Universiteit Leiden
The Netherlands

Computer Science

Improving Migraine Diagnosis by using Binary and Multi-Class Classification of Machine Learning.

Burak Özdemir

Supervisors:
Dr. M. van Leeuwen & MSc I. Papagianni

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

17/05/2023

Abstract

Machine learning is increasingly applied in various fields such as our environment, social life, education, and healthcare. However, the diagnosis of diseases is still a challenging task. This thesis focuses on improving the diagnosis of migraines using machine learning. To achieve this, we use binary and multiclass classification techniques to classify patients as migraine with aura, migraine without aura, or no migraine. Our data set includes multiple labels and is based on a questionnaire used in a previous study. We aim to find the best model for predicting migraines after the pre-processing of the data and the use of various models such as Random Forest, Logistic Regression, and SVM. Our goal is to minimize false negatives and improve the sensitivity, specificity and accuracy of migraine diagnosis in comparison to the model used in the original study conducted by Oosterhout et al. By comparing the performance of a multiclass classification with a combined binary classification, we aim to improve the overall accuracy of migraine diagnosis for patients. The results show that for all three classifications, three or more models have more accurate predictions than the Extended Questionnaire predictions. In conclusion, it can be stated that the use of machine learning techniques can result in an improved diagnosis of migraines compared to the current algorithm.

Contents

1 Introduction

Automation and the use of artificial intelligence for medical diagnosis have received a lot of attention in the past decade, supported by the increased knowledge and development of technology. This development allows more complicated and multifactorial diseases to be approached that do not have a clear cure or diagnosis [12]. Migraine is an example of this problem, with possible new technological solutions for diagnosis.

There is a growing interest in the medical industry in improving the diagnosis of migraine using machine learning. Machine learning algorithms can create models by analyzing large amounts of data and identify patterns that humans may be unable to discern. These make machine learning algorithms well suited for tasks such as diagnosis, which require the analysis of complex data and numerous other factors [19].

Machine learning has many uses in the field of automation and optimization, which can be applied to the problem of diagnosing headaches. Several studies, for example Oosterhout et al. [26], have been carried out to automate the diagnosis of migraine by patients who have difficulty clarifying the origin of their headaches.

1.1 Challenges and opportunities in the diagnosis of migraine

Headache is a phenomenon that every person feels or has felt at some point in their life. According to the World Health Organization (WHO), last year, 50-75% of adults experienced some form of headache [27]. The reasons for headaches can vary, from too much work pressure to being sick [24]. Headaches are a phenomenon that occurs in the brain, the most complex organ of the body. Due to this fact, examining headaches is not an easy task.

The classification of headache disorders has two main categories: migraine without aura (MO) and migraine with aura (MA). Migraine usually occurs in phases, as shown in Figure 6. The four stages of a migraine attack are prodrome, aura, headache and postdrome. Before the headache, there is a prodrome phase that can involve signs such as mood swings, exhaustion, irritability, stiff neck and changes in appetite or thirst. The aura phase, which occurs only in people suffering migraine with aura, is distinguished by unique neurological symptoms such as sensory or visual problems. Moderate to severe pain, nausea, vomiting, and sensitivity to light and sound are hallmarks of the headache phase. The postdrome phase, which is the last phase and follows the headache phase, is characterized by exhaustion, disorientation, a sense of "brain fog" or changes in mood. The International Classification of Headache Disorders (ICHD) is a system for classifying different types of headache disorders, including migraines with and without aura, which takes into account the patient's medical history, physical examination, and imaging studies [20].

The criteria for diagnosing a patient with migraine (with and without aura) can be supported by the diagnostic criteria mentioned in Figure 4, based on the ICHD-3 criteria. The figure shows that migraine with aura is distinguished by the presence of particular neurological symptoms, such as sensory disturbances (tingling or numbness), visual disturbances (flashing lights, zigzag lines, or blind spots), or other neurological symptoms that occur before or during headache. The headache often follows these symptoms, which usually lasts for 5 to 60 minutes. On the other hand, migraine without aura is characterized by recurrent headaches that are frequently accompanied by other symptoms such as nausea, vomiting, and sensitivity to light and sound. These headaches frequently cause moderate to severe pain and linger for 4-72 hours on average. There are no distinct

neurological symptoms that manifest before or during the headache, unlike the aura migraine.

Diagnosis of migraine has been a difficult task since the discovery of this phenomenon, as the cause of this condition cannot be classified. There is no clear biomarker that helps classify the migraine that haunts a patient. There are many possible causes of headaches and it is still easy to mark the differences between the types of headaches. To solve this problem, the International Classification of Headache Disorders (ICHD) has created a list of criteria that defines a patient's type of headache from migraine without aura, migraine with aura, tension headache, cluster headache and some others. But at least 14.7% of the entire world population [25] has symptoms that can lead to any type.

During the current diagnosis of migraine, a physician will ask the patient about his symptoms and search for physical indicators of migraine during a clinical interview and physical exam to determine whether the patient has a migraine. Imaging tests such as CT scans or magnetic resonance imaging can occasionally be used to eliminate other potential explanations of the patient's symptoms [8].

The fact that these techniques are primarily based on the patient's ability to precisely explain their symptoms and the healthcare professional's capacity to understand those symptoms is one of the main drawbacks. This can be difficult because patients may not always remember or be able to describe their symptoms accurately and because migraines frequently have a range of different symptoms that can vary from person to person.

These techniques can also be expensive and time consuming and may not always provide a clear diagnosis. For example, imaging tests might show normal signs for people who suffer from migraine, and clinical interviews might not always produce enough data to reach a reliable diagnosis. As a result, there is a need for more precise and effective techniques to diagnose migraine.

The study by Oosterhout et al. [26] focuses on improving the medical diagnosis of migraine, which currently needs a neurologist to physically examine. When the questionnaire used in the study [26] is completed, the algorithm based on the ICHD-2 criteria was run on each diagnosis to classify the patients. The algorithm had the possible outcomes: no migraine, migraine with aura and migraine without aura. They successfully diagnosed migraine aura in self-reported migraineurs with a low false positive rate, but only 80% of the cases were correctly diagnosed. This method was based on the ICHD-2 criteria, which has already been updated to the ICHD-3 criteria showing much better specificity and sensitivity (96% and 98%, respectively) [11]. Unfortunately, the application of this study [26] has limitations and errors in the performance that may lead to falls diagnosis. For example, the sample size of the study is relatively small, including a total of 2397 participants, of which 1067 were selected to continue in the study, which limits the generalizability of the findings. In addition to that, the study is based on self-reported data, self-reported migraineurs, which can be affected by recall bias and social desirability bias. Similarly, Oosterhout et al. refers to them as the 'virtual Munchhausen syndrome'. The virtual Munchhausen syndrome refers to instances where individuals self-refer to studies for which they do not meet the eligibility criteria, which can potentially compromise the validity of the study's results. [26] These limitations suggest that the study [26] can be improved.

1.2 Aim and objectives

Our study aims to explore and improve the algorithms that Oosterhout et al. [26] has implemented, using machine learning as the main tool to correctly classify patients whether they have migraine

with aura or without aura, or no migraine. We try to achieve this by using classification methods and create a predictive model based on the data set consisting of responses to the questionnaire used in [26]. The data set is enhanced with labels that will be used to conduct supervised learning in which training and testing of the algorithms are performed.

We will begin by pre-processing our data to enable us to perform multiple classification tasks on it. Our initial model will aim to predict whether a patient has migraine or not. Following this, we will perform a second binary classification to determine if a migraine patient has migraine with or without aura. This approach will enable us to divide our labels into two sub-labels, each of which will be used for binary classification tasks. We will also conduct a multiclass binary classification and compare its sensitivity, specificity and accuracy with our previous model. Our objective is to compare the performance of a multiclass classification with a combination of binary classification tasks.

In order to accomplish this, we will incorporate several additional models such as Random Forest, XGBoost, SVM and others on our dataset to evaluate their performance in comparison to our initial binary classification models. Our ultimate goal is to identify the best model that can accurately predict migraines with the least number of false negatives, compared to the model used in previous studies [26].

How can machine learning improve the sensitivity and specificity of a model for predicting migraine patients compared to the classification performed in the study by Oosterthout et al. [26]?

The following sections of this thesis are organized as follows. In Chapter 2, we review the relevant literature on the use of machine learning for migraine diagnosis, including past studies that applied machine learning methods. The vocabulary used in machine learning and migraine-related topics will be explained in Chapter 3, along with the definition of the issue and potential pitfalls. In Chapter 4, we give a detailed explanation of our research procedures and resources, including the dataset, machine learning strategies, and evaluation metrics. Chapter 5 of the thesis presents the results of our investigation, including the potency of the examined machine learning models. Finally, Chapter 6 covers the implications of our findings for the diagnosis of migraines and future research possibilities are discussed in this chapter.

1.3 Approach and contributions

First, our research expands the existing literature on the use of machine learning techniques to improve disease diagnosis. Specifically, we will focus on the application of these techniques to the diagnosis of migraine.

Second, this study examines the use of a binary classification in a multiclass problem by dividing the problem into two binary classification problems. We evaluate both models to conclude if our approach improves the predictions compared to the results of the previous study. We aim to explore how using two binary classification tasks compares to using a multiclass classification. We will perform a multiclass classification on the same data after performing our initial binary classification. This will highlight the variations and perhaps indicate the approach that performs the best.

Third, we attempt to use multiple machine learning models and evaluate each of them on accuracy, sensitivity and specificity. This will support our final conclusion on which model performs the best.

2 Related Work

As mentioned above, the study [26] has been conducted to improve the diagnosis of migraine. This method was based on the ICHD-2 criteria, which has already been updated to the newer ICHD-3 criteria, showing much better specificity and sensitivity (96% and 98%, respectively) [11].

2.1 The questionnaire by LUMINA

2.1.1 The SCREEN

To assist with early screening of headache patients, the LUMC Leiden Headache Group (LUMINA) has created an online questionnaire with questions based on ICHD-3 criteria. The screening flow for patients in the LUMINA program consists of two parts. The first part is a short screening questionnaire (SCREEN) consisting of 32 questions, for example, if the patient had severe headaches in the past 12 months and what the severity of the headache was. [26] The purpose of SCREEN is to screen the participant for migraine, tension-type headache, and cluster headache. The group of participants who potentially have migraine is forwarded to the second part of the screening. Of the 32 questions asked during SCREEN, 9 questions are used in the ICHD-2 algorithm, which has the objective of classifying probable migraine patients. The algorithm used in the study is based on a set of conditional logics that works its way through the questionnaire.

A previous project by a student about the LUMINA questionnaire resulted in a 'Mini Screen' that is an abbreviated version of the original LUMINA SCREEN questionnaire. The Mini Screen questionnaire, which is a shortened version of the original questionnaire, was created using the results of the Lasso regression analysis. The validity of the Mini Screen was then evaluated using diagnostic factors such as sensitivity and specificity by applying a confusion matrix. Furthermore, this research also investigated three methods to reduce the lifetime depression screening process [13].

The results of this study showed that the Mini Screen questionnaire includes eight questions that have a high predictive value for migraine. The sensitivity and specificity of the Mini Screen were found to be 89% and 84% respectively, compared to 95% and 35% for the original questionnaire. The advantages of this study include a large study population and an accurate "golden standard" for migraine diagnosis. However, limitations include the lack of depression diagnoses made by a physician in the second part of this research [13]. However, using more complex machine learning algorithms would potentially result in more accurate predictions for the original SCREEN.

2.1.2 The Extended Questionnaire

The second part of the LUMINA questionnaire is the Extended Questionnaire (EQ), which incorporates an additional 127 questions on migraine headache and aura characteristics. Compared to SCREEN, the extended questionnaire has more detailed questions about the characteristics of the patient, migraine headaches and aura characteristics. The purpose of the extended questionnaire is to subpart the list of potential migraine patients to patients with migraine and patients without migraine. The migraine group is further specified into patients who have migraine with aura (MA) and migraine without aura (MO).

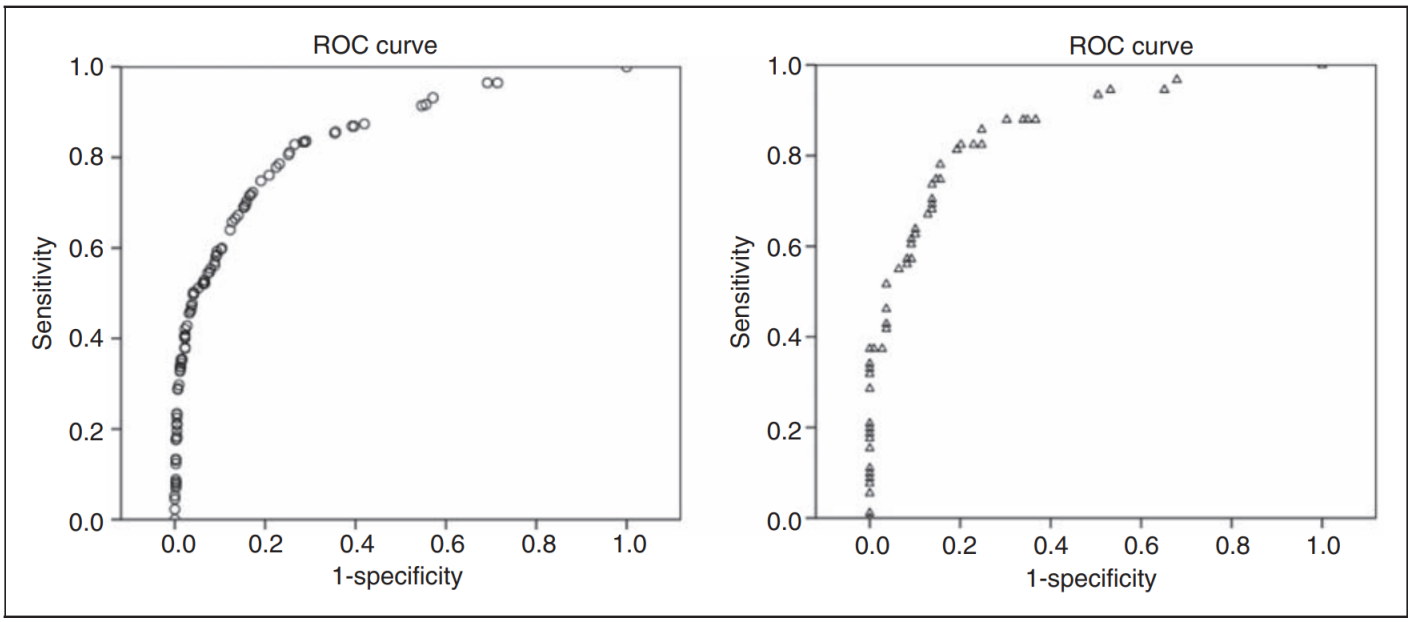


Figure 1: The study created a Receiver Operating Characteristic (ROC) curve for the prediction rule in the initial training sample of 838 patients and in the validation sample of 200 patients. The ROC curve is used to evaluate the performance of a binary classifier. The area under the ROC curve (C-statistic) for the prediction rule was 0.85 (95% confidence interval (CI) 0.83-0.88) in the training sample and 0.87 (95% CI 0.82-0.92) in the validation sample. This indicates that the prediction rule was able to accurately predict the outcome in both the training and validation samples [26].

2.2 The study by Oosterhout et al.

Van Oosterhout et al. conducted a validation study on the LUMINA questionnaire with a sample size of N = 2397 participants. Participants in this study can be divided into two subgroups, participants forwarded by the General Practitioner (GP) and participants who join voluntarily. The group that is forwarded by the GP has a clinical diagnosis, which we will use as our golden standard for evaluating the predictions generated by the algorithms. The study process is illustrated in Figure 3, which will be explained in more detail later.

From the sample size of N = 2397, 1067 participants are randomly selected to be used in the study [26]. These participants were first contacted by telephone and interviewed for 10 - 15 minutes with detailed questions about the headache and aura criteria. Based on their answers to detailed questions about headaches, a diagnosis is made for each participant and they are invited to a digital form containing the EQ. The questions in the EQ are multiple choice. After completion of the EQ, the answers are fed to the algorithm based on the ICHD-2 migraine criteria, which is an older version of the ICHD-3 algorithm. From this algorithm individual diagnoses are predicted and returned as no migraine, migraine without aura or migraine with aura [26]. The predictions made by the algorithm also need to be validated. Oosterhout et al. split the data set into a sample group (80% of the data, N = 838) and a validation group (20% of the data, N = 200). The sample group is used to create a model based on logistic regression, which contains subcategories that predict whether there is aura or not. The validation group is then used to check whether the predictions made by the model are significant ($p < 0{,}20$). Sensitivity and specificity are used as metrics to

evaluate the performance of a model.

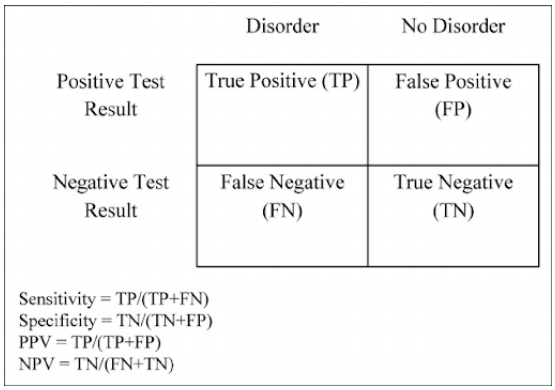


Figure 2: Table describing the results and formula to calculate sensitivity and specificity. [21]

Sensitivity is the proportion of true positive cases (people with the disease who test positive) out of all cases that actually have the disease. It is calculated as follows: Sensitivity = (True positive) / (True positive + False negative). Specificity is the proportion of true negative cases (people without the disease who test negative) out of all cases that actually do not have the disease. It is calculated as follows: Specificity = (True negatives) / (True negatives + False positives), also shown in Figure 2.

