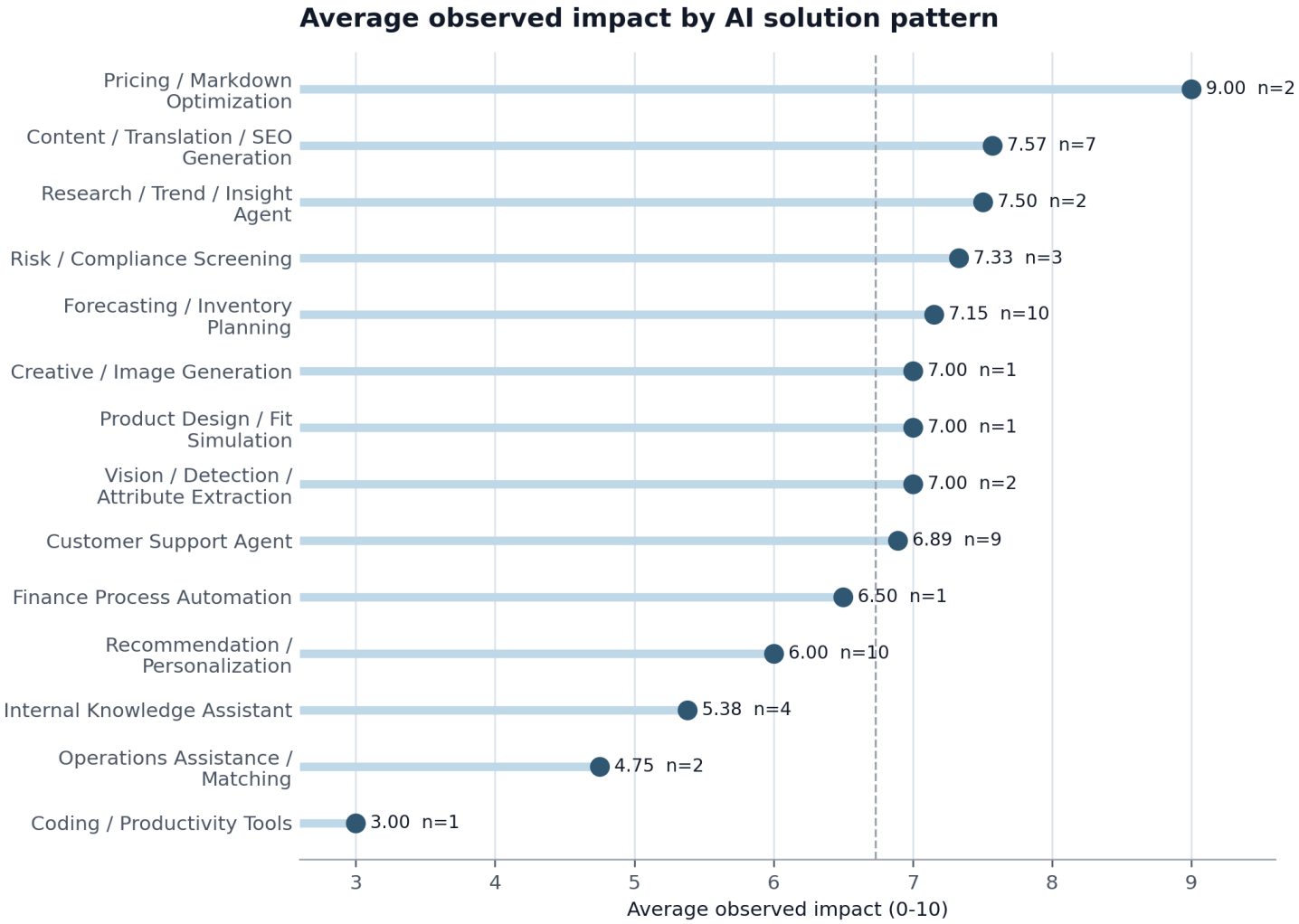


Figure 4.10: Average observed impact by AI solution pattern. The dashed line marks the overall observed-impact average of 6.73. Row labels include the number of use cases in each solution-pattern category.

Figure 4.10 shows that several of the highest-impact solution patterns are relatively specific. Pricing / markdown optimisation has the highest average observed impact at 9.00, but this is based on only two use cases and should therefore be interpreted as exploratory. Content / translation / SEO generation has a high average impact of 7.57 across seven use cases, and forecasting / inventory planning has an average impact of 7.15 across ten use cases. Risk / compliance screening and customer

support agents also sit above the overall average, although with different support levels.

The result should not be read as a definitive ranking of AI solution patterns. Some categories have only one or two observed cases, and the average impact value can therefore be sensitive to single observations. However, the better-supported high-impact categories are still informative. Content / translation / SEO generation, forecasting / inventory planning, and customer support agents all appear as relevant categories in the studied use-case portfolio, but they differ in speed, implementation burden, and likely value mechanism.

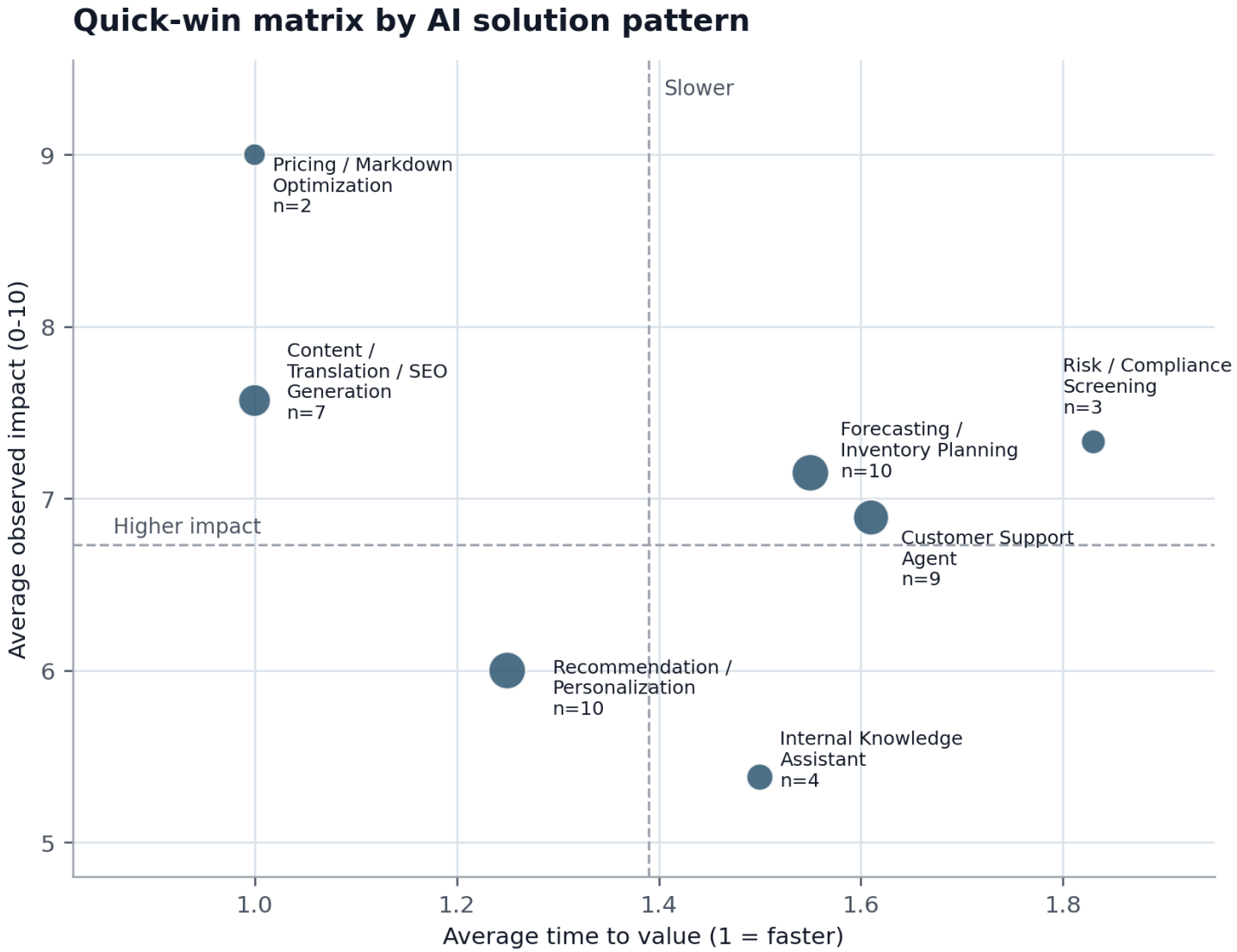


Figure 4.11: Quick-win matrix by AI solution pattern. The dashed lines mark the overall average impact and average time-to-value values for the plotted categories. Circle size represents the number of use cases in each category.

Figure 4.11 makes this tradeoff more visible by combining average observed impact

with average time to value. Content / translation / SEO generation is the clearest high-impact and fast category in the better-supported part of the plot, with average impact of 7.57, average time to value of 1.00, and seven use cases. Forecasting / inventory planning and customer support agents also have relatively high observed impact, but their average time to value is slower at 1.55 and 1.61 respectively. Recommendation / personalization is faster, at 1.25, but has a lower average observed impact of 6.00 in this subset.

The figure should be interpreted as a portfolio map rather than as a mechanical prioritisation rule. A fast category can be useful for learning and momentum, while a slower category can still be more strategically important if it affects core retail processes. The plot mainly shows that the studied AI portfolio contains different types of opportunities. Some are closer to quick wins, while others look more like heavier operating improvements.

Table 4.4: Delivery-source distribution among observed-impact use cases.

Delivery source	Use cases	Share
Internal	27	50.0%
Off-the-shelf	16	29.6%
External	11	20.4%

Table 4.5: Delivery-model profile by delivery source. Impact is measured on the 0–10 observed-impact scale. Time to value and maintenance cost are measured on coded 1–3 scales, where higher values indicate slower time to value or heavier maintenance.

Delivery source	Use cases	Avg. impact	Avg. time to value	Avg. maint. cost
Internal	27	7.28	1.30	1.37
External	11	6.32	1.45	1.36
Off-the-shelf	16	6.19	1.50	1.78

Tables 4.4 and 4.5 show that internal delivery was the largest delivery source in the impact-rated subset, representing 27 of 54 use cases, or 50.0 percent. It also had the highest average observed impact, at 7.28, compared with 6.32 for externally delivered use cases and 6.19 for off-the-shelf use cases. Internal delivery also had the fastest average time to value in this comparison, at 1.30.

This does not prove that internal delivery is generally superior. It may partly reflect selection effects, since companies are more likely to build or adapt internally when they have stronger internal capability, clearer ownership, or more strategically important use cases. Even so, delivery source appears to capture more than who supplied the tool. In this sample, it also says something about how closely the use case could be adapted to the retailer's own data, workflows, and operating priorities.

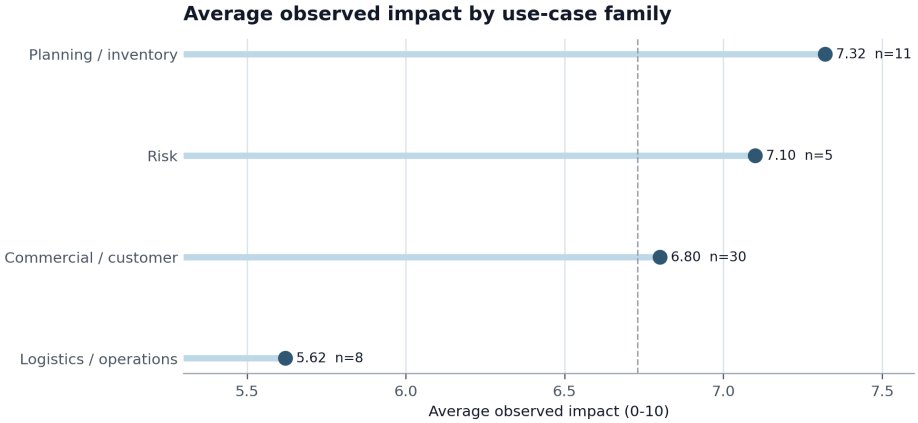


Figure 4.12: Average observed impact by use-case family. The dashed line marks the overall observed-impact average of 6.73.

Finally, Figure 4.12 shows the same impact logic at a broader use-case-family level. Planning / inventory has the highest average observed impact at 7.32, followed by risk at 7.10 and commercial / customer at 6.80. Logistics / operations is lower in this view, at 5.62. This reinforces the pattern that planning and inventory-related AI use cases are among the stronger observed areas in the study, while also reminding us that impact depends on the way a use case is implemented, not only on the label attached to it.

4.3 Qualitative results

The quantitative results above describe patterns in coded company and use-case attributes. The qualitative material adds a different layer. It helps explain how respondents described AI adoption, what kinds of difficulties they emphasised, and why some use cases appeared easier to discuss than to implement in practice. The results in this section are organized around four recurring themes: AI execution, readiness, use-case value, and governance and operating model.

These findings should be read as qualitative patterns rather than statistically estimated effects. They are based on interview notes, open-ended responses, and repeated themes across respondents. Some points are strongly recurring, while others are more contextual. Together, they help explain what is difficult about retail AI adoption beyond the structured attribute values.

4.3.1 AI execution

4.3.1.1 AI's bottleneck is execution depth, not idea generation

The strongest broad pattern is that most retailers no longer seem blocked by a lack of ideas. Instead, the recurring challenge is moving from isolated AI experiments or narrow proofs of concept into production-grade, trusted, integrated ways of working. Across the material, interviewees repeatedly describe friction around implementation

depth, integration, ownership, workflow redesign, governance, and long-run operating responsibility rather than around initial inspiration. The practical implication is that retail AI roadmapping should not only ask which use cases look attractive. It should also ask which use cases the organisation is realistically capable of embedding into real operations.

4.3.1.2 Business pull is what turns AI into traction

The material repeatedly suggests that AI initiatives work best when there is a real business-side need, clear local commitment, and someone who genuinely wants the solution to succeed in day-to-day work. Several interviews point away from AI being something that should be pushed by a central technical team alone. Instead, strong momentum seems to arise when domain teams experience the problem as their own and see the use case as directly useful in their work. This theme also explains why some apparently modest use cases become important: they are concrete enough for a real team to adopt, improve, and talk about.

4.3.1.3 Process change often outweighs the technical challenge

Many comments imply that technical build difficulty is only one layer of the real implementation burden. Even when a use case is technically feasible, the organisation may still need to change how work is divided, how people trust outputs, which escalations happen automatically, how review is done, how responsibility is assigned, and what people are expected to do instead of the manual task AI has absorbed. Several respondents explicitly describe situations where the harder part was not the model itself but the surrounding process, the adoption curve, the frontline behaviour change, or the organisational redesign needed to capture the value.

The material also suggests that process change is partly about role identity. Some employees are not only asked to use a new tool; they are asked to trust a system with work that previously expressed their expertise, judgement, or control. This is visible in comments around buyers, customer-service teams, store or warehouse employees, designers, and other operational users. For executives, this means that AI adoption should be treated as a people and role design issue, not only as process optimisation.

4.3.1.4 Trust and oversight must be designed per use case

The interviews do not support a one-size-fits-all position on autonomy. In some bounded environments, respondents are comfortable letting AI act with little direct human intervention. In other areas, especially those involving sensitive customer communication, body-related recommendations (fashion), uncertain output quality, or business-critical decisions, human review is seen as necessary. What matters most seems to be risk, reversibility, reputational exposure, and whether a bad answer can be corrected cheaply. This suggests that roadmapping should treat human-in-the-loop design as a core use-case choice, not as an afterthought. The same logic also applies to readiness: several comments suggest that retailers are often ready for AI that recommends, summarises, drafts, routes, or supports, but less ready for AI that acts broadly across systems with delegated authority.

Trust also depends on quality thresholds, not only on whether a human is in the loop. Several comments point to concerns around hallucinations, confidence, wrong recommendations, output drift, and whether an AI system knows when it should not answer. For executives, the practical question is therefore not only "should a human review this?", but also what level of accuracy, confidence, fallback behaviour, and monitoring is required before the organisation can reasonably rely on the system. The trust threshold also changes by retail context. A wrong answer in a low-risk internal workflow is different from a wrong answer in pharmacy, body-related recommendations, sensitive customer service, financial decisions, or confidential employee and customer data.

4.3.1.5 Frontline adoption decides whether AI becomes real

Several comments show that AI adoption is not decided only in central teams, IT, analytics, or executive forums. Many use cases become real only when store employees, warehouse staff, customer-service teams, buyers, marketers, franchisees, or other operational users actually change how they work. These groups may have different incentives, trust levels, time constraints, and comfort with AI than central teams do. This matters because an AI solution can look strong from a head-office perspective while still depending on adoption in daily retail work. The roadmap should therefore ask where the user sits in the organisation, how much their behaviour must change, whether the tool makes their work easier in a way they recognize, and who supports adoption after launch.

One framing that appears in the material is that AI may be more acceptable when it is positioned as coaching or support rather than surveillance or control. In frontline environments such as warehouses, stores, or customer service, AI can help employees understand what to improve, but the adoption logic changes if the same system is perceived as monitoring performance or replacing judgement. This makes communication, intent, and local trust central to implementation.

4.3.1.6 Product and supplier data are retail AI bottlenecks

The general data-readiness theme has a more retail-specific version: many AI use cases depend heavily on product data, supplier data, master data, product attributes, product images, descriptions, categories, and other forms of commercial context. Several comments suggest that these data assets can be fragmented, supplier-dependent, manually entered, stored as free text, or inconsistent across systems. This matters because retail AI is often product-context hungry. Use cases such as search, recommendations, product enrichment, translation, markdowns, forecasting, assortment planning, customer support, and store operations can all depend on product information being structured and trustworthy. For executives, this suggests that product-data quality is not a back-office detail; it is a strategic enabler for a wide range of AI use cases.

4.3.2 Readiness

4.3.2.1 Data readiness means connected, governed context

The interviews rarely suggest that retailers simply have no data. A much more common pattern is that they have data spread across systems, uneven data quality by domain, weak accessibility across boundaries, inconsistent definitions, unclear ownership, or legal constraints on how data can be combined and used. In several cases, the core issue is not raw data scarcity, but that AI needs context that existing architectures and governance routines cannot yet provide cleanly. This matters because it changes the executive framing. The readiness question is less about "do we have data?" and more about whether the right data can be trusted, accessed, combined, governed, and operationalised in the specific workflow where AI is supposed to create value.

The material also suggests that data readiness includes data ownership and data sovereignty. Several executives appear concerned with where data is stored, whether sensitive information leaves the organisation, whether EU or US jurisdiction applies, and whether external AI providers can access company data or use it to train models. This means that AI readiness is partly a legal and sourcing question: retailers need to know not only whether data exists, but also who controls it, where it goes, and what providers may do with it.

4.3.2.2 Start where repetitive operational pain is already visible

When interviewees talk concretely about AI creating value, they usually refer to workflows with one or more of the following properties:

- high transaction volume
- repeated decisions
- information overload
- manual text handling
- repetitive customer interaction
- operational planning under uncertainty

This is why the material keeps returning to areas such as customer service, forecasting, replenishment, pricing, product-content work, knowledge retrieval, order handling, and process support in logistics or store operations. The qualitative evidence suggests that these domains feel "AI-realistic" because they sit close to visible pain and already contain work that people find tedious, slow, inconsistent, or difficult to scale manually.

Retail scale also changes the economics of what counts as meaningful. In large flows of products, customers, orders, inventory, stores, or service contacts, even small improvements in forecasting, pricing, shrinkage, conversion, service cost, or working capital can become strategically important. This means that some AI cases should be judged less by whether they look dramatic and more by whether they improve a repeated high-volume mechanism.

4.3.2.3 Readiness varies by workflow more than by company

The interviews also suggest that it can be misleading to describe an entire retailer as simply "ready" or "not ready" for AI. Within the same company, one department may be curious, data-rich, commercially motivated, and willing to change, while another may be more constrained by legacy systems, manual habits, weak process ownership, or limited time to learn. Several comments point to different levels of change appetite, technical maturity, or governance clarity across functions. This matters for roadmapping because company-level readiness is too blunt for many practical decisions. A retailer may be ready for AI in customer service, content operations, or forecasting, while not yet ready for broader autonomous agents, personalization, or cross-functional process automation.

The material also hints at a size-related readiness difference. Smaller or more focused organisations may sometimes move faster because fewer people, systems, and governance layers need to be coordinated. Larger retailers may have larger value pools and more data, but they also face more complexity around security, scale, role uncertainty, and coordination across functions. This should not be treated as a universal rule, but it is still a relevant lens when comparing AI execution capacity. Retail operating models add another layer. Franchised, decentralized, or store-led organisations may need to convince independent retailers, franchisees, store managers, or local teams to change behaviour, even when the central organisation believes the AI case is strong.

4.3.2.4 The right AI pattern is not always GenAI

The qualitative material repeatedly suggests that executives should not treat generative AI as the default answer to every AI opportunity. Some problems are better suited to traditional machine learning, optimisation, deterministic automation, embedded analytics, rule-based controls, or workflow tooling. In several comments, respondents distinguish between use cases where GenAI is valuable because language, summarisation, generation, or flexible reasoning is needed, and use cases where a more deterministic or predictive approach is more reliable. This is a useful anti-hype insight for retail AI roadmapping.

4.3.2.5 Different AI cases need different measurement logic

The interview material suggests that AI value is not always measurable through one simple ROI logic. Some use cases have clean operational metrics, such as labour saved, tickets deflected, forecast accuracy, shrinkage reduction, throughput, service level, or conversion. Others rely on proxy metrics, such as first-response resolution,

adoption, user satisfaction, quality of generated content, or reduced manual review. A third group creates value through learning, confidence, sustainability, reputation, or organisational capability, which can be important but harder to isolate financially. For executives, this means that the AI portfolio should not force every use case through the same measurement frame. Scaled operational cases should have clearer value metrics, while learning-oriented or adoption-enabling cases may need a different evaluation logic.

4.3.3 Use-case value

4.3.3.1 AI can make previously uneconomical work scalable

The qualitative material does not only describe AI as a way to automate tasks that humans already perform. It also points to a second kind of value: AI can make work possible that was previously too expensive, too slow, or too labour-intensive to do at scale. Examples in the material include deeper supplier or pricing analysis, broader sales-opportunity scanning, richer product-data enrichment, more frequent content localization, more systematic customer-service analysis, and faster sorting of large information sets. For executives, this is a useful distinction. Some AI use cases should be evaluated as efficiency improvements to existing processes, while others should be evaluated as capability expansion: the organisation can start doing valuable analytical or operational work that it theoretically wanted before, but could not justify with human capacity alone.

4.3.3.2 The workflow wrapper often determines AI value

Several examples imply that the model itself is only one part of the AI use case. The surrounding workflow often matters just as much: the user interface, the escalation logic, the APIs connected to the model, the approval points, the fallback process, and the way employees are expected to act on the output. In practice, a technically capable model can still create weak business value if it is not wrapped in a usable operating process. For executives, this means that AI use-case evaluation should include a "workflow wrapper" question: what will people see, trust, approve, correct, or act on? This is especially important for customer service agents, internal support tools, product-content workflows, store-facing assistants, and other cases where value depends on the model being embedded in daily work rather than merely generating a technically plausible output.

4.3.3.3 AI policy should enable safe use, not only restrict it

The material suggests that AI governance is not only about restricting risky use. It is also about creating safe and practical conditions for employees to use AI productively. Policies, approved tools, enterprise accounts, training, and clear guidance can make the difference between controlled learning and fragmented shadow usage. This matters because many retailers appear to face a dual challenge: they want employees to experiment and build familiarity, but they also need to control data exposure, provider risk, privacy, and inconsistent use. A useful AI policy therefore needs to be

enabling as well as restrictive. It should answer not only "what is forbidden?", but also "what is safe to try, with which tools, for which types of data, and under what human responsibility?"

4.3.3.4 AI reveals undocumented process knowledge

An additional theme in the qualitative material is that AI implementation can act as a forcing function for organisational clarity. Several comments suggest that when a retailer tries to build an AI assistant, agent, recommender, or decision-support tool, it quickly becomes visible which routines are not properly documented, which exceptions are handled through informal human judgement, and which definitions or process steps are still unclear. This matters because AI does not only require data; it also requires structured organisational context. In some cases, the work of preparing an AI use case therefore becomes a way of discovering weak process documentation, inconsistent business rules, unclear escalation paths, or implicit knowledge that has previously lived inside experienced employees rather than in systems.

4.3.3.5 Fast vendor change favors modular AI architecture

Some of the qualitative material reveals another important executive concern: the AI landscape changes quickly, providers evolve, costs move, and capabilities that once required a custom build may later become standard platform features. This makes long-term commitment feel uncertain. Several respondents therefore seem drawn towards modularity, iterative build-up, and architectures that can be improved or swapped out over time instead of tightly locking the organisation into one brittle solution design.

4.3.3.6 Some AI cases build capability before direct ROI

The material suggests that not all valuable AI initiatives are judged in the same way. Some have clear cost, labour, revenue, or service metrics. Others are valuable because they help the organisation learn how to work with AI, build confidence, or create political proof that AI can be useful. The latter may not always look strongest on a narrow financial screen, but they can still have important strategic value if they increase later execution capacity. In a few cases, value also appears through sustainability, reputation, or external signaling rather than through immediate financial return, which reinforces the need to classify use cases by value type before judging them.

4.3.3.7 AI leadership mandate is not the same as use-case ownership

The material suggests that two forms of ownership matter. Local business ownership is needed for a specific use case to be adopted and improved in the workflow where it belongs. But several comments also point to the need for a broader leadership mandate: senior leaders, executives, or clearly mandated individuals who can push through organisational resistance, fund enabling infrastructure, set expectations, and make AI adoption feel like part of the company's direction rather than a collection of local experiments. This distinction matters because a use case can

have an enthusiastic local owner and still fail to scale if the broader organisation does not allocate time, remove blockers, or clarify priorities. Conversely, central leadership ambition can remain abstract if it is not connected to concrete business-side ownership.

4.3.4 Governance and operating model

4.3.4.1 Data ownership and model-provider risk shape AI adoption

Retailers are concerned not only with whether an AI tool works, but with where company data is stored, who can access it, and whether external providers can use it to train models. This becomes especially important for LLM-based tools and cloud AI, where data may move across vendor, jurisdictional, and infrastructure boundaries.

For executives, AI sourcing is therefore also a data-governance and risk decision. The playbook should distinguish between low-risk internal experimentation, sensitive use cases requiring stronger contractual or technical controls, and use cases where data residency, ownership, or provider training policies may become gating criteria.

4.3.4.2 Agentic AI is an operating-model question

The more agent-oriented comments are among the most strategically interesting. Here, the conversation shifts from isolated tasks to questions such as:

- which agents should exist
- what data each agent may access
- what actions agents may take
- how agents coordinate across functions
- who owns and supervises them
- how permissioning, auditability, and security should work

The material suggests that agentic AI raises an operating-model problem as much as a technology problem. Several respondents seem to view this as promising but still structurally unresolved.

The same point applies to data access rights. As AI systems become more agent-like, retailers need to decide not only what data exists, but which agents, employees, tools, and external providers may access which data, for which purpose, and under which guardrails. In this sense, data access becomes a part of operating-model

design rather than a purely technical permissioning question. Agentic and customer-service-oriented use cases also expose hidden implementation work around scenario mapping. Several comments suggest that the time-consuming part is often not the initial model or interface, but mapping business rules, exception cases, escalation paths, conversation flows, "what happens if" situations, and guardrails.

4.3.4.3 Explainability is needed where AI affects consequential decisions

Several AI use cases require users to understand not only the output, but also why the system produced it and where its limits are. This is especially relevant in pricing, replenishment, workforce planning, customer communication, and other operational decisions where trust, accountability, and reversibility matter.

The playbook should not treat explainability as equally important for every AI use case. It should be strongest where decisions are hard to reverse, customer-facing, financially consequential, or likely to be challenged by employees, managers, or regulators.

4.3.4.4 AI capability must spread without becoming ungoverned

Another recurring theme is that central AI expertise remains necessary, but it is not sufficient on its own. Several comments suggest that retailers will need more AI-literate employees across merchandising, marketing, supply chain, finance, customer service, store operations, and other domains.

To make full use of AI, organisations need domain experts in these areas who also possess data, AI, and technology competence. These individuals are critical for identifying strong use cases, avoiding missed opportunities, assessing what is feasible, and, perhaps most importantly, implementing AI solutions in ways that fit the specific domain, department, and context. Without this fit, much of the potential value is either lost or never uncovered.

At the same time, respondents also emphasise the continued need for central guardrails around security, platform choices, integration, and production quality. The emerging model therefore looks less like pure centralization or pure democratization, and more like a hybrid: domain-level adoption is distributed, while infrastructure and governance remain centrally coordinated.

4.3.4.5 Exploration only matters if it leads to scale discipline

The interviews do not support a simple universal answer to the "broad or deep" question. Some respondents emphasise the value of many small proofs of concept to discover where AI really works. Others argue that most companies waste money if they spread themselves too thin. Taken together, the material points towards a more nuanced posture: selective breadth for learning, but disciplined depth for cases that are meant to produce lasting value. In practical terms, this means organisations should explicitly separate:

- learning-oriented experiments

- cases intended to become scaled operating capabilities

4.3.4.6 Strong AI opportunities come from several discovery routes

The qualitative material suggests that strong AI opportunities do not emerge from one single source. Some are triggered by operational pain, where a department already knows that a workflow is slow, costly, repetitive, or hard to scale. Others come from strategic ambition, external inspiration, new hires, technology partners, or leaders seeing what similar companies have tried. A third route is more analytical: decomposing workflows into steps and asking where AI could remove, accelerate, or improve specific activities. This connects closely to the breadth-versus-depth question. Broad exploration can be useful when it helps the organisation discover where AI actually fits, but the material suggests that discovery becomes stronger when people have enough AI literacy to identify opportunities in their own work.

4.3.4.7 Vendor hype makes AI sourcing a strategic choice

The material suggests that retailers are not only evaluating AI use cases; they are also evaluating a fast-moving solution market where many vendors, platforms, and existing enterprise systems now claim AI capability. This creates a practical decision problem: it can be difficult to separate genuine workflow value from generic AI branding, especially when a solution is offered as a point product, an embedded platform feature, or a partner-built customisation. This matters because sourcing decisions become part of AI strategy. Retailers need to decide when to buy standard functionality, when to wait for AI to be embedded in existing enterprise systems, when to use a partner, and when a case is sufficiently core or differentiated to justify internal ownership.

4.3.4.8 Bandwidth scarcity can block willing organisations

The interviews also suggest a distinction between willingness to change and capacity to change. Some organisations or functions appear positive towards AI, but still lack the time, bandwidth, or managerial attention needed to learn new tools, redesign workflows, test use cases, or absorb new operating routines. This makes AI adoption partly a resource-allocation problem, not only a culture or capability problem. This is important for roadmapping because it changes what "readiness" means. A company may have competent people and a positive attitude, but still fail to progress if the same people are already fully occupied by daily operations, system projects, or commercial priorities. In those cases, a smaller but well-owned use case may be more realistic than a broader transformation initiative that requires sustained attention from overloaded functions. On the other hand, it could also signal that the company must reprioritise and allocate more human capital towards AI transformation.

4.3.4.9 Customer acceptance constrains customer-facing AI

Customer-facing AI use cases introduce a separate adoption question: whether customers actually accept the AI-mediated experience. The material includes examples where retailers are careful about using AI in customer service or customer interactions

because the interaction itself is part of the customer relationship. In these cases, the risk is not only that the model is wrong, but that the customer feels poorly treated, less understood, or pushed away from human support at the wrong moment. This makes customer acceptance an important complement to technical accuracy and ROI. Some AI tools may be suitable for internal productivity or backstage automation before they are suitable for visible customer interaction. Others may work well if they are clearly bounded, fast, helpful, and supported by a smooth escalation path to a human.

4.4 Baseline impact modelling

4.4.1 Model framing

The baseline impact model was evaluated as a row-level regression model, where each observation represented one company–use-case combination and the target variable was the coded impact score on the original 0–10 scale. As described in the methodology chapter, the evaluation used grouped cross-validation at company level. This means that the reported out-of-fold predictions were made for companies that were held out from the corresponding training fold, rather than for new rows from companies already seen during training.

This distinction matters for the interpretation of the results. If the model had been evaluated only on randomly split rows, it could have learned company-specific patterns and then looked better than it actually was when applied to a new company. The grouped evaluation is more demanding, but also more aligned with the practical use of the framework, where the model should support reasoning about AI use cases for companies that may not have been present in the training data.

The model comparison was also evaluated using several metrics rather than only one. Mean absolute error (MAE) and root mean squared error (RMSE) measure ordinary point prediction error. Global Spearman correlation measures whether the model preserves the broad rank ordering of use-case impact across all out-of-fold predictions. In addition, the top-1 hit rate measures how often the model identifies one of the highest-impact use cases within a company. This mix of metrics is important because the later playbook use case is closer to prioritisation support than exact numerical forecasting.

4.4.2 Consolidated non-fit experiment set

The reported baseline impact modelling results consolidate the non-fit experiments from the modelling runs carried out during the project. In total, the consolidated report-facing result set contained 138 deduplicated non-fit experiment rows. During this consolidation, 508 fit-family rows were excluded and 6 duplicate rows were removed. The word non-fit is used here to mean that the feature set was based on directly coded company and use-case attributes, rather than on computed compatibility attributes such as data fit, capability fit, or sourcing fit.

This exclusion was made because the report-facing impact model should mainly show what could be learned from the observed structured attributes themselves. The computed fit attributes were useful during method development, but they also embed additional analytical assumptions. For the central result presentation, it was therefore more transparent to focus on feature sets that were closer to the original coded interview material. In practice, this also makes the model comparison easier to interpret, because a strong result can be tied back to ordinary company and use-case attributes rather than to a second derived fit layer.

4.4.3 Feature-set labels

Several of the strongest experiments use abbreviated feature-set names. These names were useful in the implementation, because the plots and saved artefacts had to remain readable, but they are not self-explanatory in the report. Table 4.6 therefore gives a short description of the main feature-set labels that appear in the reported top results. The table is intended as a readable glossary rather than as a complete technical specification. The full attribute composition of each report-mentioned feature set is provided in Appendix A.1.1.

Table 4.6: Feature-set labels used in the reported baseline impact modelling results.

Feature set	Description
`opf`	A 12-attribute operational feasibility set combining company data readiness, IT competence, change capacity, process governance maturity, and use-case-side burden attributes such as data need, data recency, build effort, process change burden, integration burden, and operational risk.
`uhlc`	A 13-attribute use-case-heavy and light-company set combining strategic and technical company attributes with solution type, KPI family, time to value, scope clarity, data need, data recency, build effort, process change burden, integration burden, and operational risk.
`tbc_dq_pgm`	A compact balanced-core variant that includes primary AI goal, data availability, data quality, IT competence, change capacity, process governance maturity, solution type, KPI family, time to value, data need, and build effort.
`tbc_dq_dacc`	A compact balanced-core variant similar to `tbc_dq_pgm`, but with data accessibility included and process governance maturity excluded.
`chlu`	A 13-attribute set combining channel profile, data availability, data accessibility, IT competence, change capacity, process governance maturity, sourcing preference, AI dependence tolerance, solution type, time to value, data need, and build effort.
`tiny_balanced_core`	A smaller nine-attribute core using primary AI goal, data availability, IT competence, change capacity, solution type, KPI family, time to value, data need, and build effort.
`reduced_raw`	The reduced raw feature set defined in the methodology configuration, using 19 selected company and use-case attributes but no computed fit attributes.
`full_raw`	The full collected raw feature set, using all active coded company and use-case attributes, but no computed fit attributes.

The most important point from this table is that the best-performing feature set, `opf`, is not the largest available feature set. Instead, it is a relatively focused set around feasibility, readiness, burden, and operational risk. This is already a useful result, because it suggests that the impact signal in the dataset is not only contained in broad strategic categories such as solution type or KPI family, but also in more practical implementation and readiness variables. For transparency, Appendix A.1.1 lists the exact included attributes for all feature sets that appear in the main top-25 ranking and in the extended top-72 appendix ranking.

4.4.4 Top-25 ranking

Table 4.7 shows the top 25 consolidated non-fit experiments ranked by MAE. To keep the table readable, long implementation names are shortened in the table itself, while the feature-set definitions are described in Table 4.6. The full top-72 ranking is provided in Appendix A.1.2. The best result is `opf` with histogram gradient boosting, with MAE 1.348, RMSE 1.736, global out-of-fold R^2 of 0.110, and global Spearman correlation of 0.372. The second-best result is `opf` with gradient boosting, with a very similar MAE of 1.352, a slightly lower RMSE of 1.714, a higher global out-of-fold R^2 of 0.132, and the strongest global Spearman value among the top-ranked models at 0.439.

This means that the absolute best model depends slightly on which metric is emphasised. Histogram gradient boosting has the lowest MAE, while ordinary gradient boosting has the strongest rank-ordering signal and the highest out-of-fold R^2 among the top rows. Since the practical purpose of the framework is prioritisation support, this difference is important. The model with the lowest absolute error is not necessarily the only model that is practically interesting.

Table 4.7: Top-25 consolidated non-fit baseline impact modelling experiments ranked by MAE.

