Østfold University College

# MASTER'S THESIS

## Machine learning models and thrombosis: An exploration into predicting bleeding events

Daniel Nikolai Fiko

July 6, 2023

Master in Applied Computer Science
Faculty of Computer Sciences, Engineering and Economics

# Machine learning models and thrombosis: An exploration into predicting bleeding events

Masters thesis

Daniel Nikolai Fiko

Department of Computer Science and Communication
Østfold University College
Halden
July 6, 2023

# Abstract

This thesis explores the suitability of machine learning (ML) applications to predict bleeding events on a dataset collected from patients with thrombosis and contemporary history of a cancer diagnosis. Simultaneously, it compares the effectiveness of various ML models in processing this data.

The study seeks to understand to which extent this dataset is sufficient to differentiate classes using state-of-the-art ML methods. Twelve different models' performances were evaluated, revealing substantial variations across models. These models included DecisionTreeClassifier, ExtraTreeClassifier, ExtraTreesClassifier, GaussianNB, KNeighborsClassifier, LinearSVC, MLPClassifier, NearestCentroid, QuadraticDiscriminantAnalysis, RadiusNeighborsClassifier, RandomForestClassifier, and RidgeClassifier. Following an in-depth evaluation, three top-performing models — MLPClassifier, DecisionTreeClassifier, and ExtraTreeClassifier — were identified based on the selected performance metric. The initial results did not meet the expectations.

Consequently, several variations of the dataset were generated to better investigate the suitability of the data for predicting bleeding events. Afterward, an extended hyperparameter tuning process was performed on the three selected top-performing models to enhance their predictive performance further. The findings suggest that while no single model consistently outperforms others across all metrics, careful selection and hyperparameter optimization can substantially improve prediction accuracy.

The study underscores the challenges of applying ML in healthcare, particularly given the constraints of available data and the necessity of thorough model optimization. Despite these complexities, this research contributes to understanding ML applications in predicting bleeding events in thrombosis patients. It provides helpful information for enhancing the accuracy of predictions that could improve individualized treatment strategies for these patients.

# Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, whose expert knowledge, dedication, and patience have guided me through the most challenging parts of this process. Your constant support, insightful feedback, and conviction that things will work out have been a big part of shaping this research.

I want to thank my loving parents, whose support and encouragement have been significant in my journey toward this master's degree. Your faith in my abilities has been a source of continuous strength and motivation. Thank you for everything.

And a big thanks to my friends and family; your support and companionship have been invaluable. Your shared laughter and moments of reassurance provided a much-needed pause and made this journey much more bearable.

Lastly, a particular word of thanks goes to my girlfriend, who has been a source of constant support and love. Your patience, understanding, and belief in my capabilities, even in the face of challenges, have given me the strength to continue when things get tough. Your love and encouragement have been my pillar of strength in times of struggle. Thank you for all your love and support through all of this.

Ten years ago, the thought of me handing in a master's thesis seemed unlikely to many, myself included. I am deeply grateful for the collective effort of everyone who has played a part in this enormous accomplishment. There were times when the end seemed far away, but I have grown and learned an incredible amount through all the obstacles. Reflecting on the journey, I am incredibly proud of how far I have come. Thank you all.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The application of machine learning (ML) in healthcare has been an area of increasing interest due to its potential to transform patient care, enabling personalized treatments and facilitating early detection of diseases. One such area of application is the prediction of bleeding in thrombosis patients, a significant clinical challenge that has far-reaching implications for patient management. In this context, this study explores the use of ML to predict bleeding events in thrombosis patients. However, the focus of this study extends beyond just the application of ML techniques. It also seeks to answer the crucial question: 'Is the dataset sufficient in size and quality to utilize machine learning methods effectively?' Furthermore, it involves comparing various machine learning models in terms of their effectiveness.

## 1.1 Motivation and research problem

Thrombosis, the formation of blood clots within the blood vessels, presents a significant healthcare challenge. While anti-coagulation therapies mitigate the risk of clot formation, they carry a substantial risk of inducing bleeding in thrombosis patients. Thus, predicting the risk of bleeding in these patients is critical, necessitating a robust, accurate, and timely approach.

The emergence of ML techniques, which can handle and derive patterns from complex, multidimensional data, has opened new avenues for addressing this issue. However, the application of ML in this context is underexplored, particularly in the comparative analysis of different ML techniques.

Further complicating matters, there is generally a lack of sufficient data available to train models for predicting bleeding in thrombosis patients. For this study, Østfold Hospital has provided a dataset to be investigated in this context.

This study takes on the clear need for systematic research to identify the most suitable ML techniques for predicting bleeding in thrombosis patients. In addition, it investigates the question of whether the dataset at hand is sufficient to utilize ML methods for this purpose effectively.

## 1.2 Objective of the research

The main objective of this study is to investigate the performance of several standard ML techniques. This study will focus on their applicability for predicting bleeding

in thrombosis patients with a concurrent history of cancer diagnosis using a dataset provided by Østfold Hospital. Each of these models, with its unique characteristics and mathematical foundations, will provide a diverse set for comparative analysis. The study will evaluate the models based on the F1 macro score, reporting several other performance metrics such as accuracy, precision, recall, F1 score, and ROC AUC score. Additionally, the study will touch upon the intricacies of hyperparameter tuning, bettering our understanding of its impact on model performance and its importance for the problem.

**Research question I** To what extent does the dataset allow for the differentiation of classes with state-of-the-art machine learning methods?

**Research question II** How do standard machine learning models compare in their effectiveness in predicting bleeding events in thrombosis patients?

## 1.3 Structure of the thesis

This thesis is divided into several chapters:

- Chapter 2: **Background** - This chapter elaborates on thrombosis, its implications, and current approaches for predicting bleeding in thrombosis patients.
- Chapter 3: **Related work** - This chapter will review previous work in the domain, focusing on ML applications in healthcare prediction and describing the models under consideration.
- Chapter 4: **Method** - This chapter will describe the dataset used, the preprocessing procedure, and the model evaluation and comparison techniques.
- Chapter 5: **Results** - This chapter will present the comparative analysis results, highlighting each model's performance against various metrics.
- Chapter 6: **Discussion** - This chapter will interpret the results, discuss the implications of the findings, and provide a comprehensive comparison of the models.
- Chapter 7: **Conclusion** - The final chapter will summarize the research findings, discuss the contributions to the field, and suggest potential paths for future work.

By the end of this study, the aim is to provide a comprehensive understanding of applying ML techniques in predicting bleeding risk in thrombosis patients. This comparative study will facilitate informed model selection and optimization, improving patient outcomes.

# Chapter 2

# Background

This chapter presents the topics to understand the scope of this study. It starts with an analysis of thrombosis, its complications, and its societal implications. The existing methods for predicting bleeding in thrombosis patients are then discussed, followed by an introduction to the selected machine learning algorithms used in the study. Finally, the concepts for model evaluation - performance metrics, cross-validation, and the confusion matrix - are defined to provide the necessary background for understanding the study's analysis and results.

## 2.1 Machine learning algorithms

The selection of suitable machine learning algorithms is an essential part of this study. Each algorithm has unique attributes that make it well-suited for specific tasks, and its performance can vary widely depending on the nature of the data. This section dives deeper into the machine learning algorithms applied in this study. A brief overview of each algorithm is presented. The reasoning for selecting these particular algorithms is discussed in chapter 4. Also, the parameter tuning process, which is fundamental for enhancing the performance of these algorithms, is elaborated in chapter 4. The chosen algorithms span from traditional ones, like decision trees, to ensemble methods, like random forests, and more complex models, like neural networks.

#### DecisionTreeClassifier

This type of supervised learning algorithm is primarily used in classification problems. It works by creating a decision tree model that predicts the value of a target variable by learning simple decision rules inferred from the data features (Mitchell, 1997, p. 52). An illustration of a simple decision tree is depicted in figure 2.1.

#### ExtraTreeClassifier

The ExtraTreeClassifier is another type of decision tree algorithm. Unlike traditional decision trees, it introduces randomness by choosing split points completely at random (Geurts et al., 2006). This provides a way to reduce the variance of the model at the cost of a slight increase in bias, often resulting in a better overall performance.

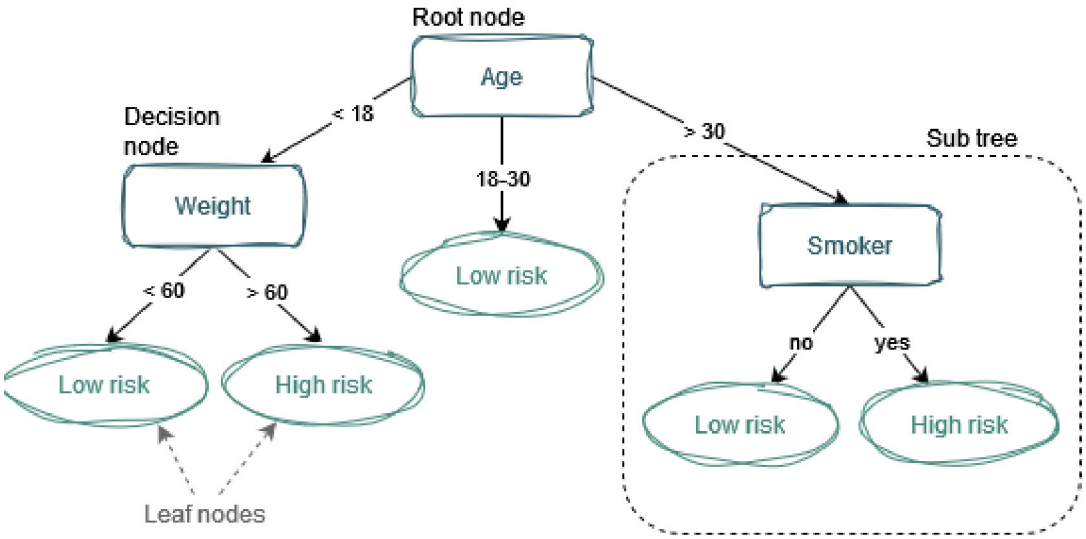



Figure 2.1: Illustration of a small decision tree and its components.

### ExtraTreesClassifier

The ExtraTreesClassifier is a type of ensemble learning technique. It constructs multiple decision trees during training and outputs the class representing the mode (most frequent) of the classes for classification tasks. It is referred to as ”extra random” because, unlike other tree-based algorithms, it picks cut points for each feature at random rather than choosing the best possible split (Geurts et al., 2006). This introduces more randomness into the model, helping to reduce the correlation between individual trees and making the model less prone to overfitting.

### GaussianNB

The GaussianNB, or Gaussian Naive Bayes, is an algorithm that applies the principles of Naive Bayes with an assumption of normal (Gaussian) distribution (scikit-learn, 2023b). This classifier assumes that the value of a particular feature is independent of the value of any other feature, hence the term ”naive.”

### KNeighborsClassifier

This is a type of instance-based learning or non-generalizing learning. The algorithm stores all instances corresponding to training data in n-dimensional space (Mitchell, 1997, p. 232). It then classifies new instances based on their distance from existing ones, taking the K-nearest ones into account.

### LinearSVC

This linear approach to support vector classification can handle both dense and sparse input. Like all Support Vector Machines, it aims to separate the data by finding the hyperplane with the largest margin (Kuhn & Johnson, 2013, p. 151).

### MLPClassifier

The MLPClassifier, short for Multi-Layer Perceptron Classifier, is a feed-forward artificial neural network model that maps input data sets onto a set of outputs (scikit-learn, 2023a). As illustrated in figure 2.2, the MLP consists of multiple layers of nodes (neurons) in a directed graph, with each layer fully connected to the next one. 'MLPClassifier' has many parameters to adjust and optimize, allowing flexibility to model non-linear dependencies in the data.

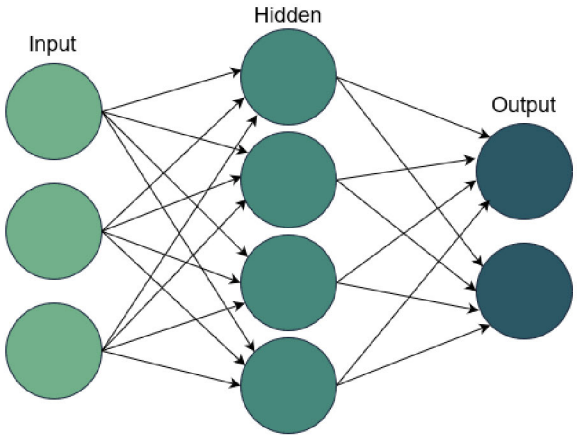



Figure 2.2: Illustration of multi-layer perception with one hidden layer and four neurons.

### NearestCentroid

This is a simple algorithm that represents each class by the centroid, also known as the geometric center, of its members (scikit-learn, 2023e). It then classifies instances by finding the class with the closest centroid.

### QuadraticDiscriminantAnalysis

This classifier fits class conditional densities to the data and uses Bayes' rule to compute probabilities (Kuhn & Johnson, 2013, p. 330). Unlike 'LinearDiscriminantAnalysis,' it does not assume that the covariance of each class is identical, allowing for more flexibility.

### RadiusNeighborsClassifier

Similar to the 'KNeighborsClassifier,' instead of considering the k-nearest neighbors, it considers all neighbors within a certain radius to classify instances (Mitchell, 1997, p. 233).

### RandomForestClassifier

This is a robust and versatile classifier that fits several decision tree classifiers on different sub-samples of the dataset and uses averaging to improve the prediction accuracy and control over-fitting (Kuhn & Johnson, 2013, p. 386).

### RidgeClassifier

This classifier learns a separate regression model for each class in multiclass scenarios. After converting the target values to -1, 1, the classifier treats the problem as multi-output regression. The model predicts a continuous value for each class, and the class with the highest predicted value is selected as the output class (scikit-learn, 2023d)..”

## 2.2 Model evaluation

Model evaluation metrics provide an essential way to quantify the performance of predictive models in machine learning. Evaluation metrics are how the quality of predictions can be measured and provide important insight into how robust and generalizable a model is. These metrics are essential in tuning model parameters, selecting the appropriate machine learning algorithm, and ultimately understanding how well the model will predict new, unseen data. This section covers the various model evaluation metrics utilized in this study. These include accuracy, precision, recall, F1 score, and area under the Receiver Operating Characteristic (ROC) curve (AUC ROC), each having its own specific interpretation and use case. The confusion matrix and cross-validation are also explained, providing an essential context for assessing model performance.

Precision, recall, and F1 score are reported using a weighted average to give a sense of the classifier's effectiveness when considering class imbalance. It is computed by taking the average of the scores for each class, with the average being weighted by the number of instances in each class (scikit-learn, 2023c). $w_i$ is the weight for each class $i$, $\text{Precision}_i$, $\text{Recall}_i$ and $\text{F1 score}_i$ are the precision, recall, and F1 score for each class $i$, respectively.

$$w_i = \frac{\text{Number of samples in class } i}{\text{Total number of samples}} \tag{2.1}$$

### Accuracy

*Accuracy* is the ratio of correctly predicted instances to the total instances in the dataset.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions made}} \tag{2.2}$$

### Precision

*Precision* is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{false positives}} \tag{2.3}$$

$$\text{Weighted Precision} = \sum_{i=1}^{N} w_i \times \text{Precision}_i \tag{2.4}$$

### Recall

*Recall* is the ratio of correctly predicted positive observations to all observations in the actual class.

$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{false negatives}} \tag{2.5}$$

$$\text{Weighted Recall} = \sum_{i=1}^{N} w_i \times \text{Recall}_i \tag{2.6}$$

### F1 score

The *F1 score* is the weighted average of precision and recall. It tries to find the balance between precision and recall.

$$\text{F1 score} = 2 \cdot \frac{\text{Precision} \cdot \text{recall}}{\text{Precision} + \text{recall}} \tag{2.7}$$

$$\text{Weighted F1 score} = \sum_{i=1}^{N} w_i \times \text{F1 Score}_i \tag{2.8}$$

### F1 macro

The *F1 macro score*, or simply F1 macro, is a measure used to assess a model's performance on multiclass classification problems. It is beneficial when the data classes are imbalanced. The F1 macro is the unweighted average of the F1 scores of each class, treating all classes equally regardless of their proportions in the dataset.

$$\text{F1 macro} = \frac{1}{N} \sum_{i=1}^{N} F1_i \tag{2.9}$$

Where $N$ is the number of classes, and $F1_i$ is the F1 score of each class.

### AUC ROC

*The Receiver Operating Characteristic* (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold values and essentially separates the 'signal' from the 'noise.' The Area Under the Curve (AUC) measures a classifier's ability to distinguish between classes and is used as a summary of the ROC curve.

This concept does not directly apply to multiclass problems, so to use this, a variation of the AUC ROC is applied to compare multiple classes; One versus One (OvO). The OvO technique calculates the average AUC ROC of all possible pairwise class combinations.

### Cross-validation

*Cross-validation* is a technique in machine learning that assesses how well a model can generalize to unseen data. It works by partitioning the dataset into several subsets or "folds," then training the model on all but one fold and testing it on the remaining fold. This process is repeated for each fold, providing a more robust measure of model

performance. Since this study is dealing with an unbalanced dataset, stratified k-fold cross-validation is utilized. This ensures that each class is equally represented across all folds, as illustrated in figure 2.3.

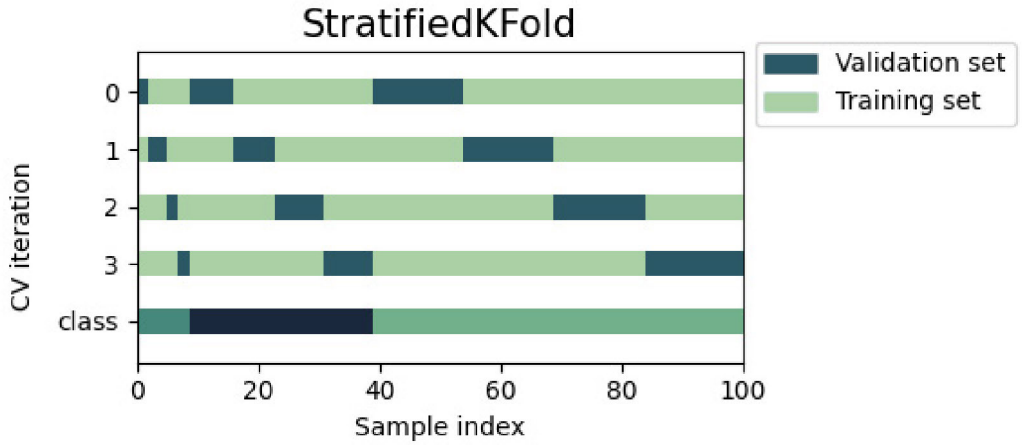



Figure 2.3: Illustration of the partitioning of a dataset into four folds using stratified k-fold cross-valdiation.

### Confusion matrix

A *confusion matrix* is a table used in statistics and machine learning to visualize the performance of a classification model. It is a $2 \times 2$ matrix ($n \times n$ for multiclass problems), and is used to compute several performance metrics, such as accuracy, precision, recall, and F1 score, and can help highlight model strengths and weaknesses. The confusion matrix is made up of four components:

1. True Positives (TP): Correctly predicted positive classes.
2. True Negatives (TN): Correctly predicted negative classes.
3. False Positives (FP): Incorrectly predicted positive classes.
4. False Negatives (FN): Incorrectly predicted negative classes.

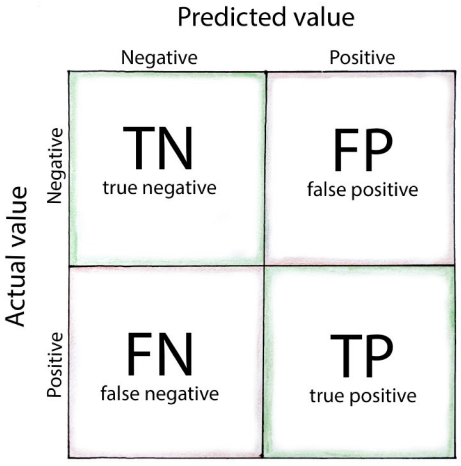



Figure 2.4: The structure of the confusion matrix.

## 2.3 Thrombosis and its societal impacts

*Thrombosis* is a general term for a blood clot forming within and blocking a blood vessel, as illustrated in figure 2.3. It can occur in any part of the vascular system, affecting veins and arteries (Kyrle & Eichinger, 2005).

Blood clots in the *veins*, the blood vessels responsible for returning blood from the body to the heart, are known as venous thrombosis. Conversely, when thrombosis occurs in the arteries, it is termed arterial thrombosis. *Arteries* are the blood vessels that transport oxygen-rich blood from the heart to the rest of the body (Johns Hopkins Medicine, 2019).

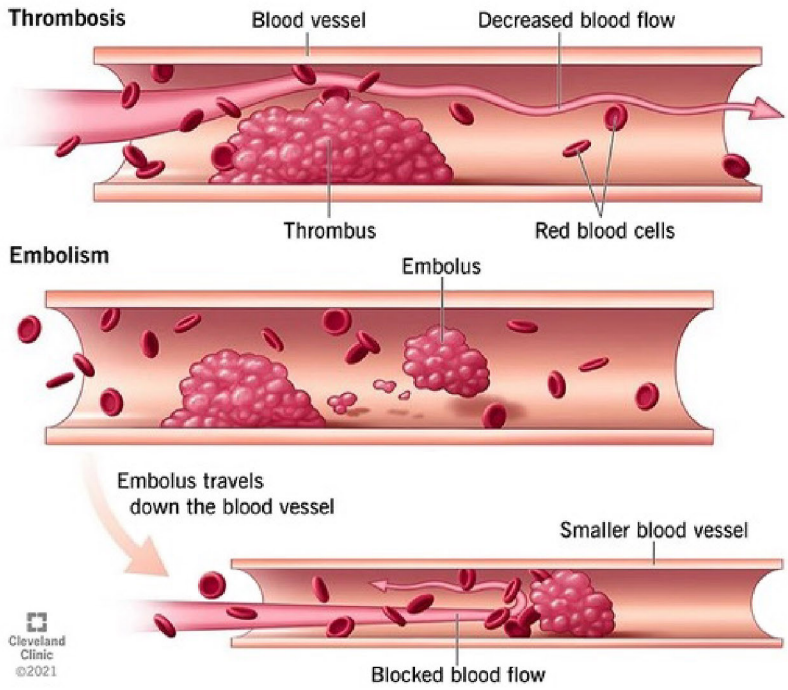



Figure 2.5: "How thrombosis can lead to a blocked blood vessel." From Cleveland Clinic. (2023, January 16). *Thrombosis illustration.* https://my.clevelandclinic.org/health/diseases/22242-thrombosis. CC Cleveland Clinic.

### Deep vein thrombosis (DVT)

This type of venous thrombosis occurs when a blood clot forms in one of the body's deep veins, usually in the legs. *DVT*s can cause pain and swelling but can also occur without symptoms (Kyrle & Eichinger, 2005). The primary concern with DVT is the risk of the clot dislodging, traveling to the lungs, and causing a pulmonary embolism (PE).

### Pulmonary embolism (PE)

*PE* is a potentially life-threatening condition when a blood clot, usually originating from a DVT, travels from where it was formed and lodges in the pulmonary arteries (National Heart, Lung, and Blood Institute, 2022). If not treated quickly, this can lead to severe issues like shortness of breath, chest pain, and sudden death.

**Venous thromboembolism (VTE)**

*VTE* is a condition that includes both DVT and PE (National Heart, Lung, and Blood Institute, 2022). It is essentially the formation of blood clots in the vein (DVT) that can dislodge and travel to the lungs (PE).

### 2.3.1 Complications and consequences of thrombosis

Thrombosis can lead to severe complications and adversely impact a patient's health and quality of life (Kahn et al., 2002). When the body forms a blood clot in a vein or artery (thrombosis), it can obstruct blood flow, leading to severe consequences if not treated promptly.

Post-thrombotic syndrome (PTS) is a common complication of venous thrombosis. This condition often arises after an episode of DVT (Kahn et al., 2002). PTS is characterized by chronic pain, swelling, and discomfort in the affected limb, often leading to significant disability and reduced quality of life (Stain et al., 2005). Over time PTS can also cause skin changes and leg ulcers.

In addition to physical health complications, thrombosis and its aftermath can significantly impact a patient's mental health and overall quality of life. Dealing with chronic pain, disability, and the constant fear of a recurrent episode can lead to psychological distress, including anxiety and depression (Kahn et al., 2002; Fischer et al., 2023).

Bleeding complications and the subsequent consequences of thrombosis significantly contribute to the severity and complexity of this condition. Bleeding complications often arise as a side effect of anticoagulant treatment, the primary therapy for thrombosis (Beckman et al., 2010). While these medications effectively prevent clot formation, they can also enhance the patient's risk of bleeding (Prandoni et al., 2002). This can lead to potentially serious complications ranging from minor to life-threatening, major bleeding. The risk of these complications increases with the intensity and duration of anticoagulant therapy (Palareti et al., 1996), as illustrated in figure 2.6.

Another serious complication of venous thrombosis is pulmonary embolism (PE), as discussed in the previous section. Even if a PE is survived, it can lead to chronic leg pain and swelling, known as post-thrombotic syndrome and chronic thromboembolic pulmonary hypertension (CTEPH) (Tapson, 2008).

### 2.3.2 Societal impacts of thrombosis

Thrombosis is a significant financial and healthcare burden globally (Beckman et al., 2010; Fernandez et al., 2015). This burden is affecting patients, healthcare systems, and society at large.

From a financial perspective, the costs associated with thrombosis treatment are substantial. These costs include diagnostic testing, hospitalization, outpatient care, anticoagulant therapies, and treatment of complications. Notably, the treatment expenses have been on an upward trajectory, exceeding the general rate of medical care inflation (Fernandez et al., 2015). Indirect costs, such as loss of productivity due to work absence or disability, further add to the economic burden (Brækkan et al., 2016).

Regarding healthcare, thrombosis has a notable impact on hospital resources and capabilities. The condition often necessitates extended hospital stays, intensive treatment

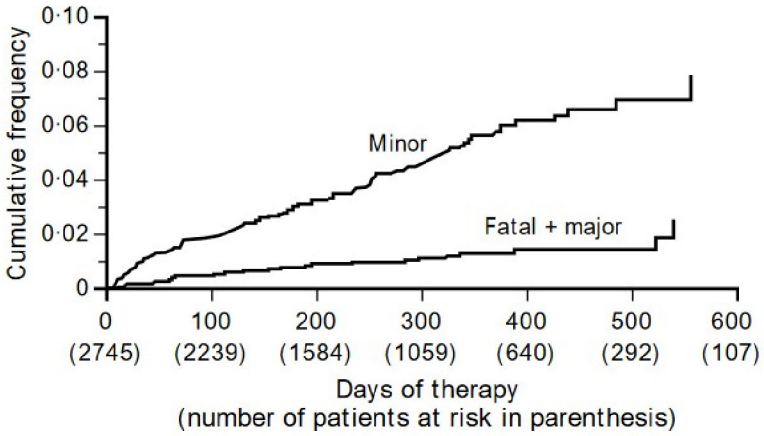



Figure 2.6: "Cumulative frequency (Kaplan-Meier curves) of fatal plus major and of minor bleeding events during outpatient anticoagulant treatment." From Palareti, G., Leali, N., Coccheri, S., Poggi, M., Manotti, C., D'Angelo, A., Pengo, V., Erba, N., Moia, M., Ciavarella, N., Devoto, G., Berrettini, M., & Musolesi, S. (1996). Bleeding complications of oral anticoagulant treatment: An inception-cohort, prospective collaborative study (ISCOAT). *The Lancet*, *348*(9025), 423–428. https://doi.org/10.1016/S0140-6736(96)01109-9. CC authors.

regimens, and long-term follow-up care, increasing strain on already stretched healthcare facilities (Fernandez et al., 2015). It also contributes to higher readmission rates and long-term morbidity, which place additional pressure on healthcare services (Kyrle & Eichinger, 2005; Fernandez et al., 2015).

Thrombosis also has broader societal implications. In the workforce, employees affected by thrombosis may face prolonged absence or disability, resulting in reduced productivity and economic output (Brækkan et al., 2016). As illustrated in figure 2.7, the condition often affects individuals in their prime working years, leading to premature withdrawal from the labor market. This can have a direct economic impact and cause disruptions to businesses and industries.

The social impacts of thrombosis also affect families, communities, and social networks, with psychological and social consequences that include stress, depression, and reduced quality of life. Furthermore, the condition carries a stigma that can lead to social isolation (Jøssang, 2020; Fischer et al., 2023).

In summary, thrombosis imposes a significant financial and healthcare burden with extensive societal consequences. Notably, bleeding complications pose a significant challenge, often as a side effect of anticoagulant treatment - the primary therapy for thrombosis. The risk of minor to major bleeding escalates with the duration and intensity of this therapy.

## 2.4 Current approaches for predicting bleeding in thrombosis patients

Traditional clinical assessment is often the first approach, with clinicians considering patient characteristics and medical history. Factors such as age, history of previous

| | No work-related disability after VTE ($n$ = 312) | Work-related disability after VTE ($n$ = 72) | Work-related disability after VTE* ($n$ = 53) |
|---|---|---|---|
| Age | 50.5 ± 10.2 | 55.5 ± 6.0 | 55.1 ± 6.0 |
| Sex (% men) | 145 (46.5) | 35 (48.6) | 24 (45.3) |
| Pulmonary embolism | 112 (35.9) | 20 (27.8) | 15 (28.3) |
| Deep vein thrombosis | 200 (64.1) | 52 (72.2) | 38 (71.7) |
| Unprovoked | 138 (44.2) | 36 (50.0) | 34 (64.0) |
| Provoked | 174 (55.8) | 36 (50.0) | 19 (36.0) |

Values are means ± standard deviations in brackets or numbers with percentages in brackets. *When the date of disability was set 1 year before the actual date of disability pension.

Figure 2.7: "Characteristics of patients with venous thromboembolism (VTE) with and without work-related disability." From Brækkan, S. K., Grosse, S. D., Okoroh, E. M., Tsai, J., Cannegieter, S. C., Næss, I. A., Krokstad, S., Hansen, J.-B., & Skjeldestad, F. E. (2016). Venous thromboembolism and subsequent permanent work-related disability. *Journal of Thrombosis and Haemostasis*, *14*(10), 1978–1987. https://doi.org/10.1111/jth.13411. CC Wiley Online Library and authors.

bleeding, kidney or liver disease, and alcohol or drug abuse, among others, are typically taken into account (De Winter et al., 2021).

Risk scoring systems have been developed to standardize and improve this assessment. Examples of such systems include the HAS-BLED and VTE-BLEED scores (Kooiman et al., 2015; Badescu et al., 2021; De Winter et al., 2021). These scoring models consider various risk factors to calculate an estimated bleeding risk. They offer a structured way to assess bleeding risks and have been shown to have predictive value.

However, although these traditional methods are helpful, they also have notable limitations. Risk scoring systems, for instance, often consider only a limited set of factors, neglecting the complexity and the multifactorial nature of bleeding risk. Traditional clinical assessments are subjective and dependent on the clinician's expertise and experience. They can therefore vary in accuracy and consistency.

These limitations indicate a need for more accurate and efficient predictive methods. With the advancements in medical technologies, techniques like machine learning are now being explored to better predict the risk of bleeding in patients with thrombosis (Mora et al., 2023; Shohat et al., 2023). This technology can consider a broader range of factors, learn from large datasets, and potentially provide more personalized risk prediction. This is a promising field, but it is still in its early stages.

# Chapter 3

# Related work

The application of machine learning to healthcare problems has significantly increased in the last few years. Advancements in algorithmic techniques drive this growth, greater healthcare data availability, and the increasing need for more efficient and personalized medical services (Jiang et al., 2017). The field of machine learning in healthcare has matured to a point where several of these approaches have been incorporated into routine clinical practice, delivering significant benefits to patient care (Alanazi, 2022).

This study has been focusing on using machine learning techniques to predict bleeding in patients with thrombosis in this growing field. This chapter presents an overview of machine learning applications in healthcare.

## 3.1 Overview of machine learning applications in healthcare

**Machine learning predicts cancer-associated deep vein thrombosis using clinically available variables.**

Jin et al., 2022 developed five ML models for cancer-associated DVT and compared the results with the Khorana score. Significant predictors were selected from randomly extracted data from about 3000 patients, and models were trained on 70% of the data. Linear discriminant analysis and logistic regression were the only machine learning models that outperformed the Khorana score. A combination with the D-dimer feature showed improved performance in all models.

**Machine learning to predict venous thrombosis in acutely ill medical patients**

A study assessing the performance of various machine learning models in relation to the IMPROVE score was conducted by Nafee et al., 2020. The IMPROVE (International Medical Prevention Registry on Venous Thromboembolism) is a recognized and verified score utilized for assessing the risk levels of acute, medically ill patients. Using the APEX trial dataset with 7,513 acutely medically ill patients, including 68 attributes, they developed a super learner model to predict venous thromboembolism by combining estimates from 5 families of candidate models. Using 39 machine learning models in 5 families of models and 10-fold cross-validation, they developed a super learning model to predict VTE, and a reduced model (rML) was also developed using 16 variables that were assumed to be associated with VTE. The candidate models' families included generalized additive models, elastic net (penalized logistic regression), extreme gradient

boosting, random forests, a Bayesian logistic regression with default priors, and a simple classification tree. Their results show that both the machine learning and rML models outperformed the IMPROVE score, with c-statistics of 0.69 and 0.68, respectively, compared to 0.59 for the IMPROVE score as illustrated in figure 3.1. This comparison shows that the machine learning and rML models were more effective in predicting VTE than the traditional IMPROVE score.

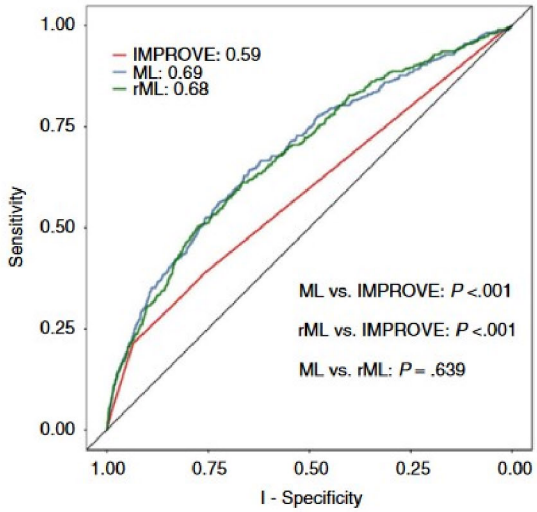



Figure 3.1: ROC curve showing the performance of ML, rML, and IMPROVE risk score in predicting combined VTE outcome. From Nafee, T., Gibson, C. M., Travis, R., Yee, M. K., Kerneis, M., Chi, G., AlKhalfan, F., Hernandez, A. F., Hull, R. D., Cohen, A. T., Harrington, R. A., & Goldhaber, S. Z. (2020). Machine learning to predict venous thrombosis in acutely ill medical patients. *Research and Practice in Thrombosis and Haemostasis*, *4*(2), 230–237. https://doi.org/10.1002/rth2.12292. CC BY-NC-ND.

**Using machine learning to predict venous thromboembolism and major bleeding events following total joint arthroplasty.**

In this study, Shohat et al., 2023 aim to create and validate a machine learning model to predict the likelihood of VTE and major bleeding events (MBE) in patients following total joint arthroplasty to support clinical decision-making.

The dataset consists of 35,963 primary and revision total joint arthroplasty patients operated between 2009 and 2020 from a single institution. 56 variables, including demographics, comorbidities, operative factors, and chemoprophylaxis, were included in the data. The researchers manually reviewed patient notes to determine the type of VTE prevention used postoperatively and the anticoagulant taken preoperatively. They applied descriptive statistics to understand data distributions and compare patients with VTE or MBE. They identified variables that increased the likelihood of developing VTE or MBE using random forest, lasso, gradient boosting trees, and support vector machines.

MBE models were tested using repeated cross-validation, with the Lasso analysis showing the highest AUC and being chosen for MBE algorithm development. The Lasso analysis revealed the ten most important factors for MBE were revision surgery, chronic

use of Warfarin preoperatively, operative duration, general anesthesia, peptic ulcer disease, allogenic blood transfusions, older age, knee joint, varicose veins, and current or past smoking.

Although the study has some limitations due to the low event rates, as there are only 308 VTE and 293 MBE patients out of the total 35,963 patients, it presents a valuable predicting model for VTE and MBE risk.

**Artificial intelligence in healthcare: Past, present and future.**

The study by Jiang et al., 2017 surveys the current status of artificial intelligence (AI) applications in healthcare, mainly concentrated on cancer, nervous system disease, and cardiovascular disease. Their findings show that support vector machines and neural networks are used in about two-thirds of the medical application of machine learning. The IBM Watson system includes both machine learning and natural language processing (NLP), showing results that are 99% coherent with the physician's decision. One can utilize narrative text and extract relevant information for machine learning prediction by implementing NLP. Two hurdles are identified, the first being regulations lacking standards, making it difficult to get approvals and assess the safety of AI systems. The second is continually accessing data to improve and develop the system after the initial training.

**Using machine learning for healthcare challenges and opportunities**

Alanazi, 2022 presents a comprehensive analysis of machine learning in healthcare, discussing various machine learning techniques and their applications in different health sectors. Machine learning algorithms have demonstrated significant potential in healthcare for predictive analysis, decision support, and efficient patient care (Alanazi, 2022). Key machine learning techniques include linear regression, random forest, support vector machines, decision trees, LASSO regression, logistic regression, K-nearest neighbors, and Naïve Bayes classifier, each having unique strengths and specific constraints.

There are many applications of machine learning in modern healthcare. These include creating clinical decision support systems (CDSS) to help clinicians predict patient outcomes and identify anomalies. For example, ensemble models have been used to predict COVID-19 patient mortality risk, and CDSS has been employed to reduce prescribing errors.

However, implementing machine learning models in healthcare is not without challenges. These include the necessity for high-quality, representative data, maintaining the interpretability of machine learning predictions, and dealing with complex legal procedures. Ethical considerations are paramount, following the principle of "do no harm" and efforts to ensure algorithmic fairness and avoid reinforcing existing inequalities.

The author concludes by developing ethical guidelines for machine learning applications in healthcare, and methodological research must be included to ensure that machine learning models do not maintain inequalities.

## 3.2 Comparative studies on machine learning algorithms for predicting medical events

### Predicting thrombosis with machine learning

Abbas, 2021 presents a study comparing the performance of different machine learning techniques in predicting thrombosis. The comparison is conducted on a full and reduced dataset. The results in figure 3.2 show that XGBoost performs the best on the entire dataset, followed by random forests, support vector machines, and artificial neural networks. Random forests perform the best on the reduced dataset, followed by XGBoost, artificial neural networks, and support vector machines. Support vector machines showed the highest recall on the full dataset, followed by XGBoost, artificial neural networks, and random forests.

| **Testing Data** | | | | | |
|---|---|---|---|---|---|
| | Accuracy | Precision | Recall | F1 Score | AUC (PR) |
| RF | 92.74 | 83.63 | 75.40 | 79.31 | 85.70 |
| XGB | 94.56 | 80.28 | 93.44 | 86.36 | 86.61 |
| SVMs | 92.14 | 71.08 | 96.72 | 81.94 | 79.65 |
| ANNs | 88.82 | 66.21 | 80.32 | 72.59 | 77.79 |
| **Training Data** | | | | | |
| | Accuracy | Precision | Recall | F1 Score | AUC (PR) |
| RF | 99.92 | 99.59 | 100.0 | 99.79 | 99.99 |
| XGB | 98.33 | 91.72 | 100.0 | 95.68 | 99.98 |
| SVMs | 90.92 | 67.22 | 99.18 | 80.13 | 85.23 |
| ANNs | 94.09 | 76.77 | 97.54 | 85.92 | 94.55 |

Figure 3.2: From Abbas, K. (2021). Predicting thrombosis with machine learning. https://hdl.handle.net/11250/2770341. CC 2021 by author.

On the reduced dataset, XGBoost exhibits the highest recall, followed by support vector machines, random forests, and artificial neural networks. The study concludes that XGBoost is the most efficient algorithm for distinguishing between thrombotic and non-thrombotic patients.

### Comparison of machine learning algorithms for clinical event prediction

In the study, Beunza et al., 2019 tested various machine learning algorithms on the Framingham Heart Study database, a publicly available database that "originated in 1948 in Framingham, Massachusetts as a prospective study of risk factors for cardiovascular disease" (Beunza et al., 2019, p. 1) available on Kaggle. The aim was to predict the risk of coronary disease. A thorough data analysis was performed, checking for missing values and selecting eight key predictors for coronary risk using an automated "stepwise"

| | **R-Studio** | | | **RapidMiner** | | |
|---|---|---|---|---|---|---|
| | Model | | | Model | | |
| | A | B | C | A | B | C |
| Decision tree | 0.53 | 0.55 | 0.53 | 0.53 | 0.5 | 0.5 |
| Boosted decision tree | 0.53 | 0.60 | 0.52 | 0.67 | 0.7 | 0.69 |
| Random forests | NA | 0.63 | 0.59 | NA | 0.71 | 0.69 |
| Support vector machines | NA | 0.68 | 0.68 | NA | 0.75 | 0.71 |
| Neural network | NA | 0.71 | 0.68 | NA | 0.73 | 0.72 |
| Logistic regression | 0.5 | 0.68 | 0.68 | 0.68 | 0.73 | 0.73 |

Table 3.1: AUC score of the different algorithms with R-studio and RapidMiner. From Beunza, J.-J., Puertas, E., García-Ovejero, E., Villalba, G., Condes, E., Koleva, G., Hurtado, C., & Landecho, M. F. (2019). Comparison of machine learning algorithms for clinical event prediction (risk of coronary heart disease). *Journal of Biomedical Informatics*, *97*, 103257. https://doi.org/10.1016/j.jbi.2019.103257. CC Elsevier user licenses[2].

technique. Three distinct data models were created using data mining techniques for comparing algorithms. Model A used the original database, Model B excluded missing values, and Model C filled in missing values with averages. These models were then comparatively analyzed using the supervised machine learning algorithms, including decision trees, random forests, support vector machines, neural networks, and logistic regression. The positive labels were balanced using the ROSE library to boost the algorithm's predictive power, increasing coronary event prevalence from 15% to 50%. This study evaluated several performance metrics: accuracy, recall, specificity, precision, negative predictive value, and area under the curve (AUC). The overall selection criterion for deciding the best hyperparameters and the data manipulation method was based on the AUC. The highest AUC obtained was 0.71 using a neural network model with R-studio and 0.75 with a support vector machines model using RapidMiner as shown in table 3.1. The different machine learning algorithms had varying pros and cons. The decision tree, for example, was fast and could handle missing data but had low prediction power. The random forest and support vector machines improved accuracy but were more complex. The neural network model required the most programming time but offered the best results. Data normalization/standardization and balancing significantly improved results, especially in cases where the incidence of the predicted event was unbalanced.

[2]beta.elsevier.com/about/policies-and-standards/open-access-licenses/elsevier-user

# Chapter 4

# Method

This chapter will provide a detailed explanation of the methods and procedures adopted in this study to answer the research questions. It will discuss the dataset, the reasoning for selecting the particular ML techniques, the parameter tuning process, and the experimental setup.

## 4.1 Dataset overview

Østfold Hospital provided the dataset used in this study as part of a research project in collaboration with Østfold University College. The research project is investigating the possibilities of using artificial intelligence and ML to provide decision support for healthcare. This section will present a detailed exploration of the dataset to understand the information within a large number of features and patient records.

The dataset contains an extensive collection of healthcare records from 1080 patients, targeted explicitly toward thrombosis patients with a concurrent history of a cancer diagnosis and their risk factors. The data contains a lot of information about each patient, associated risks, and bleeding episodes. The language used in the dataset is primarily Norwegian.

Each record is related to an individual patient, distinguished by a unique ID. This allows for the identification of multiple records from the same individual. The hospital ensured that all data were thoroughly de-identified before being shared with Østfold University College, removing personal identifiers and assigning anonymized identification numbers to each patient. A total of 245 features are distributed across a selection of data types, shown in table 4.1. Each feature provides different kinds of information, including but not limited to:

**Patient demographic information:** These features include binary variables such as gender (male/female) and continuous variables like age and Body Mass Index (BMI).

**Clinical indicators:** These features provide medical data on the patient's health, such as SpO2%, PESI score, respiratory rate, pulse rate, blood pressure (both systolic and diastolic), CRP, and D-dimer result.

**Risk factors:** These include indicators such as smoking status and variables indicating different types of known thrombophilia, past medical conditions, or trauma events. It also contains information about the patient's travel history, hormone intake, surgical history, family disposition, etc.

**Treatment records:** This includes a range of treatment types and associated information such as start and end dates for different drugs (Dabigatran, Edoksaban, Apixaban, Rivaroxaban, Marevan, Dalteparin, Enoxaparin, etc.), use of thrombolytic therapy, and complications arising during the treatment.

**Bleeding episodes:** Variables like bleeding date and degree of bleeding capture information about bleeding episodes experienced by the patients. The degree of bleeding serves as the prediction label.

| **Features** | **Count** |
|---|---|
| Temporal | 112 |
| Binary | 64 |
| Numerical | 33 |
| Categorical | 31 |
| Text | 5 |
| **Total** | **245** |

Table 4.1: Distribution of feature datatypes in the unprocessed study dataset.

Finally, the dataset includes a total of 3778 records, categorized into four different classes: 'No bleeding,' 'Minor bleeding,' 'Clinically relevant minor bleed,' and 'Major bleeding,' their distribution is presented in table 4.2. This allows ML models to predict patient bleeding events based on a wide selection of factors. The data were preprocessed, cleaned, and transformed as necessary to ensure their suitability for ML algorithms.

Understanding the dataset's overall structure is critical for effectively modeling and accurately predicting patient bleeding events.

Early in the data exploration process, an essential step was taken to refine the dataset based on instructions received in a document from the hospital. This document served as a guide, providing context and specifics about the relevance and utility of features in the dataset. Following these guidelines, 152 columns were deemed unnecessary or irrelevant for predicting bleeding events and thus were removed from the dataset. This process of initial feature reduction significantly simplified the data set, reducing its dimensionality and focusing the analysis on the most relevant information.

In addition to the feature reduction, another cleaning step involved the removal of duplicated records. 870 records were found to be exact copies of other entries in the

| **Classes** | **Records** |
|---|---|
| No bleeding | 2967 |
| Minor bleeding | 90 |
| Clinically relevant minor bleed | 448 |
| Major bleeding | 273 |
| **Total** | **3778** |

Table 4.2: Distribution of records in the unprocessed study dataset by bleeding type.

dataset. These duplicated entries were dropped from the dataset to prevent any potential bias in the analysis.

This cleaning process transformed the dataset's structure. Table 4.3 illustrates this procedure's before and after comparison. After removing unnecessary features and duplicated rows, the dataset was narrowed down to 2908 records and 94 features.

| | **Original** | **Removed** | **Remaining** |
|---|---|---|---|
| **Records** | 3778 | 870 | 2908 |
| **Features** | 245 | 152 | 94 |

Table 4.3: Comparison of dataset structure before and after preliminary cleaning.

### 4.1.1 Relevance to research questions

With its complexity and variety, this dataset is suited to answer the research questions (RQs) posed in chapter 1. It offers a wide range of relevant features to the study and provides opportunities for exploration and analysis.

Considering RQ1, a range of numerical, categorical, temporal, and binary features offers a rich input to determine how much the dataset allows for class differentiation using state-of-the-art ML methods.

**The categorical features** in the dataset capture a range of patient-related information. For instance, the 'Diagnosis' feature contains one or multiple diagnoses related to thrombosis. 'Recent trauma ICD-10 code' is associated with traumatic events or injuries requiring medical intervention. Other features such as 'Orthopedic procedure code' and 'Other surgery procedure code' serve to identify the specific orthopedic or other surgical procedures the patient has undergone. The 'Trauma description' feature provides context to the incidents or conditions that might have contributed to a patient's current medical state.

The dataset also records the type of thrombophilia and any other relevant conditions under 'Thrombophilia diagnosis' and 'Other type.' Patient smoking habits are captured in the 'Smoking' feature. It classifies patients based on their smoking status: 'No,' 'Former,' 'Yes,' or, in instances where the information is missing, 'Unknown.' 'Other 2' and 'Contraceptives, hormones, systemic' provide insight into the types of birth control pills or hormonal treatments a patient might use.

Additional factors like varicose veins, obesity, and infections, among others, are cataloged under 'Other cause (specify).' The corresponding ICD-10 code is documented in 'Icd-10 code' for patients diagnosed with cancer. The 'Familial disposition' feature indicates whether the patient has a familial preposition for thrombosis, i.e., if there are instances of thrombosis among the patient's family members, further specified in 'Familial relationship.' 'Type of thrombolysis' and 'Tenecteplase / Alteplase' features describe the specific thrombolysis treatments administered to the patient. The patient's gender is indicated in the 'Gender' feature.

Lastly, the outcome variable, 'Degree of bleeding,' categorizes the type of bleeding experienced by the patient, dividing it into four distinct groups: No bleeding, Minor

bleeding, Major bleeding, and Clinically relevant minor bleed, as previously illustrated in table 4.2.

**Numerical features** in the dataset provide insight into each patient's health condition, physical attributes, and medical indicators, offering an objective framework to differentiate bleeding classes. Unique IDs are given to each patient for tracking individual records and maintaining patient anonymity. Diagnostic and oxygen saturation levels (SpO2% value) indicate oxygen availability in a patient's blood. At the same time, the Pulmonary Embolism Severity Index (PESI) score is a prognosis of the mortality risk for patients diagnosed with pulmonary embolism (Jiménez et al., 2010). Standard vital signs, such as respiratory and pulse rates, are also included, representing the number of breaths and heartbeats per minute.

Blood pressure readings are specified in diastolic and systolic values. Physical attributes are represented by Body Mass Index (BMI) and the patient's height and weight. Other laboratory values, such as 'CRP' and 'D-dimer values,' are included. 'Days hospitalized' reflects the number of days a patient was hospitalized in conjunction with a diagnosis. Lastly, the 'Age' feature informs us how old the patient is.

**Numerous temporal features** are present in the dataset. These features are primarily about the diagnosis and treatment of thrombosis. The dataset shows the chronological progression of the patient's diagnoses and treatments. For instance, it records the date when a primary diagnosis was made ('Diagnosis date') and the dates of any previous thrombosis diagnoses. Furthermore, the data also notes dates tied to specific activities, such as flights lasting between 4-8 hours.

One of the most detailed aspects of the dataset is the timeline of various treatments. The dataset records the start and end dates for numerous medications - including Dabigatran, Edoxaban, Apixaban, Rivaroxaban, Marevan, Dalteparin, Encosaparin, and others. These paired entries sketch out the distinct periods of medication usage, giving us a picture of the patient's treatment regimens. Finally, the dataset also identifies the date of bleeding events ('Bleeding event date').

**Binary features** in the dataset provide information about each patient's unique condition and risk factors. One group of these features is related to travel and mobility. For example, some patients have undergone recent trips involving long flights ('Flight >8h', 'Flight <4h', 'Flight 4-8h') or vehicle journeys ('Travel by vehicle >4h'), which are potential risk factors in thrombosis (Kyrle & Eichinger, 2005, p. 1164) as well as experienced immobilization due to a medical condition('Is immobilized').

Another group of binary features reflects the patient's medical history. This includes prior incidences of thrombosis ('Previous other thrombosis' and surgeries (both orthopedic and other) within the last 12 weeks, indicated in 'Orthopedic surgery last 12 weeks' and 'Other surgery in the last 12 weeks', respectively.

Some binary features touch upon the patient's current medication regime, such as birth control pills or hormone replacement therapy. Furthermore, certain binary features register important events during the patient's treatment. This includes bleeding complications during treatment and whether the patient received outpatient treatment.

Lastly, some features capture more specific situations, such as trauma requiring hospitalization, casting, or immobilization within the last 12 weeks. Collectively, these

binary features represent an extensive part of the data and contribute to understanding each patient's risk profile and medical background.

**Missing data** presented a significant challenge in the dataset, with approximately half of the features having 80% or more missing values. This degree of missing values is graphically represented in figure 4.1. Missing data inhibits class differentiation by reducing the availability of information for accurately distinguishing between classes and could potentially be a significant influence on the first RQ.

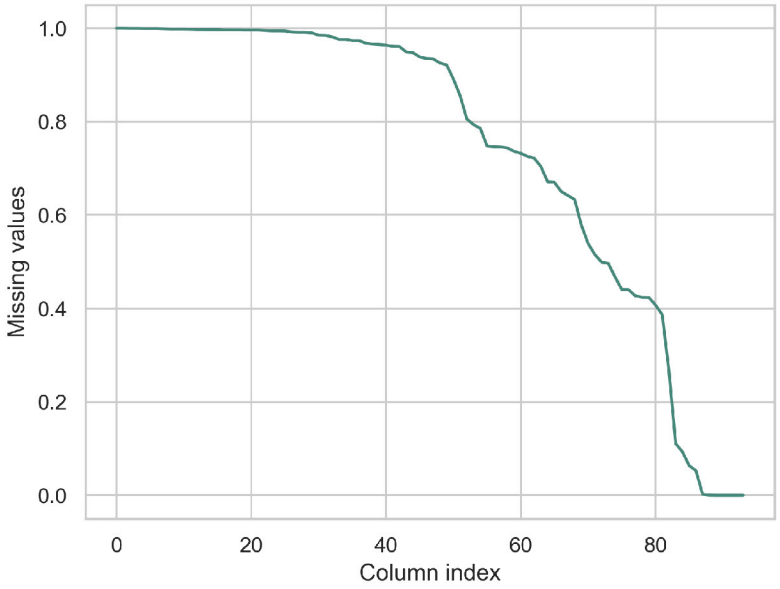



Figure 4.1: Lineplot illustrating the number of missing values across all the columns in the dataset.

RQ2 focuses on the comparative effectiveness of standard machine learning models in predicting bleeding events in thrombosis patients. The collection of features, particularly the classification of bleeding events in the dataset, is the foundation for systematically training, evaluating, and comparing the effectiveness of various machine learning models in their prediction task. Given the diversity of data, candidate models can be effectively trained and their predictive performance carefully compared. Therefore, the dataset is directly applicable to investigating RQ2.

## 4.2 Experiment setup

### 4.2.1 Preprocessing

In the preparation of the dataset, Python was the primary tool used. The Pandas library was extensively utilized for data manipulation and processing, while the Numpy library was applied for mathematical computations. Pandas is a powerful data manipulation library (pandas development team, 2022) in Python that provides a flexible and efficient data structure (dataframe) for handling and analyzing data (McKinney, 2010). Numpy

(Numerical Python) is another Python library that supports a wide range of mathematical functions for scientific computing tasks (Harris et al., 2020).

The dataset was delivered as an Excel file from the hospital. Initially, the Excel file was cleaned manually to remove irrelevant information. This also involved deleting the first few rows containing group headers and descriptions irrelevant to the analyses.

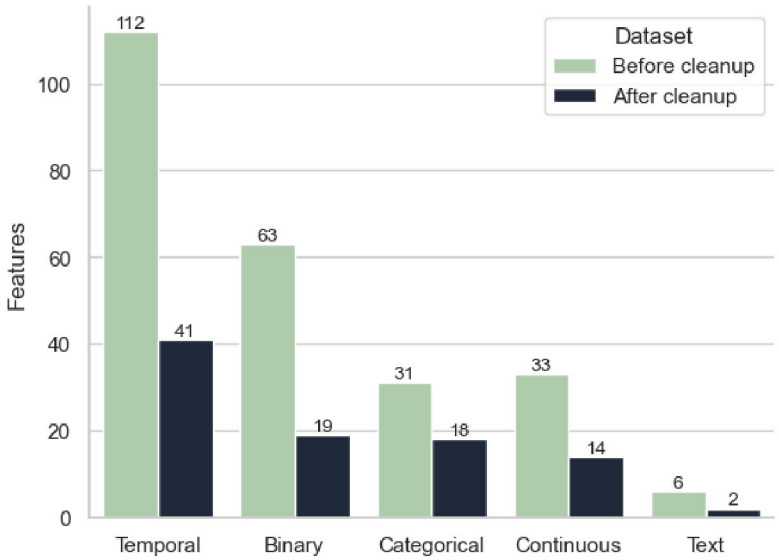



Figure 4.2: Number of features by type, before and after initial cleanup.

In addition, certain columns not contributing valuable information for the study were removed, as discussed previously in this chapter. The number of features by categories before and after this initial cleanup is illustrated in figure 4.2. After these manual cleanup steps, the Excel file was imported into a Pandas dataframe. The next step was to address data redundancy, potentially leading to biases in the study result. Any duplicated rows in the dataframe were identified and removed. The last step in this initial data cleanup was to remove five patients with no cancer diagnosis, as they did not fall within the target group for this study.

Improvements for readability were made to the DataFrame, such as changing column names to English and adding semantic value, for example, changing 'Prosedyrekode' (procedure code) to 'Orthopedic procedure code.' Categories in columns such as 'Smoking' were translated from their original language to English. Similar translations were made for several columns, including 'Family relationship description' and 'Other risk factor.'

To utilize the information in the 'Diagnosis date' feature, the day of the year (1-365) was calculated from the dates, as the date itself did not provide any helpful information. Similar calculations determined the number of days since a previous thrombosis event. Boolean flags were then created to indicate whether a patient had experienced previous thrombosis events.

New features were created, including 'Diagnosis day of year,' 'Days since previous thrombosis 2', and 'Days since previous thrombosis 1', and the now obsolete associated date columns were deleted. One 'SpO2% value' saved as a string ('96-100') was replaced with the mean value 98.0. 'Other risk factors' containing text values were converted to binary values, where '1' indicated a risk factor was provided. An iteration was performed

| **Degree of bleeding** | **Substituted value** |
|---|---|
| No bleeding | 0 |
| Clinically relevant minor bleed | 1 |
| Major bleeding | 2 |
| Minor bleeding | 3 |

Table 4.4: Encoding 'Degree of bleeding' values to numerical values performed during preprocessing.

through a list of all treatments and their associated start and end dates, creating a new boolean feature for each one to indicate if a patient had received that specific treatment.

Encoding was performed on the 'Gender' column, with '0' for males and '1' for females. The 'ICD-10 cancer code' column was transformed only to contain the leading numeric digits in the ICD-10 code. The columns datatype was then converted to a numeric type. The values in the 'BMI' column were separated initially using a comma; this was replaced with a period to convert the column to a float type. The 'Degree of bleeding' column was encoded to numeric values according to table 4.4. Missing 'Degree of bleeding' values signified no bleeding, so the missing values were filled with values of '0'.

Following the initial data cleaning steps, the dataset was divided into training and testing sets to safeguard against data leakage during the subsequent imputation process. The division was performed using the 'train_test_split' function from sklearn, employing stratified sampling with a test size of 0.2 and a random state of 42. Since each patient could have varying degrees of bleeding (the target value) recorded on different rows, the split was executed based on a list of unique patient IDs, each associated with their most significant degree of bleeding. Following this, all rows corresponding to a specific patient ID from the original dataset were collected and placed into the training or testing sets according to the split list. Ultimately only one row for each patient will remain in the dataset, so the current imbalance in the train/test split ratio is unimportant. Once the extra rows for each patient are removed, the balance between the training and testing sets will be correctly adjusted.

The next step in the data preprocessing involved imputing missing values on the training and testing sets. The columns 'Orthopedic procedure code,' 'Other surgery procedure code,' and 'Trauma description' each had a corresponding boolean column indicating whether the patient had experienced the procedure or event. It was found that the values in the associated columns were not always coherent upon analyzing the data. Therefore, the three columns mentioned were encoded, with '1' if the corresponding values were true and '0' if not. Missing values in 'Thrombophilia diagnosis' were filled similarly by iterating each row, checking the columns 'Known thrombophilia' and 'Other type,' and setting the value to '1' if either is true.

Columns where missing values equaled 'no' were filled with '0'. 'Familial relationship' was filled with 'yes' if the associated 'Familian disposition' column were true; elsewhere, 'unknown.' Missing smoking status was filled with 'unknown,' 'Other diagnosis cause' with 'no,' and 'Days hospitalized' were filled with a temporary value of '-1' to facilitate segmentation later on. Then all columns used to assist the imputation process were dropped. For the remaining boolean and numeric columns, missing values were filled

with the mode and mean of the respective column for the training set. These values were stored for later usage in filling the test set to prevent data leakage. Missing 'BMI' values were calculated using the formula for BMI:

$$BMI = \frac{weight(kg)}{(height(cm)/100)^2} \tag{4.1}$$

The last step in the imputation process involved consolidating binary values of the columns 'Orthopedic procedure code,' 'Other surgery procedure code,' 'Thrombophilia diagnoses, ' and 'Bleeding type' by grouping the data by the patient ID and diagnosis date and applying the max value to each column to ensure consistency for each unique date across patients.

Then the days hospitalized, cancer diagnosis, smoking status, familial relationship, other diagnosis cause, and thrombosis diagnosis features were one hot encoded. 'Days hospitalized' were segmented into the following bins before one hot encoded:

**Short stay:** 0-3 days, **Medium stay:** 4-7 days, **Long stay:** 8-14 days, **Extended stay:** 15-30 days, **Very extended stay:** >30 days.

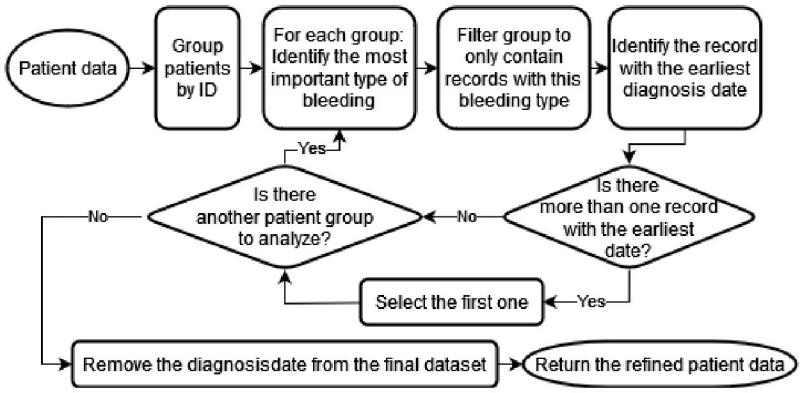



Figure 4.3: This flowchart visually represents the multi-step process used to select a single row of data per patient.

The 'ICD-10 cancer codes' and 'Thrombosis diagnosis' were segmented into categories according to a document provided by Østfold Hospital, rendered in table 4.6 and 4.5 respectively. Please note that a medical professional has not verified the translations of the diagnoses from Norwegian to English. Therefore, for the sake of transparency, these translations are presented in table A.1 in the appendix of this document.

During the one-hot encoding process, some values in the training set were not found in the testing set. Consequently, the training set ended up with fewer columns than initially. A function was therefore applied to ensure that the test set not only had the same columns as the training set, but they also appeared in the same sequence.

In the final step of the data preprocessing, a procedure was followed, according to a set of rules provided by the hospital, to ensure that one row per patient was accurately selected for the study. The procedure is illustrated with the flowchart in figure 4.3.

Beginning with a collection of patient data, the procedure first identified the most severe bleeding type registered for each patient. The data was then filtered, isolating only the rows corresponding to this highest-severity bleeding type. Then the earliest diagnosis

date among these rows was determined. If multiple entries corresponded to this earliest date, the first of these entries was prioritized. This systematic procedure ensured that a single most significant row was selected for each patient. The distribution of bleeding types before and after the data preprocessing are shown in figure 4.4.

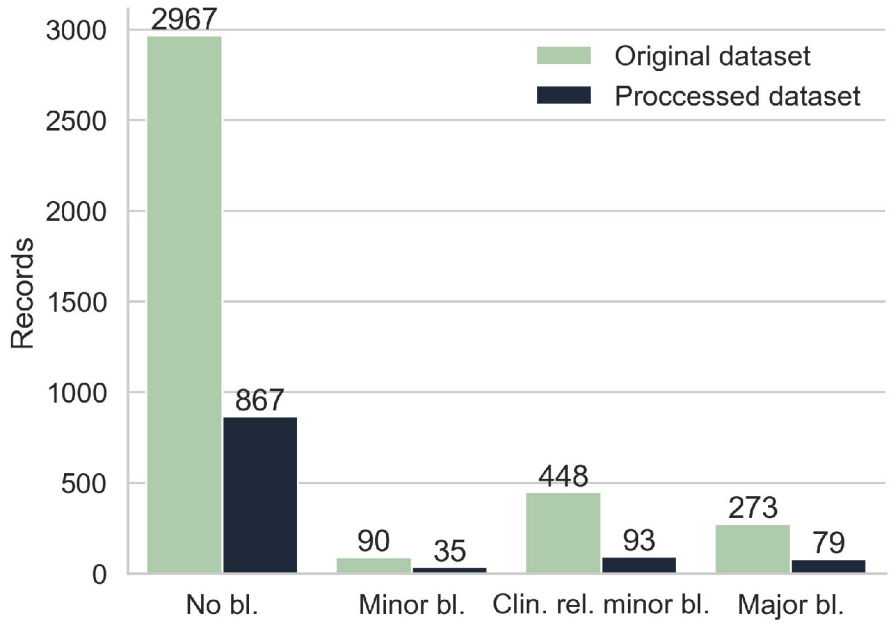



Figure 4.4: The distribution of target classes in the original and processed dataset.

| Group | Diagnosis |
|---|---|
| 1 | DVT, Muscle vein thrombosis |
| 2 | Portal vein thrombosis, Hepatic vein thrombosis, Mesenteric vein thrombosis, Splenic vein thrombosis, Inferior vena cava thrombosis |
| 3 | Pulmonary embolism |
| 4 | Ovarian vein thrombosis, Renal vein thrombosis, Other, Superficial thrombophlebitis, Upper arm thrombosis, Jugular vein thrombosis |

Table 4.5: Groups used for segmenting thrombosis diagnoses before one-hot encoding.

### 4.2.2 Normalizing data

Numerical features in the dataset were normalized using the StandardScaler function from the Scikit-learn library. The process was integrated into a pipeline for execution with the grid search cross-validation. This scaled the features to get a normal distribution with a mean of zero and a standard deviation of one. The normalization can improve the performance of multiple ML algorithms.

A ColumnTransformer was set up with StandardScaler applied to these columns. The remainder of the features ('remainder=passthrough') were left unscaled, which means they

were unaffected by this preprocessing step and allowed to "pass through" the transformer. By integrating the preprocessing steps into a pipeline, consistency in applying these steps across different models and datasets can be ensured, helping in the reproducibility and efficiency of the analysis. This pipeline configuration also ensured there was no data leakage between the training and testing data, by only fitting the scalers to the training data.

| **ICD-10 code interval** | **Label (description)** |
|---|---|
| [0, 15) + [30, 33) | ENT (Ear-nose-throat) |
| [15, 18) | U-GIT (Upper gastrointestinal) |
| [18, 22) | L-GIT (Lower gastrointestinal) |
| [22, 26) | LPB (Liver-pancreas and bile) |
| [26, 30) | G and OL -GIT (Gastrointestinal tract and overlapping lesion of digestive system) |
| [33, 40) | RS and MT (Respiratory or mediastinal) |
| [40, 50) | SCT (Skin, bone, and other connective tissue) |
| [50, 51) | Breast |
| [51, 60) | FGT (Female genital) |
| [60, 64) | MGT (Male genital) |
| [64, 69) | UT (Urinary) |
| [69, 73) | CNS (Eye, brain, and other CNS tissue) |
| [73, 76) | ET (Endocrine) |
| [76, 81) | Sc and NS (Secondary or unspecified) |
| [81, 97) | Hem (Lymphoid, hematopoietic or related tissue) |
| [97, 98) | MultyPrime (Multiple primary sites) |

Table 4.6: Intervals used for mapping ICD-10 cancer code to labels before one-hot encoding.

### 4.2.3 Selecting models for evaluation

In the process of selecting ML algorithms for this study, the starting point was Scikit-learn's (Pedregosa et al., 2011) inherently multiclass models due to the multiclass nature of the dataset's target variable. However, not all these models were suitable for the task. The first exclusion was the 'LabelPropagation' and 'LabelSpreading' algorithms, typically used for semi-supervised learning tasks. As this study is dealing with a fully labeled dataset, the semi-supervised algorithms were redundant, and they were therefore eliminated.

'LinearDiscriminantAnalysis' was the next to be excluded. Despite being a potent tool under the right circumstances, it was deemed unsuitable for this study.

'LinearDiscriminantAnalysis' makes specific assumptions about the dataset, specifically that the features are normally distributed and that the classes have identical covariance matrices. When the dataset was examined using the Shapiro-Wilk test for normality and Levene's test for equality of variances, it was clear that the dataset did not meet these assumptions, and the model was deemed unfit for this particular task. Lastly, 'RidgeClassifierCV' and 'LogisticRegressionCV' were excluded as these classes already contain built-in cross-validation. Including them in the grid search would result in redundant cross-validation computations; hence they were replaced with their non-CV counterparts, 'RidgeClassifier' and 'LogisticRegression.' Multiclass models that were not inherently multiclass were not considered. This was primarily to keep the complexity and computational time within manageable boundaries.

After this elimination process, the list of models to be tested was narrowed down to the following 12 models:

1. DecisionTreeClassifier (DTC)
2. ExtraTreeClassifier (ETC)
3. ExtraTreesClassifier (ETSC)
4. GaussianNB (GNB)
5. KNeighborsClassifier (KNC)
6. LinearSVC (LSVC)
7. MLPClassifier (MLPC)
8. NearestCentroid (NC)
9. QuadraticDiscriminantAnalysis (QDA)
10. RadiusNeighborsClassifier (RNC)
11. RandomForestClassifier (RF)
12. RidgeClassifier (RC)

These models cover a wide range of ML techniques and offer different approaches to the problem of predicting bleeding in thrombosis patients. The following section will dive deeper into the hyperparameter tuning of the algorithms.

### 4.2.4 Hyperparameter tuning and top-performing model selection

The selection of hyperparameters for the initial grid search was primarily chosen after an extensive literature review and exploring successful use cases for similar problems found through online resources.

The aim was to balance a comprehensive exploration of the hyperparameter space and computational feasibility. Therefore, the range of each hyperparameter was defined to be broad enough to capture a variety of potential model behaviors but limited enough to keep the computational time within manageable bounds.

'random_state' and 'class_weight' were set for all models that supported it. Setting random_state to a consistent number like 42 ensures that the output of the models is reproducible. Furthermore, setting 'class_weight = "balanced"' is essential when dealing with imbalanced classes. This setting automatically adjusts weights proportional to class frequencies in the input data, which can help improve the model's performance on the under-represented class.

In table 4.7, the specific hyperparameters used for all the models in the initial grid search are presented. The selection of these hyperparameters represents an educated starting point rather than an optimal solution. Different sets of hyperparameters could potentially lead to better performance.

| **Parameter** | **DTC** | **ETC** | **ETSC** | **RFC** |
|---|---|---|---|---|
| max_features | 'sqrt', None | 'sqrt', None | | 'sqrt', None |
| n_estimators | | | 100, 200 | 2, 10, 20, 100 |
| class_weight | 'balanced' | 'balanced' | 'balanced' | 'balanced' |
| random_state | 42 | 42 | 42 | 42 |
| criterion | 'gini', 'entropy' | 'gini', 'entropy' | 'gini', 'entropy' | |
| max_depth | None, 10 | None, 10 | None, 10 | 3, 5, 10, 15, None |
| **Parameter** | **RC** | **MLPC** | **LSVC** | **GNB** |
| class_weight | 'balanced' | | 'balanced' | |
| alpha | 0.1, 1.0, 2 | | | |
| random_state | 42 | 42 | 42 | |
| hidden_layer_sizes | | (50, 50, 50), (50, 100, 50), (100,) | | |
| activation | | 'tanh', 'relu' | | |
| solver | | 'lbfgs', 'adam' | | |
| C | | | 0.1, 1 | |
| multi_class | | | 'crammer_singer' | |
| var_smoothing | | | | 1e-09, 1e-08 |
| **Parameter** | **QDA** | **NC** | **KNC** | **RNC** |
| reg_param | 0.0, 0.1 | | | |
| shrink_threshold | | 0.2, None | | |
| metric | | 'euclidean', 'manhattan' | | |
| n_neighbors | | | 3, 5 | |
| weights | | | 'uniform', 'distance' | 'uniform', 'distance' |
| radius | | | | 1, 500, 2000 |
| outlier_label | | | | None, 'most_frequent' |

Table 4.7: The hyperparameters used for each model in the initial grid search.

Each model was subjected to a grid search. The models were trained and tested on various combinations of hyperparameters using stratified 10-fold cross-validation to identify the parameters that resulted in the best performance. This initial testing of model performance provides a fundamental basis for assessing the ability of the dataset to differentiate classes using ML and selecting the top models for further optimization. Based on these results, the top three models will then be selected based on the F1 macro score.

### 4.2.5 Dataset compositions

Identifying the top-performing models provides initial insights into how well machine learning methods can differentiate classes in the dataset, thereby addressing RQ1. However, the selection process contains more than choosing the best-performing model with default settings on a given dataset.

In the initial grid search, potential overfitting was considered across all models to uncover any model that might perform artificially well on the dataset and not generalize well to unseen data, an essential part of RQ1.

Given the concerns about overfitting, an investigation into the composition of the dataset is a part of the study. The next phase is 'model optimization,' contributing to the RQs by evaluating the top ML models on a grid search with extended parameters and testing on various dataset variations. This confirms the models' robust and versatile performance regarding class differentiation.

These dataset variations include a reduced feature set, an over-sampled dataset using the Synthetic Minority Over-sampling Technique (SMOTE), a stratified split with random shuffling, and a different split ratio of 90/10. The investigation of these various steps is expected to answer RQ1, assessing to what extent state-of-the-art ML methods can differentiate classes amidst changing data conditions while countering potential overfitting tendencies.

Oversampling with SMOTE aims to balance class distribution by creating synthetic examples in the minority class, providing more information for the model to learn from. This was done by applying Imbalanced-learn's SMOTE algorithm to the initial dataset, creating 1,917 new samples, and effectively adjusting the class balance to 694 of each record.

The reshuffled and the 90/10 test/train split were created to assess the model's stability to varying selection of samples and proportions of training and testing data. This ensures that any poor performance is not caused by 'bad luck' in the dataset split.

#### Feature reduction

The last dataset variation was a reduced feature dataset. The feature reduction process aimed to improve the model's performance and complexity. This involved several steps, such as calculating feature importance and permutation importance, then executing an iterative feature elimination process.

Initially, feature importance was calculated, ranking each feature based on the reduction of impurity using the random forest classifier. The permutation importance was calculated using sklearn's 'permutation_importance' function, providing a robust measure of a feature's importance by evaluating the decrease in a model's performance when the feature's value is randomly shuffled.

A dataframe was constructed to capture various metrics to gain insights into the significance of each feature: feature importance measures, variance, correlation with the target variable, and the count of missing values per feature. This structured approach facilitated a comprehensive understanding of each feature's potential predictive power and relevance to the study. Non-informative features with zero feature importance and low variance (<0.01) were marked for elimination.

Then an iterative feature elimination process was executed, where a cross-validated grid search was conducted using 'GridSearchCV' to tune the parameters of a random forest classifier. The grid search was wrapped in a custom function that took the hyperparameters to tune and train data as parameters. Using cross-validation, the model was optimized for the F1 macro score, also reporting ROC AUC, recall, precision, and accuracy for each combination of hyperparameters.

After each iteration of the grid search, features that had an importance less than a specified threshold (initially 0.001) were eliminated. The process was repeated for several iterations until no features had importance below the threshold. The change in performance across iterations was visualized using line plots to compare the training and test scores for the different metrics. A confusion matrix was also plotted to assess the model's performance visually.

Overall, this feature reduction process combined several techniques to create a more efficient, interpretable dataset to increase class differentiation while reducing complexity and potential overfitting.

### 4.2.6 Extended hyperparameters grid search

The top-performing models are then further evaluated using an extended grid search. This helped confirm that the best parameters were within the chosen range, improving the chance of optimizing the model's performance. The aim is to optimize the top models to increase their effectiveness on the given data and ensure their robustness and reliability when encountering new, unseen data.

In the initial model selection, the hyperparameters for each of the top three models were configured within a defined range. This allowed for a thorough comparison to assess their impact on the model's efficiency in predicting bleeding events as guided by RQ2. The choice of hyperparameters was driven by the need to balance computational feasibility and extensive parameter space exploration.

Following the initial model selection, the top three models are then subjected to an extended tuning process with GridSearchCV to facilitate a comprehensive exploration of hyperparameters for each top model. This helps to ensure optimal performance by finding the correct configuration of parameters that produces the most effective predictions while reducing overfitting.

After the extended grid search, each top-performing model will be evaluated on the entire dataset to verify the performance and confirm their effectiveness in predicting bleeding episodes - an essential aspect called for in RQ2. This allows for a more robust evaluation of the models' performances under various conditions. The selection of top models and hyperparameters is presented in the following chapter.

# Chapter 5

# Results

This chapter will present the findings of this study. The essence is understanding how these models perform in different situations, their strengths and weaknesses, and the extent to which the dataset allows for the differentiation of classes. To do this, stratified 10-fold cross-validation has been employed with a grid search, using the F1 macro score to evaluate the model's performance while also considering the accuracy, recall, precision, F1 score, and ROC AUC score. The first part of this chapter is dedicated to presenting these results for each of the 12 selected models for the initial model selection. Each model's performance will be presented using the metrics defined above.

The second part will examine the performance of these top models with different dataset variations to better understand their strengths and weaknesses. Each model's behavior under different dataset compositions will be systematically evaluated, providing key insights to answer RQ1 regarding the suitability of the dataset to predict bleeding and RQ2 addressing the effectiveness of the models.

The last part of this chapter will evaluate the top models' performance when trained and tested on the entire dataset. This will provide a comprehensive assessment of their ability to predict bleeding events in thrombosis patients under conditions that challenge the robustness of their predictive capabilities. This part will therefore provide essential insights into answering the research questions, particularly RQ2, while also addressing the performance issues regarding RQ1.

By the end of this chapter, you should have a thorough understanding of these 12 machine learning models and their performance on the problem in this thesis.

## 5.1 Overview of initial results

The initial model testing was performed on 12 different models: DecisionTreeclassifier (DTC), ExtraTreeclassifier (ETC), ExtraTreesclassifier (ETSC), GaussianNB (GNB), KNeighborsclassifier (KNC), LinearSVC (LSVC), MLPclassifier (MLPC), NearestCentroid (NC), QuadraticDiscriminantAnalysis (QDA), RadiusNeighborsclassifier (RNC), RandomForestclassifier (RFC), and Ridgeclassifier (RC).

The evaluation metrics in table 5.1 reveal a substantial difference in the model's performance. The scores are a calculated mean of the cross-validations of ten splits performed during the grid search. The train score is the result of the prediction made on the training part of the data in each split, and the validation score is the outcome of predicting the validation, or holdout, part in each split. As discussed in chapter 4, this

study is particularly interested in the F1-macro score, which considers both precision and recall and treats all classes equally important.

| Model | Accuracy | Precision | Recall | F1-score | F1-macro | ROC AUC |
|---|---|---|---|---|---|---|
| DTC | 0.687 (0.038)<br>*1.000 (0.000)* | 0.689 (0.031)<br>*1.000 (0.000)* | 0.687 (0.038)<br>*1.000 (0.000)* | 0.686 (0.031)<br>*1.000 (0.000)* | 0.287 (0.048)<br>*1.000 (0.000)* | 0.528 (0.039)<br>*1.000 (0.000)* |
| ETC | 0.657 (0.058)<br>*1.000 (0.000)* | 0.687 (0.030)<br>*1.000 (0.000)* | 0.657 (0.058)<br>*1.000 (0.000)* | 0.670 (0.044)<br>*1.000 (0.000)* | 0.269 (0.056)<br>*1.000 (0.000)* | 0.515 (0.042)<br>*1.000 (0.000)* |
| ETSC | 0.742 (0.033)<br>*0.971 (0.006)* | 0.674 (0.017)<br>*0.975 (0.004)* | 0.742 (0.033)<br>*0.971 (0.006)* | 0.705 (0.023)<br>*0.972 (0.006)* | 0.251 (0.028)<br>*0.945 (0.010)* | 0.614 (0.027)<br>*1.000 (0.000)* |
| GNB | 0.187 (0.030)<br>*0.231 (0.009)* | 0.740 (0.047)<br>*0.836 (0.002)* | 0.187 (0.030)<br>*0.231 (0.009)* | 0.233 (0.045)<br>*0.262 (0.012)* | 0.150 (0.031)<br>*0.217 (0.006)* | **0.617 (0.055)**<br>*0.831 (0.006)* |
| KNC | 0.746 (0.020)<br>*0.813 (0.003)* | 0.675 (0.037)<br>*0.767 (0.025)* | 0.746 (0.020)<br>*0.813 (0.003)* | 0.704 (0.021)<br>*0.741 (0.007)* | 0.254 (0.040)<br>*0.288 (0.029)* | 0.558 (0.042)<br>*1.000 (0.000)* |
| LSVC | 0.236 (0.057)<br>*0.314 (0.039)* | **0.749 (0.075)**<br>*0.819 (0.011)* | 0.236 (0.057)<br>*0.314 (0.039)* | 0.282 (0.081)<br>*0.339 (0.056)* | 0.197 (0.037)<br>*0.295 (0.025)* | nan |
| MLPC | 0.738 (0.024)<br>*0.944 (0.015)* | 0.697 (0.039)<br>*0.945 (0.016)* | 0.738 (0.024)<br>*0.944 (0.015)* | 0.716 (0.029)<br>*0.941 (0.018)* | **0.311 (0.078)**<br>*0.855 (0.044)* | 0.581 (0.073)<br>*0.976 (0.010)* |
| NC | 0.143 (0.133)<br>*0.158 (0.154)* | 0.432 (0.355)<br>*0.669 (0.058)* | 0.143 (0.133)<br>*0.158 (0.154)* | 0.125 (0.180)<br>*0.137 (0.195)* | 0.087 (0.043)<br>*0.098 (0.052)* | nan |
| QDA | 0.802 (0.011)<br>*0.830 (0.036)* | 0.670 (0.028)<br>*0.841 (0.038)* | 0.802 (0.011)<br>*0.830 (0.036)* | **0.725 (0.012)**<br>*0.764 (0.058)* | 0.241 (0.028)<br>*0.346 (0.173)* | 0.506 (0.011)<br>*0.889 (0.025)* |
| RNC | **0.808 (0.006)**<br>*1.000 (0.000)* | 0.653 (0.009)<br>*1.000 (0.000)* | **0.808 (0.006)**<br>*1.000 (0.000)* | 0.722 (0.008)<br>*1.000 (0.000)* | 0.223 (0.001)<br>*1.000 (0.000)* | 0.539 (0.073)<br>*1.000 (0.000)* |
| RFC | 0.505 (0.058)<br>*0.846 (0.013)* | 0.676 (0.031)<br>*0.896 (0.006)* | 0.505 (0.058)<br>*0.846 (0.013)* | 0.567 (0.046)<br>*0.860 (0.011)* | 0.267 (0.051)<br>*0.729 (0.016)* | 0.560 (0.062)<br>*0.930 (0.007)* |
| RC | 0.424 (0.077)<br>*0.553 (0.017)* | 0.731 (0.044)<br>*0.835 (0.004)* | 0.424 (0.077)<br>*0.553 (0.017)* | 0.508 (0.071)<br>*0.607 (0.016)* | 0.264 (0.048)<br>*0.452 (0.013)* | nan |

Table 5.1: Results from the initial grid search over all considered models. The mean and standard deviation validation scores are reported with the mean and standard deviation train scores in italics.

The results show a significant variation in model performance, which suggests that no single model offers the best results across all performance metrics. However, RadiusNeighborsClassifier, LinearSVC, QuadraticDiscriminantAnalysis, MLPClassifier, and GaussianNB demonstrate top performances in at least one metric. On the other hand, LinearSVC, NearestCentroid, and RidgeClassifier deliver less promising results with the ROC AUC metric. The ROC AUC metric is calculated using a One-vs-One approach, which averages the ROC AUC for all possible pairwise combinations of classes. In multiclass classifications, if one or more classes have few or no instances in a split or the model predicts only a single class, the ROC AUC cannot calculate a score and returns NaN. Given the adjacent accuracy and recall scores for these models, it's likely that their poor performance is related to this.

Figure 5.1 clearly illustrates the considerable divergence between the training and validation F1 macro scores across several models. This indicates the prevalent overfitting tendencies displayed by these models, which demonstrate high performance on the training set but struggle to generalize effectively to unseen validation data.

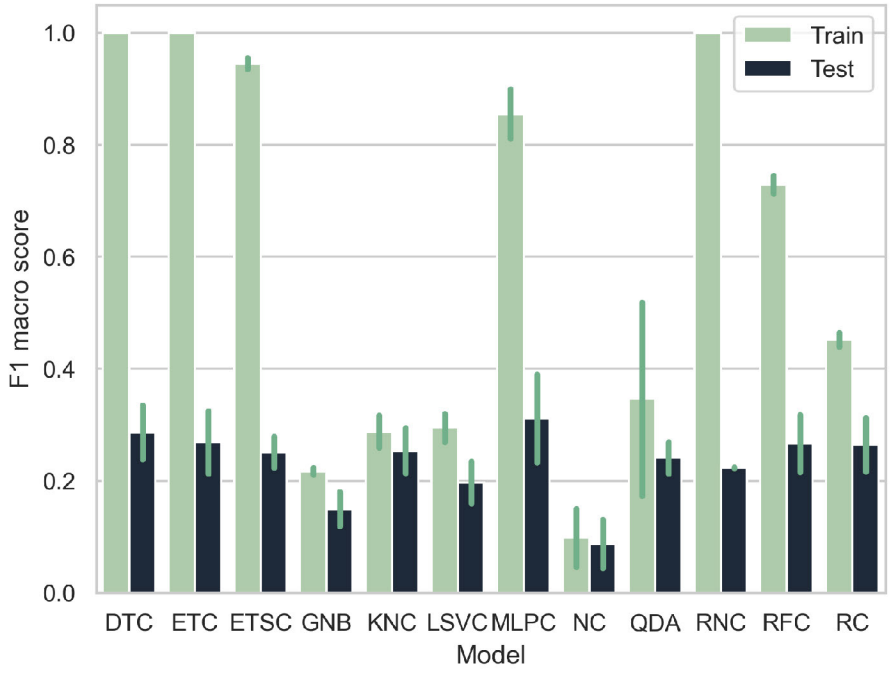



Figure 5.1: Comparative barplot of F1 macro scores (train and validation) for all models in the initial grid search. Error bars show the standard deviation.

### DecisionTreeClassifier

The DecisionTreeClassifier showed a robust precision and recall score of 0.68 though a relatively mediocre ROC AUC score of 0.528. The model has a considerable difference between its testing and training scores. It has a relatively fast fit time of 13.36 ms. The parameters that gave the best results on the validation data were: **Criterion:** entropy, **max depth:** None, **max features:** sqrt.

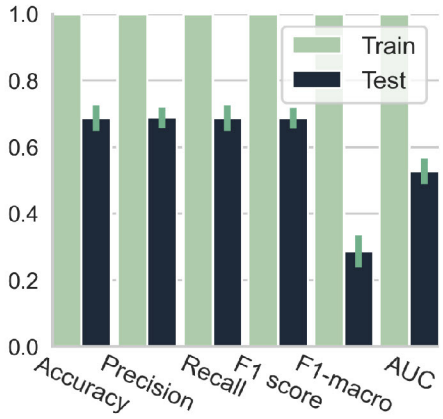
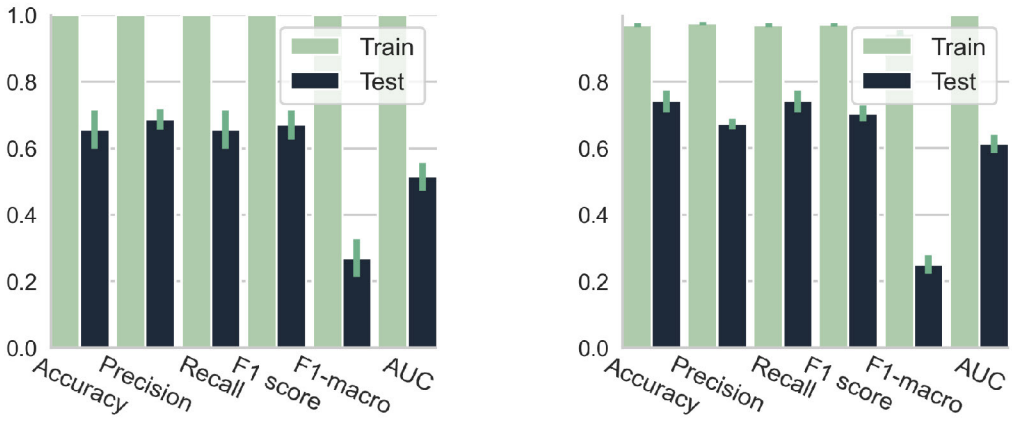



Figure 5.2: Performance of the DecisionTreeClassifier in the initial grid search. Error bars show the standard deviation.

### ExtraTreeClassifier

The ExtraTreeClassifier, similarly to the DecisionTreeClassifier, performed relatively well in accuracy and recall but displayed a less impressive ROC AUC score of 0.515. Like the DecisionTreeClassifier, the ExtraTreeClassifier shows a considerable difference between testing and training scores. It also benefits from a relatively quick mean fit time of 16.75 ms.

The parameters that gave the best results on the validation data were: **Criterion:** gini, **max depth:** None, **max features:** None.




(a) ExtraTreeClassifier

(b) ExtraTreesClassifier

Figure 5.3: Performance of the ExtraTreeClassifier and ExtraTreesClassifier in the initial grid search. Error bars show the standard deviation.

### ExtraTreesClassifier

The ExtraTreesClassifier demonstrates a more restrained difference between training and testing scores, signifying a better generalization capability than the DecisionTreeClassifier and ExtraTreeClassifier. However, the training time of 215.82 ms exceeds the DecisionTreeClassifier and ExtraTreeClassifier.

Further refinement of the ExtraTreesClassifier could be modifications to the 'n_estimators' parameter, which controls the number of trees in the forest, and the 'max_depth parameter'. An increase in 'n_estimators' generally increases the model's performance. However, this comes with the trade-off of enhanced computational expense. Therefore, finding an optimal balance between these factors is crucial. The parameters that gave the best results on the validation data were: **Criterion:** gini, **max depth:** 10, **n estimators:** 100.

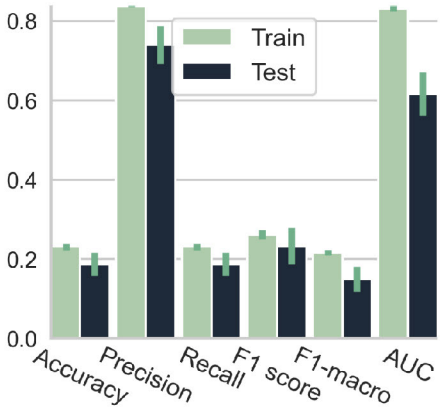



Figure 5.4: Performance of the GaussianNB in the initial grid search. Error bars show the standard deviation.

## 5.1.1 GaussianNB

Even though the Gaussian Naive Bayes classifier has the highest ROC AUC score amongst all the models, at 0.617, it showed a noticeably low F1-score macro score of 0.150. This could signify that the model has difficulties establishing an effective balance between precision and recall during class prediction. The relatively low contrast between the training and testing scores indicates that this model can better generalize to previously unseen data.

Regarding training time, the Gaussian Naive Bayes model is relatively efficient (15.65 ms), showing a smaller gap between training and testing scores than the previous models. The parameters that gave the best results on the validation data were: **Var smoothing:** 1e-08.

### KNeighborsClassifier

The KNeighborsClassifier showed a noteworthy overall performance. Still, there is potential for enhancement in its ROC AUC score of 0.558. The KNeighborsClassifier performs proficiently across all metrics, hinting at a well-balanced classifier. The parameters that gave the best results on the validation data were: **N neighbors:** 3, **weights:** distance.

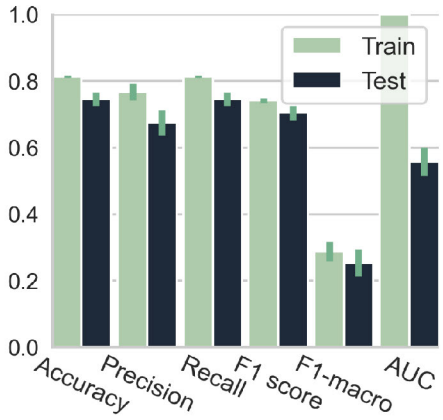



Figure 5.5: Performance of the KNeighborsClassifier in the initial grid search. Error bars show the standard deviation.

### LinearSVC

The LinearSVC model achieved the highest precision score of 0.749, indicating a low false positive rate. Despite this, the model failed to compute a ROC AUC score, potentially due to the previously discussed issues.

In contrast to the other models, the LinearSVC model requires a longer training time of 4.9 seconds. Despite the substantial computational costs, its performance on the ROC AUC metric remains poor.

The parameters that gave the best results on the validation data were: **C:** 0.1, **multi class:** crammer_singer.

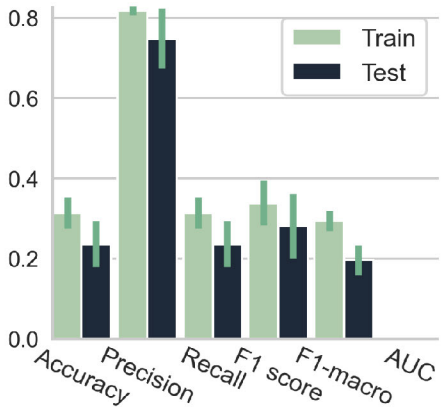


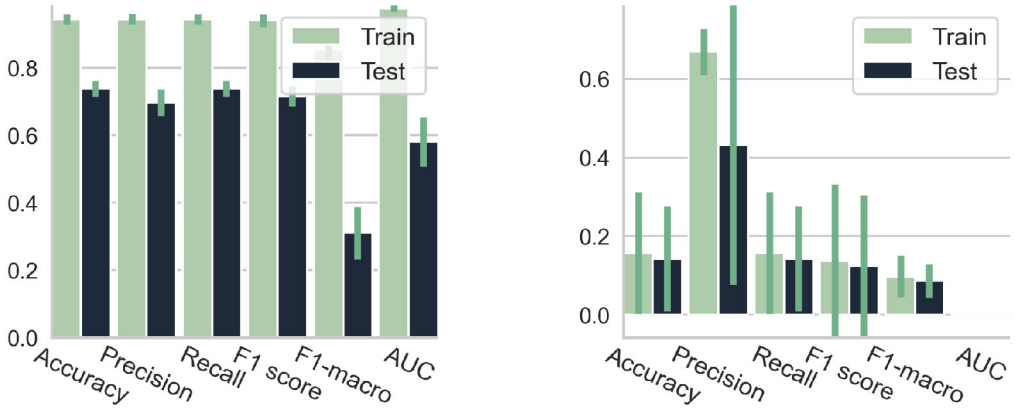

Figure 5.6: Performance of the LinearSVC in the initial grid search. Error bars show the standard deviation.

### MLPClassifier

The MLPClassifier model displayed the highest performance, with an F1 macro score of 0.311. This score implies that this model is best at getting an equal balance between precision and recall for each class without a bias toward the majority class.

Yet, the MLPClassifier model shows a considerable gap between training and testing scores, indicating overfitting. Its training time is relatively long compared to the other models, at 1.36 seconds.

The parameters that gave the best results on the validation data were: **Activation:** tanh, **hidden layer sizes:** (50, 50, 50), **solver:** adam.




(a) MLPClassifier

(b) NearestCentroid

Figure 5.7: Performance of the MLPClassifier and NearestCentroid in the initial grid search. Error bars show the standard deviation.

### NearestCentroid

The NearestCentroid model faced difficulties across most performance metrics, showing the lowest F1 macro score out of all the models. Additionally, the model failed to return a score for the ROC AUC metric. The parameters that gave the best results on the validation data were: **Metric:** euclidean, **shrink threshold:** 0.2.

### QuadraticDiscriminantAnalysis

The QuadraticDiscriminantAnalysis model exhibited noteworthy results, securing the highest F1-score (0.725) among the evaluated models. However, its ROC-AUC score (0.506) was the lowest, indicating potential challenges with effective class discrimination.

The QDA model's performance was less than satisfactory across all metrics compared to most other models. The parameters that gave the best results on the validation data were: **Reg param:** 0.0.

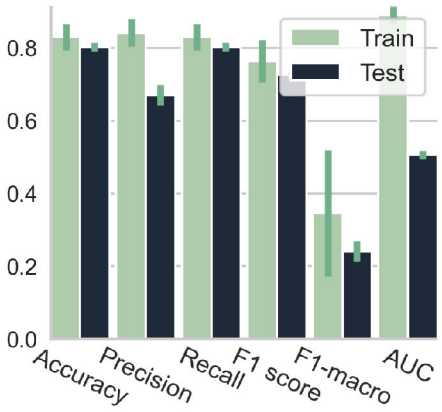



Figure 5.8: Performance of the QuadraticDiscriminantAnalysis in the initial grid search. Error bars show the standard deviation.

### RadiusNeighborsClassifier

The RadiusNeighborsClassifier showed a notable accuracy score of 0.808, which is the best of all the models. It displayed less promising ROC AUC and F1-macro scores, which were 0.539 and 0.223, respectively, highlighting potential difficulties in distinguishing between different classes. There is a significant difference between the training and testing

scores, indicating a tendency of overfitting. Given its nature, the RadiusNeighborsClassifier may not be the most suitable classifier for high-dimensional datasets. Parameters with best results on the validation data were: **Weights:** distance, **outlier label:** most_frequent, **radius:** 500.

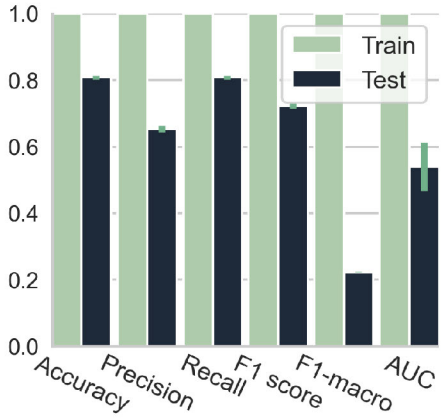


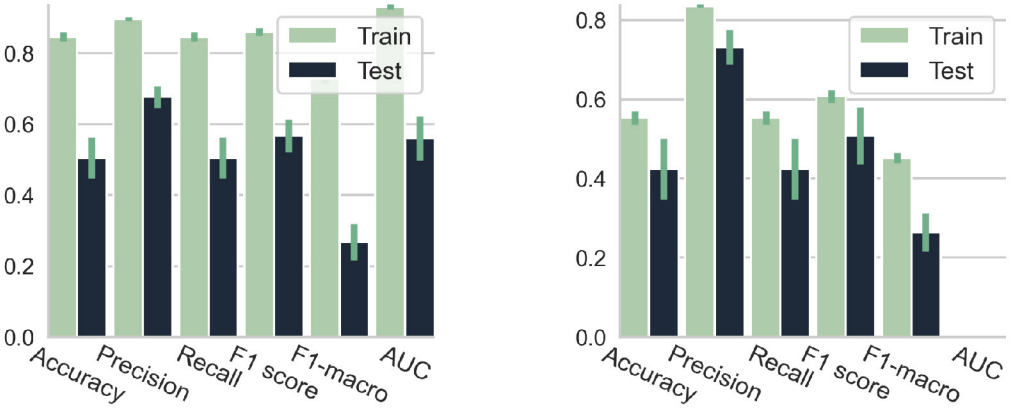

Figure 5.9: Performance of the RadiusNeighborsClassifier in the initial grid search. Error bars show the standard deviation.

### RandomForestClassifier

The RandomForestClassifier displayed moderate performance, though it recorded a slightly lower ROC-AUC score of 0.560. This suggests it has room for improvement in distinguishing between different classes. The RandomForestClassifier model provided acceptable performance across most of the evaluated metrics. While its training duration is comparatively longer (17.76 ms) than most of the models, it may be considered justifiable when considered against the model's performance. Parameters with best results on the validation data were: **Max depth:** None, **max features:** sqrt, **n estimators:** 2.

### RidgeClassifier

The RidgeClassifiercould not compute the ROC AUC score. Its performance across the remaining metrics was modest, with an accuracy score of 0.424. It is important to remember that RidgeClassifier is a linear model that assumes the relationship between the input and output variables is linear. If this assumption is incorrect, it could contribute to the results we see here.**Alpha:** 0.1 achieved the best results on the validation data.




(a) RandomForestClassifier

(b) RidgeClassifier

Figure 5.10: Performance of the RandomForestClassifier and RidgeClassifier in the initial grid search. Error bars show the standard deviation.

## 5.2 Top model selection

Following the initial grid search, the top three models were selected based on their F1 macro score. These top-performing models are:

1. **MLPClassifier:** The MLPClassifier is the top-performing model according to the F1 macro score. This suggests a better ability to classify instances from all classes accurately. However, a noticeable difference between training and testing scores indicates a potential overfitting issue.
2. **DecisionTreeClassifier:** The DecisionTreeclassifier ranks second in regards to the F1 macro score. There is an even bigger gap between training and testing scores compared to the MLPClassifier. Though, the DecisionTreeClassifier offers the advantage of easy interpretation and visualization, which can be important in understanding the underlying decision-making process.
3. **ExtraTreeClassifier:** The ExtraTreeClassifier also presents a relatively good F1 macro score. Similar to the DecisionTreeClassifier, it has a noticeable difference between training and testing scores. However, ExtraTreeClassifier could be less prone to overfitting than DecisionTreeClassifier due to its randomness in splitting nodes.

Based on the selection process, these three models have been identified as the top-performing models. That being said, achieving high model performance is an iterative process. Their current results are encouraging, but they don't suggest the end of the model refinement. Even though these models outperformed the others in the current setup, there is room for improvement and fine-tuning to increase class differentiation capabilities and model effectiveness.

| **Parameter** | **Values** |
|---|---|
| Hidden layer sizes | (10,), (50,), (10, 10), (50, 50), (50,50,50), (50,100,50), (100,) |
| Activation | tanh, relu |
| Solver | sgd, adam |
| Alpha | 0.0001, 0.05 |
| Learning rate | constant, adaptive |
| Max iter | 500 |
| Random state | 42 |

Table 5.2: Hyperparameters for the MLPClassifier used in the extended grid-search.

For the MLPClassifier, the hidden layer sizes, activation function, and solver were specified. For DecisionTreeClassifier and ExtraTreeClassifier, the maximum depth, criterion for the quality of a split, and the number of features to consider when looking for the best split were specified.

After the initial model selection, an extended hyperparameter tuning was performed. For the MLPClassifier, the learning rate and an alpha parameter for L2 regularization were introduced. This additional step is aimed at adjusting the learning process and controlling overfitting.

The hidden layers sizes search space was increased, and the maximum number of iterations was defined with a higher value than default. Since the lbfgs solver was associated with the poorest outcomes in the initial grid search, it was replaced with the stochastic gradient descent (sgd) solver to explore the possibilities.

Table 5.2 presents the hyperparameter values used in further experiments with the MLPClassifier.

For both DecisionTreeClassifier and ExtraTreeClassifier, new parameters were introduced during this phase. The splitter to choose the strategy used to split at each node, the minimum number of samples required to split an internal node, and the minimum number of samples needed to be at a leaf node were included.

These parameters add more controls to manage the size and complexity of the decision trees. The option of unlimited max features did not provide any promising results and was replaced with log2. The hyperparameters used with the DecisionTree and ExtraTreeClassifier are presented in table 5.3

| **Parameter** | **Options** |
|---|---|
| Criterion | gini, entropy |
| Splitter | best, random |
| Max depth | None, 3, 5, 10 |
| Min samples split | 2, 5, 10 |
| Min samples leaf | 1, 2, 5 |
| Max features | sqrt, log2 |
| Random state | 42 |
| Class weight | balanced |

Table 5.3: Hyperparameters for the DecisionTree and ExtraTreeClassifier used in the extended grid-search.

## 5.3 Performance with different dataset variations

To evaluate the robustness and adaptability of the top models, they are assessed on different compositions of the dataset. The datasets used are the initial dataset, reduced features, over-sampled using Synthetic Minority Over-sampling Technique (SMOTE), reshuffled with an 80/20 train/test split, a 90/10 split, and the entire dataset.

By examining how these top models perform across the different data subsets, we better understand their strengths and weaknesses. Different dataset compositions can impact the top models' ability to differentiate between classes. By evaluating the models across these varied dataset compositions, we can measure the models' robustness and ability to generalize to new data, providing insights to answer RQ1.

### 5.3.1 Initial dataset

The initial dataset is the same as used in the model selection process, with an 80/20 train/test split and a fixed random state. Table 5.4 presents the results from the best-performing estimator for each of the top models from the grid search cross-validation based on their F1 macro score. These estimators were then refitted to the entire training dataset. The predictions made by these refitted models on the test data are presented in table 5.5.

| Model | Accuracy | Precision | Recall | F1 score | F1-macro | ROC AUC |
|---|---|---|---|---|---|---|
| MLPC | 0.712 (0.048)<br>*0.968 (0.028)* | 0.701 (0.035)<br>*0.970 (0.027)* | 0.712 (0.048)<br>*0.968 (0.028)* | 0.704 (0.032)<br>*0.968 (0.030)* | 0.324 (0.057)<br>*0.919 (0.070)* | 0.567 (0.074)<br>*0.988 (0.014)* |
| DTC | 0.687 (0.038)<br>*1.000 (0.000)* | 0.689 (0.031)<br>*1.000 (0.000)* | 0.687 (0.038)<br>*1.000 (0.000)* | 0.686 (0.031)<br>*1.000 (0.000)* | 0.287 (0.048)<br>*1.000 (0.000)* | 0.528 (0.039)<br>*1.000 (0.000)* |
| ETC | 0.687 (0.038)<br>*1.000 (0.000)* | 0.689 (0.031)<br>*1.000 (0.000)* | 0.687 (0.038)<br>*1.000 (0.000)* | 0.686 (0.031)<br>*1.000 (0.000)* | 0.287 (0.048)<br>*1.000 (0.000)* | 0.528 (0.039)<br>*1.000 (0.000)* |

Table 5.4: Results for the extended grid search CV, using the initial dataset. The mean and standard deviation validation scores are reported with the train scores in italics.

| Model | Accuracy | Precision | Recall | F1 score | F1-macro | ROC AUC |
|---|---|---|---|---|---|---|
| MLPC | 0.758 | 0.712 | 0.758 | 0.720 | 0.286 | 0.529 |
| DTC | 0.693 | 0.688 | 0.693 | 0.689 | 0.292 | 0.533 |
| ETC | 0.693 | 0.688 | 0.693 | 0.689 | 0.292 | 0.533 |

Table 5.5: Performance metrics of each of the best estimators after refitting with GridSearchCV, evaluated on the initial test set.

#### MLPClassifier

The MLPClassifier displays an improvement in all evaluated metrics except for the ROC AUC score, which shows a slight decrease. This could indicate overfitting, further supported by an increase in all metrics for the training set. The refitted MLPClassifier shows promising results on the training data but performs poorly on the test data, as illustrated in figure 5.11, with an F1 macro score of 0.286. This is a noticeable decrease from the cross-validation score. The parameters that gave the best results on the

validation data were: **Activation:** tanh, **alpha:** 0.0001, **hidden layer sizes:** (50, 50, 50), **learning rate:** adaptive, **max iter:** 500, **solver:** adam.


Figure 5.11: Confusion matrices for the MLPClassifier applied to the initial training and test dataset.

### DecisionTreeClassifier

The DecisionTreeClassifier displayed identical results in both the initial and extended parameter grid searches. When refitting the model on the entire training set, there is an indication that the model is overfitting to the training data with similarly poor results as the MLPClassifier on the test data, illustrated in figure 5.12. Although, there is a slight increase in the number of correctly predicted patients with bleeding. Unlike the MLPClassifier, the DecisionTreeClassifier displays a higher F1 macro score of 0.292 with the test data. The parameters that gave the best results on the validation data were: **Criterion:** entropy, **max depth:** None, **max features:** sqrt, **min samples leaf:** 1, **min samples split:** 2, **splitter:** best.


Figure 5.12: Confusion matrices for the DecisionTreeClassifier applied to the initial training and test dataset.

### ExtraTreeClassifier

Interestingly, the ExtraTreeClassifier performs similarly to the DecisionTreeClassifier, a minor improvement from the initial grid search. When the ExtraTreeClassifier is refitted on the training data, we get an identical matrix to the DecisionTreeClassifier, illustrated in figure 5.13. The DecisionTreeClassifier also displays an increased F1 macro score of 0.292. The parameters that gave the best results on the validation data were: **Criterion:**

entropy, **max depth:** None, **max features:** sqrt, **min samples leaf:** 1, **min samples split:** 2, **splitter:** best.


Figure 5.13: Confusion matrices for the ExtraTreeClassifier applied to the initial training and test dataset.

### 5.3.2 Reduced dataset

The reduced dataset consists of 47 columns in the initial dataset, as described in chapter 4. Table 5.6 presents the results from the grid search cross-validation. The predictions made by the refitted models on the test data are shown in table 5.7.

| Model | Accuracy | Precision | Recall | F1 score | F1-macro | ROC AUC |
|---|---|---|---|---|---|---|
| MLPC | 0.753 (0.021)<br>*0.995 (0.002)* | 0.694 (0.021)<br>*0.995 (0.002)* | 0.753 (0.021)<br>*0.995 (0.002)* | 0.717 (0.019)<br>*0.995 (0.002)* | 0.286 (0.044)<br>*0.988 (0.005)* | 0.560 (0.046)<br>*1.000 (0.000)* |
| DTC | 0.512 (0.058)<br>*0.714 (0.051)* | 0.690 (0.018)<br>*0.884 (0.009)* | 0.512 (0.058)<br>*0.714 (0.051)* | 0.572 (0.049)<br>*0.754 (0.043)* | 0.298 (0.033)<br>*0.614 (0.045)* | 0.562 (0.039)<br>*0.972 (0.011)* |
| ETC | 0.512 (0.058)<br>*0.714 (0.051)* | 0.690 (0.018)<br>*0.884 (0.009)* | 0.512 (0.058)<br>*0.714 (0.051)* | 0.572 (0.049)<br>*0.754 (0.043)* | 0.298 (0.033)<br>*0.614 (0.045)* | 0.562 (0.039)<br>*0.972 (0.011)* |

Table 5.6: Results from the extended grid search CV, using the reduced dataset. The mean and standard deviation validation scores are reported with the training scores in italics.

| Model | Accuracy | Precision | Recall | F1 score | F1-macro | ROC AUC |
|---|---|---|---|---|---|---|
| MLPC | 0.763 | 0.657 | 0.763 | 0.704 | 0.235 | 0.484 |
| DTC | 0.577 | 0.654 | 0.577 | 0.611 | 0.242 | 0.471 |
| ETC | 0.577 | 0.654 | 0.577 | 0.611 | 0.242 | 0.471 |

Table 5.7: Performance metrics of each of the best estimators after refitting with GridSearchCV, evaluated on the reduced test set.

#### MLPClassifier

Compared to the results from the initial dataset, the MLPClassifier has a noticeable reduction in the F1 macro score of 0.286 from the cross-validation and even lower when evaluated on the test data with an F1 macro score of 0.235. As illustrated in figure

5.14, the MLPClassifier struggles to predict classes correctly. Looking at the confusion matrix for the training data, we can see that it's overfitting even more than the initial dataset results. The parameters that gave the best results on the validation data were: **Activation:** tanh, **alpha:** 0.05, **hidden layer sizes:** (100,), **learning rate:** adaptive, **max iter:** 500, **solver:** adam.

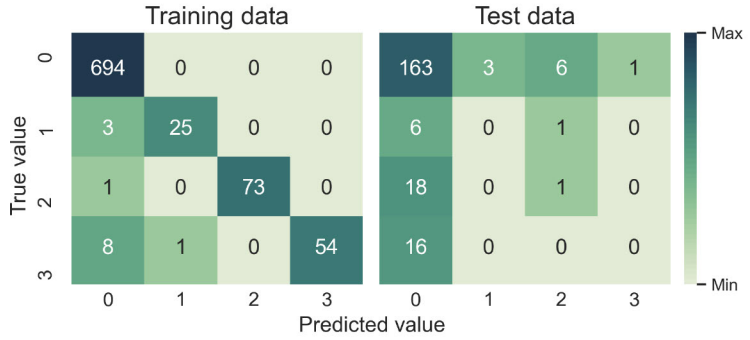



Figure 5.14: Confusion matrices for the MLPClassifier applied to the reduced training and test dataset.

### DecisionTreeClassifier

The DecisionTreeClassifier exhibits a marginally better F1 macro score of 0.298, decreasing the contrast between the train and test scores compared to the initial dataset results. Though, the confusion matrix in figure 5.15 does not present promising results. The model performs worse on the test data, with an F1 macro score of 0.242. The parameters that gave the best results on the validation data were: **Criterion:** entropy, **max depth:** 10, **max features:** log2, **min samples leaf:** 1, **min samples split:** 2, **splitter:** best.


Figure 5.15: Confusion matrices for the DecisionTreeClassifier applied to the reduced training and test dataset.

### ExtraTreeClassifier

Like the DecisionTreeClassifier, the ExtraTreeClassifier demonstrates an increase in the F1 macro score and an improvement in the contrast between training and testing scores. The two models display identical results for the reduced dataset, as we also saw with the initial dataset. The parameters that gave the best results on the validation data were:

**Criterion:** entropy, **max depth:** 10, **max features:** log2, **min samples leaf:** 1, **min samples split:** 2, **splitter:** best.


Figure 5.16: Confusion matrices for the ExtraTreeClassifier applied to the reduced training and test dataset.

### 5.3.3 Up-sampled dataset

An up-sampled version of the training data was created to explore potential model performance given balanced classes, effectively balancing the representation of minority classes using the Synthetic Minority Over-sampling Technique (SMOTE). Table 5.8 presents the evaluated metrics from a grid search cross-validation on this dataset.

Despite the cross-validation results yielding the highest scores among all the subsets, the confusion matrices in figure 5.18, 5.19 and 5.20, hints of that the models are overfitting. Looking at the results from predictions of the test data in table 5.9, we see a rather significant drop in the F1 macro score. SMOTE generates synthetic samples; hence, the resulting augmented dataset is inherently more complex. This complexity can create conditions for model overfitting as they tune their parameters to the noise

| Model | Accuracy | Precision | Recall | F1 score | F1-macro | ROC AUC |
|---|---|---|---|---|---|---|
| MLPC | 0.923 (0.016)<br>*1.000 (0.000)* | 0.924 (0.016)<br>*1.000 (0.000)* | 0.923 (0.016)<br>*1.000 (0.000)* | 0.923 (0.016)<br>*1.000 (0.000)* | 0.923 (0.016)<br>*1.000 (0.000)* | 0.983 (0.004)<br>*1.000 (0.000)* |
| DTC | 0.826 (0.019)<br>*1.000 (0.000)* | 0.826 (0.019)<br>*1.000 (0.000)* | 0.826 (0.019)<br>*1.000 (0.000)* | 0.823 (0.020)<br>*1.000 (0.000)* | 0.823 (0.020)<br>*1.000 (0.000)* | 0.884 (0.013)<br>*1.000 (0.000)* |
| ETC | 0.826 (0.019)<br>*1.000 (0.000)* | 0.826 (0.019)<br>*1.000 (0.000)* | 0.826 (0.019)<br>*1.000 (0.000)* | 0.823 (0.020)<br>*1.000 (0.000)* | 0.823 (0.020)<br>*1.000 (0.000)* | 0.884 (0.013)<br>*1.000 (0.000)* |

Table 5.8: Results from the extended grid search CV, using the up-sampled dataset. The mean and standard deviation validation scores are reported with the training scores in italics.

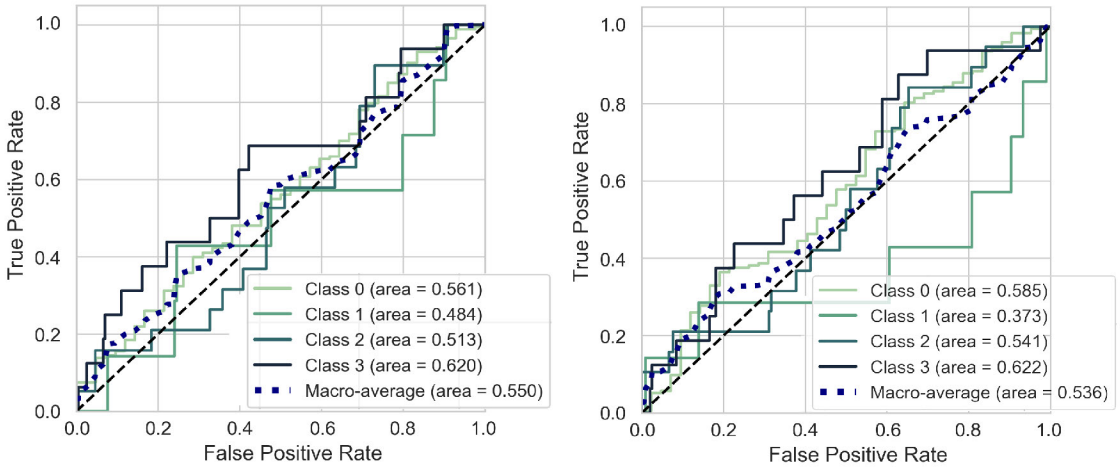
| Model | Accuracy | Precision | Recall | F1 score | F1-macro | ROC AUC |
|---|---|---|---|---|---|---|
| MLPC | 0.688 | 0.696 | 0.688 | 0.690 | 0.301 | 0.531 |
| DTC | 0.567 | 0.660 | 0.567 | 0.609 | 0.217 | 0.476 |
| ETC | 0.567 | 0.660 | 0.567 | 0.609 | 0.217 | 0.476 |

Table 5.9: Performance metrics of each of the best estimators after refitting with GridSearchCV on the up-sampled dataset, evaluated on the initial test set.

and artificial nuances of the synthetic samples. Consequently, while the models display superior performance on the SMOTE training data, they fail to generalize effectively to the original, unseen test data.

### MLPClassifier

As illustrated in 5.18, the MLPClassifier displays a marginal increase in correctly classified bleeding cases compared to the initial dataset. On the other hand, it shows a noticeable reduction of contrast between training and validation scores. The curves illustrated in figure 5.17 depict the trade-off between sensitivity (True Positive Rate) and specificity (False Positive Rate) for different decision thresholds. In an ideal scenario, we hope to see the ROC curve close to the upper left corner, indicating an effective class separation. The parameters that gave the best results on the validation data were: **Activation:** tanh, **alpha:** 0.0001, **hidden layer sizes:** (100,), **learning rate:** constant, **max iter:** 500, **solver:** adam.




(a) Initial dataset

(b) Up-sampled dataset

Figure 5.17: ROC curves and macro-averages for each class using the MLPClassifier trained on both the initial and up-sampled datasets. The testing was conducted on the test data set.

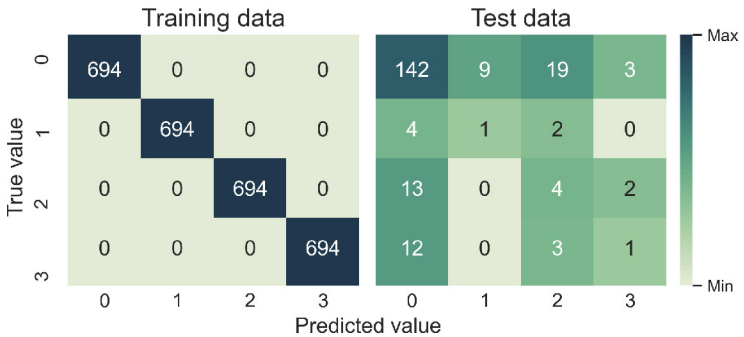



Figure 5.18: Confusion matrices for the MLPClassifier applied to the up-sampled training and initial test dataset.

### DecisionTreeClassifier

Figure 5.19 illustrates the predictions with the DecisionTreeClassifier refitted on the training data. Unlike the MLPClassifier, we see a decreased performance in predicting bleeding and non-bleeding patients. The parameters that gave the best results on the validation data were: **Criterion:** gini, **max depth:** None, **max features:** sqrt, **min samples leaf:** 1, **min samples split:** 2, **splitter:** random.

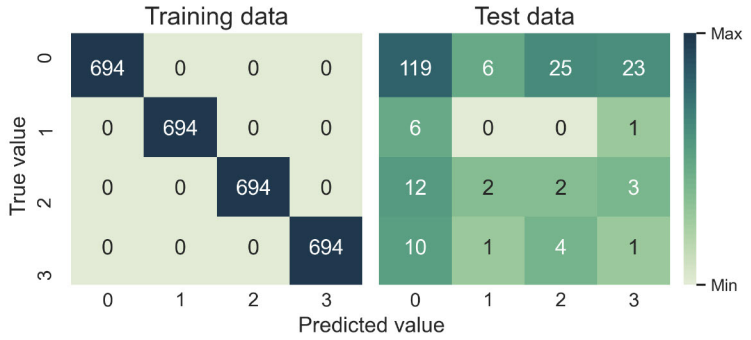



Figure 5.19: Confusion matrices for the DecisionTreeClassifier applied to the up-sampled training and initial test dataset.

### ExtraTreeClassifier

As anticipated, the ExtraTreeClassifier maintains identical results with its related DecisionTreeClassifier with the up-sampled dataset, showing a decreased contrast in train/validation scores. The parameters that gave the best results on the validation data were: **Criterion:** gini, **max depth:** None, **max features:** sqrt, **min samples leaf:** 1, **min samples split:** 2, **splitter:** random.

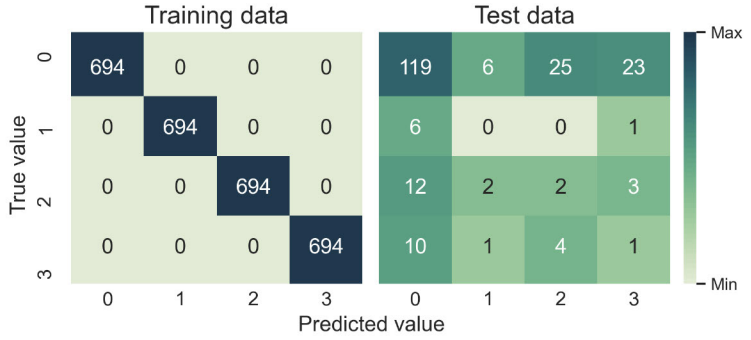



Figure 5.20: Confusion matrices for the ExtraTreeClassifier applied to the up-sampled training and test dataset.

## 5.3.4 reshuffled dataset

As discussed in chapter 4, we are dealing with a relatively small dataset, and the composition of the train and test subsets could affect the results. To ensure that the previous results were not simply a product of this particular split, an additional analysis is conducted on a reshuffled dataset, using the same 80/20 split as the initial dataset.

The evaluated metrics from the grid search cross-validation are presented in table 5.10, and the predictions made by the refitted models on the test data are presented in table 5.11.

| Model | Accuracy | Precision | Recall | F1 score | F1-macro | ROC AUC |
|---|---|---|---|---|---|---|
| MLPC | 0.695 (0.046)<br>*0.986 (0.016)* | 0.686 (0.025)<br>*0.986 (0.015)* | 0.695 (0.046)<br>*0.986 (0.016)* | 0.689 (0.032)<br>*0.985 (0.016)* | 0.278 (0.042)<br>*0.962 (0.040)* | 0.510 (0.041)<br>*0.996 (0.007)* |
| DTC | 0.587 (0.055)<br>*0.882 (0.011)* | 0.692 (0.028)<br>*0.926 (0.005)* | 0.587 (0.055)<br>*0.882 (0.011)* | 0.630 (0.041)<br>*0.892 (0.010)* | 0.289 (0.085)<br>*0.796 (0.012)* | 0.534 (0.061)<br>*0.991 (0.001)* |
| ETC | 0.587 (0.055)<br>*0.882 (0.011)* | 0.692 (0.028)<br>*0.926 (0.005)* | 0.587 (0.055)<br>*0.882 (0.011)* | 0.630 (0.041)<br>*0.892 (0.010)* | 0.289 (0.085)<br>*0.796 (0.012)* | 0.534 (0.061)<br>*0.991 (0.001)* |

Table 5.10: Results from the extended grid search CV, using the reshuffled dataset. The mean and standard deviation validation scores are reported with the training scores in italics.

| Model | Accuracy | Precision | Recall | F1 score | F1-macro | ROC AUC |
|---|---|---|---|---|---|---|
| MLPC | 0.498 | 0.699 | 0.498 | 0.562 | 0.289 | 0.618 |
| DTC | 0.530 | 0.670 | 0.530 | 0.587 | 0.219 | 0.478 |
| ETC | 0.530 | 0.670 | 0.530 | 0.587 | 0.219 | 0.478 |

Table 5.11: Performance metrics of each of the best estimators after refitting with GridSearchCV, evaluated on the reshuffled test set.

### MLPClassifier

The MLPClassifier demonstrates a decrease across all metric scores compared to the results from the corresponding initial dataset. However, the F1 macro score from the test data is marginally better. The confusion matrices in figure 5.21 show an increase in correctly classified major bleeding cases. The parameters that gave the best results on the validation data were: **Activation:** tanh, **alpha:** 0.0001, **hidden layer sizes:** (50, 50, 50), **learning rate:** constant, **max iter:** 500, **solver:** adam.

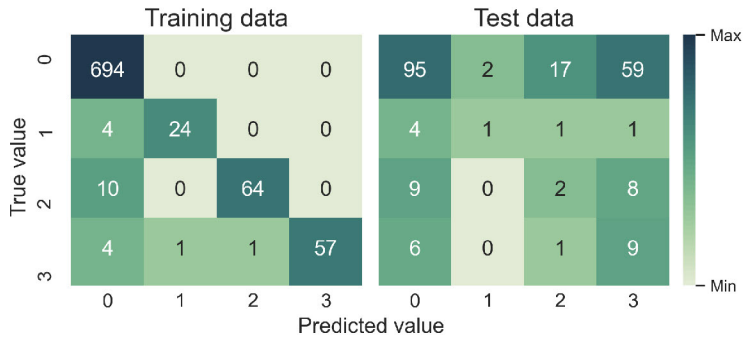



Figure 5.21: Confusion matrices for the MLPClassifier applied to the reshuffled training and test dataset.


Figure 5.22: Confusion matrices for the DecisionTreeClassifier applied to the reshuffled training and test dataset.

#### DecisionTreeClassifier

The DecisionTreeClassifier shows an improvement across the precision, F1 macro, and ROC AUC scores and a decrease in accuracy, recall, and F1 scores. The F1 macro score decreases when predicting the test data. The parameters that gave the best results on the validation data were: **Criterion:** entropy, **max depth:** None, **max features:** sqrt, **min samples leaf:** 1, **min samples split:** 5, **splitter:** best.

#### ExtraTreeClassifier

Like the results for the up-sampled dataset, the ExtraTreeClassifier shows less contrast between training and validation scores with the reshuffled data. The parameters that gave the best results on the validation data were: **Criterion:** entropy, **max depth:** None, **max features:** sqrt, **min samples leaf:** 1, **min samples split:** 5, **splitter:** best.


Figure 5.23: Confusion matrices for the ExtraTreeClassifier applied to the reshuffled training and test dataset.

### 5.3.5 90/10-split dataset

A dataset with a more significant training portion was also constructed to evaluate how increased exposure to training data would impact the performance of the models. A more extensive training set can help improve the model's generalization ability. Table 5.12 displays the results from the cross-validation on this expanded training dataset. The predictions made by the refitted models on the test data are presented in table 5.13.

| Model | Accuracy | Precision | Recall | F1 score | F1-macro | ROC AUC |
|---|---|---|---|---|---|---|
| MLPC | 0.714 (0.052)<br>*0.930 (0.028)* | 0.701 (0.039)<br>*0.934 (0.025)* | 0.714 (0.052)<br>*0.930 (0.028)* | 0.705 (0.033)<br>*0.928 (0.029)* | 0.295 (0.084)<br>*0.831 (0.073)* | 0.584 (0.074)<br>*0.972 (0.018)* |
| DTC | 0.664 (0.062)<br>*1.000 (0.000)* | 0.679 (0.032)<br>*1.000 (0.000)* | 0.664 (0.062)<br>*1.000 (0.000)* | 0.669 (0.046)<br>*1.000 (0.000)* | 0.285 (0.041)<br>*1.000 (0.000)* | 0.532 (0.032)<br>*1.000 (0.000)* |
| ETC | 0.664 (0.062)<br>*1.000 (0.000)* | 0.679 (0.032)<br>*1.000 (0.000)* | 0.664 (0.062)<br>*1.000 (0.000)* | 0.669 (0.046)<br>*1.000 (0.000)* | 0.285 (0.041)<br>*1.000 (0.000)* | 0.532 (0.032)<br>*1.000 (0.000)* |

Table 5.12: Results from the extended grid search CV, using the 90/10-split dataset. The mean and standard deviation validation scores are reported with the training scores in italics.

| Model | Accuracy | Precision | Recall | F1 score | F1-macro | ROC AUC |
|---|---|---|---|---|---|---|
| MLPC | 0.750 | 0.661 | 0.750 | 0.701 | 0.250 | 0.513 |
| DTC | 0.750 | 0.667 | 0.750 | 0.706 | 0.244 | 0.505 |
| ETC | 0.750 | 0.667 | 0.750 | 0.706 | 0.244 | 0.505 |

Table 5.13: Performance metrics of each of the best estimators after refitting with GridSearchCV, evaluated on the 90/10-split test set.

### MLPClassifier

When comparing the cross-validation scores with the initial dataset, the MLPClassifier's performance metrics display minimal difference, except the F1 macro score showing a dip and the ROC AUC score seeing an improvement. The confusion matrices in figure 5.24 display a moderate ability to predict the classes on the training data correctly. The parameters that gave the best results on the validation data were: **Activation:** tanh, **alpha:** 0.0001, **hidden layer sizes:** (50, 50, 50), **learning rate:** adaptive, **max iter:** 500, **solver:** adam.


Figure 5.24: Confusion matrices for the MLPClassifier applied to the 90/10-split training and test dataset.

### DecisionTreeClassifier

The DecisionTreeClassifier displays a marginally lower score across all metrics than the initial dataset, and the confusion matrices in figure 5.25 indicates overfitting. The predictions made on the test set do not differ significantly from the comparable results

with the initial dataset, given the change in the ratio between training and testing data. The parameters that gave the best results on the validation data were: **Criterion:** entropy, **max depth:** None, **max features:** log2, **min samples leaf:** 1, **min samples split:** 2, **splitter:** best.

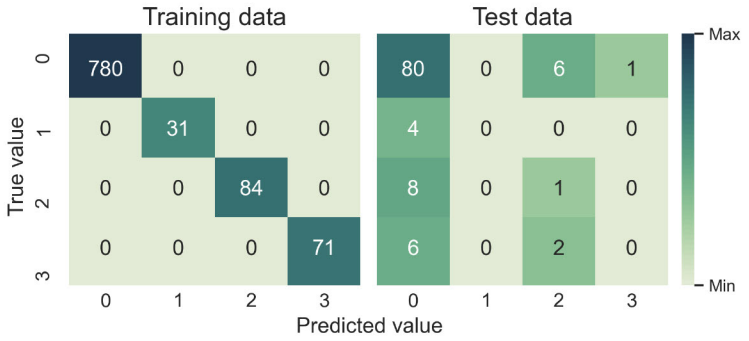



Figure 5.25: Confusion matrices for the DecisionTreeClassifier applied to the 90/10-split training and test dataset.

### ExtraTreeClassifier

With the 90/10 split dataset, the ExtraTreeClassifier does not show any differentiation from the DecisionTreeClassifier. The parameters that gave the best results on the validation data were: **Criterion:** entropy, **max depth:** None, **max features:** log2, **min samples leaf:** 1, **min samples split:** 2, **splitter:** best.

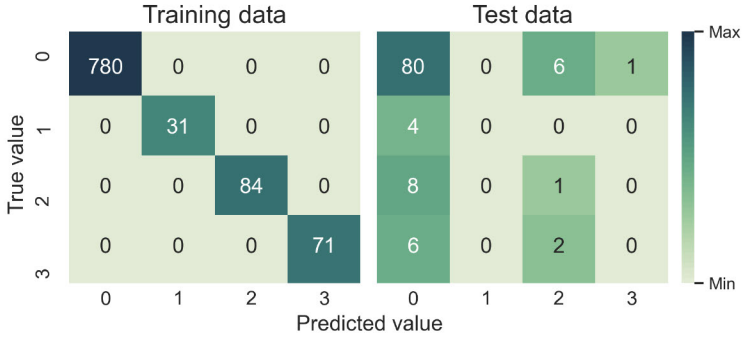



Figure 5.26: Confusion matrices for the ExtraTreeClassifier applied to the 90/10-split training and test dataset.

## 5.4 Performance on entire dataset

This section presents the performance of the top models tested on the entire dataset. The motivation for this approach is rooted in the persistently underwhelming results shown across the various subsets analyzed so far. These outcomes have suggested that the available data may be insufficient for generating robust, predictive models. This investigation aims to comprehensively evaluate the model's capabilities by fitting the models to the full dataset. A persisting poor performance on the entire dataset will further substantiate the hypothesis that insufficient data is a major limiting factor in

model development. The grid search results performed on the whole dataset are presented in table 5.14.

By comparing the performance of the top models on the entire dataset, we can better understand how these standard ML models differ in their effectiveness in predicting bleeding events, providing a robust foundation for answering RQ2.

| Model | Accuracy | Precision | Recall | F1 score | F1-macro | ROC AUC |
|---|---|---|---|---|---|---|
| MLPC | 0.759 (0.024)<br>*0.936 (0.026)* | 0.690 (0.036)<br>*0.938 (0.024)* | 0.759 (0.024)<br>*0.936 (0.026)* | 0.717 (0.021)<br>*0.930 (0.030)* | 0.287 (0.074)<br>*0.844 (0.072)* | 0.575 (0.060)<br>*0.984 (0.008)* |
| DTC | 0.670 (0.046)<br>*1.000 (0.000)* | 0.682 (0.024)<br>*1.000 (0.000)* | 0.670 (0.046)<br>*1.000 (0.000)* | 0.675 (0.034)<br>*1.000 (0.000)* | 0.276 (0.043)<br>*1.000 (0.000)* | 0.522 (0.030)<br>*1.000 (0.000)* |
| ETC | 0.670 (0.046)<br>*1.000 (0.000)* | 0.682 (0.024)<br>*1.000 (0.000)* | 0.670 (0.046)<br>*1.000 (0.000)* | 0.675 (0.034)<br>*1.000 (0.000)* | 0.276 (0.043)<br>*1.000 (0.000)* | 0.522 (0.030)<br>*1.000 (0.000)* |

Table 5.14: Results from the extended grid search CV, using the entire dataset. The mean and standard deviation validation scores are reported with the training scores in italics.

| Model | Accuracy | Precision | Recall | F1 score | F1-macro | ROC AUC |
|---|---|---|---|---|---|---|
| MLPC | 0.842 | 0.861 | 0.842 | 0.794 | 0.451 | 0.899 |
| DTC | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |
| ETC | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 |

Table 5.15: Performance metrics of each of the best estimators after refitting with GridSearchCV, evaluated on the entire test set.

### MLPClassifier

The MLPClassifier experiences a slight increase in accuracy, recall, F1, and ROC AUC score, leaving the precision score with a slight dip. In this instance, the confusion matrices in figure 5.27 are identical, which is expected as were using the same dataset for training and testing. The parameters that gave the best results on the validation data were: **Activation:** relu, **alpha:** 0.0001, **hidden layer sizes:** (100,), **learning rate:** constant, **max iter:** 500, **solver:** adam.

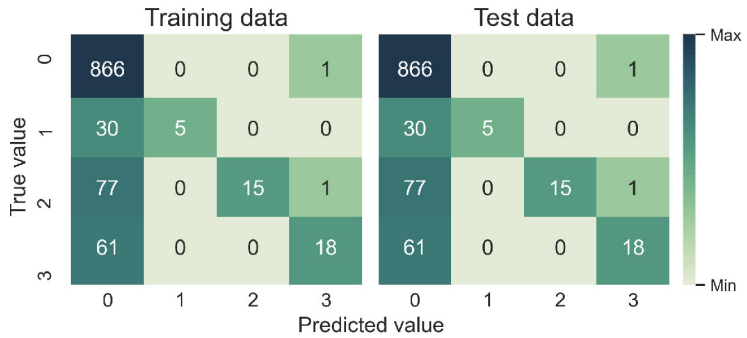



Figure 5.27: Confusion matrices for the MLPClassifier applied to the full dataset.

### DecisionTreeClassifier

For the DecisionTreeClassifier, we see an all-over decrease across all cross-validation metrics. As we saw in the preceding section, the confusion matrices in figure 5.28 illustrate that the model might be overfitting the data. The parameters that gave the best results on the validation data were: **Criterion:** gini, **max depth:** None, **max features:** log2, **min samples leaf:** 1, **min samples split:** 2, **splitter:** random.

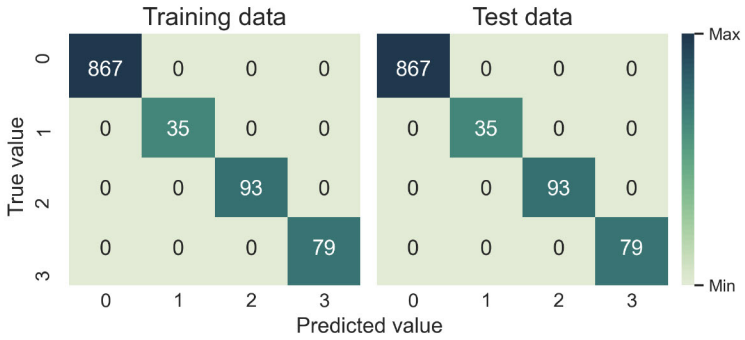



Figure 5.28: Confusion matrices for the DecisionTreeClassifier applied to the full dataset.

### ExtraTreeClassifier

The ExtraTreeClassifier demonstrated the same results as the DecisionTreeClassifier. The parameters that gave the best results on the validation data were: **Criterion:** gini, **max depth:** None, **max features:** log2, **min samples leaf:** 1, **min samples split:** 2, **splitter:** random.

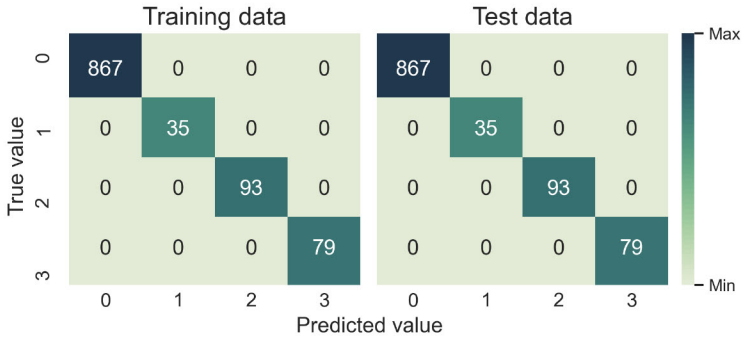



Figure 5.29: Confusion matrices for the ExtraTreeClassifier applied to the full dataset.

# Chapter 6

# Discussion

This chapter analyzes and interprets the results obtained from the performance evaluations of the 12 selected machine learning models. The discussion aims to answer the following research questions introduced in chapter 1:

1. To what extent does the dataset allow for the differentiation of classes with state-of-the-art machine learning methods?
2. How do standard machine learning models compare in their effectiveness in predicting bleeding events in thrombosis patients?

## 6.1 Class differentiation

The results from the experimentation suggest a significant challenge in class differentiation within the datasets. In regards to the first research question, the majority of models encountered issues with overfitting and class differentiation.

In the individual analysis of the results, it was found that most models showed potential overfitting issues and difficulties distinguishing between classes. The top three models, MLPClassifier, DecisionTreeClassifier, and ExtraTreeClassifier, were selected based on their F1 macro score, ensuring the model's performance was not overly influenced by its ability to predict the most common class.

The MLPClassifier's F1 macro score on the reduced dataset decreased noticeably, indicating struggles with predicting classes correctly. The Decision Tree and ExtraTreeClassifiers showed minor improvements and less overfitting but presented similar results.

Using SMOTE up-sampling for balanced class distribution, all three models seemed to overfit and struggled to generalize to the unseen test data. However, the MLPClassifier did show a slight increase in correctly classified bleeding cases, suggesting potential improvements with a balanced dataset. The reshuffled dataset showed decreased performance for the MLPClassifier and improved performance for the Decision Tree and ExtraTreeClassifiers. However, the contrast between training and validation scores for all models raised concerns about their generalizability. Increasing the training dataset to a 90/10 split led to a slight degradation in performance in the MLPClassifier and marginally lower scores in the Decision Tree and ExtraTreeClassifiers.

The MLPClassifier showed a ROC AUC score of 0.529 on the initial dataset and a slightly higher score of 0.531 on the up-sampled dataset. These scores, marginally above

the 0.5 baselines, indicate that even the best-performing model struggles to separate the classes in the data effectively.

However, we can see that the ROC curves of the MLPClassifier are significantly closer to the diagonal line than the desired upper left corner. This affirms the difficulties the model has when trying to discriminate between classes. The similarity of the curves in figure 5.17(a) and 5.17(b) suggests that addressing the class imbalance through up-sampling did not improve the model's class differentiation ability.

This difficulty in class differentiation could stem from various factors. Further studies and exploration are necessary to verify the precise reasons and potentially find solutions to this challenge.

When examining the performance of the models, it is also essential to consider the potential impact of the preprocessing steps. Handling missing values is a significant preprocessing element that could impact the class differentiation ability. A high proportion of the dataset was missing and imputed using mean values. While this is a common strategy for handling missing data, it has potential drawbacks.

Imputation with mean values ignores any correlation between features, potentially adding noise and reducing the overall variability of the dataset. This can distort the relationships between features and obscure the patterns the models try to learn. It's plausible that this could be part of why the models struggled to differentiate between classes effectively.

## 6.2 Prediction effectiveness

The results showed a degree of variation when evaluating the efficacy of the MLP, Decision Tree, and ExtraTreeClassifier in predicting bleeding events among thrombosis patients. The graph in figure 6.2 illustrates the changes in the performance across the different datasets.

### 6.2.1 MLPClassifier

The MLPClassifier emerged as the top-performing model during the initial grid search, with the highest F1 macro score. However, extended grid search on the initial dataset led to a decrease in most performance metrics, except precision and F1 macro. The MLPClassifier experienced a further decline in its F1 macro score in the reduced feature dataset, signifying difficulties in class predictions.

Applying SMOTE up-sampling for class distribution balance showed a slight improvement in predicting bleeding events. Still, the MLPClassifier struggled with generalizing to unseen data. Its performance on the reshuffled and 90/10 split datasets also decreased, indicating potential overfitting.

Surprisingly, the MLPClassifier shows a drop in the F1 macro cross-validation score. This decrease suggests that the model may not predict effectively across all classes. The dip in precision score further indicates potential limitations in the MLPClassifier's ability to predict bleeding instances correctly. The results are otherwise not particularly promising, hinting at a bias towards the no-bleeding class.

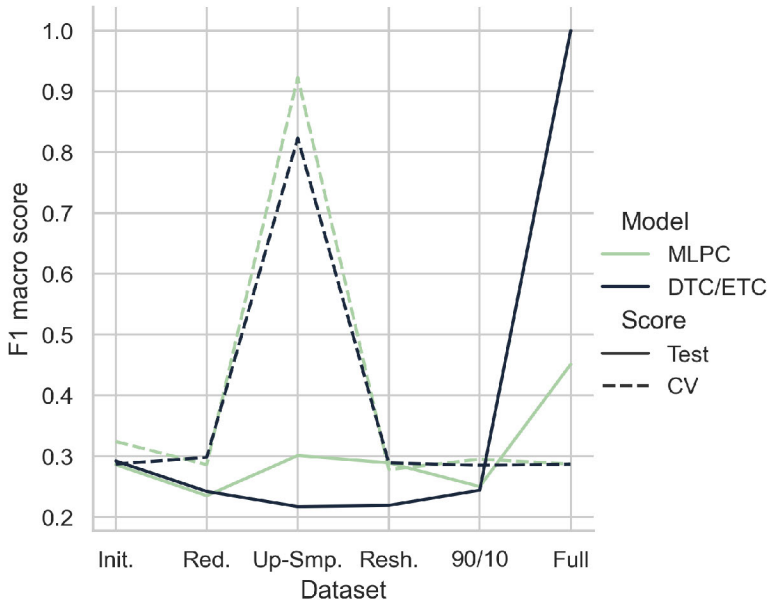



Figure 6.1: Line graph illustrating the changes in cross-validation and test scores for the top three models across the dataset variations.

### 6.2.2 DecisionTreeClassifier

The DecisionTreeClassifier showed promise in certain areas, achieving a precision and recall score of 0.68 during the initial grid search. However, a less impressive ROC AUC score of 0.528 suggest limitations in its ability to distinguish between different classes effectively. The DecisionTreeClassifier had a considerable difference between its testing and validation scores, indicating that it is likely overfitting to the training data. This is typical for a high-variance model, suggesting that the model could be too tailored to the training data, which might include noise and outliers, and thus struggles to generalize effectively to unseen data. Despite these limitations, the DecisionTreeClassifier has a relatively fast fit time of 13.36 ms, which could make it a practical choice for real-time predictions. However, the issue of overfitting remains a concern.

The DecisionTreeClassifiers' performance remained consistent in both the initial and extended grid search, indicating the optimal parameters for this classifier being covered in the initial search range. The DecisionTreeClassifier improved F1 macro and ROC AUC scores on the reduced and reshuffled datasets, signifying a better balance across classes and discriminative ability between them.

More importantly, is a reduced contrast between the train and validation scores. This indicates that the model has less overfitting. However, overfitting became apparent when testing the entire dataset, characterized by a decrease across all metrics and a substantial difference between testing and training scores.

### 6.2.3 ExtraTreeClassifier

Similar to the DecisionTreeClassifier, the ExtraTreeClassifier performed well in accuracy and recall during the initial grid search, and the performance remained consistent in both

initial and extended grid searches. This suggests that the models could extract similar patterns from the dataset. However, the ROC AUC score of 0.515 presented potential difficulties in class discrimination.

The ExtraTreeClassifier's performance improved with the reduced dataset, showing an increased F1 macro score and lower contrast between training and validation scores. Performance remained similar to the DecisionTreeClassifier with the up-sampled dataset, showing a decreased difference in train/validation scores. A more balanced dataset indicates a potential for these tree-based models to excel. Still, a decrease across all metrics and a substantial increase in contrast between training and validation scores on the full dataset suggested potential overfitting, similar to what was observed with the DecisionTreeClassifier.

**In summary,** the MLPClassifier showed more variation in performance across different datasets variations but boasted the highest score in most cases. In contrast, the Decision Tree and ExtraTreeClassifiers exhibited more consistent performances. However, all classifiers showed signs of overfitting and struggled with generalizing to unseen data, indicating a potential need for more data or improved feature selection techniques. Using the F1 macro score as a selection criterion effectively identified models with balanced performance across all classes, not just the majority class.

# Chapter 7

# Conclusion

This study investigated the extent to which the dataset facilitates the differentiation of classes using state-of-the-art ML methods and the application of ML models to predict bleeding events in patients with thrombosis. However, due to the observed overfitting characteristics and subpar results, questions were raised regarding the sufficiency of the dataset.

The study initially evaluated 12 different models, revealing substantial variations across them. The top-performing models, MLPClassifier, DecisionTreeClassifier, and ExtraTreeClassifier, were identified based on their F1 macro scores, an essential metric for unbalanced multiclass classification problems. Despite attempts to further enhance the predictive performance of these models through an extended hyperparameter tuning process, the results were not as promising as expected.

These findings, connected with the persistent issue of overfitting, pointed toward possible weaknesses in the dataset. While there were some improvements in prediction accuracy, no single model consistently outperformed others across all metrics. This underscores the complexity and difficulty of the task.

The comparative analysis provided a deeper understanding of how different ML models can predict bleeding events in thrombosis patients. It offers valuable information for enhancing the prediction effectiveness, which could improve individualized treatment strategies for these patients. However, the results also highlighted the challenges of applying ML techniques in a complex medical context.

The study has also underscored the importance of understanding the intricacies of hyperparameter tuning and its impact on model performance. It has demonstrated that while the dataset allows some degree of class differentiation using state-of-the-art ML methods, the impact of missing data and the limitations inherent in the models reduced their effectiveness in predicting bleeding events in thrombosis patients.

Despite the less-than-optimal results, the study has contributed to the field by comprehensively comparing ML models in a healthcare context, specifically in predicting bleeding risk in thrombosis patients. It has shown that ML techniques can be applied in healthcare prediction, ultimately improving patient outcomes. However, the outcomes also emphasized the pressing need to address data-related issues, such as a lack of sufficient data or a high volume of missing data.

While the research results did not meet initial expectations, the study has provided valuable insights into the challenges and complexities of applying ML models in predicting bleeding events in thrombosis patients, especially in the light of significant missing data.

It has highlighted areas for improvement and has paved the way for future research in this area. The findings of this research can potentially guide future studies and contribute to the ongoing efforts to improve healthcare outcomes through ML, particularly emphasizing the importance of high-quality, sufficient datasets.

## 7.1 Future work and recommendations

While this study has made meaningful steps in applying ML techniques to predict bleeding risk in thrombosis patients, there are several possible paths for future work.

Handling missing data with mean imputation could contribute to the underwhelming performance of the models. This underlines the importance of examining data preprocessing choices and their potential effects on model performance. More advanced imputation techniques could potentially mitigate some of these issues.

Future studies could focus on exploring advanced imputation methods, improving the quality of data collection to reduce the extent of missing data, and extracting textual data using natural language processing.

Secondly, further research could explore using more advanced ML models or ensemble methods to improve prediction accuracy. Additionally, incorporating more diverse and extensive datasets could enhance the robustness of the models.

By continuing to explore these paths, we can leverage ML to improve patient outcomes in thrombosis and other medical conditions.

# Bibliography


Palareti, G., Leali, N., Coccheri, S., Poggi, M., Manotti, C., D'Angelo, A., Pengo, V., Erba, N., Moia, M., Ciavarella, N., Devoto, G., Berrettini, M., & Musolesi, S. (1996). Bleeding complications of oral anticoagulant treatment: An inception-cohort, prospective collaborative study (ISCOAT). *The Lancet*, *348*(9025), 423–428. https://doi.org/10.1016/S0140-6736(96)01109-9

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.

Kahn, S. R., Hirsch, A., & Shrier, I. (2002). Effect of postthrombotic syndrome on health-related quality of life after deep venous thrombosis. *Archives of Internal Medicine*, *162*(10), 1144–1148. https://doi.org/10.1001/archinte.162.10.1144

Prandoni, P., Lensing, A. W. A., Piccioli, A., Bernardi, E., Simioni, P., Girolami, B., Marchiori, A., Sabbion, P., Prins, M. H., Noventa, F., & Girolami, A. (2002). Recurrent venous thromboembolism and bleeding complications during anticoagulant treatment in patients with cancer and venous thrombosis. *Blood*, *100*(10), 3484–3488. https://doi.org/10.1182/blood-2002-01-0108

Kyrle, P. A., & Eichinger, S. (2005). Deep vein thrombosis. *The Lancet*, *365*(9465), 1163–1174. https://doi.org/10.1016/S0140-6736(05)71880-8

Stain, M., Schönauer, V., Minar, E., Bialonczyk, C., Hirschl, M., Weltermann, A., Kyrle, P., & Eichinger, S. (2005). The post-thrombotic syndrome: Risk factors and impact on the course of thrombotic disease. *Journal of Thrombosis and Haemostasis*, *3*(12), 2671–2676. https://doi.org/10.1111/j.1538-7836.2005.01648.x

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

Tapson, V. F. (2008). Acute pulmonary embolism. *New England Journal of Medicine*, *358*(10), 1037–1052. https://doi.org/10.1056/NEJMra072753

Beckman, M. G., Hooper, W. C., Critchley, S. E., & Ortel, T. L. (2010). Venous thromboembolism: A public health concern. *American Journal of Preventive Medicine*, *38*(4), S495–S501. https://doi.org/10.1016/j.amepre.2009.12.017

Jiménez, D., Aujesky, D., Moores, L., Gómez, V., Lobo, J. L., Uresandi, F., Otero, R., Monreal, M., Muriel, A., Yusen, R. D., & RIETE Investigators. (2010). Simplification of the pulmonary embolism severity index for prognostication in patients with acute symptomatic pulmonary embolism. *Archives of Internal Medicine*, *170*(15), 1383–1389. https://doi.org/10.1001/archinternmed.2010.199

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A.,

Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* [OCLC: ocn827083441]. Springer.

Fernandez, M., Hogue, S. L., Preblick, R., & Kwong, W. J. (2015). Review of the cost of venous thromboembolism. *ClinicoEconomics and Outcomes Research*, 451. https://doi.org/10.2147/CEOR.S85635

Kooiman, J., van Hagen, N., Iglesias del Sol, A., Planken, E. V., Lip, G. Y. H., van der Meer, F. J. M., Cannegieter, S. C., Klok, F. A., & Huisman, M. V. (2015). The HAS-BLED score identifies patients with acute venous thromboembolism at high risk of major bleeding complications during the first six months of anticoagulant treatment. *PLoS ONE*, *10*(4), e0122520. https://doi.org/10.1371/journal.pone.0122520

Brækkan, S. K., Grosse, S. D., Okoroh, E. M., Tsai, J., Cannegieter, S. C., Næss, I. A., Krokstad, S., Hansen, J.-B., & Skjeldestad, F. E. (2016). Venous thromboembolism and subsequent permanent work-related disability. *Journal of Thrombosis and Haemostasis*, *14*(10), 1978–1987. https://doi.org/10.1111/jth.13411

Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., Wang, Y., Dong, Q., Shen, H., & Wang, Y. (2017). Artificial intelligence in healthcare: Past, present and future. *Stroke and Vascular Neurology*, *2*(4). https://doi.org/10.1136/svn-2017-000101

Beunza, J.-J., Puertas, E., García-Ovejero, E., Villalba, G., Condes, E., Koleva, G., Hurtado, C., & Landecho, M. F. (2019). Comparison of machine learning algorithms for clinical event prediction (risk of coronary heart disease). *Journal of Biomedical Informatics*, *97*, 103257. https://doi.org/10.1016/j.jbi.2019.103257

Johns Hopkins Medicine. (2019, November 19). *Thrombosis*. https://www.hopkinsmedicine.org/health/conditions-and-diseases/thrombosis

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Jøssang, I. (2020). *"legge om livet". en kvalitativ studie av erfaringer fra rehabilitering og det å leve med synsvansker etter hjerneslag.* (Master thesis). https://www.duo.uio.no/handle/10852/80707

Nafee, T., Gibson, C. M., Travis, R., Yee, M. K., Kerneis, M., Chi, G., AlKhalfan, F., Hernandez, A. F., Hull, R. D., Cohen, A. T., Harrington, R. A., & Goldhaber, S. Z. (2020). Machine learning to predict venous thrombosis in acutely ill medical patients. *Research and Practice in Thrombosis and Haemostasis*, *4*(2), 230–237. https://doi.org/10.1002/rth2.12292

Abbas, K. (2021). Predicting thrombosis with machine learning. https://hdl.handle.net/11250/2770341

Badescu, M. C., Ciocoiu, M., Badulescu, O. V., Vladeanu, M.-C., Bojan, I. B., Vlad, C. E., & Rezus, C. (2021). Prediction of bleeding events using the VTE-BLEED risk score in patients with venous thromboembolism receiving anticoagulant therapy (review). *Experimental and Therapeutic Medicine*, *22*(5), 1344. https://doi.org/10.3892/etm.2021.10779

De Winter, M. A., Van Es, N., Büller, H. R., Visseren, F. L., & Nijkeuter, M. (2021). Prediction models for recurrence and bleeding in patients with venous thromboembolism: A systematic review and critical appraisal. *Thrombosis Research*, *199*, 85–96. https://doi.org/10.1016/j.thromres.2020.12.031

Alanazi, A. (2022). Using machine learning for healthcare challenges and opportunities. *Informatics in Medicine Unlocked*, *30*, 100924. https://doi.org/10.1016/j.imu.2022.100924

Jin, S., Qin, D., Liang, B.-S., Zhang, L.-C., Wei, X.-X., Wang, Y.-J., Zhuang, B., Zhang, T., Yang, Z.-P., Cao, Y.-W., Jin, S.-L., Yang, P., Jiang, B., Rao, B.-Q., Shi, H.-P., & Lu, Q. (2022). Machine learning predicts cancer-associated deep vein thrombosis using clinically available variables. *International Journal of Medical Informatics*, *161*, 104733. https://doi.org/10.1016/j.ijmedinf.2022.104733

National Heart, Lung, and Blood Institute. (2022, September 19). *Venous thromboembolism - what is venous thromboembolism?* https://www.nhlbi.nih.gov/health/venous-thromboembolism

pandas development team, T. (2022). *Pandas-dev/pandas: Pandas* (Version 1.5.2). Zenodo. https://doi.org/10.5281/zenodo.7344967

Cleveland Clinic. (2023, January 16). *Thrombosis illustration.* https://my.clevelandclinic.org/health/diseases/22242-thrombosis

Fischer, S., Meisinger, C., Linseisen, J., Berghaus, T. M., & Kirchberger, I. (2023). Depression and anxiety up to two years after acute pulmonary embolism: Prevalence and predictors. *Thrombosis Research*, *222*, 68–74. https://doi.org/10.1016/j.thromres.2022.12.013

Mora, D., Mateo, J., Nieto, J. A., Bikdeli, B., Yamashita, Y., Barco, S., Jimenez, D., Demelo-Rodriguez, P., Rosa, V., Yoo, H. H. B., Sadeghipour, P., Monreal, M., & Investigators,
bibinitperiod t. R. I. d. E. T. ( (2023). Machine learning to predict major bleeding during anticoagulation for venous thromboembolism: Possibilities and limitations. *British Journal of Haematology*, *201*(5), 971–981. https://doi.org/10.1111/bjh.18737

scikit-learn. (2023a). *1.17. neural network models (supervised)* [Scikit-learn]. https://scikit-learn/stable/modules/neural_networks_supervised.html

scikit-learn. (2023b). *1.9. naive bayes* [Scikit-learn]. https://scikit-learn/stable/modules/naive_bayes.html

scikit-learn. (2023c). *3.3. metrics and scoring: Quantifying the quality of predictions* [Scikit-learn]. https://scikit-learn/stable/modules/model_evaluation.html

scikit-learn. (2023d). *Sklearn.linear_model.RidgeClassifier* [Scikit-learn]. https://scikit-learn/stable/modules/generated/sklearn.linear_model.RidgeClassifier.html

scikit-learn. (2023e). *Sklearn.neighbors.NearestCentroid* [Scikit-learn]. https://scikit-learn/stable/modules/generated/sklearn.neighbors.NearestCentroid.html

Shohat, N., Ludwick, L., Sherman, M. B., Fillingham, Y., & Parvizi, J. (2023). Using machine learning to predict venous thromboembolism and major bleeding events following total joint arthroplasty. *Scientific Reports*, *13*, 2197. https://doi.org/10.1038/s41598-022-26032-1

# Appendix A

# Translation of diagnoses

Translations of the thrombosis diagnoses are presented in table A.1.

| **Diagnosis (Norwegian)** | **Diagnosis (English)** |
|---|---|
| DVT | Deep Vein Thrombosis |
| Muskelvenetrombose | Muscle Vein Thrombosis |
| Vena porta trombose | Portal Vein Thrombosis |
| Vena hepatis trombose | Hepatic Vein Thrombosis |
| Vena mesenterica trombose | Mesenteric Vein Thrombosis |
| Vena lienalis trombose | Splenic Vein Thrombosis |
| Vena cava trombose | Inferior Vena Cava Thrombosis |
| Lungeemboli | Pulmonary Embolism |
| Vena ovarica trombose | Ovarian Vein Thrombosis |
| Vena renalis trombose | Renal Vein Thrombosis |
| Annet | Other |
| Overfladisk tromboflebitt | Superficial Thrombophlebitis |
| Overarm trombose | Upper Arm Thrombosis |
| Vena jugularis trombose | Jugular Vein Thrombosis |

Table A.1: Translation of thrombosis diagnosis from Norwegian to English

# Appendix B

# Feature reduction results

Table B.1 presents the features with zero importance according to the trained random forest model.

| **Feature** | **Importance** |
|---|---|
| hospitalization_Very_extended_stay | 0.0 |
| Other_diagnose_cause_Infection | 0.0 |
| Flight >8h | 0.0 |
| Familial_relationship_Second-degree relative | 0.0 |
| Familial_relationship_Unknown | 0.0 |
| treatment_Trombolyse | 0.0 |
| treatment_Marevan 2 | 0.0 |
| Trauma description | 0.0 |
| Other_diagnose_cause_IBD | 0.0 |
| Other_diagnose_cause_Obesity | 0.0 |
| Travel by vehicle >4h | 0.0 |
| treatment_Acetylsalisylsyre | 0.0 |
| Other_diagnose_cause_Peripheral venous catheter | 0.0 |
| Other_diagnose_cause_Varicose veins | 0.0 |
| Days since previous thrombosis 1 | 0.0 |
| Days since previous thrombosis 2 | 0.0 |
| Diagnosis_group2 | 0.0 |
| treatment_Dalteparin 3 | 0.0 |
| Flight <4h | 0.0 |
| Thrombophilia diagnose | 0.0 |
| hospitalization_Never_hospitalized | 0.0 |
| treatment_Apixaban 2 | 0.0 |

| **Feature** | **Importance** |
|---|---|
| Had previous thrombosis 1 | 0.0 |
| treatment_Dabigatran 2 | 0.0 |
| treatment_Edoksaban 2 | 0.0 |
| treatment_Enoksaparin 2 | 0.0 |
| Cancer_code_RS and MT | 0.0 |
| Cancer_code_SCT | 0.0 |
| CRP_is_filled | 0.0 |
| treatment_Rivaroxaban 3 | 0.0 |
| hospitalization_Short_stay | 0.0 |
| Cancer_code_CNS | 0.0 |
| Cancer_code_ET | 0.0 |
| treatment_Rivaroxaban 2 | 0.0 |
| Have myeloproliferative disease | 0.0 |
| Cancer_code_MultyPrime | 0.0 |
| Flight 4-8h | 0.0 |
| Had previous thrombosis 2 | 0.0 |

Table B.1: Features with zero importance reported by the random forest classifier.