CHALMERS

Derome

Optimizing Pricing Strategies Using Machine Learning: A Prototype Development for Derome

Enhancing Business Intelligence with AI-Driven Pricing Precision and Strategic Decision-Making

Bachelor's Thesis in Computer Engineering

Kitty Ha
Che Long Tran

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Bachelor's Thesis 2025

Optimizing Pricing Strategies Using Machine Learning: A Prototype Development for Derome

Enhancing Business Intelligence with AI-Driven Pricing Precision and Strategic Decision-Making

Kitty Ha

Che Long Tran

CHALMERS

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden 2025

Optimizing Pricing Strategies Using Machine Learning: A Prototype Development for Derome
Enhancing Business Intelligence with AI-Driven Pricing Precision and Strategic Decision-Making
Kitty Ha
Che Long Tran

© Kitty Ha, 2025.
© Che Long Tran, 2025.

Supervisor: Robert Zarins, Project Manager at Derome Bygg & Industri AB
Supervisor: Oana Geman, Department of Computer Science and Engineering
Examiner: Nicholas Smallbone, Department of Computer Science and Engineering

Bachelor's Thesis 2025
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A logo of the company Derome.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Optimizing Pricing Strategies Using Machine Learning: A Prototype Development for Derome
Enhancing Business Intelligence with AI-Driven Pricing Precision and Strategic Decision-Making
Kitty Ha
Che Long Tran
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

This project explored the application of machine learning models to analyze and forecast sales performance using internal transaction data from Derome, a Swedish company. The study implemented clustering, classification, and regression techniques to build interpretable, high-performing prototypes aimed at enhancing decision-making and revenue forecasting.

For clustering, k-means was applied to annual aggregates of product groups based on sales volume and outcome. Using the elbow method, three distinct performance tiers were identified, validated by a high silhouette score of 0.89. These tiers were assigned to transactions and used as labels in a classification task. Among the evaluated models, a decision tree classifier achieved the highest accuracy of 85.86% on unseen data for the year 2024 and outperformed a strong baseline classifier. Feature importance analysis revealed that the supplier was the most influential factor in predicting a transaction's associated performance tier.

Regression modeling focused on predicting monthly sales using two approaches: forecasting based on individual transaction-level predictions, and direct forecasting from aggregated historical data. Extreme gradient boosting regressor demonstrated the best overall performance, achieving a mean absolute percentage error (MAPE) of 8.96% between the predicted and true values for 2024 using the second approach. It was also used to forecast 2025 sales, achieving a MAPE of 8.07% when compared to the company's predicted revenue for that year, indicating strong alignment with business expectations.

Despite limitations such as a restricted hyperparameter search, exclusive reliance on internal data, and initial gaps in domain knowledge, the project successfully delivered functional prototypes. These systems provide a foundation for future development and demonstrate the practical value of machine learning in improving business forecasting and decision-making.

Overall, the project met its objectives by uncovering customer behavior patterns, optimizing internal workflows through machine learning, and supporting pricing and strategic planning with interpretable, data-driven models.

Keywords: classification, clustering, customer behavior, decision tree, k-means, machine learning, pricing strategy, regression, sales forecasting, XGBoost

Acknowledgments

This project was supported by Derome and carried out in close collaboration with the company supervisor, Robert Zarins. Working with Derome provided us with the opportunity to apply our academic knowledge to a meaningful real-world problem. Through this partnership, we have gained knowledge that contributes both to our professional and personal development.

We are sincerely grateful to both Derome and the company supervisor for their time. They provided valuable insight into the organization, explained how the business can integrate with machine learning, and offered the necessary data to achieve the project goals. The company supervisor, Robert Zarins, played a crucial role in this project. He offered continuous support, domain expertise, and constructive feedback throughout our work. This collaboration greatly deepened our understanding of the industry and we are truly thankful to Robert Zarins for his trust, mentorship, and support.

We would also like to thank the Department of Computer Science and Engineering at Chalmers University of Technology for their support. In particular, we are grateful to our academic supervisor, Oana Geman, for her guidance during the writing process of the bachelor's thesis. In addition, we appreciate the university for providing excellent resources and an academic environment throughout our three years in the bachelor's program. This foundation has not only prepared us well for this project, but will also serve as a strong base for the next steps in our careers.

Kitty Ha and Che Long Tran, Gothenburg, May 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AI	Artificial Intelligence
BI	Business Intelligence
CRAAPP	Currency, Relevance, Authority, Accuracy, Purpose and Publication
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
FN	False Negative
FP	False Positive
GPU	Graphics Processing Unit
K-NN	K-Nearest Neighbor
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MSE	Mean Squared Error
SEK	Swedish Krona
TN	True Negative
TP	True Positive
USB	Universal Serial Bus
WCSS	Within-Cluster Sum of Squares
XGBoost	Extreme Gradient Boosting

Nomenclature

Below is the nomenclature of indices, parameters, and variables that have been used throughout this thesis.

Indices

i	Index for an instance
max	Index of the maximum value
min	Index of the minimum value
$norm$	Index for the normalized value

Parameters

K	Subset coefficient of K-fold cross-validation
k	Number of clusters in K-means algorithm
N	Total number of instances in the dataset

Variables

μ	The mean of the dataset
σ	The standard deviation of the dataset
F	F1-score
FN	False negative
FP	False positive
MAE	The mean absolute error
$MAPE$	The mean absolute percentage error
MSE	The mean squared error
P	Precision

R	Recall
TP	True positive
X	A feature's original value
X_{max}	A maximum value of a feature
X_{min}	A minimum value of a feature
X_{norm}	The value of the normalization
y_i	The actual value for instance i
$\hat{y}_i$	The predicted value for instance i
Z	The standardized value

Contents

List of Figures

List of Tables

1
Introduction

This chapter provides an overview of the collaborating company, outlines the project's purpose and objectives, discusses its limitations, and explains the approach to source validation.

1.1 Derome

Derome is a family-owned business that prioritizes long-term sustainability, innovation and responsibility. The company has approximately 2,500 employees and has an annual revenue of around 10 billion Swedish kronor (SEK). Derome is Sweden's largest family-owned company in the wood industry. Founded in 1946 by Karl-Eric Andersson, the company is now managed by the third generation. Derome transforms raw materials from the forest into building components for homes and renewable energy solutions [1].

This project was conducted in collaboration with Derome to develop a machine learning prototype aimed at optimizing the company's pricing strategies. To enhance operational efficiency, Derome is exploring the integration of artificial intelligence (AI) and machine learning (ML) into its business intelligence (BI) system to refine pricing strategies. The increasing availability of AI and machine learning technologies presents an opportunity for Derome to leverage these technologies to improve pricing precision, adaptability, and market responsiveness.

1.1.1 AI and Machine Learning in Industry

Businesses across different industries are increasingly adopting and implementing AI and machine learning to improve operational efficiency, optimize workflows, and increase profits. A key focus is the development of machine learning models for pricing strategies. By integrating AI and machine learning models, many companies aim to enhance competitiveness, reduce costs, and create more intelligent business systems [2].

Dynamic pricing is one of the existing pricing strategies. According to Nowak and Pawłowska-Nowak [2], this method adjusts the price of the product depending on customer behavior, competitor pricing, and market dynamics. Examples of industries applying dynamic pricing strategies are airlines, e-commerce, hotels, and transportation. In addition, it has been shown to be successful and beneficial for both the company and the customer. In the case of an e-commerce company like Amazon, where there is an enormous amount of products, dynamic pricing strate-

gies are used to maximize revenue. Amazon has implemented an automatic pricing system that updates prices every 15 minutes [3].

Various artificial intelligence and machine learning models can be implemented to support pricing strategies in companies. For dynamic pricing, algorithms such as decision trees, the k-nearest neighbor (k-NN) algorithm [2], and regression models [4] are commonly applied. A relevant study is that of Nowak and Pawłowska-Nowak [2] which evaluated decision trees for dynamic pricing and discussed their use in e-commerce, noting that they are easy to interpret and therefore understandable to managers. However, they can be prone to overfitting, especially when working with small datasets. The study itself had limitations, including a limited dataset size and a lack of variability. These findings underline the importance of carefully selecting and validating algorithms when applying them to real-world dynamic pricing problems.

1.2 Purpose and Objectives

The purpose of this project is to examine and develop machine learning model prototypes. The goal is to support the optimization of pricing strategies within the company by generating insights from internal data that inform strategic business decisions. The following key objectives will be addressed during the project:

1. Identify patterns and anomalies in customer behavior.
2. Improve internal workflows and reduce inefficiencies using machine learning techniques.
3. Evaluate model performance with a focus on pricing accuracy and strategic decision support.

1.3 Limitations

To ensure feasibility within the project time-frame, the development is limited. The project's constraints are as follows:

- Only available internal data is utilized.
- Developing a machine learning model prototype which will be an AI-driven pricing strategy evaluation. It will not cover a full-scale deployment of AI into Derome's existing systems.
- Evaluation of general AI and machine learning methods with an emphasis on business implementations.
- Assessing and recommending AI and machine learning models without extensive large data preprocessing.

1.4 Sources

To ensure the credibility of the sources utilized in this study, the CRAAPP test is applied. This model evaluates sources based on currency, relevance, authority, accuracy, purpose, and publication [5]. Using this evaluative approach, the study guarantees that all referenced materials are reliable, current, and aligned with recognized academic standards.

2
Theory

This chapter presents relevant theoretical and technical studies on machine learning.

2.1 Machine Learning

2.1.1 Supervised Learning and Unsupervised Learning

Machine learning is a field that focuses on the construction of predictive models by observing data. Machine learning techniques can be categorized based on the type of supervision: supervised and unsupervised learning [6]. In supervised learning, models are trained on datasets containing input-output pairs, with the goal of learning a mapping that can accurately predict outputs for new, unseen data. There are two primary types of predictive tasks in supervised learning: classification, where the model learns to assign inputs to discrete category labels, and regression, where the model predicts continuous numerical values [7].

In contrast, unsupervised learning deals with datasets that lack labeled outputs. The goal is to uncover hidden patterns or structures using clustering algorithms like density-based spatial clustering of applications with noise (DBSCAN) and k-means. DBSCAN is a clustering method used to differentiate between high-density and low-density clusters, while k-means aims to form tight, spherical clusters [6]. Both have their strengths and weaknesses depending on the application.

2.1.2 Dataset

To ensure that a model generalizes well to new data, it is standard practice to divide the available dataset into three subsets: training, validation, and test sets [6].

The training set is used to fit the model, allowing it to learn patterns in the data by optimizing internal parameters. The validation set serves to fine-tune the model's hyperparameters and helps monitor for overfitting. Effective hyperparameter tuning is essential for achieving good model performance, and hyperparameter optimization techniques such as grid search are commonly applied. Grid search [8] systematically explores all combinations of predefined hyperparameter values to identify the most effective configuration. However, this method can become computationally inefficient when dealing with a large number of hyperparameters. The test set is used to evaluate the model's performance on unseen data [6]. Unseen data refers to data that the model has not encountered during the training set, providing an unbiased estimate of its generalization ability.

2.1.3 Overfitting and Underfitting

Overfitting and underfitting are issues in machine learning that affect a model's ability to generalize to unseen data [9]. Overfitting occurs when a model is overly complex and captures noise in the training data. This results in excellent training performance but poor test performance because of high variance [10]. In contrast, underfitting transpires when a model is too simple to identify underlying patterns in the data, resulting in poor performance on both training and test sets due to bias.

2.1.4 Bias and Variance

Bias and variance refer to the extent to which a model's predictions or estimations are accurate and reliable. Bias describes a systematic error caused by incorrect assumptions built into the model [6], which often leads to underfitting [10]. In contrast, an unbiased model generally produces predictions that are close to the actual values.

However, even an unbiased model can still make inconsistent predictions across different datasets due to variance. Variance is an error caused by the model's sensitivity to fluctuations in the training data [10]. High variance can result in a model that performs significantly better on training data than on unseen test data (overfitting). Therefore, finding the right balance between bias and variance is crucial for building a model that generalizes well on unseen data.

2.1.5 Data Leakage

Data leakage is a common problem in machine learning. It occurs when the training, validation, and testing datasets are not properly separated, causing them to be dependent on each other. This leads to an overly optimistic estimate of model performance during training. According to Drobnjaković et al. [11], this can be especially dangerous in high-stakes applications such as risk prediction, where the consequences of relying on inaccurate models can be severe. In real-world scenarios, data leakage can become a major issue and should be carefully avoided. To mitigate these problems, datasets are split into training, validation, and test sets. An example of a dataset split can be 80% for training and 20% for validation [12]. However, the dataset can also be split into other ratios.

2.2 Machine Learning Models

2.2.1 K-means

K-means is a clustering algorithm utilized in unsupervised machine learning, where data points are grouped into k distinct clusters [6]. The value of k is a parameter that needs to be specified by the user. The algorithm allocates each data point to the cluster with the nearest centroid. The primary objective is to reduce the distance between the data points and centroids.