The study then determined the best cutoff point for the prediction score using a Receiver Operating Characteristic (ROC) curve. The study identified the point on the ROC curve that had the highest sensitivity and specificity and then used that information to establish the ideal cut-off point, as shown in Figure 1

Unfortunately, the application of this study has limitations, as mentioned in Chapter 1.1, which may lead to falls diagnosis.

2.3 Other studies

Another study by Kwon et al. [15] used more complex machine learning models, one of the first to use machine learning with the ICHD-3 criteria, to automate headache classification. It achieved a more accurate sensitivity and specificity of 88% and 94%, respectively, compared to the study [26], which used the updated ICHD-3 criteria to make diagnoses using a questionnaire with a sample size of n=2000. The interesting part of the study is the XGBoost classifier consisting of four layers, each layer being a type of headache. This resulted in greater sensitivity and specificity for the prediction of migraine. However, the diagnosis of other types of headaches was less significant, with scores less than 50%. One of the limitations of this study was again the sample size, which was not sufficient to classify other types of headaches as accurately as migraine.

Our study has a much larger sample size of n=4000, which could mean that we can count on a better result when using machine learning to predict migraine diagnosis.

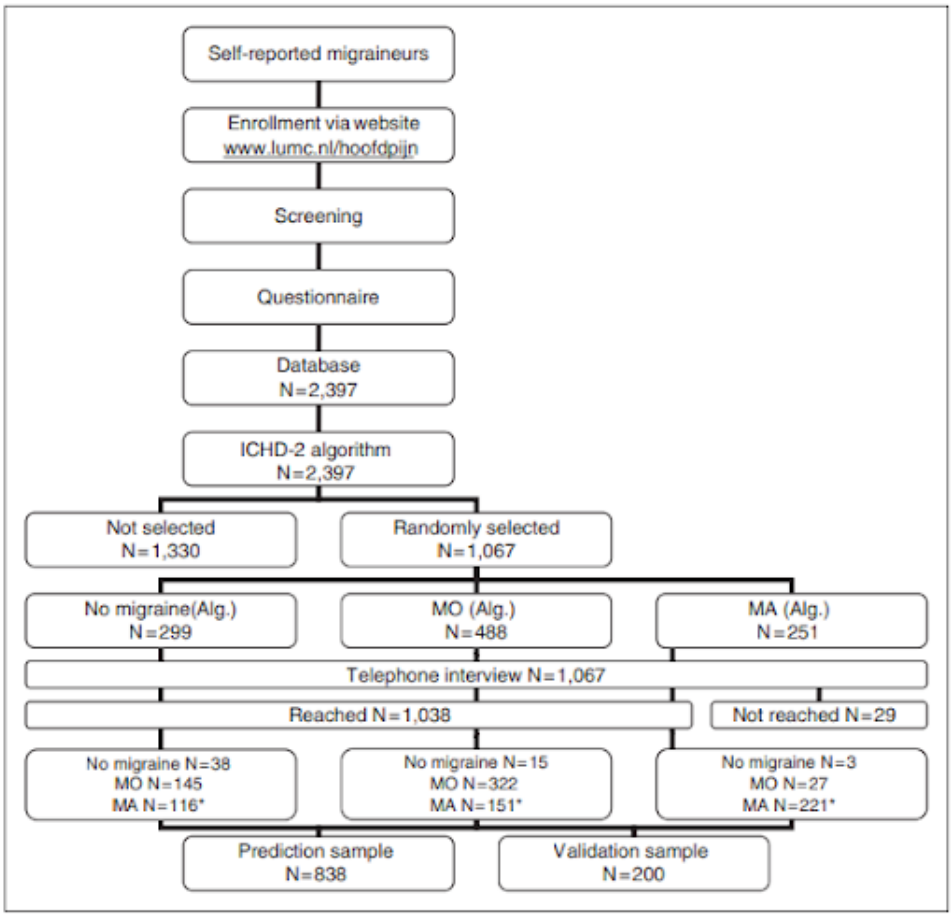


Figure 3: Study flow conducted by van Oosterhout et al. using the SCREEN and extended questionnaire [26].

3 Migraine with machine learning

3.1 Machine Learning

Machine learning refers to the use of algorithms that allow computers to improve their performance on a task. By means of Artificial Intelligence (AI), people are able to train'models' with historical data to make accurate predictions for a problem. Classification is the process of determining, depending on specific characteristics or traits, to which category or group an object or data point belongs. In contrast, predicting is the act of analyzing data to produce forecasts or projections about upcoming events or results. The goal of every classifier is to generalize well from a training data set to some unseen test data set with a low bias and low variance. However, there are some obstacles during this process, called overfitting. Overfitting occurs when a model is too well trained on a given dataset. Figure 5 visualizes this issue. We can see that in the graph of 'overfitting', the red curve, representing the model, goes through each data point. This model is too fitted to the training data and is unable to generalize well to new data. A model that cannot be generalized will not classify or predict, as is intended to do [28]. The variance, spread of all data points to the unseen data, is too high, and the bias too low. To limit the overfitting of a model, we use a

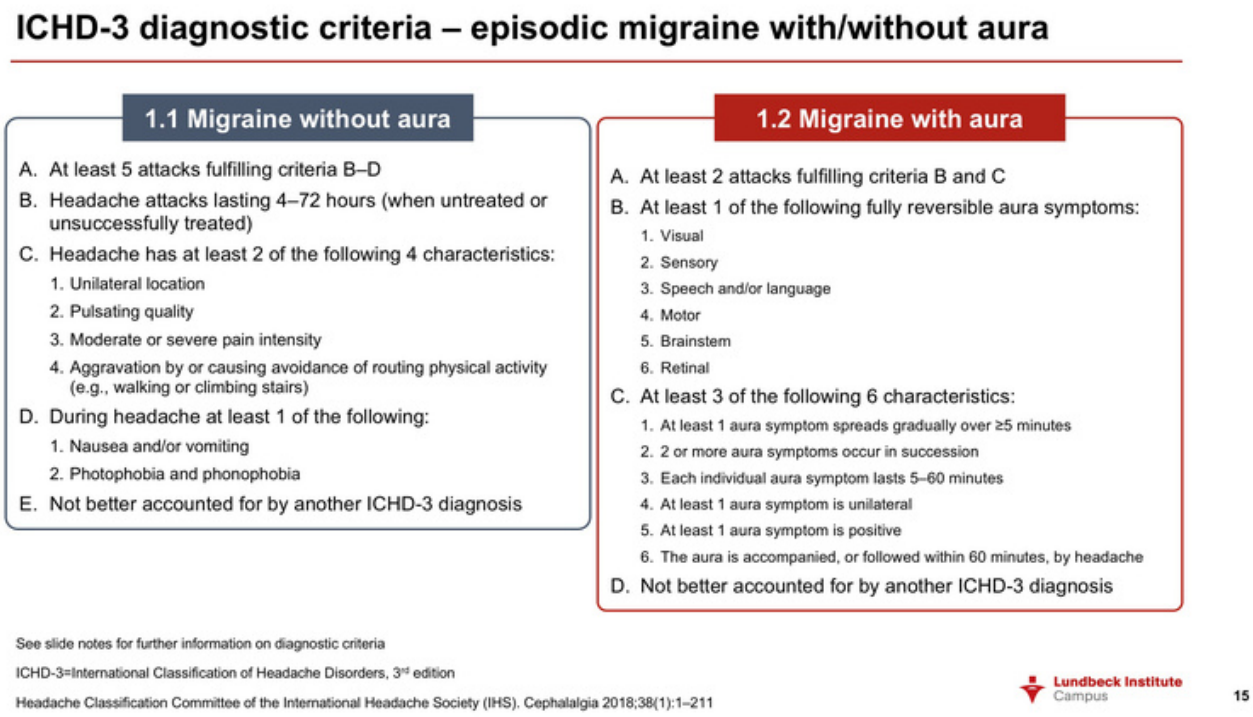


Figure 4: ICHD-3 diagnostic criteria migraine with/without aura [6].

resampling technique called k-fold cross-validation, which makes multiple copies of the training set and divides it into a train and validation set. This method allows one to estimate the accuracy of the model on unseen data [5].

Classification techniques are often used to build models through training. To understand how it operates and interpret its output, the final model should be both understandable and accurate. Although models with better accuracy are typically easier to interpret, they could also be more complex. Complex models can have a greater variety of parameters changed during training to improve performance. This increase has a lower error rate during training with the dataset that is used for it [10]. The trade-off between interpretability and complexity, shown in Figure 7, is visualized for common algorithms used in classification. When an algorithm has a high interpretability, like Decision Trees which are more interpretable, there is a decrease in the ability to learn complex patterns in the data. In contrast to this, algorithms such as support vector machines and neural networks, which are more complex models, will learn complex patterns easier but will make it harder to understand the implementation [5]. There is a machine learning theorem called 'No Free Lunch', which implies that there is no single algorithm that outperforms all other algorithms in all problems. That is why studies occasionally run multiple algorithms on a specific problem and compare them to see which suits the best for our problem. A supervised learning technique tries to determine the best linear fit between the dependent variable (the outcome) and the independent variables (all the other features in the dataset).

When selecting a machine learning algorithm using Python libraries, we aim to find the most interpretable and accurate model that can effectively solve our problem. Previous work [15] has employed well-known models to accurately diagnose a range of medical conditions using the following

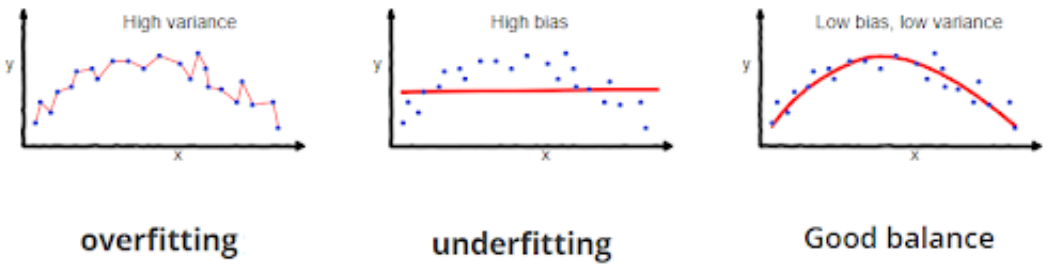


Figure 5: Visualization of overfitting and underfitting of data [4].

machine learning methods:

- **Decision trees** (DT). This method is a type of machine learning algorithm that is used for data classification. These algorithms, which are commonly used for tasks such as diagnosis and prediction, build a tree-like model of decisions based on the characteristics of the data. Their advantages include the ability to handle both categorical and numerical data, the ability to handle multi-output problems, and the ability to be easily interpretable. Disadvantages include their tendency to overfit, particularly with noisy or complex data, and the fact that minor differences between instances in the data can result in a completely different tree. The maximum depth of the tree and the minimum number of samples required to split an internal node are two parameters. Decision trees can be used to classify objects into binary and multiclass categories [2].

- **Support vector machines (SVM)**. Another type of machine learning algorithm used for classification tasks is SVM. They operate by locating the hyperplane in a high-dimensional space that separates different types of data. Advantages include their ability to work with non-linear decision boundaries, high-dimensional data, and infrequent data. The disadvantages include their sensitivity to kernel selection and the need to tune hyperparameters. The kernel (linear, polynomial, or radial basis function) and the regularization parameter C are among the parameters [9].

- **Artificial neural networks (ANN)**. This method is a type of machine learning algorithm that mimics the way the brain functions and is structured. These models consist of a network of interconnected artificial neurons that are capable of receiving, processing, and transmitting information. ANNs are commonly used in various machine learning applications, such as classifying and predicting outcomes [23].

However, other algorithms that will be mentioned and used in our study can be explained as follows:

- **Logistic Regression**. This method is a supervised learning algorithm for problem solving. It is a linear model that employs a logistic function to estimate the probability that a given

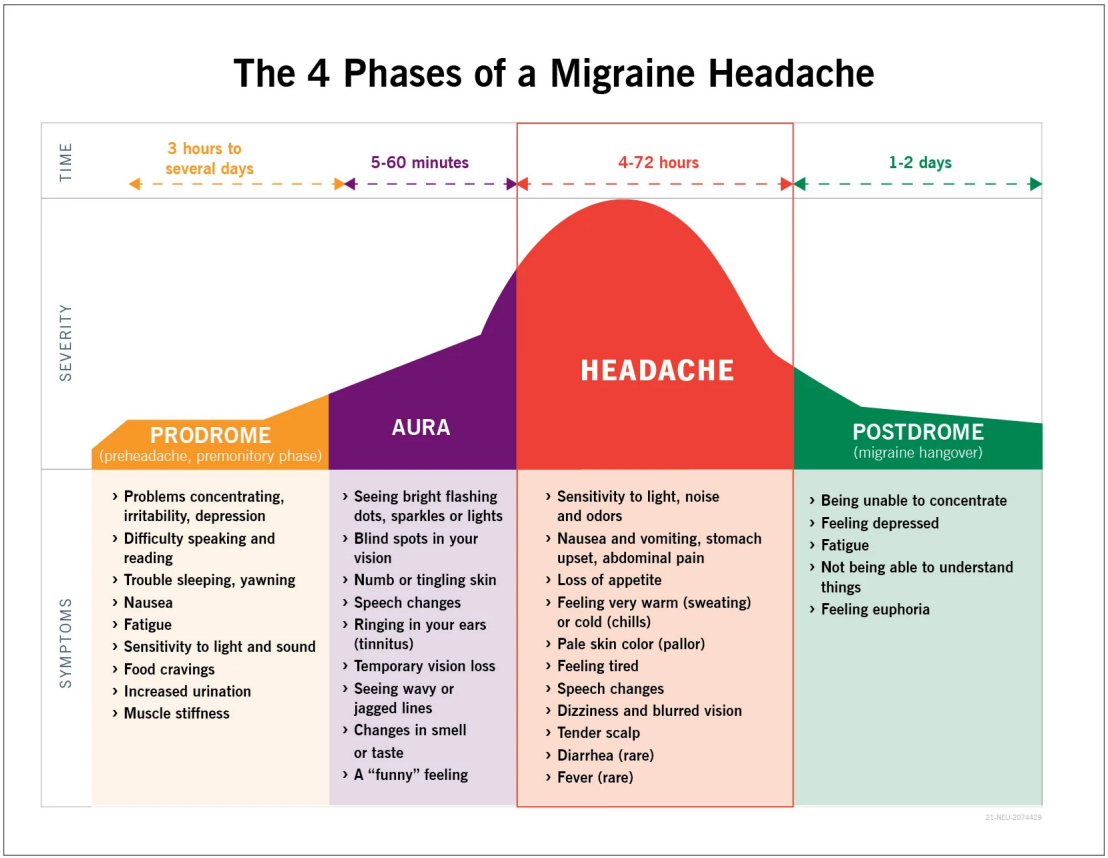


Figure 6: Visualization of the different phases of a migraine attack [1].

input falls into a specific class. The model's benefits include its easy interpretability, good performance on a wide range of problems, and the ability to regularize the model to avoid overfitting. One disadvantage is that it lacks complexity and cannot handle non-linear decision boundaries. Regularization strength and optimization are two parameters that are used in Logistic Regression. The model is applicable to binary and multiclass classification problems [14].

- **Naive Bayes**. This method is part of probabilistic algorithms that apply Bayes' theorem with the 'naive' assumption that every pair of features is independent. Advantages include their simplicity, good performance with high-dimensional data, and their ability to handle continuous and discrete data. Disadvantages include the unrealistic assumption of feature independence and poor performance with small amounts of data. Parameters include the type of distribution used for each feature (e.g. Gaussian or Bernoulli). Gaussian Naive Bayes can be used to classify both binary and multiclass problems [29].

- **Random Forest**. This method is built upon the Decision trees. From a randomly chosen subset of the training set, a set of decision trees is built. The final test object class is then defined by averaging the votes from various decision trees. Decision trees have a propensity to overfit their training set, which Random Forest corrects. Advantages include their ability to handle large data sets with higher dimensionality, their ability to handle unbalanced and

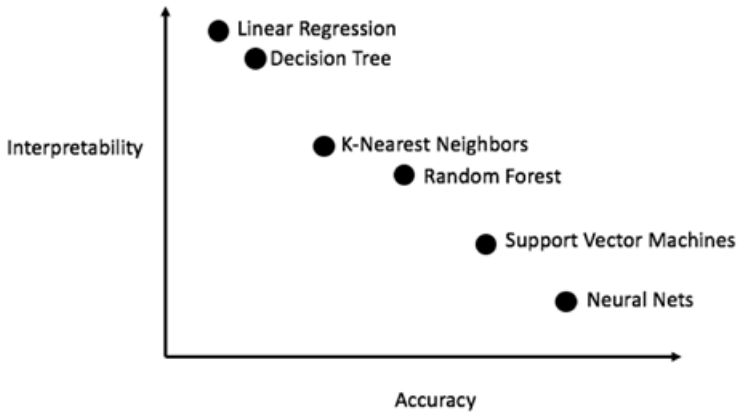


Figure 7: The trade-off between interpretability and accuracy of machine learning Algorithms [18].

missing data, and their ability to estimate feature importance. Disadvantages include their complexity and their longer training time. The parameters are: number of trees in the forest and the number of features taken into account at each split. Random Forest can be used for binary and multiclass classification problems [3].