Rank	Feature set	Model	MAE	RMSE	OOF R^2	Spearman	Top-1
1	OPF	Hist. gradient boosting	1.348	1.736	0.110	0.372	0.474
2	OPF	Gradient boosting	1.352	1.714	0.132	0.439	0.579
3	OPF	Random forest	1.394	1.780	0.064	0.275	0.421
4	UHLC	Gradient boosting	1.418	1.851	-0.012	0.333	0.579
5	TBC DQ + PGM	Hist. gradient boosting	1.421	1.864	-0.026	0.144	0.658
6	UHLC	SVR	1.437	1.871	-0.033	-0.015	0.684
7	TBC minus time to value	SVR	1.438	1.867	-0.029	0.014	0.632
8	OPF	Ridge	1.439	1.834	0.006	0.144	0.316
9	TBC plus data quality	SVR	1.445	1.873	-0.035	0.031	0.632
10	TBC DQ + data access	SVR	1.445	1.873	-0.036	0.017	0.684
11	Tiny balanced core	SVR	1.446	1.879	-0.043	0.019	0.632
12	TBC DQ + PGM	SVR	1.450	1.878	-0.041	-0.001	0.632
13	TBC plus scope clarity	SVR	1.453	1.878	-0.042	-0.003	0.632
14	Reduced raw	SVR	1.453	1.875	-0.038	-0.116	0.684
15	Full raw	SVR	1.457	1.887	-0.052	-0.165	0.579
16	OPF	SVR	1.459	1.866	-0.028	-0.029	0.526
17	TBC plus data quality	Hist. gradient boosting	1.459	1.874	-0.037	0.124	0.632
18	TBC minus build effort	SVR	1.460	1.899	-0.065	-0.066	0.579

Rank	Feature set	Model	MAE	RMSE	OOF R^2	Spearman	Top-1
19	CHLU	Random forest	1.461	1.833	0.008	0.127	0.632
20	OPF	Extra trees	1.470	1.875	-0.038	0.137	0.526
21	Reduced raw	Gradient boosting	1.474	1.882	-0.046	0.267	0.579
22	TBC DQ + data access	Hist. gradient boosting	1.476	1.884	-0.048	0.134	0.658
23	TBC DQ + data access	Random forest	1.476	1.878	-0.042	0.047	0.579
24	TBC DQ + PGM	Random forest	1.478	1.868	-0.030	0.068	0.579
25	CHLU	Elastic net	1.483	1.856	-0.017	0.040	0.583

There are three main patterns in Table 4.7. Firstly, the `opf` feature set appears repeatedly among the strongest models, especially in combination with tree-based ensemble methods. Secondly, the strongest simple model, ridge with `opf`, is still competitive in MAE terms, even though it is clearly weaker on some prioritisation metrics. Thirdly, the full raw feature set does not dominate the ranking. This is important because it suggests that adding more attributes does not automatically improve performance in this small-sample setting. A more focused feature set can reduce noise and may help the model use the limited data more effectively.

4.4.5 Top-25 metric summary

Figures 4.13–4.16 show the same top-25 result set from four different metric perspectives. The interpretation changes somewhat depending on whether the emphasis is on prediction error, explanatory performance, broad ranking, or within-company prioritisation behaviour. Appendix A.1.3 contains the corresponding plots for the wider top-72 result set.

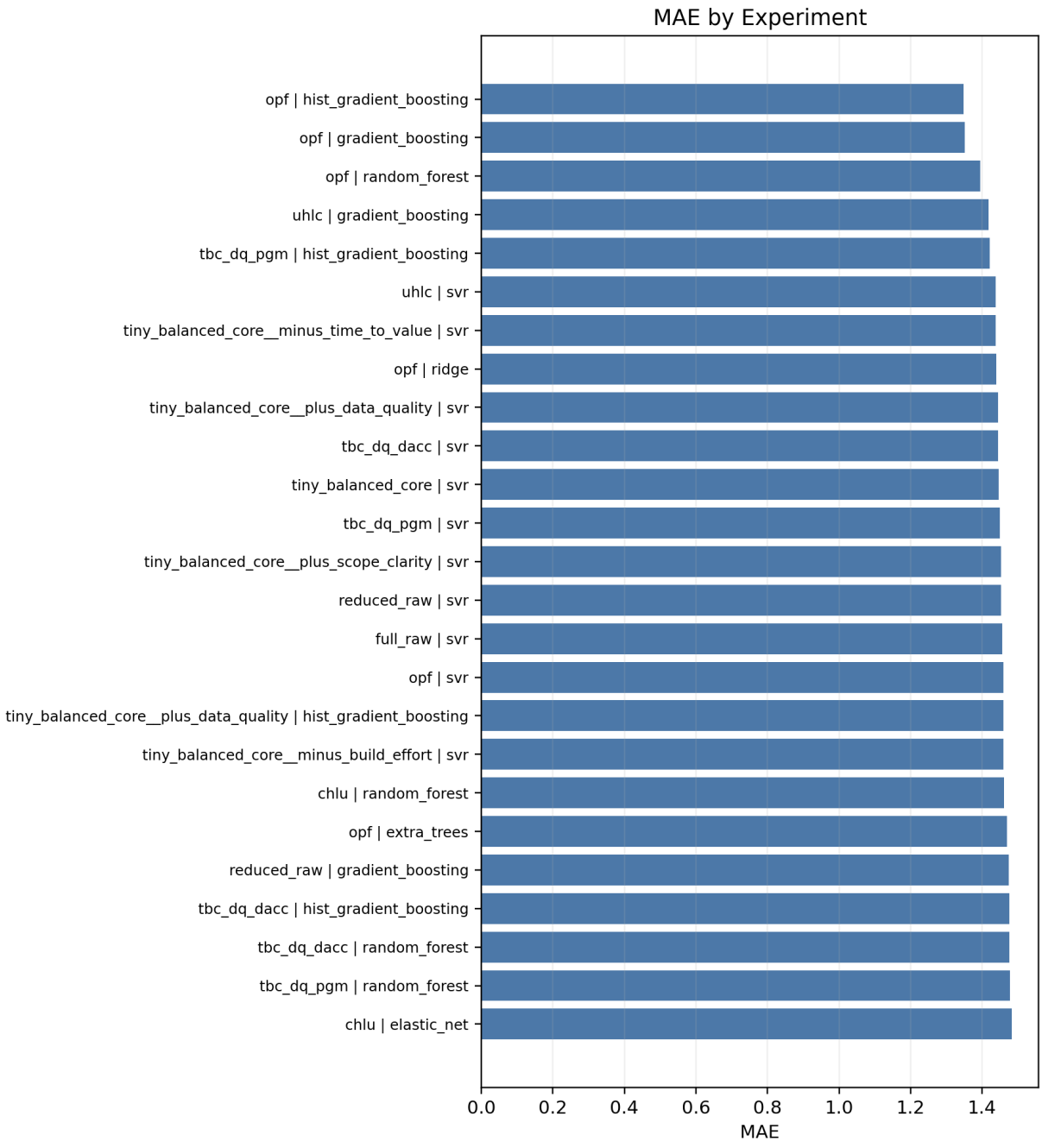


Figure 4.13: Top-25 consolidated non-fit impact modelling experiments ranked by mean absolute error. The plot shows that the strongest results are concentrated around `opf`, especially when paired with gradient-boosting-style ensemble models.

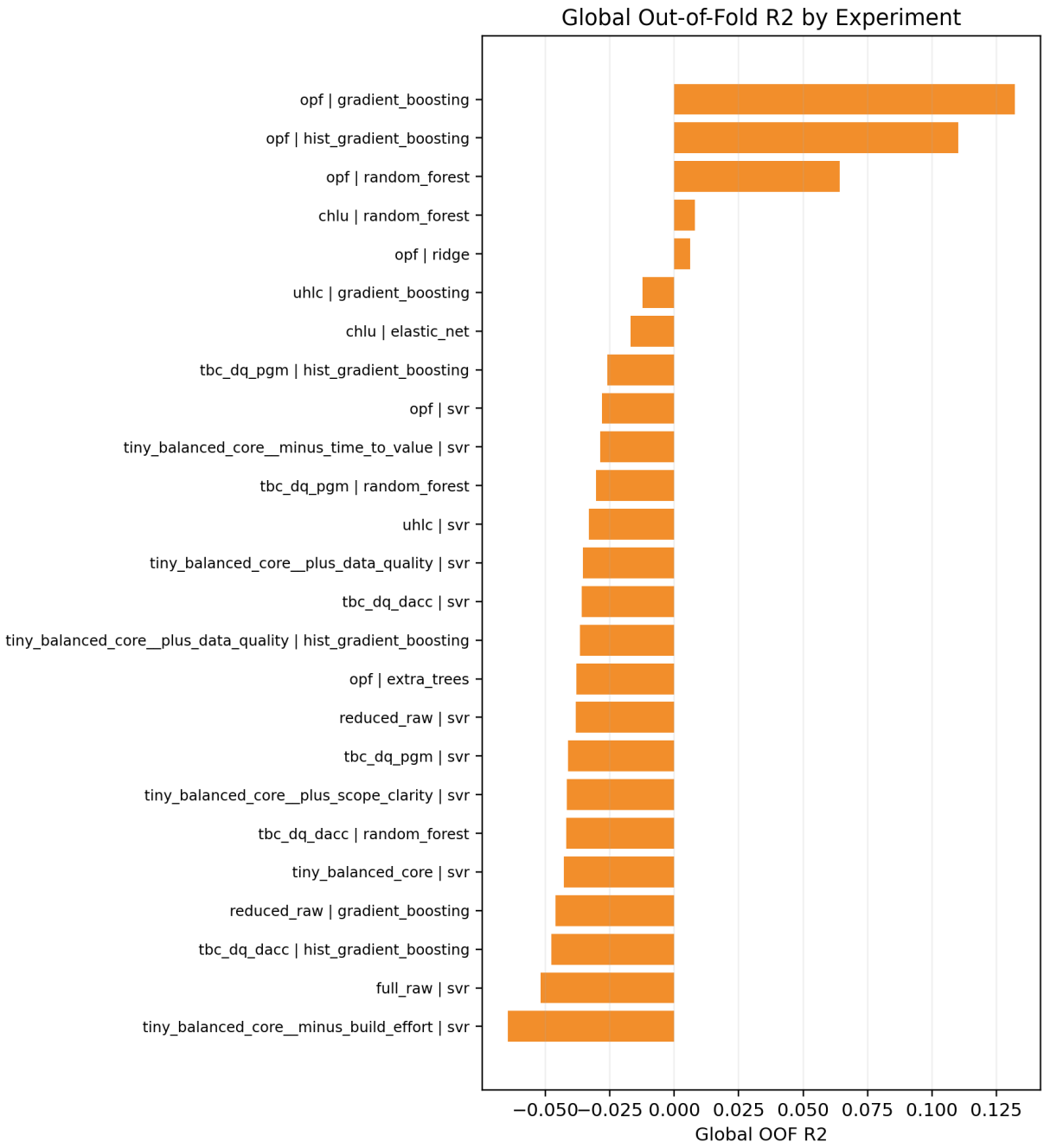


Figure 4.14: Global out-of-fold R^2 for the top-25 consolidated non-fit impact modelling experiments. Even the strongest values are modest, which indicates useful but limited explanatory performance in the small interview-derived dataset.

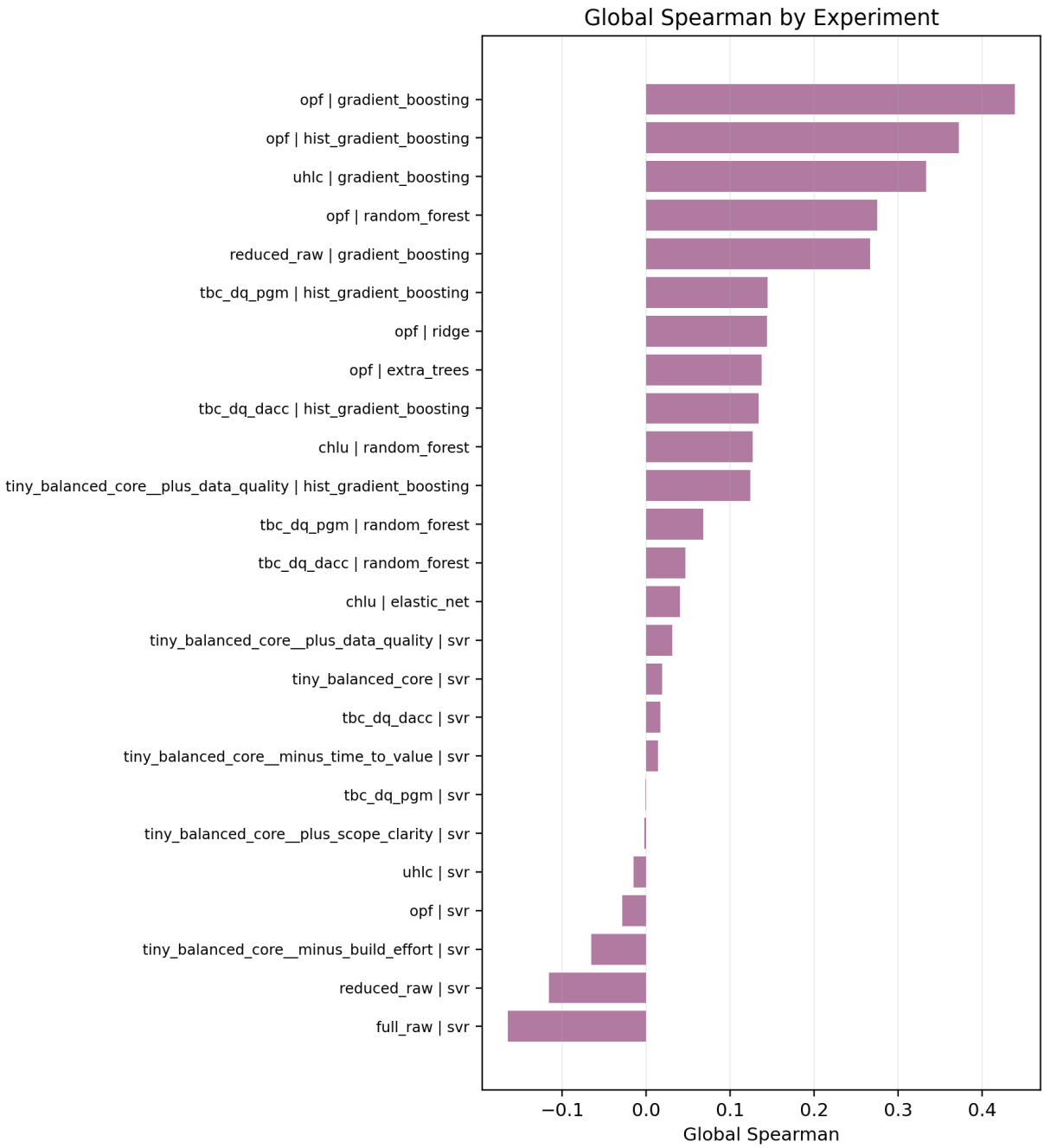


Figure 4.15: Global Spearman rank correlation for the top-25 consolidated non-fit impact modelling experiments. The figure reports how well each experiment preserves the relative ordering of use-case impact across the pooled out-of-fold predictions.

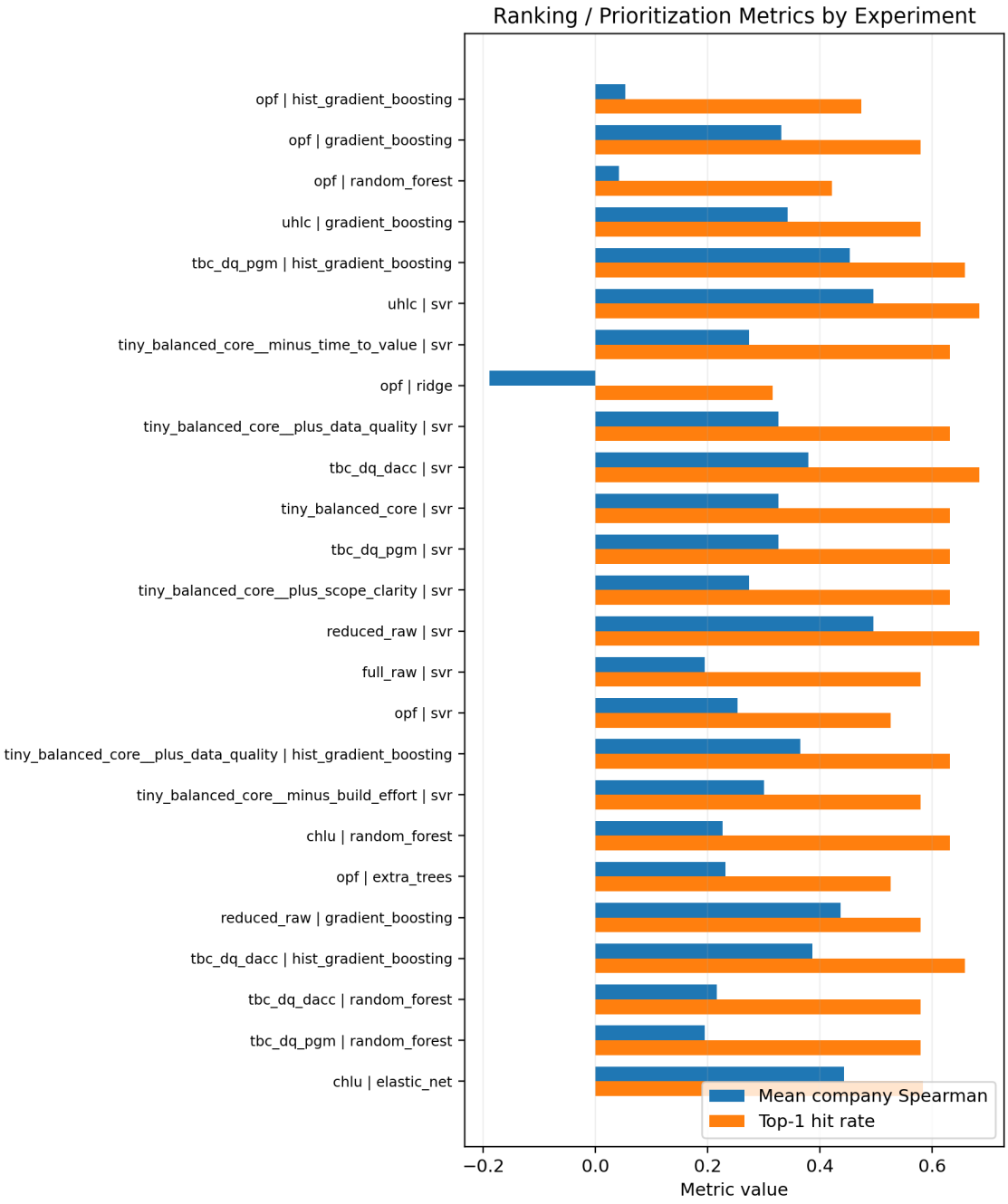


Figure 4.16: Ranking-oriented metrics for the top-25 consolidated non-fit impact modelling experiments. The figure supplements error-based metrics with prioritisation-oriented measures such as mean company Spearman and top-1 hit rate.

The MAE plot in Figure 4.13 shows that the best models are relatively close to each other in absolute error, but that the first three positions are all `opf` paired with tree ensembles. This reinforces the table result and suggests that the operational feasibility style of feature set is a robust candidate among the tested non-fit representations.

The R^2 plot in Figure 4.14 is more cautious. Only a few models have positive out-of-fold R^2, and even the strongest values remain low. This does not make the modelling result useless, but it does limit what can be claimed. The model appears to capture some structure in impact, but a large part of the variation is still unexplained. A possible reason is that the impact score depends on company-specific implementation details, strategic choices, and contextual factors that are difficult to capture fully in a structured coding sheet.

The Spearman and ranking plots in Figures 4.15 and 4.16 are especially relevant for the use-case prioritisation setting. For example, `opf` with gradient boosting has almost the same MAE as `opf` with histogram gradient boosting, but it has stronger global Spearman and a higher top-1 hit rate. In practice, this suggests that model selection should not only be based on MAE. If the model is used to support ranking and roadmapping, then the ability to preserve useful ordering is at least as important as reducing the average absolute prediction error.

4.4.6 Model-family comparison

The top-25 ranking gives the main performance view, but it can overemphasize model families that happen to appear many times in the strongest part of the ranking. Table 4.8 therefore shows the best non-fit result for each model family. This makes the comparison more balanced, because each model family is represented once.

Table 4.8: Best consolidated non-fit impact modelling result for each model family.

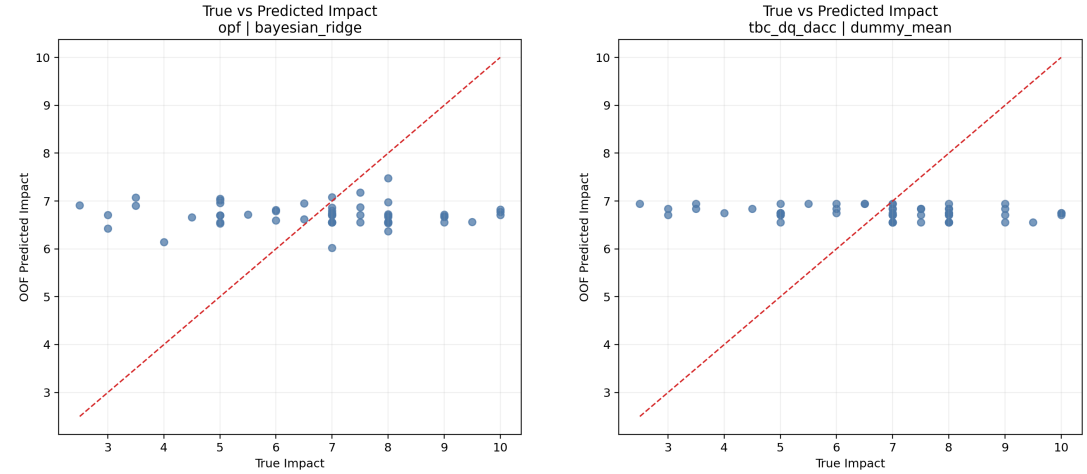
No.	Feature set	Model	MAE	RMSE	OOF R^2	Spearman	Top-1
1	OPF	Bayesian ridge	1.503	1.866	-0.028	-0.083	0.579
2	TBC DQ + data access	Dummy mean	1.514	1.880	-0.044	-0.277	0.461
3	CHLU	Elastic net	1.483	1.856	-0.017	0.040	0.583
4	OPF	Extra trees	1.470	1.875	-0.038	0.137	0.526
5	OPF	Gradient boosting	1.352	1.714	0.132	0.439	0.579
6	OPF	Hist. gradient boosting	1.348	1.736	0.110	0.372	0.474
7	TBC DQ + data access	Lasso	1.514	1.880	-0.044	-0.277	0.461
8	TBC plus scope clarity	MLP regressor	1.997	2.514	-0.866	0.055	0.526
9	CHLU	PLS regression	1.488	1.853	-0.014	0.193	0.632
10	OPF	Random forest	1.394	1.780	0.064	0.275	0.421
11	OPF	Ridge	1.439	1.834	0.006	0.144	0.316
12	UHLC	SVR	1.437	1.871	-0.033	-0.015	0.684

The model-family comparison gives a more nuanced picture than the top rows alone. The strongest tree-based models are clearly better than the dummy baseline and the weakest regularised models. However, the best ridge model is not far behind the weaker tree models in MAE, which means that the relationship is not simply that more complex models always dominate. Rather, the result suggests that the

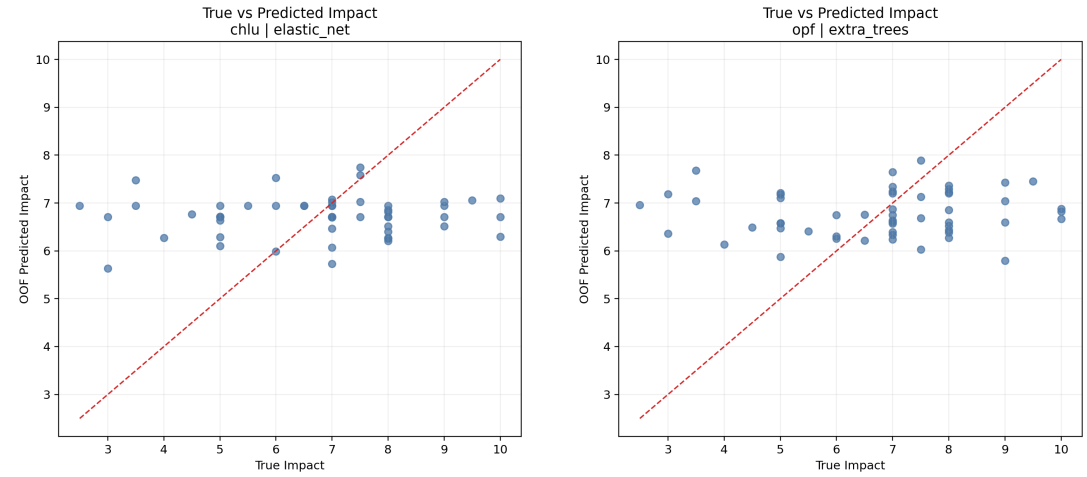
combination of a focused feature set and a model that can capture some nonlinearity is useful, while excessive flexibility is not necessarily helpful.

This can be seen most clearly in the multilayer perceptron result. The best MLP result has MAE 1.997 and a strongly negative out-of-fold R^2. This is much worse than the best tree-based models and also worse than the simple baselines. A likely explanation is that the dataset is too small and too heterogeneous for a neural-network-style model to learn stable patterns. In this setting, the MLP has enough flexibility to behave poorly, but not enough data to make that flexibility useful.

Figures 4.17–4.19 show true-versus-predicted diagnostics for the best result from each model family. These plots make the same point visually. Many models compress predictions towards the middle of the impact scale, and the models differ less by producing perfectly accurate predictions and more by how much useful variation they recover around that middle.

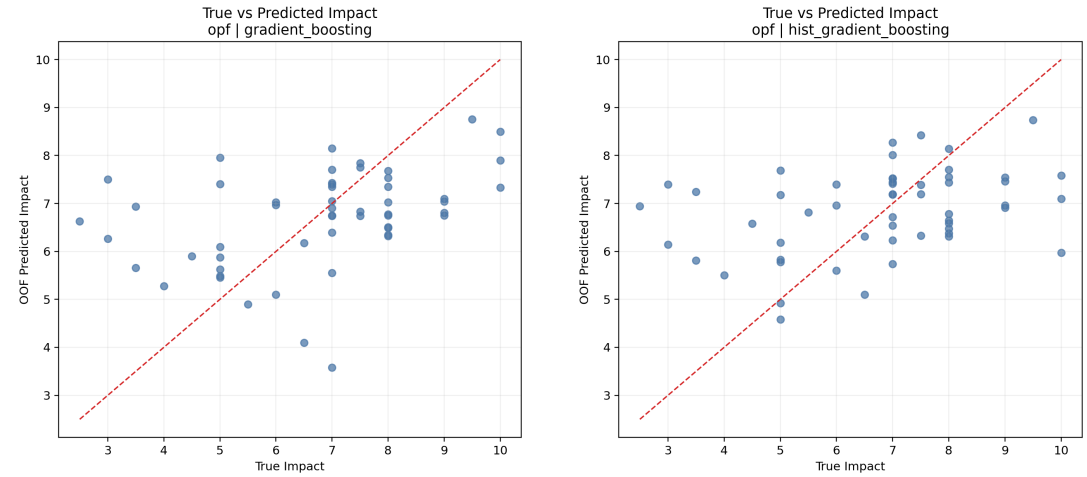


(a) **(b)**



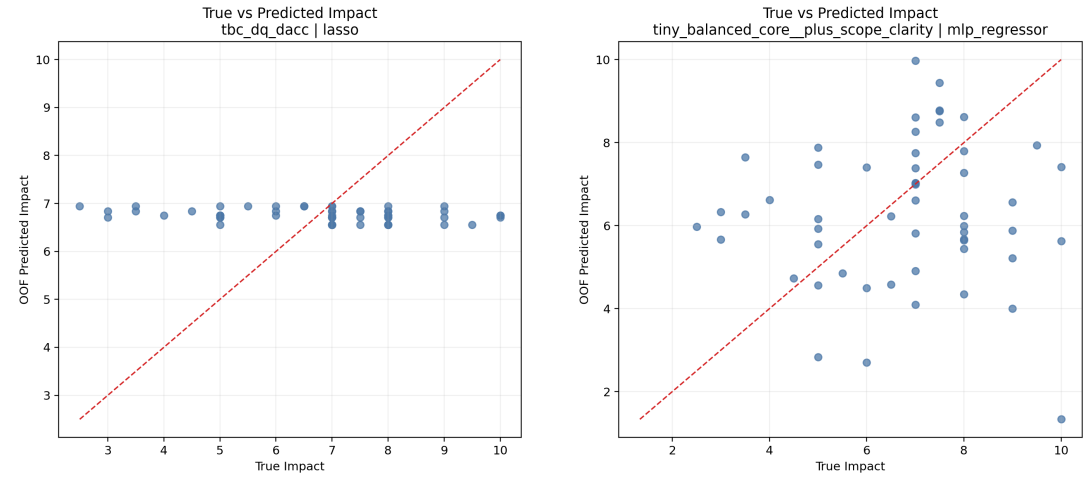
(c) **(d)**

Figure 4.17: True-versus-predicted impact diagnostics for the best non-fit result from four model families. The panels show that several simple or weakly regularised baselines compress predictions towards the middle of the impact scale.



(a)

(b)



(c)

(d)

Figure 4.18: True-versus-predicted impact diagnostics for the strongest gradient-boosting results and two contrasting model families. The boosting models with `opf` provide the strongest overall results, while the lasso and MLP examples illustrate the limitations of overly simple or data-hungry approaches in this small-sample setting.

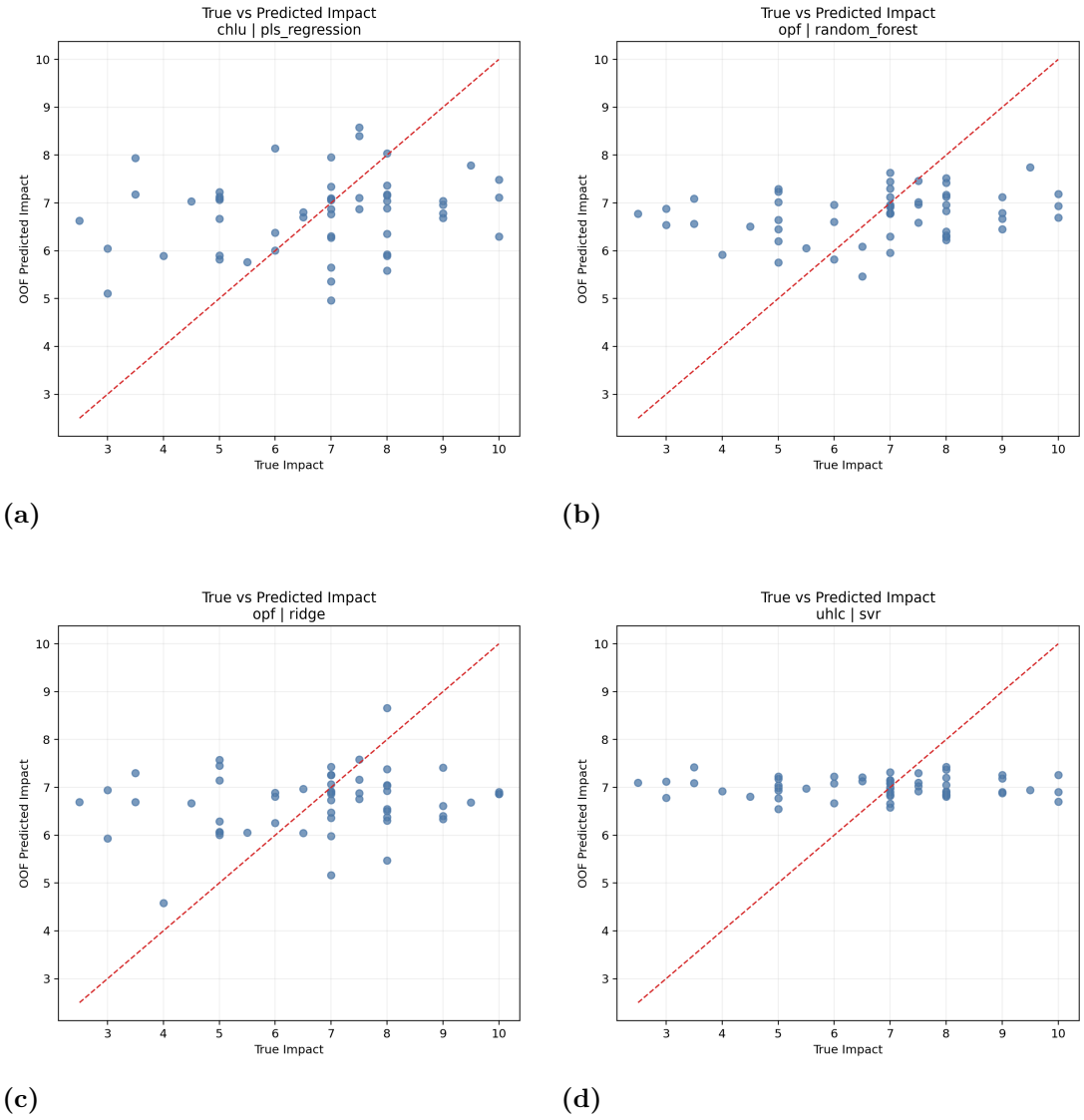


Figure 4.19: True-versus-predicted impact diagnostics for the remaining best-by-model-family results. The comparison shows that the strongest simple and kernel-based models can be competitive on MAE, but that the strongest tree ensembles recover more of the broad variation in observed impact.

4.4.7 Selected winner diagnostics

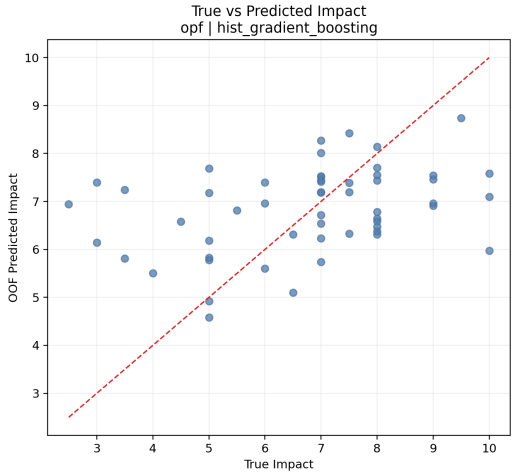
The final diagnostic inspection used seven selected non-fit experiments. These were not selected only to show the single best model, but to include the strongest observed MAE results and a few nearby alternatives that help interpret the difference between absolute error and ranking behaviour. Table 4.9 lists these experiments.

Table 4.9: Selected final non-fit impact modelling experiments used for detailed diagnostic inspection.

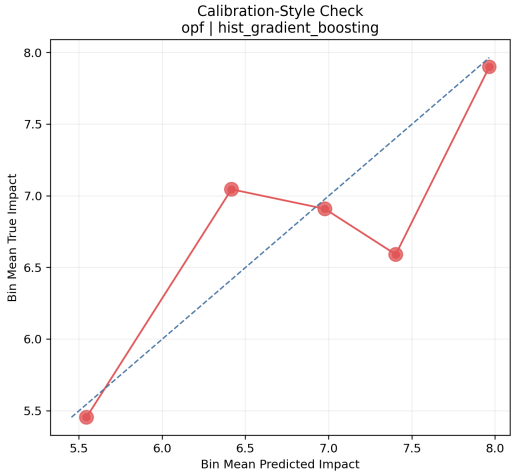
No.	Feature set	Model	MAE	RMSE	OOF R^2	Spearman	Top-1
1	OPF	Hist. gradient boosting	1.348	1.736	0.110	0.372	0.474
2	OPF	Gradient boosting	1.352	1.714	0.132	0.439	0.579
3	OPF	Random forest	1.394	1.780	0.064	0.275	0.421
4	UHLC	Gradient boosting	1.418	1.851	-0.012	0.333	0.579
5	TBC DQ + PGM	Hist. gradient boosting	1.421	1.864	-0.026	0.144	0.658
6	UHLC	SVR	1.437	1.871	-0.033	-0.015	0.684
7	TBC minus time to value	SVR	1.438	1.867	-0.029	0.014	0.632

Figures 4.20–4.26 show diagnostic triplets for these seven selected experiments. Each triplet contains a true-versus-predicted plot, a calibration-style plot, and a residual plot. The most visible pattern is that even the best models still compress predictions towards the middle of the impact scale. This is not surprising, because the model is trained on a small dataset with many different types of use cases and companies. Extreme impact scores are also more difficult to predict, because they often depend on contextual implementation details that are only partly represented in the structured features.

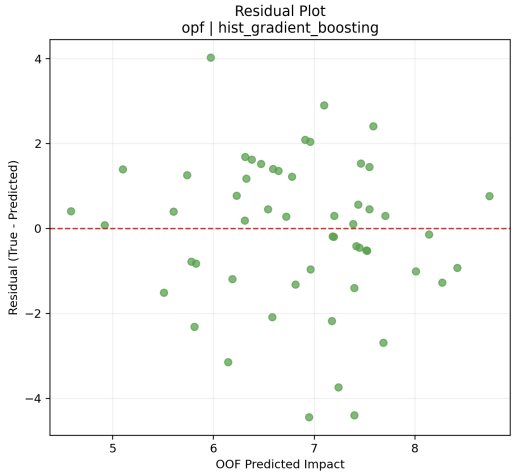
The diagnostic triplets also show why it is useful to inspect more than the MAE ranking. Some selected experiments are close in absolute error but differ more clearly in their rank-ordering behaviour and calibration. The tree-based ensemble models generally show more useful structure in the predicted values, while the SVR alternatives remain competitive on MAE but provide weaker global rank-ordering signal.