Inertia, also known as WCSS (Within-Cluster Sum of Squares) is the sum of squared distances between each data point and the centroid of the cluster it has been allocated to [13]. In the k-means algorithm, the objective is to minimize inertia, which measures how spread out the data points are within each cluster. By reducing inertia, the algorithm ensures that the data points are aligned closely with their respective centroids. This indicates tighter and cohesive clustering.

The elbow method is a useful technique to find the ideal number of clusters [14]. It is essential to determine the optimal number of clusters when clustering, as it can significantly influence the quality and interpretability of the resulting clusters. The name of the technique comes from the plot that looks like an elbow [13]. The elbow method uses WCSS to identify the optimal number of clusters. When the number of clusters increases, the WCSS typically decreases, forming an elbow-like shape in the plot. The optimal number of clusters is usually found at the point where this bend occurs.

2.2.2 Decision Tree

A decision tree, visualized in Figure 2.1, is a supervised machine learning algorithm used for both classification and regression tasks. It works by splitting the dataset into smaller subsets based on feature values. This creates a tree-like structure where each internal node represents a decision, and each leaf node corresponds to an outcome [6]. The goal is to split the data such that each resulting node is as pure as possible, meaning the data points in each node are similar in target value. Determining the best split can be time-consuming, especially when the dataset contains many features.

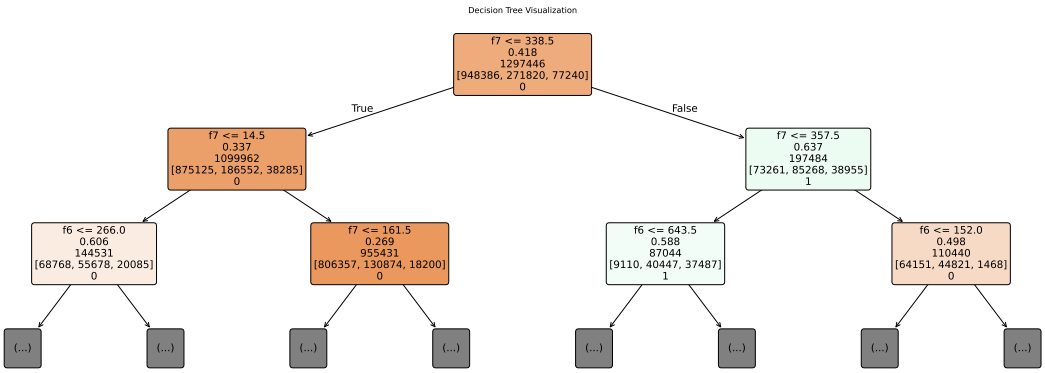


Figure 2.1: Visualization of the decision tree used to classify transactional performance tiers in the project.

2.2.3 Random Forest

The idea of combining multiple machine learning models is called an ensemble, and an ensemble made of decision trees is known as a random forest [15]. Like a standard decision tree, it can be used for both classification and regression tasks. However, this algorithm differs in how it makes decisions, by aggregating the results of multiple trees, either by taking the majority vote for classification or the average for regression.

The key concept is to use bagging, where each tree is trained on a random subset of the data. Additionally, at each split, the tree only considers a random subset of the features. This approach helps reduce the risk of overfitting, which is a common issue with single decision trees that can learn noise in the training data [15].

The advantages of using random forest are its robustness and accuracy. However, it loses in interpretability. Random forest is more complex compared to a single decision tree, which is more straightforward and easier to explain.

2.2.4 XGBoost

Extreme gradient boosting (XGBoost) is another tree-based machine learning model. This ensemble, rather than training trees independently like in random forest, builds trees sequentially to reduce the error of the previous tree [16]. This algorithm uses supervised learning. XGBoost is widely known for its strong learning capability and computational speed, making it one of the better-integrated technologies with very good performance.

The model includes a pruning function that reevaluates the trees and reduces the number of nodes that do not improve the model's performance. This helps make the model less complex and can reduce overfitting [16]. During training, the algorithm supports parallel operations, which makes it very fast. In addition, it incorporates techniques to ensure strong predictive performance while keeping the model complexity under control.

2.2.5 Logistic Regression

Logistic regression is usually used for binary classification and it is a supervised machine learning algorithm [6]. It uses a logistic function, also known as the sigmoid function (Figure 2.2) [17]. This function computes probabilities between 0 and 1, indicating the likelihood that a sample belongs to a given class. Additionally, logistic regression can be extended to multiclass classification [18]. It uses a one-vs-rest strategy, where a separate logistic regression model is trained for each class, and the class with the highest predicted probability is selected.

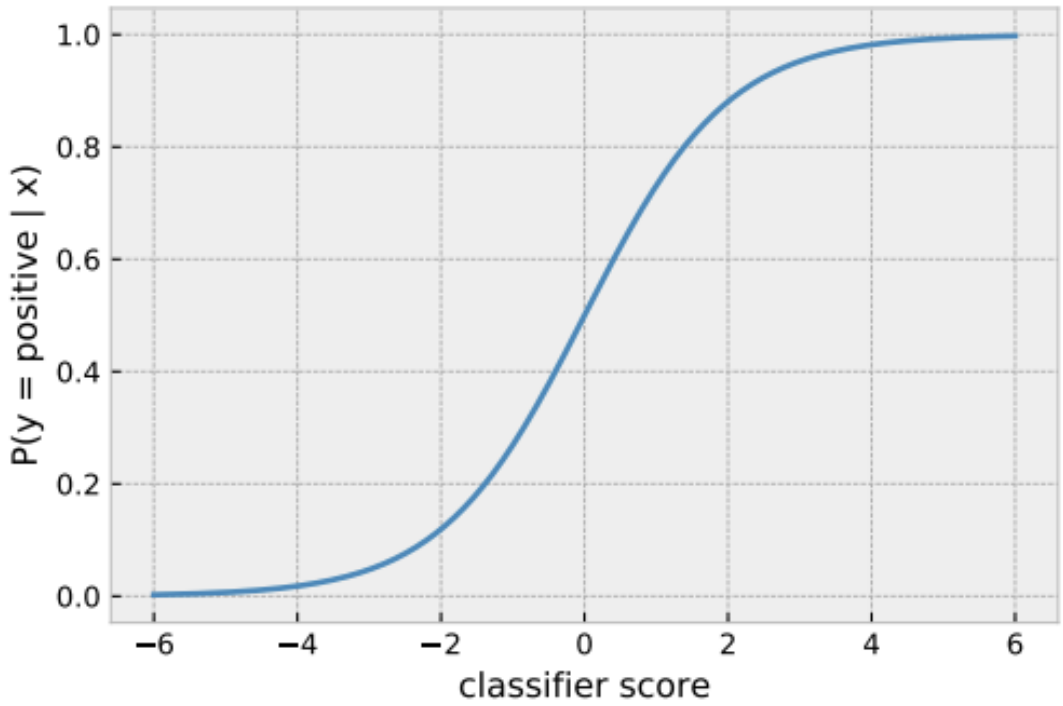


Figure 2.2: The logistic (sigmoid) function illustrating the probability output. This figure is from Johansson's PowerPoint [17].

2.2.6 Perceptron

The perceptron is a binary classifier that has gained popularity due to its simplicity [19]. It is a supervised machine learning algorithm trained on labeled data from two classes and learns a decision boundary using a linear function called the discriminant function.

During training, the algorithm typically initializes the weights and bias with small random values. For each input, it computes a weighted sum of the inputs and adds the bias. This sum is then passed through an activation function to generate a prediction. The perceptron commonly uses a step function as its activation function, which outputs 1 if the weighted sum is greater than or equal to zero, and 0 otherwise. This binary output enables the perceptron to make decisions between two classes.

The predicted output is compared to the true label, and the error is used to update the weights and bias. This process is repeated across many training examples to improve the model's accuracy. Figure 2.3 illustrates this learning process.

Similar to logistic regression, the perceptron can be extended for multiclass classification using a one-vs-rest strategy.

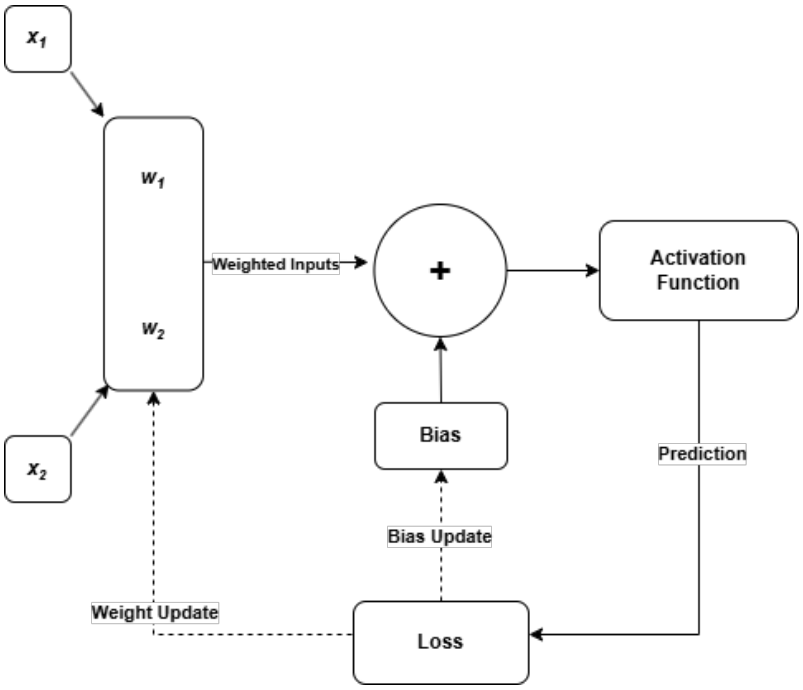


Figure 2.3: Illustration of the perceptron learning process, showing how weights and biases are updated during training.

2.3 Data Preprocessing Techniques

2.3.1 Normalization and Standardization

Normalization and standardization are important scaling techniques in data preprocessing for optimizing the model's performance by changing data values into another form [20]. Normalization scales and standardizes the data values between the range 0 and 1 [21]. By scaling the data values, it makes the data more reliable, leading to improved accuracy of the results [22]. This is particularly important for models such as k-means, which rely on distance measurements to function effectively. If the data values are not properly scaled, it can cause a feature to become more significant and dominate the other features.

The equations are provided by Sujon et al. [20], the normalization equation is given by Equation 2.1. X_{norm} is the normalized value, X is the original feature value, and X_{max} and X_{min} represent the feature's maximum and minimum values, respectively.

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{2.1}$$

In addition, standardization scales the dataset to have a mean of zero and a standard deviation of one [20]. As shown in Equation 2.2, Z represents the standardized value, X is the original feature value, μ is the dataset mean, and σ is the standard deviation.

$$Z = \frac{X - \mu}{\sigma} \tag{2.2}$$

2.3.2 Missing Data

Handling missing data is a crucial part of data preprocessing, as it can significantly impact precision and model performance [23]. Managing missing values often requires domain knowledge, since different situations call for different approaches. Common techniques include removing the data point, filling in with the average value, or using the most frequent label. While these are widely used, applying them without understanding the context can lead to misleading results. A particular method might not make sense depending on the feature or the dataset. Therefore, it is important to be thoughtful when dealing with missing data. If handled poorly, it can introduce bias and reduce the accuracy of predictions.

2.3.3 One-Hot Encoding

One-hot encoding is a technique used for handling categorical variables [24]. Each category is represented by a vector mostly consisting of zeros, with a one indicating the presence of that category in a given data point. This method is commonly used in classification tasks, as many machine learning algorithms, such as logistic regression, require numerical input. One-hot encoding ensures that categorical data can be effectively represented in a format suitable for these algorithms.

2.4 Evaluation and Optimization

When predictive systems are developed, it is rare for them to achieve perfect performance. Therefore, it is essential to measure how effectively the models perform [6]. For this reason, selecting an appropriate evaluation protocol is crucial.

2.4.1 Performance Metrics

The confusion matrix is a table used to evaluate the performance of a classification model. It shows the relationship between the actual classes and predicted classes [25]. From the matrix, one can identify key metrics such as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), which provide insight into the model's accuracy and classification behavior. Precision and recall are evaluation metrics for classification models that are calculated based on values from the confusion matrix.