- **K-Nearest Neighbors (KNN)**. KNN is a nonparametric technique for classification and regression. The closest k training examples in the feature space compose the input. Depending on the kind of issue, the output can either be a class membership or a continuous value. Advantages of KNN include that it is simple to implement, works well with a small number of features, and can handle multiclass problems. Disadvantages include the fact that it can be computationally expensive and sensitive to the scale of the data. The value of k, which determines how many nearest neighbors will be taken into account when making a prediction, is the main parameter when using KNN.

- **XGBoost**. EXtreme Gradient Boosting, is an ensemble learning method for gradient boosting decision trees. Large datasets can greatly benefit from its effective and scalable implementation of the XGBoost algorithm. The algorithm works by iterating to build a decision tree model and adding new models to the set to correct the errors of the previous models. The final ensemble model is a weighted combination of all individual decision trees [7].

  Advantages of XGBoost include its ability to handle large datasets, its high accuracy, and its ability to handle missing data. It also includes a number of regularization parameters that can be used to prevent overfitting.

  Disadvantages include its complexity, which can make it difficult to interpret the results, and the fact that it can be sensitive to the choice of parameters.

  XGBoost can be used for both binary and multiclass classification problems. The parameter "num_class" can be set to 2 for binary classification or to the number of classes in a multiclass problem.

3.2 Problem Statement

Currently, the automatic diagnosis of migraine patients program has a disadvantage. The current model used in the program labels patients who have a clinic diagnosis stating that they suffer from migraine, as nonmigraine patients (Table 1, 'B'). This is also called a "False Negative" and can be explained in Figure 2. The EQ has a sensitivity of 79% and a specificity of 69% [26]. A sensitivity of 79% and a specificity of 69% for the model in the study tells us that the model performs relatively well in identifying patients with migraine (true positive rate) and those without migraine (true negative rate), respectively. A sensitivity of 79% means that the model correctly identifies 79% of migraine patients (true positives) among all patients with the disease (true positives + false negatives). This means that the model is likely to miss around 21% of the patients who actually have migraine. A specificity of 69% means that the model correctly identifies 69% of the patients who do not have migraine (true negatives) among all patients who do not have the disease (true negatives + false positives). This means that the model is likely to incorrectly identify around 31% of the patients who do not have migraine as having the disease. A diagnostic test should have high sensitivity because this reduces the risk of missing a patient who has the condition, but it must also have high specificity, as this reduces the risk of false positive results.

The main question of the project is as follows: Can machine learning improve the classification of probable migraine patients compared to the current algorithm?

Before answering our main question, we need more specification about the performance of the algorithms we will use. We can also investigate if the groups which our dataset consists of, the GP group and the voluntary group, show differences when trained on the model.

So, a secondary objective is: How does the classification of migraine and no migraine compare to the classification of no migraine, MO and MA?

However, a prediction problem remains. Table 1 shows the various possible outcomes of the LUMINA questionnaire and the outpatient diagnosis. Currently, the predictions of the Mini-screen, questionnaire and outpatient diagnosis for groups A and D have shown accurate predictions. If the questionnaire predicts that a patient has migraine, this will also be detected during outpatient diagnosis. However, not every patient who actually suffers from migraine is labeled a possible migraine patient by the questionnaire, group B in Table 1.

Table 1: A table with predictions: migraine or no migraine from the LUMINA-questionnaire and clinic-diagnosis.

		LUMINA-questionnaire	
		Migraine	No migraine
Clinic-diagnosis	Migraine	A	B
	No migraine	C	D

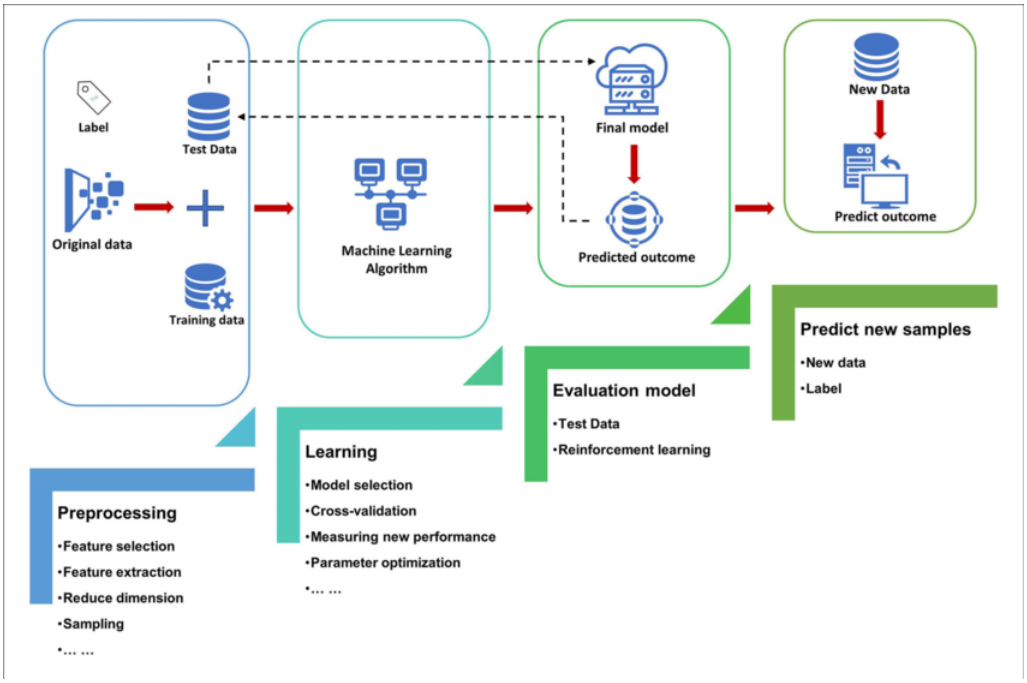


Figure 8: Roadmap for machine learning systems [22].

4 Methodology

4.1 Methods

To carry out our study efficiently and find an answer to our research question, we have a roadmap, as shown in Figure 8, which describes the order in which to carry out the study. The process of creating machine learning models involves several steps, including preprocessing, learning, and evaluation.

- Preprocessing includes tasks such as feature selection and feature extraction, which involve selecting and extracting relevant information from the data to be used in the model.
- Learning involves model selection, cross-validation, and parameter optimization. Model selection is the process of choosing a suitable model for the task at hand. The models that we will use are mentioned in 3.1. Cross-validation is used to evaluate the performance of the model, and parameter optimization is used to adjust the parameters of the model to improve its performance.
- Evaluation of the model is done by testing it on a set of data that was not used in the training process, it is called Test Data. This step is used to evaluate the performance of the model using various evaluation metrics, such as sensitivity, specificity, accuracy, and f1 score. These metrics are used to evaluate how well the model can correctly classify or predict the results of the data.

We will conduct our experiment in two phases. The first phase involves using binary classification to train seven models, as mentioned in the chapter 3.1, and to predict whether patients have migraine

or not. Patients labeled with migraine will then be entered into a second binary classification process, using the same seven algorithms, which will be retrained on data from patients with migraine, this time to predict whether the patient has migraine with aura (MA) or migraine without aura (MO). In the second phase of the experiment, we will use multiclass classification to retrain the seven models, this time with labels: no migraine, MA and MO, for multiple classes. We will compare the performance of the two phases using metrics such as sensitivity, specificity, accuracy, and f1 score to determine whether there is a difference between two binary classification tasks and a single multiclass classification. The clinic-diagnosis will act as the target of the model. We will create the models by first training them in our data set, which will be used to make predictions about the diagnoses of patients. We will use machine learning models, including Random Forest Classifier, Decision Tree Classifier, Support Vector Machine, Logistic Regression, K-Nearest Neighbor, Naive Baye, and XGBoost to train our data. We will train all of the mentioned algorithms, even if some of them perform well on large datasets and some do not. Knowing that some models perform poorly on multiclass classification or some on binary classification tasks, as mentioned in Section 3.1, gives us the ability to examine whether the other models actually perform as promised according to their specifications.

To train our models, we will use k-fold cross-validation, which is a method to evaluate the performance of machine learning models. The idea behind k-fold cross-validation is to divide the data into k 'folds' or subsets, train the model k times and then evaluate it k times, using a different fold as the test set and the remaining k-1 folds as the training set each time. This makes it possible to train the model on a variety of data subsets, making it less susceptible to overfitting. After going through all the folds, we end up with the averaged sensitivity, specificity, accuracy, and f1 score that is averaged over the k-fold (5 times) of the model. This is important as it allows us to check the stability of the model, especially if the data is small or the model's performance varies a lot depending on the training set. By evaluating the model multiple times on different subsets of the data, we can get a better understanding of how well the model will perform on unseen data.

Another important analysis we will conduct is the feature importance analysis for each model in each classification. Feature importance analysis assesses the significance of specific features in a dataset, helping us understand which elements are crucial for predictions or classifications. In our study, we will use feature importance to:

- Enhance the sensitivity and specificity of migraine diagnosis by identifying the most relevant features, we can focus on the data that contribute the most to accurate predictions, ultimately improving our machine learning models' performance in diagnosing migraines.

- Guide model selection by comparing the interpretability and relevance of features across different algorithms, we can better understand their decision-making processes. Models showing low interpretability are less sensible to use in situations that require understanding of the underlying decision making of a model. For example, models used in healthcare should be easy for doctors to read and understand to confirm the accuracy of a prediction. Less interpretable models are not favorable, as not understanding the reasoning of a prediction can result in a fatal outcome if the prediction is not correctly revised by a doctor. Models that have high interpretability are easier to handle in cases of false predictions.

This analysis allows us to compare different machine learning algorithms in terms of their relevance and interpretability by identifying the most relevant features and gaining insight into

each model's decision-making process. The insights gained from feature importance can be applied to feature selection, feature engineering, and the removal of redundant or irrelevant features. Techniques such as Decision Tree Classifier, Random Forest Classifier, Logistic Regression, XGBoost, and Naive Bayes can be used to measure feature importance, while the Support Vector Machine and K-Nearest Neighbors are excluded from this analysis due to the lack of an implemented method in Python.

4.2 Prediction and Evaluation

After training and fitting our model, we will make predictions by running the test set on each model and comparing it with the target, the clinic-diagnosis. Our predictions should match the outcome of the clinic-diagnosis. We will use the formulas in Figure 2 to calculate the sensitivity, specificity, accuracy, and f1 score of each model. The accuracy and f1 score can be calculated using the available method 'accuracy_score' in Python. To evaluate each model, we will use the calculated sensitivity, specificity, accuracy and f1 score of each model and plot these metrics on a bar graph for the seven models that will be used. To compare our models more visually, we will create a confusion matrix, as shown in Table 2 and Table 3, to summarize the results of the predictions. This will make the number of false positives and false negatives clearer. We will show the two matrices for each model and then compare the two matrices to interpret the performance of each model. The model with the highest sensitivity, specificity, accuracy and f1 score will be presented as the best model for our experiment.

Table 2: An example of a confusion matrix is presented to evaluate the accuracy of migraine predictions made by different algorithms. The clinic-diagnosis, taken as the ground-truth value, is compared to the predictions made by the current ICHD-2 algorithm (EQ) and the predictions made by the various models (Prediction). This matrix allows for a clear comparison of the number of false positive and false negative results for each algorithm, providing insight into the performance of each model.

		LUMINA-questionnaire	
		Clinic-diagnosis +	Clinic-diagnosis -
Algorithm	EQ/Prediction +	A/E	B/F
	EQ/Prediction -	C/G	D/H

Table 3: An example of a confusion matrix is presented to evaluate the accuracy of the second binary classification predictions made by different algorithms. The matrix compares migraine predictions with aura (MA) and migraine without aura (MO) with the predictions made by the current ICHD-2 algorithm (EQ) and the predictions made by the various models (prediction). This matrix provides a clear comparison of the number of false positive and false negative results for each algorithm, giving insight into the performance of each model.

		LUMINA-questionnaire	
		MA	MO
Algorithm	EQ/Prediction +	A/E	B/F
	EQ/Prediction -	C/G	D/H

Table 4: Parameters describing the data with A (first binary classification: migraine or no migraine), B (second binary classification: aura or no aura) and C (multiclass classification: no migraine, migraine with aura or migraine without aura). The two dataframes for the separate filling methods do not differ in the values for the given parameters.

	A	B	C
Number of participants	4629	2946	4629
Number of features	124	124	124
Averaged age	24	24	24
Number of men/women	1230/3398	-	1230/3398
Number labeled as no migraine	1683	-	1683
Number labeled as migraine	2946	-	-
——— Migraine without aura	-	1094	1094
——— Migraine with aura	-	1852	1852

5 Experiments

5.1 Preprocessing

Before training our models, we need to clean and preprocess the data to make sure they are ready for use in each classification. This includes handling missing values, eliminating empty columns, and keeping only the numerical columns that are needed for our analysis. We will also remove any columns that contain non-numerical data, such as strings, as the algorithms we will be using can only process numerical data. This preprocessing step is important to ensure the accuracy and effectiveness of our models.

As for the source of our data, we will use an existing dataset provided by LUMINA, which consists of two separate datasets, AA and BB. These datasets will be used to train our models. The AA data set is based on the SCREEN questionnaire, which participants filled out before participating in the study. This dataset contains 37 features and 4444 patients, of which 31 features

have data used to diagnose migraine. There are five key characteristics that cannot be excluded from the data set that will be used for training, which are based directly on the ICHD-3 criteria, as a LUMC student mentioned [17]. These are SCREENA, SCREENC, SCREENE, SCREENJ, SCREENM and FAMC. The BB data set is based on the extended questionnaire, which is given to patients after completing the SCREEN questionnaire. This dataset consists of 252 features and 4650 patients. The extended questionnaire contains more features that provide additional data for diagnosing patients. The extended questionnaire has a larger number of patients than the data set that contains the SCREEN questionnaire. The difference in the number of patients is due to the fact that there are two types of participants in the LUMINA study: patients referred to the study by polyclinics, who first complete the SCREEN questionnaire and then the extended questionnaire, and are identified in both data sets by the ID number 105; and patients who participate in the research, who only need to complete the EQ, and receive an ID number of 104 or another number.

To combine these two datasets, we need to create a unique ID for each patient to match the columns in the two datasets. We can do this by combining the ID and IDAA columns in both datasets to create a unique ID, and then use this unique ID to merge the two datasets into a single dataset called Merged DF.

To define our label for training our models, we will use the predictions of the LUMINA study described in [26]. The study made three diagnoses for each patient: the first after the SCREEN questionnaire (CRIT01HPIJN), the second after a phone call with a caller (CRIT02HPIJN) and the third by a headache specialist (CRIT03HPIJN). The most accurate diagnosis among these three is subsequently placed in a separate column, designated CRIT04HPIJN. We will use CRIT04HPIJN as our primary label source, as it represents the most accurate value among the three diagnoses. First, the value of CRIT03HPIJN is assigned to CRIT04HPIJN. If this column is empty, the value of CRIT02HPIJN is used to populate CRIT04HPIJN. If CRIT02HPIJN also has no value, CRIT01HPIJN is utilized to fill the gap. This process results in a completely populated label column named CRIT04HPIJN.

We observed that several columns contain missing values and some even have zero values. The total number of missing values in our merged_df amounts to 645,852, which represents 49% of the total cell count (1,315,000) in our data set. Missing values are problematic because the models cannot effectively handle them. The models may discard rows with missing values, substantially reducing our sample size and leading to sub-par performance. To address the missing values in these columns, we removed all zero-valued columns from the merged DataFrame. This resulted in the elimination of 39 columns. Additionally, since the models we plan to use only process numerical data, we excluded 20 columns containing strings from the merged df, leaving us with 419,543 missing values.

Addressing missing values in categorical data can be more complex than in numerical data, as numerical imputation methods such as mean or median imputation cannot be applied. There are several methods to handle missing values in categorical data, including multiple imputation, constant imputation, mode imputation, and deletion [16]. Deletion involves removing rows with empty values, but this can decrease the data set sample size.

Constant imputation involves replacing missing values with a fixed value, such as "unknown" or "other." Mode imputation replaces missing values with the most frequent value (mode) in the column. Multiple imputation utilizes various imputation models to fill in missing values. These models are trained on available data and generate multiple imputed data sets. The results of the different data

sets are combined to create a final data set with imputed values. This approach can be more effective than single-imputation methods, as it accounts for the uncertainty introduced by imputing missing values. To employ the multiple imputation technique, we used the IterativeImputer() method from the fancyimpute library. The IterativeImputer method fits an imputation model for each column with missing values and iterates the process until all missing values are replaced with new values. However, this method can be computationally demanding, particularly for datasets with a high number of missing values, which may require hours to fill the missing values in a dataset.

To ensure that we understand which features and patients cause a large amount of missing values, we analyzed the percentage of missing data for columns and rows. This resulted in Table 5. We observed that at least 132 columns have over 10% missing data and 98 columns have over 40% missing data. This amount does not differ much, but 82 columns have more than 40% missing data, indicating that a small number of columns share the same rows with missing values in specific columns. The Extended Questionnaire contains columns that a small percentage of patients have completed. The number of rows that have over 40% missing data is 2295, supporting that columns with more than 40% missing data cause the large number of missing values. We will exclude these 98 columns from our dataset to prevent any overfitting can be caused by handeling missing values. 6 columns that are important for the ICHD-3 criteria, mentioned to us by a fellow researcher from the migraine department at the LUMC, are not removed from the dataset. This leaves us with a remaining 145 columns, as shown in Table 9. After the deletion, we are left with only 72,066 missing values, which is 6% of our total data set.