(a) True vs predicted



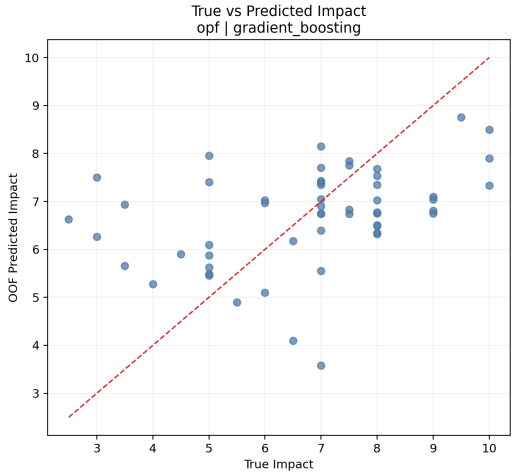
(b) Calibration-style check



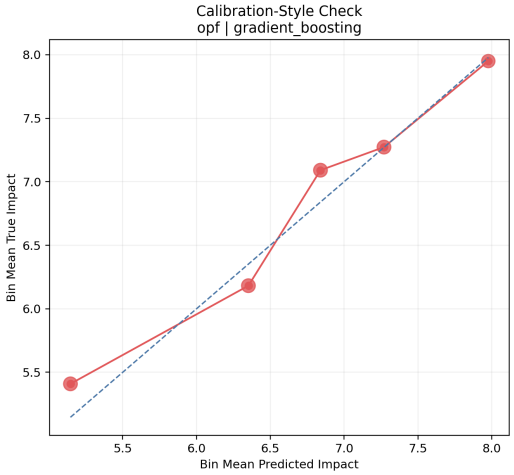
(c) Residuals

(d)

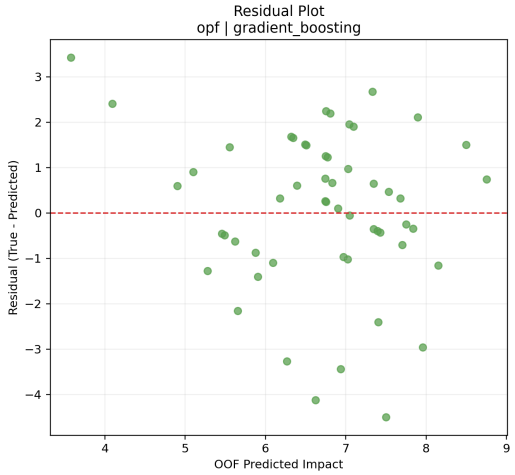
Figure 4.20: Diagnostic triplet for selected winner w01, `opf` with histogram gradient boosting. This experiment has the lowest observed MAE among the consolidated non-fit models and shows useful signal, while still compressing predictions towards the middle of the impact scale.



(a) True vs predicted



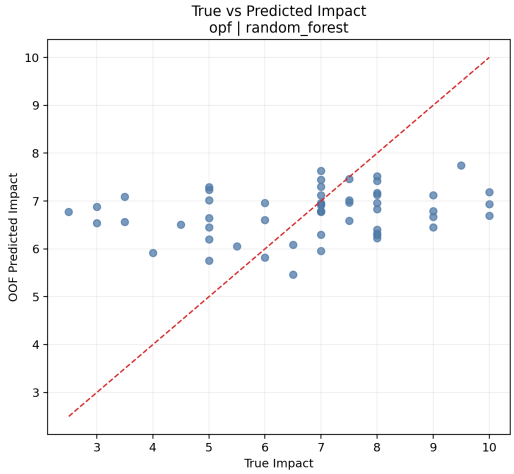
(b) Calibration-style check



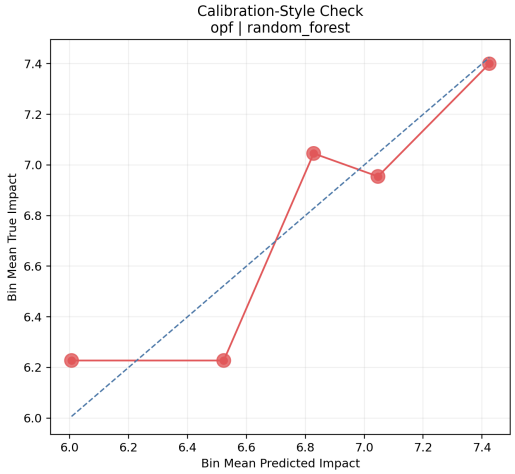
(c) Residuals

(d)

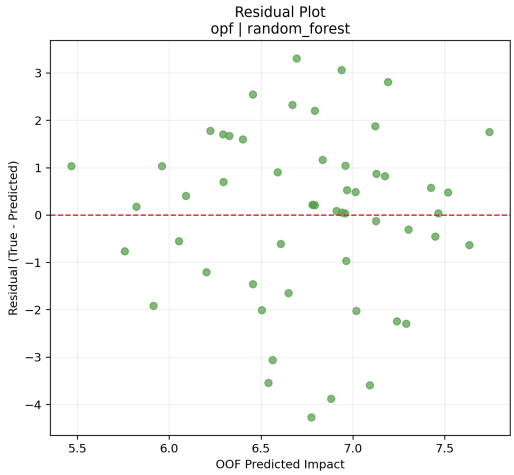
Figure 4.21: Diagnostic triplet for selected winner w02, `opf` with gradient boosting. This experiment has nearly the same MAE as w01, but stronger global Spearman correlation and out-of-fold R^2, making it particularly relevant from a prioritisation perspective.



(a) True vs predicted



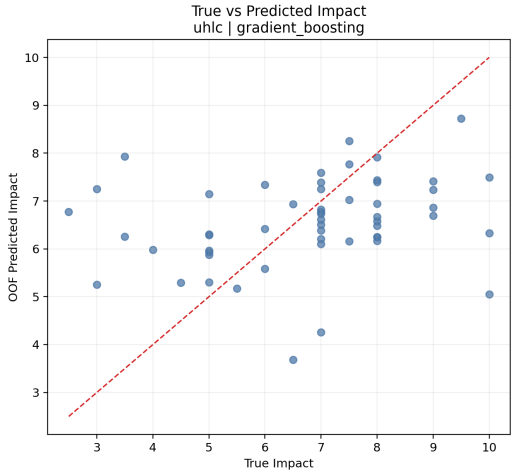
(b) Calibration-style check



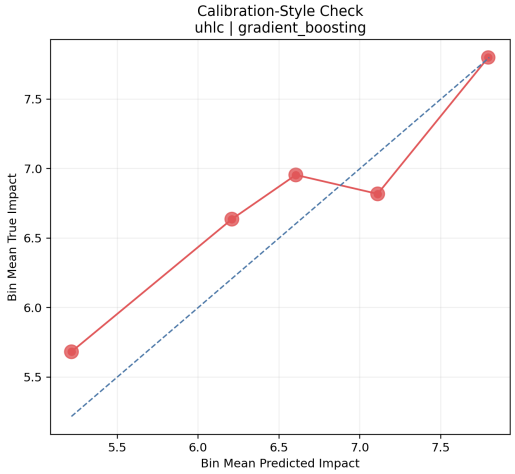
(c) Residuals

(d)

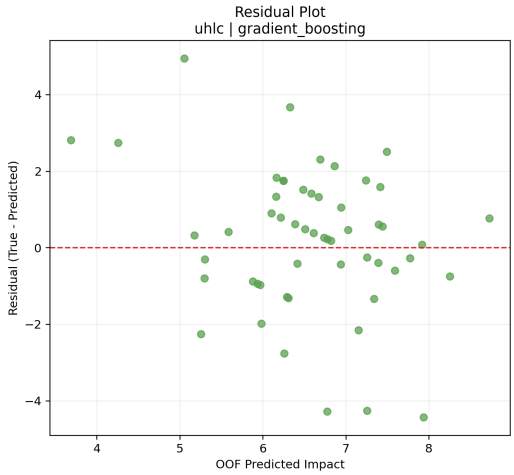
Figure 4.22: Diagnostic triplet for selected winner w03, `opf` with random forest. The model remains competitive in absolute error, but its weaker rank-ordering metrics show that similar MAE values can still imply different prioritisation usefulness.



(a) True vs predicted



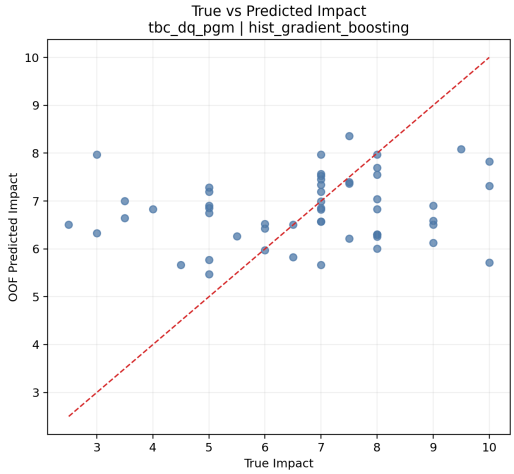
(b) Calibration-style check



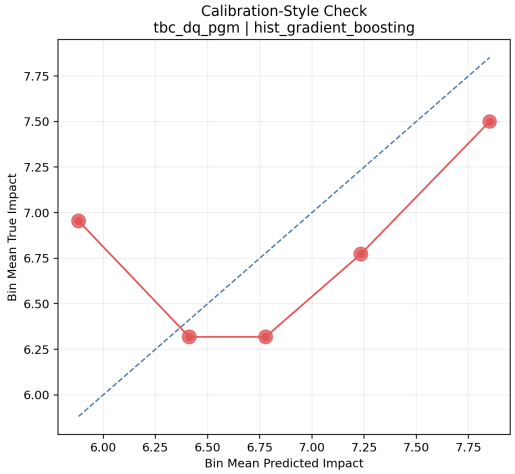
(c) Residuals

(d)

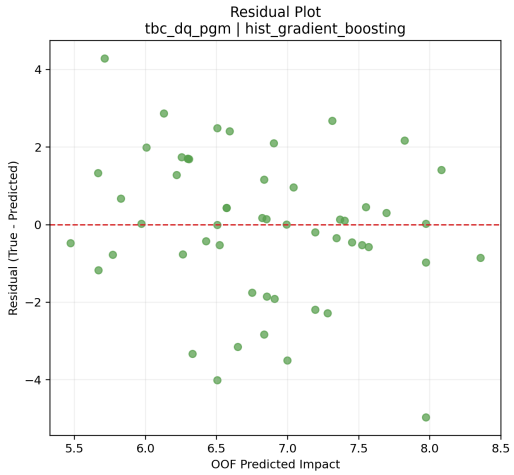
Figure 4.23: Diagnostic triplet for selected winner w04, `uhlc` with gradient boosting. This experiment shows that a more use-case-heavy feature set can remain competitive, although its out-of-fold R^2 is weaker than the strongest `opf` models.



(a) True vs predicted



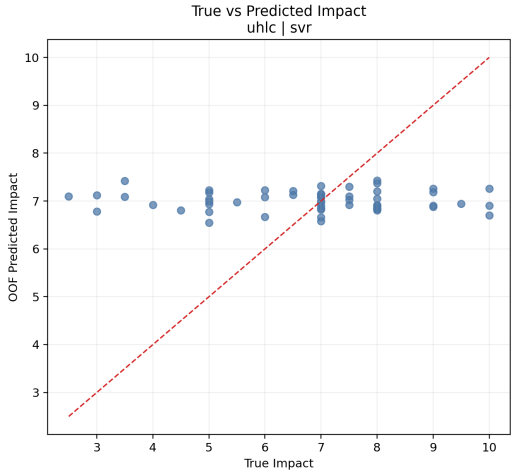
(b) Calibration-style check



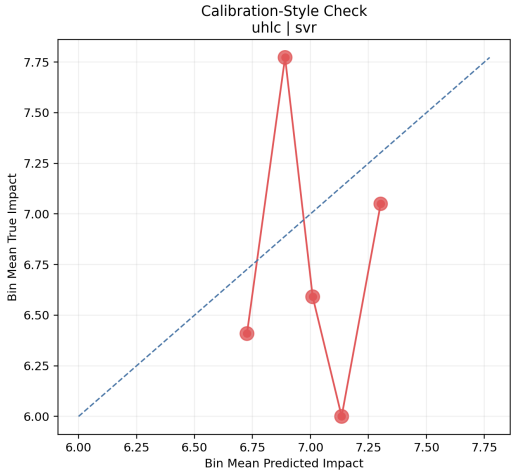
(c) Residuals

(d)

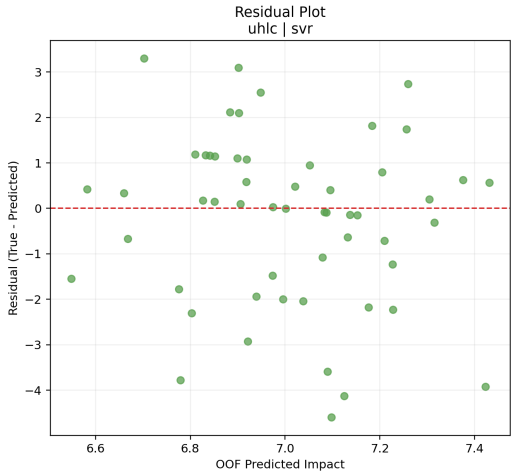
Figure 4.24: Diagnostic triplet for selected winner w05, `tbc_dq_pgm` with histogram gradient boosting. This compact balanced-core variant performs close to the strongest models and has the highest top-1 hit rate among the selected winners.



(a) True vs predicted



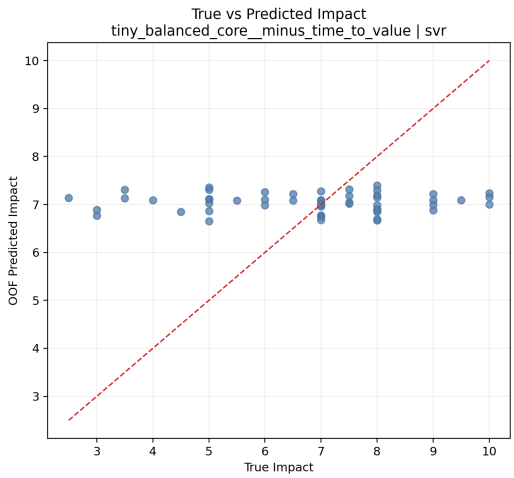
(b) Calibration-style check



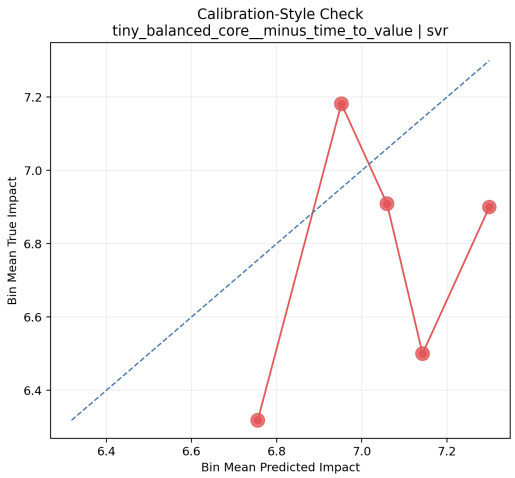
(c) Residuals

(d)

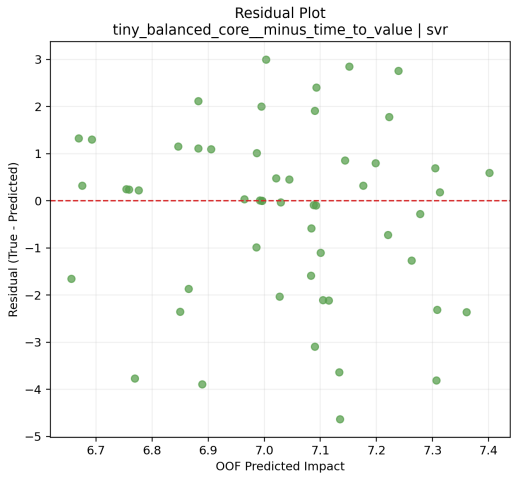
Figure 4.25: Diagnostic triplet for selected winner w06, `uhlc` with SVR. The experiment is close to the top models in MAE, but its weak global rank-ordering signal makes it less convincing as a prioritisation model.



(a) True vs predicted



(b) Calibration-style check



(c) Residuals

(d)

Figure 4.26: Diagnostic triplet for selected winner w07, `tiny_balanced_core` without time to value, with SVR. The result reinforces that a competitive MAE alone is not enough to treat a model as strong for prioritisation, since the ranking metrics remain limited.

4.4.8 Baseline model interpretation

Overall, the baseline impact modelling results suggest that the structured interview attributes contain a real but limited signal for impact-oriented prioritisation. The clearest result is that the strongest non-fit experiments are not based on the full raw feature set, but on a more focused operational feasibility feature set. This suggests that impact in the studied material is connected to the practical conditions around a use case: data readiness, IT competence, change capacity, governance maturity, data need, implementation burden, integration burden, and operational risk.

The second main result is that tree-based ensemble models, especially gradient boosting and histogram gradient boosting, extract more useful signal than most simpler baselines. This could be because the relationship between company readiness, use-case burden, and expected impact is not purely linear. For example, high data need may not be a problem if data availability and IT competence are also high, but it may become a problem when those company-side conditions are weak. A tree-based model can represent this type of interaction more naturally than a basic linear model.

At the same time, the results should not be overstated. The best out-of-fold R^2 values are still modest, and the diagnostic plots show clear prediction compression towards the middle of the impact scale. This means that the model should not be used as a stand-alone decision engine. Rather, it is better interpreted as one component in a broader analytical framework. It can help identify patterns, compare feature-set assumptions, and support prioritisation logic, but it still needs to be combined with qualitative interpretation, coefficient analysis, cluster-strength analysis, and managerial judgement.

The baseline modelling results therefore provide a useful transition to the next technical result layer. Since the best models appear to capture some impact signal, the next question is which attributes seem to contribute to that signal and whether those attributes are conceptually meaningful. That question is addressed through the coefficient and interpretability analysis.

4.5 Impact model interpretability

The baseline impact modelling results show that the structured interview attributes contain some predictive signal, but they do not by themselves explain where that signal comes from. Therefore, the next step was to inspect the fitted models more directly. This section presents the impact model interpretability analysis, using coefficient-based interpretation for selected linear models, SHAP-based interpretation for selected tree and boosting models, and permutation importance as a robustness-oriented check for nonlinear models.

The purpose of this section is not to establish causal drivers of AI use-case impact. The dataset is too small, too heterogeneous, and too interview-derived for that type of conclusion. Instead, the purpose is more modest and more practical. We want

to understand which attributes repeatedly appear to influence model behaviour, whether those attributes make substantive sense, and whether they can support transparent candidate rules for later cluster-strength experimentation. In other words, the interpretability analysis is used as a diagnostic layer between the baseline impact model and the later cluster-strength module.

4.5.1 Coefficient-based interpretation

The coefficient-based interpretation was carried out for the best selected non-fit experiment from each relevant linear model family. This included ridge regression, Bayesian ridge regression, elastic net, PLS regression, and lasso. The purpose was not only to compare the linear models as predictors, since that had already been done in the baseline modelling section, but also to inspect whether the fitted coefficients pointed towards conceptually meaningful attributes.

Figure 4.27 shows the attribute-level coefficient magnitude heatmap. The values are aggregated from encoded features back to conceptual attributes, meaning that one-hot and multi-hot columns belonging to the same original attribute are summarised together. This view shows which attributes the linear models used most strongly, regardless of whether the direction was positive or negative. The most visible attributes include `data_recency_requirement`, `data_quality`, `build_effort`, `it_competence`, `operational_risk`, and parts of the broader use-case attribute set.

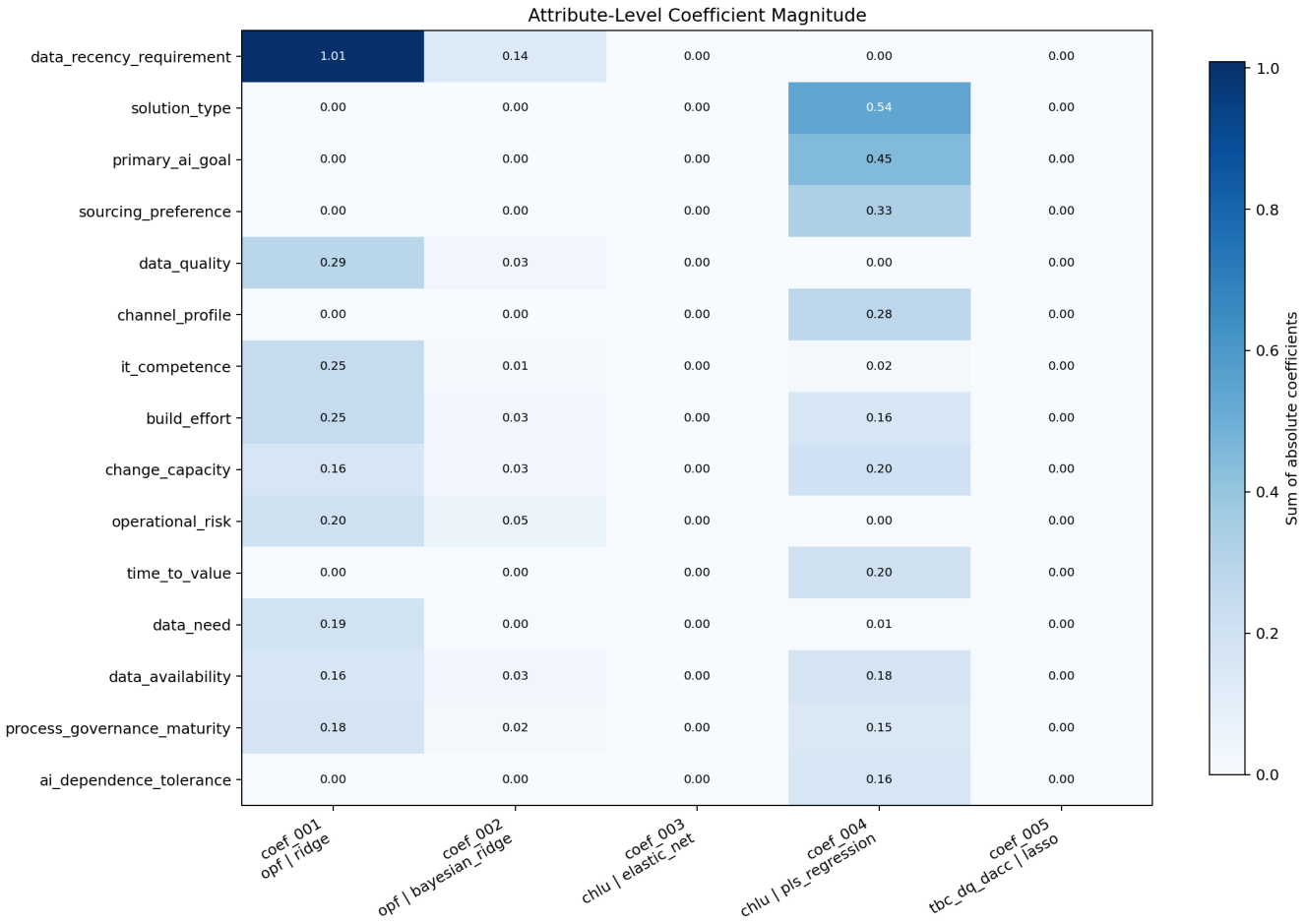


Figure 4.27: Attribute-level coefficient magnitude heatmap for selected linear impact models. The heatmap aggregates encoded feature coefficients back to conceptual attributes and therefore shows which attributes were used most strongly by each linear model, without showing whether the direction was positive or negative.

Figure 4.28 instead shows the signed coefficient aggregate. This plot is more directly related to interpretation, because redder values indicate a positive net direction and bluer values indicate a negative net direction in the fitted linear model. However, this view must be read carefully. For numerical or ordinal attributes, the interpretation is relatively direct: a positive coefficient means that a higher encoded value is associated with a higher predicted impact, while a negative coefficient means the opposite. For categorical attributes, the interpretation is less direct, because different encoded categories can point in different directions and partially cancel out when aggregated.

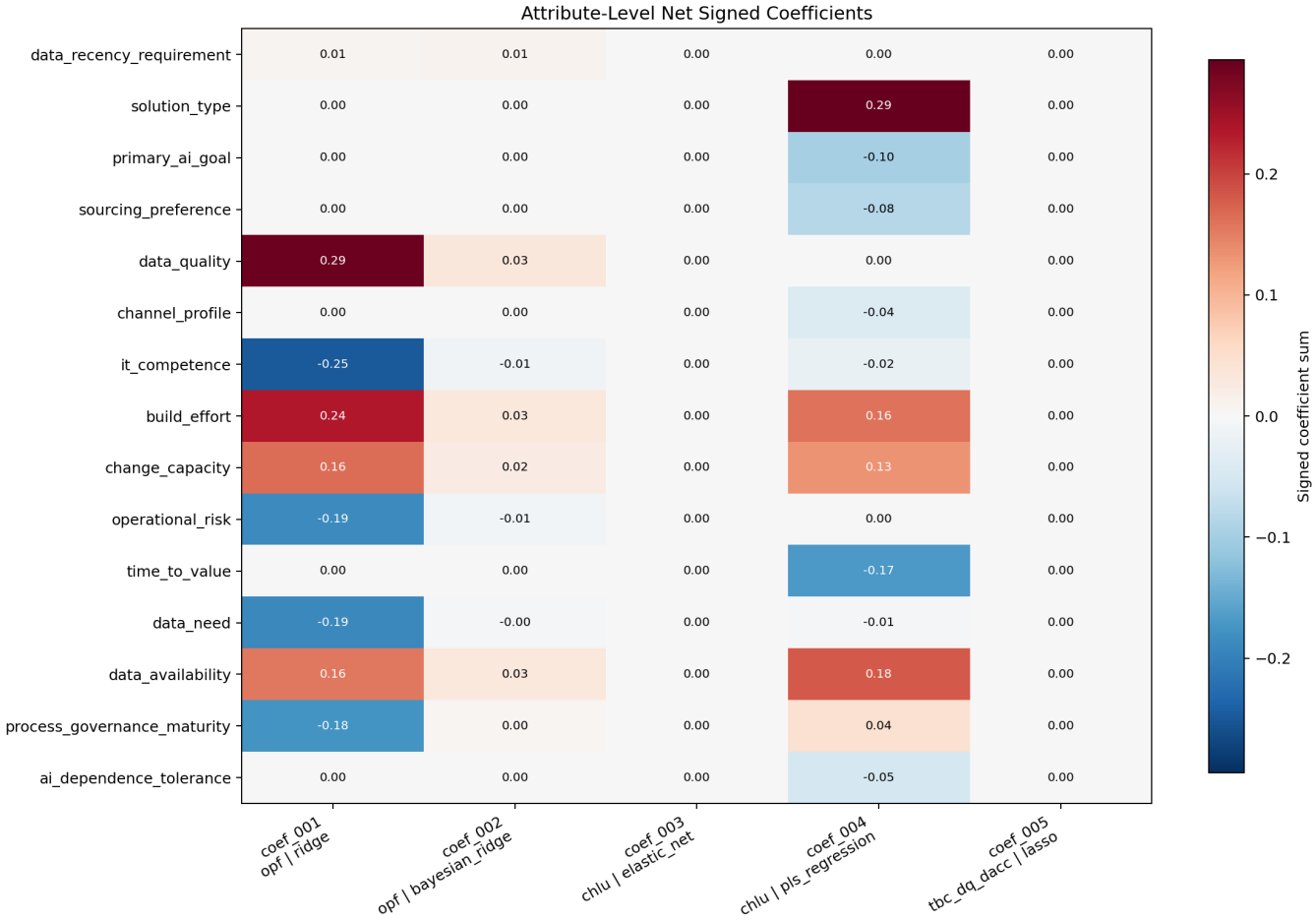


Figure 4.28: Attribute-level signed coefficient heatmap for selected linear impact models. Red indicates a positive net coefficient direction and blue indicates a negative net coefficient direction. For categorical attributes, the aggregate can hide category-level differences, so selected feature-level coefficient plots are also inspected.

The clearest positive coefficient signal came from `data_quality`, especially in the ridge model using the `opf` feature set. This is a plausible result, because better data quality should make an AI use case easier to implement and more likely to produce useful output. `build_effort` also appeared with a positive direction in some of the linear models. This should not be interpreted as effort being good by itself. A more reasonable interpretation is that more ambitious or more developed use cases may also have higher potential impact. In practice, this means that effort can partly act as a proxy for use-case depth or scope.

The `data_recency_requirement` attribute was also important, but it illustrates why the feature-level plots are necessary. In the aggregate signed heatmap, the attribute can look close to neutral in some models, because its encoded categories point in different directions. Figure 4.29 shows this more clearly for the ridge model. In that model, the realtime category has a positive coefficient, while the static and no-data categories have negative coefficients. This is a more meaningful interpretation than simply saying that data recency is positive or negative overall.

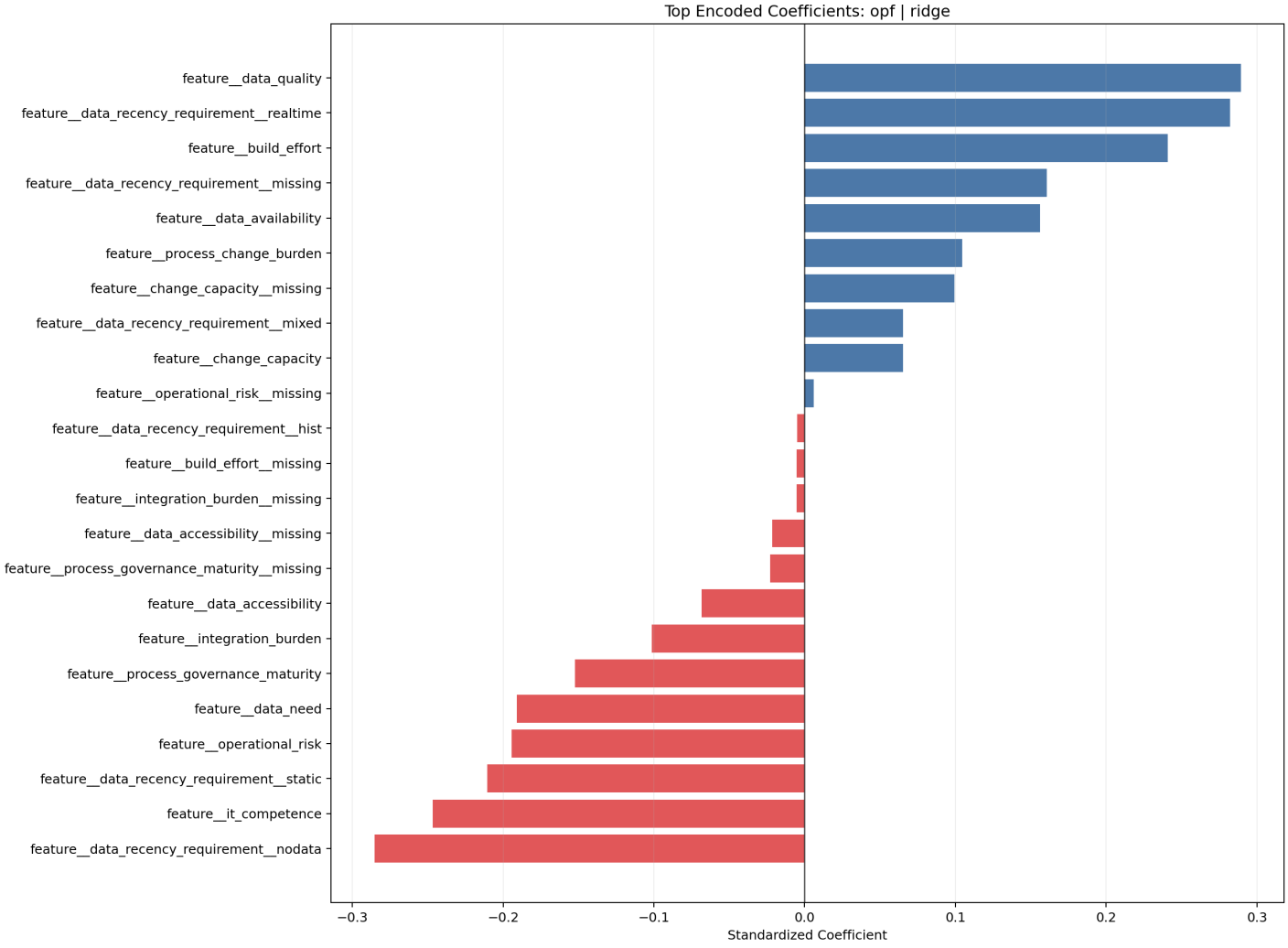


Figure 4.29: Feature-level coefficient plot for the `opf` ridge model. The plot shows why categorical attributes should be inspected at encoded-feature level: for example, realtime data recency is positive, while static or no-data recency categories are negative.

Figure 4.30 gives a second feature-level view, this time for the PLS regression model using the `chlu` feature set. This model was not the strongest overall baseline model, but it is useful diagnostically because it gives a different linear view of the same modelling problem. In this view, `time_to_value` appears with a negative direction, which means that longer time to value is associated with lower predicted impact. This is conceptually plausible, since use cases that take longer to produce value are less attractive from a prioritisation perspective, all else equal.

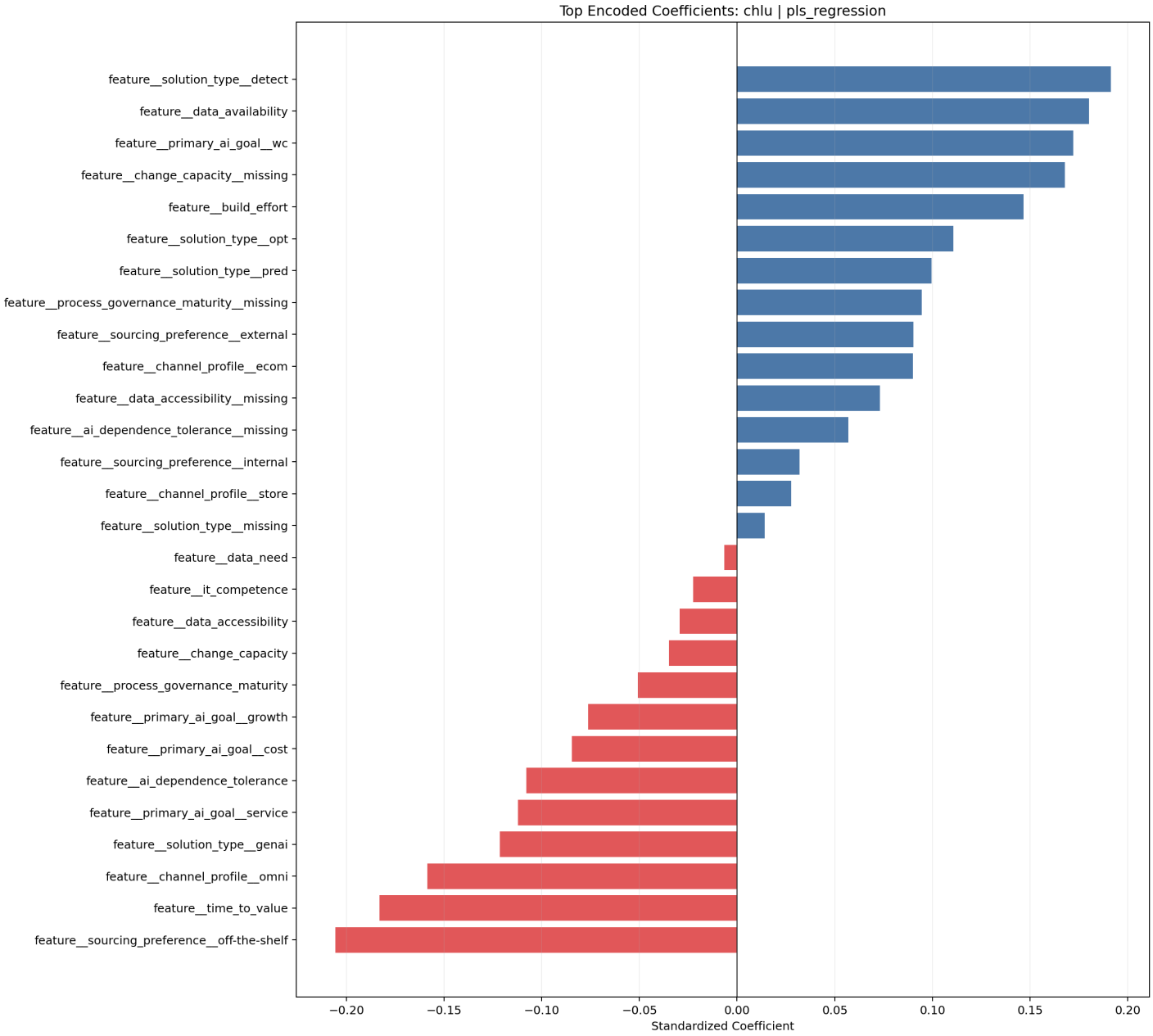


Figure 4.30: Feature-level coefficient plot for the `chlu` PLS regression model. This plot is mainly used as a diagnostic complement to the ridge view, especially for the negative direction of `time_to_value`.

At the same time, not all coefficient signals were straightforward. For example, `it_competence` and `process_governance_maturity` appeared with negative directions in some linear specifications. This goes against what one would normally expect, since stronger IT competence and stronger process governance should generally support implementation. A possible explanation is that high-capability companies in the sample may also have selected more ambitious or more difficult use cases, which could make the simple coefficient direction misleading. Another possible explanation is collinearity between company attributes in a small dataset. Therefore, the coefficient results were used cautiously. They are useful for identifying candidate patterns, but they are not strong enough to be treated as causal explanations.

For transparency, the remaining generated coefficient-level plots are included in Appendix A.2. The main text only shows the feature-level plots that are most directly relevant for interpreting the candidate adjustment rules.

4.5.2 SHAP-based interpretation

The SHAP-based analysis was used for the selected nonlinear tree and boosting models. These models were more difficult to interpret directly from fitted parameters, but they were among the strongest baseline impact models. SHAP was therefore used to inspect how much each encoded feature contributed to the fitted prediction, and the encoded feature contributions were then aggregated back to conceptual attributes.

Figure 4.31 shows the attribute-level mean absolute SHAP values. A higher value means that the attribute changed the model prediction more strongly on average. It does not say whether the attribute increased or decreased the prediction; it only says that the model relied on it. The most consistent attributes in this view are `data_recency_requirement`, `data_quality`, `build_effort`, `process_change_burden`, `time_to_value`, and `operational_risk`. This is interesting because several of these attributes also appeared in the coefficient analysis or in the strongest `opf` feature-set family.

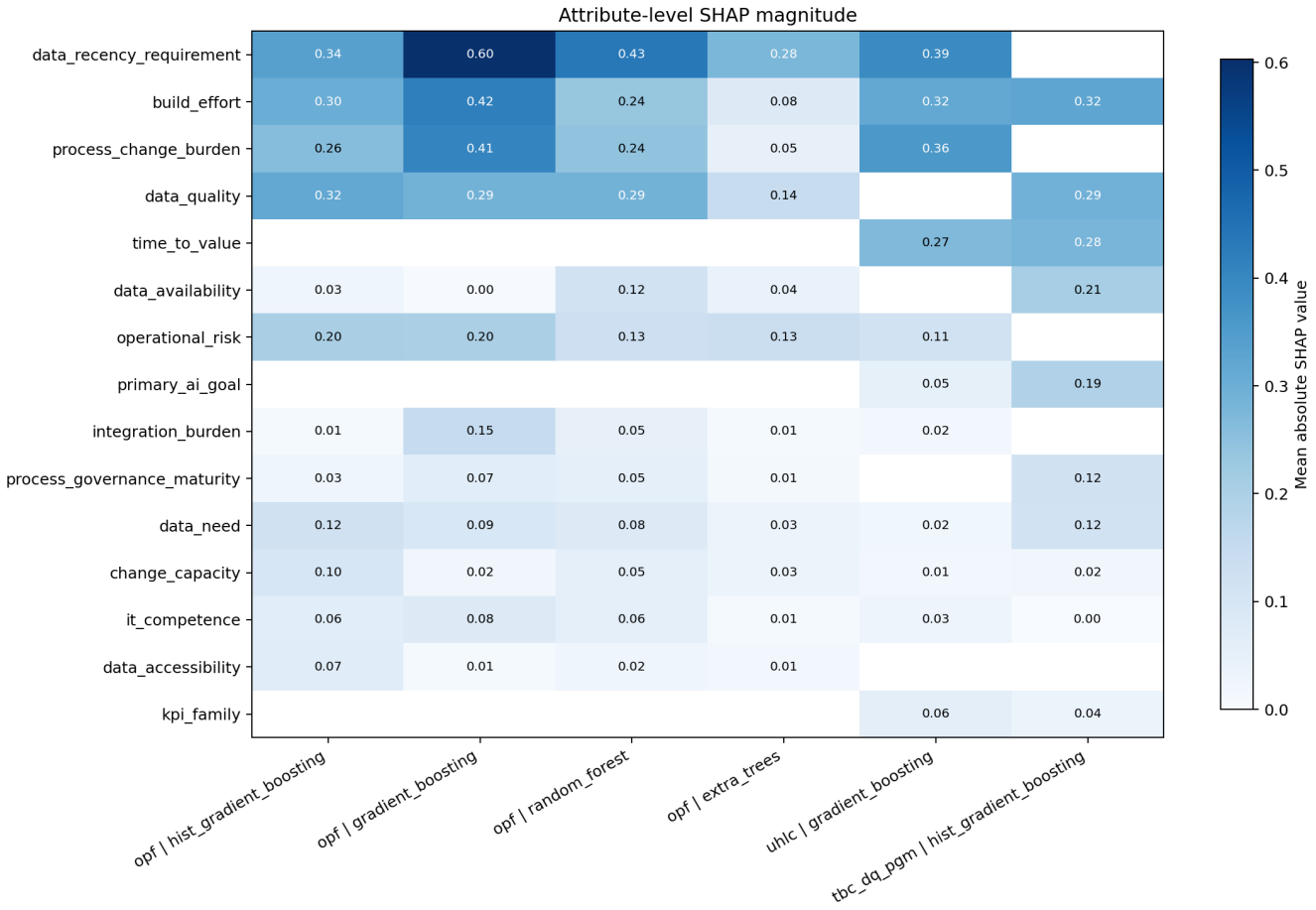


Figure 4.31: Attribute-level mean absolute SHAP values for selected nonlinear impact models. Higher values indicate that the attribute contributed more strongly to fitted predictions on average, but not whether the direction was positive or negative.

Figure 4.32 adds a directional diagnostic by looking at how encoded feature values relate to SHAP values. This makes the interpretation more useful, but also more fragile. For ordinal attributes such as `data_quality`, a positive direction is relatively easy to read: higher data quality tends to push the prediction upward in the fitted model. For categorical attributes such as `data_recency_requirement`, the direction is again more complicated because different categories can behave differently. In the detailed SHAP outputs, realtime data recency was the dominant positive category, while static and no-data categories tended to be weaker or negative.

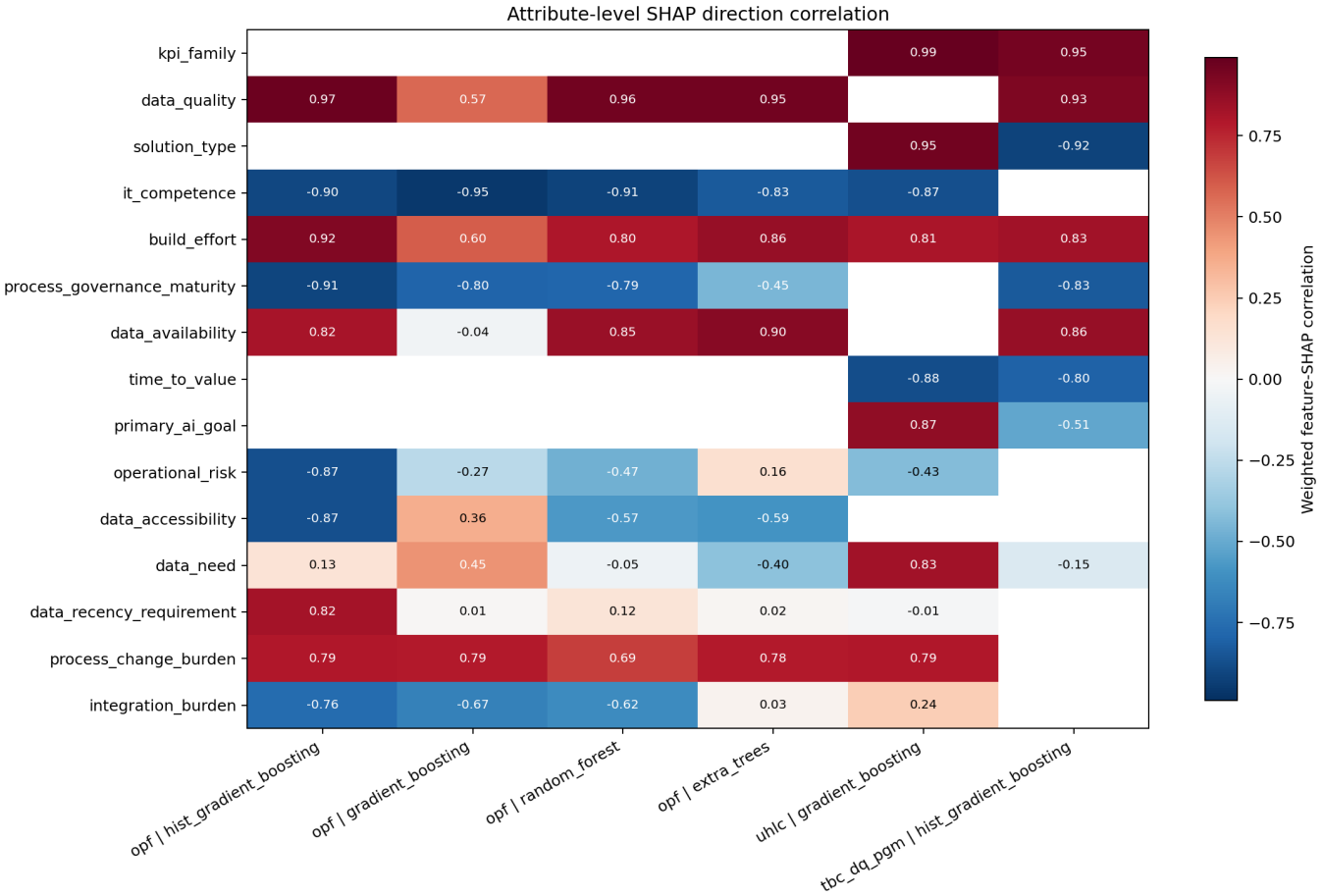


Figure 4.32: Attribute-level SHAP direction diagnostic for selected nonlinear impact models. The direction is based on the relationship between encoded feature values and SHAP values. It is most directly interpretable for ordinal attributes and should be read more cautiously for categorical attributes.

The SHAP results support several intuitive interpretations. Higher `data_quality` is associated with higher predicted impact. Shorter `time_to_value` is generally more favourable, while longer time to value is less favourable. Higher `operational_risk` tends to reduce predicted impact. These patterns are conceptually reasonable and align with how a prioritisation framework would normally treat these variables.

Other SHAP patterns require more careful interpretation. Both `build_effort` and `process_change_burden` appear important and often positive. This should not be read as saying that implementation burden is good. Rather, these variables may capture the fact that larger, deeper, or more transformational use cases have more potential to create impact if they are actually implemented well. In this sense, the model may be picking up the difference between small, easy use cases and more substantial use cases. However, this is still a model-derived interpretation, not a direct causal statement.

4.5.3 Permutation-importance robustness check

Permutation importance was used as a robustness-oriented complement to SHAP. The logic is different from SHAP: instead of decomposing predictions into feature contributions, permutation importance measures how much model error increases when the values of a conceptual attribute are shuffled. If shuffling an attribute causes a large increase in MAE, the model was relying on that attribute for prediction.

Figure 4.33 shows the attribute-level permutation importance heatmap. The most important point is the overlap with the SHAP magnitude view. `data_recency_requirement`, `data_quality`, `build_effort`, `process_change_burden`, and `time_to_value` repeatedly appear as important across several nonlinear models. This overlap makes these attributes more credible as candidate interpretability signals than if they had appeared only in one model or one method.

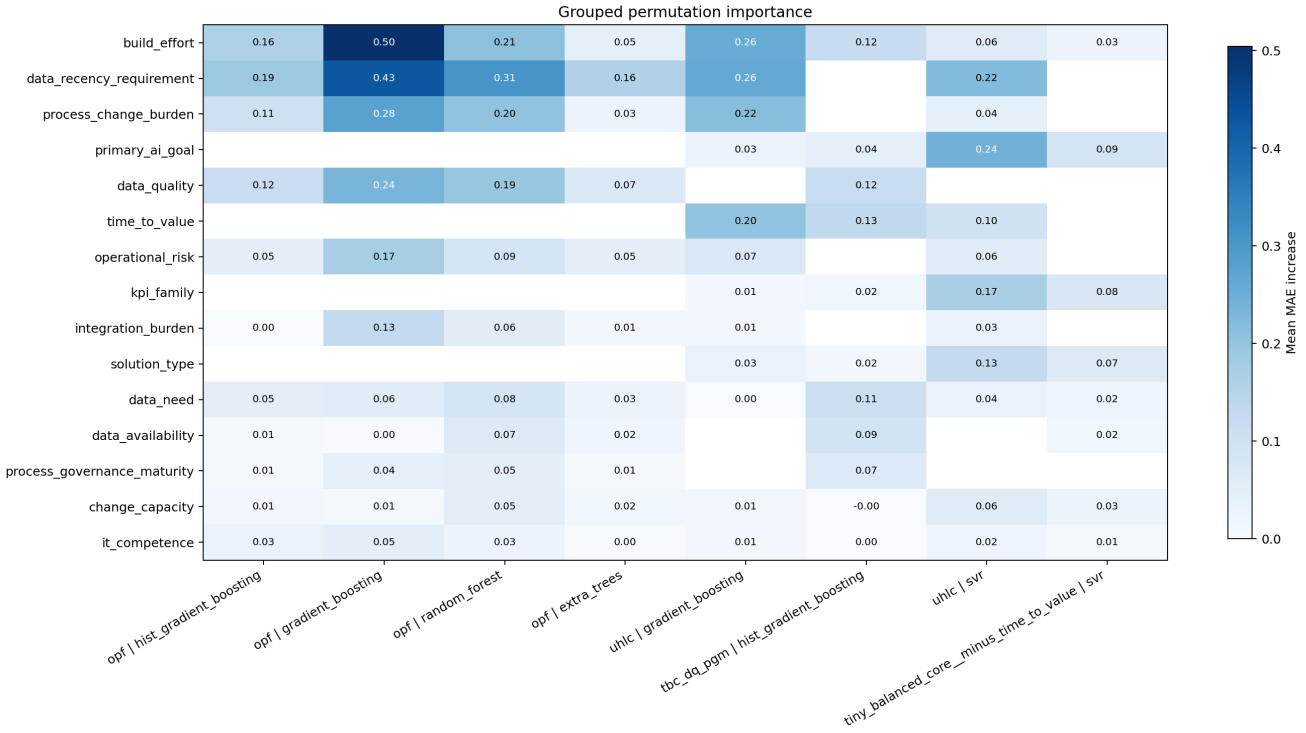


Figure 4.33: Attribute-level permutation importance for selected nonlinear impact models. Values represent the mean increase in MAE after permuting the attribute group. Higher values indicate that the fitted model relied more strongly on that attribute.

Permutation importance was also used for selected SVR models, because the selected SVR experiments were close to the stronger models in MAE and one of them had a relatively high top-1 hit rate. However, SHAP was not used for the SVR models, because model-agnostic SHAP would have added more complexity and would have been easier to overinterpret in this small-sample setting. The MLP model was also left out of the interpretability analysis, since its baseline modelling performance was weak and it did not provide a useful report-facing case for explaining impact.

It is important to note that permutation importance does not provide direction. If `data_quality` has high permutation importance, this means that the model relied on data quality, but not whether high data quality pushed predictions upward or downward. Therefore, permutation importance was mainly used to check whether attributes highlighted by SHAP and coefficients were also important to prediction performance.

4.5.4 Candidate manual-adjustment logic

The final step in the interpretability analysis was to translate selected model-derived signals into candidate manual-adjustment settings for later cluster-strength experimentation. This translation was deliberately conservative. The adjustment rules should not be understood as learned causal weights, and they should not be interpreted as direct recommendations for companies. Instead, they are transparent hypotheses that can be tested in the configurable experimental part of the cluster-strength module, where prototype completion, manual adjustments, smoothing, and related settings can be varied explicitly. They apply only to predicted missing values in the cluster-strength completion layer, not to observed impact values.

Two candidate sets were defined. The first set was mainly coefficient-derived and is shown in Table 4.10. It emphasises attributes where the signed coefficient results and feature-level coefficient plots gave a reasonably interpretable direction. The strongest examples are higher data quality, realtime data recency, lower operational risk, shorter time to value, and the positive but cautious build-effort signal.

Table 4.10: Coefficient-derived candidate manual-adjustment settings for later cluster-strength experimentation.

Candidate condition	Delta
`data_recency_requirement` = `realtime`	+0.20
`data_recency_requirement` = `static`	−0.15
`data_recency_requirement` = `nodata`	−0.20
`data_quality` ≥ 2.5	+0.15
`data_quality` ≤ 1.5	−0.15
`operational_risk` ≥ 2.5	−0.15
`operational_risk` ≤ 1.5	+0.10
`time_to_value` ≤ 1.5	+0.08
`time_to_value` ≥ 2.5	−0.10
`build_effort` ≥ 2.5	+0.08

The second set was derived from the combined SHAP and permutation-importance view and is shown in Table 4.11. This set is slightly broader, because `build_effort`, `process_change_burden`, `data_quality`, `time_to_value`, `operational_risk`, and `data_recency_requirement` all appeared repeatedly in the nonlinear interpretation outputs. The main difference compared with the coefficient-derived set is that `process_change_burden` is included here. This is because it appeared clearly in the SHAP and permutation views, while it was not strong enough in the coefficient view to justify inclusion there.

Table 4.11: SHAP- and permutation-derived candidate manual-adjustment settings for later cluster-strength experimentation.

Candidate condition	Delta
`data_recency_requirement` = `realtime`	+0.25
`data_recency_requirement` = `static`	−0.20
`data_recency_requirement` = `nodata`	−0.20
`data_quality` ≥ 2.5	+0.20
`data_quality` ≤ 1.5	−0.20
`build_effort` ≥ 2.5	+0.15
`process_change_burden` ≥ 2.5	+0.15
`time_to_value` ≤ 1.5	+0.12
`time_to_value` ≥ 2.5	−0.15
`operational_risk` ≥ 2.5	−0.15

These adjustment sets are intentionally small in magnitude relative to the original 0–10 impact scale. This is because the interpretability results are not strong enough to justify large mechanical changes. The adjustment values are better understood as sensitivity settings. If a completed cluster-strength run becomes much more or much less interesting only because of these small manual deltas, then that result should be

treated carefully. On the other hand, if a relationship remains visible under both no-adjustment and adjustment-enabled settings, then that gives stronger support for the relationship as an interpretable pattern.

4.5.5 Interpretive caution

Overall, the interpretability analysis suggests that several practical attributes matter in the fitted impact models. The most repeated signals concern data recency, data quality, build effort, process change burden, time to value, and operational risk. These attributes are also conceptually meaningful in a retail AI roadmapping setting, since they describe both the quality of the data foundation and the difficulty or depth of implementation.

However, the results also show why this layer should be interpreted cautiously. Some coefficient directions were counterintuitive, especially for company-side capability attributes such as `it_competence` and `process_governance_maturity`. This does not necessarily mean that IT competence or process governance reduce impact. A more likely explanation is that the small sample, correlated attributes, and different ambition levels across companies make simple directional interpretation unstable. For example, companies with higher capability may have discussed more ambitious use cases, which can make their use cases harder to predict from structured attributes alone.

This caution is also consistent with the baseline modelling results. The best models captured some signal, but the out-of-fold R^2 values were still modest and the diagnostic plots showed prediction compression towards the middle of the impact scale. Therefore, the interpretability analysis should be read as an exploratory diagnostic layer. It helps identify plausible candidate levers, but it does not prove that these levers determine impact. The next technical result layer, the cluster-strength module, can then test how company-grouping and use-case-grouping relationships behave under both cleaner observed-only settings and more experimental settings in which prototype completion, manual adjustments, smoothing, and related configuration choices are enabled.

4.6 Cluster-strength results

The cluster-strength module was used to move from row-level impact modelling towards more interpretable company-grouping × use-case-grouping relationships. In this section, an edge refers to one company group combined with one use-case group, and the edge strength should be read as an estimated mean impact on the original 0–10 impact scale. This is a different result object than the baseline impact model. The baseline model evaluates whether individual company–use-case rows can be predicted, while the cluster-strength module asks whether broader company and use-case archetypes form stronger or weaker combinations.

The results are divided into two parts. The first part, presented here, is the

configuration-enabled cluster-strength analysis. It is mainly a technical sensitivity experiment, because prototype completion, smoothing, manual adjustments, weighting, and neighbour count are varied systematically. The second part, which is treated separately later, is the observed-only cluster-strength analysis. That later part is closer to direct study evidence, because it relies more narrowly on observed interview-coded use cases. This distinction is important. The configuration-enabled analysis helps show how the framework behaves, but it should not by itself be treated as the final source of business conclusions.

4.6.1 Configuration-enabled analysis

4.6.1.1 Experiment setup

The configuration-enabled analysis combined selected company grouping strategies with selected use-case grouping strategies across nine configuration variants. In total, this produced 900 successful cluster-strength runs, corresponding to 10 company grouping strategies, 10 use-case grouping strategies, and 9 variants. The company grouping strategies were retail industry type, size class, sourcing preference, supply-chain complexity, change capacity, IT competence, primary AI goal, channel profile, data readiness tier, and execution-readiness tier. The use-case grouping strategies were organisational support area, AI solution pattern, business department owner, human-in-the-loop need, solution type, KPI family, use-case family, delivery source, implementation-complexity tier, and data need by recency requirement.

The purpose of the experiment was not to find one single final configuration. Rather, the purpose was to test how sensitive the cluster-strength outputs were to important modelling choices. The prototype feature space used for this analysis was the full raw structured attribute representation, excluding computed fit attributes. This choice was made to keep the prototype completion layer close to the original coded company and use-case attributes, instead of letting the completion layer depend on the derived fit attributes discussed earlier in the methodology.

Table 4.12 summarises the nine variants. The baseline variant used prototype completion with five neighbours, uniform neighbour weighting, global smoothing with pseudo-count 5, and no manual adjustment rules. The other variants then changed one or a few settings relative to this baseline in order to test specific sensitivities. Two variants applied the candidate manual-adjustment settings from the interpretability analysis, one based on the coefficient-derived settings in Table 4.10 and one based on the SHAP/permutation-derived settings in Table 4.11. Other variants disabled prototype completion, disabled smoothing, changed smoothing strength, changed distance weighting, or changed the number of neighbours.

Table 4.12: Configuration variants used in the configuration-enabled cluster-strength analysis.

Variant	Configuration meaning
Baseline	Prototype completion enabled with five neighbours, uniform neighbour weighting, global edge smoothing with pseudo-count 5, and no manual adjustments.
Coeff. adj.	Same as the baseline, but with the coefficient-derived manual-adjustment settings from Table 4.10.
SHAP/perm. adj.	Same as the baseline, but with the SHAP- and permutation-derived manual-adjustment settings from Table 4.11.
Observed only	Prototype completion disabled, no neighbour weighting used, global smoothing still applied with pseudo-count 5, and no manual adjustments.
No smoothing	Same as the baseline, but with global edge smoothing disabled.
Smoothing 2	Same as the baseline, but with a weaker smoothing pseudo-count of 2 instead of 5.
Inv.-distance	Same as the baseline, but with inverse-distance neighbour weighting instead of uniform neighbour weighting.
$k = 3$	Same as the baseline, but with three neighbours instead of five. Uniform neighbour weighting, smoothing pseudo-count 5, and no manual adjustments are retained.
$k = 10$	Same as the baseline, but with ten neighbours instead of five. Uniform neighbour weighting, smoothing pseudo-count 5, and no manual adjustments are retained.

The observed-only variant in this section should not be confused with the stricter observed-only study-insight runs discussed later. Here, the observed-only variant still sits inside the same configuration-enabled batch and still uses the same smoothing logic as the other variants. It is included mainly to show what happens when prototype completion is switched off inside the otherwise comparable experimental setup.

4.6.1.2 Variant sensitivity

Table 4.13 compares the nine variants at a high level. The first three metric columns summarise the distribution of edge strengths across the 100 grouping-pair runs within each variant. The mean absolute change column compares each variant to the baseline at edge level. The final column compares the observed top-5 supported edges in each variant with the corresponding baseline top-5 lists. This gives a compact view of whether the strongest supported edges stay similar when the configuration settings are changed.

Table 4.13: Variant-level sensitivity comparison for the configuration-enabled cluster-strength experiments.

Variant	Mean edge	Std. edge	Range	Mean abs. change vs baseline	Observed top-5 overlap vs baseline
Baseline	6.80	0.28	1.22	0.00	1.00
Coeff. adj.	6.90	0.31	1.36	0.10	0.90
SHAP/perm. adj.	6.98	0.34	1.48	0.17	0.88
Observed only	6.76	0.34	1.24	0.23	0.61
No smoothing	6.82	0.40	1.82	0.14	0.94
Smoothing 2	6.81	0.33	1.42	0.06	0.97
Inv.-distance	6.82	0.29	1.28	0.04	0.92
$k = 3$	6.81	0.35	1.50	0.13	0.81
$k = 10$	6.80	0.21	0.90	0.13	0.80

Note. The zero-observed share was 0.48 for every variant. This is the share of edge cells with no directly observed company–use-case rows behind them, and it is constant because all variants use the same observed data support. What changes across variants is how the framework completes, smooths, weights, or adjusts the edge values. The observed top-5 overlap compares the highest-ranked supported edges in each variant with the baseline variant.

The first result from Table 4.13 is that the manual-adjustment variants increase the average edge-strength level. The baseline mean edge is 6.80, while the coefficient-adjusted variant has mean 6.90 and the SHAP/permutation-adjusted variant has mean 6.98. This is expected, because several of the manual rules add positive deltas for attributes that appear in stronger use cases, such as high data quality, realtime data recency, and high build effort. However, the increase should not be interpreted as the adjusted variant being better. It simply means that the adjustment layer moves the completed values upward under the selected rules.

The second result is that smoothing and neighbour count have visible effects on the spread of edge strengths. The no-smoothing variant has the largest average edge range, 1.82, and the largest standard deviation, 0.40. This means that smoothing compresses the edge distribution, which is also the purpose of smoothing in sparse settings. In contrast, the $k = 10$ variant has the lowest range, 0.90, and the lowest standard deviation, 0.21. This is also reasonable, because using more neighbours makes the prototype predictions less local and more averaged. In practice, this shows a clear tradeoff: stronger smoothing and larger neighbourhoods reduce instability, but they can also reduce the separation between stronger and weaker edges.

The third result is that the observed-only variant changes the top-edge structure more than most other variants. Its observed top-5 overlap with the baseline is 0.61, which is lower than the other variants. This suggests that prototype completion materially affects which edges appear strongest, even when the same observed data support is used. This is not necessarily a problem, because the prototype layer is intended to complete sparse matrices. However, it reinforces the reason for keeping

configuration-enabled results separate from the cleaner observed-only results. Almost half of the possible edge cells have no directly observed row behind them, so the completed edge structure should be interpreted as framework behaviour under explicit assumptions, rather than as direct evidence alone.

4.6.1.3 Top and stable supported edges

The next step was to inspect which edges became strongest under the different variants. A simple top-edge list can be misleading if it is based on sparse cells. Therefore, the first edge-level table applies a minimum observed support threshold of five observed rows. Table 4.14 reports the strongest supported edge in each variant under this threshold.

Table 4.14: Strongest supported edge per configuration variant, requiring at least five observed rows.

Variant	Strongest supported edge	Strength	n
Baseline	Channel profile: E-commerce-led x Use-case family: Planning / Inventory	7.45	5
Coeff. adj.	Channel profile: E-commerce-led x Use-case family: Planning / Inventory	7.67	5
SHAP/perm. adj.	Channel profile: E-commerce-led x Use-case family: Planning / Inventory	7.84	5
Observed only	Execution-readiness tier: High x Delivery source: Internal	7.57	14
No smoothing	Channel profile: E-commerce-led x Use-case family: Planning / Inventory	7.49	5
Smoothing 2	Channel profile: E-commerce-led x Use-case family: Planning / Inventory	7.47	5
Inv.-distance	Channel profile: E-commerce-led x Use-case family: Planning / Inventory	7.46	5
$k = 3$	Channel profile: E-commerce-led x Use-case family: Planning / Inventory	7.49	5
$k = 10$	Channel profile: E-commerce-led x Use-case family: Planning / Inventory	7.35	5

The most visible pattern in Table 4.14 is that the combination of e-commerce-led companies and planning/inventory use cases appears as the strongest supported edge in almost all variants. The only exception is the observed-only variant, where high execution readiness combined with internal delivery becomes the strongest edge. This result is informative, but it should still be read cautiously. The e-commerce-led planning/inventory edge has five observed rows, which is enough to pass the support filter used here but still not a large sample. Therefore, it is better interpreted as an interesting supported pattern than as a robust conclusion on its own.

Table 4.15 applies a stricter support filter of at least ten observed rows and then de-duplicates edges across variants by keeping the strongest variant for each logical

edge. This view favors edges that are supported by a larger number of observed company–use-case rows. It therefore provides a more conservative view of the strongest configuration-enabled relationships.

Table 4.15: Strongest de-duplicated edges after requiring at least ten observed rows.

Edge	Best variant	Strength	n
Data readiness tier: High x Use-case family: Planning / Inventory	SHAP/perm. adj.	7.69	11
Data readiness tier: High x AI solution pattern: Forecasting / Inventory Planning	SHAP/perm. adj.	7.64	10
Execution-readiness tier: High x Delivery source: Internal	Observed only	7.57	14
Data readiness tier: High x Business department owner: Supply Chain / Logistics / Planning	SHAP/perm. adj.	7.56	12
Change capacity: High x Delivery source: Internal	Observed only	7.54	16
Supply-chain complexity: High x Delivery source: Internal	Observed only	7.51	15
Data readiness tier: High x Delivery source: Internal	Observed only	7.49	19
Channel profile: E-commerce-led x Implementation-complexity tier: Medium	Observed only	7.49	12
IT competence: High x Business department owner: Supply Chain / Logistics / Planning	SHAP/perm. adj.	7.46	10
Channel profile: E-commerce-led x Delivery source: Internal	SHAP/perm. adj.	7.44	10

The stricter support-filtered table shifts the emphasis somewhat. Instead of only showing the e-commerce-led planning/inventory edge, it highlights a broader set of readiness and execution-related patterns. High data readiness appears with planning/inventory use cases, forecasting/inventory-planning patterns, supply-chain/logistics ownership, and internal delivery. High change capacity, high supply-chain complexity, and high execution readiness also appear together with internal delivery. This supports the interpretation that the configuration-enabled framework is repeatedly finding stronger edges where company readiness or operational complexity is paired with use cases that are more internally embedded or planning-oriented.

At the same time, many of the strongest values in Table 4.15 come from the SHAP/permutation-adjusted variant. This is expected, because that variant explicitly adds positive deltas for several attributes that are common in more data-intensive or implementation-heavy use cases. Therefore, the table should not be read as pure observed evidence that these edges are stronger. It is more accurate to say that these edges remain strong after applying a configuration that emphasises the interpretability-derived adjustment logic. The observed-only rows in the table are especially important because they show that some internal-delivery patterns are also strong without prototype completion.

The final edge-level view is stability across variants. Table 4.16 summarises the selected edges used in the cross-variant stability heatmap later in the section. The table includes both stronger and relatively weaker supported edges. This matters because the cluster-strength framework should not only identify high values; it should also show whether lower or more moderate edges remain moderate across configuration settings.

Table 4.16: Selected supported edges used for cross-variant stability inspection.

Type	**Edge**	**Mean**	**Min–max**	n
Stronger	High data readiness x Internal delivery	7.24	7.08–7.49	19
Stronger	High IT competence x Internal delivery	7.08	6.97–7.27	22
Stronger	High data readiness x Forecasting / Inventory Planning	7.26	7.02–7.64	10
Stronger	E-commerce-led x Forecasting / Inventory Planning	7.44	7.09–7.79	4
Stronger	E-commerce-led x Planning / Inventory	7.50	7.28–7.84	5
Stronger	High data readiness x Planning / Inventory	7.31	7.14–7.69	11
Stronger	High change capacity x Internal delivery	7.24	7.06–7.54	16
Stronger	E-commerce-led x Internal delivery	7.27	7.10–7.44	10
Rel. weaker	Omni-led x Logistics / Operations	5.90	5.76–6.14	3
Rel. weaker	Off-the-shelf sourcing x Internal Knowledge Assistant	6.03	5.87–6.41	3
Rel. weaker	Medium change capacity x Internal Knowledge Assistant	6.06	5.86–6.33	3
Rel. weaker	Medium execution readiness x Internal Knowledge Assistant	6.12	5.98–6.29	4
Rel. weaker	Medium data readiness x Logistics / Operations	6.15	6.06–6.33	4
Rel. weaker	Off-the-shelf sourcing x Logistics / Operations	6.16	6.06–6.41	6
Rel. weaker	Medium execution readiness x Logistics / Operations	6.21	5.98–6.41	7
Rel. weaker	Omni-led x Generative AI	6.22	6.07–6.37	6

The stronger stable edges in Table 4.16 are mostly concentrated around high data readiness, high IT competence, high change capacity, e-commerce-led companies, internal delivery, and planning/inventory-oriented use cases. The relatively weaker rows are not low in absolute terms, since most still sit around 6 on the impact scale.

Rather, they are weaker compared with the stronger rows in the same configuration-enabled experiment. This distinction matters because the cluster-strength scale is still an impact scale, not a probability of success or a relative utility index.

4.6.1.4 Selected matrix heatmaps

Tables are useful for ranking edges, but they do not show the surrounding matrix structure. Therefore, eight selected matrix heatmaps were produced to show how the edge-strength values are distributed across important company-grouping × use-case-grouping combinations. These heatmaps were selected after inspecting the output, with attention to observed support, edge variation, interpretability, and whether the matrix helped explain one of the more important technical patterns.

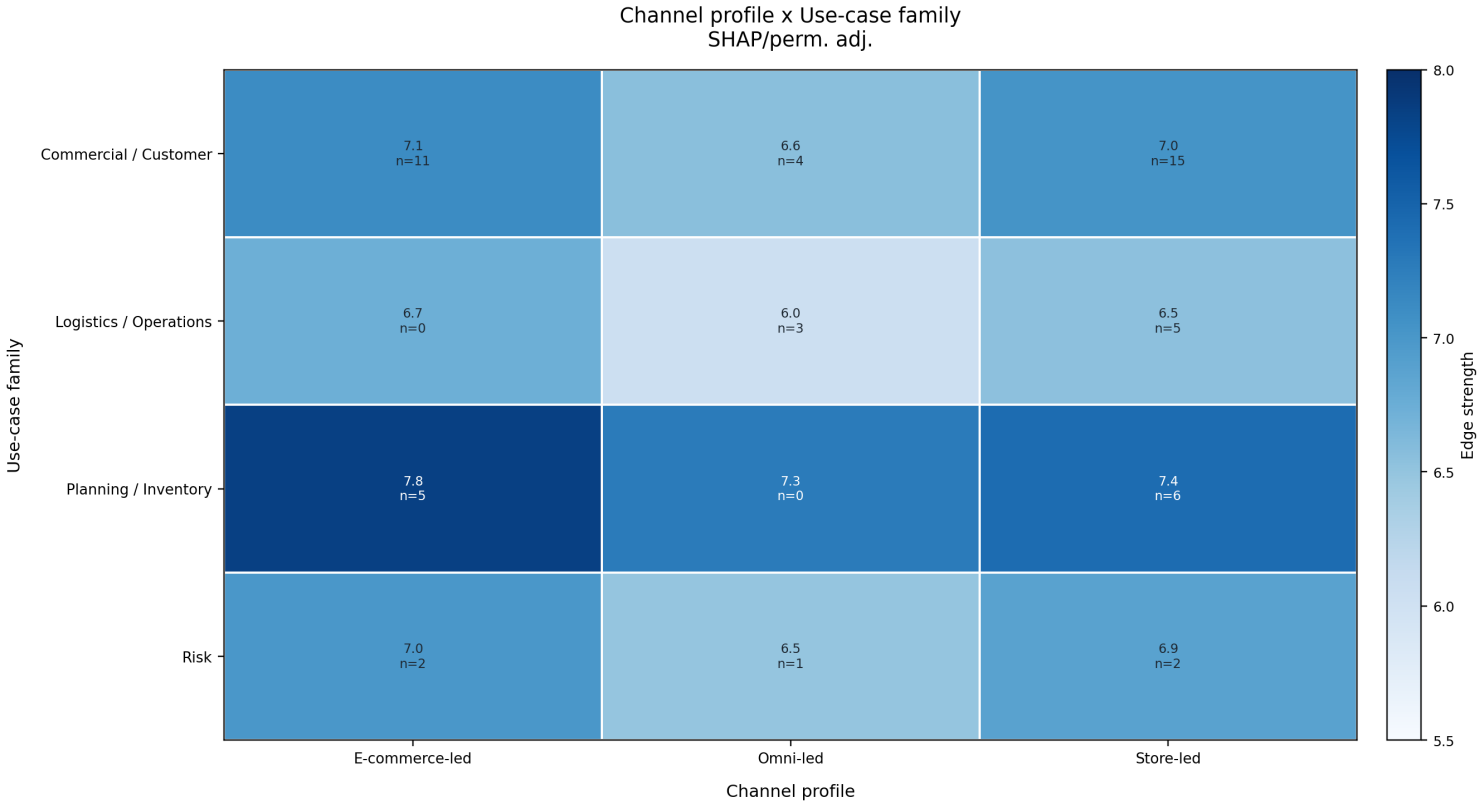


Figure 4.34: Configuration-enabled cluster-strength heatmap for channel profile and use-case family. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison.

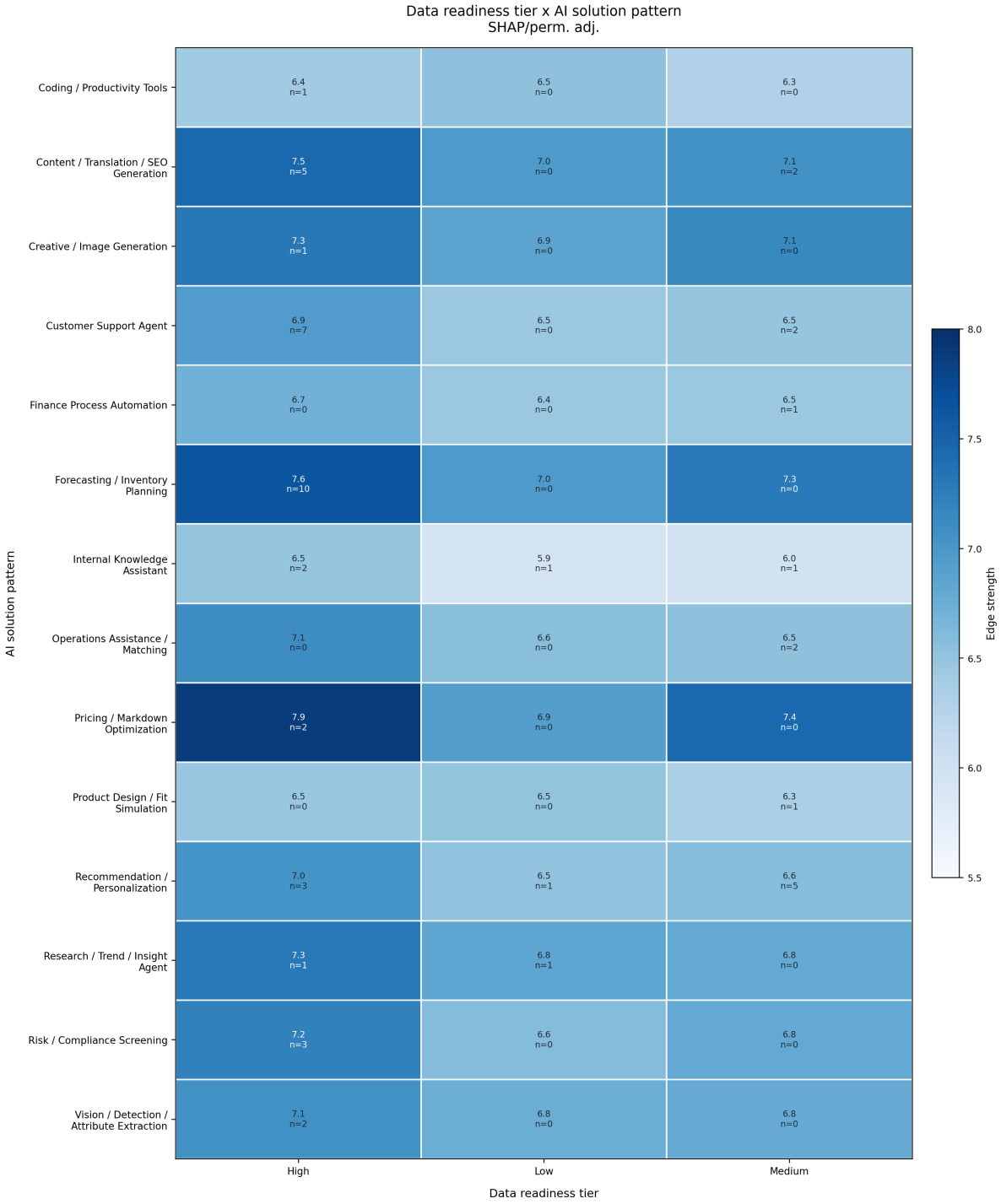


Figure 4.35: Configuration-enabled cluster-strength heatmap for data readiness tier and AI solution pattern. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison.

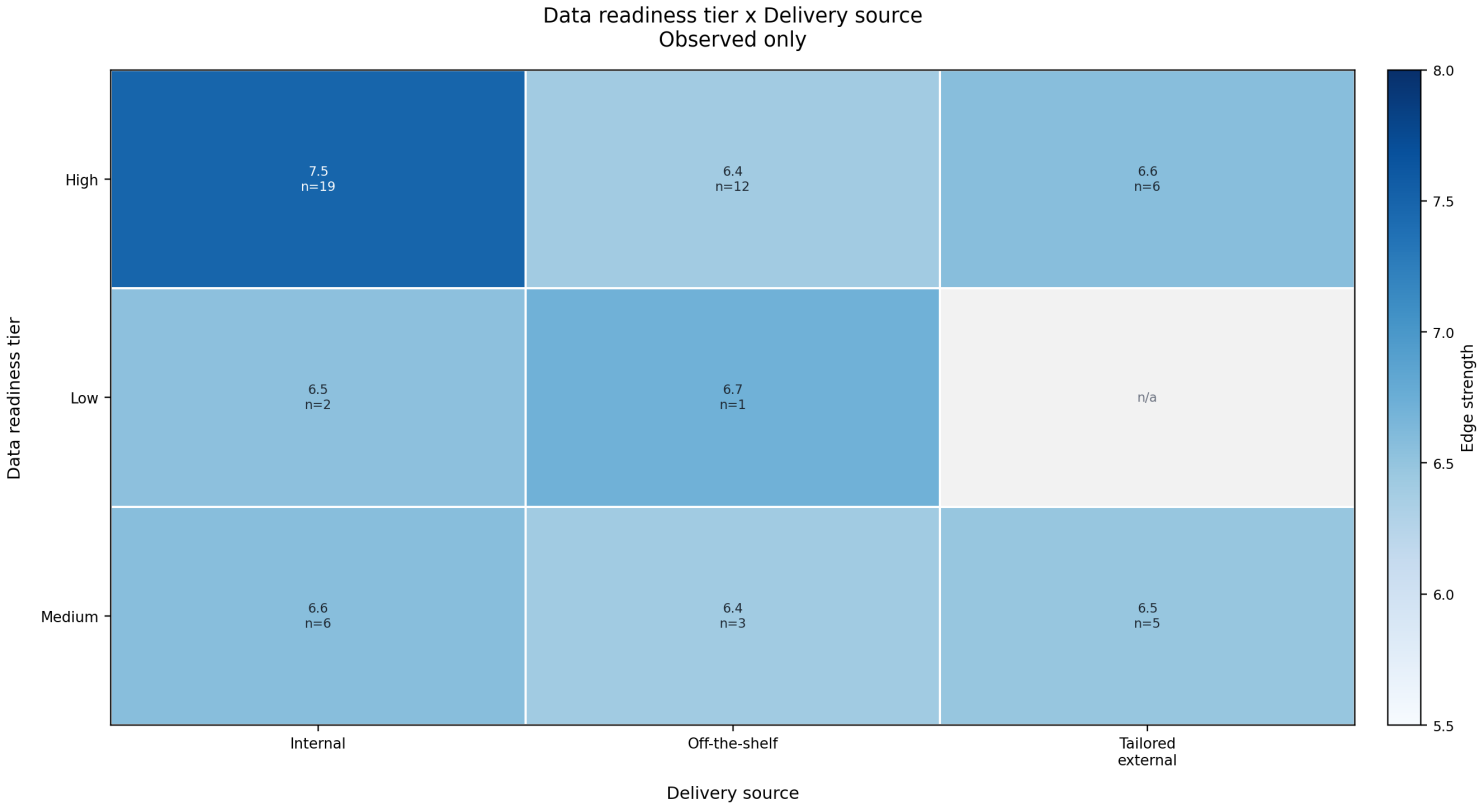


Figure 4.36: Configuration-enabled cluster-strength heatmap for data readiness tier and delivery source. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison.

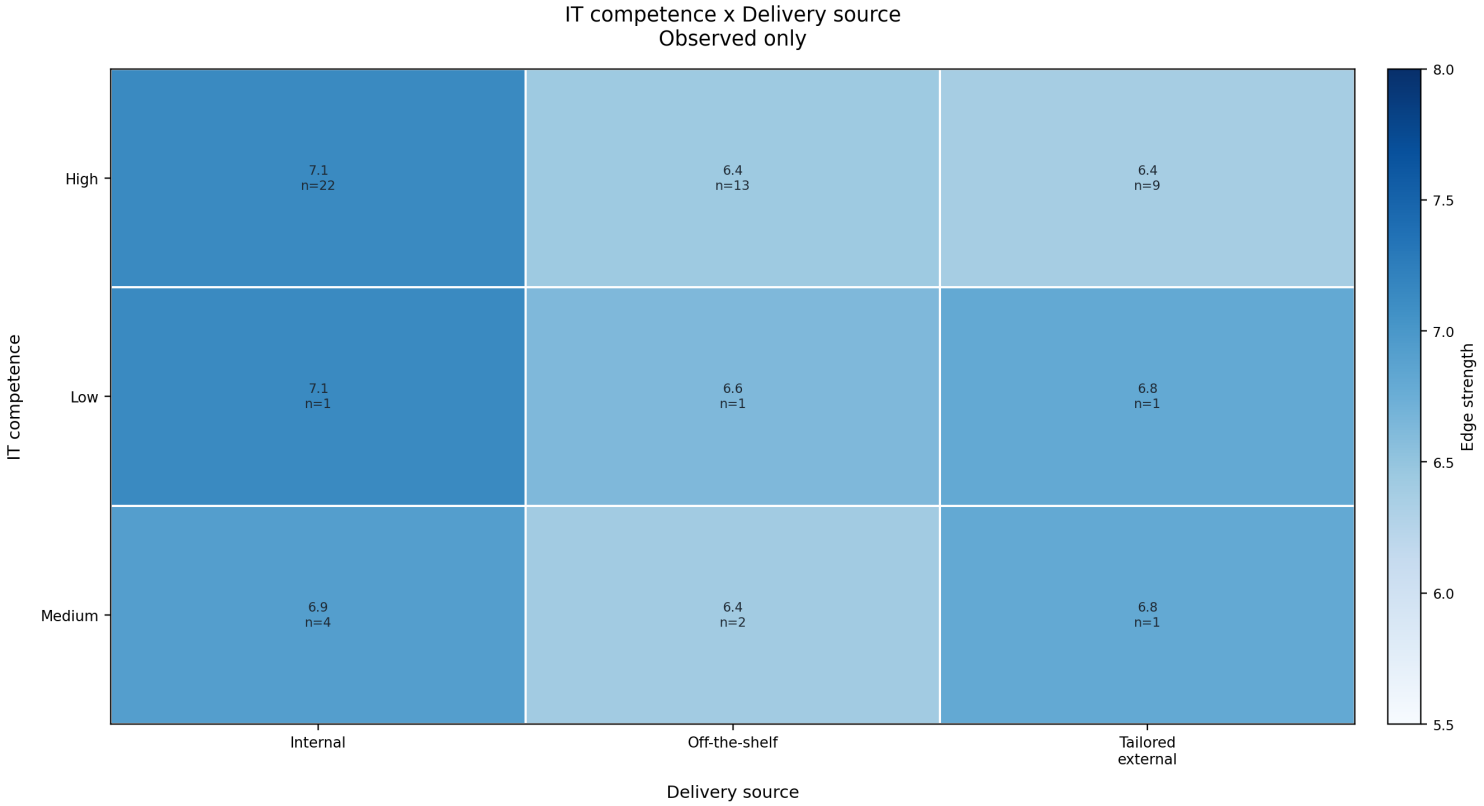


Figure 4.37: Configuration-enabled cluster-strength heatmap for IT competence and delivery source. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison.

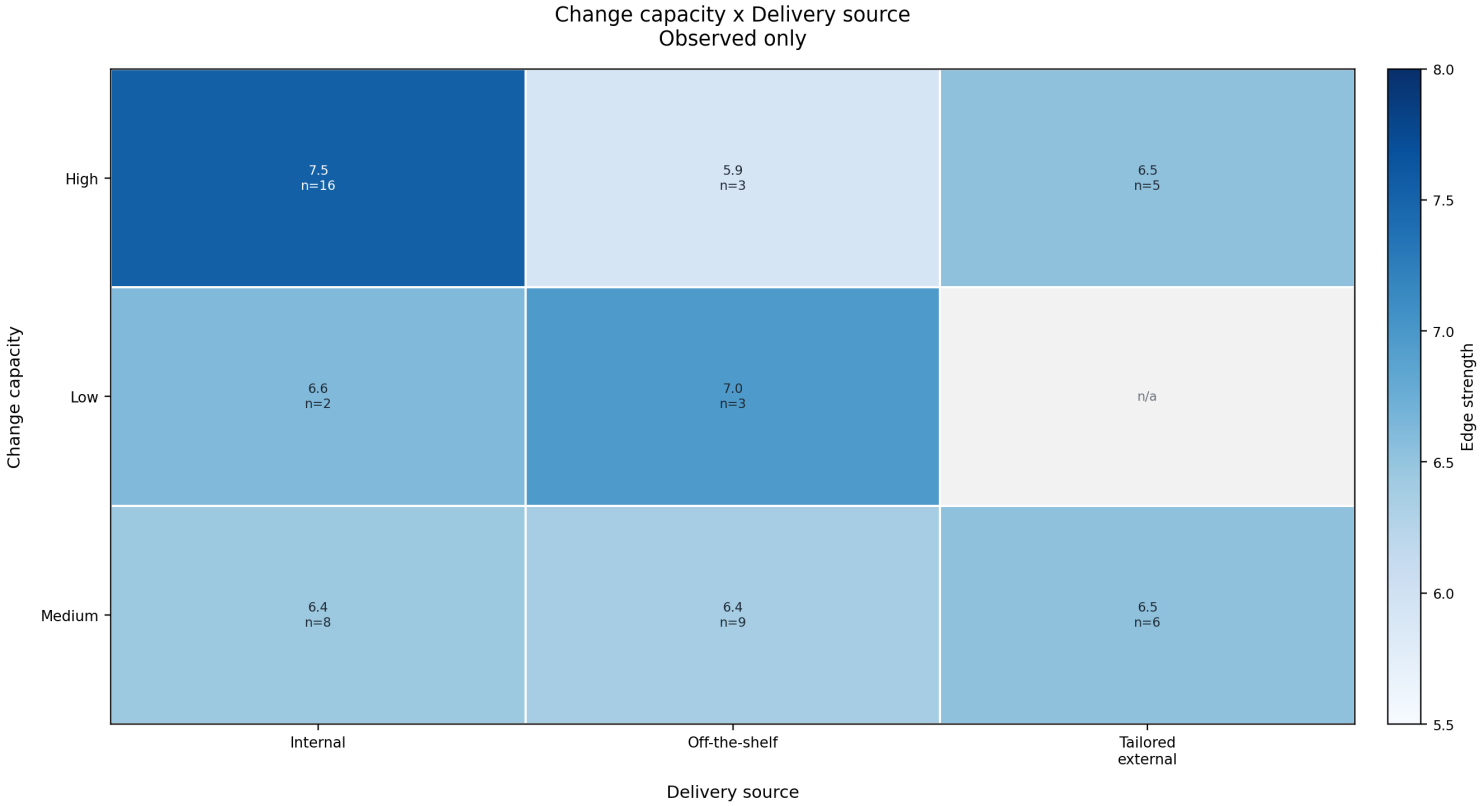


Figure 4.38: Configuration-enabled cluster-strength heatmap for change capacity and delivery source. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison.

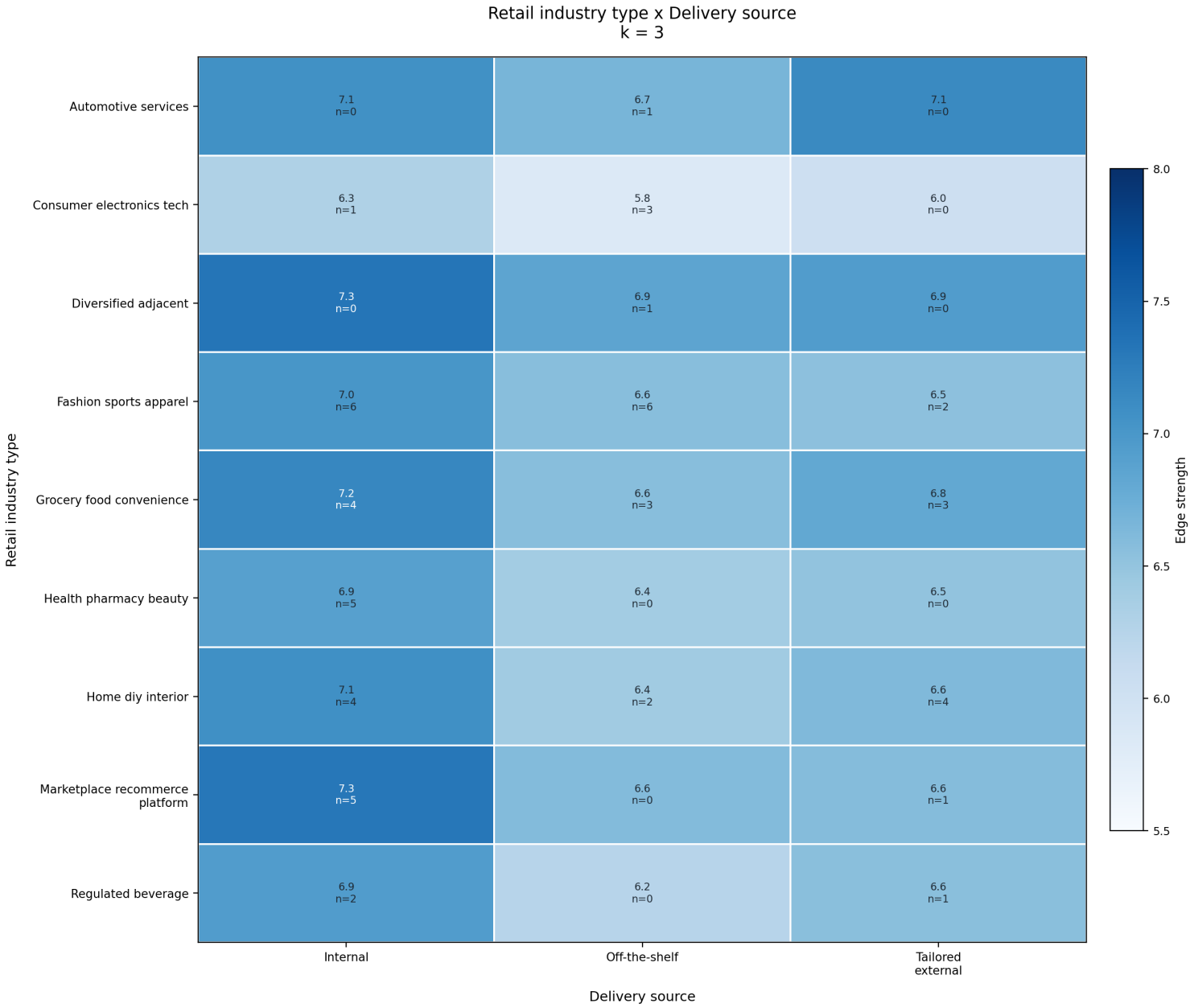


Figure 4.39: Configuration-enabled cluster-strength heatmap for retail industry type and delivery source. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison.

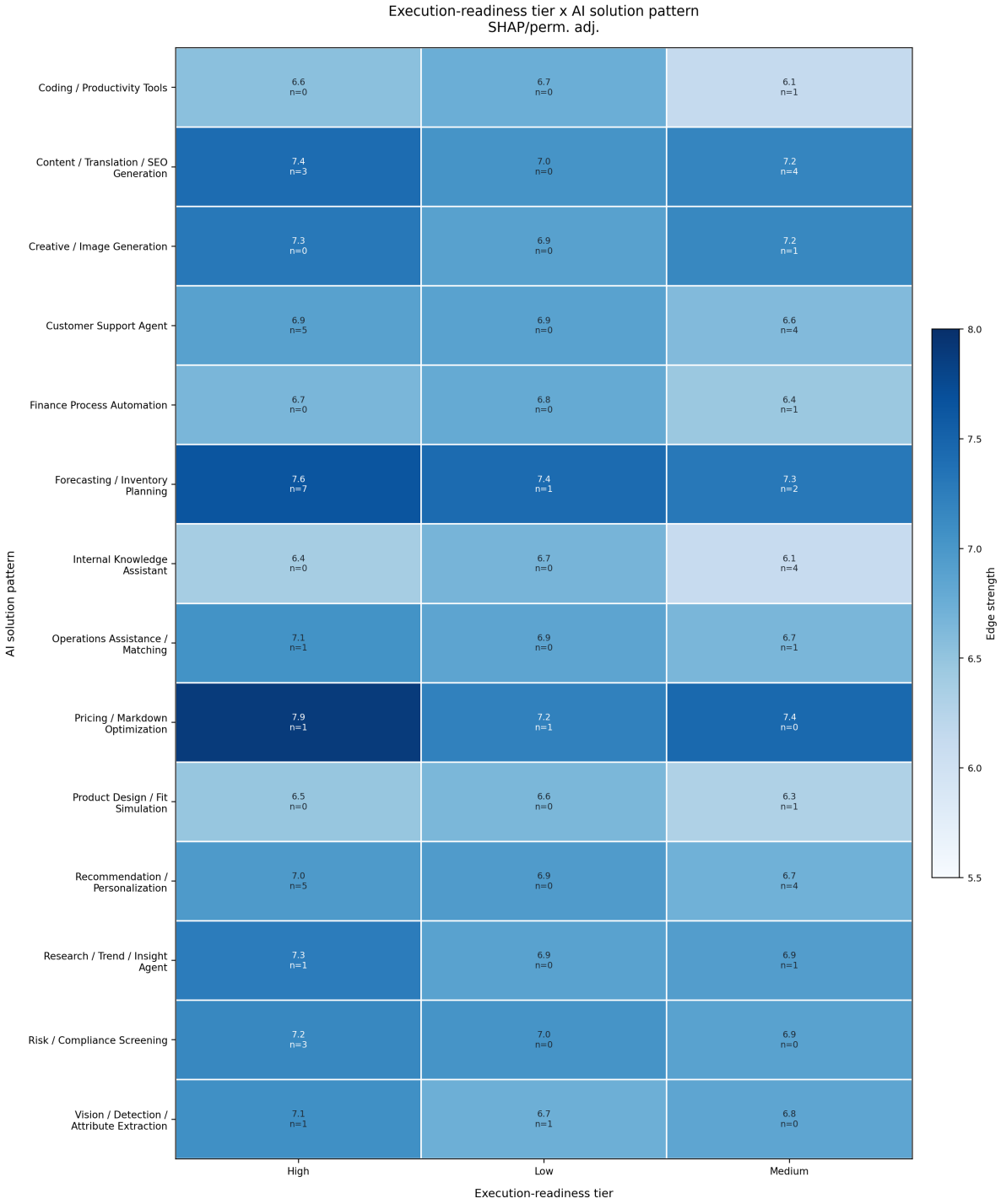


Figure 4.40: Configuration-enabled cluster-strength heatmap for execution-readiness tier and AI solution pattern. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison.

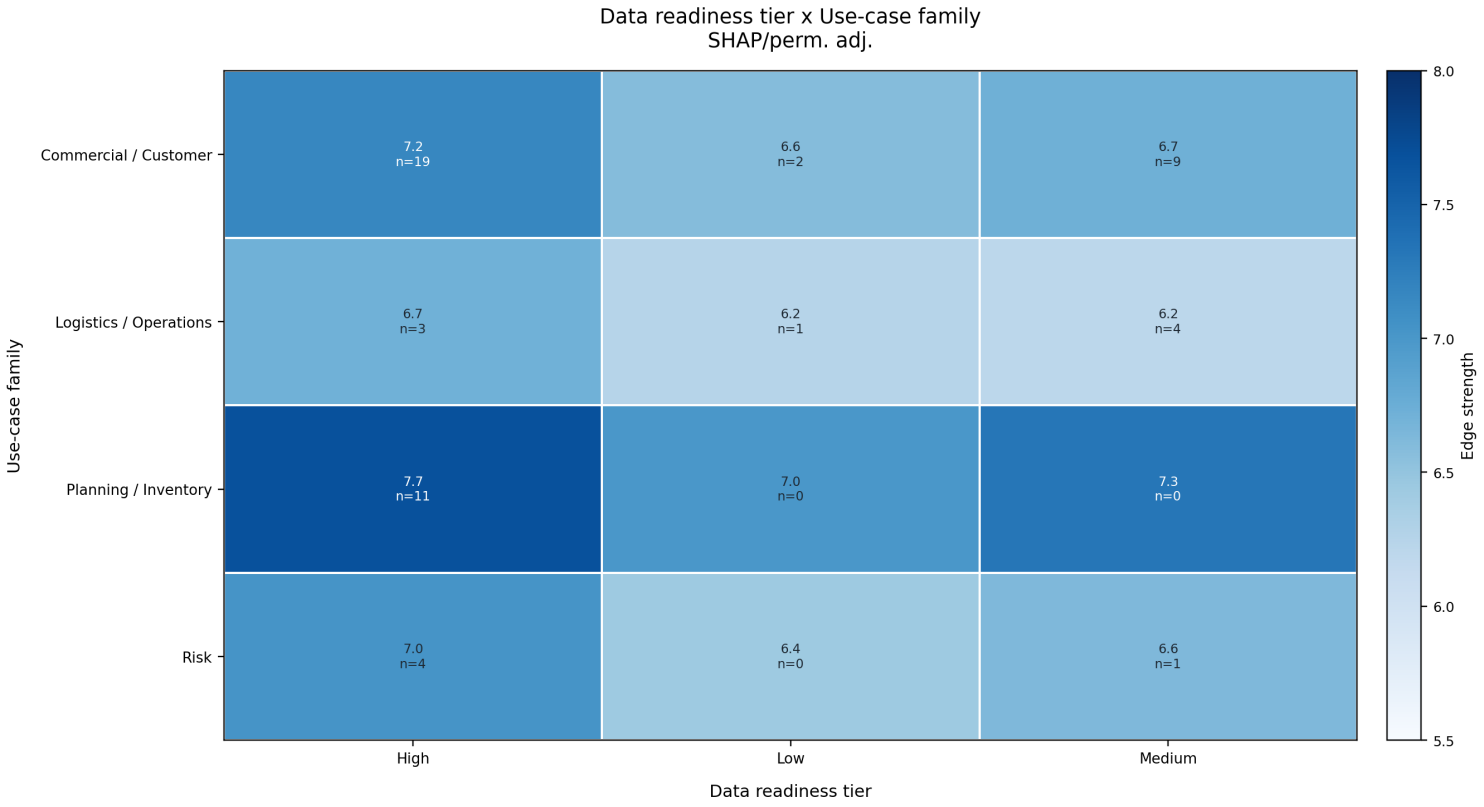


Figure 4.41: Configuration-enabled cluster-strength heatmap for data readiness tier and use-case family. Cell labels show edge strength and observed row count n. Grey n/a cells indicate combinations where no edge value was available after missing or unknown categories were removed. The colour scale is fixed at 5.5–8.0 for comparison.

The heatmaps make several patterns easier to see than the edge tables alone. Figure 4.34 shows that the e-commerce-led channel profile has a strong edge with planning/inventory use cases. This is consistent with the top-edge results in Table 4.14. However, the figure also shows that the pattern is not simply that e-commerce-led companies are high everywhere. The logistics/operations and risk cells are lower, and the observed support varies across cells. This makes the matrix more informative than a single top-edge row.

Figures 4.35 and 4.41 show a similar readiness-oriented pattern. High data readiness tends to be associated with stronger planning/inventory and forecasting/inventory-planning edges. This is conceptually plausible, because these use cases often require more reliable and timely data. At the same time, some cells with high numerical values have low observed support, and some cells are completed by the framework rather than directly observed. Therefore, the result should be treated as a useful pattern in the configuration-enabled analysis, not as a direct estimate of the population relationship.

Figures 4.36, 4.37, and 4.38 show the delivery-source dimension from three company-readiness perspectives. In these matrices, internal delivery often appears stronger for high-readiness or high-capability company groups. This is also visible in Table 4.15, where high data readiness, high change capacity, and high execution readiness all combine strongly with internal delivery. A possible interpretation is that internal or internally adapted solutions can be especially valuable when the company has the capability to absorb and operationalise them. However, this does not mean that internal delivery is always better. It may also reflect that companies capable of internal delivery select more strategically important use cases.

Figure 4.39 adds a more sector-oriented view. Internal delivery is comparatively strong for several retail industry types, although some of the cells have limited observed support. This figure is therefore mainly useful as an illustration of how the same delivery-source logic differs across industry contexts. Finally, Figure 4.40 shows how execution readiness interacts with AI solution patterns. The stronger cells again appear around planning/inventory and some content or optimisation-oriented patterns, while internal knowledge assistant edges tend to sit lower in several related views.

4.6.1.5 Cross-variant stability heatmap

The selected matrix heatmaps show individual grouping-pair runs. The next question is whether selected edges remain similar across configuration variants. Figure 4.42 therefore plots the selected stronger and relatively weaker supported edges from Table 4.16 across all nine variants.

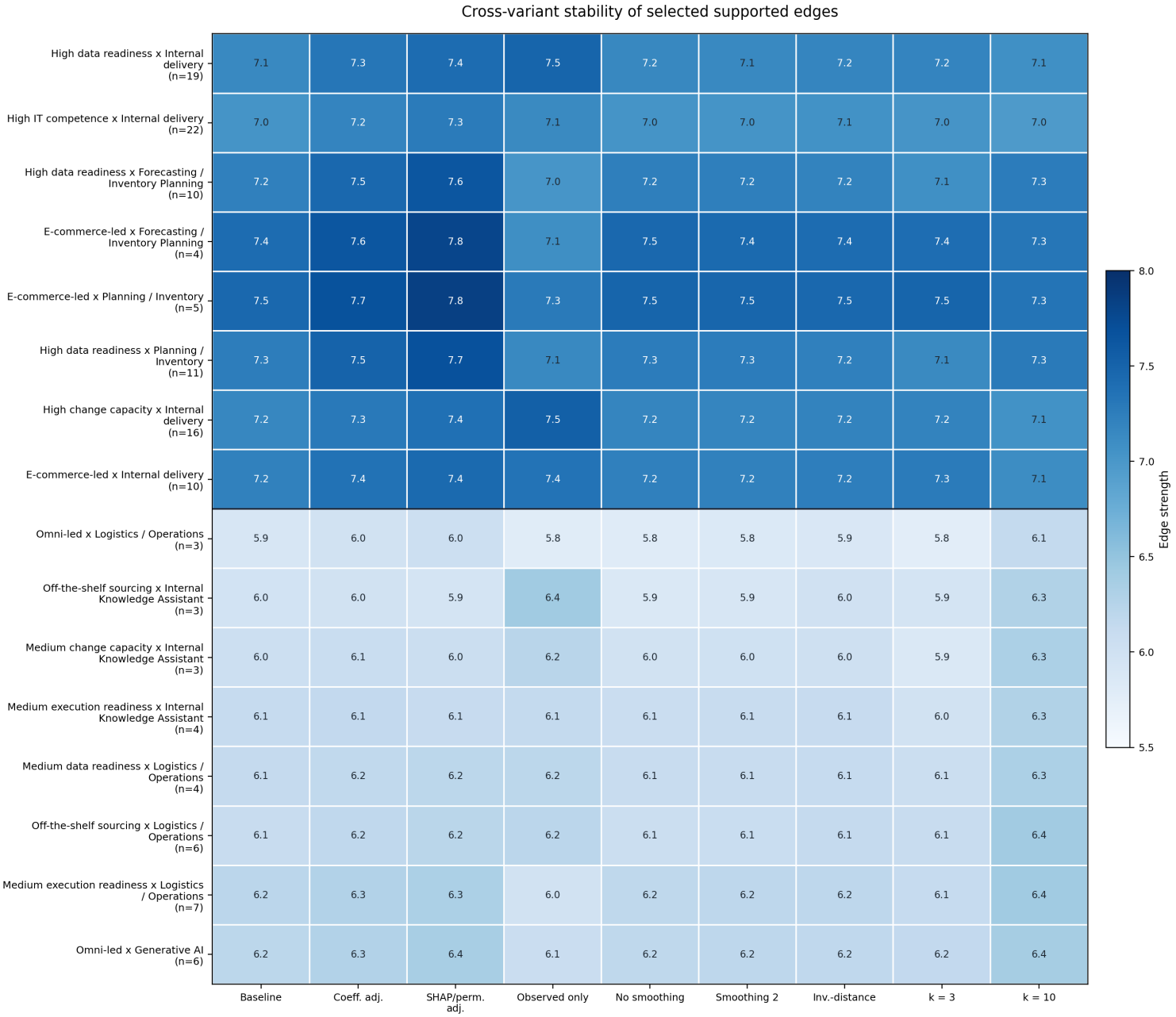


Figure 4.42: Cross-variant stability heatmap for selected supported cluster-strength edges. Cells show edge strength by configuration variant. Row labels include observed row count n. The colour scale is fixed at 5.5–8.0 for comparison.

The stability heatmap shows that several of the stronger edges remain high across many configuration choices. For example, high data readiness combined with internal delivery remains around 7.1–7.5, high IT competence combined with internal delivery remains around 7.0–7.3, and e-commerce-led companies combined with planning/inventory or forecasting/inventory-planning use cases remain among the higher rows. This is important because it suggests that these patterns are not only artefacts of one single configuration setting.

At the same time, the heatmap also shows configuration sensitivity. The manual-adjustment variants often lift the stronger rows, especially when the edge involves data readiness or implementation-intensive use cases. The observed-only variant can move some rows downward or upward relative to the prototype-completed variants. The $k = 10$ variant tends to compress the values somewhat, which is consistent with the variant-level sensitivity table. Therefore, the stability result should be interpreted as partial support for recurring patterns, not as proof that the numerical edge strengths are fixed.

The weaker rows are also informative. Rows such as omni-led channel profile combined with logistics/operations use-case family, off-the-shelf sourcing preference combined with internal knowledge assistant, and medium readiness or medium execution-readiness groups combined with internal knowledge assistant stay relatively lower across the variants. Again, these are not necessarily bad use-case categories. They are weaker in this configuration-enabled edge-strength comparison. This distinction is important, because the later roadmapping interpretation should not say that these use cases lack value. It should instead ask under what conditions they become more or less attractive.

4.6.1.6 Technical takeaway

Overall, the configuration-enabled cluster-strength analysis produced three main technical results. Firstly, the framework is sensitive to configuration choices in the way expected from the methodology. Manual adjustments lift and widen the edge distribution, no smoothing increases spread, and a larger neighbour count compresses edge strengths. This is a useful validation of the implementation, because the output changes in a direction that is consistent with the purpose of each configuration setting.

Secondly, observed support is a central boundary condition. The zero-observed share is about 0.48 across variants, meaning that almost half of the edge cells in the broad configuration-enabled batch have no directly observed company–use-case rows behind them. This does not make the analysis invalid, because sparse completion was one reason for building the framework. However, it means that the configuration-enabled results should be read as sensitivity-oriented pattern discovery. They are less direct than the observed-only results that will be discussed separately.

Thirdly, some patterns appear repeatedly despite this sensitivity. The most visible repeated patterns concern high data readiness, high execution readiness, high change

capacity, internal delivery, and planning/inventory-oriented use cases. These results fit the broader logic of the project: AI use-case prioritisation seems to depend not only on the use-case category, but also on whether the company has the data, technical competence, and organisational capacity to turn that use case into operational value. The next step is therefore to compare these configuration-enabled patterns with the cleaner observed-only cluster-strength results, where the interpretation can be closer to direct empirical evidence.

4.6.2 Observed-only analysis

The observed-only analysis deliberately removes the configuration-enabled completion layer. In this stricter version, an edge is simply the average observed impact among existing, non-synthetic rows between one company group and one use-case group. Observed coverage is the share of the full possible pair universe represented by observed rows in that edge. This makes the results easier to interpret as direct study evidence, although sparse support still limits how strongly individual edges should be read.

Table 4.17: Observed-only delivery-source and internal-adaptation edges. Coverage is the observed share of the relevant company-use-case pair universe.

Edge	**Strength**	n	**Cov.**
Grocery / food / convenience x Internal	8.500	4	0.025
Marketplace / recommerce / platform x Internal	7.800	5	0.093
Home / DIY / interior x Internal	7.750	4	0.021
Grocery / food / convenience x External	7.667	3	0.045
Fashion / sports / apparel x Internal	7.417	6	0.028
Enterprise x Internal	7.611	9	0.028
Large x Internal	7.167	12	0.028
Mid x Internal	7.000	6	0.044
High change capacity x Internal	7.781	16	0.042
High data readiness x Internal	7.684	19	0.037

The delivery-source edges suggest that internal delivery was repeatedly associated with higher observed impact, especially among companies with stronger data readiness or change capacity. This should not be interpreted as a universal sourcing rule. It is better read as evidence that internally shaped AI can be valuable when the company has the capability, ownership, and data conditions needed to adapt the use case closely to its own workflows.

Table 4.18: Observed-only AI solution pattern and use-case archetype edges. Coverage is the observed share of the relevant company-use-case pair universe.

Edge	Strength	n	Cov.
Fashion / sports / apparel x Content / Translation / SEO Generation	7.300	5	0.089
Grocery / food / convenience x Forecasting / Inventory Planning	6.500	4	0.067
Health / pharmacy / beauty x Recommendation / Personalization	6.333	3	0.111
Large x Forecasting / Inventory Planning	7.333	3	0.019
Large x Customer Support Agent	7.300	5	0.039
Large x Content / Translation / SEO Generation	7.200	5	0.045
Store-led x Content / Translation / SEO Generation	7.500	5	0.042
E-commerce-led x Forecasting / Inventory Planning	7.500	4	0.044
E-commerce-led x Optimisation	8.000	4	0.063
Large x Optimisation	8.500	4	0.036

Several use-case archetype patterns also appear. Forecasting and inventory planning appears repeatedly in high-value cells, especially for grocery-like, e-commerce-led, large, and data-ready contexts. Content, translation, and SEO generation is also strong in fashion and store-led contexts. These patterns support the idea that retail AI roadmaps should start from retail context rather than from a universal ranking of AI categories.

Table 4.19: Observed-only readiness and implementation-complexity edges. Coverage is the observed share of the relevant company-use-case pair universe.

Edge	Strength	n	Cov.
High data readiness x Content / Translation / SEO Generation	7.600	5	0.038
High data readiness x Risk / Compliance Screening	7.333	3	0.053
High data readiness x Forecasting / Inventory Planning	7.150	10	0.053
High data readiness x Customer Support Agent	6.857	7	0.041
High data readiness x Medium implementation complexity	7.286	21	0.038
High data readiness x High implementation complexity	6.750	4	0.053
High data readiness x Low implementation complexity	6.625	12	0.030
Medium data readiness x Medium implementation complexity	6.312	8	0.028

The readiness and implementation-complexity edges point in a similar direction. High data readiness combines strongly with several use-case archetypes, and high data readiness with medium implementation complexity is one of the clearest supported observed-only cells. This suggests that value is often found between trivial and very difficult use cases: substantial enough to matter, but still feasible enough to operationalise.

4.6.2.1 Observed-only takeaway

Taken together, the observed-only cluster-strength analysis gives a more business-facing version of the cluster-strength results. Internal delivery appears repeatedly

strong, especially when combined with high data readiness, high change capacity, larger company size, or several retail industry types. Forecasting / inventory planning, optimisation-oriented use cases, content / translation / SEO generation, and customer-support-agent patterns appear as recurring high-value archetypes, although their strength depends on company and channel context.

These results should not be interpreted as causal rules. The study uses a limited number of companies and observed use cases, and several edges have limited support. Still, the observed-only analysis gives a transparent quantitative basis for later discussion. It supports the idea that retail AI roadmapping should not only ask which AI use cases are attractive in general. It should also ask which company contexts, delivery models, and implementation conditions make a specific AI use case more likely to create value.

5

Discussion

The Results chapter presented several different kinds of evidence: descriptive portfolio patterns, qualitative interview results, baseline impact modelling, model interpretability analysis, and cluster-strength analysis. These result types do not have the same role in the thesis. Some of them are close to direct empirical description, while others are more technical and should be understood as exploratory modelling evidence. Because of this, the Discussion chapter is divided into two main interpretive parts before returning to the research questions, limitations, and future work.

The first part discusses the analytical framework from a more technical and methodological perspective. It focuses mainly on the baseline impact model, the interpretability analysis, and the cluster-strength framework. The purpose is to discuss what these technical results can and cannot support. The second part, which follows later, shifts the interpretation towards retail AI roadmapping and playbook-oriented implications. That part uses the technical results as background, but draws more directly on the qualitative results, the descriptive portfolio patterns, and the observed-only cluster-strength results.

5.1 Technical interpretation

5.1.1 Manager-rated expected impact

A first point concerns the meaning of the target variable itself. In the modelling sections, the target was the coded impact value on the original 0–10 scale. However, this value should not be read as an objectively measured financial or operational outcome. It is better understood as a manager-rated expected or perceived impact assessment collected through the interview process. In the Results chapter, the phrase observed impact means that the value was observed in the study data. It does not mean that the impact was independently verified after implementation.

This distinction changes how the modelling results should be interpreted. If the target is treated as realised business impact, then the model would appear to make stronger claims than the data can support. The study did not track use cases longitudinally, and it did not measure before-and-after effects in revenue, cost, service level, forecast accuracy, or working capital. Instead, the study captured how senior retail respondents assessed the expected or perceived impact of concrete AI use

cases. The target variable is therefore subjective, interview-derived, and shaped by managerial judgement.

At the same time, this does not make the target uninteresting. For the purpose of AI roadmapping, managerial expectations are themselves valuable. They say something about which AI initiatives senior decision-makers perceive as important, credible, and worth pursuing. A model that predicts these assessments is therefore not only predicting technical performance. More precisely, it is modelling a structured version of managerial prioritisation. This is a narrower claim than predicting realised impact, but it is still relevant for a thesis about AI use-case prioritisation in retail.

5.1.2 Common-rater and subjectivity risks

The subjective target also creates a second methodological issue. In many cases, the same respondent helped describe the use case attributes and assess the expected impact. This creates a possible common-rater effect. If a respondent is enthusiastic about a use case, they may rate its expected impact highly and also describe its attributes in a more favourable way. Similarly, if a respondent is sceptical or uncertain, both the descriptive attributes and the impact score may become more cautious.

This matters especially for the baseline impact model and the interpretability analysis. A model may find a relationship between, for example, data quality and impact partly because data quality really matters for AI use-case value. But it may also find that relationship because respondents who believe strongly in a use case tend to describe the surrounding conditions more positively. The model cannot fully separate these two explanations. With the current data, the correlations should therefore be read as patterns in the interview-coded material, not as independently validated causal mechanisms.

There is also a more subtle version of the same issue. Some respondents may have interpreted attributes through the lens of the specific use case being discussed. For example, if a use case was seen as strategically important, the respondent may have been more willing to describe its data access, ownership, or implementation path as manageable. Another respondent, describing a similar use case in a less mature organisation, may have scored both feasibility and expected impact lower. This means that part of the signal in the dataset may reflect managerial confidence and organisational ambition, not only objective attribute differences.

This does not invalidate the analysis, but it affects the level of claim that should be made. The technical results are strongest when they are used to identify plausible prioritisation signals, compare representations, and structure later interpretation. They are weaker if read as proof that a certain attribute mechanically increases or decreases impact. For that reason, the later roadmapping discussion should combine the model-based results with the qualitative material and with the cleaner observed-only descriptive patterns.

5.1.3 Predictive signal in a small dataset

The baseline impact modelling results indicate that the structured attributes contain some predictive signal, but the signal is limited. The strongest consolidated non-fit model, histogram gradient boosting with the `opf` feature set, reached MAE 1.348 and RMSE 1.736. The ordinary gradient boosting model with the same feature set had almost the same MAE at 1.352, slightly lower RMSE at 1.714, and the strongest global Spearman value among the top-ranked models at 0.439. These are useful results in the context of a small interview-derived dataset, but they are not strong enough to treat the model as a reliable stand-alone predictor.

The modest out-of-fold R^2 values are especially important here. Even the strongest values were low, with the top rows around 0.11–0.13. This suggests that the model captured some structure, but that much of the variation in manager-rated impact remained unexplained. This is not surprising. A 55-row impact-rated dataset is small for supervised learning, and the observations cover different companies, use-case types, departments, delivery models, and maturity levels. In addition, some of the most important determinants of impact are probably difficult to encode in a structured interview sheet. These include local ownership, implementation depth, political support, timing, budget, hidden system constraints, and the quality of the actual solution.

The diagnostic plots support the same interpretation. Across the selected winner diagnostics, many models compressed predictions towards the middle of the impact scale. This is a common pattern when a model has limited data and uncertain signal. Extreme values are harder to predict, so the model tends to avoid very high or very low predictions. In practical terms, that means the model can help identify broad tendencies, but it is less able to distinguish precisely between, for example, a moderately strong use case and an exceptional one.

For the thesis, this is still a useful outcome. The baseline model does not need to be perfect to contribute to the framework. It shows that structured interview attributes can contain meaningful prioritisation information, especially when the feature set is focused. However, the modelling result also shows why the framework should remain exploratory.

5.1.4 Prioritisation over exact prediction

Because the model is imperfect as an exact predictor, the most realistic use of the baseline impact model is prioritisation support. In a retail AI roadmap, the practical question is often not whether a use case will score exactly 7.2 rather than 6.8. The more useful question is whether one group of use cases appears more promising than another, or whether a model can help identify candidates that deserve closer qualitative and managerial inspection.

This is why the ranking-oriented metrics matter. MAE and RMSE measure numerical prediction error, which is important, but they do not fully capture the roadmapping

task. Spearman correlation and top-1 hit rate are closer to the use case of ranking and prioritisation. For example, the `opf` gradient boosting model had a slightly worse MAE than histogram gradient boosting, but it had stronger global Spearman and higher top-1 hit rate. If the model is used to support prioritisation, that difference matters.

There is still a caveat. A good top-1 result does not mean that the model understands the use-case portfolio in a deep way. In a small dataset, top-1 hit rate can be affected by how many use cases each company has, how concentrated the impact scores are, and whether one use case is clearly higher than the others. Spearman can also be unstable when the number of observations per company is small. Still, these metrics are valuable because they shift the evaluation away from pure point prediction and towards the practical roadmapping problem.

Overall, the model is most defensible as a structured prioritisation aid. It can help organize evidence and test whether certain feature representations contain useful signal. It should not replace expert judgement, interview interpretation, or company-specific analysis. This is also consistent with the broader thesis logic: the model is one layer in a combined framework, alongside qualitative interpretation and cluster-strength analysis.

5.1.5 Model complexity helped, but only partly

The model-family comparison shows that more flexible models helped in some cases, but the result is not a simple argument for maximum complexity. The strongest results came from tree-based ensemble models, especially gradient boosting and histogram gradient boosting with the `opf` feature set. This makes conceptual sense. The relationship between company readiness, use-case burden, delivery source, and expected impact is unlikely to be purely linear. For example, high data need may be acceptable in a company with strong data quality and IT competence, but more problematic in a company with weaker data conditions. Tree-based models can represent this type of interaction more naturally than linear regression.

However, the improvement over simpler models should be interpreted carefully. Ridge regression with `opf` reached MAE 1.439 and RMSE 1.834. This is worse than the best gradient boosting result, but not dramatically worse in absolute-error terms. The difference is larger in ranking behaviour, where gradient boosting reached global Spearman 0.439 while ridge reached 0.144. This suggests that the flexible model did add useful ordering signal, but it also means that the simpler model was not irrelevant. If the performance gap had been smaller, the simpler model would have been preferable because it is easier to interpret and less likely to rely on unstable interactions (Occam's razor).

The weaker model families reinforce the same point. The MLP regressor performed poorly, with MAE 1.997 and strongly negative out-of-fold R^2. This is a rather clear sign that neural-network-style flexibility was not useful in this setting. The model had too little data relative to its flexibility, and the interview-derived rows were too

heterogeneous for that flexibility to become helpful. In this case, a more complex model did not add intelligence; it mainly added instability.

The technical lesson is therefore a cautious one. Some nonlinearity appears useful, especially in focused feature sets, but the dataset is far too small to justify a general preference for complex models. The best modelling posture is to compare flexible and simple models side by side, then prefer the simplest model that captures the needed signal. In this study, gradient boosting was useful because it clearly performed better on several metrics, but the interpretation should still remain conservative.

5.1.6 Interpretability is diagnostic, not causal

The interpretability analysis adds useful technical understanding, but it also has clear limits. Coefficients, SHAP values, and permutation importance all describe model behaviour. They do not prove that an attribute causally determines expected impact. This distinction is especially important because the target variable is manager-rated and because the same respondent may have influenced both the input attributes and the target.

The most repeated interpretability signals were still meaningful. Data recency requirement, data quality, build effort, process change burden, time to value, and operational risk appeared repeatedly across the coefficient, SHAP, and permutation-importance views. Several of these are easy to connect to retail AI implementation. Better data quality should support stronger AI use cases. Shorter time to value is attractive from a prioritisation perspective. Higher operational risk can reduce expected impact because it raises the threshold for trust, control, and production use.

Other patterns are more difficult to read. Build effort and process change burden sometimes appeared as positive signals, especially in nonlinear models. This should not be interpreted as saying that burden is good in itself. A more plausible explanation is that these attributes also capture use-case depth. A small and simple use case may be easy to implement but have limited potential value, while a more substantial use case may require more effort and process change because it touches a more important operational process. The model may therefore be mixing implementation difficulty with opportunity size.

Some coefficient results were also counterintuitive, especially for company-side capability attributes such as IT competence and process governance maturity. These results are useful precisely because they warn against overinterpretation. A negative or unstable coefficient does not necessarily mean that IT competence is harmful. It may instead reflect correlated attributes, different ambition levels, small sample effects, or the fact that more capable companies discussed more demanding use cases. In a larger and more controlled dataset, these relationships might look different.

For that reason, the interpretability outputs should be treated as diagnostic evidence. They help identify candidate attributes that deserve attention, and they make the

model less opaque. But they should not be used as direct causal weights or as direct managerial rules. The value of the interpretability analysis is that it helps connect the predictive model to transparent reasoning, while still showing where the reasoning is fragile.

5.1.7 Manual adjustments as sensitivity settings

The candidate manual-adjustment rules in Tables 4.10 and 4.11 should be interpreted in the same cautious way. They were derived from the model-interpretability analysis, but they were not learned causal parameters. They were deliberately small, transparent sensitivity settings used in the configuration-enabled cluster-strength experiments.

This matters because the adjustment rules sit between two different parts of the framework. On the one hand, they use signals from the impact model interpretability analysis. On the other hand, they affect the prototype-completion layer in the cluster-strength module. If these adjustments were treated as definitive, the framework would risk turning uncertain model behaviour into seemingly precise edge values. The safer interpretation is that the adjustments are hypotheses: if certain attributes repeatedly appear important in the model, then it is reasonable to test whether small directional changes based on those attributes affect the cluster-strength outputs.

The results support this sensitivity interpretation. The manual-adjustment variants increased the average edge-strength level in the configuration-enabled cluster-strength analysis. This was expected, because several adjustment rules added positive deltas for attributes associated with stronger model predictions. But a higher mean edge value does not mean that the adjusted configuration is more correct. It means that the selected assumptions changed the completed values in a predictable way.

In that sense, the main value of the adjustment layer is that it makes assumptions explicit. Instead of hiding completion logic inside a single technical pipeline, the framework can show what happens when plausible adjustment rules are introduced. If an edge remains visible across no-adjustment and adjustment-enabled variants, it becomes more credible as a pattern. If it appears only under one adjusted configuration, it should be treated more carefully.

5.1.8 Cluster strength as pattern discovery

The cluster-strength framework changes the unit of analysis from individual rows to broader company-grouping × use-case-grouping relationships. This fits the purpose of the thesis, since AI roadmapping often depends on patterns such as which use-case families fit which company contexts, rather than on predicting one isolated row. Still, the two cluster-strength modes need to be interpreted differently.

The configuration-enabled analysis is mainly a technical sensitivity experiment. It includes prototype completion, smoothing, manual adjustments, neighbour weighting, and different neighbour counts. This makes it possible to evaluate how the framework

behaves under different assumptions. For example, the Results chapter showed that no smoothing increased spread, larger neighbourhoods compressed edge strengths, and manual adjustments lifted the average edge level. These patterns are technically informative because they show that the framework reacts in plausible directions when its configuration settings are changed.

At the same time, prototype completion introduces uncertainty. When the dataset is sparse, the nearest neighbours used for completion may still be substantively different from the missing company-use-case pair. In other words, the nearest available case can still be far away in the underlying business sense. A completed value may therefore look precise while actually depending on a weak similarity assumption. This is why the configuration-enabled results should not be read as direct empirical facts. They are better read as stress tests and pattern-discovery outputs.

The observed-only cluster-strength analysis is easier to defend as direct evidence, because it uses only directly observed company-use-case rows and computes simple average impact inside each edge. However, this version also has limitations. Some cells have small observed row counts, and coverage differs across edges. A high edge with $n = 3$ or $n = 4$ may be interesting, but it should not be treated like a population-level estimate. The observed-only analysis is therefore more empirical than the configuration-enabled analysis, but it is still sparse.

This distinction is central for the later roadmapping implications. The configuration-enabled results show how the analytical framework behaves when assumptions are changed. The observed-only results are closer to the empirical material and are therefore the stronger basis for business-facing interpretation. In practice, this means that the cluster-strength framework is most useful when the support level, modelling assumptions, and qualitative explanation are kept visible together.

5.2 Retail AI roadmapping

The second interpretive part turns from the technical framework to the practical implications for retail AI roadmapping. The purpose is different from the previous section. Here, the main question is not whether the model predicts impact with high precision, but what the combined results suggest about how retailers should think when they prioritise, sequence, and scale AI use cases. For that reason, this part relies most directly on the descriptive portfolio results, the observed-only cluster-strength results, and the qualitative findings. The modelling results still matter, but mainly as background evidence about which attributes appeared meaningful and where the framework should be interpreted carefully.

5.2.1 Roadmaps should start from readiness and retail context

The results suggest that retail AI roadmaps should start from the retailer's context, rather than from a generic list of attractive AI use cases. Company readiness differed

across data availability, data quality, data accessibility, IT competence, and change capacity. The qualitative material also showed that readiness is often specific to a workflow. A company may be ready for AI in one domain because the data, ownership, and process logic are already in place, while another domain in the same company may still be difficult to scale because the process is fragmented or poorly documented.

The observed-only cluster-strength results support the same point. Different retail industry types showed different promising AI solution patterns. For example, fashion / sports / apparel had a strong observed edge with content / translation / SEO generation, grocery / food / convenience had a strong observed edge with forecasting / inventory planning, and health / pharmacy / beauty had a strong observed edge with recommendation / personalization. These edges should not be interpreted as universal rules, because the support levels are still limited. However, they are useful signals because they make the broader roadmapping point concrete: the first question should be what kind of retailer and workflow the use case is supposed to fit.

This also means that readiness should not be treated as a simple yes-or-no condition. A retailer with weaker data accessibility may still be able to run bounded GenAI use cases that use narrow document sets or low-risk workflows. Conversely, a retailer with strong data availability may still struggle with AI if ownership, decision rights, or change capacity are weak. A roadmap should therefore separate general company readiness from use-case-specific readiness. The relevant question is not only whether the company is mature in general, but whether the specific workflow has the data, ownership, and implementation conditions needed for the AI use case to become usable.

5.2.2 AI portfolios should separate quick wins, strategic bets, and learning cases

The portfolio results also suggest that AI use cases should not be evaluated along one dimension only. Use cases differ in expected impact, time to value, process-change burden, integration burden, maintenance cost, operational risk, and ROI clarity. A content-generation use case with fast time to value is not the same type of roadmap object as a forecasting model that affects inventory planning. Both may be valuable, but they require different expectations, different governance, and different implementation effort.

A clearer roadmap distinction is therefore to separate quick wins, strategic bets, and learning cases. Quick wins are use cases that can create visible progress with limited implementation depth, for example certain content, translation, or internal productivity use cases. Strategic bets are use cases that may require more work, but touch larger operational value pools, such as forecasting, replenishment, pricing, or customer-service automation. Learning cases are somewhat different. Their direct ROI may be more uncertain, but they can help the organisation build AI literacy, confidence, and practical experience with tools, data access, review routines, and

governance.

This distinction is important because broad GenAI adoption does not by itself say where the largest operational value is. GenAI adoption was widespread in the study, and GenAI-based use cases represented a large share of the coded use-case material. At the same time, ML-based use cases had higher average observed impact in the coded material. This does not mean that ML is always better than GenAI. It suggests instead that retailers should avoid treating one technology category as the default answer. Some problems need language generation or summarisation. Others need prediction, optimisation, classification, rules, workflow tooling, or a combination of several techniques.

Forecasting and inventory planning is a useful example of a strategic-bet category. It appeared repeatedly in the quantitative results and had several strong observed-only cluster-strength signals. However, it should not be presented as a light quick win. Compared with some faster GenAI-oriented use cases, it often requires stronger data history, clearer process ownership, maintenance, integration with planning routines, and trust in the recommendations. This is exactly why it belongs in the strategic-bet part of the roadmap. The category can be valuable, but only if the retailer is willing and able to connect the model output to real planning decisions.

5.2.3 Delivery model should depend on readiness, ownership, and strategic importance

The results repeatedly suggest that internally shaped AI use cases can be valuable when the company has the capability and ownership needed to adapt the solution. Internal delivery had the highest average observed impact and also appeared strong in observed-only cluster-strength edges for high data readiness and high change capacity. Internal delivery also had the fastest average time to value in Table 4.5, although this should be interpreted carefully because companies may choose internal delivery for use cases where they already have stronger capability, clearer ownership, or better conditions for implementation.

The implication is not that every retailer should build internally. Instead, the delivery model should be treated as a strategic design choice. Off-the-shelf tools may be suitable for generic needs, especially when the workflow is not strategically differentiating and the data sensitivity is low. External partners may be useful when capability is missing or when speed is more important than internal ownership. Internal development or internal adaptation becomes more relevant when the use case is close to proprietary data, differentiated workflows, or decisions that are central to the retailer's competitive position.

The delivery-source pattern should therefore be read as a capability and ownership question. An internally delivered use case is often closer to the retailer's own data, workflows, and operating priorities. That can make it easier to adapt the solution into something useful, but only when the company has the technical competence, business ownership, and change capacity to absorb the work. Without those conditions,

internal delivery can also become slow, under-resourced, or dependent on a small number of technical specialists.

5.2.4 The central bottleneck is moving from use case to operating practice

One of the clearest practical findings is that the difficult part is often the movement from an AI use case to an operating practice. This is one of the clearest links between the quantitative and qualitative material. In the portfolio results, process-change burden was the most prominent implementation-burden dimension: 44.4 percent of the in-depth use cases had high process-change burden, while integration burden was low for 66.0 percent of use cases. This contrast matters because it suggests that many AI use cases are not blocked mainly by the technical act of connecting systems. They are blocked by the work required to change routines, responsibility, review flows, and user behaviour around the AI output.

The qualitative results make this point more concrete. Section 4.3.1 describes execution depth, business pull, workflow wrappers, frontline adoption, and bandwidth scarcity as recurring themes. These are practical implementation issues. A use case may be technically feasible and still fail to create value if no business team owns it, if employees do not trust it, if the escalation path is unclear, or if the output does not fit into how decisions are actually made. In this sense, the roadmap should not stop at selecting the use case. It should also define what has to change around the use case for it to become part of daily work.

Customer-support agents illustrate this well. As a category, they appear promising in the portfolio results, but the qualitative material suggests that the value depends on more than the underlying model. The use case needs response-quality thresholds, fallback behaviour, escalation rules, customer acceptance, and sometimes API connections or access to order, product, delivery, or policy data. A thin chatbot can become more of a visible experiment than a productivity tool if it cannot answer enough questions reliably. A more carefully designed customer-support agent can reduce workload and improve consistency, but only if the workflow around the agent is designed with the service organisation.

Internal knowledge assistants give another example. The qualitative results suggest that preparing an internal assistant can expose undocumented process knowledge, weak documentation, and unclear business rules. That can be a positive side effect, because the company is forced to examine whether its existing knowledge base is correct and usable. However, it also means that the AI use case inherits problems from the underlying organisation. If the process is unclear, the assistant may surface that lack of clarity rather than solve it. In a roadmap, these use cases should therefore include time for knowledge cleaning, process clarification, ownership design, and model setup together.

This is also where the breadth-versus-depth issue becomes important. The interviews suggest that broad experimentation can be useful for learning, but only if it eventually

leads to disciplined selection and deeper implementation. A portfolio with many proofs of concept may create activity and awareness, but it can also consume time without changing operations. Conversely, going deep too early on a poorly chosen use case can lock resources into something with limited value. A balanced roadmap should therefore separate exploration from scaling. Exploration helps identify where AI fits. Scaling requires ownership, workflow redesign, measurement logic, and sustained attention.

5.2.5 Scaling AI requires explicit operating-model choices

Once AI use cases move beyond isolated experiments, the findings suggest that retailers need more explicit operating-model choices. In this thesis, operating model refers to the practical management system around AI use cases: who owns them, who approves or corrects outputs, what data they may use, how risk is monitored, how employees are trained, and how responsibility is divided between central technical expertise and business domains. This is where many of the qualitative AI challenges come together.

One important choice concerns how AI competence should be distributed. The qualitative material suggests that central AI or data teams remain important, especially for platform choices, security, production quality, and technical standards. At the same time, central teams cannot identify and sustain every valuable use case alone. Retailers also need AI-literate domain experts in merchandising, supply chain, marketing, finance, customer service, store operations, and other areas. These people are closer to the work and can often see where AI would actually help. The challenge is to spread AI capability without allowing fragmented tool use, inconsistent data handling, or duplicated initiatives.

Agentic AI makes this operating-model question more important. The qualitative results show that agent-like systems raise questions about permissioning, delegated authority, auditability, escalation, and data access. These are not purely technical questions. If an AI agent can search internal documents, draft customer communication, create purchase suggestions, or act across systems, the company needs to decide what the agent is allowed to do and who is responsible when the output is wrong. For this reason, agentic AI should probably be roadmapped after the retailer has defined which decisions or actions can be delegated, which require review, and which should remain outside the AI system. Otherwise, the company risks building technically advanced agents before it has decided the boundaries of responsibility. In lower-risk settings, a retailer may accept more automation. In sensitive settings, the same level of autonomy may be inappropriate.

Trust and explanation also need to vary by use case. The Results chapter showed that explainability is especially relevant when AI affects consequential decisions, such as pricing, replenishment, workforce planning, customer communication, or sensitive recommendations. In those settings, users may need to understand why a recommendation was made and when it should be challenged. Customer-facing AI

adds another threshold, because poor answers may reduce customer trust, increase escalation to service staff, or create extra work rather than reduce it. For lower-risk internal productivity tools, the explanation requirement may be lighter. The practical implication is that a roadmap should not define one universal control level for AI. It should define control, review, escalation, and monitoring based on the consequence of the use case.

Data ownership and vendor risk should be treated in the same way. Several qualitative findings show that executives care about where data is stored, whether external providers can access it, and whether it can be used to train models. This becomes especially relevant for LLM-based tools and cloud AI services. A retailer may reasonably allow low-risk experimentation in some areas while requiring stricter contractual, technical, or jurisdictional controls for sensitive company data. The governance model should therefore make safe use possible while still protecting data, customers, employees, and strategically important information.

Overall, scaling AI seems to require a hybrid operating model. Local business ownership is needed because use cases live in specific workflows. Central coordination is still needed because AI introduces shared risks around platforms, data access, security, monitoring, and production quality. If either side is missing, the roadmap becomes weaker. Pure centralization can miss local opportunities and reduce adoption. Pure local experimentation can become fragmented and risky. The more realistic implication is that retailers need both: domain-level ownership for concrete use cases and central guardrails for the common infrastructure and risk questions.

5.2.6 The playbook logic is a sequence of fit questions

Taken together, the results suggest that retail AI roadmapping should be treated as a sequence of fit questions rather than as a universal ranking of use cases. The study does contain average impact values and several strong observed-only edges. These are useful, but they should not be converted mechanically into a ranked list that applies to every retailer. The same AI pattern can have different value depending on data readiness, retail context, delivery model, workflow ownership, and implementation depth.

A practical playbook based on the findings would therefore start with context. What kind of retailer is the company, and where are its largest repeated operational pains? The retail-industry and channel patterns suggest that fashion, grocery, health, store-led, and e-commerce-heavy contexts may create different natural starting points. The next question is readiness. Does the relevant workflow have the data quality, accessibility, ownership, and process clarity needed for AI to work? If not, the first roadmap step may be data or process preparation rather than the AI solution itself.

The third question is portfolio role. Is the use case intended as a quick win, a strategic operating bet, or a learning case? This classification matters because the success criteria differ. A quick win should create visible progress with limited implementation depth. A strategic operating bet should be allowed to require more data, process

change, and governance if the value pool is large enough. A learning case should be judged partly by whether it builds capability and confidence, rather than only by immediate measurable return.

The fourth question is delivery route. Should the use case be bought, built internally, externally supported, or internally adapted around a vendor tool? The evidence suggests that internal delivery can be strong when readiness and change capacity are high, but this should not be turned into a universal rule. The delivery route should match the sensitivity of the data, the strategic importance of the workflow, the internal capability of the company, and the maturity of available vendor solutions.

Finally, the roadmap should ask what operating model is required before scaling. Who owns the use case after launch? What level of review or control is needed? What data can the system access? How will errors be monitored? How will employees learn to use the tool? How will the organisation prevent promising pilots from remaining isolated experiments? These questions may look less exciting than the model or the use-case idea, but the results suggest that they are central to realised value.

The overall discussion therefore points towards a cautious playbook logic. Retailers should look for use cases where value potential and execution fit reinforce each other. The strongest candidates are not necessarily the most fashionable AI applications or the simplest pilots. They are use cases where the company context, data foundation, workflow pain, delivery route, ownership model, and implementation capacity fit together well enough for the AI system to become part of real retail operations.

5.3 Research questions

The discussion can now be tied back to the three research questions introduced in the Introduction. The questions are answered at different levels of certainty. Some answers are more technical, because they concern whether the structured representation can be modelled. Other answers are more practical, because they concern how the analytical outputs can support retail AI roadmapping. In both cases, the answers should be read in light of the empirical setting of the thesis: a relatively small, interview-derived dataset based on senior retail respondents.

RQ1 asked how a structured interview-based dataset can be used to model the expected impact of AI use cases in retail. The thesis shows that this can be done, but with a careful interpretation of what the model output means. The organisation $\times$ use-case representation made it possible to encode company attributes, use-case attributes, and observed impact ratings into a supervised learning problem. The baseline impact modelling results showed that there was some predictive signal in the structured data, especially when focused non-fit feature sets were used with tree-based ensemble models. However, the modest out-of-fold R^2 values and the prediction compression in the diagnostic plots show that the model should not be treated as an exact impact predictor. A more reasonable answer to RQ1 is therefore that structured interview data can support impact-oriented prioritisation analysis,

but mainly as an exploratory ranking and decision-support layer.

RQ2 asked how company attributes, use-case attributes, and computed fit attributes can be organized into a representation that supports more context-sensitive prioritisation. The thesis answers this by separating the representation into company-side context, use-case-side requirements, and fit-oriented logic. This structure made it possible to compare different feature sets in the baseline modelling, inspect attribute importance through interpretability analysis, and reorganize the same material into company-grouping × use-case-grouping relationships in the cluster-strength framework. The final report-facing modelling results relied mainly on directly coded non-fit attributes, since these were easier to interpret and closer to the collected material. The computed fit attributes remain relevant as part of the broader framework design, but the results suggest that they should be used carefully and transparently rather than treated as automatically superior inputs. Overall, the attribute structure worked best when it helped explain fit between a retailer, a workflow, and a use case, rather than when it tried to produce one universal score.

RQ3 asked how the outputs of the structured analytical framework can be translated into practically usable guidance for retail AI prioritisation and playbook-oriented decision support. The main answer is that the outputs should be translated into a roadmapping logic, not a fixed ranking. The descriptive portfolio patterns identify broad differences between AI solution patterns, delivery sources, readiness levels, and implementation-burden profiles. The observed-only cluster-strength results show several context-specific fit signals, such as stronger planning and inventory patterns in grocery-like contexts and stronger internal-delivery edges among more ready companies. The qualitative results then explain why these patterns may occur in practice, for example through data foundations, ownership, workflow fit, implementation depth, and operating-model choices. Taken together, these outputs support a playbook that asks a sequence of fit questions: what kind of retailer is this, where is the workflow pain, what data and capability are available, what portfolio role should the use case have, what delivery route is suitable, and what operating model is needed after launch.

Overall, the three research questions are answered most strongly when the evidence types are combined. The modelling results show that the structured data contains some prioritisation signal. The cluster-strength results translate row-level observations into interpretable company-context and use-case-category patterns. The qualitative material explains the practical conditions behind those patterns. None of these layers is sufficient on its own, but together they support the thesis objective of building a structured and practically interpretable framework for retail AI use-case prioritisation.

5.4 Limitations and boundary conditions

The main limitations of the study follow from the data collection design and the size of the empirical material. The study is based on 33 Nordic retail companies and 55 in-depth impact-rated use cases. This is a useful empirical base for an exploratory MSc thesis, but it is still small for machine learning, especially because the use cases differ in category, maturity, delivery model, business function, and company context. Several cluster-strength edges also have limited observed support. This means that individual high or low cells should be treated as signals for interpretation, not as stable population estimates.

A second limitation is that the target variable is interview-derived. The observed impact rating captures manager-rated expected or perceived impact, not independently measured realised impact after implementation. In addition, the same respondent often contributed both attribute assessments and the impact rating, which creates a possible common-rater effect. This may be part of the reason why some attributes appear predictive. The modelling and interpretability results should therefore be understood as patterns in structured managerial assessment data, rather than as causal evidence about what determines realised AI value.

The technical framework also has boundary conditions. The baseline impact models were evaluated using grouped cross-validation, but the sample was still small, and the strongest models only explained a limited share of the variation in impact ratings. The SHAP, coefficient, and permutation-importance analyses helped interpret model behaviour, but they do not prove causal relationships. The configuration-enabled cluster-strength analysis was useful for testing sensitivity to prototype completion, smoothing, manual adjustments, weighting, and neighbour count, but its completed edges should not be read in the same way as directly observed averages. For this reason, the more business-facing interpretation in this thesis relied mainly on descriptive portfolio patterns, qualitative findings, and observed-only cluster-strength results.

5.5 Future work

Future work should first test the framework on a larger and more independent dataset. A larger sample with more companies, more use cases per company, and more repeated use-case categories would make it possible to evaluate whether the same feature sets, model families, and cluster-strength edges remain stable. It would also be valuable to collect realised outcome data over time, for example measured cost savings, forecast accuracy changes, service-level effects, conversion effects, or working-capital impact. That would make it possible to compare manager-rated expected impact with actual post-implementation outcomes.

A second direction is to refine the technical framework itself. Future work could test the fit attributes more systematically, compare alternative feature-set designs, and evaluate the cluster-strength module with clearer observed-support thresholds. It

would also be useful to test whether the playbook logic improves actual prioritisation decisions in workshops or portfolio reviews. In that setting, the framework could be evaluated by whether it helps managers choose more realistic use cases, identify missing readiness conditions, and avoid scaling weakly owned or poorly supported AI initiatives.

A third possible extension concerns the qualitative interview material. The open-ended notes and attribute comments contain more detail than could be fully used in the current report. One future development would be to anonymize this text material, attach non-identifying metadata such as retail industry type, revenue class, channel profile, and use-case category, and store it in a vector database. A retrieval-augmented generation setup could then retrieve relevant anonymized examples when producing playbook material or decision-support text. This could make the qualitative evidence more usable in later applications, but it would require careful anonymization, source traceability, prompt design, and output checking. The system should only use retrieved material as evidence when the underlying anonymized source can be inspected and understood.

6

Conclusion

This thesis set out to develop a structured and reusable framework for prioritising AI use cases in retail. The starting point was a practical problem: many retail companies can identify many possible AI initiatives, but it is harder to decide which initiatives fit the company, which ones are realistic to scale, and which ones should be prioritised first. To address this, the thesis combined interview-based data collection with structured attribute coding, baseline impact modelling, model interpretability analysis, cluster-strength analysis, and qualitative interpretation.

The empirical basis consisted of interviews with 33 Nordic retail companies. Across these interviews, respondents listed 151 AI use cases, of which 55 were coded in greater depth with structured attributes and manager-rated expected impact. This made it possible to represent the decision problem as an organisation $\times$ use-case dataset, where each row described both the company context and the characteristics of a specific AI use case. A main conclusion is that this type of structured interview data can support AI use-case prioritisation, but it should be treated as decision-support material rather than as a precise prediction system.

The technical results showed that the structured attributes contained some signal for modelling expected impact. Tree-based ensemble models, especially gradient boosting and histogram gradient boosting, performed best in the consolidated non-fit modelling setup. However, the prediction performance was still modest, and the diagnostic plots showed compression towards the middle of the impact scale. This means that the model was more useful for prioritisation support and pattern inspection than for exact impact prediction. The interpretability analysis added another layer by showing which attributes appeared repeatedly in model behaviour, including data recency requirement, data quality, build effort, process change burden, time to value, and operational risk. These signals were useful, but they were diagnostic rather than causal.

The cluster-strength framework extended the analysis by moving from row-level prediction to broader company-grouping $\times$ use-case-grouping relationships. The configuration-enabled analysis showed how the framework behaved under different assumptions, such as prototype completion, smoothing, manual adjustments, weighting, and neighbour count. The observed-only analysis then gave a cleaner empirical view by relying only on directly observed rows. In that stricter setting, several

patterns were especially relevant for roadmapping. Internal delivery appeared strong in companies with high data readiness and high change capacity. Forecasting and inventory planning appeared as a strong operational archetype, especially where data readiness was high. Retail industry context also mattered, with different AI solution patterns appearing more natural in different retail settings.

The qualitative findings helped explain why these quantitative patterns appeared. Retailers repeatedly described AI adoption as constrained by data foundations, ownership, workflow fit, implementation depth, business pull, and operating-model design. Several respondents also raised questions about data ownership, data residency, provider risk, explainability, customer acceptance, and agent-like systems. This means that the practical challenge is not simply to choose the most attractive use-case category. The harder task is to match the use case with the retailer's data conditions, process maturity, delivery route, internal capability, and governance needs.

Taken together, the thesis supports a context-sensitive roadmapping logic. An AI playbook for retailers should not start with one universal ranking of use cases. It should instead ask a sequence of fit questions. What kind of retailer is this? Where is the repeated operational pain? Is the relevant data trusted, accessible, and owned? Is the use case a quick win, a strategic operating bet, or a learning case? Should it be built internally, bought, externally supported, or adapted around a vendor tool? What operating model is needed for the use case to become part of daily work after launch?

The contribution of the thesis is therefore twofold. Technically, it shows how interview-based AI use-case material can be transformed into a layered analytical framework with modelling, interpretability, and cluster-level pattern discovery. Practically, it shows how this framework can support more structured reasoning about retail AI roadmaps. The framework does not remove the need for managerial judgement. Rather, it gives that judgement a clearer empirical and analytical structure. In a setting where AI opportunities are many, but execution capacity is limited, that structure may be one of the most useful outputs of the study.

Bibliography

Breiman, L. (2001). Random Forests. *Machine Learning*, 45, 5–32. https://link.springer.com/article/10.1023/A:1010933404324

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of KDD*. https://arxiv.org/abs/1603.02754

Culot, G., et al. (2024). Artificial intelligence in supply chain management: A systematic literature review. *International Journal of Production Economics*. https://www.sciencedirect.com/science/article/pii/S0166361524000605

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805

Helo, P., & Hao, Y. (2022). Artificial intelligence in operations management and supply chain management. *Production Planning & Control*. https://www.tandfonline.com/doi/full/10.1080/09537287.2021.1882690

Kandakoglu, M., Walther, G., & Ben Amor, S. (2024). The use of multi-criteria decision-making methods in project portfolio selection: a literature review and future research directions. *Annals of Operations Research*. https://link.springer.com/article/10.1007/s10479-023-05564-3

Ke, G., et al. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *NeurIPS*. https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree

Kellerer, H., Pferschy, U., & Pisinger, D. (2004). *Knapsack Problems*. Springer. https://link.springer.com/book/10.1007/978-3-540-24777-7

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. https://arxiv.org/abs/1705.07874

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings-of-t

he-Fifth-Berkeley-Symposium-on-Mathematical-Statistics-and/chapter/Some-methods-for-classification-and-analysis-of-multivariate-observations/bsmsp/1200512992

Nemhauser, G. L., & Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*. Wiley. https://onlinelibrary.wiley.com/doi/book/10.1002/9781118627372

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *NeurIPS*. https://arxiv.org/abs/1706.09516

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1, 206–215. https://www.nature.com/articles/s42256-019-0048-x

Shmueli, G. (2010). To explain or to predict? *Statistical Science*, 25(3), 289–310. https://projecteuclid.org/journals/statistical-science/volume-25/issue-3/To-Explain-or-to-Predict/10.1214/10-STS330.full

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21–27. https://ieeexplore.ieee.org/document/1053964

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1), 39–59. https://www.iiia.csic.es/People/enric/AICom.html

Punj, G., & Stewart, D. W. (1983). Cluster analysis in marketing research: Review and suggestions for application. *Journal of Marketing Research*, 20(2), 134–148. https://journals.sagepub.com/doi/10.1177/002224378302000204

Nickerson, R. C., Varshney, U., & Muntermann, J. (2013). A method for taxonomy development and its application in information systems. *European Journal of Information Systems*, 22, 336–359. https://www.tandfonline.com/doi/abs/10.1057/ejis.2012.26

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. https://arxiv.org/abs/1908.10084

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20, 53–65. https://www.sciencedirect.com/science/article/pii/0377042787901257

Toorajipour, R., et al. (2021). Artificial intelligence in supply chain management: A systematic literature review. *Journal of Business Research*, 122, 502–517. https://ideas.repec.org/a/eee/jbrese/v122y2021icp502-517.html

A

Appendix

A.1 Baseline impact modelling supplement

This appendix contains supplementary material for the baseline impact modelling results in Section 4.4. The purpose is to keep the main Results chapter readable while still making the model comparison auditable. The appendix therefore includes the full attribute composition of the feature sets used in the reported baseline impact modelling results, the extended top-72 non-fit model ranking, and the full top-72 summary plots.

A.1.1 Feature-set inventory

Table A.1 lists the exact structured attributes included in each feature set appearing in the main baseline modelling results and in the extended top-72 ranking. The table is intentionally more detailed than the feature-set glossary in the Results chapter. The main text uses shorter descriptions, while this appendix gives the concrete attribute lists used by the encoded modelling pipeline.

Table A.1: Full feature-set inventory for report-mentioned non-fit baseline impact modelling feature sets.

Feature set	Included attributes
OPF	Implementation label: `o p f`. Attributes: `data_availability`, `data_quality`, `data_accessibility`, `it_competence`, `change_capacity`, `process_governance_maturity`, `data_need`, `data_recency_requirement`, `build_effort`, `process_change_burden`, `integration_burden`, `operational_risk`.

Feature set	Included attributes
UHLC	Implementation label: `u h lc`. Attributes: `primary_ai_goal`, `it_competence`, `change_capacity`, `solution_type`, `kpi_family`, `time_to_value`, `scope_clarity`, `data_need`, `data_recency_requirement`, `build_effort`, `process_change_burden`, `integration_burden`, `operational_risk`.
TBC DQ + PGM	Implementation label: `t b c _ d q _ p gm`. Attributes: `primary_ai_goal`, `data_availability`, `data_quality`, `it_competence`, `change_capacity`, `process_governance_maturity`, `solution_type`, `kpi_family`, `time_to_value`, `data_need`, `build_effort`.
TBC DQ + data access	Implementation label: `t b c _ d q _ d a c c`. Attributes: `primary_ai_goal`, `data_availability`, `data_quality`, `data_accessibility`, `it_competence`, `change_capacity`, `solution_type`, `kpi_family`, `time_to_value`, `data_need`, `build_effort`.
TBC minus time to value	Implementation label: `t i n y _ b a l a n c e d _ c o r e _ _ m i n u s _ t i m e _ t o _ v a l u e`. Attributes: `primary_ai_goal`, `data_availability`, `it_competence`, `change_capacity`, `solution_type`, `kpi_family`, `data_need`, `build_effort`.
TBC plus data quality	Implementation label: `tiny_balanced_core__plus_data_quality`. Attributes: `primary_ai_goal`, `data_availability`, `data_quality`, `it_competence`, `change_capacity`, `solution_type`, `kpi_family`, `time_to_value`, `data_need`, `build_effort`.
Tiny balanced core	Implementation label: `tiny_balanced_core`. Attributes: `primary_ai_goal`, `data_availability`, `it_competence`, `change_capacity`, `solution_type`, `kpi_family`, `time_to_value`, `data_need`, `build_effort`.
TBC plus scope clarity	Implementation label: `t i n y _ b a l a n c e d _ c o r e _ _ p l u s _ s c o p e _ c l a r i t y`. Attributes: `primary_ai_goal`, `data_availability`, `it_competence`, `change_capacity`, `solution_type`, `kpi_family`, `time_to_value`, `scope_clarity`, `data_need`, `build_effort`.

Feature set	Included attributes
Reduced raw	Implementation label: `reduced_raw`. Attributes: `primary_ai_goal`, `channel_profile`, `data_availability`, `data_accessibility`, `it_competence`, `change_capacity`, `process_governance_maturity`, `privacy_mindedness`, `solution_type`, `kpi_family`, `time_to_value`, `scope_clarity`, `data_need`, `data_recency_requirement`, `integration_burden`, `process_change_burden`, `build_effort`, `operational_risk`, `privacy_friendliness`.
Full raw	Implementation label: `full_raw`. Attributes: `primary_ai_goal`, `size_class`, `channel_profile`, `supply_chain_complexity`, `data_availability`, `data_quality`, `data_accessibility`, `it_competence`, `change_capacity`, `process_governance_maturity`, `privacy_mindedness`, `compliance_constraint_level`, `sourcing_preference`, `ai_dependence_tolerance`, `workflow_change_willingness`, `usecase_family`, `solution_type`, `delivery_source`, `kpi_family`, `time_to_value`, `roi_metric_clarity`, `scope_clarity`, `organizational_breadth_relevance`, `data_need`, `data_recency_requirement`, `build_effort`, `process_change_burden`, `integration_burden`, `maintenance_cost`, `human_in_the_loop_need`, `operational_risk`, `privacy_friendliness`, `compliance_sensitivity`, `required_interpretability`, `reversibility_of_wrong_decisions`.
TBC minus build effort	Implementation label: `tiny_balanced_core__minus_build_effort`. Attributes: `primary_ai_goal`, `data_availability`, `it_competence`, `change_capacity`, `solution_type`, `kpi_family`, `time_to_value`, `data_need`.
CHLU	Implementation label: `chlu`. Attributes: `primary_ai_goal`, `channel_profile`, `data_availability`, `data_accessibility`, `it_competence`, `change_capacity`, `process_governance_maturity`, `sourcing_preference`, `ai_dependence_tolerance`, `solution_type`, `time_to_value`, `data_need`, `build_effort`.

A.1.2 Top-72 non-fit model ranking

Table A.2 extends the main top-25 ranking to the top 72 consolidated non-fit experiments. The table is included to make the model-family and feature-set patterns more transparent beyond the strongest rows shown in the main Results chapter.

Table A.2: Extended top-72 consolidated non-fit baseline impact modelling experiments ranked by MAE.

Rank	Feature set	Model	MAE	RMSE	OOF R^2	Spearman	Top-1
1	OPF	Hist. gradient boosting	1.348	1.736	0.110	0.372	0.474
2	OPF	Gradient boosting	1.352	1.714	0.132	0.439	0.579
3	OPF	Random forest	1.394	1.780	0.064	0.275	0.421
4	UHLC	Gradient boosting	1.418	1.851	-0.012	0.333	0.579
5	TBC DQ + PGM	Hist. gradient boosting	1.421	1.864	-0.026	0.144	0.658
6	UHLC	SVR	1.437	1.871	-0.033	-0.015	0.684
7	TBC minus time to value	SVR	1.438	1.867	-0.029	0.014	0.632
8	OPF	Ridge	1.439	1.834	0.006	0.144	0.316
9	TBC plus data quality	SVR	1.445	1.873	-0.035	0.031	0.632
10	TBC DQ + data access	SVR	1.445	1.873	-0.036	0.017	0.684
11	Tiny balanced core	SVR	1.446	1.879	-0.043	0.019	0.632
12	TBC DQ + PGM	SVR	1.450	1.878	-0.041	-0.001	0.632
13	TBC plus scope clarity	SVR	1.453	1.878	-0.042	-0.003	0.632
14	Reduced raw	SVR	1.453	1.875	-0.038	-0.116	0.684
15	Full raw	SVR	1.457	1.887	-0.052	-0.165	0.579
16	OPF	SVR	1.459	1.866	-0.028	-0.029	0.526
17	TBC plus data quality	Hist. gradient boosting	1.459	1.874	-0.037	0.124	0.632
18	TBC minus build effort	SVR	1.460	1.899	-0.065	-0.066	0.579
19	CHLU	Random forest	1.461	1.833	0.008	0.127	0.632
20	OPF	Extra trees	1.470	1.875	-0.038	0.137	0.526
21	Reduced raw	Gradient boosting	1.474	1.882	-0.046	0.267	0.579
22	TBC DQ + data access	Hist. gradient boosting	1.476	1.884	-0.048	0.134	0.658
23	TBC DQ + data access	Random forest	1.476	1.878	-0.042	0.047	0.579
24	TBC DQ + PGM	Random forest	1.478	1.868	-0.030	0.068	0.579
25	CHLU	Elastic net	1.483	1.856	-0.017	0.040	0.583
26	TBC plus data quality	Random forest	1.486	1.848	-0.008	0.124	0.684
27	Tiny balanced core	Hist. gradient boosting	1.486	1.913	-0.081	0.067	0.553
28	TBC plus scope clarity	Hist. gradient boosting	1.486	1.913	-0.081	0.067	0.553
29	CHLU	SVR	1.487	1.882	-0.046	0.020	0.737
30	CHLU	PLS regression	1.488	1.853	-0.014	0.193	0.632
31	OPF	PLS regression	1.488	1.861	-0.022	0.223	0.579
32	UHLC	Random forest	1.489	1.864	-0.026	0.102	0.474
33	TBC plus data quality	Ridge	1.493	1.886	-0.050	0.105	0.579
34	UHLC	Hist. gradient boosting	1.493	1.923	-0.092	0.156	0.474
35	TBC DQ + data access	Gradient boosting	1.497	1.938	-0.109	0.113	0.526
36	CHLU	Extra trees	1.501	1.879	-0.043	0.043	0.526
37	Tiny balanced core	Random forest	1.502	1.886	-0.050	0.075	0.474
38	TBC DQ + data access	Ridge	1.503	1.901	-0.068	0.093	0.579
39	OPF	Bayesian ridge	1.503	1.866	-0.028	-0.083	0.579
40	TBC minus time to value	Hist. gradient boosting	1.503	1.915	-0.083	0.070	0.553
41	Full raw	Random forest	1.507	1.874	-0.037	0.152	0.526
42	Reduced raw	Random forest	1.509	1.899	-0.065	0.013	0.526
43	CHLU	Bayesian ridge	1.510	1.872	-0.035	-0.074	0.684
44	Full raw	Extra trees	1.511	1.884	-0.048	0.086	0.579
45	TBC minus time to value	Elastic net	1.513	1.880	-0.044	-0.273	0.461
46	OPF	Elastic net	1.513	1.882	-0.046	-0.117	0.373

Rank	Feature set	Model	MAE	RMSE	OOF R^2	Spearman	Top-1
47	TBC DQ + data access	Dummy mean	1.514	1.880	-0.044	-0.277	0.461
48	TBC DQ + data access	Lasso	1.514	1.880	-0.044	-0.277	0.461
49	TBC DQ + PGM	Dummy mean	1.514	1.880	-0.044	-0.277	0.461
50	TBC DQ + PGM	Lasso	1.514	1.880	-0.044	-0.277	0.461
51	CHLU	Dummy mean	1.514	1.880	-0.044	-0.277	0.461
52	CHLU	Lasso	1.514	1.880	-0.044	-0.277	0.461
53	UHLC	Dummy mean	1.514	1.880	-0.044	-0.277	0.461
54	UHLC	Lasso	1.514	1.880	-0.044	-0.277	0.461
55	OPF	Dummy mean	1.514	1.880	-0.044	-0.277	0.461
56	OPF	Lasso	1.514	1.880	-0.044	-0.277	0.461
57	Tiny balanced core	Dummy mean	1.514	1.880	-0.044	-0.277	0.461
58	Tiny balanced core	Lasso	1.514	1.880	-0.044	-0.277	0.461
59	TBC minus build effort	Dummy mean	1.514	1.880	-0.044	-0.277	0.461
60	TBC minus time to value	Dummy mean	1.514	1.880	-0.044	-0.277	0.461
61	TBC minus time to value	Lasso	1.514	1.880	-0.044	-0.277	0.461
62	TBC plus scope clarity	Dummy mean	1.514	1.880	-0.044	-0.277	0.461
63	TBC plus scope clarity	Lasso	1.514	1.880	-0.044	-0.277	0.461
64	TBC plus data quality	Dummy mean	1.514	1.880	-0.044	-0.277	0.461
65	TBC plus data quality	Lasso	1.514	1.880	-0.044	-0.277	0.461
66	Reduced raw	Dummy mean	1.514	1.880	-0.044	-0.277	0.461
67	Reduced raw	Lasso	1.514	1.880	-0.044	-0.277	0.461
68	UHLC	Elastic net	1.515	1.880	-0.044	-0.273	0.443
69	Reduced raw	Elastic net	1.515	1.880	-0.044	-0.273	0.443
70	TBC DQ + PGM	Ridge	1.516	1.915	-0.082	0.057	0.579
71	TBC plus scope clarity	Random forest	1.517	1.907	-0.074	-0.011	0.474
72	Reduced raw	Bayesian ridge	1.521	1.884	-0.048	-0.114	0.474

A.1.3 Top-72 summary plots

Figures A.1–A.4 show the same summary metrics as the main Results section, but for the broader top-72 non-fit experiment set. These plots are included mainly as an audit view. They make it possible to inspect whether the patterns discussed in the

top-25 results remain visible when the ranking window is widened.

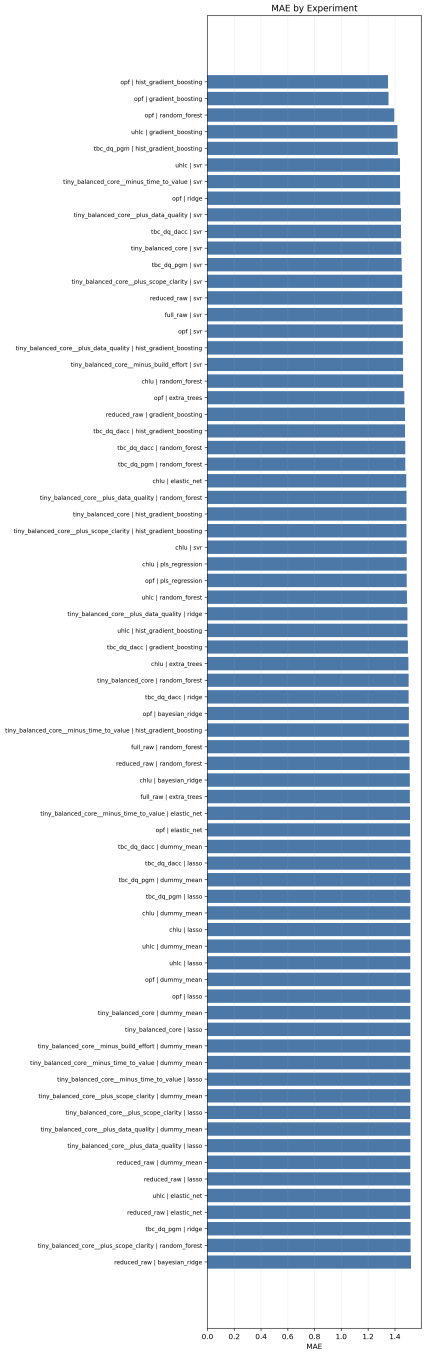


Figure A.1: Extended top-72 MAE ranking for consolidated non-fit baseline impact modelling experiments.

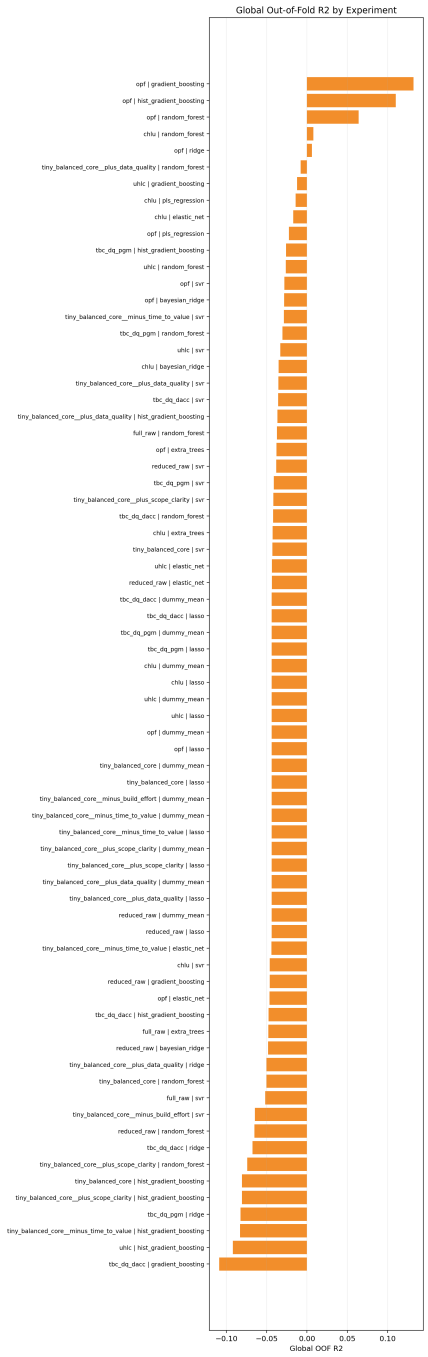


Figure A.2: Extended top-72 global out-of-fold R^2 summary for consolidated non-fit baseline impact modelling experiments.

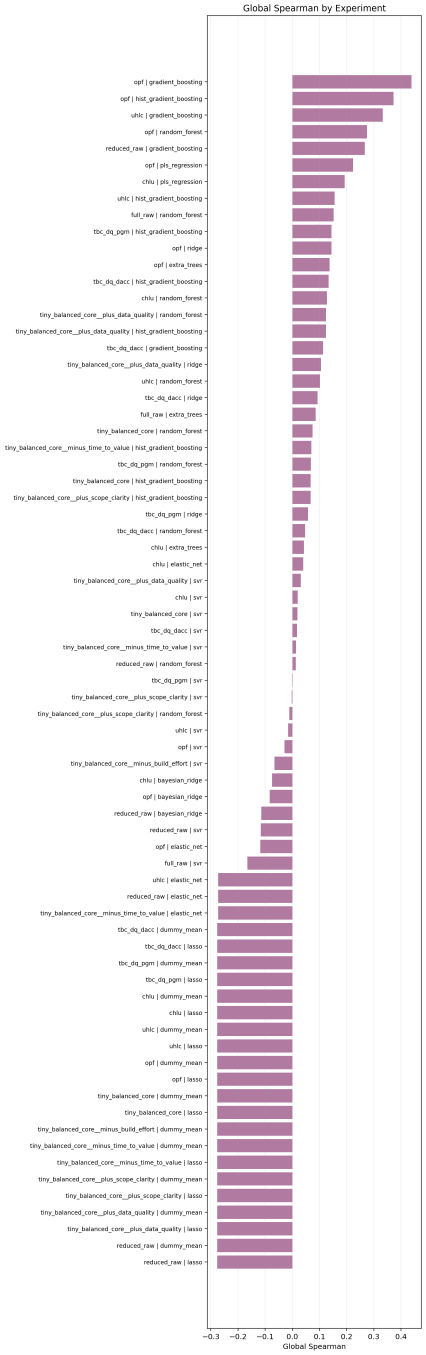


Figure A.3: Extended top-72 global Spearman summary for consolidated non-fit baseline impact modelling experiments.

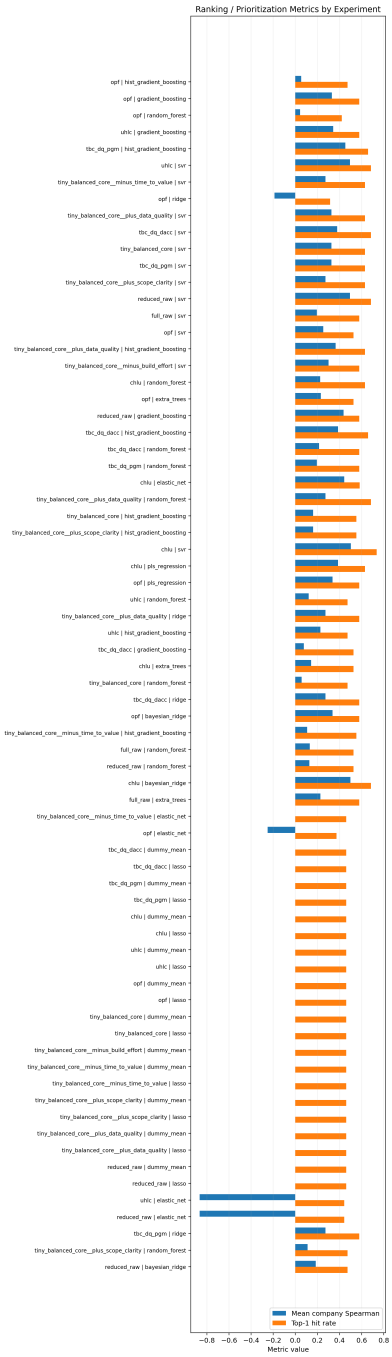


Figure A.4: Extended top-72 ranking-metrics summary for consolidated non-fit baseline impact modelling experiments.

A.2 Interpretability supplement

This appendix section contains supplementary plots for the impact model interpretability analysis in Section 4.5. The main Results chapter only includes selected feature-level plots, because the central argument is based mostly on the aggregate coefficient, SHAP, and permutation heatmaps. The appendix therefore provides the more detailed feature-level outputs for transparency.

A.2.1 Coefficient plots

Figures A.5, A.6, and A.7 show the generated feature-level coefficient plots. Two of these are also discussed in the main Results chapter, because they are directly relevant for the proposed manual-adjustment settings. They are repeated here so that the generated coefficient-detail plots are collected in one place.

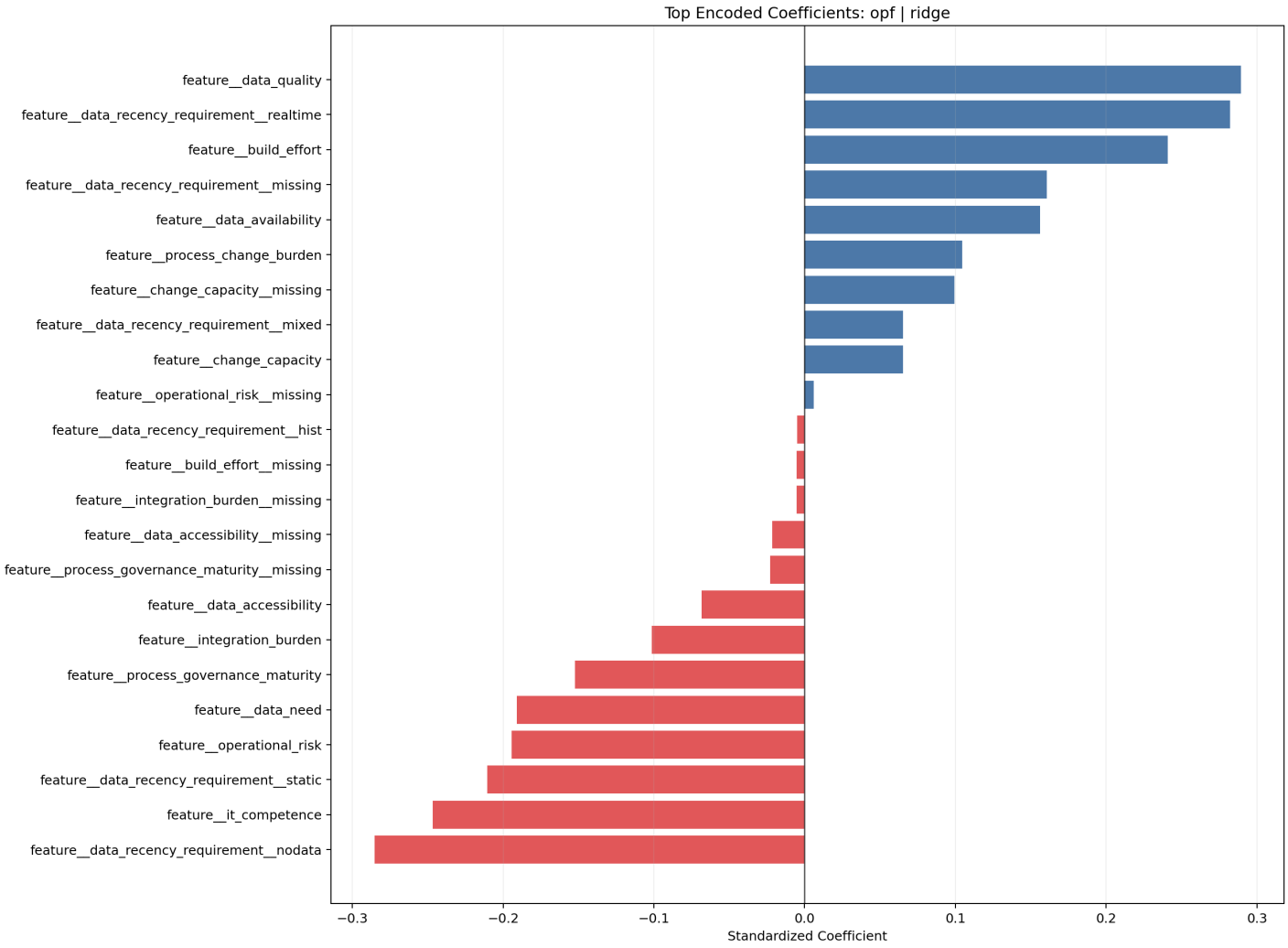


Figure A.5: Feature-level coefficient plot for `opf` with ridge regression.

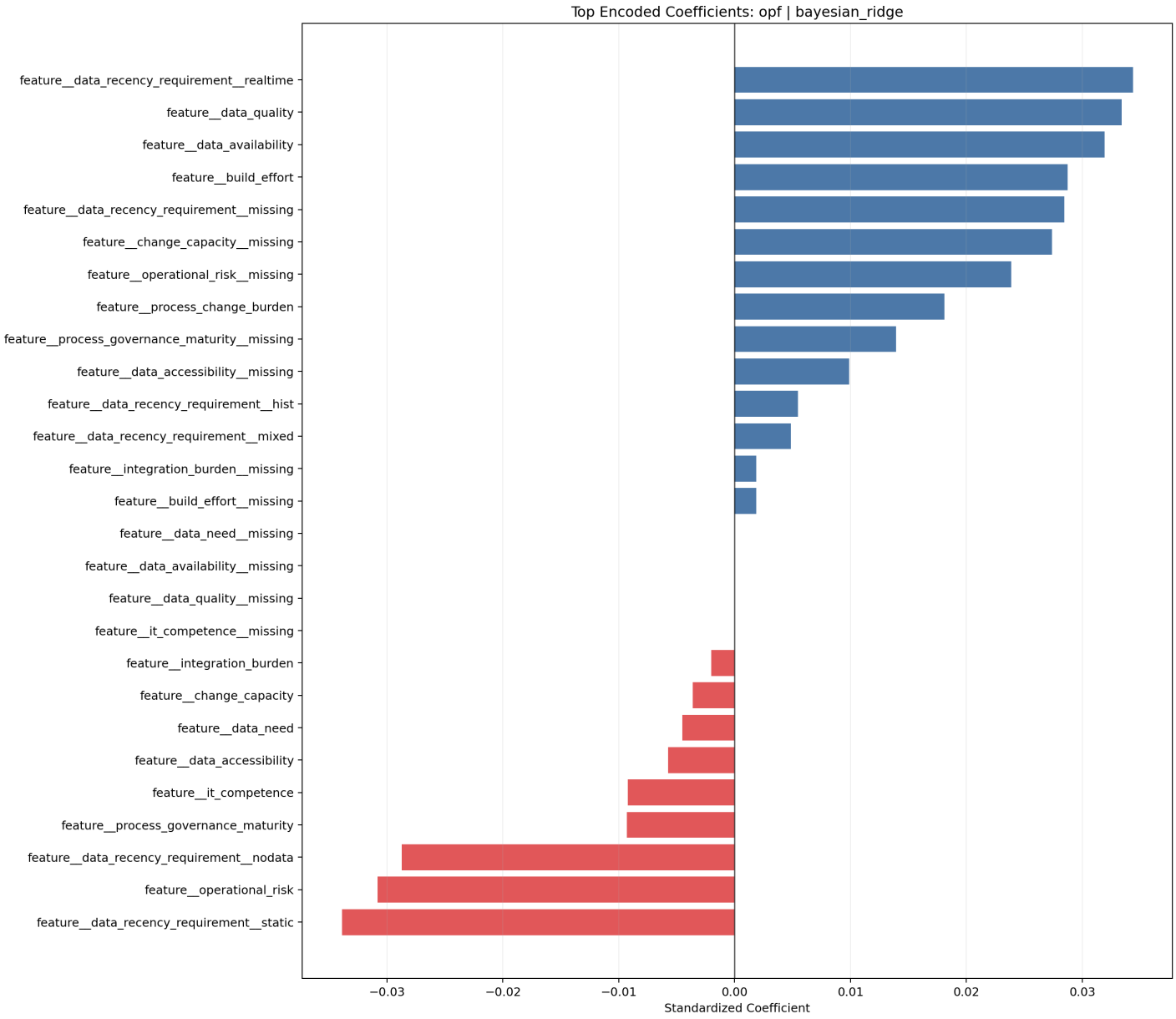


Figure A.6: Feature-level coefficient plot for `opf` with Bayesian ridge regression.

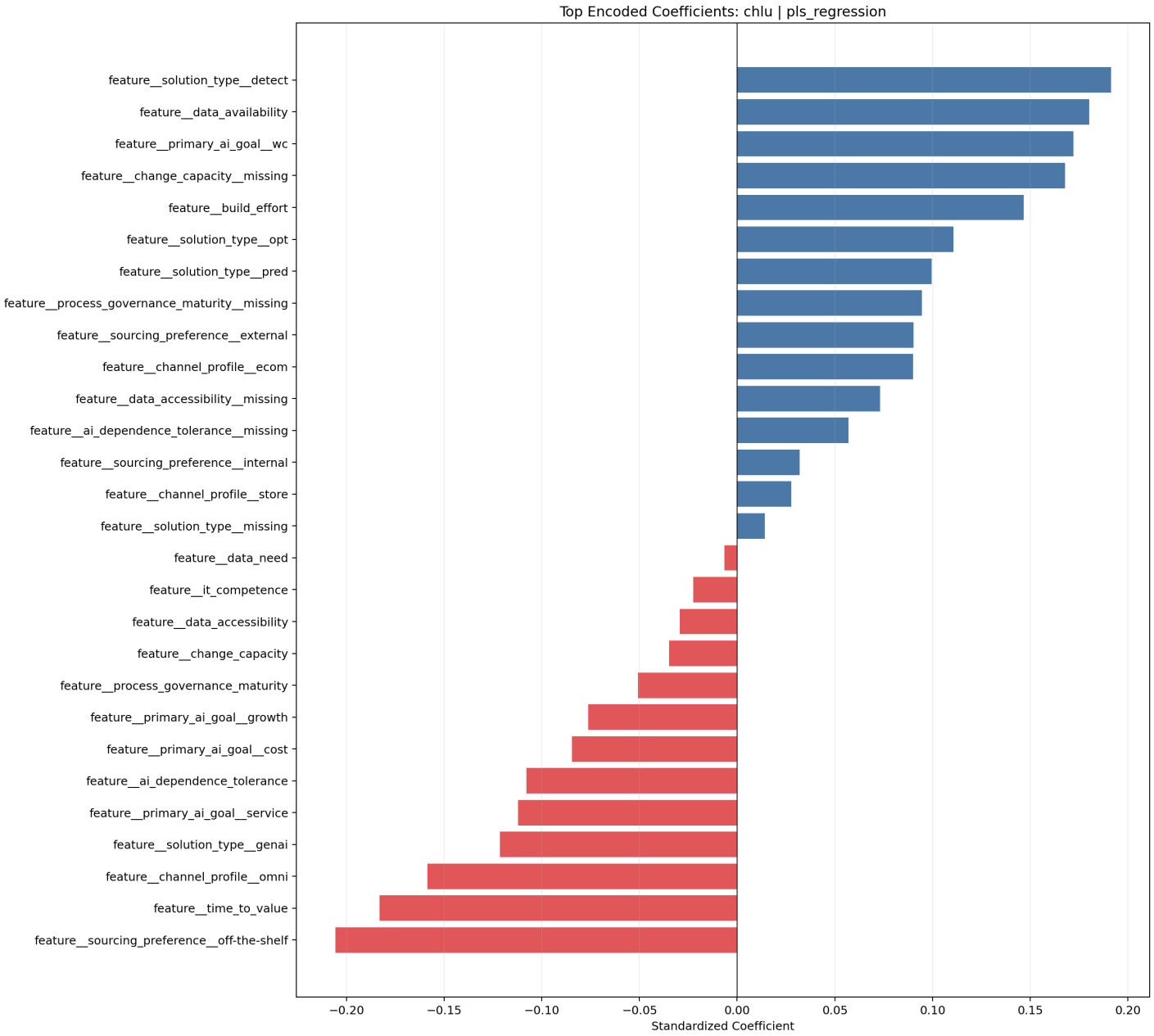


Figure A.7: Feature-level coefficient plot for `chlu` with PLS regression.

A.2.2 SHAP plots

Figures A.8–A.13 show the detailed feature-level SHAP plots for the selected non-linear tree and boosting models. These figures are useful for inspecting encoded categories directly, especially for attributes such as `data_recency_requirement`, where different categories can point in different directions.

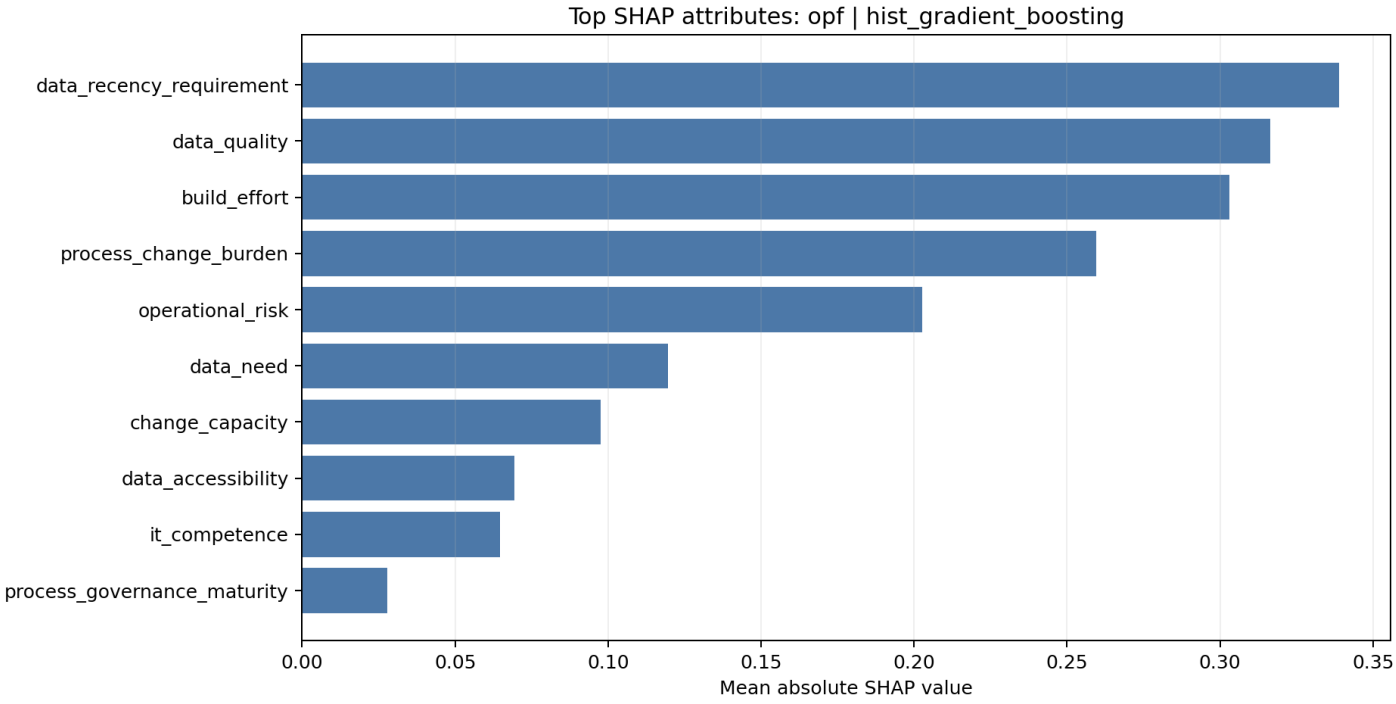


Figure A.8: Feature-level SHAP plot for `opf` with histogram gradient boosting.

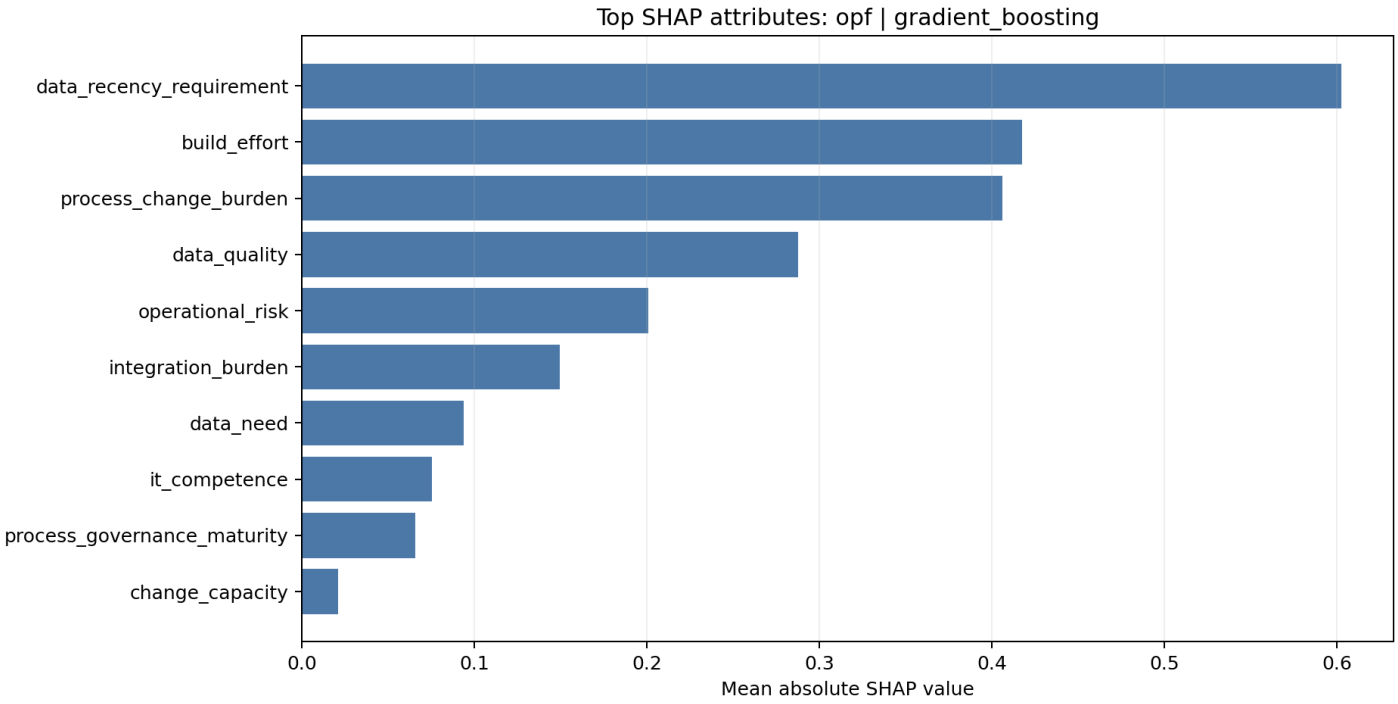


Figure A.9: Feature-level SHAP plot for `opf` with gradient boosting.

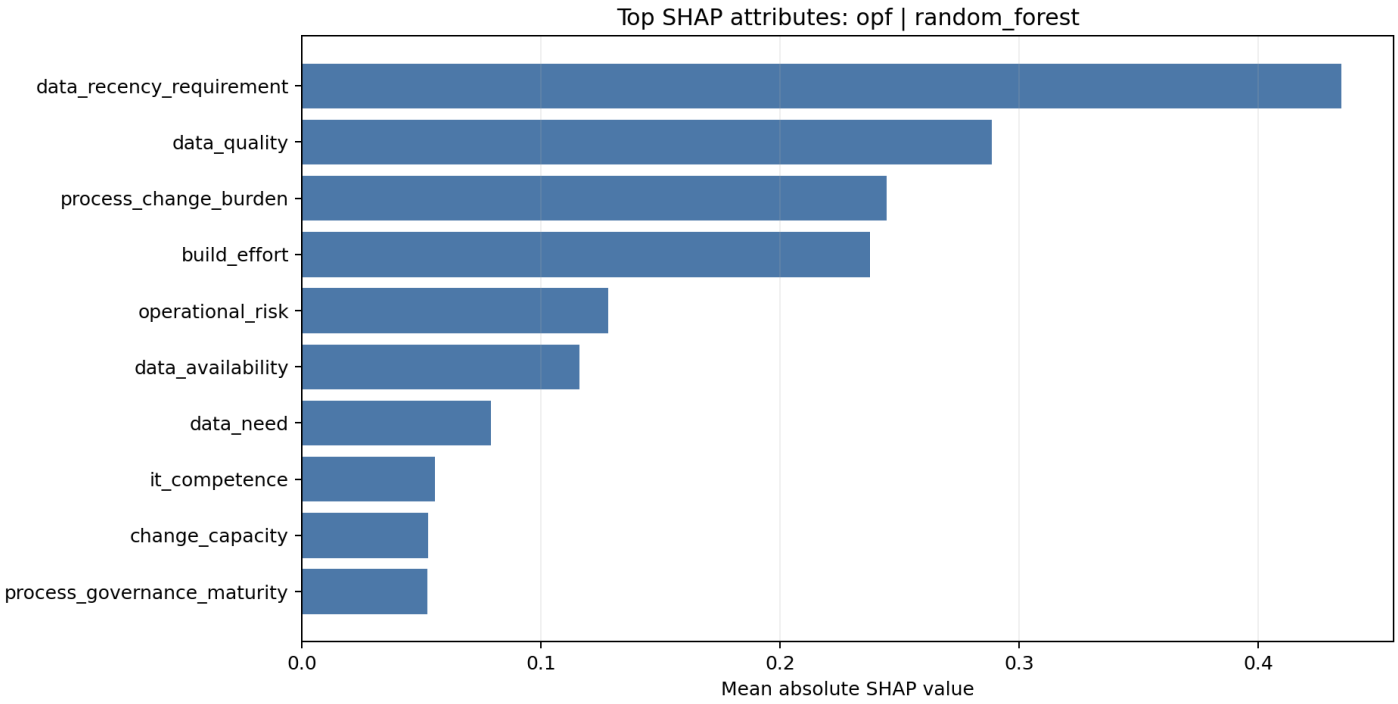


Figure A.10: Feature-level SHAP plot for `opf` with random forest.

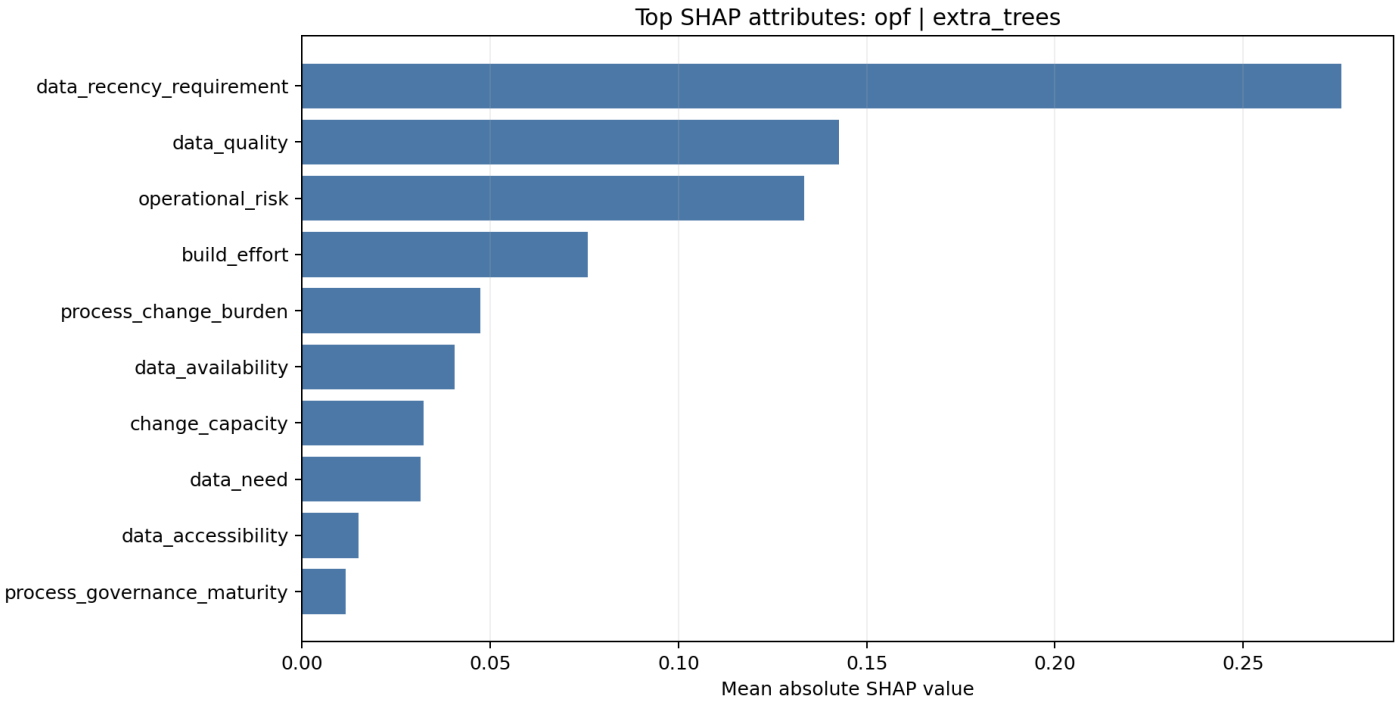


Figure A.11: Feature-level SHAP plot for `opf` with extra trees.

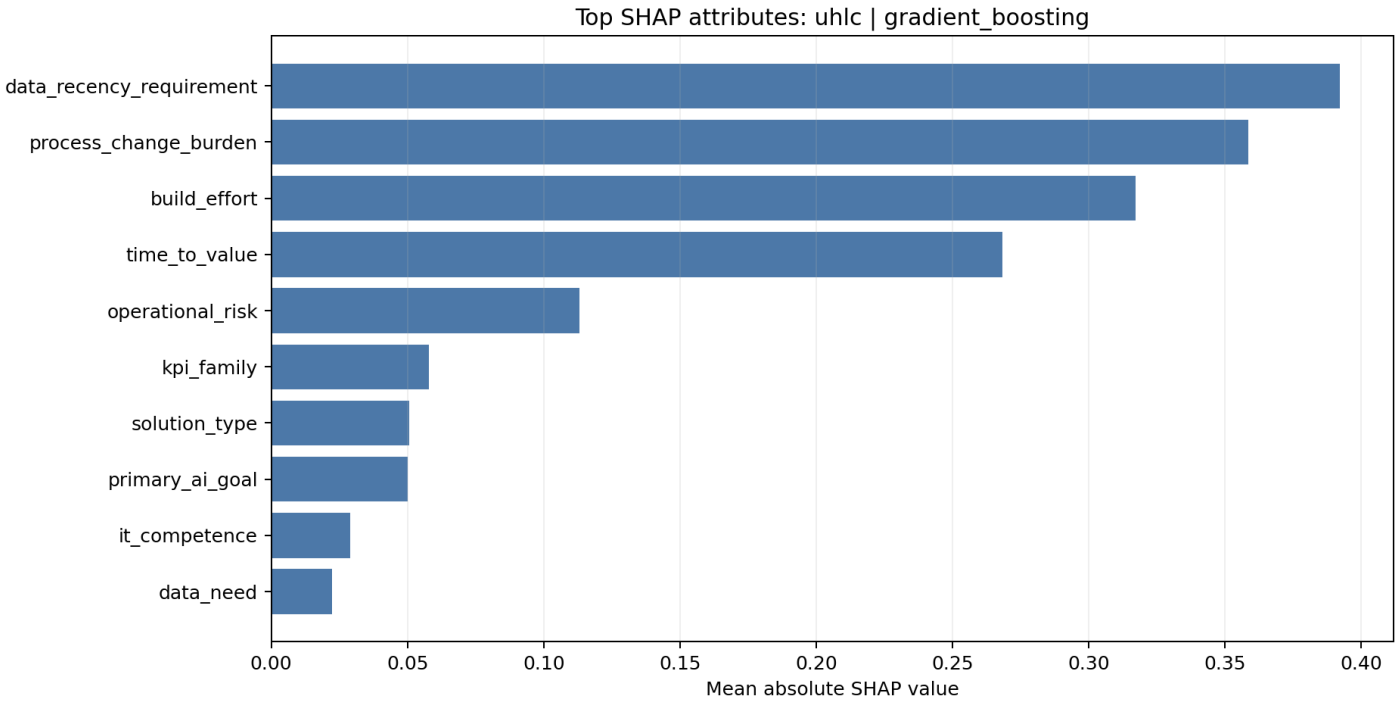


Figure A.12: Feature-level SHAP plot for `uhlc` with gradient boosting.

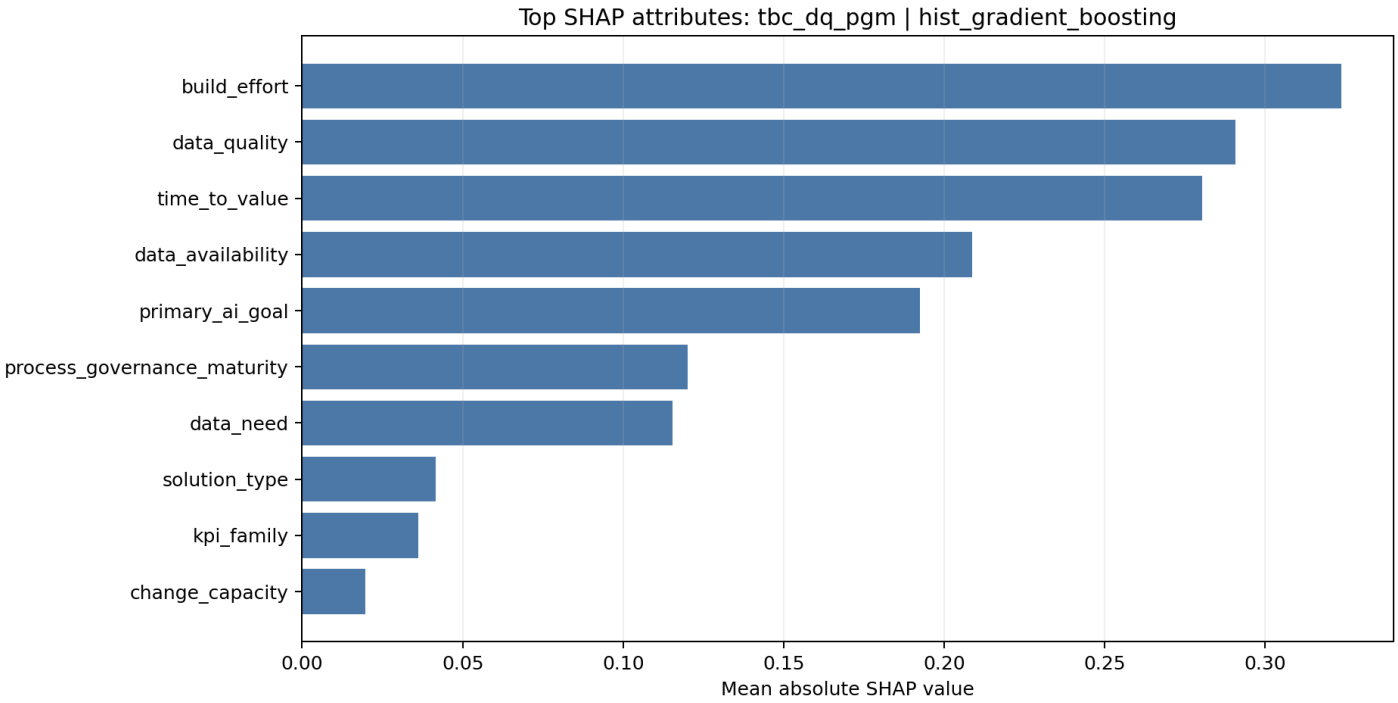


Figure A.13: Feature-level SHAP plot for `tbc_dq_pgm` with histogram gradient boosting.

A.2.3 Permutation plots

Figures A.14–A.21 show the feature-level permutation-importance plots. The first six correspond to the same tree and boosting models used in the SHAP analysis, while the final two correspond to selected SVR models. These plots are included mainly as an audit view of which feature groups caused the largest prediction-error increase when permuted.

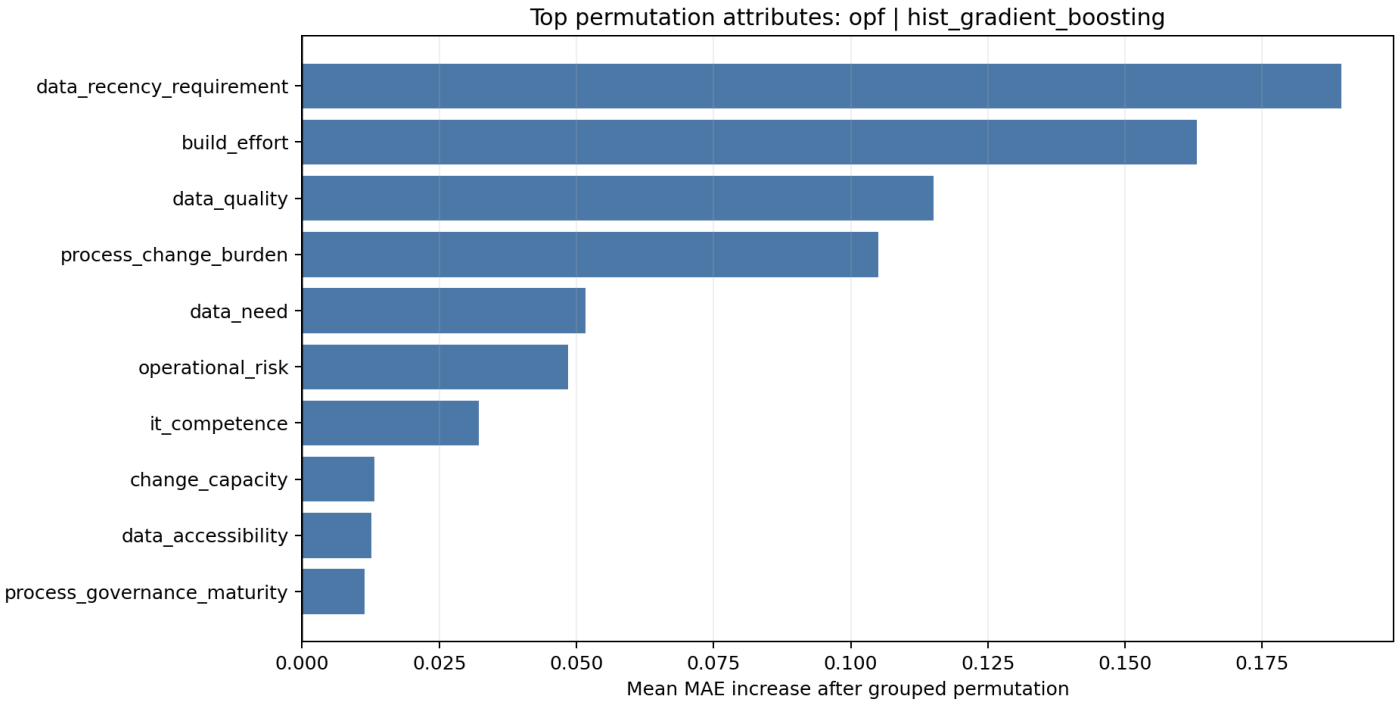


Figure A.14: Feature-level permutation-importance plot for `opf` with histogram gradient boosting.

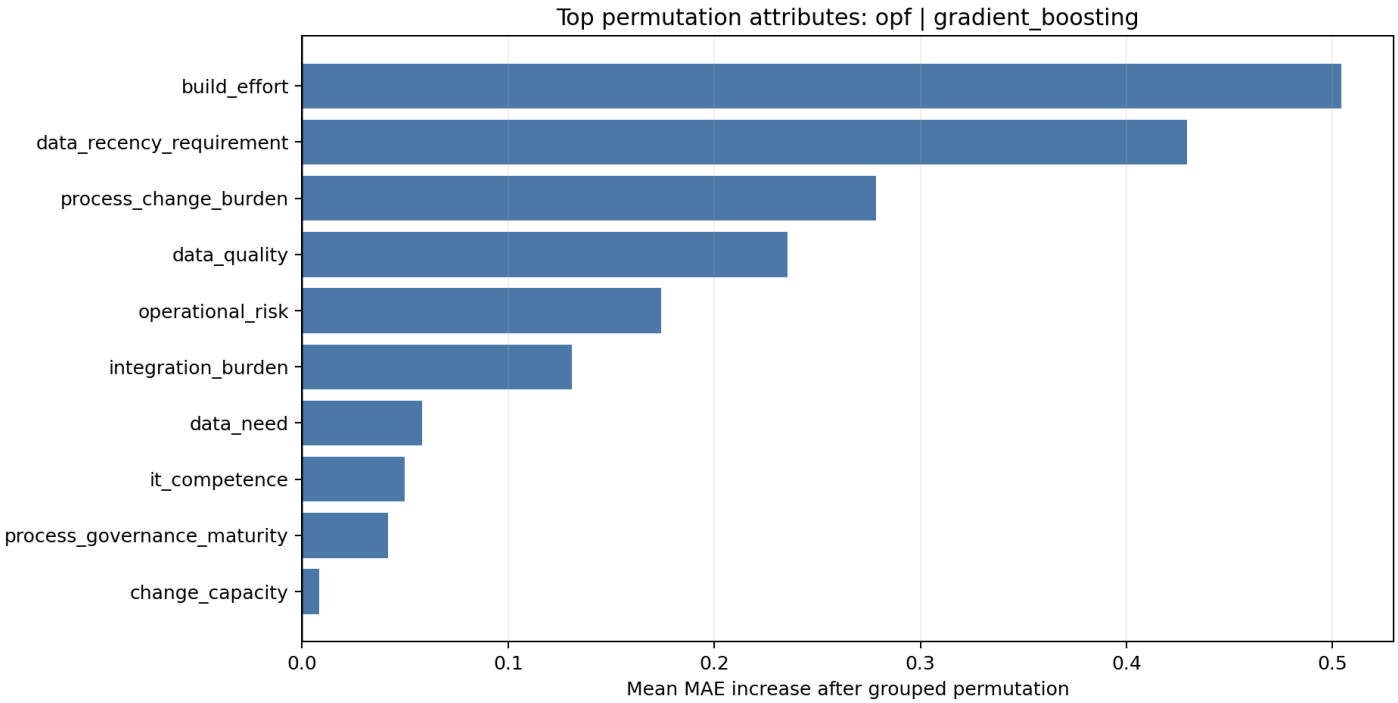


Figure A.15: Feature-level permutation-importance plot for `opf` with gradient boosting.

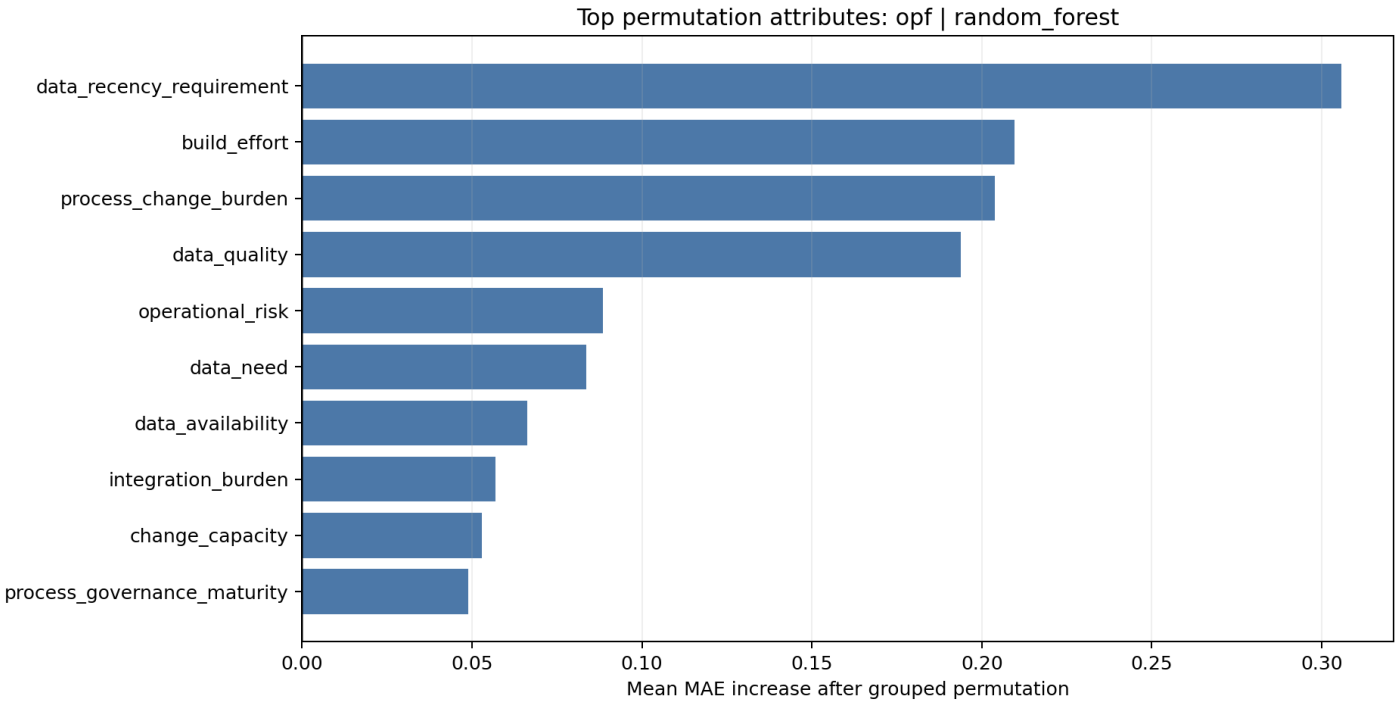


Figure A.16: Feature-level permutation-importance plot for `opf` with random forest.

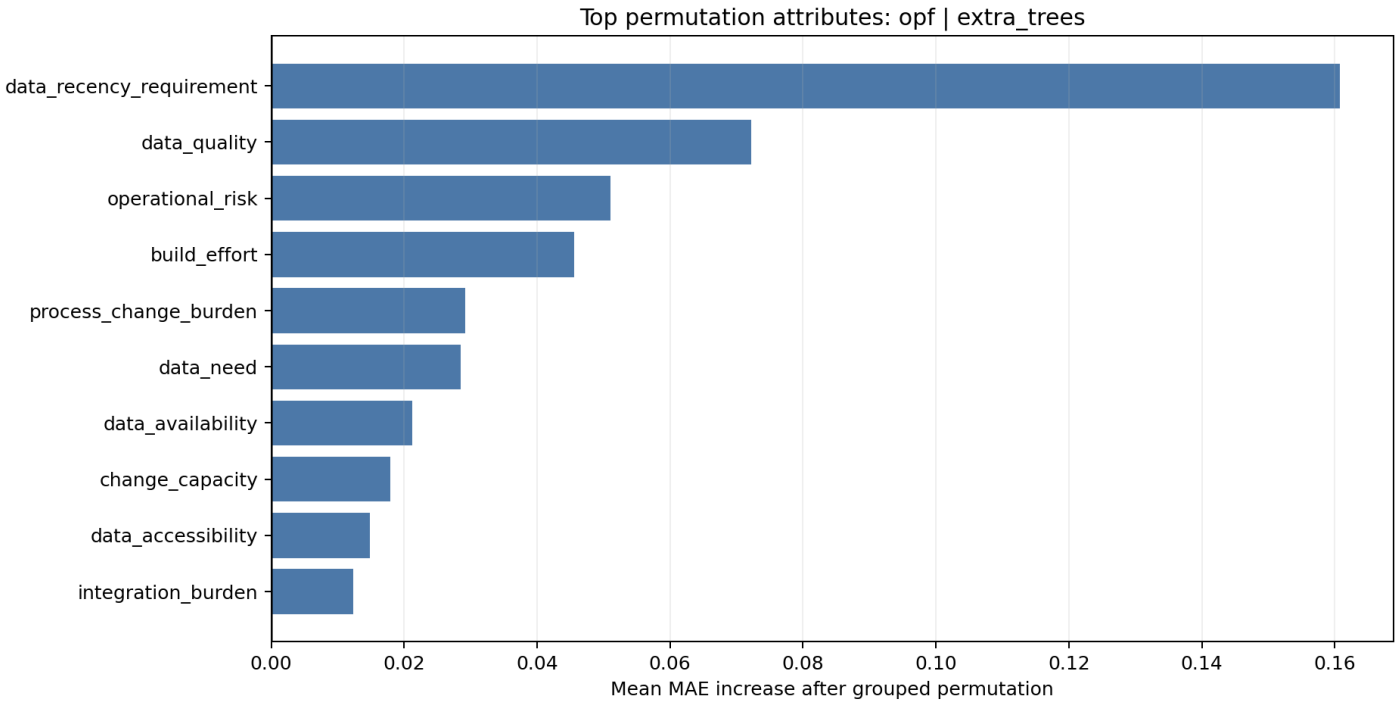


Figure A.17: Feature-level permutation-importance plot for `opf` with extra trees.

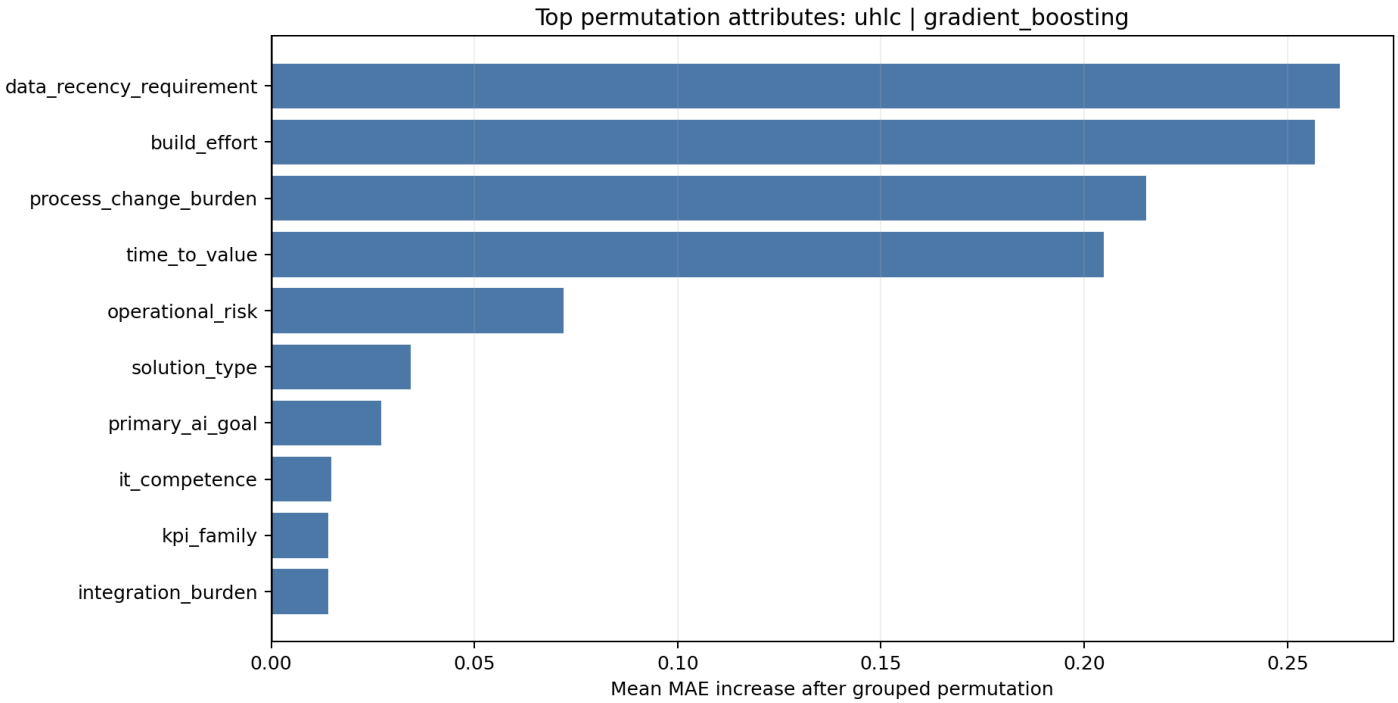


Figure A.18: Feature-level permutation-importance plot for `uhlc` with gradient boosting.

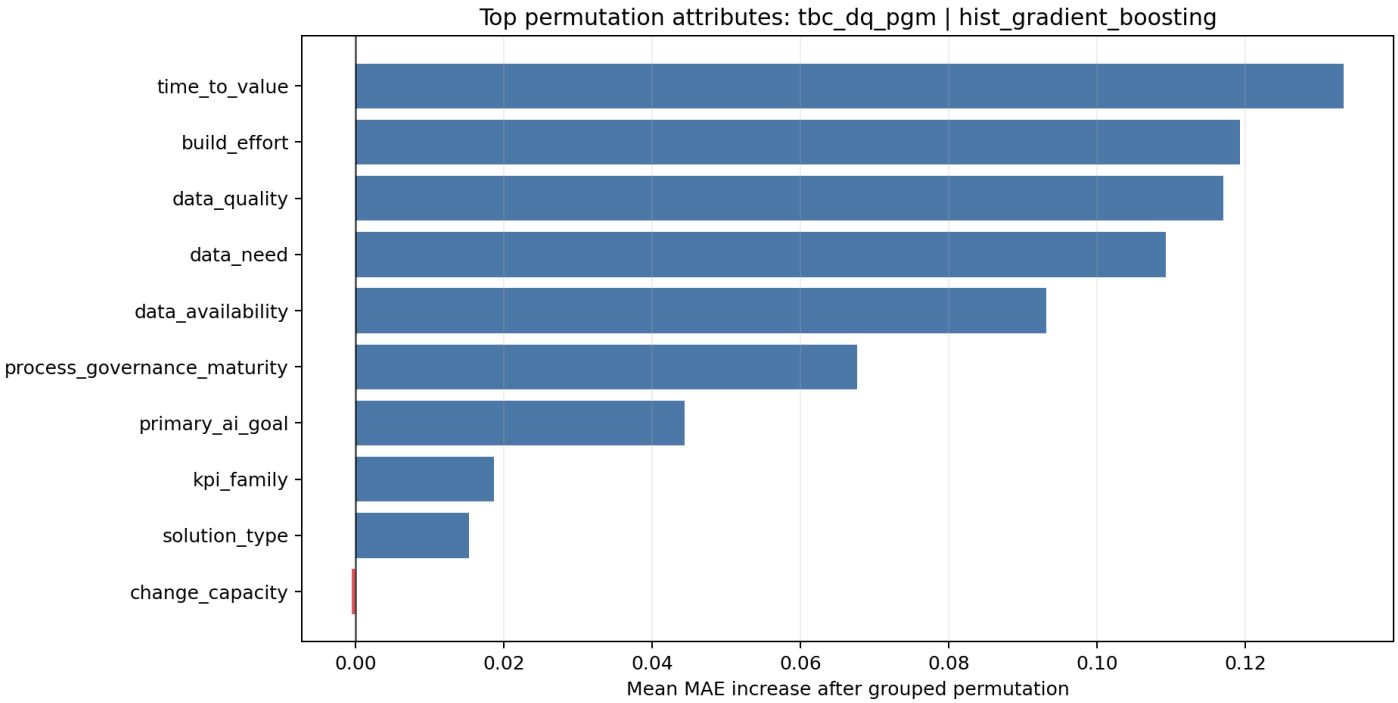


Figure A.19: Feature-level permutation-importance plot for `tbc_dq_pgm` with histogram gradient boosting.

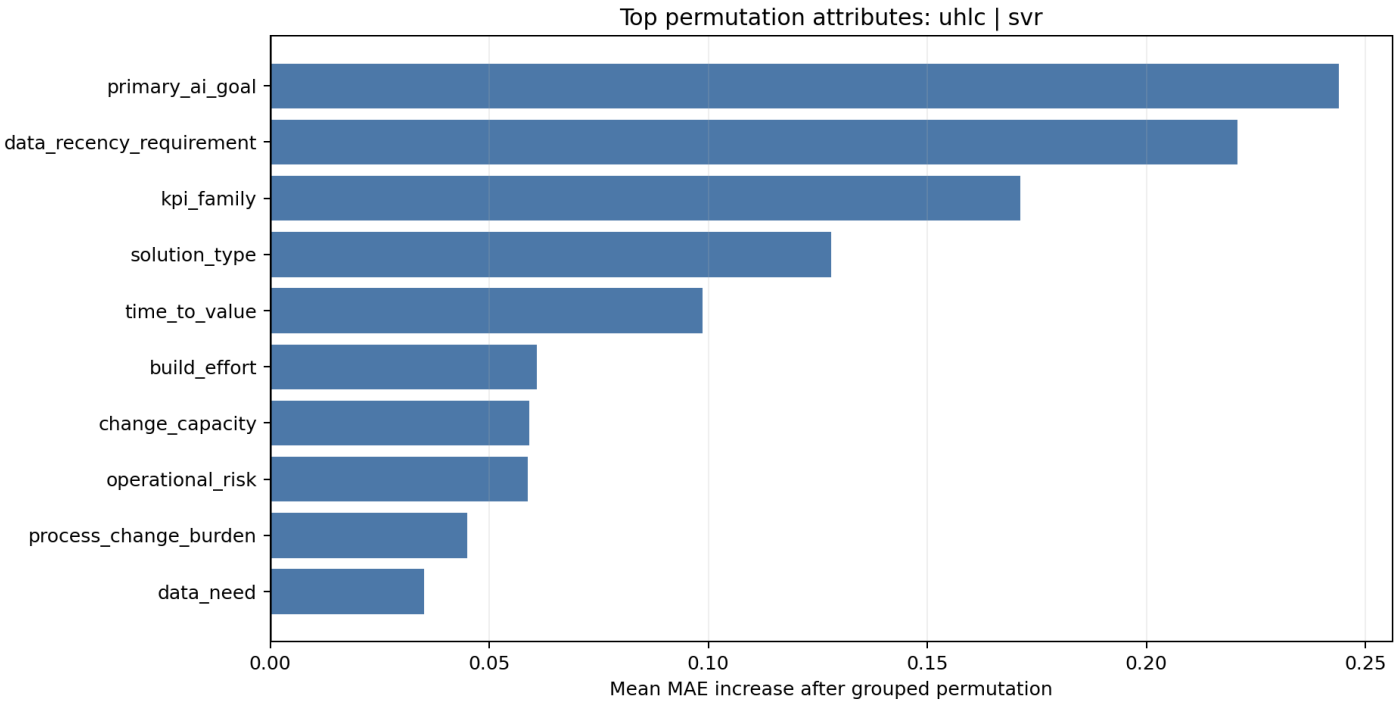


Figure A.20: Feature-level permutation-importance plot for `uhlc` with SVR.

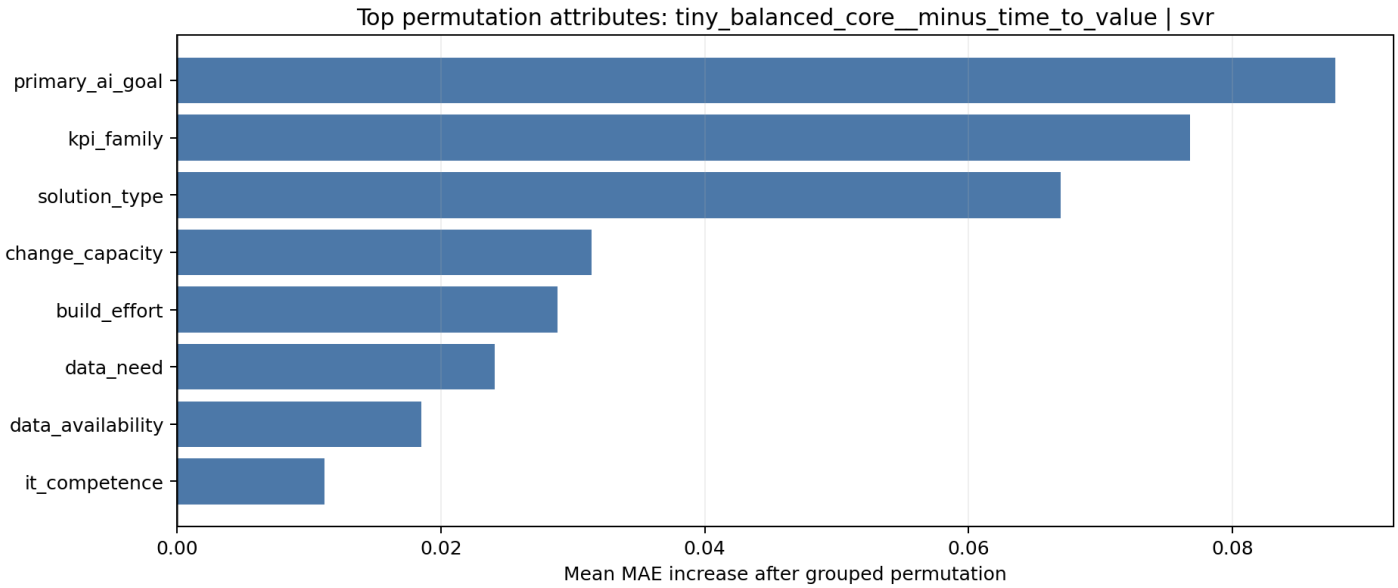


Figure A.21: Feature-level permutation-importance plot for `tiny_balanced_core` without time to value, with SVR.

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden – www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY