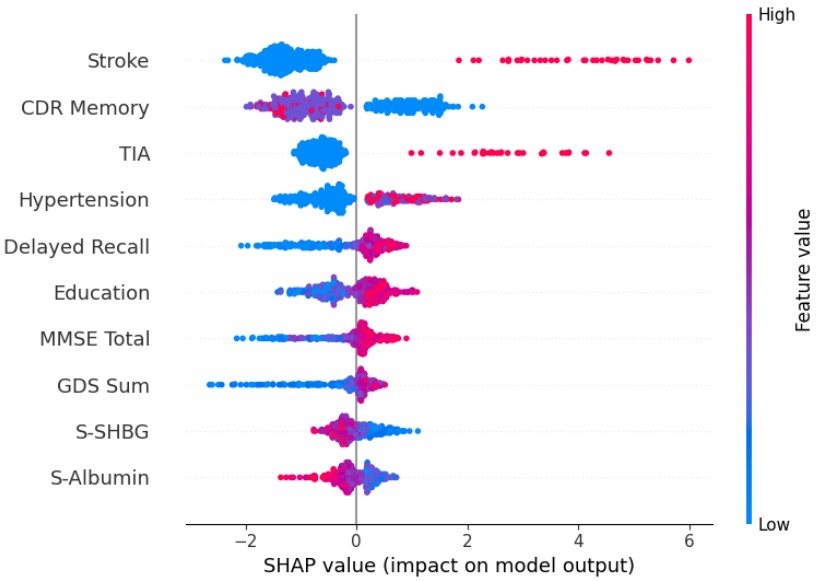
CHALMERS
UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG



Using Machine Learning to Develop Diagnostic Models for Cognitive Diseases

Advancing Neurodegenerative Diagnostics with Interpretable Machine Learning Models

Master's thesis in Computer Science and Engineering

JAKOB SVENSSON

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2025

Master's thesis 2025

Using Machine Learning to Develop Diagnostic Models for Cognitive Diseases

Advancing Neurodegenerative Diagnostics with Interpretable Machine Learning Models

JAKOB SVENSSON

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

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

Using Machine Learning to Develop Diagnostic Models for Cognitive Diseases
Advancing Neurodegenerative Diagnostics with Interpretable Machine Learning Models
JAKOB SVENSSON

Supervisor: Petronella Kettunen, GU
Examiner: Simon Olsson, CSE

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

Cover: Figure showing the distribution of SHAP values for individual instances in the top ten features with highest mean absolute SHAP values and how higher (red) or lower (blue) feature values impact the model's prediction towards SSVD (positive class) or AD (negative class).

Typeset in LaTeX
Gothenburg, Sweden 2025

Using Machine Learning to Develop Diagnostic Models for Cognitive Diseases
Advancing Neurodegenerative Diagnostics with Interpretable Machine Learning Models
JAKOB SVENSSON
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Vascular Cognitive Disease (VCD), including Subcortical Small Vessel Disease (SSVD), remains one of the most underdiagnosed cases of dementia. Given the clinical need for early, accurate, and explainable diagnosis, this study explores machine learning techniques trained on real-world clinical data leveraged from the Gothenburg Mild Cognitive Impairment (MCI) study to uncover key variables in VCD diagnosis. The methodology was structured into three key parts: data preprocessing, model training and evaluation, and model explainability. The original dataset was partitioned into three subsets to reflect distinct clinical settings: primary care, specialist care, and research data, each with varying levels of feature availability. Given the high rate of missing values, multiple imputation techniques were explored and assessed.

The model training part involved evaluating the performance of various machine learning algorithms across specific diagnostic tasks and clinical settings. These machine learning algorithms included two gradient boosting tree algorithms XGBoost and LightGBM, Support Vector Machine (SVM), Random Forest (RF), and Naive Bayes. The model with the highest average F1 score (ranging from 0 to 1) across multiple iterations was selected for final deployment and further refined through additional training. An Explainable AI (XAI) approach, named SHapley Additive exPlanations (SHAP), was applied to the final model to ensure transparency and clinical relevance and identify the most influential features contributing to classification outcomes.

The majority of the final models could classify diagnoses with high precision and recall, achieving high F1 scores. Some of the variables were previously known to be associated with the diseases. Furthermore, new variables not previously linked to the disease were identified, prompting further research. In conclusion, the machine learning pipeline built in this study has the potential to act as a classifier to distinguish VCD from AD and clinical prestages of cognitive impairment in a clinical setting. Furthermore, it can be utilized to identify key variables associated with distinguishing between these diseases.

Keywords: Cognitive diseases, machine learning, explainable ai (XAI), classification, missing data imputation.

Acknowledgements

I am sincerely grateful to my supervisor Petronella Kettunen and Emir Basic at the Memory Clinic in Gothenburg for their invaluable guidance and support throughout the course of this project.

Jakob Svensson, Gothenburg, 2025-06-17

Contents

List of Figures **xi**

List of Tables **xiii**

1 Introduction **1**
1.1 Background . . . 1
1.2 Aim . . . 3
1.3 Specification of the Issue Being Investigated . . . 3
1.4 Limitations . . . 4

2 Theory **5**
2.1 Two-Sample Kolmogorov-Smirnov Test . . . 5
2.2 Missing Data Mechanisms: MCAR, MAR, and MNAR . . . 6
2.3 Imputation Methods . . . 7
2.3.1 K-Nearest Neighbors (KNN) Imputation . . . 7
2.3.2 Multiple Imputation by Chained Equations (MICE) . . . 8
2.4 Synthetic Minority Over-sampling Technique (SMOTE) . . . 9
2.5 Machine Learning Models . . . 10
2.5.1 Random Forest (RF) . . . 10
2.5.2 Gradient Boosted Trees . . . 10
2.5.2.1 XGBoost . . . 11
2.5.2.2 LightGBM . . . 11
2.5.3 Support Vector Machine (SVM) . . . 11
2.5.4 Naive Bayes . . . 11
2.6 Explainable AI (XAI) . . . 12
2.6.1 SHapley Additive exPlanations (SHAP) . . . 12

3 Methods **15**
3.1 Pre-Processing . . . 15
3.1.1 Dataset Generation . . . 16
3.1.2 Handling Missing Values . . . 17
3.2 Model Training and Evaluation . . . 19
3.2.1 Handling Class Imbalance . . . 19
3.2.2 Machine Learning Models . . . 21
3.2.3 Evaluation Metrics . . . 21
3.3 Model Explainability and Feature Extraction . . . 22

4 Results **23**
4.1 Handling Missing Values 23
4.2 Model Evaluation 28
4.3 Feature Importance 30

5 Conclusion **35**
5.1 Discussion 35
5.2 Conclusion 37

Bibliography **39**

A Appendix 1 **I**

List of Figures

1.1 Table showing the number of participants, age, sex, and total MMSE score for each diagnosis subgroup for the baseline test 1

2.1 The red and blue lines each correspond to two separate empirical cumulative distribution functions and the black arrow represents the KS-statistic or the maximum absolute difference between them. From [9]. 5
2.2 A two-dimensional illustration of KNN imputation. The central red point (missing Feature 3) uses its 5 nearest neighbors (circled), found via Feature 1 and Feature 2, to estimate the missing value. 7
2.3 Numerical illustration of the MICE process. From [15] (CC BY 4.0). . 8
2.4 Scatter plots illustrating two-dimensional data before and after applying SMOTE. 9

3.1 A flowchart outlining the methodology used in this study, divided into three main stages: (a) Pre-Processing, (b) Model Training and Evaluation, and (c) Model Explainability. 15
3.2 Venn diagram highlighting the relationship between the variables (feature sets) used in all three clinical settings. 16
3.3 A flowchart outlining the full training phase. The orange blocks represent the model selection part and the blue blocks represent the final training part using the best performing model. 20

4.1 Visualizations of missing data patterns in the dataset. 24
4.2 Heatmaps showing the KS-statistic achieved and SSVD labels and variables remaining at different thresholds for column and row elimination where a row or a column with a missing value level over this threshold was eliminated. 25
4.3 Distributions of the six variables with the highest levels of missingness in the specialist care dataset, shown before and after imputation using MICE (4.3a) and KNN imputation (4.3b). 27
4.4 Confusion matrices showing the results of classification between SSVD and three other cognitive diseases and healthy control, averaged over five cross-validation folds. 29

4.5 SHAP analysis for AD vs SSVD classification highlighting the top 10 most discriminative features for the PC, SC, and RE settings. Left panels (a, c, e) display the mean absolute SHAP values for these features, with error bars indicating standard deviations across five cross-validation folds. Right panels (b, d, f) present SHAP summary plots, illustrating the distribution of SHAP values for each feature and how higher (red) or lower (blue) feature values impact the model's prediction towards SSVD (positive class) or AD (negative class). . . . 32

A.1 Distributions of a sample of 9 categorical variables in the research dataset, shown before and after imputation using KNN with clipping to nearest category (A.1a) and Custom KNN imputation using the mode for categorical variables (A.1b). II

List of Tables

3.1 Number of variables included in each setting. 16

4.1 Percentage of missing values before and after forward filling time-invariant variables, merging some variables, imputing age, and removing rows and columns with over 90% missing values. 23
4.2 Two-Sample KS test using MICE and KNN imputation. D represents the average KS-statistic over all columns before and after imputation. 25
4.3 Performance metrics for KNN imputation and MICE on the SC dataset. 26
4.4 Model Performance Metrics (HC vs SSVD) using SC dataset 28
4.5 Model Performance Metrics (MCI vs SSVD) using SC dataset 28
4.6 Model Performance Metrics (AD vs SSVD) using SC dataset 30
4.7 Model Performance Metrics (MIX vs SSVD) using SC dataset 30
4.8 Results of SMOTE analysis on the SC dataset. The table shows F1 and ROC AUC using the best performing model for each scenario with and without the use of SMOTE during training. 31
4.9 Performance metrics on AD vs SSVD classification using the entire feature set. 33
4.10 Performance metrics on AD vs SSVD classification using only the top 10 features with the highest mean absolute SHAP values. 33
4.11 Performance metrics on AD vs SSVD classification using only the top 30 features with the highest mean absolute SHAP values. 33

1

Introduction

This study uses data from 10 years of following patients and controls in the Gothenburg Mild Cognitive Impairment (MCI) study at the Memory Clinic in Gothenburg. The Gothenburg MCI study that started in 1999 has studied the effect and diagnostic procedure in Alzheimer's disease and related disorders in order to evolve the current nosological knowledge in the field of cognitive impairment [1]. The Gothenburg MCI study has made significant strides towards understanding cognitive impairment and dementia. The Gothenburg MCI study comprises 954 participants (406 male, 548 female) with a mean age of 65 years. The cohort represents a spectrum of cognitive health, from Healthy Controls (HC) and individuals with preclinical conditions like Subjective Cognitive Impairment (SCI) and Mild Cognitive Impairment (MCI), to patients with dementia diagnoses such as Alzheimer's Disease (AD), Mixed Dementia (MIX), and Subcortical Small Vessel Disease (SSVD). Figure 1.1 displays demographic data and the total scores of the baseline Mini Mental State Examination (MMSE) for each subgroup. The total MMSE score, derived from the MMSE test, is used to quantify an individual's cognitive status where a lower score indicates cognitive decline.

Diagnosis	HC	SCI	MCI	AD	MIX	SSVD
Number of participants	117	231	327	97	60	30
Median Age (range)	65 (31-77)	62 (27-79)	66 (43-80)	67 (54-81)	71 (51-79)	70 (51-79)
Sex (M/F, %)	47/70 (40/60)	92/137 (41/59)	139/187 (43/57)	33/64 (34/66)	24/36 (40/60)	20/10 (67/33)
Median MMSE Total (range)	29 (25-30)	29 (25-30)	28 (22-30)	25 (15-29)	25 (19-30)	25 (19-29)

Figure 1.1: Table showing the number of participants, age, sex, and total MMSE score for each diagnosis subgroup for the baseline test

Despite the advances in diagnostic procedures generated by studies such as the Gothenburg MCI study, accurately distinguishing between the specific types of cognitive impairments remains a challenge due to the amount of overlapping symptoms and variability in disease progression.

1.1 Background

SSVD is a common and often underdiagnosed form of vascular cognitive disease (VCD) with the latter making up around 25% of all dementia cases [2]. VCD is

characterized by damages to brain blood vessels and tissue and causes cognitive and motor impairments, and psychological symptoms such as anxiety and depression.

At the moment, making an accurate diagnosis of VCD requires brain imaging data from sources such as magnetic resonance imaging (MRI) along with detailed symptomatology, cognitive testing, and evaluation of cardiovascular disease and risk factors, which is both expensive and not always accessible, making VCD a highly underdiagnosed disease. This is deeply unfortunate as VCD, unlike other forms of dementia such as Alzheimer's disease (AD), can possibly be prevented and treated if an early diagnosis of the disease is established [3].

Traditional diagnostic techniques such as clinical assessments, using a specially trained physician, have been used to consider specific diagnostics in patients with a Global Deterioration Scale (GDS) level of 4 and above, indicating moderate to severe dementia. This diagnostic procedure relies heavily on specialized data such as certain levels of White Matter Hyperintensities (WMF) in the brain that are gathered from MRI imaging and assessed using a modified Fazekas scale [4]. These imaging techniques play a crucial role in detecting cognitive impairment in patients and are used, along with biomarkers, cerebrospinal fluid (CSF) and neuropsychological tests, to determine specific types of cognitive impairments such as SSVD or MIX, which corresponds to patients with both AD and SSVD pathologies. However, due to the unavailability of MRI imaging in both primary and specialist care, SSVD is generally believed to be missed in patients with unclear symptoms.