% missing values	>10%	>20%	>30%	>40%	>50%	>60%	>70%	>80%	>90%
n columns	132	126	115	98	82	79	78	67	43
n rows	4641	4569	4196	2295	580	21	4	2	0

Table 5: Table describing number of columns and rows containing missing values for percentages in the merged df dataset

In this chapter, multiple methods are mentioned that help fill in the missing values. However, two other methods are used for handeling missing values, called the Forward and Backward fill, supplied by the fillna() method. The fillna() method can locate missing values in the DataFrame and replace them with other values in the same column. The fillna() method can fill the values in two ways: bfill (backward fill) and ffill (forward fill). Bfill uses the most recent valid observation in the previous row to fill in missing values, while ffill uses the next valid following row. Both methods can be used to fill missing values in a DataFrame or Series by calling the method on the data. In our experiment, we will use both of these filling methods to fill missing values in our data. By using both methods, we can understand the performance difference and overfitting that the two methods potentially bring with them. Filling in the missing values with fillna() resulted in some rows that still contained missing values. The two dataframes had separate rows that contained the missing values, which resulted in removing these rows (20) from both dataframes to create an equality between the two.

To gain further insight into how the models have predicted the outcomes, we compare the predictions of the models with the diagnosis predicted by the ICHD-2 algorithm, also known as the EQ, mentioned by Oosterhout et al. [26]. We have created confusion matrices for each classification. The first binary classification has only two labels, no migraine or migraine. A dataframe is created with only the no migraine and migraine, MO and MA combined, from the CRIT01HPIJN features

that are based on the EQ predictions. Using this dataframe, the confusion matrix is compiled and used for Tables 6 and 8. The same method is used to create the dataframe for the confusion matrices for the second binary classification and the multiclass classification. The second binary classification has a dataframe that has two labels: MO and MA, and the data frame for the multiclass classification keeps all the labels without excluding any.

As we will have two binary and a multiclass classification tasks, with each classification having other needs, we have created three separate datasets for use for the matching classification.

First binary classification: Migraine or not Migraine

To start with the first binary classification, we need to use only two labels: 0 for no migraine and 1 for migraine. The column CRIT04HPIJN in our dataset contains three labels that are gathered using the diagnosis in CRIT03HPIJN, CRIT02HPIJN, and CRIT01HPIJN. To simplify, we will keep the 0 label that refers to no migraine or no diagnosis and combine labels 1 and 2 into one label: 1 for migraine. Currently, we have assumed that label 0 refers to the diagnosis that there is no migraine. We have created a new column called CRIT04HPIJNCLASS in our merged dataframes to store these new labels.

However, it is important to remember that we do not want to use any of the labels from other columns that contain diagnosis information for our classification, not even the new column CRIT04HPIJNCLASS, as it may negatively impact the accuracy of our model. So, to achieve this, we have created two new datasets (bf and ff), by dropping the first 17 columns, the columns that contain labels such as CRIT0X, CRIT0XHPIJN and UITSLAG, and the newly created CRIT04HPIJNCLASS column from the merged (for both bf and ff variants). This will ensure that our model is trained only on the relevant data. Table 4 shows the parameters of the dataset.

Second binary classification: aura or no aura

The first binary classification task was to differentiate between no migraine and migraine for the diagnosis of patients. The second binary classification is to distinguish between migraine without aura and migraine with aura. To achieve this, we created a column called "CRIT04HPIJNAURA" that contains all the values from the column "CRIT04HPIJN", excluding those that are labeled as no migraine. We have made the label "0" to store the MO-labeled patients and the label "1" to store the MA patients. However, removing the no-migraine label from our prediction label resulted in fewer instances (2346 instances compared to 4929 in the first binary classification), which may increase the risk of overfitting and poor performance on unseen data. Nevertheless, a smaller dataset may benefit models that perform well on smaller datasets, such as Support Vector Machines. After creating a separate column for our two labels, the process is similar to the one performed in the first binary classification, by remving any labels that originate from a diagnosis and called this dataset binary2 (for both bf and ff variants). Table 4 shows the parameters of binary2.

Multiclass classification: not Migraine, Migraine with aura or Migraine without aura

We have prepared two data sets for multiclass classification by removing any labels that originate from a diagnosis. This includes 17 columns that contain possible diagnosis labels and ID numbers, as well as other columns without categorical values. The datasets we will use for training our models

are: the bf-variant where missing values are filled using the backfill method and the ff-variant for which the forwardfill method is used to fill the missing values. Also summarized in Table 4.

5.2 Results

Table 6: In this confusion matrix, seven different algorithms are being used for a first and second binary classification tasks: the Random Forest Classifier (RFC), the Decision Tree Classifier (DTC), the Support Vector Machine (SVM), the Logistic Regression (LR), the K-Nearest Neighbor (KNN), the Naive Bayes (NB), and the XGBoost (XGB). The EQ is the diagnosis predicted by the algorithm in the study by Oosterhout et al. [26], column CRIT01HPIJNof the two merged dataframes, shown as ('backfilled' - 'forwardfilled') next to the '/' inside the confusion matrices. The 'Prediction' is the diagnosis predicted by the model, which is compared to the Clinic-diagnosis labels of (A) no migraine (-) and migraine (+), and (B) migraine without aura (MO) and migraine with aura (MA) from the binary2 data set. The total number of predictions is (A) 4629 and (B) 2946.

		A				B	
		Clinic-diagnosis -	Clinic-diagnosis +			MO	MA
Algorithm	Prediction/EQ -	(1614—1613) / 1070	(69—70) / 597	Algorithm	Prediction/EQ MO	(972—973) / 776	(122—121) / 530
RFC	Prediction/EQ +	(38—40) / 613	(2908—2906) / 2349	**RFC**	Prediction/EQ MA	(24—26) / 318	(1828—1826) / 1322

		A				B	
		Clinic-diagnosis -	Clinic-diagnosis +			MO	MA
Algorithm	Prediction/EQ -	(1593—1593) / 1070	(90—90) / 597	Algorithm	Prediction/EQ MO	(992—990) / 776	(102—104) / 530
DTC	Prediction/EQ +	(135—112) / 613	(2811—2834) / 2349	**DTC**	Prediction/EQ MA	(74—84) / 318	(1778—1768) / 1322

		A				B	
		Clinic-diagnosis -	Clinic-diagnosis +			MO	MA
Algorithm	Prediction/EQ -	(1124—1123) / 1070	(559—560) / 597	Algorithm	Prediction/EQ MO	(513—524) / 776	(581—570)/ 530
SVM	Prediction/EQ +	(228—223) / 613	(2718—2723) / 2349	**SVM**	Prediction/EQ MA	(101—113) / 318	(1751—1739)/ 1322

		A				B	
		Clinic-diagnosis -	Clinic-diagnosis +			MO	MA
Algorithm	Prediction/EQ -	(1110—1113) / 1070	(573—570) / 597	Algorithm	Prediction/EQ MO	(569—596) / 776	(525—498) / 530
LR	Prediction/EQ +	(350—360) / 613	(2596—2586) / 2349	**LR**	Prediction/EQ MA	(254—270) / 318	(1598—1582) / 1322

		A				B	
		Clinic-diagnosis -	Clinic-diagnosis +			MO	MA
Algorithm	Prediction/EQ -	(1611—1614) / 1070	(72—69) / 597	Algorithm	Prediction/EQ MO	(994—995) / 776	(100—99) / 530
XGB	Prediction/EQ +	(47—53) / 613	(2899—2893) / 2349	**XGB**	Prediction/EQ MA	(49—51) / 318	(1803—1801) / 1322

		A				B	
		Clinic-diagnosis -	Clinic-diagnosis +			MO	MA
Algorithm	Prediction/EQ -	(1084—1091) / 1070	(599—592) / 597	Algorithm	Prediction/EQ MO	(619—649) / 776	(475—445) / 530
NB	Prediction/EQ +	(429—428) / 613	(2517—2518) / 2349	**NB**	Prediction/EQ MA	(408—431) / 318	(1444—1421) / 1322

		A				B	
		Clinic-diagnosis -	Clinic-diagnosis +			MO	MA
Algorithm	Prediction/EQ -	(982—982) / 1070	(701—701) / 597	Algorithm	Prediction/EQ MO	(599—607) / 776	(495—487) / 530
KNN	Prediction/EQ +	(264—270) / 613	(2682—2676) / 2349	**KNN**	Prediction/EQ MA	(279—284) / 318	(1573—1568) / 1322

5.2.1 The two Binary Classification

First Binary Classification

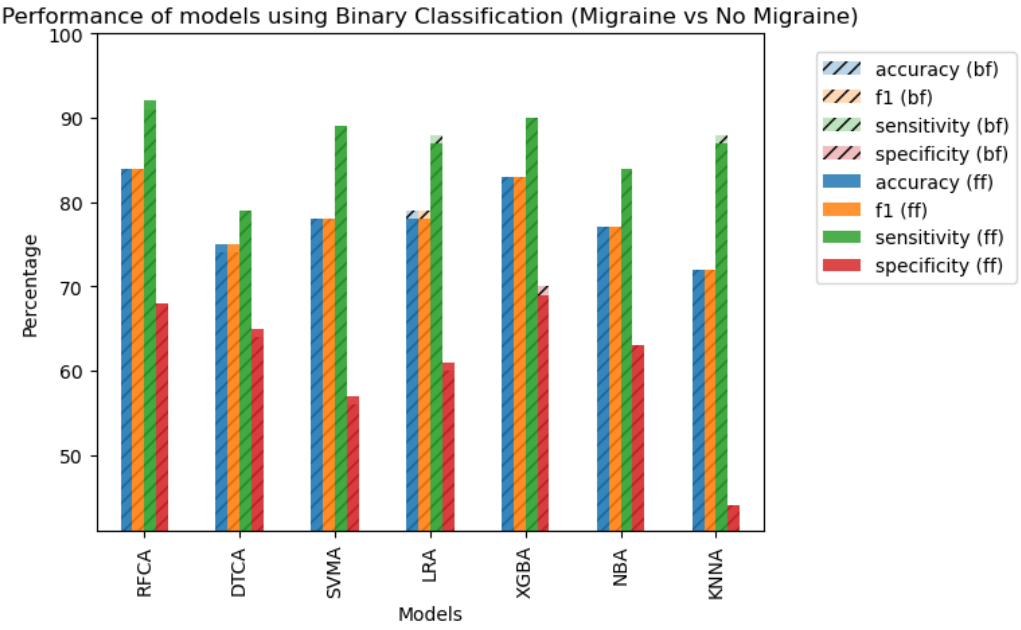


Figure 9: In our first binary classification task, we evaluated the performance of the models: the Random Forest Classifier (RFCA), the Decision Tree Classifier (DTCA), the Support Vector Machine (SVMA), the Logistic Regression (LRA), K-Nearest Neighbor (KNNA), Naive Baye (NBA), and XGBoost (XGBA) using two datasets, forward (ff) and backward (bf) filled, containing two labels: migraine and no migraine. The performance of these models was measured using four metrics: Accuracy, F1 score, Specificity, and Sensitivity.

Initially, we started by initializing our models for the first binary classification. The default parameters of all models will be kept the same during our first experiment, except for Logistic Regression, for which, we will increase the maximum iteration number from 100 to 10000. This is necessary as the model was unable to complete the analysis of our dataset when using the default setting. By increasing the maximum iteration number to 10000, we aim to ensure that the model can fully analyze our dataset.

We have used our training function including all our seven models, and for each model, the loop goes through the different 5-fold of data that is created. For each fold, we get the indices of the data that are used for training and testing and assign them to the corresponding variables X_train, X_test, y_train, and y_test. Then, using these variables, we fit the model and calculate its sensitivity, specificity, accuracy, and, as an extra metric, the f1 score and take the mean over the 5-folds and save it in metricsdf. The reason for using the f1 score is that it takes into account both the number of true positive and the number of false positive results. It is best used when both precision and recall are important and have a balance with the problem. The other three metric each only focus on one of the properties of the confusion matrix, for example, sensitivity looks at the true positive from all positive results, and sensitivity at the percentage of true negative from all negatives. The F1 score can give a more weighted picture of the performance of a classifier than

the accuracy.

All metrics mentioned above are generated from the two separate merged dataframes (backfilled and forwardfilled) and stored in a new one, shown in Table 7, and used to visualize and analyze the models further. In our case, a bar chart is a great way to visualize the results of a comparison between multiple models in terms of sensitivity, specificity, accuracy and f1 score. Each of the seven models is represented by a separate bar, and the length of the bar corresponds to the value of the metric for that model in percentages, resulting in figures such as Figure 9. The figure contains two overlay plots, the first representing the models trained with the backfilled dataframe, and the second one trained on the forwardfilled dataframe. In this way, the difference between the two filling methods can be compared and examined.

The process of analyzing the importance of different features in a dataset is known as feature importance analysis. It is used to understand which features are most significant when making predictions or classifications, as well as how each feature affects the performance of the model as a whole. However, only some of these models have the capability to save the feature importance, such as Random Forest Classifier, Decision Tree Classifier, Logistic Regression, Naive Baye, and XGBoost. For these models, we will save the feature importance for each iteration through the fold, taking the mean over the 5-folds. This information will then be used to create a bar chart to visualize the top 10 most important features and the least 10 important features, making it easier to understand which features are most relevant to the problem. However, the Support Vector Machine and K-Nearest Neighbor do not have an easily accessible method to save feature importances using the coefficients of the model, so they will not be included in the feature importance analysis.

Figure 9 shows that the metrics of the K-Nearest Neighbor (KNN) model are generally the lowest among the other six models, with a sensitivity of 88% and specificity of 44% according to Table 7. The ff variant shows a smaller sensitivity, 87%. This aligns with the findings in 3.1 that K-Nearest Neighbor does not perform well on larger datasets and many features, which is evident in our case. The specificity of the KNN model is particularly low, indicating that the use of this model for prediction could lead to more false positives, resulting in misdiagnosis of people without migraine as having migraine. Additionally, Figure 7 illustrates that KNN is a model with a lower accuracy, which can result in overfitting of the data and thus making predictions less accurate and resulting in more false positives. However, it should be noted that the KNN model has a higher true sensitivity, which refers to a high number of true positives in the predictions, as shown in Table 6, where the false negatives of the KNN models are higher than the predictions of the EQ. However, the model succeeds in predicting fewer false positives than the EQ. The trade-off between true sensitivity and specificity is evident in this case.

However, the other models have a higher specificity than the KNN model but still with a lower accuracy for their predictions than expected. The Random Forest Classifier (RFC) shows the highest accuracy value of 84%, with the XGBoost (XGB) model close to it with a value of 83%. The sensitivity of the RFC model is yet again higher than the specificity, respectively 91% and 68%, with a clear trade-off between the two metrics. Table 6 shows that the RFC model has 600 more true positives and true negatives, compared to the EQ. The bf variant shows a small difference in one or two more true positives and negatives and one or two less false positives and negatives. False negatives and false positives of the model are the lowest among the other six models. The RFC shows that it performs as it promises to, in Section 3.1, where it is mentioned that the model can handle large datasets with higher dimensionality, which is valid in our dataset with

147 features. The scores for the four metrics for RFC and XGB are close to each other, with the same ability to handle large datasets as RFC, mentioned in Section 3.1. However, if we look at the importance of the features of the two models, Figure 10 we can see that some features that have a high coefficient in the two models are visible in both models. The feature that has a high influence in the RFC, HPIJND, is not in the top 10 most important features of XGB, but the features related to BESCHRX have a high importance in both models. This means that these features are highly related to the diagnosis of migraine by our models. Something that stands out is the influence of the LEEFTIJD feature in the RFC model, as it is one of the most important features. The DTC and NB models seem to share the same importance for the LEEFTIJD feature, as shown in Figures 17 and 18.

The features related to CAFEINEX seem to have low importance in the RFC and XGB models. These features do not assist the model's process of trying to find the correlation between features.

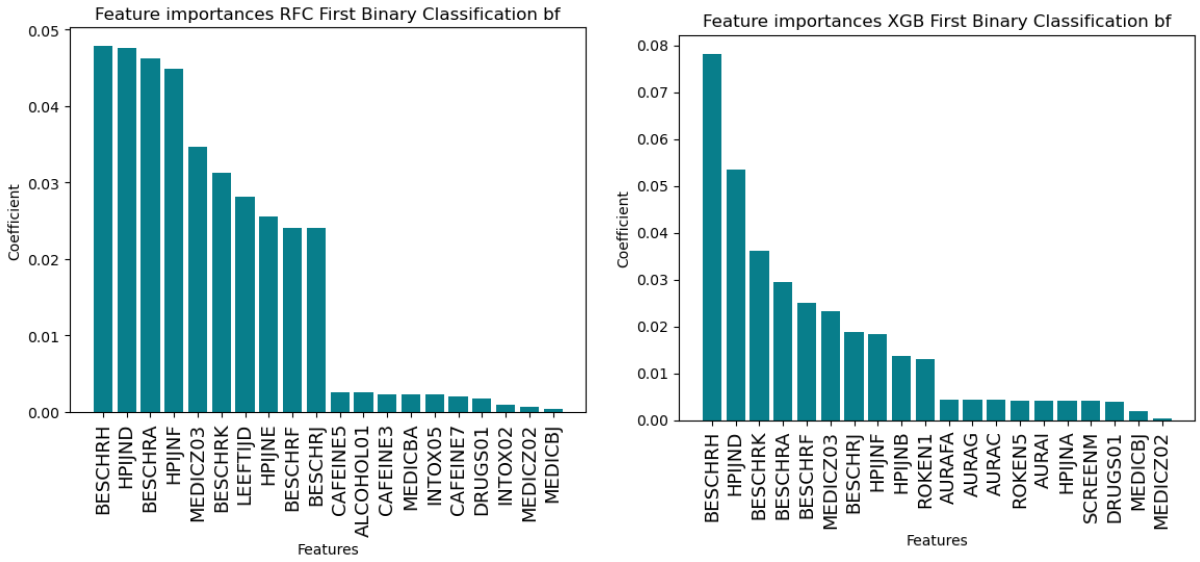


Figure 10: The feature importance in coefficients for the models Random Forest Classifier (RFA) and XGBoost (XGBA) in the first binary classification. The bf variants of both models are used, as the most and least important features of both variants do not differ in the parts where we examine.

The Decision Tree Classifier (DTC), Logistic Regression (LR), Support Vector Machine (SVM) and Naive Bayes (NB) seem to perform similarly on the four performance metrics, with a slight exception for the SVM model where the trade-off between sensitivity 89% and specificity 56% (bf variant) is clearly visible. SVM with high sensitivity but low specificity will lead to a high number of false positives, as seen in Table 6, where the SVM model has a high number of false positives, compared to the other models, but still fewer false positives compared to the EQ. The DTC model shows in Figure 9 a performance similar to the other three models, but Table 6 indicates otherwise. The model performed well, with a good balance between sensitivity and specificity. It predicted the third highest number of true positives and true negatives among all models, with a sensitivity of 79% and specificity of 64% (using the ff variant of the dataframe). This means that the model correctly identified a high proportion of positive cases while also avoiding many false positives.

When comparing the models trained on the forward (ff) and backward (bf) variants of the

dataframe, variant ff showed a higher specificity, with 2% more correct identifications of positive cases compared to variant bf. This resulted from the fact that variant ff had fewer false positives than variant bf, as shown in the confusion matrix table.

Furthermore, the DTC model trained on the ff variant of the dataframe correctly predicted 23 more patients compared to the bf variant. This suggests that the ff variant may have been more suitable for the model, possibly due to its ability to better preserve the temporal structure of the data or capture the underlying patterns in the missing values. This suggests that the choice of imputation technique should depend on the specific characteristics of the data and the problem being solved.

In general, according to Figure 9 and Table 6, it appears that both the Random Forest Classifier (RFC) and XGBoost (XGB) show the best performance compared to the other models, with sensitivity 91% and specificity 68% for RFC, and for XGB sensitivity 90% and specificity 70%.

Table 7: The performance of the models for the backfilled (left table) and forward filled (right table) dataframes measured in percentages using four metrics: accuracy, f1 score, specificity, and sensitivity. We evaluated the performance of the models: the Random Forest Classifier (RFC), the Decision Tree Classifier (DTC), the Support Vector Machine (SVM), the Logistic Regression (LR), K-Nearest Neighbor (KNN), Naive Baye (NB), and XGBoost (XGB) for the first binary classification (XXXA), the second binary classification (XXXB) and the multiclass classification (XXXC)

Algorithm	Accuracy	F1 score	Sensitivity	Specificity
DTCA	73	73	79	62
DTCB	70	70	76	60
DTCC	58	58	71	82
NBA	77	77	84	63
NBB	66	66	73	56
NBC	60	60	75	80
KNNA	72	72	88	44
KNNB	61	61	74	39
KNNC	50	50	61	77
LRA	79	79	88	60
LRB	70	70	82	51
LRC	62	62	70	87
RFCA	83	83	91	68
RFCB	78	78	90	58
RFCC	70	70	75	94
SVMA	78	78	89	56
SVMB	71	71	90	38
SVMC	63	63	63	92
XGBA	83	83	90	70
XGBB	78	78	86	66
XGBC	71	71	83	90

Algorithm	Accuracy	F1 score	Sensitivity	Specificity
DTCA	74	74	79	64
DTCB	69	69	75	58
DTCC	57	57	71	81
NBA	77	77	84	63
NBB	67	67	75	56
NBC	61	61	77	79
KNNA	72	72	87	44
KNNB	61	61	72	41
KNNC	49	49	62	78
LRA	78	78	87	61
LRB	70	70	82	50
LRC	62	62	71	88
RFCA	84	84	92	68
RFCB	78	78	91	56
RFCC	70	70	76	93
SVMA	78	78	89	57
SVMB	71	71	89	39
SVMC	62	62	63	93
XGBA	83	83	90	69
XGBB	78	78	85	66
XGBC	71	71	84	90

Second Binary Classification

We again start with initializing our model, so the previously trained model does not interfere with our new classification. We will keep the same parameters as described in the first binary classification for all the models.

The results are shown in Figure 11, which illustrates the impact of the decreased data set size on the performance of specifically the SVM and KNN models, with 2361 instances in the second binary classification shown in Table 4 compared to 4629 in the other classification. The KNN model has decreased from 88% to 74& in sensitivity. The specificity has also decreased by 5%, as shown

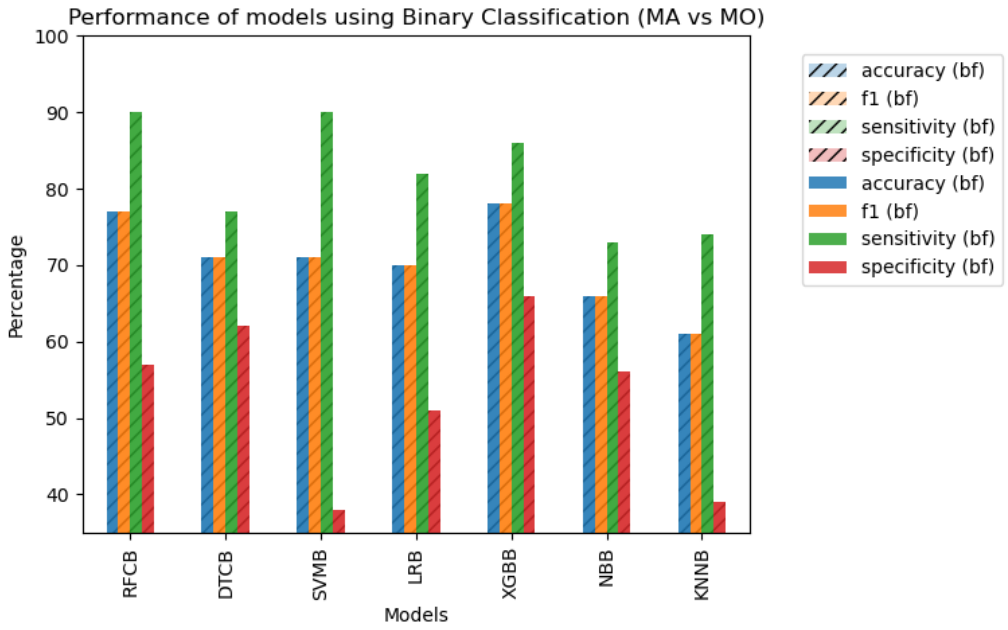


Figure 11: In our second binary classification task, we evaluated the performance of the models: the Random Forest Classifier (RFB), the Decision Tree Classifier (DTB), the Support Vector Machine (SVMB), the Logistic Regression (LRB), K-Nearest Neighbor (KNNB), Naive Baye (NBB), and XGBoost (XGBB) using two datasets, forward (ff) and backward (bf) filled, containing two labels: migraine with aura (MA) and migraine without aura (MO). The performance of these models was measured using four metrics: Accuracy, F1 score, Specificity, and Sensitivity

in Table 7. This decrease can be explained by the smaller sample size that is fed to the model. The model during the first binary classification may have been overfitted on the larger dataset, resulting in higher sensitivity and specificity. However, a smaller dataset may be harder to train on, resulting in less accurate predictions, supported by Table 7 which shows a decrease of 12% in the accuracy of the model. The specificity of the SVM model shows a really low score of 38% but a really high sensitivity of 90%, but a similar decrease compared to the KNN, from 56% to 38%, suggesting that the smaller dataset may not be sufficient to generalize the data set well. The less representative data set may have led to lower specificity for the model.

The LR and NB models seem to score slightly better than the previous two models, according to Table 7 and 6. The LR model with a sensitivity and specificity of respectively 82% and 51%, compared to NB with 73% and 56% seems to perform slightly worse than the first binary classification. The smaller sample size seems to affect the performance of the LR and NB models. The NB model predicts far more people falsely with MA than the LR model, which makes the LR model in this case better than the EQ predictions and the NB model worse. When we examine the feature importance of the two models, shown in Figure 12, none of the most important features seem to match, suggesting that the models do not recoginize the same pattern during training. What they do have in common, are the ROKEN1 and INTOX02 features that share the same position of being among the least important features in both models.

The RFC and XGB models show identical accuracy, f1 score scores, 78%, without leaving out that the sensitivity and specificity trade-off of these models are the highest, the sensitivity of the

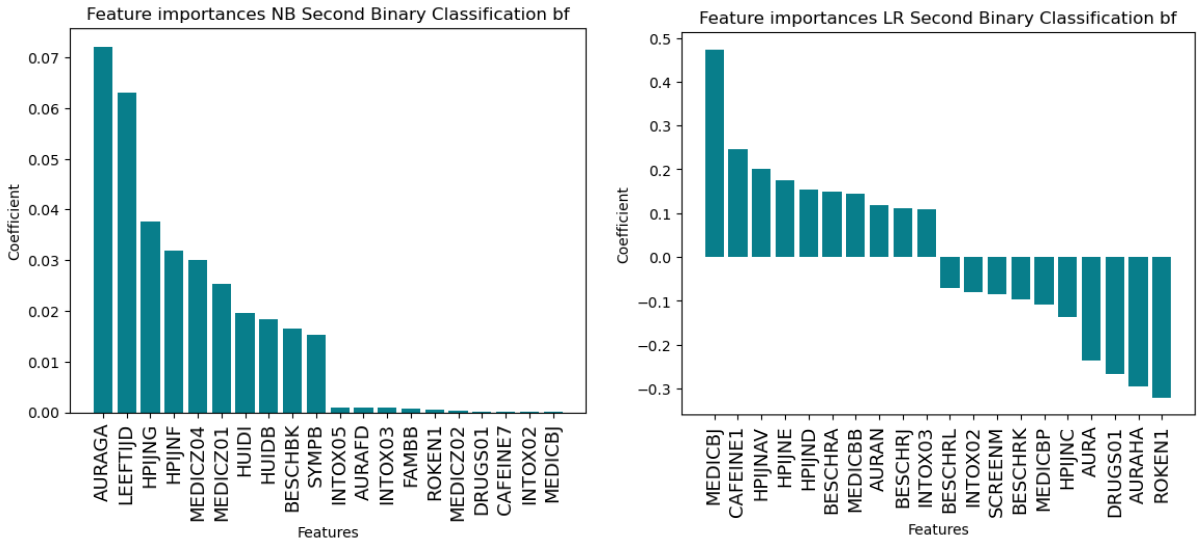


Figure 12: The feature importance in coefficients for the models Naives Bayes (NB) and Logistic Regression (LR) in the second binary classification. The bf variants of both models are used, as the most and least important features of both variants do not differ in the parts where we examine.

RFC and XGB are respectively 91% and 86%, and the specificity respectively 56% and 66%. The RFC model with the higher sensitivity classifies 1828 (bf variant) people truly with MA, as shown in Table 6. The higher specificity of the XGB model is also a result of the model predicting fewer false positives compared to the RFC model. When we look at the feature importance in Figure 13, the most important features of the two models are quite similar, referring to features that contain AURAX in them, referring to the fact that both models seem to find relations in the data that are close to each other.

In this classification, the RFC and XGB models perform, again, the best among the seven models.

5.3 Multiclass Classification

When initializing the models before starting the training, we use the same parameters as described in previous classifications, except for the Logistic Regression and XGBoost. The multiclass classification classifies the data set with three labels. For Logistic Regression to handle this, the mult_class parameter is set to "multinomial" to allow the model to handle multiple labels. XGBoost's parameter "num_class" is also set to 3 for the same reason. Other models can already handle multiclass classification.

To compare the results and performance of the binary and multiclass classification, we will use the same models: Random Forest Classifier (rf), Decision Tree Classifier (dt), Support Vector Machine (svm) and Logistic Regression (lr). As predict data, we will use the CRIT04HPIJN column from our merged dataset. The CRIT04HPIJN label, as previously discussed, contains three labels based on the other three CRIT0XHPIJN diagnoses. To give more insight on the distribution of the labels in the data, we will report the count of each label included in the data.

Table 8: In this confusion matrix, seven different algorithms are being used for a multiclass classification task: the Random Forest Classifier (RFC), the Decision Tree Classifier (DTC), the Support Vector Machine (SVM), the Logistic Regression (LR), the K-Nearest Neighbor (KNN), the Naive Bayes (NB), and the XGBoost (XGB). The EQ is the diagnosis predicted by the algorithm in the study by Oosterhout et al. [26], column CRIT01HPIJN of the two merged dataframes, shown as ('backfilled' - 'forwardfilled') next to the '/' inside the confusion matrices. The 'Prediction' is the diagnosis predicted by the model, which is compared to the Clinic-diagnosis labels of no migraine (-), migraine without aura (MO), and migraine with aura (MA) from the CRIT04HPIJN dataset of the merged dataframe. The total number of predictions is 4629.

		Clinic-diagnosis -	Clinic-diagnosis MO	Clinic-diagnosis MA
	Prediction/EQ -	(1631—1637) / 1070	(6—7) / 173	(46—39) / 424
Algorithm	Prediction/EQ MO	(36—39) / 147	(933—933)/ 530	(125—116)/178
RFC	Prediction/EQ MA	(45—35) / 466	(24—22) / 391	(1783—1795)/1250

		Clinic-diagnosis -	Clinic-diagnosis MO	Clinic-diagnosis MA
	Prediction/EQ -	(1593—1580) / 1070	(25—34) / 173	(65—69) / 424
Algorithm	Prediction/EQ MO	(50—50) / 147	(952—947)/ 530	(92—97)/178
DCT	Prediction/EQ MA	(89—80) / 466	(69—63) / 391	(1694—1709)/1250

		Clinic-diagnosis -	Clinic-diagnosis MO	Clinic-diagnosis MA
	Prediction/EQ -	(1277— 1268) / 1070	(65—52) / 173	(341—363) / 424
Algorithm	Prediction/EQ MO	(196—194) / 147	(401—406)/ 530	(497—494)/178
SVM	Prediction/EQ MA	(274—266)/ 466	(86—97)/ 391	(1492—1489)/1250

		Clinic-diagnosis -	Clinic-diagnosis MO	Clinic-diagnosis MA
	Prediction/EQ -	(1234—1222) / 1070	(120—111) / 173	(329—350) / 424
Algorithm	Prediction/EQ MO	(181—191) / 147	(483—495)/ 530	(430—408)/178
LR	Prediction/EQ MA	(297—295) / 466	(218—227) / 391	(1337—1330)/1250

		Clinic-diagnosis -	Clinic-diagnosis MO	Clinic-diagnosis MA
	Prediction/EQ -	(1623—1625) / 1070	(20—15) / 173	(40—43) / 424
Algorithm	Prediction/EQ MO	(26—21) / 147	(974—980)/ 530	(94—93)/178
XGB	Prediction/EQ MA	(38—39) / 466	(36—45) / 391	(1778—1768)/1250

		Clinic-diagnosis -	Clinic-diagnosis MO	Clinic-diagnosis MA
	Prediction/EQ -	(1067—1058) / 1070	(231—254) / 173	(385—371) / 424
Algorithm	Prediction/EQ MO	(150—140) / 147	(522—565)/ 530	(422—389)/178
NB	Prediction/EQ MA	(234—228) / 466	(305—329) / 391	(1313—1295)/1250

		Clinic-diagnosis -	Clinic-diagnosis MO	Clinic-diagnosis MA
	Prediction/EQ -	(1240—1229) / 1070	(164—187) / 173	(279—267) / 424
Algorithm	Prediction/EQ MO	(250—225) / 147	(550—576)/ 530	(294—293)/178
KNN	Prediction/EQ MA	(361—342) / 466	(301—313) / 391	(1190—1197)/1250

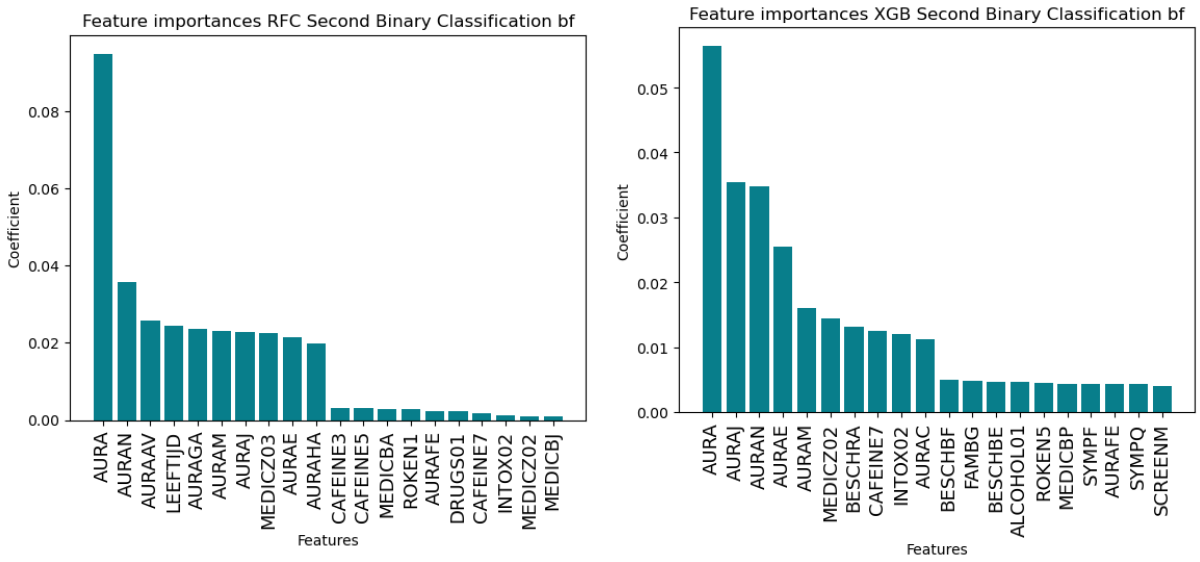


Figure 13: The feature importance in coefficients for the models Random Forest Classifier (RFC) and XGBoost (XGB) in the second binary classification. The bf variants of both models are used, as the most and least important features of both variants do not differ in the parts where we examine.

- 0 (no migraine) : 1683 labels
- 1 (MO) : 1094 labels
- 2 (MA) : 1852 labels

We have a total of 4629 diagnoses that were used to train the multiclass classification, including three labels: no migraine, migraine with aura, and migraine without aura. For each model, two separate models are trained using the bf and ff dataframes. The classification resulted in the bar chart in Figure 14.

Looking at Figure 14, we can see that all models score better for specificity than the sensitivity value. This means that there will be more false negatives compared to false positives in the predictions of the models. A reason for this could be that the data is unbalanced, making it harder for the models to identify the least common class among the three classes. Table 8 shows that the LR, NB, and KNN models have many false predictions, while the RFC and XGB models have fewer false negatives, as they have a higher accuracy. According to Table 7, the sensitivity and specificity of the RFC are 75% and 94%, while those of the XGB are 83% and 90%. This means that the RFC model has fewer false predictions compared to the XGB model.

The RFC, DCT, and XGB models are the only ones that predict the three labels more accurately than the EQ predictions, as shown in Table 8. When we compare the important features of the RFC and XGB models in Figure 15, we can see that the RFC and XGB models have HPIJND, AURA, BESCHRA/K as important features. The other features don't tell us much about the similar scores. An interesting feature is BESCHRH, which is important for the RFC model, but not for the XGB model. This could mean that the RFC model uses this feature more to find a pattern, while the XGB model finds other features more important for training.

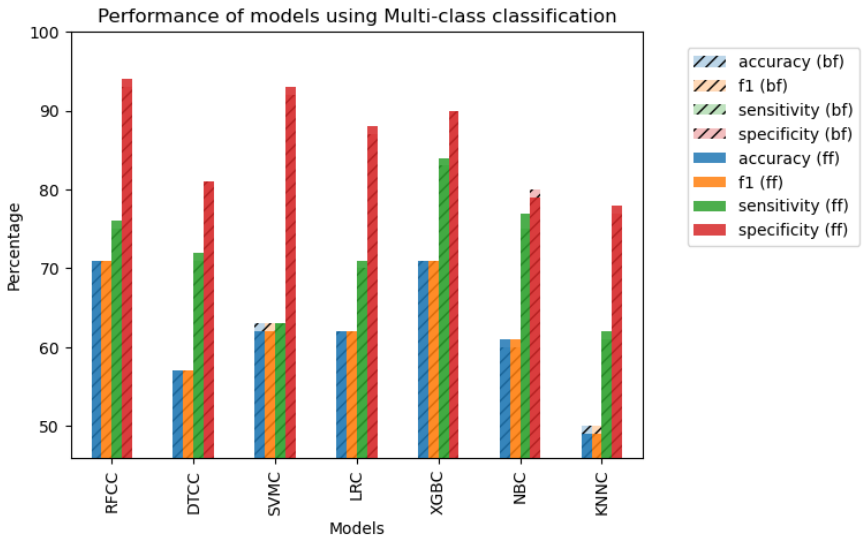


Figure 14: In our multiclass classification task, we evaluated the performance of the models: the Random Forest Classifier (RFC), the Decision Tree Classifier (DTC), the Support Vector Machine (SVMC), the Logistic Regression (LRC), K-Nearest Neighbor (KNNC), Naive Baye (NBC), and XGBoost (XGBC) using two datasets, forward (ff) and backward (bf) filled, containing three labels: no migraine, migraine with aura (MA) and migraine without aura (MO). The performance of these models was measured using four metrics: Accuracy, F1 score, Specificity, and Sensitivity.

The models with the lowest accuracy are the DCT (58%) and NB (61%) models. The DCT model can easily overfit when the data is noisy and complex, which can result in a completely different tree that performs poorly on the data.

Table 8 shows that predictions for no migraine and MA labels are more than the EQ predictions in all cases, except for the NB and KNN models. However, the models do not perform as well in predicting the MO labels. This could be because the model cannot effectively identify instances of the minority class if the features used to train the model are not helpful enough for the minority class. Additionally, the model might be overfitting to the majority class, causing high specificity but poor performance in the minority class.

When we compare the bf and ff variants in the multiclass classified models, the DTC and NB models appear to have higher accuracy and the f1 score when trained with the ff variant of the dataframe, as shown in Figure 14. On the other hand, the sensitivity and specificity of this variant show lower scores compared to the bf variant. The sensitivity and specificity of SVM and KNN are boosted by the ff variant, but this results in worse accuracy in the models. Overall, there is no variant that outperforms the other variant.

In conclusion, the choice of the algorithm can greatly affect the overall performance and accuracy of the predictions in a multiclass classification task. The RFC and XGB models perform the best, followed by the DTC model, while the LR, NB and KNN models have the lowest performance among all algorithms, with accuracies of 61%, 66%, and 58%, respectively. The high specificity and low sensitivity observed in the models can be attributed to the unbalanced nature of the data, referring to the importance of mentioning the imbalance of the classes in the multiclass classification.

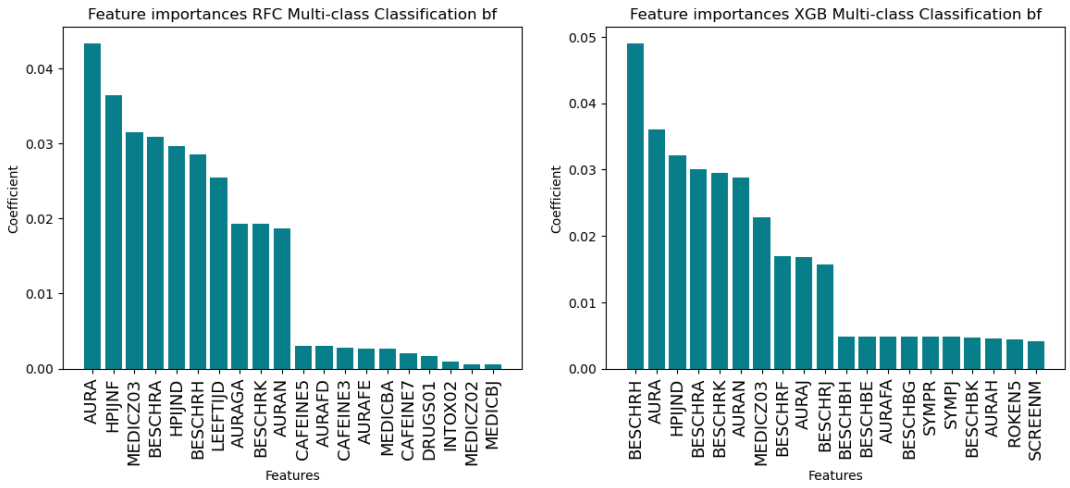


Figure 15: The feature importance in coefficients for the models the Random Forest Classifier (RFC) and the XGBoost (XGBC) in the multiclass classification. The bf variants of both models are used, as the most and least important features of both variants do not differ in the parts where we examine

Binary versus Multiclass Classification

We conducted two types of classification experiments to test the hypothesis that separating a classification problem into multiple binary classification tasks improves accuracy. To evaluate this, we performed two binary classification tasks and one multiclass classification. Comparing the results of these classifications shows that both binary classification tasks and multiclass classification produce really different performance scores, as shown in Figure 16 and Table 7. The Support Vector Machine and K-Nearest Neighbor models performed poorly in both classifications, particularly in terms of specificity in the second binary classification. However, in the multiclass classification, the specificity of these two models was more accurate than its sensitivity and accuracy. This suggests that the Support Vector Machine and K-Nearest Neighbor may only be able to correctly predict one class, in particular the classification for having migraine or not, shows higher scores compared to the aura classification, and identify other classes less accurately. The Decision Tree Classifier, Logistic Regression, and Naive Baye had for both classification their ups and downs, but showed the lowest accuracy in the multiclass classification, leaving us with an overall lower performance than the remaining two models. Both binary classification tasks have a higher score in the accuracy, f1 score, and sensitivity for almost all models, compared to the multiclass classification. This suggests that classification tasks with more than two classes are less accurate than classifications with only two classes.

All models, shown in Table 7, have lower accuracies in multiclass classification compared to the two binary classification tasks. The remaining two models used in both classifications that yielded similar results were the Random Forest Classifier and XGBoost performing the best in both classifications, but with XGBoost when averaged over the three classifications, having the highest sensitivity and specificity compared to the other models.

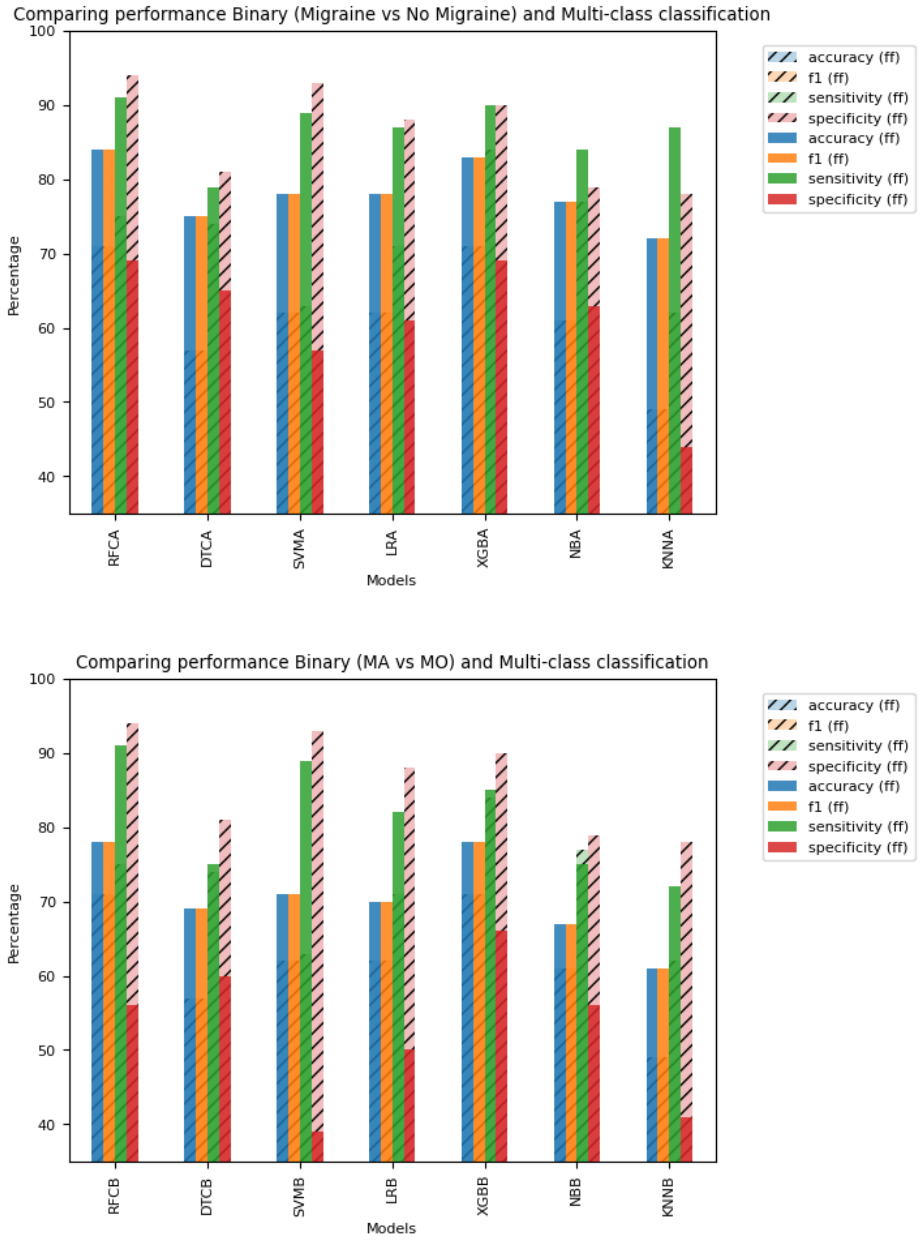


Figure 16: The figures compare the performance of our models: Random Forest Classifier (RFC), Decision Tree Classifier (DTC), Support Vector Machine (SVM), Logistic Regression (LR), K-Nearest Neighbor (KNN), Naive Baye (NB), and XGBoost (XGB) on two different tasks using two datasets, forward filled (ff). The first figure contrasts the binary classification task (migraine or no migraine) with the multiclass classification task (no migraine, migraine with aura (MA), and migraine without aura (MO)), while the second figure contrasts the binary classification task (migraine with aura (MA) or migraine without aura (MO)) with the multiclass classification task. In both figures, striped bars represent multiclass classification performance, and solid bars represent binary classification performance. Performance was evaluated using four metrics: Accuracy, F1 score, Specificity, and Sensitivity.

6 Conclusions and Discussion

After thorough experimental evaluation of different machine learning models, we were able to evaluate their performance and compare them based on accuracy, f1 score, sensitivity, and specificity. The results of this study have highlighted the potential of machine learning techniques in improving the diagnosis of diseases, particularly in the case of migraine.

We found that the binary classification approach is a feasible method to use on a multiclass problem, our study shows that the results obtained by using two binary classification problems are comparable to the results 5.2 obtained by using a multiclass classification problem. However, when comparing our results from machine learning models with the results that were given by the extended questionnaire of [26], our models showed visible differences, more accurate predictions, and less false negatives. This leads us to affirm that machine learning can indeed improve the diagnosis of migraine compared to the current algorithm. There were, however, some limitations due to shortcomings in computational power and data. The data label provided by LUMINA has labeled the diagnosis that referred to no migraine or did not have clear results, such as no migraine. This left us with a label that may not be as accurate as we think. These labels need some control before further research can be done on them. Another limitation is the lack of computational power to run the imputation on the large number of missing values. To fill this gap, we used other filling-missing-values methods, which obviously should not be the case. Moreover, it is important to consider the potential effect of imbalanced classes in our data set, mainly during multiclass classification, as it could have affected the performance of the models. Techniques such as oversampling and undersampling could be considered to address this issue and potentially lead to better model performance, especially in terms of sensitivity for the minority class.

One potential direction for further research could also be to analyze the feature importances and variable interactions which could be used to identify the most relevant factors that influence the prediction of the diagnosis of migraine and to gain deeper insights into the underlying mechanisms of the disease, more specifically. To improve the models, the features that have the least importance could be excluded from the data set that is given to the models. Another approach could be to consider the use of ensemble techniques such as stacking, bagging, and boosting that combine the predictions of multiple models to improve overall performance. This can be done by training several models and then combining their predictions in a way that enhances the performance on unseen data.

In general, this study contributes to the growing body of research on the implementation of machine learning techniques in the medical field and provides valuable insights into the potential of these techniques for the prediction of migraine. We believe that these results will serve as a catalyst for further research and the development of new and improved diagnostic tools for this condition.

References

[1] Migraine headaches. https://my.clevelandclinic.org/health/diseases/5005-migraine-headaches, Accessed on 8th March, 2023.

[2] L. Breiman. Classification and regression trees. *Wadsworth*, 1984.

[3] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] E. Briscoe and J. Feldman. Conceptual complexity and the bias/variance tradeoff. *Cognition*, 118(1):2–16, 2011.

[5] J. Brownlee. How to choose a feature selection method for machine learning. *Machine Learning Mastery*, 10, 2019.

[6] L. I. Campus. History, Definitions and Diagnosis, Dec. 2022.

[7] 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*, pages 785–794, 2016.

[8] M. Clinic. Migraine headache: Diagnosis and treatment. https://www.mayoclinic.org/diseases-conditions/migraine-headache/diagnosis-treatment/drc-20360207, n.d.

[9] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[10] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE, 2018.

[11] C. H. Göbel, S. C. Karstedt, T. F. Münte, H. Göbel, S. Wolfrum, E. R. Lebedeva, J. Olesen, and G. Royl. Ichd-3 is significantly more specific than ichd-3 beta for diagnosis of migraine with aura and with typical aura. *The journal of headache and pain*, 21(1), 2020.

[12] M. W. Green and R. Colman. Complicated migraine. In *Headache and Migraine Biology and Management*, pages 51–60. Elsevier, 2015.

[13] T. Hofman. The mini screen for diagnosing migraine. Project Wetenschapsstage, Leiden Universiteit, location: C:Universiteitde jaarProject, 2021.

[14] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant. Applied logistic regression. *John Wiley Sons*, 2013.

[15] J. Kwon, H. Lee, S. Cho, C.-S. Chung, M. J. Lee, and H. Park. Machine learning-based automated classification of headache disorders using patient-reported questionnaires. *Sci. Rep.*, 10(14062):1–8, Aug. 2020.

[16] R. J. Little and D. B. Rubin. *Statistical analysis with missing data*. Wiley, 2019.

[17] B. v. d. N. LUMC, Arend. SCREEN en Uitgebreide vragenlijst incl algoritmen, 2022. file:///C:/Users/BurakOzdemir/Documents/Leiden%20Universiteit/3de%20jaar/Bachelor%20Project/papers.

[18] P. R. Matthew Stewart. Guide to interpretable machine learning. *Medium*, May 2022.

[19] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[20] H. C. C. of the International Headache Society. The international classification of headache disorders, 3rd edition (beta version). *Cephalalgia*, 33(9):629–808, 2018.

[21] ResearchGate. Calculation of sensitivity, specificity, and positive and..., July 2022.

[22] ResearchGate. Machine learning Paradigms, July 2022.

[23] D. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by error propagation. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1:318–362, 1985.

[24] A. Sajee. Model complexity, accuracy and interpretability. *Medium*, October 2022.

[25] T. Steiner, L. Stovner, T. Nilsen, T. Vos, C. Steiner, and V. Pfaffenrath. Global burden of migraine: the global burden of disease study 2016. *JAMA*, 320(7):741–758, 2018.

[26] W. van Oosterhout, C. Weller, A. Stam, F. Bakels, T. Stijnen, M. Ferrari, and G. Terwindt. Validation of the web-based Lumina questionnaire for recruiting large cohorts of migraineurs. *Cephalalgia*, 31(13):1359–1367, 2011.

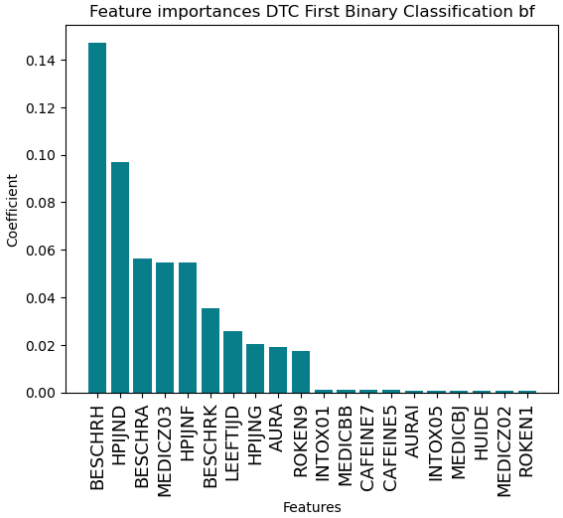
[27] WHO. Headache disorders. July 2022.

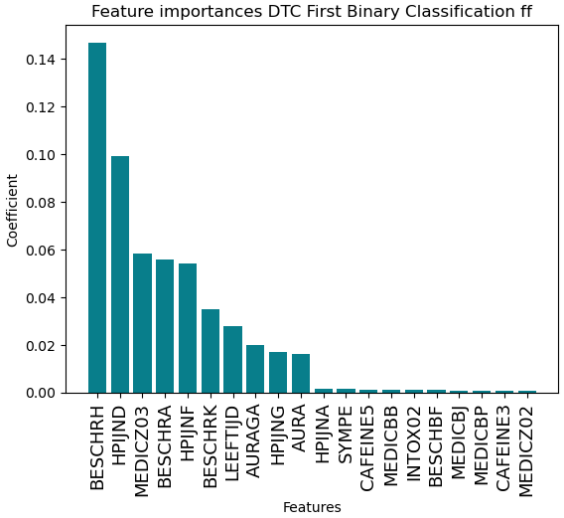
[28] X. Ying. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168:022022, 02 2019.

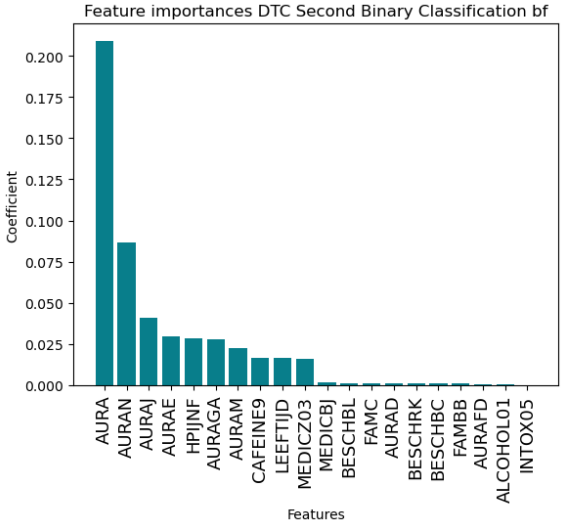
[29] T. Zhang and Y. Yang. The optimality of naive bayes. *Proceedings of the 17th International Conference on Machine Learning (ICML-00)*, pages 703–710, 2004.

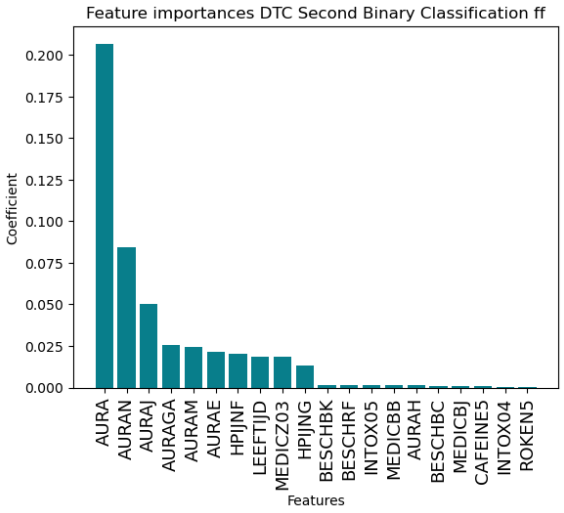
7 Appendix

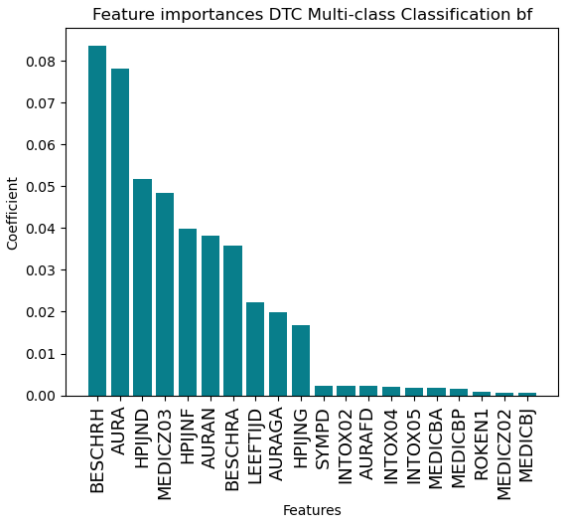
Figure 17: The importance of features for binary and multiclass classification in the Decision Tree Classifier (DTC) model, represented by coefficients, is determined for both backfilled (bf) and forwardfilled (ff) dataframes. The analysis includes the first binary, second binary, and multiclass classification scenarios.











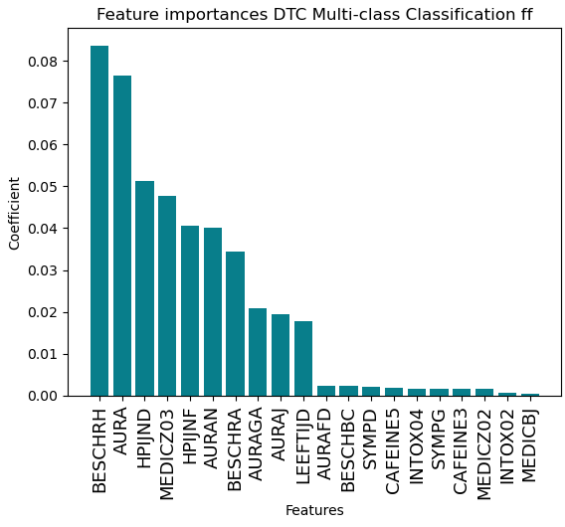


Figure 18: The importance of features for binary and multiclass classification in the Naives Bayes (NB) model, represented by coefficients, is determined for both backfilled (bf) and forwardfilled (ff) dataframes. The analysis includes the first binary, second binary, and multiclass classification scenarios.

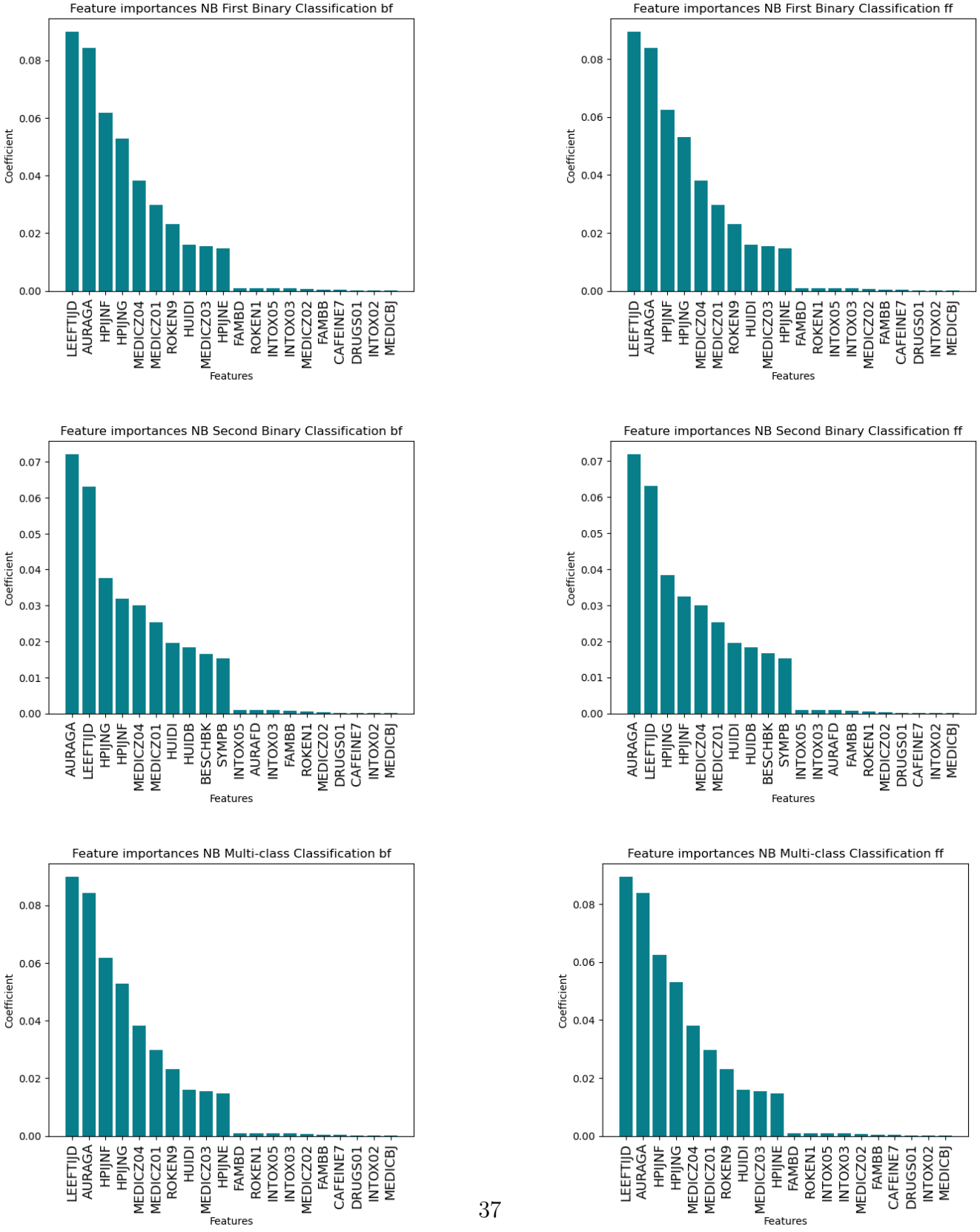
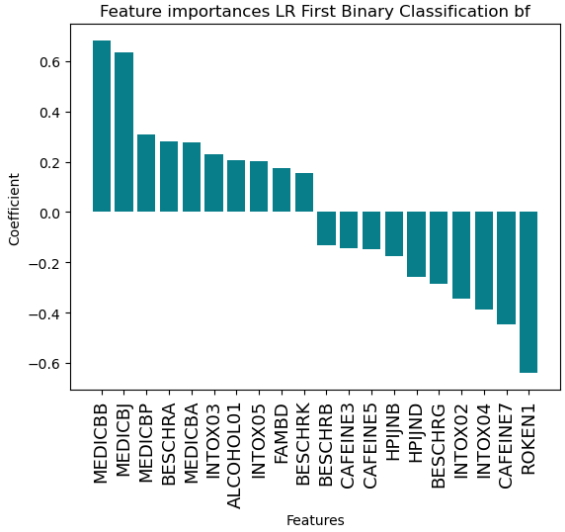
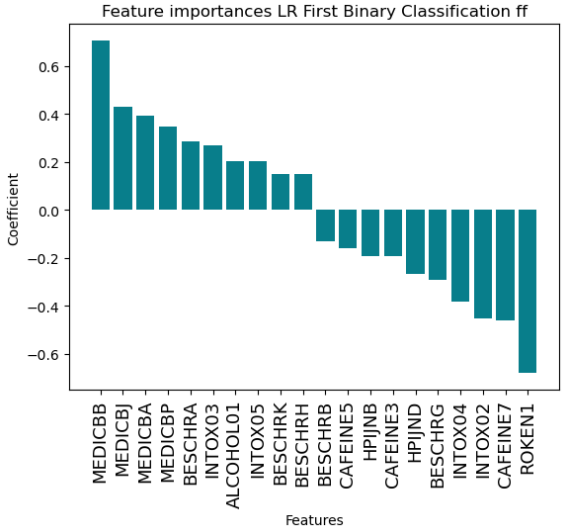
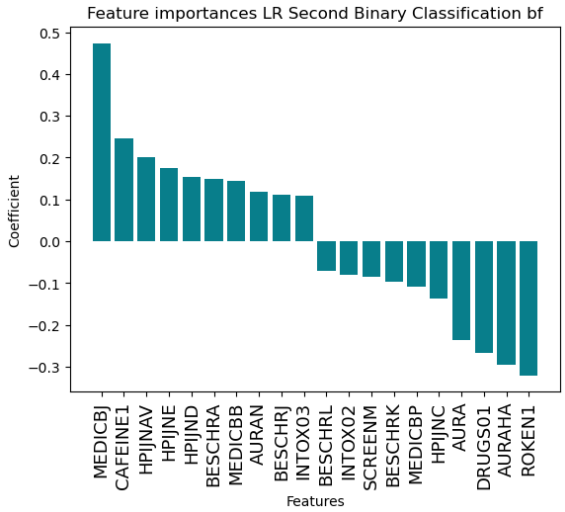
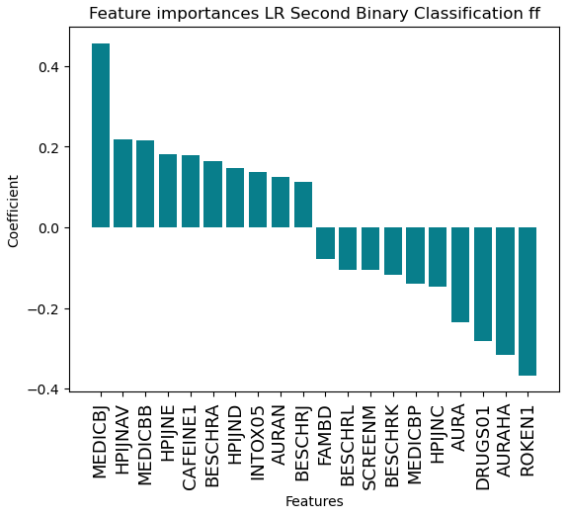


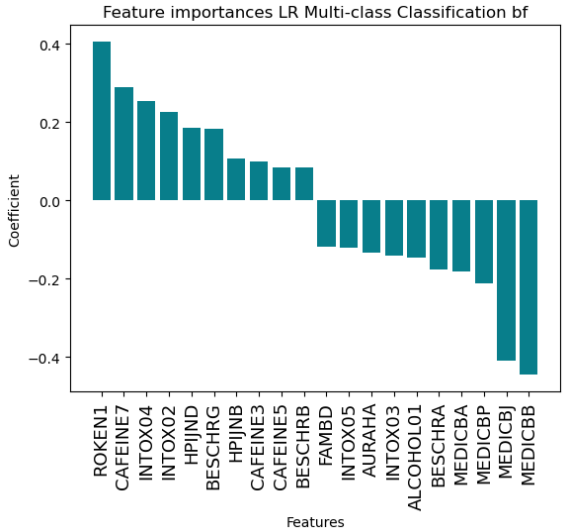
Figure 19: The importance of features for binary and multiclass classification in the Logistic Regression (LR) model, represented by coefficients, is determined for both backfilled (bf) and forwardfilled (ff) dataframes. The analysis includes the first binary, second binary, and multiclass classification scenarios.











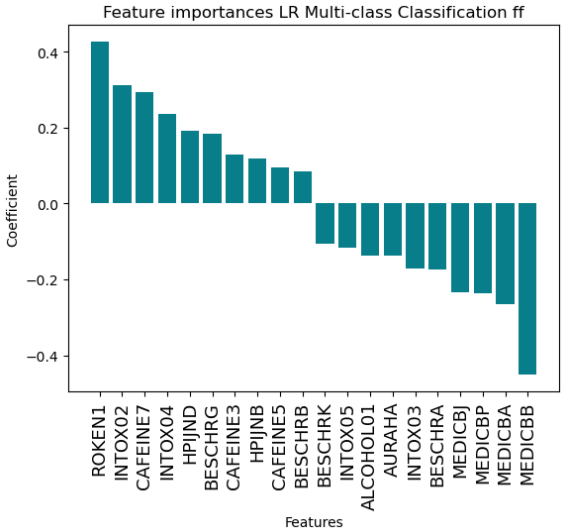
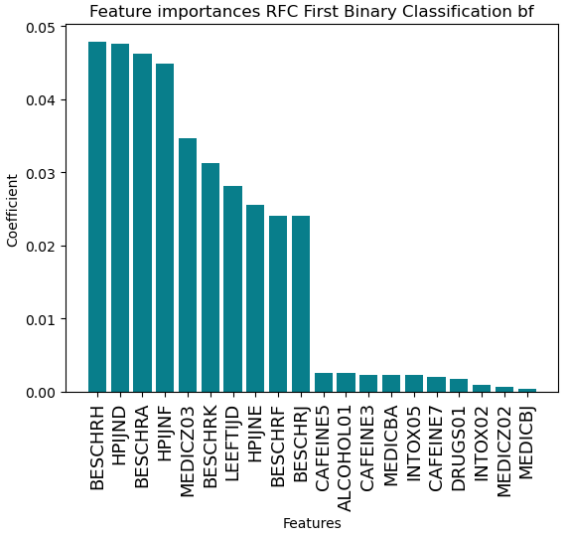
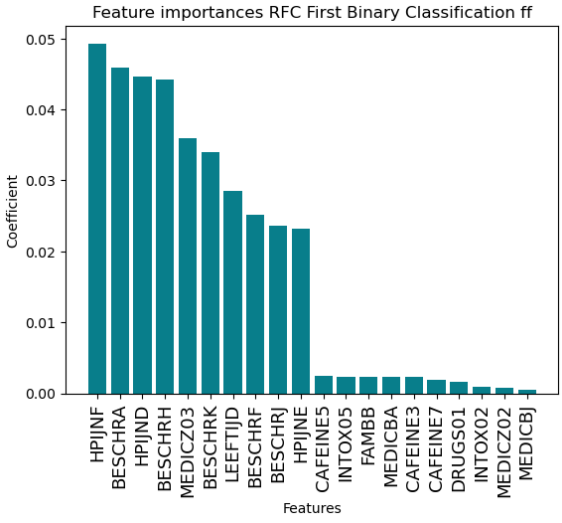
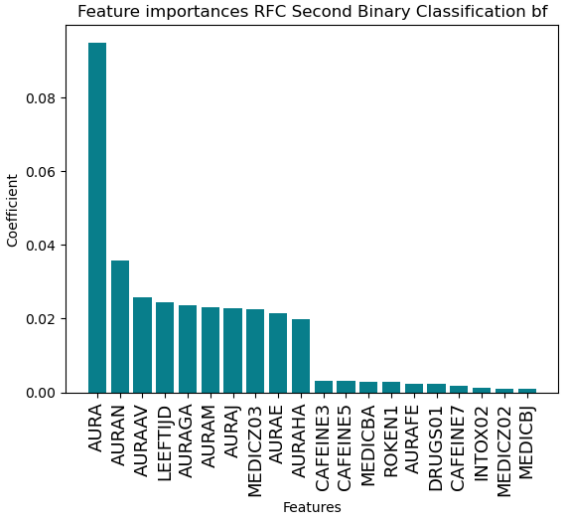
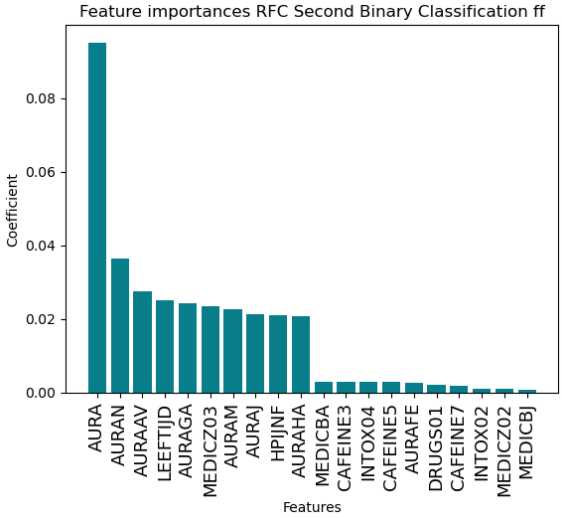


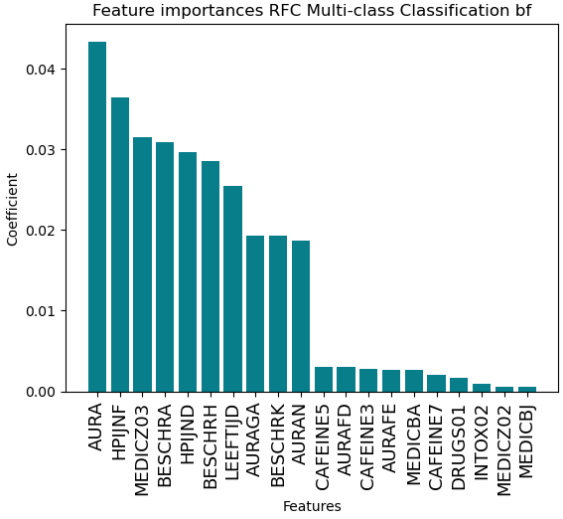
Figure 20: The importance of features for binary and multiclass classification in the Random Forest Classifier (RFC) model, represented by coefficients, is determined for both backfilled (bf) and forwardfilled (ff) dataframes. The analysis includes the first binary, second binary, and multiclass classification scenarios.











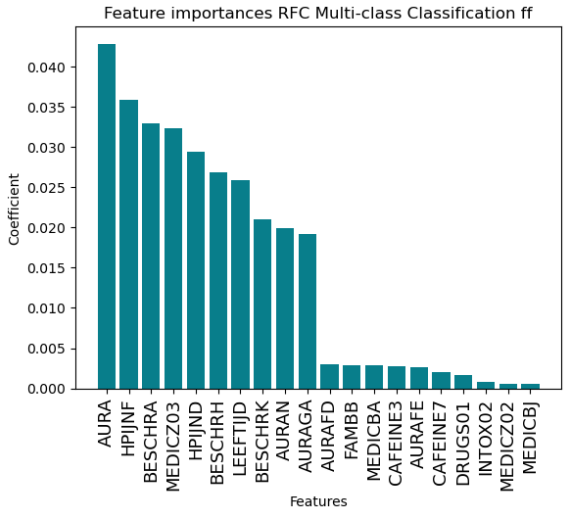
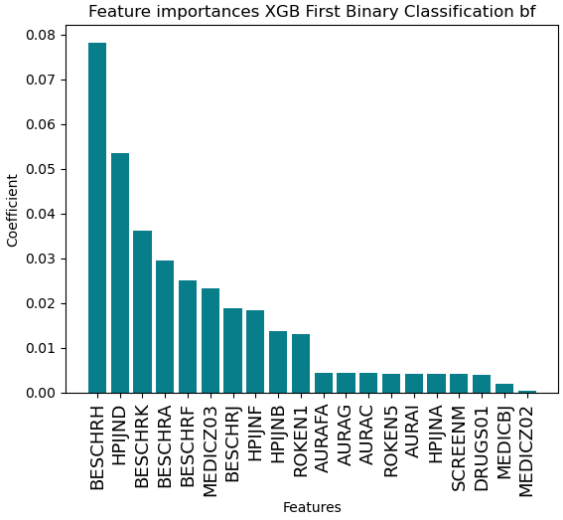
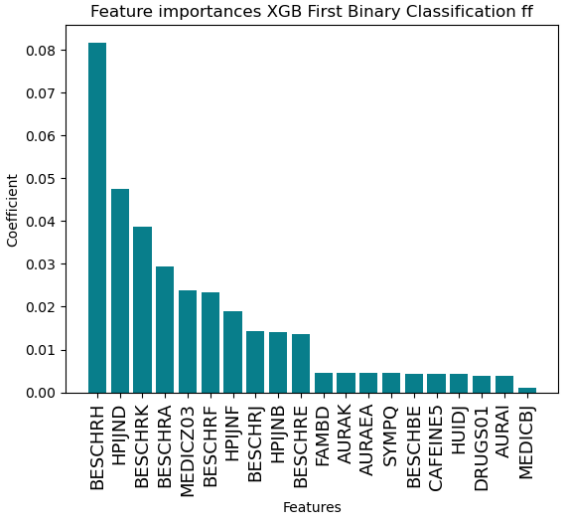
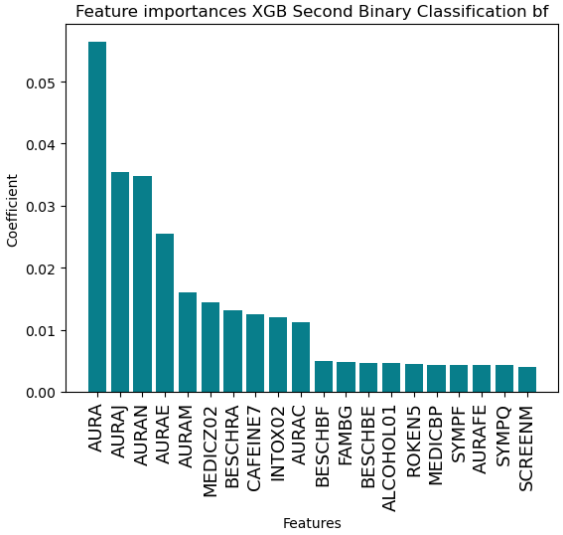
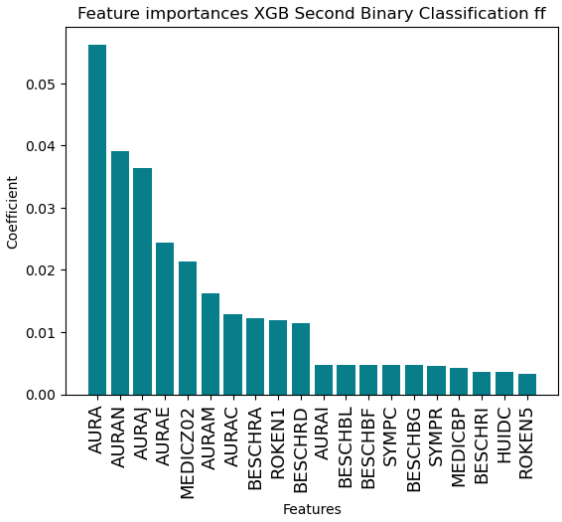


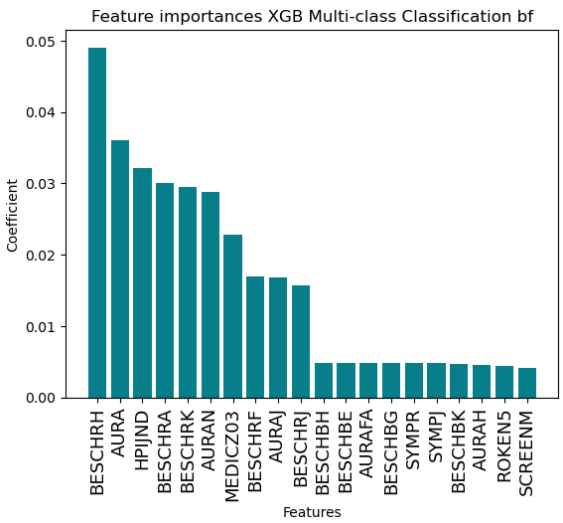
Figure 21: The importance of features for binary and multiclass classification in the XGBoost (XGB) model, represented by coefficients, is determined for both backfilled (bf) and forwardfilled (ff) dataframes. The analysis includes the first binary, second binary, and multiclass classification scenarios.











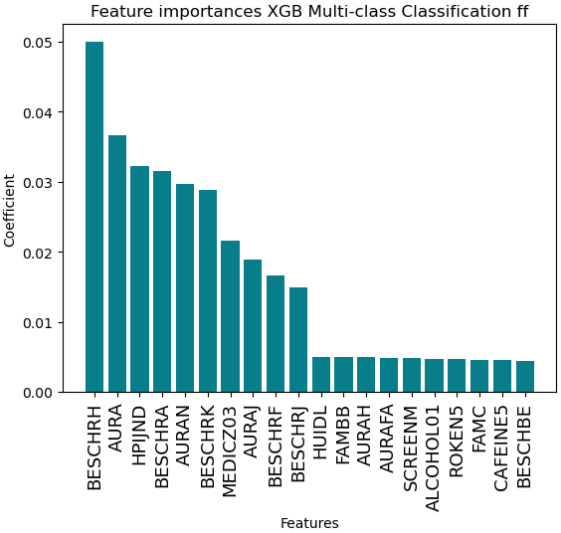


Table 9: This table describes the final, pre-processed dataset with names, the domain values (minimum and maximum), and the data types of the 145 columns in the merged df.

Name column	Min-value	Max-value	Dtype
RECORD_	1	4445	float64
CASEID_	1	4445	float64
ID	101	601	float64
IDAA	1	9243	float64
GEBJAAR	1926	2013	float64
GESLACHT	0	1	float64
SCRDIAGEIND	0	20	float64
CRIT01HPIJN	0	8	float64
CRIT02HPIJN	0	12	float64
CRIT03HPIJN	0	12	float64
CRITTEMPHPIJN	0	1	float64
CRIT04HPIJN	0	2	float64
CRIT05HPIJN	0	1	float64
SCREENA	1	1	float64
SCREENC	0	10	float64
SCREENE	0	9	float64
SCREENJ	0	9	float64
SCREENM	0	9	float64
FAMC	0	9	float64
UITSLAG	0	1	float64
CRIT01	0	8	float64
CRIT03	0	12	float64
CRIT04	0	2	float64
HPIJNA	1	2	float64
HPIJNB	0	3	float64
HPIJNC	0	9	float64
HPIJND	1	7	float64
HPIJNE	1	6	float64
HPIJNF	0	93	float64
HPIJNG	0	93	float64
HPIJNAV	0	2	float64
BESCHRA	0	4	float64
BESCHRB	0	9	float64
BESCHRC	0	9	float64
BESCHRD	0	9	float64
BESCHRE	0	9	float64
BESCHRF	0	9	float64
BESCHRG	0	9	float64
BESCHRH	0	9	float64
BESCHRI	0	9	float64
BESCHRJ	0	9	float64
BESCHRK	0	9	float64
BESCHRL	0	9	float64
BESCHRM	0	9	float64
HUIDA	0	9	float64
HUIDB	0	9	float64
HUIDC	0	9	float64
HUIDD	0	9	float64
HUIDE	0	9	float64
HUIDF	0	9	float64
HUIDG	0	9	float64
HUIDH	0	9	float64
HUIDI	0	9	float64
HUIDJ	0	9	float64
HUIDK	0	9	float64
HUIDL	0	9	float64
BESCHBOBSOLETE	0	8	float64
BESCHBC	0	9	float64
BESCHBD	0	9	float64
BESCHBE	0	9	float64
BESCHBF	0	9	float64
BESCHBG	0	9	float64
BESCHBH	0	9	float64
BESCHBI	0	9	float64
BESCHBJ	0	9	float64
BESCHBK	0	9	float64
BESCHBL	0	9	float64
SYMPB	0	9	float64
SYMPC	0	9	float64
SYMPD	0	9	float64
SYMPE	0	9	float64
SYMPF	0	9	float64
SYMPG	0	9	float64
SYMPH	0	9	float64
SYMPI	0	9	float64
SYMPJ	0	9	float64
SYMPL	0	9	float64
SYMPK	0	9	float64
SYMPM	0	9	float64
SYMPN	0	9	float64
SYMPO	0	9	float64
SYMPP	0	9	float64
SYMPQ	0	9	float64
SYMPR	0	9	float64
MEDICA	0	2	float64
MEDICBA	0	1	float64
MEDICBJ	0	1	float64
MEDICBB	0	1	float64
MEDICBP	0	1	float64
MEDICZ01	0	31	float64
MEDICZ02	0	31	float64
MEDICZ03	0	31	float64
MEDICZ04	0	31	float64
INTOX01	0	1	float64
INTOX02	0	1	float64
INTOX03	0	1	float64
INTOX04	0	1	float64
INTOX05	0	1	float64
ROKEN1	0	2	float64
ROKEN5	0	2	float64
ROKEN9	-3.7	153.1	float64
CAFEINE1	0	1	float64
CAFEINE2	0	15	float64
CAFEINE3	0	1	float64
CAFEINE4	-1	20	float64
CAFEINE5	0	1	float64
CAFEINE7	0	1	float64
CAFEINE9	-3	28	float64
ALCOHOL01	0	1	float64
DRUGS01	0	2	float64
LEEFTIJD	0	83	float64
FAMILA	0	9	float64
FAMILB	0	9	float64
FAMBA	0	11	float64
FAMBB	0	7	float64
FAMBC	0	9	float64
FAMBD	0	6	float64
FAMBG	0	11	float64
AURA	0	9	float64
AURAB	0	9	float64
AURAC	0	9	float64
AURAD	0	9	float64
AURAE	0	9	float64
AURAF	0	9	float64
AURAG	0	9	float64
AURAH	0	9	float64
AURAI	0	9	float64
AURAJ	0	9	float64
AURAK	0	9	float64
AURAL	0	9	float64
AURAM	0	9	float64
AURAN	0	9	float64
AURAO	0	9	float64
AURAAV	0	9	float64
AURAEV	0	9	float64
AURAEA	0	9	float64
AURAFA	0	9	float64
AURAFB	0	9	float64
AURAFC	0	9	float64
AURAFD	0	9	float64
AURAFE	0	9	float64
AURAGA	0	99	float64
AURAHA	0	3	float64
AURAHB	0	9	float64
CRITFINAL	0	2	float64
CRIT04HPIJNCLASS	0	1	float64
CRIT04HPIJNAURA	0	1	float64