Precision (P) is the measurement of how accurate the model is when it predicts a certain class. It tells how many of the predicted positives are actually correct. The Equation 2.3 defines precision [25], [26].

$$P = \frac{TP}{TP + FP} \tag{2.3}$$

While recall (R) measures how many of the positives in the class were correctly identified. [26]. Recall is defined by the Equation 2.4 [25].

$$R = \frac{TP}{TP + FN} \tag{2.4}$$

F1-score is a performance metric that is the harmonic mean of the two performance metrics, precision and recall [25], [27], given by Equation 2.5.

$$F = \frac{2 \cdot P \cdot R}{P + R} \tag{2.5}$$

The mean squared error (MSE) is typically used for evaluating a regression model. This performance metric measures the average of the squared differences between the predicted values and the actual values, shown in Equation 2.6 [25], [28]. The total number of instances in the dataset is given by N and i is the index of an instance. The variable y_i is the actual value of instance i and $\hat{y}_i$ is the predicted value for instance i.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{2.6}$$

The mean absolute error (MAE) is another metric used to evaluate regression models. It calculates the average of the absolute difference between the predicted values and the true values [25],[29]. This is expressed in Equation 2.7.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{2.7}$$

The mean absolute percentage error (MAPE) is also used for evaluating regression models. It measures the average percentage error between the predicted and actual values, providing a sense of relative error. The formula is given in Equation 2.8 [28], [29].

$$\text{MAPE} = \frac{100\%}{N} \sum_{i=1}^{N} \left| \frac{\hat{y}_i - y_i}{y_i} \right| \tag{2.8}$$

Silhouette score is a performance metric that evaluates the quality of clustering [30]. It ranges between -1 and 1 where a high score means good clustering. In addition, it can also aid in finding the ideal number of clusters k. A score close to negative one indicates that the data point is assigned to the wrong cluster. A score near one means it is placed in the correct cluster. If the score is around zero, it may fit into another cluster.

2.4.2 Cross-validation

Cross-validation is a technique for evaluating the model's performance and examining how well the model generalizes to unseen data. There are different methods when using this technique. However, the focus will be on the K-fold cross validation [31].

In K-fold cross-validation, the dataset is randomly divided into K equal-sized subsets, known as folds. Each fold takes a turn as the test set while the remaining

$K-1$ folds are used for training [32]. This process is repeated K times, and the results are averaged to estimate model performance. A common choice for K is 10 [31], but values like 3 or 5 may also be used depending on the dataset size and application.

2.4.3 Regularization

Regularization is used to mitigate overfitting in models. Overfitting can occur when a model picks up noise from the training data, causing certain feature weights to be overly emphasized in the prediction process. Regularization reduces model complexity by penalizing large weights, helping the model generalize better to unseen data.

Common types of regularization include Lasso (L1 regularization) and Ridge (L2 regularization) [33]. Both methods add a penalty term to the loss function, but they penalize the weights differently. Ridge regression shrinks the weights toward zero but does not eliminate them entirely, while Lasso can reduce some weights exactly to zero. This makes Lasso useful for feature selection, especially when it's believed that only a subset of features is relevant.

Despite their differences, both techniques aim to improve model accuracy and generalization [34].

3
Methods

This chapter presents the methodological framework applied in this study.

3.1 Dataset Description

The dataset was provided by the company supervisor. It originally contained raw data from all customer transactions and interactions from January 2022 to April 2025, sourced from the company's internal data systems. The dataset consisted of approximately 3 million data points and 46 features, capturing various aspects of each transaction, such as who made the transaction, location, price, earnings, whether the price was adjusted, and more.

These features varied in type, including both categorical and numerical variables, and some were not available across all transaction types. As a result, when extracting and combining the data into a single Excel file, each data point contained at least one missing feature value.

To protect both the company's interests and the confidentiality of the research, no actual values or figures that could be considered sensitive or proprietary have been disclosed. For instance, Figure 3.1 shows only the historical sales trend from January 2022 to April 2025, while omitting the specific sales values. Rather than focusing on exact numbers, the primary aim of this project is to demonstrate the value and potential of applying machine learning techniques to real-world company data.

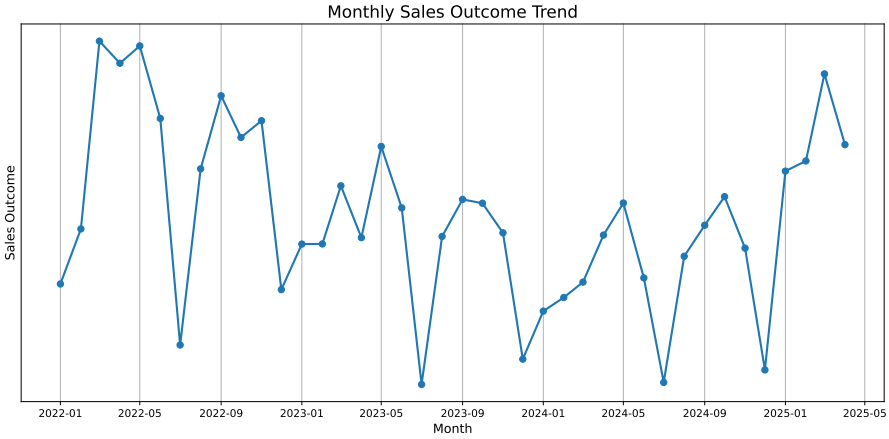


Figure 3.1: Monthly sales outcome trend (January 2022–April 2025), revealing recurring seasonal patterns and shifts in sales intensity over time.

3.2 Data Preprocessing

Since the raw data was large and diverse, it was difficult to work with, and due to the lack of domain knowledge and limited time, the company supervisor helped with filtering the data and selecting the features deemed useful for the models that were going to be developed. This created a subset of the original data that did not contain missing values because the data now consisted of the same transaction type. The new dataset, which was used and worked with, had a little more than 2 million rows and 15 features.

3.2.1 Initial Cleaning and Feature Selection

However, even after this initial filtering, further preprocessing was necessary. One step involved handling the date field, which was stored as a string and had lost its temporal meaning, despite providing useful context for the models and predictions aimed to be built. To ensure proper handling of the data, two separate preprocessing pipelines were created based on the model type. For tree-based models, string features were encoded as categorical variables, relying on the trees' inherent ability to split based on information gain, which made standardization unnecessary. This approach also reduced training time compared to one-hot encoding, which would have significantly increased the number of feature columns and computational cost. For other models, where distances between data points are important, a separate pipeline was used where all string features were one-hot encoded and numerical features were normalized. By structuring the preprocessing this way, it ensured accurate modeling while minimizing unnecessary computation.

3.3 Model Selection

For model selection, it is important to note that the company supervisor suggested and requested that specific model types be explored: a clustering model and a regression model, using internal corporate data. Since this task was to build a prototype, the purpose was more to explore and test which models would achieve the best performance and give the company better insight.

3.3.1 Clustering and Performance Tier Classification

For the clustering task, the data was examined. After discussions with the company supervisor, it was decided to aggregate product groups by sales outcome and sales volume for each year. The goal was to cluster the product groups into different performance tiers.

Based on the resulting diagram, the data appeared to exhibit a relatively circular structure (see Figure 3.2). Therefore, the k-means algorithm was selected, as it performs well with circular distributions, in contrast to alternatives such as DBSCAN. To determine the optimal number of clusters, the elbow method was used.

After applying the algorithm with the selected number of clusters, the resulting cluster labels were added to the dataset, allowing each transaction to include

information about the performance tier of its associated product group.

The tier label was then treated as a new target variable, and several classification models were trained to predict it using transaction-level features. The aim was to analyze customer behavior and assess whether it was possible to identify patterns associated with specific performance tiers.

The models tested were decision tree, XGBoost classifier, random forest classifier, logistic regression, and perceptron. These models were chosen because they are conceptually easy to understand and are also commonly used for similar tasks.

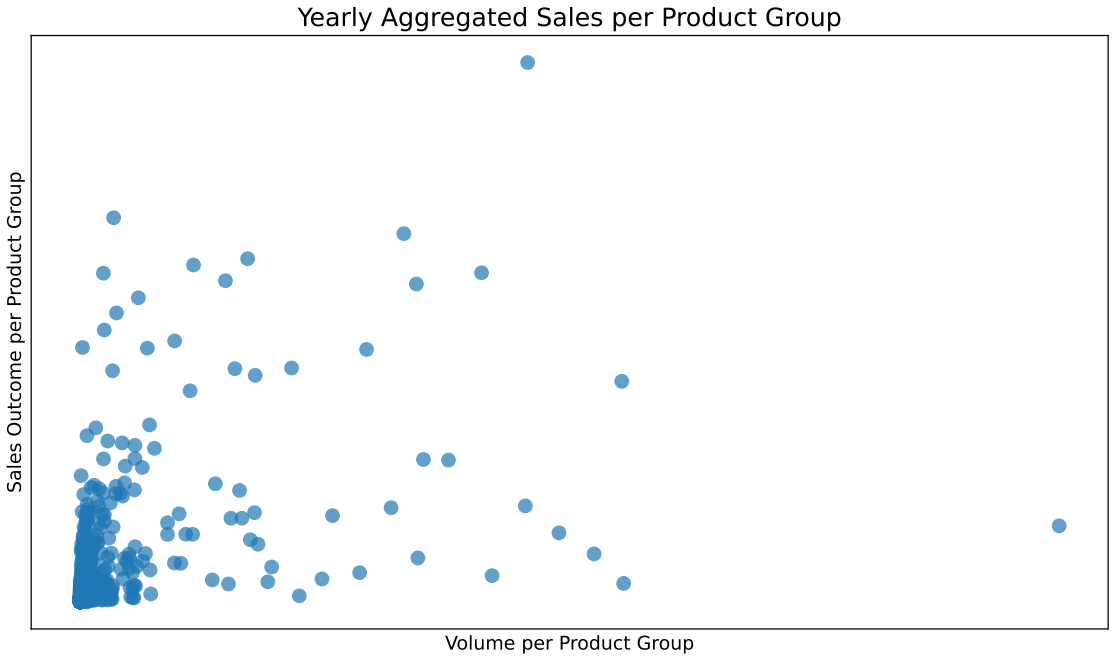


Figure 3.2: Relationship between total sales volume and total sales outcome per product group, aggregated annually. Each point represents a product group, with values summed across the year.

3.3.2 Sales Outcome Prediction

For the regression task, the goal was to develop a model to predict the sales outcome. This prediction would serve as an estimate to support the company's internal forecasting processes and help identify features that may influence sales performance.

The models used were linear regression, random forest regressor, and XGBoost regressor, which are well-known and commonly used models for this type of task. Linear regression served as a simple baseline that is easy to interpret, while random forest and XGBoost were chosen because they can handle more complex patterns and usually perform well with structured data.

3.4 Model Training

Using the newly prepared dataset of product groups aggregated by sales outcome and sales volume, k-means models were evaluated across a clustering range from 1 to 10. The resulting inertia values for each cluster count were recorded and visualized in a graph. The number of clusters that best aligned with the elbow method was selected. The corresponding performance tier for each product group was then added to the dataset for use in the classification task.

To optimize the classification models, grid search was used with 5-fold cross-validation. For tree-based models, max depth values between 2 and 20 were tested to balance underfitting and overfitting (see Table 3.1). Logistic regression's regularization strength (C) was tested using the values 0.01, 0.1, 1, and 10 to capture a wide range of penalties. For the perceptron, both L1 and L2 penalties with alpha values of 0.01, 0.001, and 0.0001 were evaluated. These ranges were selected based on common practice and early exploratory tests that helped narrow down the search space.

To prevent data leakage, features such as product group, sales volume, and sales outcome were intentionally excluded from the model inputs, as they directly influenced the tier assignment. Instead, the analysis focused on evaluating whether other features could serve as reliable indicators of performance tier, and how accurately they could predict it.

Table 3.1: Hyperparameter tuning ranges used in grid search for classification models.

Model	**Hyperparameter**	**Tested Values**
Decision tree, Random forest classifier, XGBoost classifier	Max Depth	2 to 20 (step size 1)
Logistic regression	C	0.01, 0.1, 1, 10
Perceptron	Penalty Alpha	Ridge, Lasso 0.01, 0.001, 0.0001

3.4.1 Regression Modeling

The regression modeling was split into two different approaches: transaction-level prediction and time-based forecasting. Both aimed to predict monthly sales outcomes, but used different data structures and feature designs. Figure 3.1 shows the historical sales outcome trend from January 2022 to April 2025, providing context for the forecasting task.

3.4.1.1 Approach 1: Transaction-Level Prediction

In the first approach, the sales outcome of each individual transaction was predicted and then aggregated at a monthly and yearly level. To evaluate the performance, the models were trained on data from 2022 and 2023 and then tested on 2024. Table 3.2 shows the models and the hyperparameters tested during grid search.

The selection of hyperparameters was guided by established literature. General tuning practices, including the use of grid search, were based on the guidelines described by Bischl et al. [35], which provide a foundation for optimizing machine learning models effectively. For tree-based models in particular, the tuning of parameters such as max_features and n_estimators was informed by Probst et al. [36], who emphasize their impact on the trade-off between bias and variance in random forest algorithms.

The negative mean absolute error was used as the scoring metric during the grid search to provide a quick, high-level view of model performance. The negative form is used because grid search aims to maximize the scoring function. If standard MAE (positive) were used, the model with the highest absolute error would be selected as the best, which is the opposite of the desired outcome. Therefore, using the negative version ensures that models with lower MAE scores are ranked higher, aligning with the goal of minimizing error. After training, the negative MAE values returned by the grid search were converted back to positive values to reflect the actual mean absolute error for interpretation.