The use of computational algorithms in medical diagnosis is a well-established concept and has been studied and developed ever since electronic computers came into use in the 1950s and 1960s [5]. With the rise of computational algorithms, three major branches of machine learning have emerged: symbolic learning, statistical methods, and neural networks. All three branches developed advanced statistical and pattern recognition methods, including the k-nearest neighbor algorithm, decision trees, and Artificial Neural Networks (ANNs).

While machine learning techniques provide good accuracy and efficiency in diagnostic tasks, their interpretability varies across different methods. In the field of medical science, model interpretability is crucial in order for clinicians to understand the reasoning behind the model and trust its decisions to identify key variables contributing to disease classification. The interpretability of a model can be enhanced by the use of Explainable AI (XAI) approaches to provide local and global explanations for machine learning prediction. However, when the interpretability of the underlying model is lacking, XAI approaches face challenges in producing robust and meaningful explanations [6].

ANNs are notorious for providing excellent accuracy at the cost of interpretability. They are great at identifying patterns in large and complex sets of data and usually yield high accuracy and precision in classification tasks. However, ANNs are built on an often huge number of interconnected neurons, forming a highly complex network that lacks human interpretability. This makes them widely recognized as "black box" models and may therefore not be suitable in implementations where high interpretability is critical [7].

Previous work in implementing machine learning techniques to identify key variables in AD research using interpretable machine learning models has been extensive, with researchers exploring various approaches to improve classification accuracy and model interpretability. In [6], several different machine learning models were implemented in trying to classify AD patients using the large and publicly available AD dataset from the National Alzheimer's Coordinating Center (NACC). Their study compared performances between a number of different machine learning models while highlighting the use of feature selection and explainable approaches such as the use of a SHapley Additive exPlanations (SHAP) model to produce both local and global explanations. Their findings demonstrated that ensemble learning methods, in particular Random Forest (RF), and kernel-based methods such as Support Vector Machine (SVM), yielded high classification performance while maintaining a balanced trade-off between accuracy and interpretability.

Despite recent advances in the application of machine learning to nosological research, relatively few studies have systematically evaluated algorithmic performance across a broad spectrum of neurodegenerative diseases. Even fewer have focused specifically on the classification and explainability of SSVD, particularly when utilizing a diverse set of variables ranging from readily available clinical measures such as age, to more complex biomarkers derived from CSF analyzes and MRI imaging.

1.2 Aim

The aim of this project is to develop machine learning models to be able to classify SSVD in patients with high precision while having a high model interpretability. By leveraging real-world data gathered from the Gothenburg MCI study, this study entails the development of machine learning models with the aim of classifying patients into the different subgroups of cognitive impairment. Furthermore, this study will explore the dataset and trained models in order to discover the features which are most prominent when classifying the patients' specific cognitive impairments, focusing on the features associated with SSVD classification. High interpretability is crucial in order for medical professionals and clinicians to trust the results and adopt them in a clinical setting.

1.3 Specification of the Issue Being Investigated

To further specify the issue being investigated, the study will focus on the following research questions:

- How can machine learning models effectively classify SSVD with high precision and interpretability?
- Can machine learning models and XAI approaches identify variables that contribute significantly to the diagnosis of SSVD?
- What methods can be used to optimize machine learning pipelines to handle heterogeneous, noisy, or missing data in this context?

These research questions will be answered by leveraging data-driven experiments, model evaluation, and feature analysis techniques.

1.4 Limitations

Despite the potential advantages associated with developing classification models for SSVD, several limitations have to be considered. One significant concern is the bias of data. The models used in this study will be trained exclusively on the data from the Gothenburg MCI study. Despite the several advantages of using this dataset such as the huge feature variation, there exists a risk of bias due to the data being gathered from patients in a single region. As a result, the models may not be generalizable to a broader population.

Another limitation in this project is real-world clinical integration. Although interpretability is a priority, this project does not further describe the implementation into real-world clinical settings. The study does not include how well the extracted features fare when used in real-world diagnosis and should not be considered definitive indicators but rather as potential factors requiring further investigation by medical professionals.

While data privacy and fairness are acknowledged, this project does not include a detailed ethical or regulatory analysis regarding medical AI deployment. Issues such as bias mitigation, explainability requirements, and legal approval processes are not within the scope of this work.

2

Theory

2.1 Two-Sample Kolmogorov-Smirnov Test

The Two-Sample Kolmogorov-Smirnov (KS) test is a nonparametric statistical approach used to test the hypothesis that two samples have the same distributional properties [8]. Nonparametric tests are defined by their lack of or minimal assumptions about the underlying data distribution applicable to the analysis being performed.

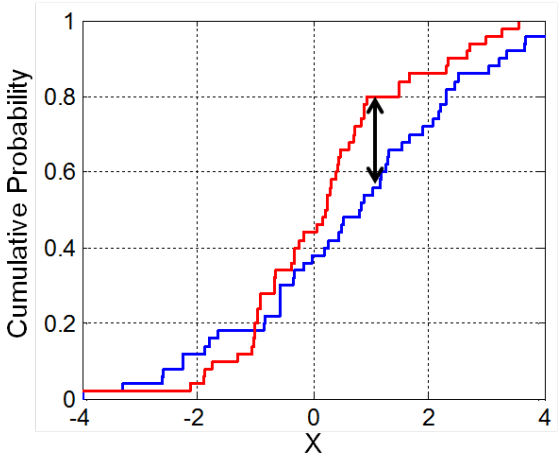


Figure 2.1: The red and blue lines each correspond to two separate empirical cumulative distribution functions and the black arrow represents the KS-statistic or the maximum absolute difference between them. From [9].

As Figure 2.1 depicts, the KS-statistic used in a Two-Sample KS-test is a quantitative measure that directly compares two Empirical Cumulative Distribution Functions (ECDFs). It measures the degree of separation of the two probability distribution by determining the maximum discrepancy that exists between the two.

Let there be two cumulative distribution functions, $F_{1,n}(x)$ and $F_{2,m}(x)$, for samples of sizes n and m, respectively. The KS-statistic $D_{n,m}$ is defined as given in Eq. 2.1.

$$D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)| \tag{2.1}$$

The $\sup_x$ notation symbolizes the maximum absolute difference for all values of x. The null hypothesis (H_0) assumes that the two samples have the same underlying distribution, and the alternative hypothesis (H_A) suggests that their underlying distributions are not the same.

For large sample sizes, the null hypothesis is rejected if the inequality given in Eq. 2.2 holds [10].

$$D_{n,m} > \sqrt{-\ln(\frac{\alpha}{2})\frac{1+\frac{m}{n}}{2m}} \tag{2.2}$$

The p-value is the smallest α for which this inequality holds. A large KS-statistic with a very low p-value indicates that there is a significant difference between the distributions and that bias due to imputation might have occurred. It is crucial, nonetheless, to accurately interpret the result of this test. Being unable to reject H_0 is not a proof that the two distributions are the same. It only implies that there is not enough statistical evidence to conclude that they are distinct. This is a significant distinction, as the measure is being used here as a heuristic tool to uncover major distortions created by imputation, rather than to validate the imputed data as in some way completely representative of the underlying, unseen data.

2.2 Missing Data Mechanisms: MCAR, MAR, and MNAR

Data missingness is usually split into three separate categories depending on the underlying cause of missingness. If there is no obvious pattern to the missingness and the missing values are independent of both observed and unobserved values, the missingness can be categorized as Missing Completely At Random (MCAR) [11].

If the missing values, however, can be explained by the observed values, the missingness is categorized as Missing At Random (MAR). For example, it might be found that patients scheduled for appointments on Fridays (an observed variable) are less likely to submit their pain scores (the variable with missing data) for that day, perhaps due to weekend plans or a different end-of-week routine. If this tendency to miss Friday submissions is consistent across different actual pain levels, then the missingness of pain scores is related to the observed day of the week, not the unobserved pain score itself.

When missingness cannot be categorized into neither MCAR nor MAR, then missingness is related to unobserved values and can be categorized as Missing Not At Random (MNAR). For example, again considering the study collecting self-reported daily pain scores, if patients experiencing very high pain scores (the variable with missing data) are too unwell or distressed to log their scores, then the missingness of the pain score data is directly related to the high severity of the pain (the unobserved values) they would have reported.

2.3 Imputation Methods

Multiple Imputation by Chained Equations (MICE) and K-nearest Neighbors Imputation (KNN imputation) are two advanced imputation methods for completing datasets. MICE operates under the assumption that the data missingness, which is to be imputed, is MAR. Using MICE when the missingness is not MAR could result in biased estimates. KNN imputation is not affected by the missingness of the data and can be used under any missingness mechanism [12].

2.3.1 K-Nearest Neighbors (KNN) Imputation

K-Nearest Neighbors (KNN) imputation is a method where, for a sample with a missing value in a particular feature, the k most similar samples (nearest neighbors) are identified from the dataset [13]. This similarity is determined using a distance metric calculated across the other available features. The missing value is then imputed using the weighted average of the neighbors' values for that feature (if numerical) or the most frequent value (mode) among the neighbors (if categorical). The "closeness" or similarity between the nearest neighbors is based on a distance metric, where Euclidean distance is the most commonly used.

Figure 2.2 illustrates how KNN imputation works in two dimensions. Here, the red point has a missing value in Feature 3 and uses its closest neighbors (Euclidean distance) to estimate the missing value, either by using the weighted average of its neighbors' values for Feature 3 (if Feature 3 is numerical) or by taking the mode (if Feature 3 is categorical).

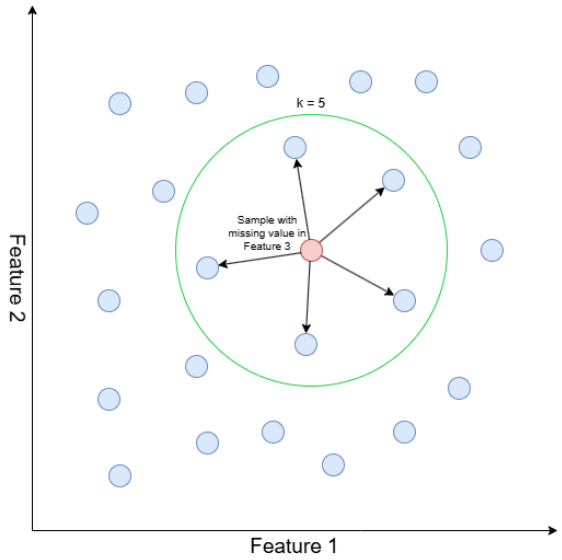


Figure 2.2: A two-dimensional illustration of KNN imputation. The central red point (missing Feature 3) uses its 5 nearest neighbors (circled), found via Feature 1 and Feature 2, to estimate the missing value.

2.3.2 Multiple Imputation by Chained Equations (MICE)

Multiple Imputation by Chained Equations (MICE) has emerged in statistical literature as one principled method of addressing and imputing missing data [14]. Unlike single imputation methods such as KNN imputation, MICE generates multiple imputations to account for the statistical uncertainty in the data and uses chained equations in order to base the imputed values on observed values.

This method is also very flexible, due to chained equations being able to handle varying types of data such as binary or continuous. The term "chained equations" refers to the iterative process of imputing values for each variable, which is then used as a predictor for imputing values in the next variable and so on [15]. This process ensures that the imputed values are based on observed values. The multiple imputation method involves filling in the missing values multiple times, which in turn creates multiple complete sets of data, each reflecting a plausible variation of the missingness pattern.

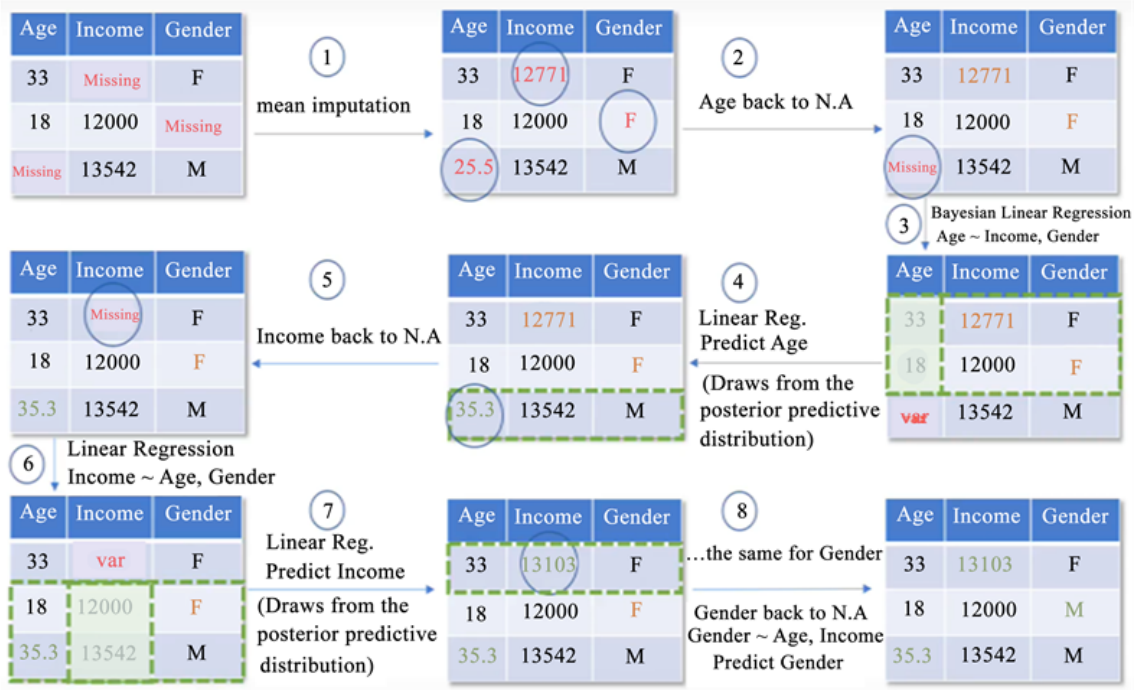


Figure 2.3: Numerical illustration of the MICE process. From [15] (CC BY 4.0).

MICE works, as illustrated in Figure 2.3, by using a simple imputation method to get a temporary value for all missing values. Then, for every variable, it treats it as the dependent variable and uses other variables as predictors. It then sets the missing values for the variable back to NaN and uses a regression model to predict a new value for all missing values for this variable, drawing from the posterior predictive distribution to maintain uncertainty. This process continues until a fixed number of iterations has been reached or the model is stable, i.e., the difference between the last two imputed values is zero or very small.

2.4 Synthetic Minority Over-sampling Technique (SMOTE)

Synthetic Minority Over-sampling Technique (SMOTE), as proposed in [16], is an over-sampling technique used to balance datasets with an imbalanced class distribution. An imbalanced class distribution in a dataset occurs when there are fewer instances of one class compared to another, often leading to models biased towards the majority class. One way of balancing the classes is to under-sample the majority class by excluding rows corresponding to the majority class until there are a similar amount of rows in both classes. This method, however, removes valuable information about the data that could be used to distinguish the two classes which is sub-optimal.

SMOTE works by over-sampling the minority class by creating "synthetic" examples rather than over-sampling with replacement. These synthetic examples are generated by interpolating between already existing examples of the minority class. In more detail, for each example or data point in the minority class, SMOTE selects one or more of its k-nearest neighbors and generates new instances along the line segments that are connecting them, as shown in Figure 2.4.

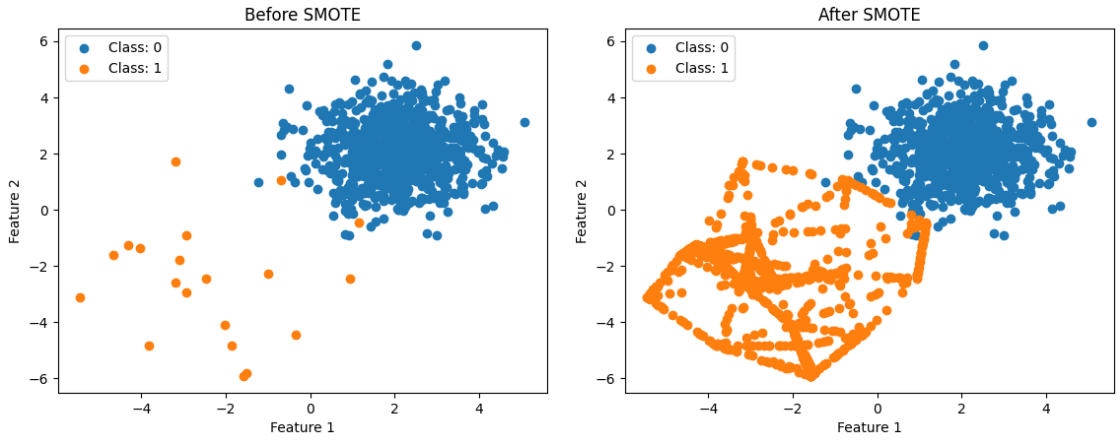


Figure 2.4: Scatter plots illustrating two-dimensional data before and after applying SMOTE.

By generating synthetic samples rather than simply duplicating existing ones, which can lead to overfitting, SMOTE helps create a more generalizable decision boundary for classification models. However, SMOTE assumes that minority class samples are well covered in their local vicinity, which is not guaranteed. Furthermore, if the minority class is highly overlapping with the majority class, SMOTE can generate synthetic samples that do not assist the classifier to distinguish between classes.

2.5 Machine Learning Models

This study employs a variety of machine learning models for classification. Each model offers different advantages depending on the nature of the data, the dimensionality, and the relationships between the variables. The following subsections describe the core principles of the machine learning models used in this study.

2.5.1 Random Forest (RF)

Random Forest (RF) is an ensemble learning method that works by constructing a large number of decision trees and outputting the mode or mean of the classes outputted by the individual trees. RF was first introduced in [17] and combines the concept of "bagging" with random feature selection to build a collection of decorrelated trees whose prediction by majority vote is more accurate than that of any individual tree.

More formally, given a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$, RF builds multiple decision trees $T_1, T_2, \ldots, T_B$, where each tree is trained on a "bootstrap" sample from D. For every node in the decision tree, a random subset of features is selected to determine the best split. This in turn introduces variability and reduces variance without significantly increasing bias.

2.5.2 Gradient Boosted Trees

Gradient Boosted Trees (GBT) is, similarly to RF, an ensemble learning method that uses decision trees as weak learners. Unlike bagging-based methods such as RF, GBT is based on the concept of boosting. Boosting focuses on reducing bias by sequentially training models that attempt to correct the errors made by previous models. First introduced in [18], Gradient Boosting optimizes a loss function over a function space by iteratively choosing a function which is typically a decision tree that points in the negative gradient direction.

Formally, at every iteration, a new decision tree $h_m(x)$ is fit to the negative gradient of the loss function L with respect to the current model prediction $F_{m-1}(x)$ as seen in Eq. 2.3 and the updated model can be seen in Eq. 2.4.

$$h_m(x) = \arg\min_{h} \sum_{i=1}^{n} \left[-\frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)} - h(x_i) \right]^2 \tag{2.3}$$

The updated model becomes:

$$F_m(x) = F_{m-1}(x) + \nu h_m(x) \tag{2.4}$$

where ν is the learning rate used to control the contribution of each tree.

2.5.2.1 XGBoost

There are several optimized implementations of GBT. Extreme Gradient Boosting (XGBoost) is a highly efficient and scalable implementation that introduces additional regularization terms to the loss function to prevent overfitting and optimization techniques such as approximate greedy algorithms for split finding [19].

2.5.2.2 LightGBM

LightGBM is another example of an implementation of GBT. Introduced by Microsoft, LightGBM further improves training speed and model efficiency by using a histogram based algorithm and leaf-wise tree growth instead of level-wise which makes it effective for large datasets and high-dimensional feature spaces [20].

2.5.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a learning method used predominantly in classification tasks. SVM attempts to find the optimal hyperplane that separates classes with the highest margin [21]. Supposing data cannot be separated linearly in input space. SVM can then make use of kernel functions, such as the radial basis function or polynomial kernels, to transform data into higher spaces where a linear separator may exist.

Formally, given labeled training data $\{(x_i, y_i)\}_{i=1}^{n}$ where $y_i \in \{-1, 1\}$, SVM solves the optimization problem seen in Eq. 2.5.

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}\cdot x_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0 \end{aligned} \tag{2.5}$$

where $\mathbf{w}$ and b define the hyperplane, ξ_i are slack variables allowing some misclassifications, and C is a regularization parameter balancing margin maximization and classification error.

2.5.4 Naive Bayes

Naive Bayes is a Bayes' Theorem-based probabilistic classifier assuming the "naive" hypothesis that features are conditionally independent of one another given the class label [22]. Despite this naive hypothesis, Naive Bayes classifiers generally perform well in practice, particularly in high-dimensional settings.

Bayes' Theorem can be seen in Eq. 2.6.

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)} \tag{2.6}$$

Since $P(x_1, \ldots, x_n)$ is constant for all classes, the classifier simplifies to choosing the class y that maximizes Eq. 2.7.

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y) \tag{2.7}$$

Where $\hat{y}$ is the class that maximizes the conditional probability $P(y \mid x_1, \ldots, x_n)$.

2.6 Explainable AI (XAI)

Explainable AI (XAI) refers to a collection of methods and techniques that allows human users to understand, interpret, and trust the outcomes generated by complex machine learning models [23]. By describing how the models arrive at the specific decisions, XAI enhances transparency and supports accountability in AI models and improves the interpretability of complex AI models. It plays a key role in evaluating accuracy, fairness, and outcomes in AI-driven decision making.

2.6.1 SHapley Additive exPlanations (SHAP)

SHapley Additive exPlanations (SHAP) is an XAI method first introduced in [24]. The SHAP method assigns so-called "SHAP values" to features in a dataset using the concept of cooperative game theory. The cooperative game theory framework can be interpreted as follows: imagine there are several players cooperating in completing a game; how would the reward for each player be assigned when each player contributed differently [25]? The reward, or the SHAP values, is assigned to each player depending on the player's marginal contribution to the game. The marginal contribution of each player is defined by four axioms:

- **Efficiency** - The total reward has to be fully distributed between all contributing players
- **Symmetry** - Players that contribute equally receives an equal reward
- **Dummy** - Players that do not contribute receives no reward
- **Additivity** - If a game consists of multiple components, a player's reward should reflect their contributions to each part individually rather than just the overall outcome.

Using this, SHAP can be utilized in healthcare to identify biomarkers and clinical features (players) contributing to a specific disease outcome (reward).

Mathematically, SHAP utilizes an additive feature attribution method that is a linear function of binary variables, as seen in Eq. 2.8 [26].

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{2.8}$$

Where $g(z')$ is the explanation model, $z' \in \{0,1\}^M$ is the binary vector representing the presence or absence of each of the M input features. ϕ_0 is the baseline prediction of the explanation model. ϕ_i is the SHAP value of feature i.

SHAP values ϕ_i for each feature i are calculated in Eq. 2.9.

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right] \tag{2.9}$$

Where F is the set of all features and S is the subset of features not including feature i. $f_S(x_S)$ is the expected model output using only features in S. This represents a weighted average contribution over all possible feature subsets.

[25] reviewed 45 studies that implemented SHAP and found that more than three-quarters of them had proposed machine learning models to be used with SHAP, with XGBoost being the most frequently used (14 studies), followed by RF (10 studies). This preference may be explained by the high-speed exact SHAP algorithm which is specifically designed for tree-ensemble models such as XGBoost, RF, LightGBM, and CatBoost. Additionally, they found that SHAP has been widely used in hospital management applications (17 studies), including mortality prediction in Intensive Care Unit (ICU) patients, hospital admissions and readmissions, surgical complications, and other adverse outcomes of treatment.

3

Methods

The methodology of the study can be divided up into three main parts: pre-processing, model training and evaluation, and model explainability. Figure 3.1 outlines these main parts and their sub-processes in a flowchart. All analyses were conducted using Python, leveraging libraries such as Scikit-learn for preprocessing and model training in addition to LightGBM and XGBoost, and Imbalanced-learn for SMOTE. Data handling and visualization were performed using Pandas and Matplotlib, respectively.

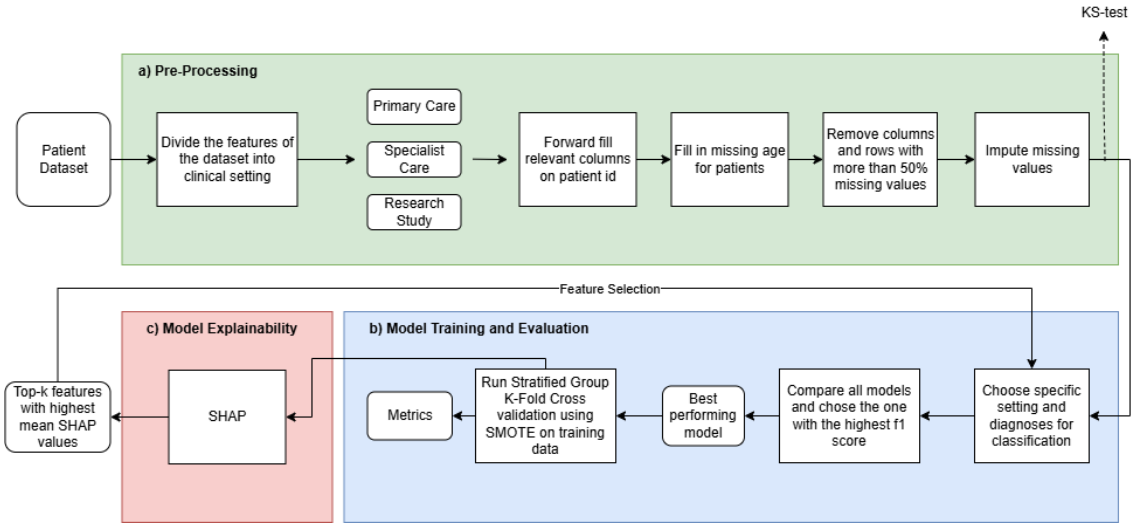


Figure 3.1: A flowchart outlining the methodology used in this study, divided into three main stages: (a) Pre-Processing, (b) Model Training and Evaluation, and (c) Model Explainability.

3.1 Pre-Processing

A series of steps must be taken to appropriately incorporate the raw data into a machine learning classification pipeline. These include using imputation methods to fill in the missing values in the data and render it complete, dropping columns (features) and rows that contain rare values, and transforming the dataset into an appropriate structure, i.e. the formation of paired data and corresponding labels paired with aligned instances.

3.1.1 Dataset Generation

The raw dataset from the Gothenburg MCI study, gathered in an Excel document, contains several sheets representing four different diagnostic domains with around 700 variables across all of them. The types of variables include integer-encoded categorical and continuous. Although most data were gathered during patients first visits, there were five rounds of testing conducted at two-year intervals. This study is interested in investigating the classification performance and feature extraction in three clinical settings: primary care (PC), specialist care (SC), and research study (RE). Thus, features accurately representing these clinical settings had to be identified and split into three separate datasets.

Identifying features for the clinical settings was done by studying literature, such as the yearly report from SweDem [27], consulting clinical experts, and analyzing the data. Each clinical setting differs in terms of the number of variables. The relationship between the variables used in each clinical setting can be seen in Figure 3.2. The final number of variables in each setting can be seen in Table 3.1, this is the number of variables before any data pre-processing has been performed.

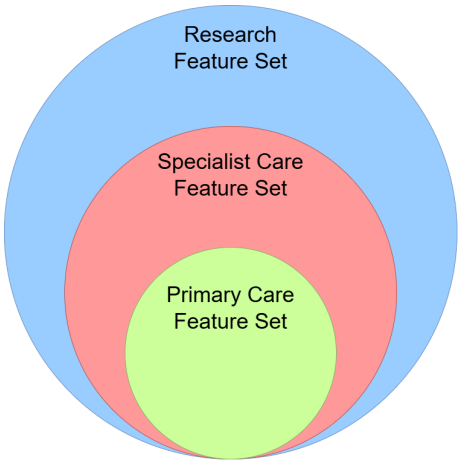


Figure 3.2: Venn diagram highlighting the relationship between the variables (feature sets) used in all three clinical settings.

Setting	# Variables
PC	71
SC	137
RE	606

Table 3.1: Number of variables included in each setting.

In primary care settings, features used are typically those that are available in general practice, such as basic anamnesis and status information, basic cognitive tests, and easily accessible biomarker data [27]. These features were chosen to reflect real-world

limitations of primary care, where advanced imaging or specialized biomarkers and neuropsychological tests may not be accessible.

The features or variables used for the specialist care setting expand upon the primary care feature set by adding specialized cognitive test features, biomarkers, and neuropsychological tests. This reflects the more thorough diagnostic process available in specialist care, where patients are referred for further evaluation after initial concerns arise.

Finally, the research feature set includes all available features from the Gothenburg MCI study and incorporates advanced imaging data such as WMH volumes and intracranial volume. This setting represents the most comprehensive data, typically only available in research environments or highly specialized clinics.

Once the feature sets were compiled for all three settings, the data were split into three separate setting datasets containing all the setting features, patient numbers, and all five rounds of testing.

3.1.2 Handling Missing Values

Missing values in large datasets are a very common problem, especially in medical datasets, and the dataset used in this study is not an exception. There are several methods used to handle missing values; this study has employed some of these methods.

The main goal of imputation in this context is to fill in missing values in a way that preserves the original statistical properties and underlying relationships between variables in the dataset. If an imputation method significantly alters the distribution of a variable, it can introduce bias. Furthermore, this bias can mislead machine learning models to learn incorrect patterns, which in turn diminishes their predictive accuracy and undermines the reliability of explainability methods such as SHAP to uncover variables with high predictive contribution. Evaluating the performance of an imputation method by measuring the distributional similarity is therefore crucial. The Two-Sample KS test serves as a quantitative measure for this specific purpose and helps to ensure that the imputation does not fundamentally distort the integrity of the data which is essential for robust and trustworthy diagnostic models.

The missing data in this dataset can be considered MAR since their occurrence is systematically related to observed variables, i.e., the year of study and the type of variable. Specifically, missingness is more common in later years, presumably because the frequency of testing diminishes with time rather than being completely random. Furthermore, there are some variables with higher rates of missing data, potentially due to changes in clinical practice, test availability, or study protocol. Since the nature of the missing values depends on these observed variables and not on the unobserved values themselves, the MAR assumption is reasonable, which allows the use of imputation techniques such as MICE or KNN imputation.

To further investigate the missingness of the data, a missingness matrix was generated to better visualize the missingness of the data where the missing values are represented

by white blocks and observed values by black blocks. If the missingness is scattered randomly, the missing values suggest MCAR. Patterns among the missing values in the matrix may indicate conditional missingness MAR or MNAR. Additionally, a missingness heatmap will be generated that indicates correlated missingness, or nullity correlation, between the features of the data. The nullity correlation ranges from -1 to 1. A nullity correlation value of -1 represents a perfectly negative relationship where if one variable appears, the other definitely does not. A value of 0 represents no correlation and a value of 1 represents a perfectly positive relationship where if one variable appears, the other one is also present.

In the first step of the missing data handling an analysis of the variables was employed to figure out if some of the variables were relaying the same information and, if so, merge these variables into one variable. For example, there were three different variables indicating if the patient had, or has, diabetes and if they were using some sort of treatment. These three variables were merged into one variable, simply indicating if the patient has had or is currently having diabetes. This reduced redundancy, simplified the dataset, and increased the amount of information captured within a single variable.

Secondly, many of the variables only had available data for the first round or rounds, where some of these are time-invariant variables. Time-invariant variables are those that do not change over time, such as sex, birth date, and some anamnesis and status variables. To handle missing values for these time-invariant variables, a forward fill method was employed where the last known values for the variable are propagated forward, maintaining the trend of the previous data. An algorithm was used to estimate the missing values in the age variable by calculating the difference in years between known age values and filling in the gaps accordingly.

Before imputation, removal of rows and columns with high counts of missing values is crucial for reliable imputation performance. A test was therefore employed in order to evaluate the influence of threshold differences on row and column elimination by percentage missing data. Rows and columns with percentage missing values greater than the thresholds were removed. The experiment systematically tested all combinations of row and column deletion thresholds at the levels of 40%, 50%, 60%, 70%, 80%, and 90%. For every threshold combination, several measurements were recorded, such as the KS statistic for measuring imputation performance, the number of variables remaining, and the number of SSVD labels remaining. The test was to discover the pair of row and column thresholds that sufficiently preserved the original data distribution while minimizing the loss of both variables and SSVD labels. Columns were removed first, followed by rows, as preserving data points was deemed more important.

Finally, two different imputation methods were utilized, including MICE and KNN imputation. A Two-Sample KS test was performed to test the effectiveness of the imputation methods by comparing the distributions of the whole dataset before and after imputation using a Kernel Density Estimation (KDE) technique. Furthermore, a comparison of the distributions of the top-6 variables with the highest amount of missing values were generated to see how the distribution of features with a high

missing rate will be handled using these imputation methods.

Standard imputation methods, such as scikit-learn's KNNImputer (KNN imputation) and IterativeImputer (MICE) do not inherently handle categorical values but rather treat them as numerical values, which can lead to misleading imputations or the creation of non-existent categories. This is especially critical for ordinal variables with a given order, but where differences between successive values do not increase uniformly (for example, the clinical difference of scoring 0 compared to 1 will probably not be the same as for the difference of scoring 1 and 2). The use of imputation methods that rely on distance or regression inherently makes the assumption that the intervals are equally spaced.

To visually assess the impact of imputation, KDE plots are used. It is acknowledged that KDE plots create a smooth, continuous curve from discrete data points, which can give the impression of "smearing". However, their purpose here is not to perfectly represent the discrete nature of the data, but to provide a visual approximation for comparing the overall shape, centering, and spread of a variable's distribution before and after imputation. This allows for the detection of significant distortions.

To manage the issue of non-existent categories, a pragmatic approach was taken. The categories were saved before imputation, and the imputed values for all categorical features were subsequently clipped to the nearest valid category. While this does not resolve the underlying assumption of equal intervals, it ensures that the final imputed dataset contains only valid categorical values. In addition to this approach, a custom KNN imputation method were employed and compared to the aforementioned technique. By adhering to theory, this method imputes categorical variables using the mode and numerical variables using the mean, each calculated from the values of their k-nearest neighbors.

3.2 Model Training and Evaluation

A flow chart of the training part of the methodology is illustrated in Figure 3.3.

For model training, a stratified group k-fold cross-validation strategy was used, where, in each iteration, the training set was split into five folds, with one fold held for testing while the rest was used for training. In addition to maintaining patient group integrity and preventing any data leakage by ensuring that the same patient does not end up in both training and test set, the approach also preserves class distribution in each split by using stratified sampling. This k-fold cross-validation approach helps evaluate the model performance across several folds and achieving a robust evaluation by averaging the metrics across all folds.

3.2.1 Handling Class Imbalance

The data set exhibited significant class imbalance, with SSVD cases being the least frequent class. Without any countermeasures, class imbalance could impact model generalization during training by biasing towards the majority class. Therefore, SMOTE was employed during model training to over-sample the minority class and

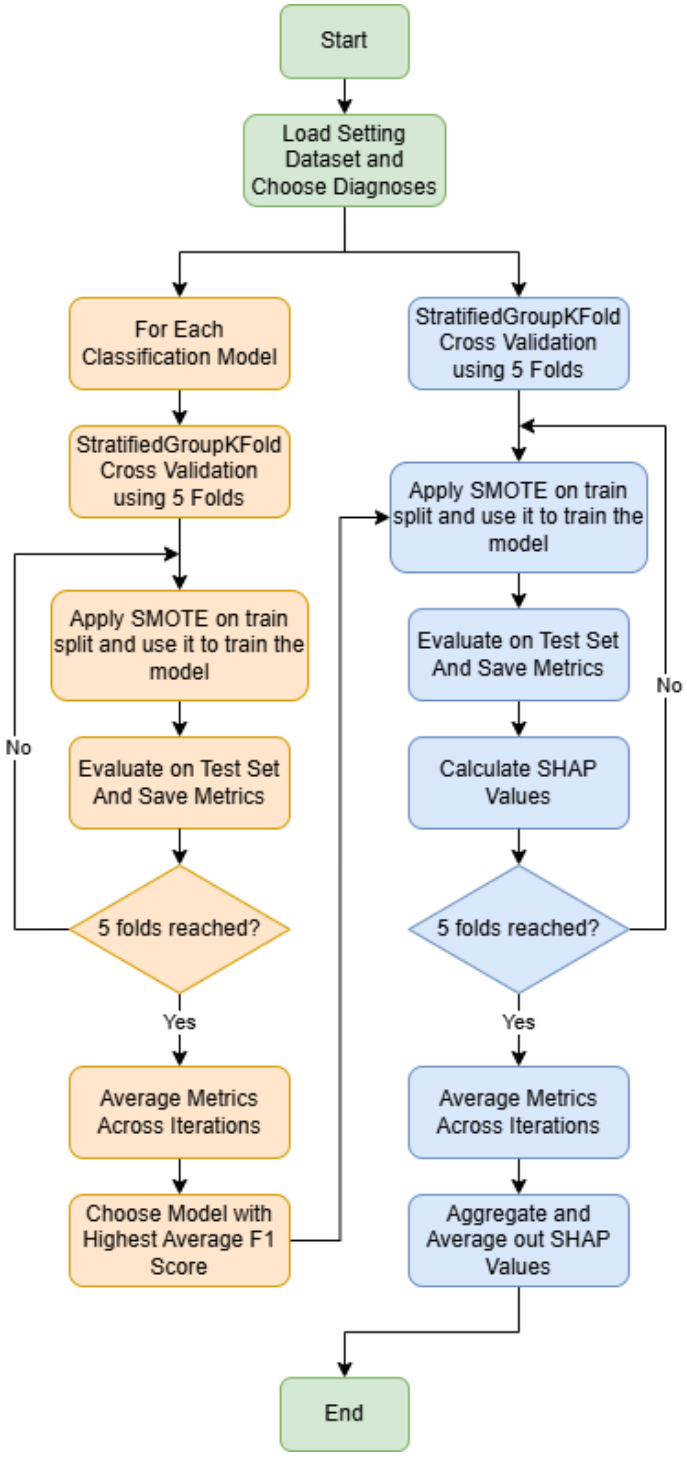


Figure 3.3: A flowchart outlining the full training phase. The orange blocks represent the model selection part and the blue blocks represent the final training part using the best performing model.

balance out the class distribution for the training set while the test set remained the same. In addition to using SMOTE, a stratified sampling technique was implemented during training to further address class imbalance challenges.

Stratified sampling is used to fairly represent each class in both the training and test splits by preserving the original class distribution in each split. This ensures that performance are not biased due to uneven class distribution across different splits. By combining both SMOTE and stratified sampling, the model was trained on a more balanced data set while the test set remained untouched, reflecting the real distribution of diagnoses. This approach aimed to improve the model's ability to generalize across all classes, particularly the underrepresented SSVD group.

3.2.2 Machine Learning Models

When training using a specific setting dataset and diagnoses, several different models were trained and evaluated using stratified group k-fold and an average F1 score over five folds. The model that achieved the highest average F1 score for the specific dataset and diagnoses was subsequently trained and evaluated again. Performance metrics, confusion matrices, and SHAP values were averaged over all five folds, achieving a more robust performance evaluation. The models used include the following: XGBoost, LightGBM, Random Forest (RF), Support Vector Machine (SVM), and Naive Bayes (NB).

RF was chosen for its ability to handle noisy clinical data while maintaining high performance and interpretability. It was also proven in [6], in addition to SVM, to perform well on medical data. XGBoost and LightGBM were included due to their high performance on structured data tasks, particularly on healthcare tasks, where they have a tendency to outperform traditional models by modeling complex feature interactions and handling missing values more gracefully. For example, studies have shown that XGBoost and LightGBM outperform other models such as Logistic Regression and SVM in myocardial infarction prediction, highlighting their potential in healthcare settings [28]. NB was included as a comparison model due to its simplicity, and a high performance by NB could indicate an approximate independency between the variables used in the classification.

3.2.3 Evaluation Metrics

Model performance was assessed using standard classification metrics, including accuracy, precision, recall, F1 score, and the area under the receiver operating characteristic curve (AUC ROC). For these metrics, values range from 0 to 1, where a score closer to 1 signifies better performance. Accuracy is a naive approach to model evaluation that measures the share of correctly predicted labels among the total number of instances. This provides a general sense of the model performance. However, on for example imbalanced datasets, this metric may be misleading as it may be biased towards the majority class while neglecting the minority class performance. Other metrics such as precision, recall and F1 score were therefore employed to more accurately represent the model performance.

Precision measures the quantity of true positive predictions among all positive predictions, indicating how reliable the model is when predicting the positive class. Recall measures the number of true positive predictions relative to all actual positive instances, indicating how well the model finds all positive instances. The F1 score is a harmonic mean of both precision and recall which balances the trade off between them, and is especially useful when the classes are imbalanced. Precision, recall, and F1 score were macro-averaged which give equal weight to each class regardless of sample size, which is a good indication of performance on imbalanced datasets.

AUC-ROC assesses the model's ability to distinguish between classes across all possible classification thresholds (between 0 and 1) and calculates the area under the curve (AUC) generated by plotting the true positive rate against the false positive rate. Values close to 1 indicates great discriminative ability and robustness and values near 0.5 suggest that the model performs no better than random chance.

3.3 Model Explainability and Feature Extraction

To identify key features for the specific classification task and dataset, the XAI approach SHAP was employed. At each fold of the best performing model for the specific task and dataset, SHAP values were generated for each feature and data point in the test set. After all folds were used, the SHAP values were averaged to get a more realistic estimate of the feature's overall contribution to the model's predictions. The standard deviation of the features SHAP values were also calculated, indicating if the feature's influence was consistent across folds or not.

The top-k features with highest mean SHAP values were plotted on a bar chart along with their standard deviations in order to visualize their impact on the model and if the impact were consistent during the training iterations. A beeswarm plot of the SHAP values for the top-k features were also employed in order to visualize the distribution of SHAP values for each data point and which values of a feature were associated with which diagnosis. The models were retrained using only the top-k features to see how well they performed using purely these features.

4

Results

4.1 Handling Missing Values

Figure 4.1 shows the missingness matrix and the missingness correlation heatmap, both using the specialist care dataset and the top two hundred variables with the highest level of missingness. Both the missingness matrix and missingness correlation heatmap show that missingness tends to occur in structured patterns and particularly across subsets of features which rules out the possibility of MCAR. This pattern is consistent with the MAR assumption, where missingness is dependent on observed variables. Although these figures alone cannot rule out the possibility of MNAR, the patterns observed and the knowledge that missingness is mostly related to testing round makes an assumption of MAR on the missingness of this dataset reasonable.

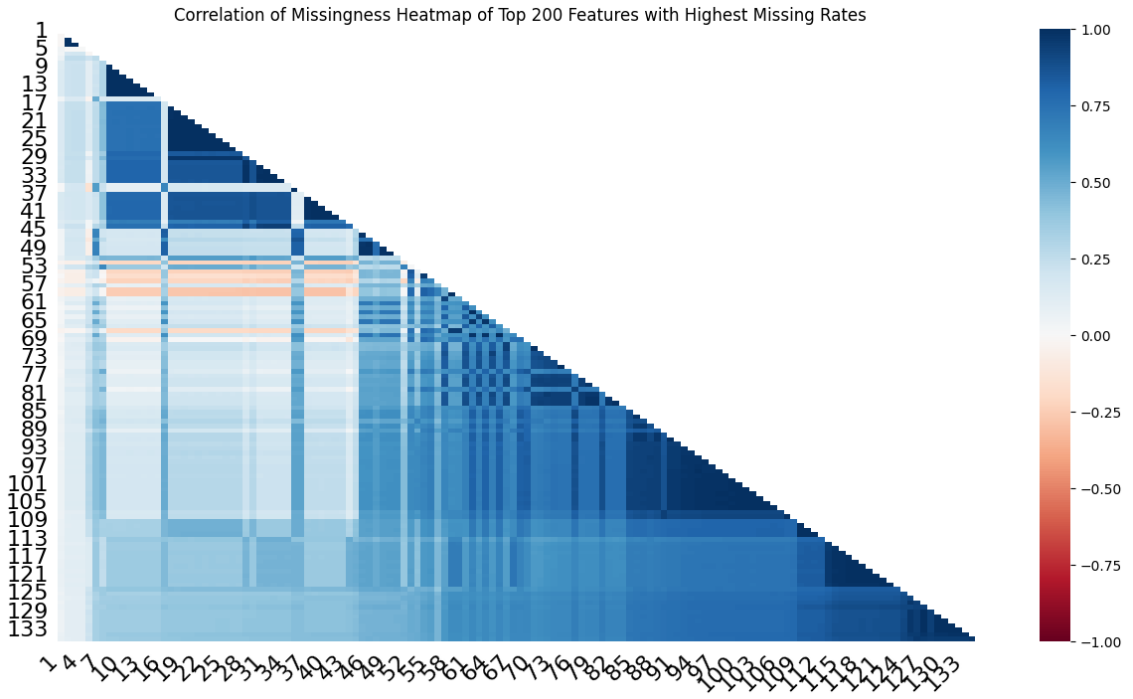
Figure 4.2 shows the variables and SSVD labels remaining using all pairs of column and row thresholds. This shows that using a row and column threshold of 0.9 (90%) retains most of the original variables while ensuring a sufficient number of SSVD labels and achieves a reasonable KS-statistic of 0.0777.

Table 4.1 shows the percentage of missing values before and after forward filling time-invariant variables, merging some variables, imputing age, and removing rows and columns with over 90% missing values.

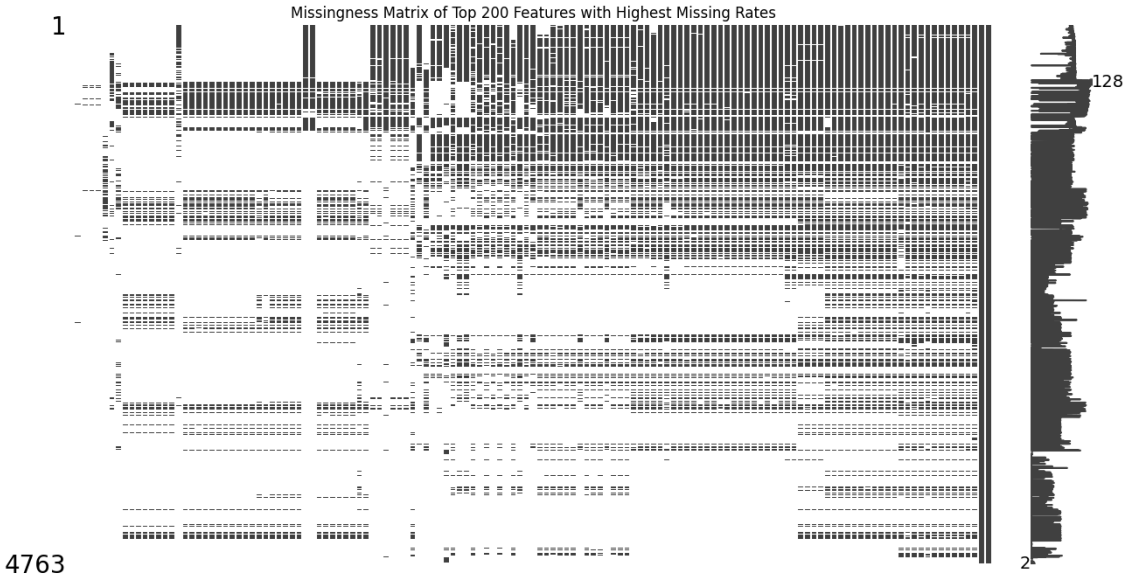
Setting	**% Missing (Before)**	**% Missing (After)**	**# Variables (Before)**	**# Variables (After)**
PC	65.47%	32.74%	71	68
SC	68.87%	40.56%	137	131
RE	75.50%	44.60%	606	487

Table 4.1: Percentage of missing values before and after forward filling time-invariant variables, merging some variables, imputing age, and removing rows and columns with over 90% missing values.

Table 4.2 shows the average KS-statistic and the p-value of the three different setting data sets averaged over all columns using MICE and KNN imputation.

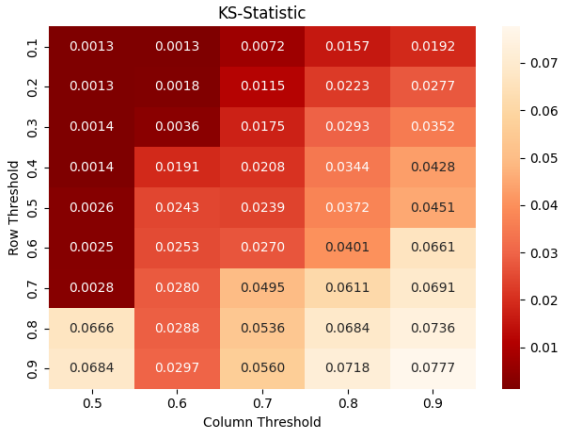


(a) Missingness correlation heatmap of the top 200 features with the highest missing rates.

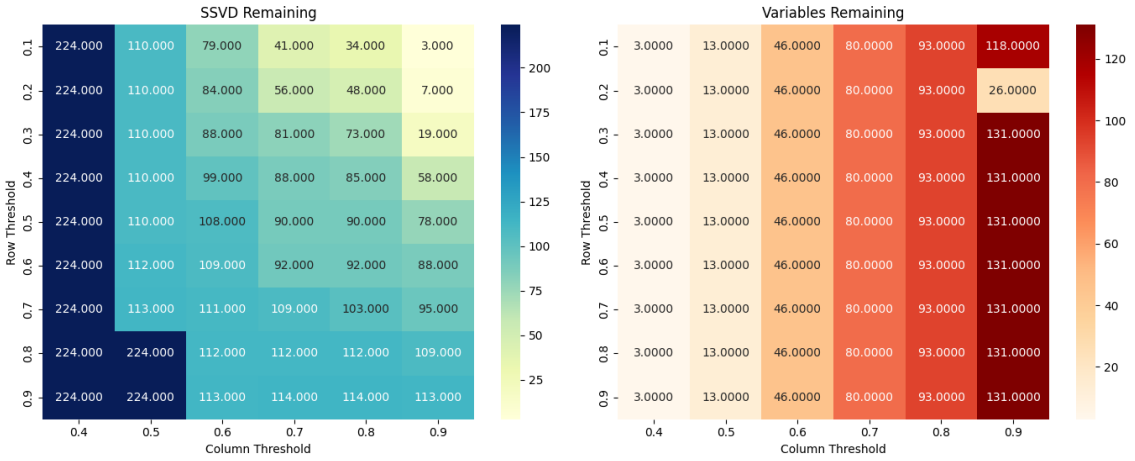


(b) Missingness matrix of the same 200 features, showing the distribution of missing values across samples.

Figure 4.1: Visualizations of missing data patterns in the dataset.



(a) Heatmap showing the KS-statistic achieved by KNN imputation at different row and column elimination thresholds.



(b) Heatmap showing the number of SSVD labels and variables retained at different row and column elimination thresholds.

Figure 4.2: Heatmaps showing the KS-statistic achieved and SSVD labels and variables remaining at different thresholds for column and row elimination where a row or a column with a missing value level over this threshold was eliminated.

Setting	**D (KNN)**	**D (MICE)**	**p-value (KNN)**	**p-value (MICE)**
PC	0.0608	0.1028	0.3264	0.3548
SC	0.0777	0.1202	0.2701	0.2522
RE	0.0898	0.1112	0.1974	0.2155

Table 4.2: Two-Sample KS test using MICE and KNN imputation. D represents the average KS-statistic over all columns before and after imputation.

The result of the Two-Sample KS tests using MICE and KNN imputation indicates that KNN imputation, on average, preserves the distribution of the original columns better than MICE. The high p-value implies that the null hypothesis, stating that the distributions before and after imputation are similar, cannot be rejected.

For MICE, the KS-statistic is less variable across PC, SC, and RE datasets compared to KNN imputation and is actually lower for the RE dataset than SC. This could indicate that MICE is more resistant to increasing dimensionality and operates better in higher-dimensional contexts than KNN. On the other hand, the comparatively stable or even improved performance of MICE with increasing features indicates that the datasets employed in this study are perhaps not very high-dimensional, since extremely high-dimensionality would have the effect of degrading the performance of both methods.

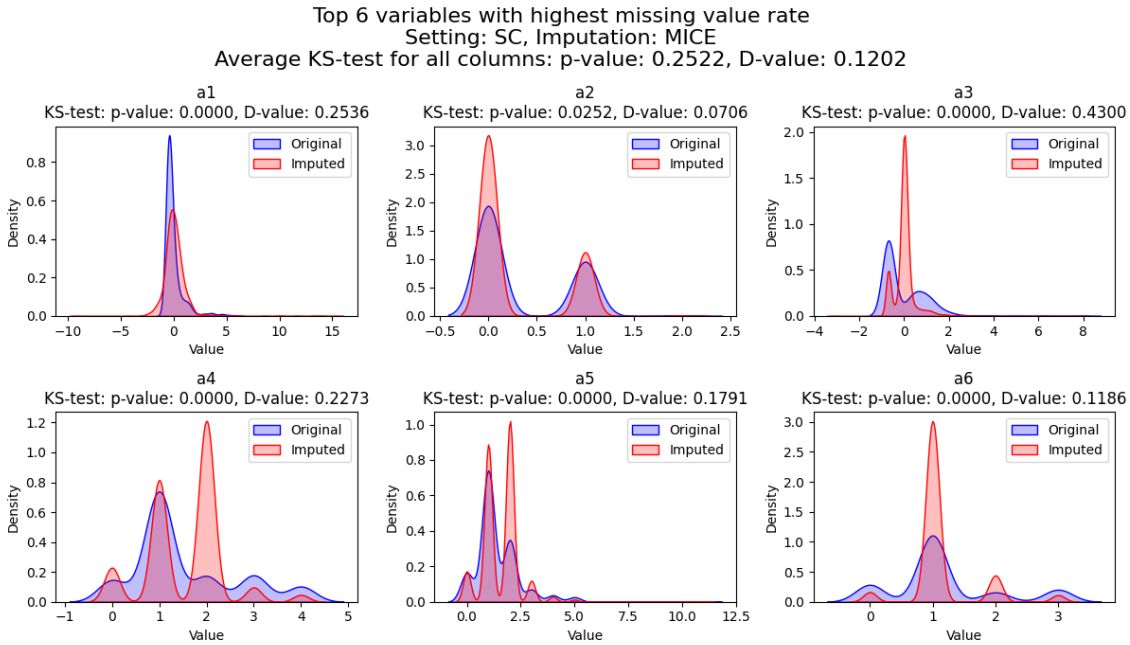
To further investigate the performance in imputation between MICE and KNN imputation, individual column distributions before and after imputation were analyzed. The distributions of the six variables with the highest levels of missingness in the specialist care dataset can be seen in Figure 4.3. Both imputation methods struggle to perfectly capture the distributions of the original dataset. The distribution plots, in addition to their KS-statistics and p-values, seem to suggest that KNN imputation does a better job than MICE in preserving the original distributions. KNN imputation also does not impute values outside the observed value range which seems to be the case for MICE, which is evident in the Figure 4.3a1 and Figure 4.3b1 distribution plots showing the same variable imputed using MICE and KNN imputation respectively.

Table 4.3 highlights the classification performance (F1 score and ROC AUC) using both imputation methods, all setting datasets, and five different classification scenarios, including SSVD in each. Models trained on data imputed by KNN imputation more frequently achieved higher F1 and ROC AUC scores. This seems to suggest that KNN imputation does a better job at preserving the predictive structure of the data.

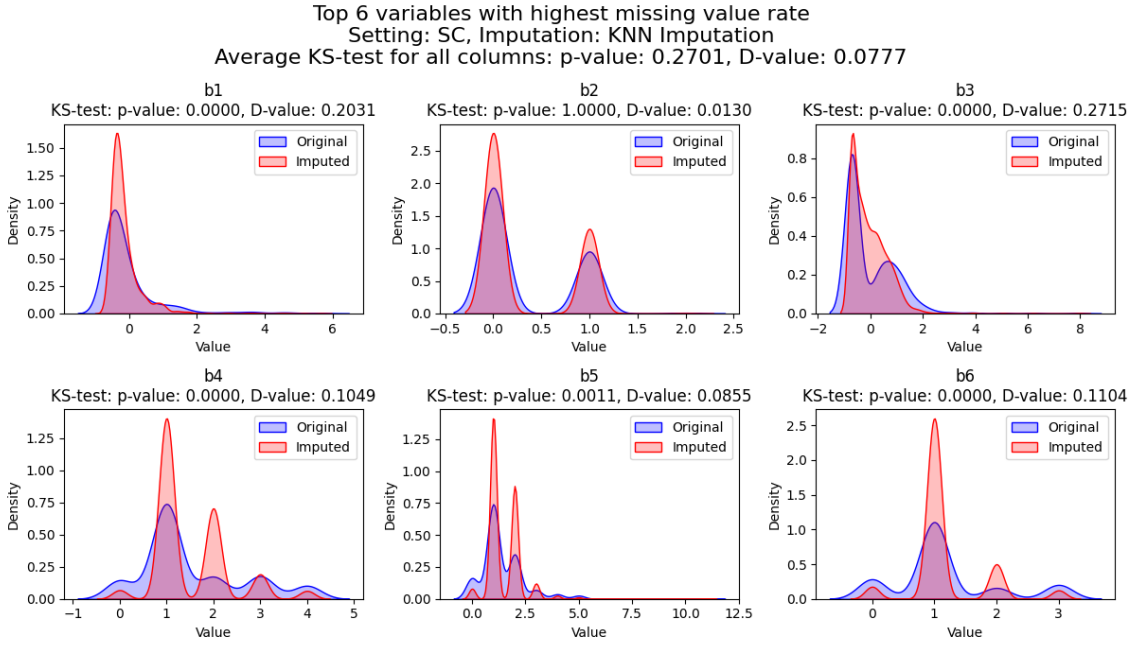
Classification Scenario	**F1 (KNN)**	**F1 (MICE)**	**ROC AUC (KNN)**	**ROC AUC (MICE)**
SSVD vs AD	0.789	0.772	0.772	0.732
SSVD vs MIX	0.749	0.743	0.775	0.768
SSVD vs HC	0.960	0.956	0.970	0.963
SSVD vs MCI	0.757	0.745	0.725	0.715

Table 4.3: Performance metrics for KNN imputation and MICE on the SC dataset.

A custom KNN imputation method was implemented, which inherently handled categorical and numerical variables. Although this method adheres to theory, by taking the mode of the nearest neighbors for missing values in categorical columns, it still performed worse than using KNN imputation and clipping imputed categorical values to the nearest category. The custom KNN implementation achieved an average KS-statistic of 0.1306 over all columns on the research dataset and, compared to KNN



(a) MICE



(b) KNN imputation

Figure 4.3: Distributions of the six variables with the highest levels of missingness in the specialist care dataset, shown before and after imputation using MICE (4.3a) and KNN imputation (4.3b).

imputation and clipping, a higher or equal KS-statistic on a sample of categorical variables. This can be seen in Appendix A.1. The KNN imputation with clipping were used in all results onward due to the observed superiority of the KNN imputation implementation compared to MICE and the custom KNN imputation method.

4.2 Model Evaluation

The following tables show the classification results using all models and several scenarios, including SSVD and using the specialist care dataset. Table 4.4 shows the classification performance of all models discriminating between HC and SSVD, where RF achieved the highest F1 score of 0.96. Table 4.5 shows classification performance between MCI and SSVD, where RF achieved the highest F1 score of 0.757. Table 4.6 is between AD and SSVD, where LightGBM achieved the highest F1 score of 0.789. Table 4.7 show the performance between MIX and SSVD, where SVM performed the best with a F1 score of 0.749. These results show that ensemble methods are highly effective in discriminating between SSVD and other cognitive diseases such as AD. This seems to be the case even when using a larger feature set in the research dataset. The only scenario when ensemble methods were not superior, were in MIX vs SSVD classification, where SVM performed much better than the ensemble methods achieving a F1 score of 0.749 compared to 0.689 for the best performing ensemble model. Classification difficulty increases progressively from HC vs SSVD to MCI vs SSVD, and then to AD vs SSVD, which is expected.

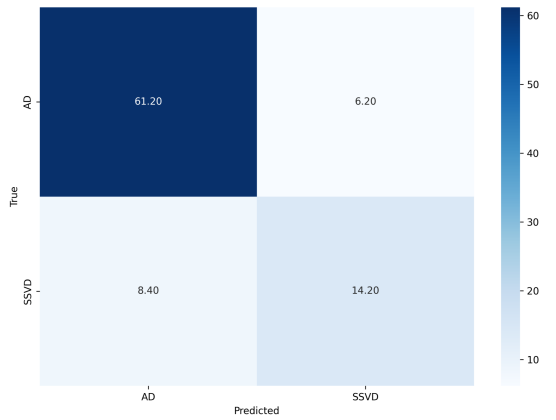
Table 4.4: Model Performance Metrics (HC vs SSVD) using SC dataset

Model Name	**Accuracy**	**Precision**	**Recall**	**F1**	**ROC AUC**
XGBoost	0.966	0.939	0.955	0.946	0.955
LightGBM	0.971	0.954	0.959	0.956	0.959
RF	0.973	0.952	0.970	0.960	0.970
SVM	0.965	0.955	0.947	0.948	0.947
NB	0.907	0.840	0.900	0.861	0.900

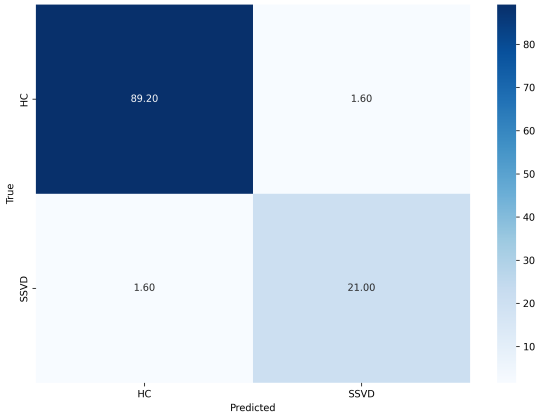
Table 4.5: Model Performance Metrics (MCI vs SSVD) using SC dataset

Model Name	**Accuracy**	**Precision**	**Recall**	**F1**	**ROC AUC**
XGBoost	0.885	0.774	0.724	0.731	0.724
LightGBM	0.885	0.776	0.714	0.728	0.714
RF	0.900	0.834	0.725	0.757	0.725
SVM	0.846	0.688	0.684	0.678	0.684
NB	0.397	0.514	0.532	0.371	0.532

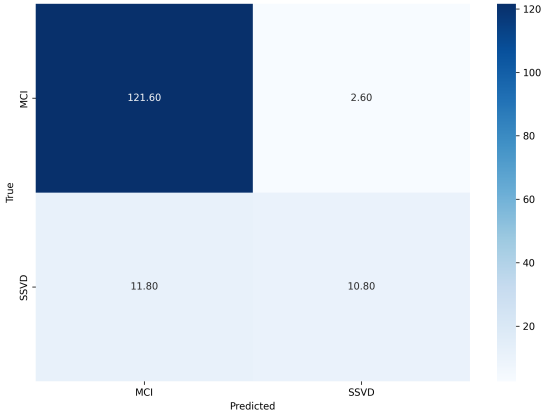
Figure 4.4 shows confusion matrices, averaged over all five cross-validation folds, as a result of classification between SSVD and three other cognitive diseases and healthy control and using the SC dataset. There is a noticeable class imbalance



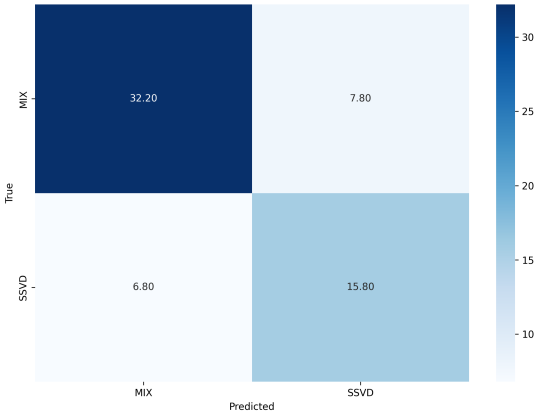
(a) Confusion matrix of AD vs SSVD classification using the SC dataset.



(b) Confusion matrix of HC vs SSVD classification using the SC dataset.



(c) Confusion matrix of MCI vs SSVD classification using the SC dataset.



(d) Confusion matrix of MIX vs SSVD classification using the SC dataset.

Figure 4.4: Confusion matrices showing the results of classification between SSVD and three other cognitive diseases and healthy control, averaged over five cross-validation folds.

Table 4.6: Model Performance Metrics (AD vs SSVD) using SC dataset

Model Name	Accuracy	Precision	Recall	F1	ROC AUC
XGBoost	0.828	0.774	0.762	0.765	0.762
LightGBM	0.852	0.814	0.772	0.789	0.772
RF	0.823	0.789	0.700	0.725	0.700
SVM	0.760	0.693	0.714	0.699	0.714
NB	0.789	0.725	0.688	0.700	0.688

Table 4.7: Model Performance Metrics (MIX vs SSVD) using SC dataset

Model Name	Accuracy	Precision	Recall	F1	ROC AUC
XGBoost	0.724	0.706	0.702	0.689	0.702
LightGBM	0.737	0.716	0.689	0.689	0.689
RF	0.732	0.735	0.694	0.684	0.694
SVM	0.769	0.754	0.775	0.749	0.775
NB	0.66	0.657	0.678	0.642	0.678

between SSVD and all four other classes which is most noticeable in Figure 4.4c where the number of MCI instances is nearly six times higher than the number of SSVD instances. This leads to the model showing signs of biasing towards the majority class due to the low recall for the minority class compared to the high recall for the majority class.

Table 4.8 shows a comparison in model performance when using SMOTE and not using SMOTE. It is evident that SMOTE enhances the classification performance, especially in distinguishing between SSVD and AD and SSVD and MCI. In all of these four scenarios, SSVD is the minority class, which means that it was the class being subjected to oversampling using SMOTE. Due to the improvement in performance using SMOTE by oversampling the SSVD class, this would indicate that SSVD samples are well covered in their local vicinity. As previously observed, the best performing model in the SSVD vs MIX scenario was SVM and this scenario using SVM did not see as much increase in performance as the other scenarios which uses ensemble models. This could indicate that SVM is more robust to class imbalance compared to the ensemble methods used.

4.3 Feature Importance

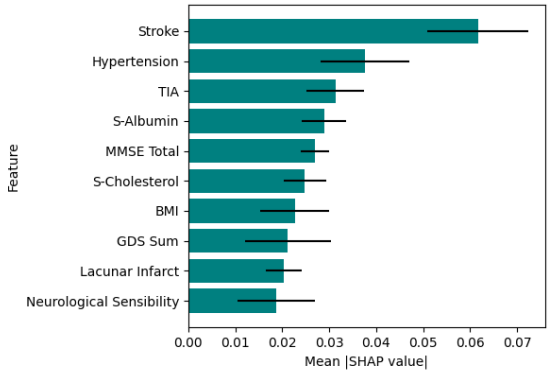
Figure 4.5 illustrates SHAP values for AD vs SSVD classification in all three settings, focusing on the 10 variables with the highest mean absolute SHAP values. The figure shows both the complete distribution of SHAP values for these top 10 variables and their mean absolute SHAP values. Additionally, the graphs on the left depict these mean absolute SHAP values along with their standard deviations across the five cross-validation folds. The graphs on the right side depict the complete distribution of SHAP values for these top 10 variables. Here, a high SHAP value pushes the

Scenario	F1 (SMOTE)	F1 (No SMOTE)	ROC AUC (SMOTE)	ROC AUC (No SMOTE)
SSVD vs AD	0.789	0.739	0.772	0.721
SSVD vs MIX	0.749	0.749	0.775	0.767
SSVD vs HC	0.960	0.958	0.970	0.966
SSVD vs MCI	0.757	0.713	0.725	0.688

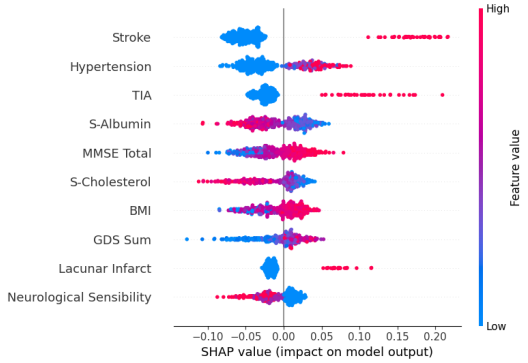
Table 4.8: Results of SMOTE analysis on the SC dataset. The table shows F1 and ROC AUC using the best performing model for each scenario with and without the use of SMOTE during training.

model's decisions towards a positive class classification, and a low SHAP value pushes it towards the negative class. In this example, the positive class is SSVD and the negative class is AD. They also show the value distribution of the variables and their relation to the SHAP values. In this example, we can see that for example, a high value in the stroke variable, indicating if the patient has had a stroke in the past, is related to higher SHAP values and subsequently SSVD (positive class). Note that the SHAP values are not directly comparable across different scenarios (datasets, feature sets, and models). This is because SHAP values measure the impact of an observation relative to a specific model's unique baseline prediction, the learned hyperparameters, and the particular dataset on which it was trained.

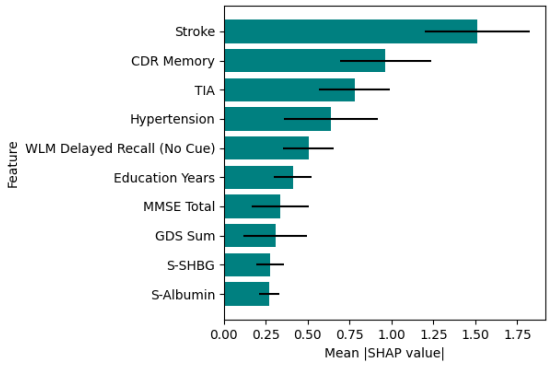
Table 4.9 and Table 4.10 show the model performance of using the entire feature set and only the top 10 features with the highest mean absolute SHAP value, respectively. These tables show how similar the performance is when the models rely on only the top 10 features, as opposed to the entire feature set. Most metrics demonstrate only marginal differences across all settings. For example, the accuracy for the SC setting, decreases minimally from 0.852 to 0.844 when using the top 10 features. While the accuracy decreases in all three settings, the F1 and ROC AUC scores are maintained or even improved when using only the top 10 features. Table 4.11 show the model performance when using the top 30 features with the highest mean absolute SHAP value. While performance using the PC dataset remains roughly the same, performance using the SC and especially the RE dataset increases dramatically, which could be explained by the reduction in complexity as less features are used compared to using the entire feature set. Another unexpected result is that the best performing model when using only the top 10 features and the PC and SC dataset was NB, which could indicate a relatively strong independence between these top 10 features.



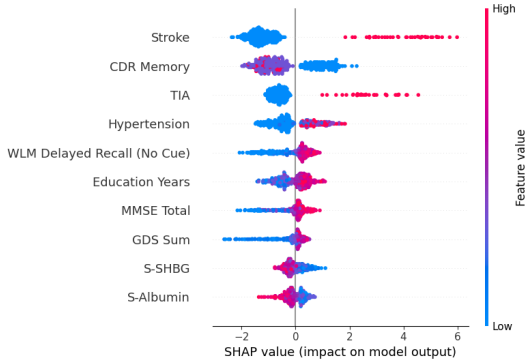
(a) PC Setting: Mean absolute SHAP values for the top 10 features.



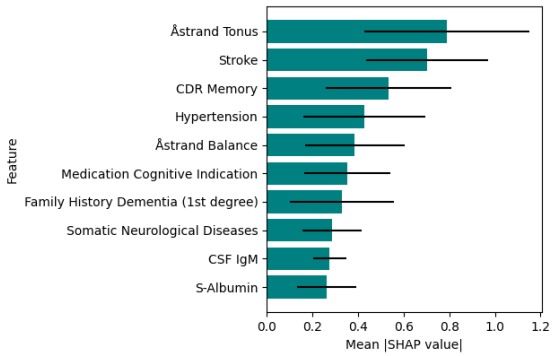
(b) PC Setting: SHAP summary plot for the top 10 features.



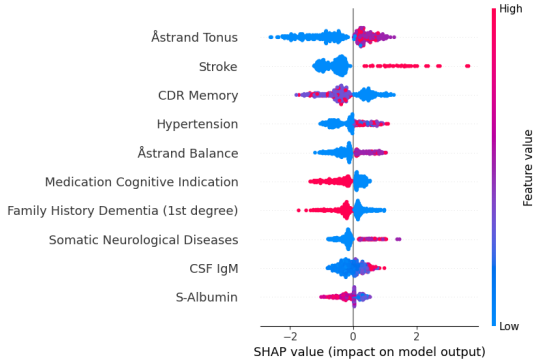
(c) SC Setting: Mean absolute SHAP values for the top 10 features.



(d) SC Setting: SHAP summary plot for the top 10 features.



(e) RE Setting: Mean absolute SHAP values for the top 10 features.



(f) RE Setting: SHAP summary plot for the top 10 features.

Figure 4.5: SHAP analysis for AD vs SSVD classification highlighting the top 10 most discriminative features for the PC, SC, and RE settings. Left panels (a, c, e) display the mean absolute SHAP values for these features, with error bars indicating standard deviations across five cross-validation folds. Right panels (b, d, f) present SHAP summary plots, illustrating the distribution of SHAP values for each feature and how higher (red) or lower (blue) feature values impact the model's prediction towards SSVD (positive class) or AD (negative class).

Setting	Accuracy	Precision	Recall	F1	ROC AUC
PC	0.817	0.778	0.692	0.714	0.692
SC	0.852	0.814	0.772	0.789	0.772
RE	0.819	0.770	0.750	0.748	0.750

Table 4.9: Performance metrics on AD vs SSVD classification using the entire feature set.

Setting	Accuracy	Precision	Recall	F1	ROC AUC
PC	0.837	0.798	0.754	0.770	0.754
SC	0.844	0.807	0.770	0.784	0.777
RE	0.813	0.747	0.763	0.751	0.763

Table 4.10: Performance metrics on AD vs SSVD classification using only the top 10 features with the highest mean absolute SHAP values.

Setting	Accuracy	Precision	Recall	F1	ROC AUC
PC	0.840	0.808	0.741	0.764	0.741
SC	0.851	0.800	0.802	0.800	0.802
RE	0.872	0.827	0.837	0.829	0.837

Table 4.11: Performance metrics on AD vs SSVD classification using only the top 30 features with the highest mean absolute SHAP values.

5

Conclusion

This chapter discusses the findings of the study in relation to the research questions, evaluates the methodologies employed, and considers the implications for clinical practice and future research. It concludes by summarizing the key contributions of this work.

5.1 Discussion

This study aimed to develop machine learning models capable of classifying subcortical small vessel disease (SSVD) with high precision and interpretability, identify significant diagnostic variables, and optimize machine learning pipelines for handling real-world complex clinical data. The research successfully demonstrated that machine learning models, especially ensemble methods like LightGBM, XGBoost and Random Forests, can effectively classify SSVD against other cognitive impairments and healthy control (HC). For instance, RF achieved a high F1 score of 0.96 when differentiating SSVD from HC using the specialist care dataset, and LightGBM a F1 score of 0.789 against AD. Support Vector Machines also showed promise, particularly in SSVD vs MIX classification.

In some cases, such as the SSVD vs MCI scenario using the SC data set as shown in Table 4.5, the performance worsened due to low recall for the minority class (SSVD) which can be attributed to the class imbalance present. When one class, such as SSVD, is heavily underrepresented, models, even with techniques such as SMOTE during training, can struggle to sufficiently learn the full spectrum of the minority class or may learn to be conservative in predicting it to avoid missclassifying the majority class. This often leads to a high number of false negatives for the minority class which lowers its recall. To mitigate this and improve the model performance on the minority class, SMOTE is a strong approach, which is evident in Table 4.8. Other strong approaches include variants of SMOTE such as Borderline-SMOTE which only oversamples minority class instances along the class boundary [29]. Undersampling the majority class is another approach to mitigate majority bias, however, this method can lead to a loss of important information if not done carefully. Although these methods can help mitigate model bias, the most effective and real-world solution would be to collect more instances of the minority class.

The variability observed in the best performing model across these different diagnostic comparisons such as RF for HC, LightGBM for AD, and SVM for MIX all when

contrasted with SSVD is noteworthy. This is likely due to the variability of the underlying data characteristics in different diagnostic pairings. For example, distinguishing SSVD from HC, where cognitive and biomarker profiles are expected to be more distinct, might favor models such as RF that are great at identifying and learning these clear separating boundaries between the classes. In contrast, distinguishing between SSVD and MIX which inherently involves overlapping pathologies, might benefit from the margin-maximizing capabilities of SVMs in specific feature spaces. This observations implies that a universally optimal machine learning model may not exist for all SSVD-related diagnostic challenges. Instead, future clinical decision support systems could be more effective if they employed a range of specialized models tailored to specific diagnostic challenges or an adaptive model which could adapt to the specific diagnostic challenge requested by adjusting its internal mechanism.

The use of SHAP as an XAI approach successfully identified variables that are known to contribute to SSVD diagnosis, some that are known to contribute to other diagnoses, and some that are not known to contribute to any of these diagnoses. Across different clinical settings in AD vs SSVD classification, variables such as stroke, hypertension and TIA consistently emerged as important discriminators which are all generally known to be associated with SSVD and is backed up by previous research in the field [30], [31]. Other identified variables such as IgM are not generally known to be associated with either AD or SSVD which prompts further research into the effect of these variables in SSVD diagnosis. To our knowledge, no associations of IgM and SSVD has been reported although low levels of IgM has been associated with stroke [32]. Interestingly, accumulation of IgM has been observed in in cutaneous small vessel vasculitis, a disease that affects the vessels in the skin [33]. The ability to identify key variables is crucial for enhancing the clinical utility and trustworthiness of the models, as it provides clinicians with insights into the factors driving the diagnostic classifications. Retraining the models using only the top 10 most influential variables resulted in only marginal performance differences, and in some cases, using the top 30 variables even improved performance which suggests a potential for more focused and efficient diagnostic tools which uses only a subset of variables.

The study systematically addressed optimizing machine learning pipelines for heterogeneous and missing data. This was done by, among other methods, comparing KNN imputation and MICE and adjusting row and column elimination thresholds to maximize the number of rows and columns retained after deletion while having a reasonable imputation performance. The results showed a superior performance for the KNN imputation method with clipping to the nearest category for categorical imputed values compared to using MICE, also with clipping. This was evidenced by the higher average KS-statistic across all columns and the individual KS-statistics and well preserved distributions for the top 6 variables with highest missing value levels. Additionally, the KNN imputed dataset for the SC setting performed better in SSVD classification tasks compared to the MICE imputed dataset. A reason for KNN imputation performing better than MICE could be due to the non-parametric nature of the KNN imputation technique compared to the use of parametric regression models in MICE. Clinical datasets often contain complex, non-linear relationships

between variables which KNN imputation can adapt to without making strong assumptions about the underlying data distributions. On the other hand, MICE relies on parametric regression models to capture the complexity of the underlying data, which, if unable to do so adequately, can result in less accurate imputations compared to KNN imputation. The custom KNN imputation method for categorical features, while theoretically sound, did not outperform the standard KNN approach with clipping. The reason for this could be that some of the categorical variables are also ordinal, which means that the order of the categories have meaning and taking the mode does not respect this order. A solution for this would be to split categorical variables into ordinal and non-ordinal variables and use the mode only for the non-ordinal variables although this was not explored in this study. The robust handling of missing data alongside techniques like SMOTE for class imbalance and stratified group k-fold cross-validation for robust evaluation, represents a significant methodological strength of this research.

The main strengths of the employed methodology include the use of real-world clinical data from the Gothenburg MCI study, the comprehensive comparison of different imputation techniques, the application of various machine learning algorithms, and the incorporation of XAI to ensure model explainability. Some weaknesses must be acknowledged however. The data originated from the same region, which may limit the generalizability of the results to a broader population. Similarly, the relatively low amount of SSVD instances compared to other cognitive impairments may also hinder model generalizability. It is also important to note that, while the top variables with the highest mean absolute SHAP values are effective at explaining the discriminative ability of these variables for the specific classification model, it is not certain these can be generalized to the real world. The assumption of MAR for missing data, while supported by observed patterns and domain knowledge, is an assumption. Additionally, while the custom KNN imputation technique for categorical variables was explored, its performance indicated a need for further refinement.

The potential for future clinical implementation of these findings is considerable. The developed models could serve as decision support tools for clinicians, aiding in distinguishing SSVD from other cognitive impairments. The identification of key diagnostic variables can also guide clinical assessment and potentially highlight diagnostic variables for SSVD. Despite this potential, practical challenges include the need for validation on larger, more diverse datasets to ensure generalizability. Integrating such AI tools into existing clinical workflows would require careful consideration of ethical implications and the development of user-friendly interfaces that can be adopted by clinicians. This study did not delve into a detailed ethical or regulatory analysis which would be crucial for real-world development.

5.2 Conclusion

The study successfully demonstrated the utility of machine learning techniques in developing precise and interpretable diagnostic models for SSVD using real-world clinical data. The main contribution of the study lie in the effective classification of SSVD from other cognitive impairments, the identification of variables which are

both known as being strongly indicative of SSVD and potentially other candidate variables, and a systematic approach to optimizing data pre-processing, particularly for handling missing values. The results underscore the potential for machine learning, augmented by XAI to enhance diagnostic accuracy in congitive diseases.

Bibliography

[1] A. Wallin, A. Nordlund, M. Jonsson, *et al.*, "The Gothenburg MCI study: Design and distribution of alzheimer's disease and subcortical vascular disease diagnoses from baseline to 6-year follow-up," *Journal of Cerebral Blood Flow Metabolism*, vol. 36, no. 1, pp. 114–131, 2016. DOI: 10.1038/jcbfm.2015.147. [Online]. Available: https://journals.sagepub.com/doi/full/10.1038/jcbfm.2015.147.

[2] P. Kettunen, *Vaskulär demens. Boken om demenssjukdomar*. Liber, 2023, ch. Vaskulär demens.

[3] A. Ritter and J. A. Pillai, "Treatment of vascular cognitive impairment," *Current Treatment Options in Neurology*, vol. 17, no. 8, p. 35, 2015. DOI: 10.1007/s11940-015-0367-0. [Online]. Available: https://link.springer.com/article/10.1007/s11940-015-0367-0.

[4] L.-O. Wahlund, F. Barkhof, F. Fazekas, *et al.*, "A new rating scale for age-related white matter changes applicable to MRI and CT," *Stroke*, vol. 32, no. 6, pp. 1318–1322, 2001. DOI: 10.1161/01.str.32.6.1318. [Online]. Available: https://www.ahajournals.org/doi/10.1161/01.str.32.6.1318.

[5] I. Kononenko, "Machine learning for medical diagnosis: History, state of the art and perspective," *Artificial Intelligence in Medicine*, vol. 23, no. 1, pp. 89–109, 2001. DOI: 10.1016/S0933-3657(01)00077-X. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S093336570100077X.

[6] A. S. Alatrany, W. Khan, A. Hussain, H. Kolivand, and D. Al-Jumeily, "An explainable machine learning approach for alzheimer's disease classification," *Scientific Reports*, vol. 14, p. 2637, 2024. DOI: 10.1038/s41598-024-51985-w. [Online]. Available: https://doi.org/10.1038/s41598-024-51985-w.

[7] J. E. Dobson, "On reading and interpreting black box deep neural networks," *International Journal of Digital Humanities*, vol. 5, pp. 431–449, 2023. DOI: 10.1007/s42803-023-00075-w. [Online]. Available: https://link.springer.com/article/10.1007/s42803-023-00075-w.

[8] J. Massey F. J., "The kolmogorov-smirnov test for goodness of fit," *Journal of the American Statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[9] Wikipedia contributors, *Kolmogorov–smirnov test — Wikipedia, the free encyclopedia*, [Online; accessed 6-May-2025], 2025. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Kolmogorov%E2%80%93Smirnov_test&oldid=1286241187.

[10] D. E. Knuth, *The Art of Computer Programming, Seminumerical Algorithms*, 3rd ed. Reading, Massachusetts: Addison-Wesley, 1998, vol. 2.

[11] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*, 2nd. Hoboken, NJ: John Wiley & Sons, 2002.

[12] S. M. Memon, R. Wamala, and I. H. Kabano, "A comparison of imputation methods for categorical data," *Informatics in Medicine Unlocked*, vol. 42, p. 101 382, 2023, ISSN: 2352-9148. DOI: https://doi.org/10.1016/j.imu.2023.101382. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352914823002289.

[13] O. Troyanskaya, M. Cantor, G. Sherlock, *et al.*, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.

[14] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: What is it and how does it work?" *International Journal of Methods in Psychiatric Research*, vol. 20, no. 1, pp. 40–49, 2011. DOI: 10.1002/mpr.329.

[15] A. Z. Alruhaymi and C. J. Kim, "Why can multiple imputations and how (mice) algorithm work?" *Open Journal of Statistics*, vol. 11, no. 5, pp. 759–777, 2021. DOI: 10.4236/ojs.2021.115045. [Online]. Available: https://www.scirp.org/journal/paperinformation?paperid=112455.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002, Originally published in 2002; available on arXiv:1106.1813. [Online]. Available: https://arxiv.org/abs/1106.1813.

[17] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. DOI: 10.1023/A:1010933404324.

[18] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001. DOI: 10.1214/aos/1013203451.

[19] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, ACM, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785.

[20] G. Ke, Q. Meng, T. Finley, *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., 2017, pp. 3149–3157. [Online]. Available: https://papers.nips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

[21] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995, ISBN: 978-0-387-94559-0. DOI: 10.1007/978-1-4757-2440-0.

[22] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press, 2012, ISBN: 978-0-262-01802-9.

[23] IBM, "What is explainable ai?," 2024, Accessed: 2025-04-08. [Online]. Available: https://www.ibm.com/think/topics/explainable-ai.

[24] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017. [Online]. Available: https://proceedings.neurips.cc/

paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[25] H. W. Loh, C. P. Ooi, S. Seoni, P. D. Barua, F. Molinari, and U. R. Acharya, "Application of explainable artificial intelligence for healthcare: A systematic review of the last decade (2011–2022)," *Computer Methods and Programs in Biomedicine*, vol. 226, p. 107 161, 2022, ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2022.107161. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260722005429.

[26] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[27] SveDem, "Årsrapport svedem 2023," Svenska Demensregistret (SveDem), 2023. [Online]. Available: https://www.ucr.uu.se/svedem/om-svedem/arsrapporter/arsrapporter/svedemarsrapport-2023-2.

[28] J. Miah, D. M. Ca, M. A. Sayed, E. R. Lipu, F. Mahmud, and S. M. Y. Arafat, *Improving cardiovascular disease prediction through comparative analysis of machine learning models: A case study on myocardial infarction*, 2023. arXiv: 2311.00517 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2311.00517.

[29] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing*, D.-S. Huang, X.-P. Zhang, and G.-B. Huang, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 878–887, ISBN: 978-3-540-31902-3.

[30] K. K. Lau, L. Li, U. Schulz, *et al.*, "Total small vessel disease score and risk of recurrent stroke," *Neurology*, vol. 88, no. 24, pp. 2260–2267, 2017. DOI: 10.1212/WNL.0000000000004042. eprint: https://www.neurology.org/doi/pdf/10.1212/WNL.0000000000004042. [Online]. Available: https://www.neurology.org/doi/abs/10.1212/WNL.0000000000004042.

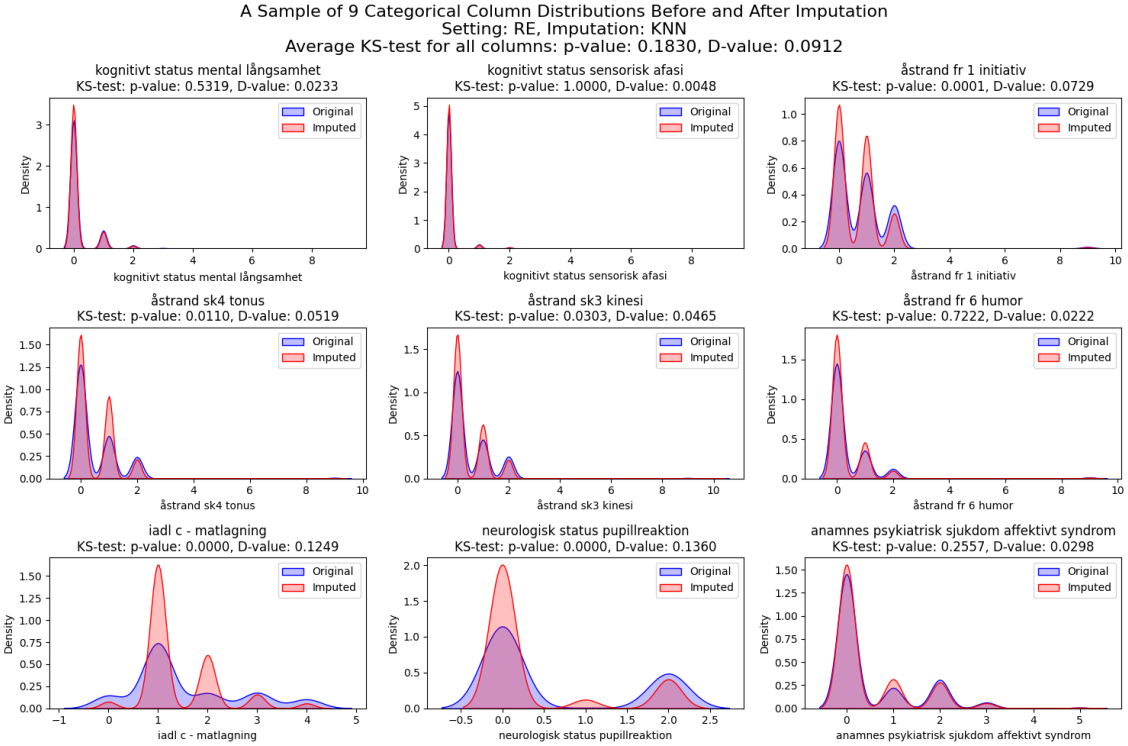
[31] H. M. Abraham, L. Wolfson, N. Moscufo, C. R. G. Guttmann, R. F. Kaplan, and W. B. White, "Cardiovascular risk factors and small vessel disease of the brain: Blood pressure, white matter lesions, and functional decline in older persons," *Journal of Cerebral Blood Flow & Metabolism*, vol. 36, no. 1, pp. 132–142, 2016. DOI: 10.1038/jcbfm.2015.121.

[32] J. Khwaja, S. D'Sa, M. C. Minnema, M. J. Kersten, A. Wechalekar, and J. M. Vos, "IgM monoclonal gammopathies of clinical significance: diagnosis and management," *Haematologica*, vol. 107, no. 9, pp. 2037–2050, 2022. DOI: 10.3324/haematol.2022.280953.

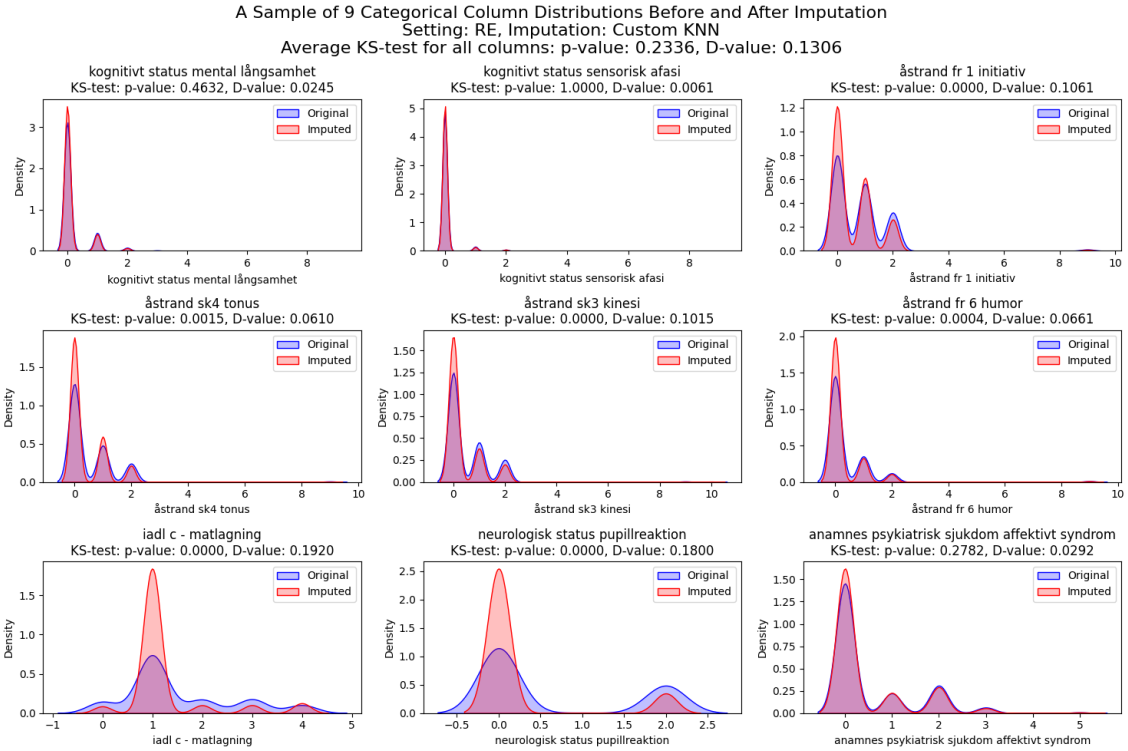
[33] M. Kawamura, Y. Mizutani, Y. Mizutani, *et al.*, "Clinical and pathological differences between skin-limited igm/igg vasculitis and skin-limited iga vasculitis," *Journal of Cutaneous Immunology and Allergy*, vol. 4, no. 2, pp. 28–33, 2021. DOI: https://doi.org/10.1002/cia2.12156. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cia2.12156. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cia2.12156.

A

Appendix 1



(a) KNN with clipping



(b) Custom KNN imputation

Figure A.1: Distributions of a sample of 9 categorical variables in the research dataset, shown before and after imputation using KNN with clipping to nearest category (A.1a) and Custom KNN imputation using the mode for categorical variables (A.1b).