The best-performing models from the grid search were then selected and retrained to predict sales outcomes for 2024. The predicted monthly sales were then compared to the actual monthly sales in 2024 using mean absolute percentage error. MAPE was chosen because, with large sales values, it can be more difficult to interpret raw error values, whereas percentage-based errors offer clearer insight.

To avoid data leakage, certain features were removed from the training set. Specifically, sales volume, sales divided by volume, and cost, as these are directly related to the target variable.

Table 3.2: Hyperparameter tuning ranges used in grid search for regression models.

Model	**Hyperparameter**	**Tested Values**
Linear regression	Fit intercept	True, False
Random forest regressor	n_estimators	100, 300
	max_depth	5, 10, None
	max_features	sqrt, log2
	min_samples_split	2, 5
	min_samples_leaf	1, 2
XGBoost regressor	n_estimators	100, 300
	learning_rate	0.05, 0.1
	max_depth	3, 5, 7, 9
	subsample	0.8, 1.0
	colsample_bytree	0.8, 1.0
	gamma	0, 0.1
	min_child_weight	1, 5

3.4.1.2 Approach 2: Time-Series Forecasting with Lag Features

The second approach focused on using sales outcomes from previous months to predict sales for a given month. The selected features included lag 1, lag 3, and

lag 12, which represent sales values from one, three, and twelve months earlier, respectively.

In addition, rolling averages over three and twelve months were incorporated to smooth short-term fluctuations and capture broader trends in the data. These features help the model detect seasonal patterns and long-term dynamics. Other features included an indicator for whether the month falls within the summer period (i.e., June, July, or August), as well as the calendar month, year, quarter, and a time index.

The time index was constructed by assigning the value zero to the first month in the dataset and incrementing it by one for each subsequent month. These features were inspired by the method presented by Makridakis et al. [37], which is based on the M5 forecasting accuracy competition. That competition focused on improving predictive accuracy using hierarchical time series data and emphasized the value of lag features, rolling statistics, and calendar-based variables in sales forecasting models.

It is worth noting that some features were unavailable for the earliest months of 2022. For example, lag 12 and the twelve-month rolling average could not be computed due to insufficient historical data. In such cases, the missing values were imputed using the mean of the corresponding feature from the rest of the dataset.

This method was applied because random forest regressor and linear regression models cannot operate on missing values. It was considered more valuable to retain these early data points by imputing reasonable estimates, rather than discarding them and further reducing the amount of historical data available for training.

The training setup for this approach was the same as in the first. Models were trained on data from 2022 and 2023 and used to predict sales for 2024. Grid search with 3-fold cross-validation was applied to evaluate different hyperparameter combinations. The negative mean absolute error was used as the scoring metric during training, and the resulting values were converted back to standard MAE for final evaluation.

Once the best hyperparameters were identified, the models were retrained on all available data before 2024 and then used to forecast monthly sales for 2024. As in the first approach, mean absolute percentage error was used to evaluate the model's performance. Mean squared error was deliberately excluded in both approaches, as sales values were already large and even MAE provided limited interpretability, serving more as a model comparison tool. Squaring the errors would have further amplified this issue, making interpretation more difficult.

An additional advantage of this approach was its ability to forecast the entire year of 2025. Since transaction data was available up to April 2025, the best-performing model was retrained using data from January 2022 through April 2025. It then predicted sales for May, which was fed back into the model to forecast June. This process was repeated sequentially, month by month, until predictions were made for the entire year.

3.5 Testing and Validation

3.5.1 Clustering Validation

The silhouette score was used to test how well the clusters identified using the elbow method performed. This was done to assess the quality and separation of the clusters. Class imbalance in the new cluster labels was considered likely, as there appeared to be a significant difference in distribution. Figure 3.2 highlights this distribution, showing higher density in the lower-left corner, where both sales volume and sales outcome are low. This pattern reflects underlying variation in the data, which the clustering algorithm should ideally be able to detect and represent.

3.5.2 Classification Model Evaluation

A baseline model was trained for the classification task. This dummy classifier predicted the majority class for all observations. It served as both a point of comparison for the final models and a sanity check. If a model cannot outperform a classifier that always predicts the majority class, it is not suitable for practical use.

After evaluating the classification models using cross-validation on the training data for predicting transaction tiers, the best-performing model was selected along with its optimal hyperparameters. Rather than applying cross-validation again, the selected model was retrained on the full training set from 2022 and 2023 and then used to predict the cluster labels for the 2024 data.

To validate these predictions, precision, recall, and F1-score were calculated to assess the model's effectiveness on unseen data. Additionally, its performance was compared to the baseline dummy classifier to determine whether the improvement was both statistically and practically meaningful, with practical significance referring to whether the performance gain could lead to useful business outcomes.

Feature importance was also analyzed using the built-in metrics provided by each model. This allowed identification of the features with the greatest influence on predicting the transaction tier. Since different models calculate importance in different ways, such as frequency of use in splits or impact on predictive performance, the values were normalized to allow for better interpretability and comparison across features.

3.5.3 Regression Model Evaluation

For the regression models, according to the company supervisor, an error margin of about 10% was considered reasonable for evaluation purposes. Although this threshold was somewhat arbitrary, it served as a reference point for assessing model performance when forecasting sales and comparing predictions to actual values using MAPE. Feature importance scores were also examined to identify which variables had the greatest influence on the predictions. These results were reviewed together with the company supervisor to determine whether the model's outputs aligned with domain knowledge and business expectations. Each model's performance was evaluated using MAPE by comparing predicted and actual monthly sales outcomes

for 2024.

Finally, the best-performing model from the second regression approach, based on historical aggregated data, was used to forecast monthly sales for 2025. The forecasted values were then compared to the company's internal budget for 2025. This budget, which represents the company's forecasted revenue for the year, was referred to as the "budget" throughout the project and will continue to be called that for consistency. It served as a proxy for the expected sales outcome for 2025.

3.6 Tools and Environment

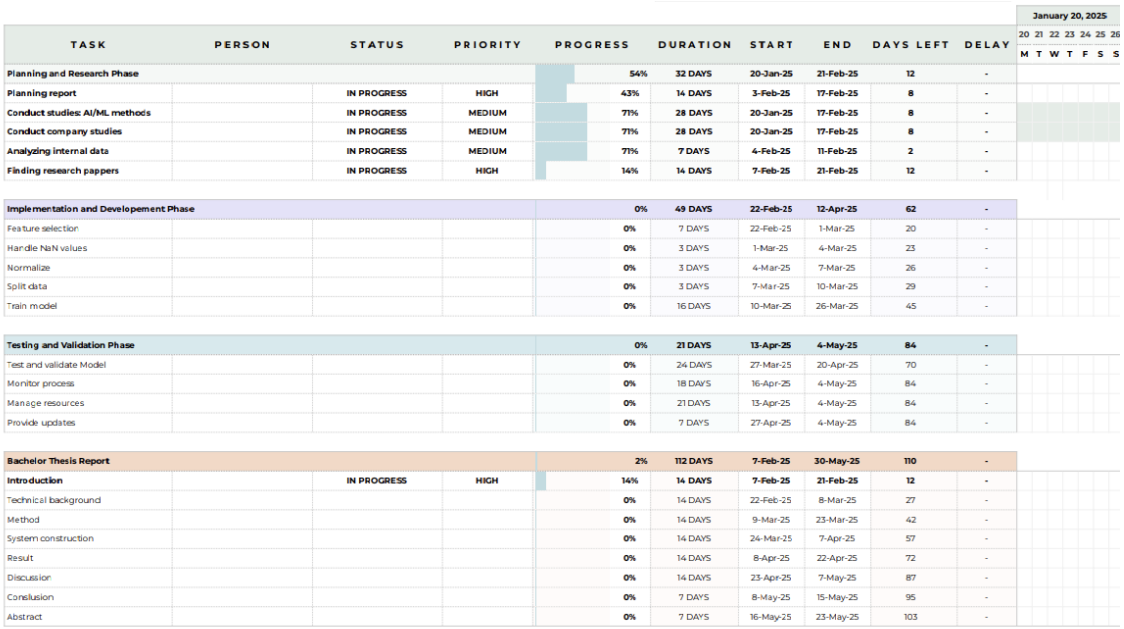
The described approach was implemented in Python version 3.12.2, using libraries such as NumPy, pandas, scikit-learn, and XGBoost. These libraries were chosen because they are standard tools in machine learning, are well documented, and make the code easier to write, read, and understand. The hyperparameters in the tables use the same names as in these libraries, for more detailed descriptions, one can refer to the official documentation. The code was executed locally on a project member's computer using Jupyter Notebook within the Anaconda version 25.1.1 environment. Direct access to the company's internal data system was not available. Instead, the data were provided via a universal serial bus (USB) stick and stored in Excel files.

3.7 Agile Methodology

Agile principles and methodology were applied from the beginning of the project to ensure that the requirements were being fulfilled and to achieve the project goals. One-week sprints were implemented, with the company supervisor serving as the product owner and the project members alternating to act as the Scrum Master each sprint.

The agile approach was deemed the most suitable for this project because of its flexibility and adaptability. This was an essential part of the iterative development of the machine learning prototype. The models were continuously refined and improved each week under the guidance of the product owner. Although initial requirements and goals were defined from the start, the project's incorporation of unfamiliar areas such as machine learning had not been previously applied within the company making it challenging to establish a detailed long-term schedule.

As a result, weekly sprints were considered more appropriate. Throughout the development process, weekly meetings were held with the product owner where tasks were given. To organize and manage the workflow, a Gantt chart, see Figure 3.3, was utilized to provide an overview of the project timeline and ensure progress tracking.



TASK	PERSON	STATUS	PRIORITY	PROGRESS	DURATION	START	END	DAYS LEFT	DELAY
Planning and Research Phase				**54%**	**32 DAYS**	**20-Jan-25**	**21-Feb-25**	**12**	-
Planning report		**IN PROGRESS**	**HIGH**	**43%**	**14 DAYS**	**3-Feb-25**	**17-Feb-25**	**8**	-
Conduct studies: AI/ML methods		**IN PROGRESS**	**MEDIUM**	**71%**	**28 DAYS**	**20-Jan-25**	**17-Feb-25**	**8**	-
Conduct company studies		**IN PROGRESS**	**MEDIUM**	**71%**	**28 DAYS**	**20-Jan-25**	**17-Feb-25**	**8**	-
Analyzing internal data		**IN PROGRESS**	**MEDIUM**	**71%**	**7 DAYS**	**4-Feb-25**	**11-Feb-25**	**2**	-
Finding research pappers		**IN PROGRESS**	**HIGH**	**14%**	**14 DAYS**	**7-Feb-25**	**21-Feb-25**	**12**	-
Implementation and Developement Phase				**0%**	**49 DAYS**	**22-Feb-25**	**12-Apr-25**	**62**	-
Feature selection				**0%**	7 DAYS	22-Feb-25	1-Mar-25	20	-
Handle NaN values				**0%**	3 DAYS	1-Mar-25	4-Mar-25	23	-
Normalize				**0%**	3 DAYS	4-Mar-25	7-Mar-25	26	-
Split data				**0%**	3 DAYS	7-Mar-25	10-Mar-25	29	-
Train model				**0%**	16 DAYS	10-Mar-25	26-Mar-25	45	-
Testing and Validation Phase				**0%**	**21 DAYS**	**13-Apr-25**	**4-May-25**	**84**	-
Test and validate Model				**0%**	24 DAYS	27-Mar-25	20-Apr-25	70	-
Monitor process				**0%**	18 DAYS	16-Apr-25	4-May-25	84	-
Manage resources				**0%**	21 DAYS	13-Apr-25	4-May-25	84	-
Provide updates				**0%**	7 DAYS	27-Apr-25	4-May-25	84	-
Bachelor Thesis Report				**2%**	**112 DAYS**	**7-Feb-25**	**30-May-25**	**110**	-
Introduction		**IN PROGRESS**	**HIGH**	**14%**	**14 DAYS**	**7-Feb-25**	**21-Feb-25**	**12**	-
Technical background				**0%**	14 DAYS	22-Feb-25	8-Mar-25	27	-
Method				**0%**	14 DAYS	9-Mar-25	23-Mar-25	42	-
System construction				**0%**	14 DAYS	24-Mar-25	7-Apr-25	57	-
Result				**0%**	14 DAYS	8-Apr-25	22-Apr-25	72	-
Discussion				**0%**	14 DAYS	23-Apr-25	7-May-25	87	-
Conslusion				**0%**	7 DAYS	8-May-25	15-May-25	95	-
Abstract				**0%**	7 DAYS	16-May-25	23-May-25	103	-

Figure 3.3: This is the project's Gantt chart at the project start. It shows the timeline and progress of the bachelor project's development. Each task is tracked with details such as status, priority, progress, duration and deadlines.

4

Results

This chapter presents the results of the project.

4.1 Clustering Model

4.1.1 Elbow Method and Cluster Selection

To perform the k-means algorithm and determine the appropriate number of clusters, the elbow method was used. The resulting graph is shown in Figure 4.1. From the graph, a significant drop can be observed in inertia when the number of clusters are increasing from 1 to 2, indicating that the data points were grouped more tightly around their respective centroids. A further, though smaller, drop occurred from 2 to 3 clusters. After that, the reduction in inertia diminished noticeably. The graph forms a clear "elbow" at 3 clusters, which is the origin of the method's name. Based on this, we selected three as the optimal number of clusters, meaning the product groups were divided into three distinct segments. The silhouette score for this clustering was 0.89, suggesting strong separation between the groups.

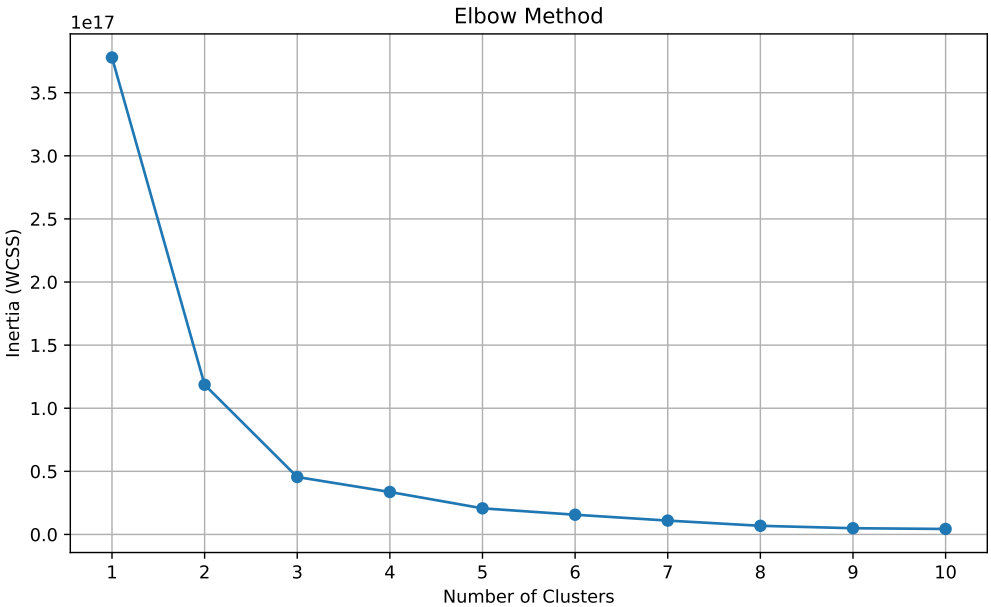


Figure 4.1: This is the Elbow Method graph used to determine the optimal number of clusters.

4.1.2 Cluster Characteristics and Class Distribution

Using this outcome, each product group was assigned to one of three tiers representing different performance levels. After running the k-means algorithm with 3 clusters and re-plotting the yearly aggregated product groups by sales outcome and volume, the clusters obtained are shown in Figure 4.2. The resulting tiers were labeled 0, 1, and 2. Tier 0 includes product groups with typically lower yearly outcome and volume, tier 2 includes those with high yearly outcome and volume, and tier 1 falls in between. Additionally, it was observed that there were fewer product groups in tier 2, even though it was the most profitable tier. The distribution of the classes is shown in Table 4.1, and it reveals a significant class imbalance, with tier 0 dominating the dataset. This is important to keep in mind for classification models.

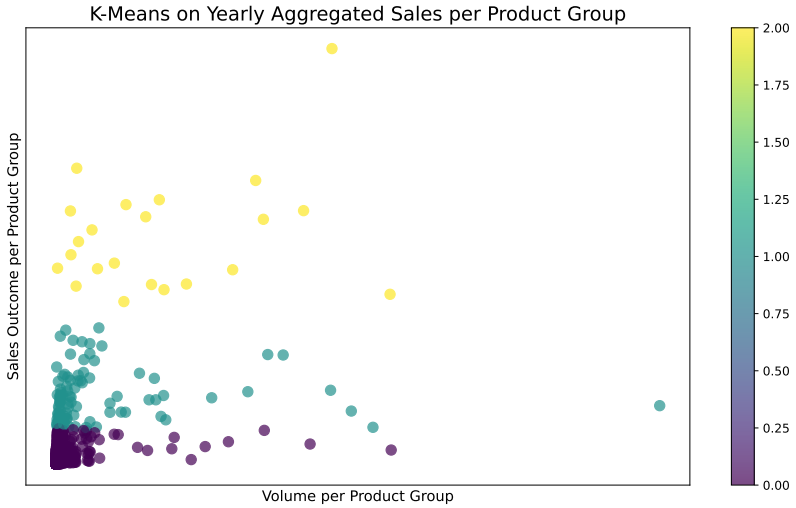


Figure 4.2: K-means clustering result on yearly aggregated product groups based on sales outcome and volume. The figure shows the separation of product groups into three tiers.

Table 4.1: Distribution of product groups across the three tiers.

Tier	Distribution
0	75%
1	20%
2	5%

4.2 Classification Model

4.2.1 Baseline and Model Comparison

To understand how well the models perform given the class imbalance, a dummy classifier was trained at the start, which achieved an accuracy of 73% as the mean of a 5-fold cross-validation. This served as a baseline to compare the other models

against. Following this, several classifiers were tested, including three tree-based models, and the results are shown in Table 4.2.

Table 4.2: Mean cross-validation accuracy for classification models.

Model	Mean Cross-Validation Accuracy
Dummy classifier	73.00%
Decision tree	84.65%
Random forest classifier	81.01%
XGBoost classifier	82.97%
Logistic regression	85.01%
Perceptron	81.75%

4.2.2 Performance Evaluation on 2024 Data

The model with the highest average cross-validation accuracy was logistic regression, with a score of 85.01% using a C parameter of 1. A close second was the decision tree, which achieved 84.65% with a max depth of 13. Since their performance was similar, both models was retrained to predict the tier for each transaction in 2024 and analyzed how they performed on unseen data.

The results for logistic regression can be seen in Table 4.3, and the results for the decision tree are shown in Table 4.4. The overall accuracy for logistic regression was 85.40%, while for the decision tree it was 85.86%. In both cases, the models outperformed the dummy classifier. However, due to the slightly higher accuracy, better F1-score, and the fact that it is easier to interpret, the decision tree was chosen for further analysis of feature importance. Figure 4.3 shows the five most important features used by the model. The most dominant factor in determining the performance tier associated with a transaction was the supplier, which was later examined with the company supervisor for further interpretation.

Table 4.3: Results for logistic regression.

Tier	Precision	Recall	F1-score
0	0.92	0.92	0.92
1	0.64	0.77	0.70
2	0.82	0.25	0.39

Table 4.4: Results for decision tree.

Tier	Precision	Recall	F1-score
0	0.93	0.92	0.92
1	0.66	0.76	0.70
2	0.67	0.36	0.47

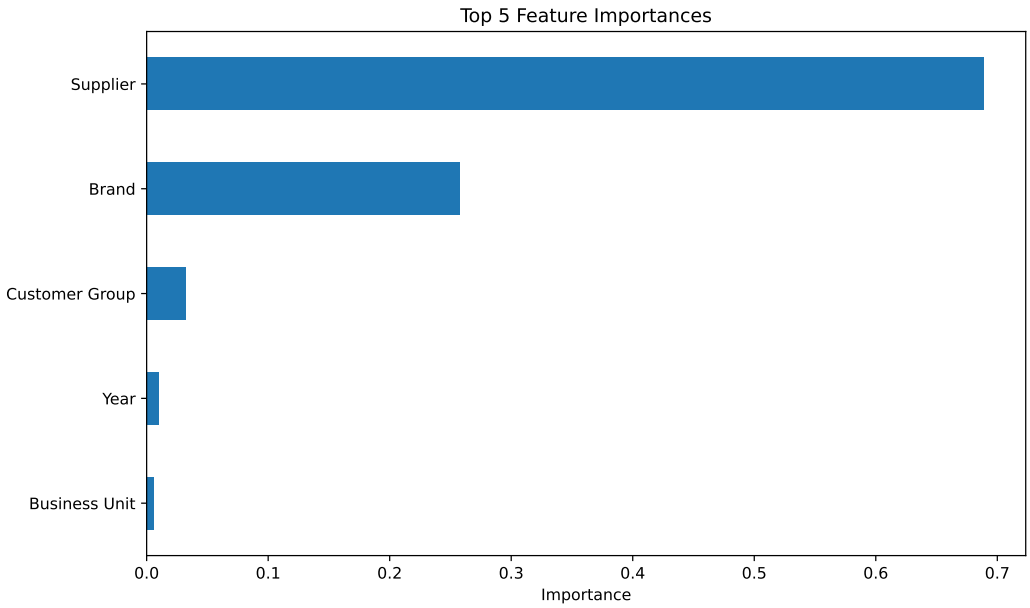


Figure 4.3: Top 5 most important features from the decision tree model used to classify transaction tiers.

4.3 Regression Model

For the regression task, two previously described approaches were evaluated to compare their results, strengths, and limitations. The first involved transaction-level predictions that were aggregated into monthly totals, while the second relied on historical aggregated data to model time-based trends directly. Performance was assessed using MAE during training for its simplicity and robustness, while MAPE was used during validation to provide percentage-based error insights that are easier to interpret in a business context.

4.3.1 Approach 1: Transaction-Level Predictions

In the first regression approach, where the sales outcome of each individual transaction was predicted at the row level, the best hyperparameters obtained from grid search are listed in Table 4.5. Based on cross-validation results, the random forest regressor performed best, with a MAE of 3,464. XGBoost regressor followed closely with a MAE of 3,721, while linear regression had the highest error at 6,330.

Table 4.5: Best hyperparameters for 2024 sales prediction using transaction-level regression model (approach 1).

Model	Hyperparameter	Best Value
Linear regression	Fit Intercept	True
Random forest regressor	n_estimators	300
	max_depth	15
	max_features	sqrt
	min_samples_split	2
	min_samples_leaf	1
XGBoost regressor	n_estimators	300
	learning_rate	0.05
	max_depth	9
	subsample	0.8
	colsample_bytree	1.0
	gamma	0
	min_child_weight	1

4.3.1.1 Monthly Aggregated Predictions

Using the optimal hyperparameters identified through grid search, the models were retrained to predict sales outcomes for 2024. As shown in Figure 4.4, XGBoost achieved the lowest MAPE among the tested models.

To gain insight into what influenced the model's predictions, feature importance was analyzed. Figure 4.5 displays the five most important features according to the XGBoost model. Product subcategory and product group showed comparable levels of importance, suggesting that both features played equally strong roles in predicting transaction-level sales outcomes.

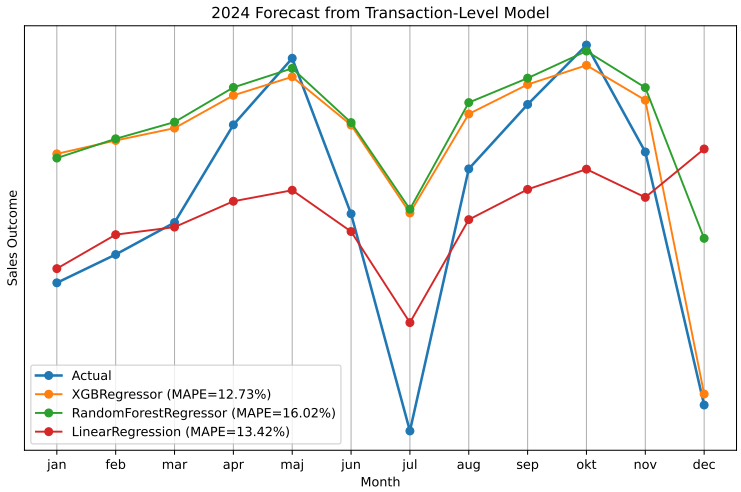


Figure 4.4: Monthly sales predictions for 2024 using the transaction-level model (approach 1), where individual transactions were predicted and then aggregated to monthly totals.

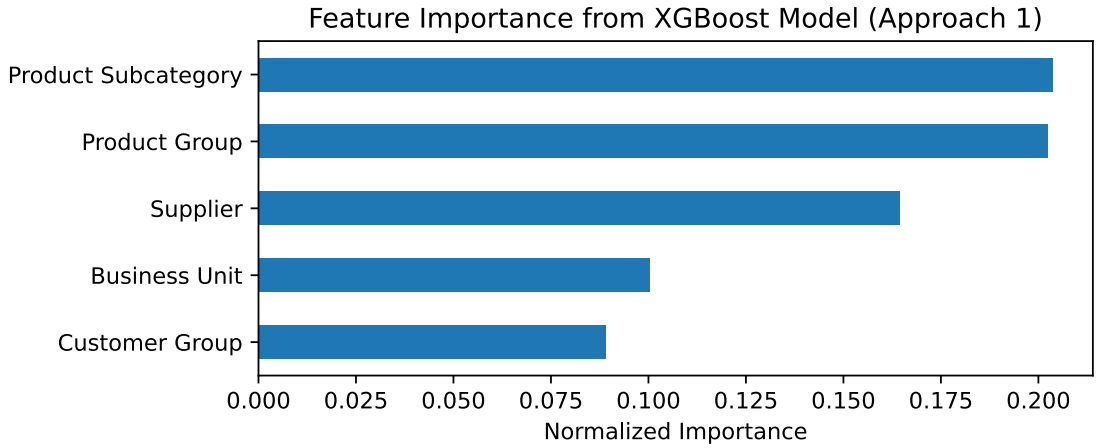


Figure 4.5: Top 5 most important features in the transaction-level model (approach 1) using XGBoost Regressor, indicating the strongest predictors of individual transaction sales outcomes.

4.3.2 Approach 2: Time-Series Forecasting

In the second regression approach, historical monthly data was used to predict sales outcomes. The same three models were applied, and the best hyperparameter values identified through grid search are listed in Table 4.6. Based on the training results, linear regression had the highest mean absolute error at 178,984,595, followed by XGBoost regressor with 44,942,119, and random forest regressor with the lowest error at 44,802,469.

Table 4.6: Best hyperparameters for 2024 sales prediction using time-series regression model with lag features (approach 2).

Model	Hyperparameter	Best Value
Linear regression	Fit Intercept	True
Random forest regressor	n_estimators	300
	max_depth	5
	max_features	sqrt
	min_samples_split	2
	min_samples_leaf	1
XGBoost regressor	n_estimators	100
	learning_rate	0.1
	max_depth	9
	subsample	0.8
	colsample_bytree	0.8
	gamma	0
	min_child_weight	1

4.3.2.1 Forecast for 2024

Using these trained models, the 2024 sales were predicted based on lagged features. The results are shown in Figure 4.6, where XGBoost once again achieved the lowest MAPE at 8.96%. Feature importance analysis in Figure 4.7 indicates that the time index and lag 12 were the two most influential features, having the highest normalized gain. Although the optimal hyperparameters differed slightly from those used in the first approach, XGBoost continued to demonstrate the most consistent performance. As a result, XGBoost regressor was selected from this approach to forecast the 2025 sales outcome.

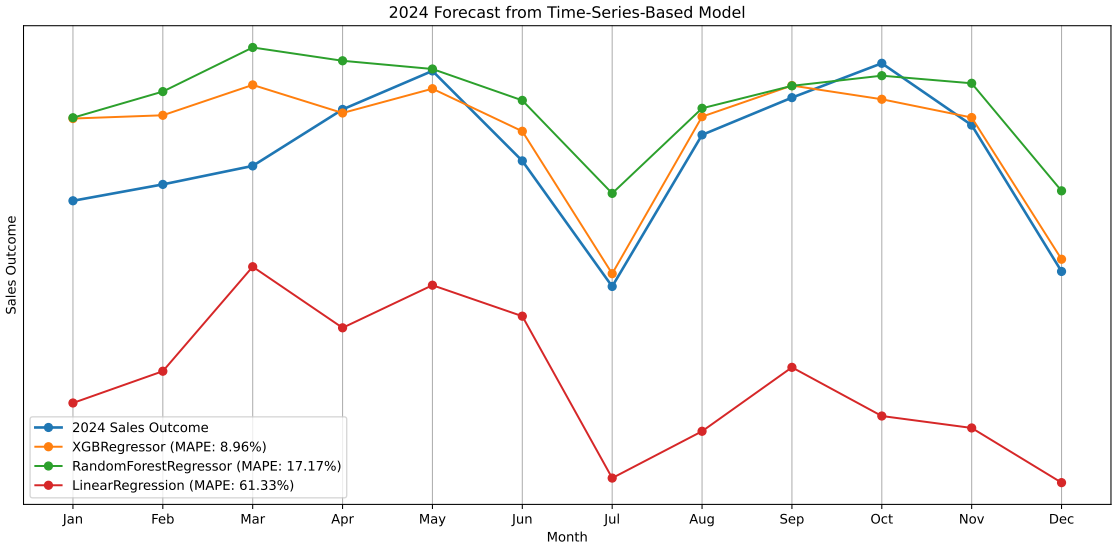


Figure 4.6: Monthly sales predictions for 2024 using the time-series model (approach 2), based on lagged historical sales features and rolling averages.

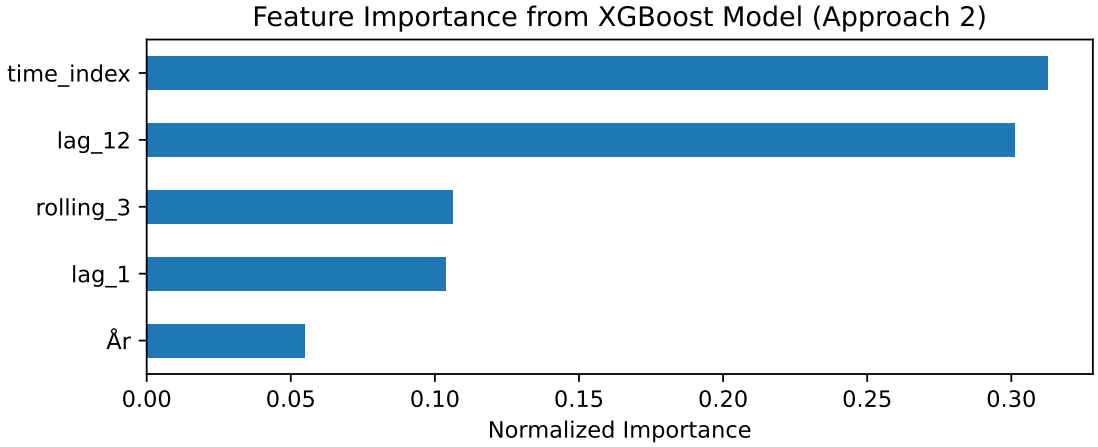


Figure 4.7: Feature importance for the XGBoost Regressor model in approach 2, based on normalized gain.

4.3.2.2 Forecast for 2025 and Budget Comparison

To generate the 2025 forecast, transaction data available up to April 2025 was used for training. From May onward, the model predicted each month's outcome sequentially, using the previous prediction as input for the next. The results are shown in Figure 4.8. On average, the model's forecast deviated from the company's internal 2025 budget by approximately 8.07%.

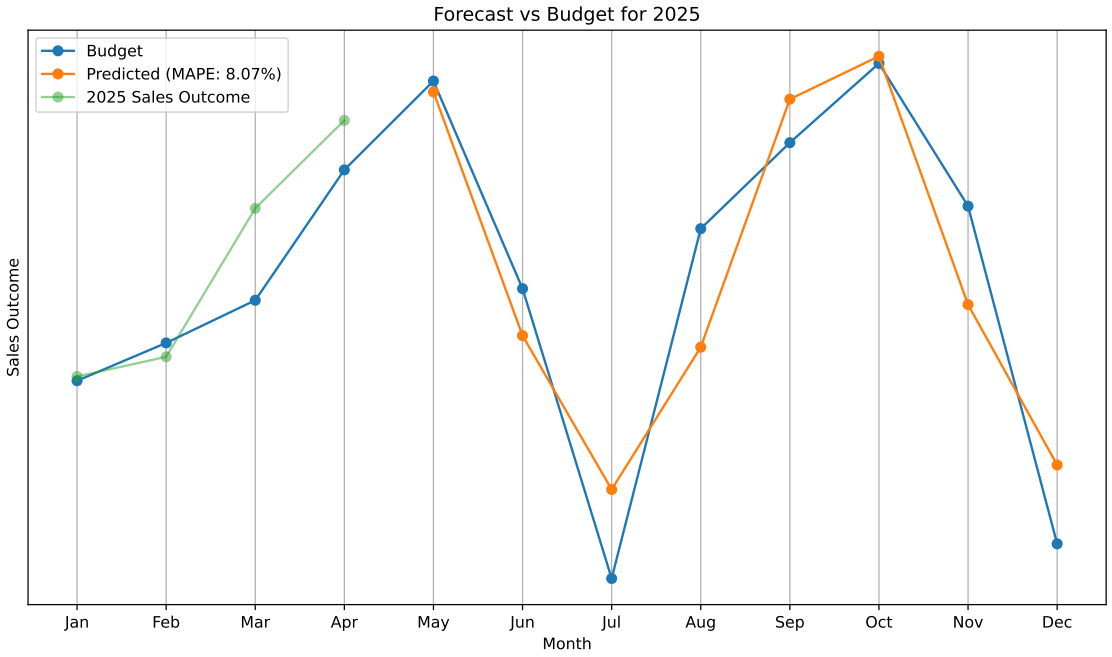


Figure 4.8: Sales forecast for 2025 using the XGBoost model from the second regression approach. The figure includes the actual sales values up to April 2025, which were used to initiate the forecast. The model was trained on data prior to May 2025, and each remaining month of 2025 was predicted sequentially.

5
Discussion

This chapter discusses the implications and interpretations of the project's results. As well as areas of improvement, future development, and ethical and sustainable considerations.

5.1 Model Selection and Design Rationale

When interpreting the results from the classification models, logistic regression and decision tree had the best performance. Given the purpose of this project, exploring machine learning opportunities for a company, there were two major factors that had to be considered during model selection.

First, the models needed to be effective for the task. That includes how well they perform, whether they offer useful insights, and if there are any risks such as bias or data leakage. These are the more technical concerns around model performance. Secondly, it was important to consider how implementable these models would be for the company. Interpretability was a key part of that; staff, administrators, and management team should be able to understand how the models work and the value they provide. This is important, because if the benefits of machine learning are easy to grasp, it increases the chances that the company will be willing to invest in future development.

Since this is a bachelor's thesis the time was limited, part of the goal was to showcase to the company what is possible with their data. This is one of the reasons why there was a strong focus on models that are easier to interpret. That trade-off between performance and transparency was considered throughout the selection process.

5.2 Clustering and the ABC Classification

For the clustering task, k-means was selected over alternatives such as DBSCAN, primarily due to the inherent structure of the data. The data points exhibited roughly circular cluster shapes (see Figure 4.2), which aligns well with k-means assumptions. Additionally, k-means allows for specifying the number of clusters explicitly, offering better control. Using the elbow method, it was determined that the optimal number of clusters was three, which was anticipated.

The company supervisor introduced the ABC classification, it is an internal system which the company uses. This system groups product groups into tiers based on sales and volume and is inspired by Lehrskov-Schmidt [38]. It divides

product groups into categories like good, better, and best, a concept the company was already familiar with.

The result of selecting three clusters using the elbow method not only made sense statistically, with a silhouette score of 0.89, but also aligned well with business logic. Not only was the ABC classification already part of the company's internal practices, but a similar tier structure could be derived directly from the data using machine learning.

5.3 Classification Results and Business Insights

After assigning each transaction a performance tier based on its associated product group, the classification task proceeded. Logistic regression and decision tree models stood out with the highest accuracy scores. Although both models outperformed the dummy classifier, it is worth noting that the dummy classifier still had a relatively high accuracy, as it always predicted the most common class. This highlights the significant class imbalance in the data, which was deliberately left uncorrected. The project members were aware of the class imbalance, but decided to preserve it because it reflects the real-world distribution of transaction tiers. This way, the models would lead to conclusions that were more realistic and aligned with how the business actually operates.

Accuracy alone was not sufficient for evaluating model performance because of the class imbalance. Other metrics was considered such as precision, recall, and F1-score. From Table 4.3 and Table 4.4, logistic regression had a higher precision when predicting tier 2, but its lower recall resulted in a weaker overall F1-score. The decision tree, on the other hand, achieved a better balance between precision and recall. Combined with its inherently interpretable structure, this made decision tree a more suitable choice. This decision supports the two priorities outlined earlier: strong technical performance and ease of interpretation for the company.

5.3.1 Understanding Key Drivers of Transaction Tiers

In addition to performance metrics, feature importance was examined. The results, shown in Figure 4.3, indicate that several features contributed significantly to predicting the performance tier for each transaction, including customer group and brand. However, the most dominant feature was the supplier. Although this was not entirely surprising to the company, it was notable that this insight emerged directly from the data rather than intuition or prior experience. This can be seen as a positive confirmation of supplier influence, now grounded in data.

With this knowledge, the company could explore further analyses around suppliers, such as identifying top-performing suppliers, understanding when they are most profitable, and detecting potential seasonal patterns. This discovery not only supports the current business intuition but also highlights another valuable use case for machine learning: enhancing business and pricing strategies through data-driven insights.

5.4 Regression Models Results and Trade-offs

As for the regression models, linear regression was included as a simple baseline. Then two tree-based models were chosen, random forest regressor and XGBoost regressor, to keep the model architecture straightforward while still being able to capture more complex relationships in the data. This resulted in a balance between interpretability and performance, especially when comparing the two forecasting approaches.

5.4.1 Limitations of Row Level Prediction

Both regression approaches come with its limitations. The first approach, predicted sales at the row level and then aggregated the predicted values into monthly totals, is more sensitive to random noise. This includes cases such as manual price adjustments or irregular transactions. It also assumes that individual transaction predictions can be accurately summed to reflect monthly outcomes, which may not always hold true. Additionally, the method lacks temporal context and may lead to overfitting based on specific transaction-level details, reducing the model's ability to generalize effectively.

The most significant limitation of this approach is that it requires access to transaction-level data for the month being predicted. This makes it unsuitable for forecasting future months, as it can only estimate sales outcomes for already recorded periods. While summing the predicted values of individual transactions did yield a MAPE of 12.73%, which is close to the company supervisor suggested 10% threshold, the approach remains constrained in its applicability.

This approach still provides useful insights. Feature importance at the transaction level highlights the factors that play a major role in influencing individual sales. In the results, the importance scores were more evenly distributed among the top five features, suggesting that several different variables played a meaningful role in predicting transaction value. These insights may be useful for developing pricing strategies, identifying key business drivers, or targeting customers more effectively, even if the model itself is not suitable for forward-looking forecasts.

5.4.2 Time-Based Forecasting and Business Implications

In the second approach, historical monthly sales data was used to predict future sales outcomes, the best-performing model for forecasting 2024 was XGBoost, achieving a MAPE of 8.96%. This performance clearly surpassed the first approach and was lower than the 10% deviation threshold suggested by the company supervisor.

One of the main strengths of this method is its ability to forecast future periods. For example, Figure 4.8 shows the model's predictions for 2025. It was observed that the predicted sales closely matched the company's internal budget, reflecting their sales expectations. On average, the 2025 forecast deviated by about 8.07%, which is considered relatively close.

This difference highlights one of the trade-offs of relying only on historical data. While the model captures trends from previous years effectively, it cannot account

for external factors such as political changes, economic shifts, or other unforeseen events. As a result, the forecast may be either too optimistic or too conservative. This limitation is further reflected in the model's most important features, lag 12 and time index, which suggests that it heavily relies on past sales outcomes and assumes that a consistent trend will continue over time.

Despite these limitations, the model still offers value. It can serve as a way to verify internal forecasts or support data-driven decision-making during planning. However, such predictions should always be interpreted with caution and used alongside expert judgment and business context.

5.5 Enhancing Progress with Domain Knowledge

One of the most important skills for successfully working with, or continuing the development of machine learning in a business setting, is domain knowledge. A key reason the project progressed smoothly was because of the agile methodology followed. That included weekly meetings and having the company supervisor act as the product owner. This setup allowed for regular communication and kept the project aligned with business needs.

More importantly, the company supervisor also provided valuable business insights that significantly improved the workflow. Guidance was provided on the important aspects of the business and how to interpret the data from a practical perspective. In addition, literature on similar tasks, such as sales forecasting, helped with identifying what features were typically important and how others approached similar problems.

Having both domain knowledge and technical machine learning skills is critical when working on applied data science tasks. This became very clear when the dataset was first received. The raw data contained every type of transaction, and at first, it was difficult to perform any meaningful analysis. There was no consistent structure, and every row had at least one missing value.

Through an iterative process and close communication with the company supervisor, the data was gradually filtered to focus on the parts that were most relevant. This back-and-forth collaboration was essential, and it was not a straightforward task. Eventually, a portion of the dataset was identified that was both clean and meaningful, providing data suitable for training the models. This does not mean that the other parts of the dataset were useless, but rather that they required more time and domain-specific understanding to work with effectively.

5.6 Areas of Improvement

As this project is a prototype, there are many areas that could be improved in future development. Some of these have already been mentioned, such as the potential to enhance the second forecasting approach by including external data sources, or applying better domain understanding to more effectively utilize the dataset. In this section, a few key improvements will be emphasized.

Firstly, during hyperparameter tuning, grid search was used with a limited set of

values. While this may have been sufficient for the purposes, it may not have yielded the absolute best configurations. One reason for this is related to hardware limitations. Since high-performance machines were not used for this project, retraining and testing models on millions of rows was time-consuming and inefficient, especially given the project time scope. As a result, trade-offs between computational cost and optimization depth were made.

Secondly, model selection could also be an area for improvement. Most of the models used were relatively basic, with the exception of random forest and XGBoost. Interpretability was prioritized because one of the goals was to show the company the benefits of machine learning in a way that was easy to understand and act upon. This likely came with a slight performance trade-off, as more complex models may have produced better results. However, these models are usually harder to interpret and require more extensive tuning, which would have extended the project time-line. While it was a limitation, it was also a practical choice given the scope and context of the project.

Lastly, the importance of transparency regarding the limitations of the project is recognized. Acknowledging these challenges is part of responsible development, and they should be considered for future iterations or deeper implementations. By identifying these areas early, future teams can build on this foundation with improved tools, models, and strategies.

5.7 Opportunities for Further Development

5.7.1 Automation and Data Integration

There are many opportunities for further development and improvements that could provide even greater business value, especially in areas like pricing strategies and sales forecasting. In this project, the data used was manually extracted from the company's internal database and delivered as Excel files. While this approach was sufficient for a prototype, a more scalable and efficient solution would involve building integrated data pipelines that connect directly to the company's data sources.

Such a system would enable access to a broader dataset. In this project, data from 2022 and onward was accessed. For some models, particularly the forecasting approach based on historical patterns, this limited time range created gaps. Early months in 2022 lacked the necessary history for computing rolling averages and lag features, and to compensate, these values was filled using feature-wise means. With a dataset covering a longer time span and greater completeness, these features could be computed more accurately, potentially improving model performance.

Implementing data automation would not only allow for more accurate inputs but would also reduce manual overhead. It would streamline the process by removing the need for manual filtering, extraction, and cleaning. Additionally, it would help keep the data current and reduce the risk of working with outdated or inconsistent files. In the early stages of this project, a considerable amount of time was spent resolving data mismatches, and a connected pipeline would have significantly reduced that burden.

5.7.2 Exploring AI-Driven Decision Support

Beyond automation, the company could also explore the use of AI agents that have a solid understanding of the environment based on the processed data. These agents could assist in making rational business decisions informed by historical patterns. There could even be multiple specialized agents focusing on different areas, such as forecasting, inventory, or pricing. These agents may operate independently or rely on each other's predictions, depending on how the system is designed.

5.7.3 Advanced Models and Long-Term Potential

Over time, with improved infrastructure and automation, there may be less need for the administrative team to directly understand the underlying machine learning technologies. The focus could shift more toward model performance and decision support. This would also open the door to experimenting with more complex models, such as neural networks, which are capable of capturing patterns that are difficult for humans to detect. Using such models would require more computational resources, more tuning effort, and longer development time, but the potential performance gains may be worth exploring in the long term.

5.8 Ethical and Sustainable Considerations in Machine Learning

This project focused on developing machine learning prototypes to support business value. However, it is equally important to consider the broader ethical and sustainability implications of such technologies.

As machine learning systems become more integrated into business operations, ensuring data privacy and accountability is critical. While this report does not explore data ethics in depth, care was taken to avoid disclosing sensitive business information. As the system develops, stronger emphasis on data governance, access controls, and compliance with relevant regulations will become increasingly important. Furthermore, if future developments incorporate sensitive customer information, users should be informed to ensure transparency and maintain trust.

Machine learning tools, especially those influencing pricing or customer outcomes, should assist rather than replace human judgment. While models are capable of detecting patterns and suggesting actions, final decisions should remain under human oversight to prevent unintended consequences. This is particularly important in domains that affect customer experience or market dynamics. Ensuring accountability requires clear documentation of how models are trained, how decisions are made, and who is responsible for evaluating their outcomes.

From an environmental perspective, machine learning can involve substantial computational costs. However, the models used in this project were relatively lightweight. These included decision tree, logistic regression, and XGBoost, with training durations limited to a few hours. To further limit computational demand, the hyperparameter search ranges were intentionally kept narrow. As a result, the environmental impact of this work was minimal.

Future development involving large-scale models, such as deep neural networks and natural language processing systems, can be highly resource-intensive and energy-consuming during both training and maintenance [39]. Promoting sustainability involves being mindful of algorithm efficiency, limiting unnecessary complexity, and balancing performance with energy use.

Moreover, while modern graphics processing units (GPU) offer improved performance for model training [40], their production relies on rare materials. Countries like China currently dominate the processing of these resources, accounting for about 90% of global refining and 60% of extraction [41]. Their extraction, refinement, and transportation contribute significantly to global carbon emissions, underscoring the need to factor supply chain impacts into sustainability considerations.

While AI and machine learning offer substantial value, environmental and ethical concerns must not be overlooked. Addressing these factors is essential not only for building efficient systems but also for contributing to a more sustainable and responsible future.

6

Conclusion

The project resulted in operating prototypes that fulfilled the initial requirements outlined at the beginning of the study. The implemented models included clustering, classification, and regression algorithms, all applied to a dataset containing transactions and key transactional attributes.

For the clustering task, product groups were aggregated annually by sales outcome and sales volume, and the k-means algorithm was applied to segment them into performance tiers. Using the elbow method, the optimal number of clusters was determined to be three. In this setup, tier 2 represented the highest-performing product groups in terms of sales and volume, while tier 0 represented the lowest. The clustering results showed strong separation and high quality, as reflected by a silhouette score of 0.89.

Each transaction was then assigned the performance tier corresponding to its associated product group. This label was used in a classification task to evaluate whether models could accurately predict a transaction's associated performance tier, and to identify which features mostly influenced that prediction.

Among the classification models, the decision tree achieved the highest accuracy on the 2024 test set, reaching 85.86%. It also recorded the highest F1-scores across all three performance tiers, with a maximum depth of 13. Although it did not perform best during training, where logistic regression had slightly better results, its performance was comparable. Upon evaluation on unseen 2024 data, the decision tree showed a marginal performance advantage over logistic regression. It also significantly exceeded the baseline dummy classifier, which achieved 73% accuracy, largely due to the strong class imbalance. These results support the model's practical viability.

Ultimately, the decision tree was selected for further analysis due to its slightly stronger performance and higher interpretability. Feature importance analysis identified the supplier as the most influential factor in determining a transaction's associated performance tier. This insight offers a valuable opportunity for deeper analysis and could help inform strategic business decisions.

Lastly, for the regression models, the goal was to forecast sales outcomes, and that was done using two different approaches. The first approach aimed to predict individual transaction sales and then aggregate them monthly to generate a forecast. The second approach directly forecast monthly sales based on historical aggregated data.

In the first approach, the XGBoost regressor performed best, achieving a MAPE of 12.73% when compared to the actual 2024 values. In the second approach, XGBoost again delivered the most accurate results, with a MAPE of 8.96% for 2024.

This made it the most reliable model overall. Furthermore, the second approach enabled future forecasting, something the first method could not support due to its reliance on transaction-level data from the year being predicted. Using XGBoost from the second approach, the 2025 sales forecast achieved a MAPE of 8.07% when compared to the company's projected revenue for that year. These results demonstrate the potential of machine learning models to support the budgeting process and leverage historical data for accurate forecasting.

Feature importance analysis from the first approach indicated that several variables played key roles in the predictions, notably product group, product subcategory, supplier, and business unit. In contrast, the second approach was primarily influenced by temporal features, especially revenue from 12 months prior and the overall time trend.

Several limitations affected the study. One key constraint was that the models were trained on a limited range of hyperparameters due to computational cost and time constraints. Additionally, the project emphasized model interpretability, as the primary goal was to give the company insight into the capabilities of machine learning. However, this focus may have come at the expense of achieving peak predictive performance.

Another limitation was that the models were trained solely on Derome's internal sales data, without incorporating external data sources. Including external factors, such as market trends, economic indicators, or competitor pricing, which could have improved model accuracy, particularly since product prices are influenced by such dynamics.

A major challenge was the initial lack of domain knowledge. While the dataset was pre-filtered by the company supervisor, it took time to fully understand the organizational structure, the role of various features, and their interrelationships. A deeper domain understanding from the start could have led to more efficient analysis and better use of the data.

Despite these limitations, the project offered valuable insights into applying academic knowledge in a real-world business setting. Beyond the technical aspects, the experience provided exposure to corporate workflows and organizational decision-making. Initially, the prototype development process was challenging due to limited business context. However, through regular meetings with the company supervisor, who offered domain-specific guidance, our understanding improved. Following agile methodology and iterative sprints, the prototypes evolved steadily, demonstrating clear progress from problem exploration to solution delivery.

A key distinction observed during the project was the difference between academic problem-solving and addressing real-world challenges. Unlike university assignments, real-world data rarely yield perfect results or definitive conclusions. This reality required a shift in mindset and highlighted the importance of iterative development and approximation in practical applications.

As machine learning becomes more embedded in business operations, it is essential to maintain a critical perspective. While these models can significantly support decision-making, they should not be used unconditionally. Human oversight remains crucial, especially in areas like pricing and sales, where accountability for decisions must be clearly defined. As such, successful implementation of AI and machine

learning requires adequate training and a shared understanding that these systems are tools to support, not replace, human judgment.

The developed prototypes offer a strong foundation for future expansion. Further improvements could involve incorporating more internal data as well as external sources. Exploring more advanced models, such as neural networks, may lead to better performance, potentially enhancing decision-making and reducing administrative workloads. However, such improvements would demand increased computational resources, more extensive hyperparameter tuning, and longer development timelines. These investments could be worthwhile over time, considering the potential for improved results and broader business impact.

In summary, the project effectively met the three core objectives defined at the beginning. First, by classifying transactions into performance tiers and analyzing key influencing factors such as supplier and brand, the project provided a foundation for identifying patterns and potential anomalies in customer behavior. Second, the implementation of interpretable machine learning models for both classification and sales forecasting demonstrated how AI can support internal workflows, improve efficiency, and offer reliable decision support. Finally, model performance was evaluated using accuracy, F1-score, MAPE for predictive tasks, and silhouette score for clustering. Feature importance enabled a deeper understanding of key drivers and allowed model reasoning to be compared with real-world business logic.

To conclude, the project demonstrates that machine learning can serve as a powerful aid in both understanding business data and guiding strategic planning.

Bibliography

[1] "Om Derome." derome.se. Accessed: Feb. 26, 2025. [Online]. Available: https://www.derome.se/

[2] M. Nowak and M. Pawłowska-Nowak, "Dynamic Pricing Method in the E-Commerce Industry Using Machine Learning," in *Artificial Intelligence and Machine Learning Applications in Industrial Systems,* vol. 14, no. 24, Dec. 2024, doi: https://doi.org/10.3390/app142411668

[3] J. Liu, Y. Zhang, X. Wang, Y. Deng, and X. Wu, "Dynamic pricing on e-commerce platform with deep reinforcement learning: A field experiment," arXiv preprint, Dec. 2019. [Online]. doi: https://doi.org/10.48550/arXiv.1912.02572

[4] G. Yamuna, D. Paul Dhinakaran, C. Vijai, P. S. J. Kingsly, Raynukaazhakarsamy and S. R. Devi, "Machine Learning-Based Price Optimization for Dynamic Pricing on Online Retail," in *Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM),* Chennai, India, 2024, pp. 1-5, doi: https://doi.org/10.1109/ICONSTEM60960.2024.10568763

[5] "LibGuides: Search and evaluate information: Source evaluation." guides.lib.chalmers.se. Accessed: Mar. 15, 2025. [Online]. Available: https://guides.lib.chalmers.se/sokaochutvarderainformation/EN/sourceevaluation

[6] M. T. Almuqati, F. Sidi, S. N. M. Rum, M. Zolkepli, and I. Ishak, "Challenges in Supervised and Unsupervised Learning: A Comprehensive Overview," in *International Journal and on Advanced Science, Engineering and Information technology,* vol. 14, no. 4, 2024, doi: https://doi.org/10.18517/ijaseit.14.4.20191

[7] J. Li, "Regression and Classification in Supervised Learning," in *Proceedings of the 2nd international conference on computing and big data (ICCBD '19),* Association for Computing Machinery, New York, NY, Oct. 2019, pp. 99-104, doi: https://doi.org/10.1145/3366650.3366675

[8] J. A Ilemobayo et al., "Hyperparameter Tuning in Machine Learning: A Comprehensive Review," vol. 26, no. 6, pp. 388-395, Jun. 2024, doi: https://doi.org/10.9734/jerr/2024/v26i61188

[9] C. Aliferis and G. Simon, "Overfitting, Underfitting and General Model Overconfidence and Under-Performance Pitfalls and Best Practices in Machine Learning and AI," in *Artificial Intelligence and Machine Learning in Health Care and Medical Sciences,* Health Informatics. Springer, Cham, 2024, pp. 477-524, doi: https://doi.org/10.1007/978-3-031-39355-6_10

[10] S. S. Skiena, *The Data Science Design Manual,* Charm: Springer, 2017, doi: https://doi.org/10.1007/978-3-319-55444-0

[11] F. Drobnjaković, P. Subotić, and C. Urban, "An Abstract Interpretation-Based Data Leakage Static Analysis," in *Theoretical Aspects of Software Engineering (TASE 2024),* W. N. Chin and Z. Xu, Eds., Lecture Notes in Computer Science, 2024, vol. 14777, pp. 109-126, Springer, Cham, doi: https://doi.org/10.1007/978-3-031-64626-3_7

[12] I. Muraina, "IDEAL DATASET SPLITTING RATIOS IN MACHINE LEARNING ALGORITHMS: GENERAL CONCERNS FOR DATA SCIENTISTS AND DATA ANALYSTS," in *Proceedings of the 7th International Mardin Artuklu Scientific Research Conference,* 2022, pp. 496-504.

[13] K. Gowsic, S. Mugunthan, S. Logavaseekarapakther, A. Puviyarasu, and R. Mohammed Farook, "ENHANCED UNSUPERVISED K-MEANS CLUSTERING ALGORITHM," in *ShodhKosh Journal of Visual and Performing Arts,* vol. 5, no. 1, Jan. 2024, doi: https://doi.org/10.29121/shodhkosh.v5.i1.2024.2867

[14] E. Umargono, J. E. Suseno, and S.K. Vincensius Gunawan, "K-Means Clustering Optimization Using the Elbow Method and Early Centroid Determination Based on Mean and Median Formula," in *Proceedings of the 2nd International Seminar on Science and Technology,* vol. 474, 2020, doi: https://doi.org/10.2991/assehr.k.201010.019

[15] L. Breiman, "Random Forests," in Machine Learning, vol. 45, pp. 5-32, 2001, doi: https://doi.org/10.1023/A:1010933404324

[16] Y. Bao and H. Wen, "Research on Prediction of Anti-Fraud in Automobile Finance Based on XGBoost Machine Learning Algorithm," in *Proceedings of the International Conference on Digital Economy*, Blockchain and Artificial Intelligence, pp. 367–375, Aug. 2024, doi: https://doi.org/10.1145/3700058.3700116

[17] R. Johansson. (2025). Logistic regression [PowerPoint slides]. Available: https://www.cse.chalmers.se/~richajo/dit866/lectures/l5/l5_3.pdf

[18] "LogisticRegression." scikit-learn.org. Accessed: May 16, 2025. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[19] M. N. Murty and R. Raghava, "Perceptron," in *Support Vector Machines and*

Perceptrons, SpringerBreifs in Computer Science. Cham: Springer, 2016, pp. 27-40. doi: https://doi.org/10.1007/978-3-319-41063-0_3

[20] K. M. Sujon, R. B. Hassan, Z. T. Towshi, M. A. Othman, M. A. Samad, and K. Choi, "When to Use Standardization and Normalization: Empirical Evidence from Machine Learning Models and XAI," in *IEEE Access,* vol. 12, pp. 135300-13531, 2024, doi: https://doi.org/10.1109/ACCESS.2024.3462434

[21] V. S. Durga and T. Jeyaprakash, "Data Transformation Techniques for Academic Datasets," *International Journal of Engineering and Advanced Technology (IJEAT),* vol. 9, no. 1, Oct. 2019. ISSN: 2249 – 8958, doi: https://doi.org/10.35940/ijeat.A9711.109119

[22] V. R. Patel and R. G. Mehta, "Performance analysis of MK-means clustering algorithm with normalization approach," in Proc. 2011 *World Congress on Information and Communication Technologies,* Mumbai, India, 2011, pp. 974-979, doi: https://doi.org/10.1109/WICT.2011.6141380

[23] D. Pandya, A. Jadeja, S. Gour, S. B. Trivedi, H. H. Patel, and P. U. Jadeja, "An Analytical Perspective of Missing Values in Machine Learning," in *Emerging Trends in Expert Applications and Security (ICE-TEAS 2024).* V.S. Rathore, V. Piuri, R. Babo, V. Tiwari (eds.), Lecture Notes in Networks and Systems, vol. 1037, pp. 285–294, Singapore: Springer, 2024, doi: https://doi.org/10.1007/978-981-97-3991-2_24

[24] J. A. Samuels, "One-Hot Encoding and Two-Hot Encoding: An Introduction," preprint, 2024, doi: https://doi.org/10.13140/RG.2.2.21459.76327

[25] R. Johansson. (2025). Evaluation of Classifiers and Regressors [PowerPoint slides]. Available: https://www.cse.chalmers.se/~richajo/dit866/lectures/l6/l6_3.pdf

[26] "Precision-Recall." scikit-learn.org. Accessed: May 16, 2025. [Online]. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

[27] L. Lavazza and S. Morasca, "Comparing ϕ and the F-measure as performance metrics for software-related classifications," *Empirical Software Engineering,* vol. 27, no. 7, 2022, doi: https://doi.org/10.1007/s10664-022-10199-2

[28] "3.4. Metrics and scoring: quantifying the quality of predictions." scikit-learn.org. Accessed: May 16, 2025. [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html

[29] T. Vu, G. S. Thirunavukkarasu, M. Seyedmahmoudian, S. Mekhilef and A. Stojcevski, "Comparative Analysis of Regression Models for Household Appliance Energy Consumption Prediction using Extreme Gradient Boosting," in *2023 33rd Australasian Universities Power Engineering Conference (AUPEC)*, Bal-

larat, Australia, 2023, pp. 1-6, doi: https://doi.org/10.1109/AUPEC59354.2023.10503204

[30] K. R. Shahapure and C. Nicholas, "Cluster Quality Analysis Using Silhouette Score," in *2020 IEEE 7th international conference on data science and advanced analytics (DSAA),* Sydney, NSW, Australia, 2020, pp. 747-748, doi: https://doi.org/10.1109/DSAA49011.2020.00096

[31] B. Ghojogh and M. Crowley, "The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial," arXiv preprint arXiv:1905.12787. [Online]. doi: https://arxiv.org/abs/1905.12787

[32] G. Simon and C. Aliferis, "Evaluation," *Artificial Intelligence and Machine Learning in Health Care and Medical Sciences,* Health Informatics. Springer, Cham, 2024, pp. 415-476, doi: https://doi.org/10.1007/978-3-031-39355-6_9

[33] J. C. Obi and I. C. Jecinta, "A Review of Techniques for Regularization," *International Journal of Research in Engineering and Science*, 2023, vol. 11, no. 1, pp. 360-367.

[34] Z. Song, "Lasso and ridge regression methods and their application in GDP deflator estimation analysis," *In International Conference on Statistics, Applied Mathematics, and Computing Science (CSAMCS 2021),* SPIE, 2022, vol. 12163, pp. 834-843, doi: https://doi.org/10.1117/12.2628045

[35] B. Bischl et al., "Hyperparameter Optimization: Foundations, Algorithms, Best Practices and Open Challenges," arXiv preprint arXiv:2107.05847, 2021. [Online]. doi: https://doi.org/10.48550/arXiv.2107.05847

[36] P. Probst, M. Wright, and A.-L. Boulesteix, "Hyperparameters and Tuning Strategies for Random Forest," arXiv preprint arXiv:1804.03515, 2019. [Online]. doi: https://doi.org/10.48550/arXiv.1804.03515

[37] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, “M5 accuracy competition: Results, findings, and conclusions,” *Int. J. Forecast.*, vol. 38, no. 4, 2022. doi: https://doi.org/10.1016/j.ijforecast.2021.11.013

[38] U. Lehrskov-Schmidt, *The Pricing Roadmap: How to Design B2B SaaS Pricing Models That Your Customers Will Love,* Houndstooth Press, 2023.

[39] D. Patterson et al., "Carbon Emissions and Large Neural Network Training," arXiv preprint arXiv:2104.10350, 2021. [Online]. doi: https://doi.org/10.48550/arXiv.2104.10350

[40] G. Koszczał, J. Dobrosolski, M. Matuszek, and P. Czarnul, "Performance and Energy Aware Training of a Deep Neural Network in a Multi-GPU Environment with Power Capping," in Zeinalipour, D. et al. (Eds.): *Euro-Par 2023: Parallel Processing Workshops, Lecture Notes in Computer Science,* vol. 14352,

Springer, Cham, 2024, doi: https://doi.org/10.1007/978-3-031-48803-0_1

[41] G. Baskaran. "What China's Ban on Rare Earths Processing Technology Exports Means." csis.org. Accessed: Jun. 21, 2025. [Online]. Available: https://www.csis.org/analysis/what-chinas-ban-rare-earths-processing-technology-exports-means

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS