UNIVERSITAT DE BARCELONA

Final Degree Project

Biomedical Engineering Degree

“Use of Machine Learning and SNOMED CT Encoded Health Problems to Predict Hospital Discharge Diagnoses”

Barcelona, 11th June 2025

Author: Cindy Chen

Director: Santiago Frid

Tutor: Juan Barrios

ABSTRACT

The accurate classification of discharge diagnoses is a critical step in clinical decision-making, as it has direct effect on patient care, hospital management, and administrative tasks. Traditionally, diagnostic coding has been a manual and time-consuming process, typically done after a patient is discharged, which could lead to delays for subsequent processes such as billing, reporting, and care optimization. Recently, the Hospital Clínic de Barcelona has integrated a structured list of health problems coded in SNOMED CT into the Electronic Health Record (EHR) from the beginning of the patient's hospitalization. This development has enabled the reuse of structured clinical data throughout the care process and has opened the door for predictive tools using Machine Learning (ML).

The goal of this research is to determine whether there's a significant relationship between reported health problems and the final ICD-10 discharge diagnoses. To explore this, data obtained from the Hospital Clínic de Barcelona was analysed, incorporating information from various clinical sources, such as demographics, laboratory results, prescriptions, and admissions records. Feature engineering was also carried out and methods based on decision trees, along with ANOVA tests, were used to identify the most relevant input variables. Subsequently, several supervised ML models, including Decision Trees (DTs), Random Forest (RF), and XGBoost were trained and evaluated.

The best performing model, a Decision Tree classifier, achieved an accuracy of 69.8%, with a recall and F1-score of 0.68, and an AUC of 0.83. While no single variable served as a dominant predictor, the results show that health problems coded in SNOMED CT, combined with other clinical and demographic data, can significantly improve the model's ability to classify discharge diagnoses.

Keywords: Machine Learning, SNOMED CT, ICD-10-CM, Supervised Learning, Multiclass Classification.

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to my tutor, Santiago Frid, for his guidance, support, and teaching throughout the entirety of this project. I am also deeply thankful to Juan Barrios for his constant willingness to share his knowledge and offer his assistance whenever it was needed.

Finally, I would like to thank my family and friends for their unconditional support, which has been fundamental in helping me complete this project. I am deeply grateful for the opportunity this experience has given me to grow both professionally and personally.

LIST OF FIGURES

LIST OF TABLES

GLOSSARY OF ABBREVIATIONS

AI	Artificial Intelligence
ANOVA	Analysis of Variance
AUC	Area Under the Curve
CAGR	Computed Annual Growth Rate
CDSS	Clinical Decision Support Systems
CEIm	Comité de Ética de la Investigación con Medicamentos
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
DRG	Diagnosis-Related Group
DTs	Decision Trees
EDA	Exploratory Data Analysis
EHR	Electronic Health Record
FN	False Negative
FP	False Positive
ICD-10	International Classification of Diseases, 10th Revision
ICD-10-CM	International Classification of Diseases, 10th Revision, Clinical Modification
KNN	K-Nearest Neighbours
LightGBM	Light Gradient Boosting Machine
LLM	Large Language Model
LR	Logistic Regression
MCC	Matthew’s Correlation Coefficient
MDC	Major Diagnostic Category
MDR	Medical Device Regulation
ML	Machine Learning
MLP	Multilayer Perceptron
NaN	Not a Number
NLP	Natural Language Processing

NN	Neural Network
PERT	Program Evaluation and Review Technique
POMR	Problem-Oriented Medical Record
POR	Problem-Oriented Record
RBF	Radial Basis Function
RF	Random Forest
ROC	Receiver Operating Characteristic
SMOTE	Synthetic Minority Over-sampling Technique
SNOMED CT	Systematized Nomenclature of Medicine Clinical Terms
SOAP	Subjective-Objective-Assessment-Plan
SVM	Support Vector Machine
SWOT	Strengths, Weaknesses, Opportunities, and Threats
TN	True Negative
TP	True Positive
WBS	Work Breakdown Structure
WHO	Word Health Organization
XAI	Explainable AI
XGBoost	Extreme Gradient Boosting

TABLE OF CONTENTS

1 INTRODUCTION

In the healthcare sector, accurate diagnoses and treatments are crucial for improving patient outcomes and optimizing hospital resources. This is because diagnostic accuracy is essential to ensure patients receive timely care and minimize the likelihood of medical errors, which can have a significant impact on health outcomes. However, healthcare systems are facing significant challenges due to the increasing complexity of diseases, the volume of data generated, and the need for quick evidence-based decision-making.

In this context, the use of Machine Learning (ML) to predict discharge diagnoses presents itself as a promising tool to improve both the accuracy and efficiency of the diagnostic process. By integrating advanced computational methods, hospitals may be able to reduce diagnostic errors, accelerate treatment plans, and better manage their resources.

1.1 Motivation

Traditionally, hospitals have relied on manual processes to code medical diagnoses and procedures, typically only assigning codes based on discharge reports. This practice presents significant limitations, the most notable one is the delay in assigning these codes, which can often take up to a month after discharge. Such delays leads to inefficiencies, especially in settings where quick decision-making and resource allocation is critical. Moreover, the lack of early coding limits the ability to adjust patient care plans in real time.

Recently, at the Hospital Clínic de Barcelona, a list of health problems coded by physicians using the Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT) has been integrated into the Electronic Health Record (EHR) system from the beginning of the care process. This innovation enables the assignment of standard codes from the very beginning of the patient care process, eliminating the need to wait until discharge. This development allows for the creation of a catalog of clinical entities that can be processed by information systems, thereby providing precise semantic meanings. These advances present a significant opportunity for the reuse of clinical information for both primary and secondary purposes. A particularly intriguing and novel aspect of this initiative is the exploration of whether health problems coded with SNOMED CT at the time of hospitalization can help predict discharge diagnoses, coded with the International Classification of Diseases, 10th Revision, Clinical Modification (ICD-10-CM), through the application of ML methodologies.

The implementation of ML algorithms to these coded health problems could uncover complex patterns and relationships among multiple clinical and demographic variables that are not easily observable to clinicians. Consequently, this approach may enhance the accuracy of predicting discharge diagnoses, thereby potentially reducing the incidence of misdiagnoses, improving patient outcomes, and facilitating more efficient resource allocation.

1.2 Objectives

The primary objective of this research is to explore the correlation between reported health problems and the final diagnoses issued at the time of hospital discharge. This research aims to determine whether a significant relationship exist between these two variables, thereby improving the understanding of how health problems may influence diagnostic decisions.

Furthermore, the study will compare the ability of various ML models to classify discharge diagnoses coded in ICD-10-CM, using health problems coded with SNOMED CT during hospitalization, along with other clinical data. By training these models with data from the Hospital Clínic de Barcelona, the study will investigate how well SNOMED CT coded health problems serve as predictors of ICD-10-CM discharge diagnoses.

Secondary objectives include evaluating model performance using a wide range of metrics, including accuracy, sensitivity, specificity, and more.

Finally, the study aims to identify the most effective model and the most significant input variables for optimizing classification performance. Special attention will be placed on clinical and demographic variables, including age, sex, vital signs, prescriptions, and laboratory results, to identify which variables have the most significant impact on the predicted diagnoses.

1.3 Scope

The scope of the project is limited to the analysis of clinical data collected at the Hospital Clínic de Barcelona over a specific period. The primary objective is to classify discharge diagnoses using various ML models, with the goal of exploring the potential of these tools to support clinical decision-making and improve the efficiency of hospital workflow. The study does not include the evaluation of post-discharge treatments or interventions.

At the spatial level, the study will be conducted solely on data from the Hospital Clínic, although the techniques developed could be replicated in other hospital settings.

1.4 Methodology

The methodology followed for this project can be divided into four main parts. First, a comprehensive literature review is conducted to understand the background and context in which this project is situated. This includes an overview of the challenges in SNOMED CT to ICD-10-CM mapping, as well as the role of ML in healthcare.

Once the data is obtained, a data pre-processing phase follows, during which the raw data is cleaned, formatted, and prepared for model training. In the next stage, suitable ML models are selected and trained to analyse the pre-processed data. Finally, the results are evaluated and discussed, offering insights into the performance of the different models and their implications for the study.

2 BACKGROUND

2.1 Electronic Health Record

An Electronic Health Record (EHR) is a digital representation of a patient’s medical history, that is continuously updated and managed by healthcare professionals. It includes essential clinical and administrative information pertinent to a patient’s care, including demographics, vital signs, diagnoses, treatment plans, medications, past medical history, allergies, immunizations, radiology reports, and laboratory and test results [1]. The implementation of EHR systems facilitates efficient access to information, potentially enhancing the workflow of clinicians [2].

2.2 Problem-Oriented Medical Record

A problem-oriented approach is one of the possibilities to organize a medical record. In the 1960s, Dr. Lawrence Weed introduced the Problem-Oriented Medical Record (POMR), also known as the Problem-Oriented Record (POR) [3], [4]. This structured method revolutionized clinical documentation by emphasizing the identification and management of individual health issues, allowing for more systematic and organized care.

A health problem is defined as any condition affecting a person’s physical, psychological, or social well-being that requires medical attention or may impact the patient’s quality of life [5]. Dr. Weed described a health problem as “anything that requires diagnosis, further management, or interferes with quality of life, perceived by the patient.” [3].

The fundamental component of the POMR is the problem list which can be defined as a dynamic, continually updated record that includes all past and present identified problems, as well as the time of occurrence and whether the problem was resolved, and links to further information on each entry in the list [6]. This structure ensures that all observations, assessments, and healthcare plans are grouped by patient problem, promoting clarity and continuity in patient care.

To further enhance data organization and communication, progress notes are often written in the Subjective-Objective-Assessment-Plan (SOAP) format [7]:

- **Subjective**: the patient’s reported symptoms and concerns.
- **Objective**: observable and measurable clinical findings.
- **Assessment**: clinician’s evaluation or diagnosis.
- **Plan**: recommended next steps in care or treatment plans.

2.3 Clinical coding systems

Medical coding is a key process in healthcare administration, as it allows for the classification and organization of patient clinical information using standardized systems. This structured data facilitates effective communication, analysis, and reporting across healthcare systems.

Two of the most widely used coding systems worldwide are SNOMED CT and ICD-10. While SNOMED CT is widely used in daily clinical documentation due to its ability to capture specific details about diagnoses, symptoms, and procedures in real time, ICD-10 is especially used at the time of patient discharge and supporting healthcare operation such as billing [8].

2.3.1 SNOMED CT

SNOMED CT is the most comprehensive and multilingual clinical terminology, encompassing over 360.000 concepts [9]. It is a coding system that offers a structured and detailed representation of clinical information, covering a wide range of healthcare elements such as diagnoses, symptoms, procedures, medications, and other concepts relevant to healthcare. This system is widely used in clinical documentation due to its benefits, including [10], [11]:

- **Granularity and specificity**: SNOMED CT offers precise descriptions of clinical concepts, allowing clinicians to document information in a very detailed manner, which improves accuracy in healthcare documentation.
- **Interoperability**: the system is designed in a way that it can be integrated with other healthcare systems and EHRs, facilitating the exchange of standardized information between different care providers and improving communication and continuity of care.
- **Data analytics**: by offering structured and computable health data, SNOMED CT supports advanced data analysis and clinical research.
- **Continuous evolution**: SNOMED CT is regularly maintained and updated to include new clinical terms and concepts. This ensures that the terminology remains current and aligned with ongoing advances in healthcare.

2.3.2 ICD-10

ICD-10 is a coding system developed by the Word Health Organization (WHO) that organizes health data into standardized categories for a wide range of clinical, administrative, and research purposes. It assigns unique alphanumeric codes to various health-related terms, including diseases, signs and symptoms, procedures, and abnormal findings. This system facilitates the classification of health information across healthcare systems and countries [12].

ICD-10 also supports the storage, retrieval, and analysis of diagnostic information which is crucial for epidemiological studies, healthcare research, and monitoring of population health [13]. It also standardizes the recording and reporting of health data, which is essential for statistical analysis, as well as for billing, reimbursement, and resource allocation within healthcare systems [14].

Some key advantages of this system include:

- **Hierarchical structure**: ICD-10 organizes diseases and other health conditions into standardized and structured groups for easier management.
- **Administrative efficiency**: it enhances coding accuracy for billing and reimbursement processes, which reduces administrative workload.
- **Focus on statistics and management**: it provides healthcare administrators with statistical insights to assess the time and resources spent on treating a medical condition.

2.3.2.1. ICD-10-CM

International Classification of Diseases, 10th Revision, Clinical Modification (ICD-10-CM) is a modified version of the ICD-10 coding system, that has been specifically adapted and expanded with more detailed codes for clinical use in the United States [15]. The translated version of ICD-10-CM is used at the Hospital Clínic de Barcelona.

Although ICD-10-CM is essential for health resource management, its focus on disease classification and statistical and administrative purposes can lead to the loss of detailed clinical information that is captured by SNOMED CT. This difference in coding approach poses challenges in using both systems in a complementary manner.

2.3.3 Current challenges in clinical coding

Although clinical coding systems offer numerous benefits, there are still several challenges that make their implementation and use difficult.

2.3.3.1. Delayed coding processes

A significant issue with clinical coding is the delay in assigning codes, particularly for ICD-10-CM. In many hospitals, coding is often performed days or even weeks after a patient's discharge, which limits the utility of coded data for real-time decision-making and resource allocation. This delay also contributes to inefficiencies in healthcare services [16].

2.3.3.2. Manual effort and error rates

Manual code conversion is susceptible to high error rates and inefficiencies. Coders must make decisions when interpreting clinical notes and assigning appropriate codes, a process that is time-consuming and prone to mistakes. These errors in code conversion can lead to significant consequences like incorrect billing, denied insurance claims, and inaccurate statistical data, all of which can negatively impact patient care and hospital finances [16].

2.3.3.3. Limitations of SNOMED CT to ICD-10 mappings

Mapping between SNOMED CT and ICD-10-CM presents significant challenges due to the fundamental differences in their structures and intended use. SNOMED CT offers a much more detailed and granular representation of clinical data, while ICD-10-CM is designed more for population-level, epidemiological and administrative use, often lacking the level of clinical granularity found in SNOMED CT. This discrepancy leads to inconsistencies and difficulties in creating accurate mapping, which can complicate the integration of clinical and administrative data within healthcare systems.

Although mapping tools have been developed to address this issue, it only provides a semi-automated generation of ICD-10-CM classification codes from clinical data encoded in SNOMED CT [17]. Moreover, these mapping are partial and fail to address complex cases like n:n mapping, where one concept may correspond to multiple other concepts, rather than a simple one-to-one mapping.

2.4 Artificial Intelligence

Artificial Intelligence (AI) refers to the development and use of computers systems capable of performing tasks that usually require human intelligence. These tasks include learning, problem-solving, reasoning, perception, and language understanding [18]. In recent years, the availability of high-performance computers and the large amount of data generated have led to advancements in the application of AI across many fields. This progress has also significantly accelerated the development of Machine Learning and Deep Learning, subfields of AI that enable systems to learn from data and continuously improve their performance over time without explicit programming (Figure 1).

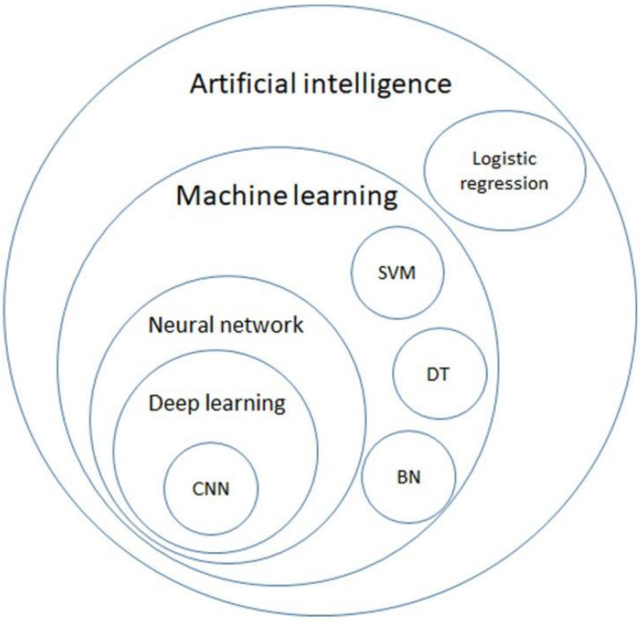


Figure 1: *Venn diagram of artificial intelligence (AI), machine learning (ML), neural network, deep learning, and further algorithms in each category* [19].

2.5 Machine Learning

Machine Learning (ML) is a subset of AI that focuses on the development of algorithms and statistical models that enable computers to perform tasks without explicit instructions. Instead, ML models learn by identifying patterns, extracting meaningful insights, and continuously improving their performance over time through experience [20].

ML is typically divided into several categories, with the two most prominent being:

- **Supervised learning**: models are trained on labelled data, meaning that each input is paired with a known output. This allows the algorithm to evaluate its performance and make adjustments during training to improve accuracy [20].
- **Unsupervised learning**: models are trained on unlabelled data, allowing the algorithm to identify hidden patterns, structures, or relationships within the data without prior knowledge of the outcomes [20].

2.5.1 Machine Learning applied to healthcare

ML applications in healthcare are diverse an range from disease prediction to treatment optimization. Some relevant applications include [21], [22]:

- **Disease prediction**: ML models can analyse historical patient data to identify risk factors and predict the likelihood of developing certain health conditions.
- **Medical imaging**: ML algorithms are capable of interpreting medical images, such as X-rays, CT scans, and MRIs, achieving accuracy levels comparable to that of radiologists.
- **Clinical decision support**: ML is widely used in Clinical Decision Support Systems (CDSS) to assist healthcare professionals in predicting patient outcomes and recommending treatments.
- **Workflow optimization**: ML can assist hospitals in managing resources more efficiently and optimizing administrative processes.
- **Readmission risk prediction**: ML models have been used to predict the likelihood of patient readmission, demonstrating the potential for improving resource allocation and reducing hospital costs [23].
- **Personalized medicine**: ML has significantly advanced personalized medicine by analysing individual patient data such as genetic information, medical histories, and lifestyle factors, to tailor treatments for each patient [24], [25].

2.6 State of the art

The use of ML in healthcare has experienced a rapid growth in recent years thanks to the advancements in computational power and the increased availability of data. Recent studies have shown that ML models trained on high-dimensional data, especially when supplemented with Natural Language Processing (NLP) techniques to extract insights from unstructured text, can significantly improve diagnostic predictions [26]. Additionally, developments in Explainable Artificial Intelligence (XAI) have made it easier for healthcare professionals to understand model predictions and interpret the results [26], [27].

The use of large and diverse datasets, such as national health databases and specific hospital data, has played a key role in improving the generalizability of predictive models. Many studies have employed ensemble methods to combine predictions from multiple models, improving robustness and minimizing bias [28]. Furthermore, research into transfer learning and federated learning has created new opportunities to share data across different institutions while maintaining patient privacy [29]. However, challenges still remain in the application of ML in healthcare, including concerns about data quality, the ethical use of patient information, and the need for model validation in real-world clinical environments [30].

As this field continues to evolve, future research is expected to focus on refining existing models, creating hybrid approaches that combine domain expertise with data-driven insights, and ensuring safe and ethical integration of these technologies into clinical practice. Accurately predicting hospital discharge diagnoses not only has the potential to improve patient care but also offers benefits in resource management, reducing readmissions, and enhancing clinical workflows.

2.6.1 Machine Learning in Discharge Diagnosis Prediction

ML has shown strong potential in predicting hospital discharge diagnoses by uncovering complex patterns in clinical and demographic data. A notable study conducted by Lin et al. (2017) [31] explores the application of AI in automating the classification of diagnosis coded from unstructured discharge notes. The goal of the study was to evaluate the performance of traditional pipelines (NLP paired with supervised ML models) with that of word embedding combined with a Convolutional Neural Network (CNN) (Figure 2) in performing a classification task to identify ICD-10-CM diagnosis codes in discharge notes.

The results revealed that in 5-fold cross-validation test, the word embedding combined with a CNN had higher testing accuracy (mean AUC 0.9696; mean F-measure 0.9086) than traditional NLP-based approaches (mean AUC range 0.8183 - 0.9571; mean F-measure range 0.5050 - 0.8739). Additionally, it showed that the convolutional layers of the CNN successfully identified a significant number of keywords and automatically extracted enough concepts to predict the diagnosis codes. The research demonstrated its ability to effectively extract and predict diagnosis codes with minimal data pre-processing, highlighting the potential of CNNs to automatically capture essential medical concepts from unstructured text.

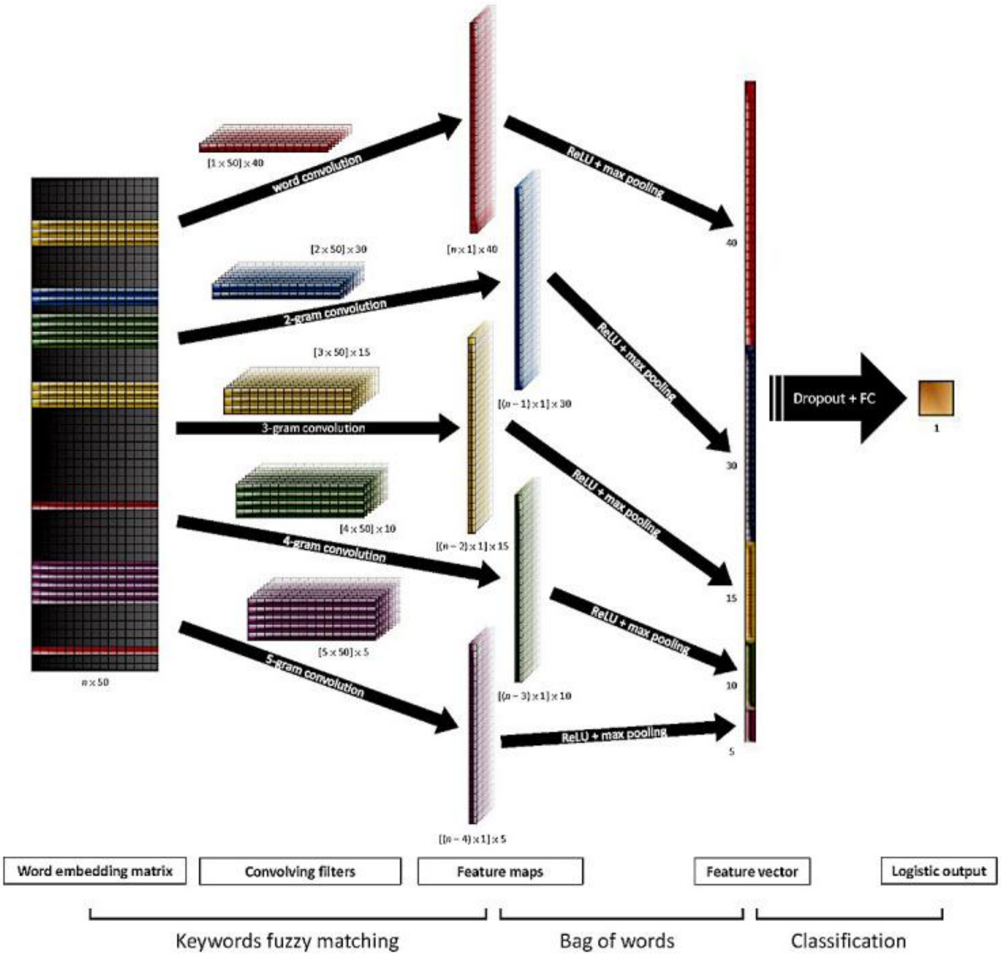
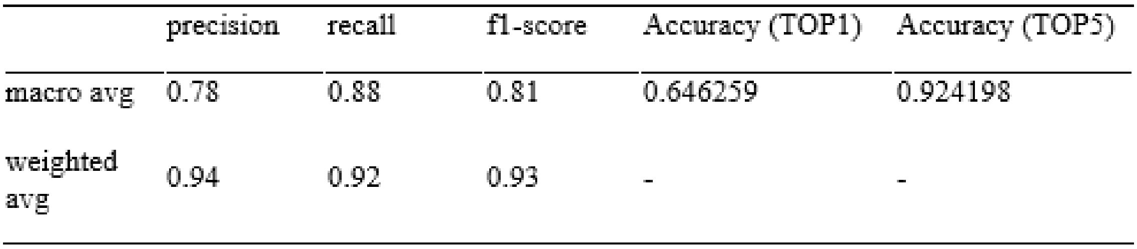


***Figure 2:** Proposed model's architecture by Lin et al. (2017) [31].*

Another interesting study is the one conducted by Park et al. (2021) [32]. This research focuses on creating an optimised ensemble model that combines Deep Neural Networks (DNN) with ML algorithms to predict diseases using laboratory test results. Their objective was to develop a model capable of accurately predict 39 specific diseases based on laboratory test data. To do so, researchers selected 86 laboratory test attributes from datasets, considering factors such as value counts, clinical importance, and missing values. Sample datasets on 5145 cases, including 325686 laboratory test results were collected. These datasets were then used to construct Light Gradient Boosting Machine (LightGBM) and Extreme Gradient Boosting (XGBoost) ML models and a DNN model. What they found was that the optimised ensemble model achieved a F1- score of 81% and a prediction accuracy of 92% for the five most common diseases (Table 1).

Table 1: *Ensemble model performance result using F1-score by Park et al. (2021)* [32].



	precision	recall	f1-score	Accuracy (TOP1)	Accuracy (TOP5)
macro avg	0.78	0.88	0.81	0.646259	0.924198
weighted avg	0.94	0.92	0.93	-	-

3 MARKET ANALYSIS

3.1 Market sector

The healthcare sector is one of the largest and most dynamic industries globally, with significant contributions to economic growth and societal well-being. In 2024, the global healthcare market was valued at 112.9 billion USD and is projected to reach 139.69 billion by 2033, exhibiting a Computed Annual Growth Rate (CAGR) of 2.4% (Figure 3) [33]. This growth is driven by several factors, including the increasing prevalence of chronic diseases, aging populations, and advancements in medical technologies.

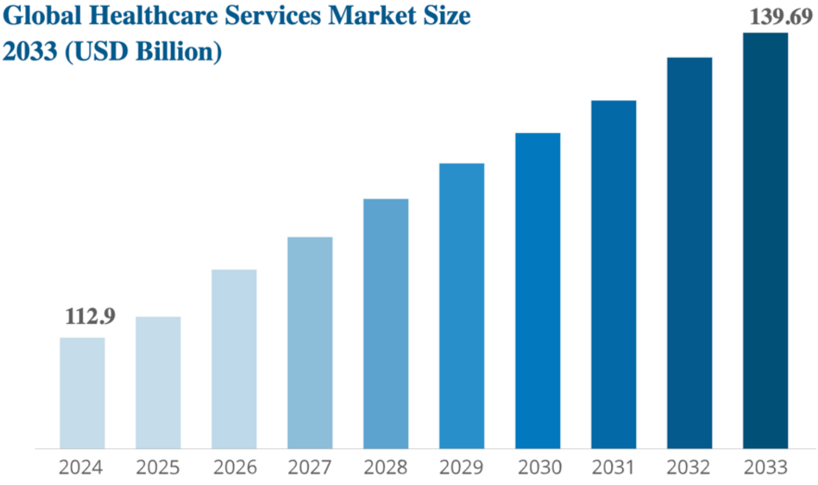


Figure 3: *Global healthcare services market size estimation for 2033 [33].*

3.2 Target market

The target market for a ML algorithm capable of predicting hospital discharge diagnoses using SNOMED CT encoded health problems reaches various sectors within healthcare and medical technology.

Primary markets include hospitals, where there is a growing demand for innovative solutions to improve real-time decision-making and optimise resource allocation, particularly in setting where timely and accurate diagnoses are critical. The integration of SNOMED CT encoded health problems with ML methodologies offers a valuable opportunity to predict discharge diagnoses early in the care process. This enables healthcare providers to adjust treatment plans, ultimately improving overall patient outcomes.

Secondary markets include insurance companies that are seeking to predict patients risks, manage claims more effectively, and reduce costs associated with misdiagnoses or prolonged hospital stays. By predicting discharge diagnoses accurately, insurers can improve their claims processes, resulting in cost savings and improved efficiency.

3.3 Future perspectives

In recent years, the growth of AI applications in healthcare has been remarkable. AI-driven innovations are being widely applied, with significant advances expected in areas such as medical imaging, drug development, disease classification and diagnostics, predictive analytics, and personalized medicine, including treatment and prescription [34].

Key trends and emerging opportunities of clinical coding and predictive analytics in healthcare in the future include [35]:

- **Personalized medicine**: predictive analytics allows for the customization of treatments based on individual patient data, enhancing the effectiveness and efficiency of care by tailoring interventions to patient needs.
- **AI and ML in clinical coding**: the use of AI and ML in clinical coding is expected to grow significantly, driven by the increasing need for real-time coding and the need to minimize errors.
- **Natural Language Processing (NLP)**: NLP technologies play a crucial role in extracting structured data from unstructured clinical notes, improving the accuracy of coding, and enabling better integration with EHR systems.
- **Real-time clinical decision support**: the integration of predictive analytics with real-time clinical decision support systems allow clinicians to receive instant recommendations during patient care, helping reduce delays and improve patient outcomes.

4 CONCEPT ENGINEERING

To reach the objectives of the research, different stages must be completed. The overall workflow of the project (Figure 4) outlines the key steps where different methodologies can be applied. This section evaluates the different proposed methods and presents the selected solution.

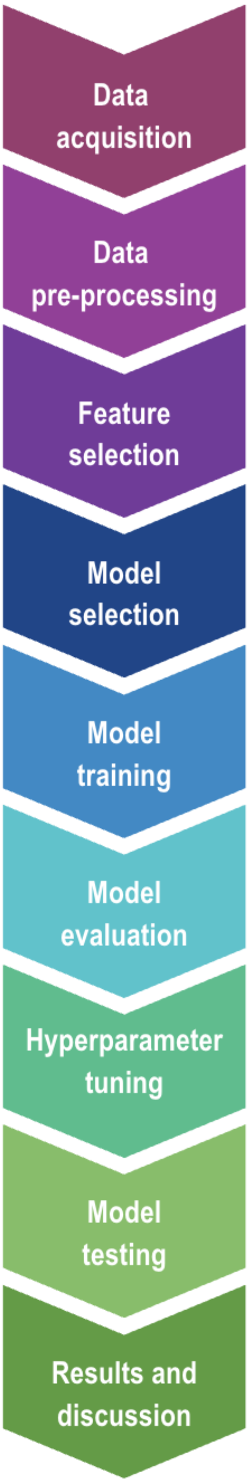


Figure 4: *Overall workflow of the project.*

4.1 Data acquisition and description

The data used in this study was obtained from the Hospital Clínic de Barcelona. All information originates from the hospital's institutional data warehouse, which serves as a centralized repository for clinical data.

Access to the data was granted with the approval of the Comité de Ética de la Investigación con Medicamentos (CEIm) of the Hospital Clínic de Barcelona. A copy of the ethics approval document is provided in Annex A.

The main sources of information include:

- **Administration events**: information related to the administration of treatments to patients. It keeps track of various aspects, such as the drugs administered, the method of administration, and the amounts involved.
- **Admission and discharge events**: contains records related to patient admissions and discharges, providing insight into the patient's entry and exit from healthcare facilities.
- **Care level events**: data related to the care levels assigned to patients throughout their medical episodes.
- **Clinical records events**: contains detailed clinical records and medical results, including test results, and measurements taken during patient episodes.
- **Demographic events**: contains demographic information about the patients, including date of birth, sex, and nationality.
- **Diagnostic events**: contains information about hospital discharge diagnoses and other diagnostic events, providing valuable insights into the medical conditions and diagnoses associated with patient episodes.
- **DRG events**: data related to Diagnosis-Related Groups (DRG), a system used to classify hospitalized patients into categories that have similar processes of care and require similar levels of hospital resources. DRGs are intended to identify the "products" that the hospital provides and are mainly used for billing and reimbursement purposes [36].
- **Encounter events**: records detailed information about patient encounters within the healthcare system.
- **Episode events**: contains important information about the start and end dates of patient episodes, which represents a continuous period of care or treatment for a patient within the healthcare system.
- **Exitus events**: captures critical information regarding patient deaths.
- **Health issues events**: contains information about the health problems of the patients.
- **Laboratory events**: contains information regarding laboratory test results and associated details like the different laboratory test performed.
- **Movement events**: tracks patient transfers between different locations or care units within the healthcare system.
- **Perfusion events**: records information regarding drug infusion treatments administered to patients during their episode of care. Infusion treatments involve the slow administration of fluids or drugs, typically via an intravenous (IV) line.

- **Prescription events**: contains information regarding patient prescriptions, such as the drugs prescribed and the dosage.

4.2 Data pre-processing

Data pre-processing is a crucial step in the data engineering process. Given that real-world datasets often contain inconsistencies, missing values, and other imperfections, pre-processing ensures that the data is clean, consistent, and appropriately formatted for the next steps.

4.2.1 Missing values

Missing data is a common issue in clinical datasets and must be carefully handled to maintain data quality and avoid introducing bias. Datasets often contain missing values, which are typically represented as blanks or NaN (Not a Number). Most ML algorithms cannot handle missing or blank values, making it necessary to apply appropriate strategies for dealing with them.

Some common approaches to handling missing data are [37]:

- **Dropping rows**: one straightforward method is to remove rows with missing values. This approach is useful when the dataset is large enough that removing records will not significantly impact the overall analysis. However, this method can result in the loss of valuable data and potentially removing key patterns or relationships from the dataset.
- **Imputing missing values**: another strategy is to impute, or fill in, the missing values with logical substitutes. There are several imputation techniques:
    - Mean: replaces missing values with the mean of the respective column. It is suitable for normally distributed data.
    - Median: fills in missing values with the median value of the column. It is often used when the data contains outliers.
    - Mode: replaces missing values with the most frequent value in the column. It is often used for categorical features or variables with repeated values.
    - K-Nearest Neighbours (KNN): imputes missing values based on the values of the nearest neighbours. It identifies the *k* most similar rows and fills the missing value with the average of the corresponding values from those rows.

4.2.2 Encoding categorical variables

Ensuring that data types are correctly assigned is crucial, as various downstream processes depend on the data type of features. Categorical features, in particular, contain label values rather than numeric values. Since most ML algorithms cannot directly handle categorical data, it must be transformed into numeric values before training a model. The most commonly used methods for encoding categorical data are [38]:

- **Label encoding**: each category is assigned a unique integer value. This method is best suited for nominal data where the order doesn’t matter as it does not respect the order of the categories.

- **Ordinal encoding**: this method is used when categories have a clear, defined order but not necessarily evenly spaced intervals. For example, “Low”, “Medium”, and “High” would be assigned numerical values reflecting their rank.
- **One-Hot encoding**: in this method, each category in a feature is transformed into a separate binary feature (1 or 0). This approach is ideal for nominal data as it prevents the model from assuming any relationship between the categories.

4.2.3 Normalization

Normalization is a technique aimed at rescaling the values of numeric features so they fall within a consistent range. This ensures that no single feature dominates due to its scale and help models train more effectively and efficiently. Many ML models perform better when the input features are within similar value ranges or distributions.

The most frequently used methods for scaling in ML are [39]:

- **Min-Max normalization**: preserves the original distribution of data but scales values to a fixed range between 0 and 1. Each value is transformed according to Eq. 1:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

- **Z-score normalization**: transforms the data so it has a mean (μ) of 0 and a standard deviation (σ) of 1. The formula is shown in Eq. 2:

$$x_{standard} = \frac{x - \mu}{\sigma} \tag{2}$$

4.3 Feature selection

Feature selection is a crucial step in the ML pipeline. It is a process aimed at identifying the features in the dataset that contribute the most in predicting the target variable. Focusing on these selected features instead of all the features, not only helps reduce the risk of overfitting but also enhances model performance and improves computational efficiency by reducing training time.

There are many methods for feature selection, some methods to consider are:

- **Tree-Based models**: algorithms such as random forest and decision trees can be used as tools to estimate feature importance. When building a decision tree, the algorithm evaluates features at different nodes to determine the best splits. A feature’s importance is based on how much it reduces impurity, which is a measure of how mixed the target classes are after the split. Features that consistently reduce impurity are deemed more important [40].
- **Analysis of Variance (ANOVA) test**: it is a statistical test used to compare the means of two or more groups and determine whether they are significantly different. In the context of feature selection, the ANOVA F-test evaluates each feature individually by calculating a

F-score, which represent the ratio of the variance between groups to the variance within groups. Features with higher F-scores are considered more relevant to the target variable [41].

- **Pearson correlation**: it measures the strength and direction of the linear relationship between two continuous variables. Features that have a strong correlation (positive or negative) with the target variable can be considered for selection. However, care must be taken to handle multicollinearity, where multiple features are highly correlated with each other, potentially leading to redundancy [42].

4.4 Supervised Machine Learning models

Supervised machine learning is a subfield of ML where the model is trained on a labelled dataset, meaning each training example is paired with a correct output. The objective of supervised machine learning is to learn a mapping from inputs to outputs that can generalize well to unseen data. This learning paradigm is commonly used for classification and regression tasks, where the model aims to predict a category or a continuous value, respectively.

In supervised learning, the training process involves minimizing a loss function that measures the discrepancy between the predicted output and the actual label. Once trained, the model can be evaluated on test data to assess its performance using metrics such as accuracy, precision, recall, and F1 score, depending on the task.

In the following subsections, a brief overview of the theorical foundations of the different supervised machine learning models considered in this project is provided.

4.4.1 Logistic Regression

Logistic Regression (LR) is a fundamental statistical model commonly used for binary classification tasks. While it is derived from regression analysis, it is designed to predict discrete outcomes, particularly binary or categorical responses. Unlike linear regression, which estimates continuous values, LR calculates the probability that a given input belongs to a particular class [43]. Although LR is inherently a binary classifier, it can be extended to handle multiclass classification [44]. One of the major advantages of this model is its interpretability, which makes it particularly useful in clinical settings, where understanding the impact of different features is essential.

4.4.2 Support Vector Machine

Support Vector Machine (SVM) is a supervised learning model primary used for classification tasks. It works by finding an optimal hyperplane that separates different classes in the data and selecting the one that maximizes the margin between classes. The data points that are closest to this hyperplane and define its position are known as support vectors [45].

Although SVMs are inherently linear classifiers, they can effectively handle non-linear relationships in the data by using kernel functions. Commonly used kernels include the Radial Basis Function (RBF), polynomial, and sigmoid kernels [46]. These functions project the input features into a higher-dimensional space where a linear separation becomes possible, allowing the model to learn complex patterns like the ones found in clinical datasets. While SVMs may be less interpretable than simpler models such as LR, they are useful in scenarios where intricate and potentially non-linear relationships exist between features and diagnostic outcomes.

4.4.3 Decision Trees

Decision Trees (DTs) are a fundamental machine learning algorithm used for classification and regression tasks. Recognized for their intuitive design and ease of interpretation, DTs simulate human decision-making by iteratively dividing data into subsets based on feature values. This tree-like structure makes decision rules easy to visualize, contributing to their widespread use across different fields [47].

DTs are composed of nodes, each internal node represents a decision based on a feature, each branch corresponds to a possible outcome of the decision, and each leaf node assigns a class label. The model begins at the root node and splits the dataset by selecting the feature that best separates the data according to a chosen criterion, such as Gini impurity, entropy, or log loss. This process continues recursively until a stopping condition is met, such as no remaining features, all data points at a node belonging to the same class, or a predefined maximum depth [48].

Despite its simplicity, DTs serve as the foundation for more advanced ensemble models like Random Forests and Gradient Boosted Trees, which combine multiple trees to improve predictive performance and generalization.

4.4.4 Random Forest

Random Forest (RF) is a robust and flexible ensemble learning algorithm commonly used for classification and regression tasks. Developed by Leo Breiman in 2001, it enhances the decision tree approach by combining multiple trees to generate more reliable, precise, and generalized predictions [49]. The algorithm utilizes bagging (bootstrap aggregating), where each tree is trained on a random subset of the training data. Additionally, it introduces extra randomness in feature selection to minimize overfitting and enhance model performance. In classification tasks, the ensemble's final prediction is obtained by aggregating the predictions of the individual trees, typically through majority voting as shown in Figure 5 [50].

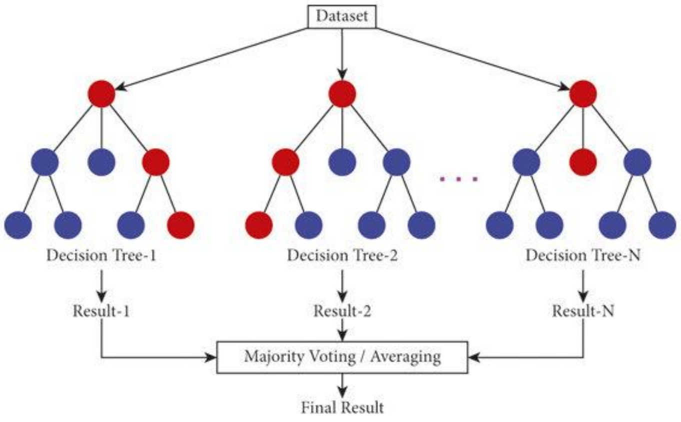


***Figure 5:** Random Forest trees illustration [50].*

4.4.5 Gradient Boosting

Gradient Boosting is an ensemble learning technique that combines multiple weak learners, usually decision trees, to form a more powerful predictive model. The core principle of gradient boosting is to train models sequentially, with each new model focusing on correcting the errors made by the previous ones. This is achieved by training each new model to fit the residuals, or errors, left by the preceding model. In each iteration, a new tree is trained using the negative gradient of the loss function concerning the current predictions, progressively minimizing the error [51].

4.4.5.1. Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is an optimised and scalable implementation of gradient boosting developed to enhance speed, efficiency, and performance. Unlike RF, which build multiple decision trees independently and aggregate their outputs, XGBoost builds trees sequentially. In this process, each new tree is trained to correct the errors made by the previous ensemble of trees by assigning higher weights to misclassified samples as shown in Figure 6 [52]. This focused learning process allows the model to capture complex data patterns and improve predictive performance over time.

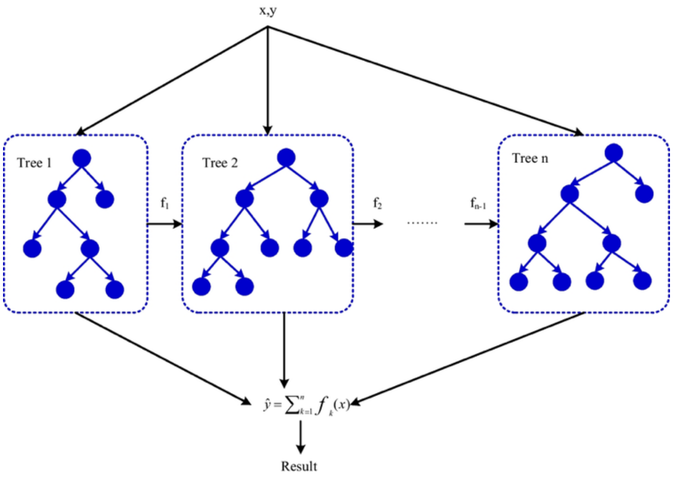


Figure 6: *Extreme Gradient Boosting (XGBoost) illustration [53].*

4.4.6 K-Nearest Neighbours

K-Nearest Neighbours (KNN) is a simple, instance-based supervised learning model used for classification and regression tasks. In classification, KNN predicts the class of a new data point by identifying the number of neighbours (*k*) closest samples in the training set, based on a distance metric such as Euclidean distance. The class most frequently represented among these neighbours is then assigned to the new point [54].

The choice of *k* is a critical hyperparameter: a small *k* may lead to overfitting and noisy predictions, while a large *k* tends to smooth out class boundaries but can cause underfitting [54]. While KNN is easy to implement and understand, it struggles with large datasets unless properly optimised, and its performance heavily depends on the chosen distance metric and how the data features are scaled.

4.4.7 Neural Networks

Deep Learning (DL) is a subset of ML, that uses neural networks composed of multiple layers to process and analyse large volumes of data. These layered networks are built to identify patterns and generate predictions automatically [55]. As a result, DL has proven to be particularly effective in healthcare due to its ability of handling and managing large and complex datasets.

In the 1950s, the perceptron algorithm was first introduced as one of the firsts attempts to replicate how a human neuron works (Figure 7). The perceptron processes an input, applies weights, and then uses an activation function to determine if the neuron becomes active and generates an output. Although a single perceptron cannot recognize complex patterns, combining multiple perceptrons into layered structures, known as Neural Networks (NN), allows the model to capture and learn much more complex data [56].

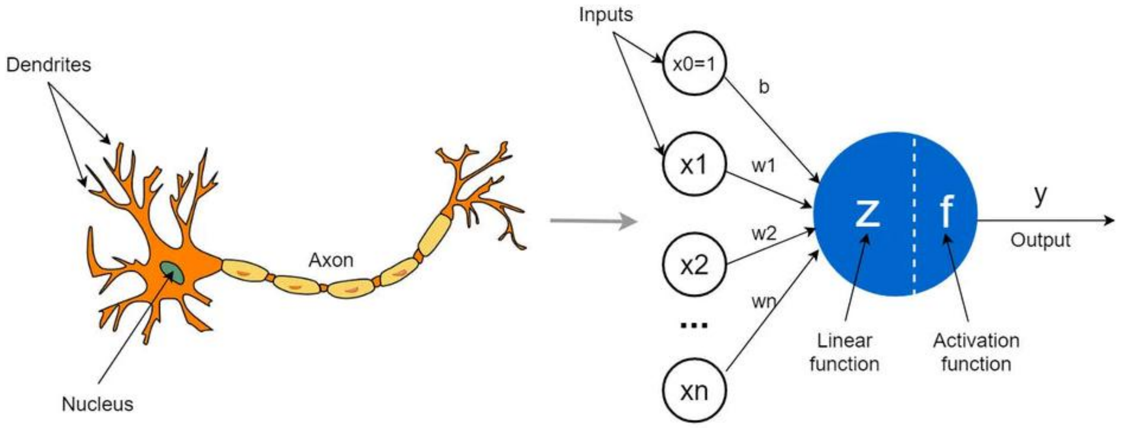


***Figure 7:** Diagram of a neuron model [57].*

4.4.7.1. Multilayer Perceptron

The Multilayer Perceptron (MLP) is a type of NN and one of the most widely used architectures in deep learning. A MLP consists of at least three layers: an input layer, one or more hidden layers, and an output layer. Each layer, except for the input, is made up of neurons that apply a non-linear activation function, allowing the network to learn complex mappings between inputs and outputs. The network trains by adjusting the weights of these connections through a process called backpropagation, which minimises a loss function typically using gradient descent. For classification tasks, the output layer usually contains one node per class [58].

MLPs are highly flexible and can model intricate patterns in clinical datasets. However, they also come with several challenges. They require large amounts of labelled data to perform well and are not easily interpretable which can be a disadvantage in clinical settings, where explainability is important.

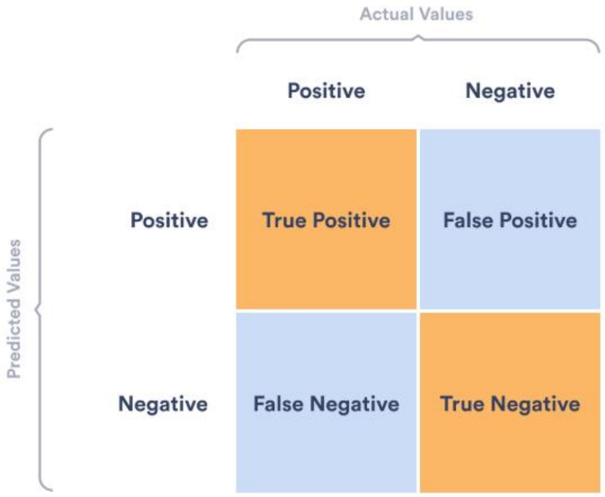
4.5 Model evaluation

This section describes some of the most widely used metrics to assess the performance of classification models. These metrics are essential for comparing the effectiveness of different models and for understanding how well a model generalizes to unseen data.

Confusion Matrix (Figure 8): is a fundamental tool for evaluating the performance of classification models. It provides a summary of the model's predictions compared to the real values of the classes. For binary classification, the confusion matrix consists of four components [59]:

- **True Positive TP)**: instances where the model correctly predicts the positive class.
- **True Negative (TN)**: instances where the model correctly predicts the negative class.
- **False Positive (FP)**: instances where the model predicts a positive class, when the actual class is negative.
- **False Negative (FN)**: instances where the model predicts a negative class, when the actual class is positive.

For multiclass classification, the confusion matrix is extended to an *n x n* matrix, where *n* is the number of classes. Each row of the matrix represents the actual class, while each column represents the predicted class. Diagonal elements indicate correct predictions, while off-diagonal elements correspond to misclassifications [59].



	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

***Figure 8:** Confusion matrix [60].*

The confusion matrix not only helps identifying the types of errors made by the model but also serves as the foundation for deriving several other evaluation metrics [61]:

Accuracy (Acc): represents the proportion of correctly classified instances out of the total number of instances. Accuracy can be calculated using the following formula (Eq. 3):

$$Acc = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{3}$$

Precision (P_n): refers to the ratio of correctly predicted positive instances to the total number of instances that was predicted as positive (Eq. 4).

$$P_n = \frac{T_P}{T_P + F_P} \tag{4}$$

Recall (R_c): indicates the proportion of actual positive instances that were correctly identified by the model (Eq. 5).

$$R_c = \frac{T_P}{T_N + F_P} \tag{5}$$

Sensitivity (S_n): represents the model's ability to correctly identify positive cases (Eq. 6).

$$S_n = \frac{T_P}{T_P + F_N} \tag{6}$$

Specificity (S_p): measures the proportion of actual negative instances that are correctly identified by the model (Eq. 7):

$$S_p = \frac{T_N}{T_N + F_P} \tag{7}$$

F-measure: the F1 score is the harmonic mean of precision and recall, providing a balanced measure of a model's accuracy identifying positive cases. The highest F score is 1, which indicates perfect precision and recall score (Eq. 8).

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

Area Under the Curve (AUC): quantifies the overall ability of a model to distinguish between classes across various threshold settings (Eq. 9). Where I_p and I_n represent positive and negative data samples, and R_i represents the rating of the ith positive samples [61].

$$AUC = \frac{\sum R_i(I_p) - I_p\left(\frac{I_p + 1}{2}\right)}{I_p + I_n} \tag{9}$$

Cohen's kappa ($\boldsymbol{\kappa}$): is frequently used to test interrater reliability. It is a metric that measures the agreement between two raters or classification models, taking into account the agreement that could happened by chance (Eq. 10). Where $\Pr(a)$ is the observed proportion of agreement, and $\Pr(e)$ is the expected proportion of agreement by chance [62].

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \tag{10}$$

Matthew's correlation coefficient (MCC): unlike accuracy, it provides a balanced measure even if the classes are of very different sizes, making it especially useful for imbalanced datasets (Eq. 11) [63].

$$MCC = \frac{T_P \times T_N - F_P \times F_N}{\sqrt{(T_P + F_P)(T_P + F_N)(T_N + F_P)(T_N + F_{N)}}} \tag{11}$$

4.5.1 Validation

To evaluate the generalizability and robustness of the model, appropriate validation strategies must be employed. These strategies help reduce overfitting and provide a more accurate estimate of model performance. Below are commonly used validation techniques:

- **Hold-Out validation**: this strategy involves randomly dividing the dataset into two sets: a training set and a test set. The model is trained on the training set and evaluated on the test set. This method is straightforward and computationally efficient, making it suitable for large datasets. However, its performance estimate can be sensitive to the specific data split, potentially leading to high variance in model evaluation. This sensitivity can result in misleading performance metrics, especially when the dataset is small or imbalanced [64].
- **K-Fold Cross-validation**: this method addresses the limitations of hold-out validation by dividing the dataset into *k* equal-sized folds. The model undergoes *k* iterations, each time training on *k-1* folds and testing on the remaining fold. This process ensures that every data point is used for both training and testing, providing a more reliable estimate of model performance [65].
- **Stratified K-Fold Cross-validation**: it is an enhancement of k-fold cross-validation that ensures each fold maintains the same proportion of each class as the entire dataset (Figure 9). This technique is particularly beneficial for imbalanced datasets, where certain classes may be underrepresented. By preserving class distribution, this method provides a more accurate assessment of model performance across all classes [65].

For this project, the stratified k-fold cross validation method is used to ensure each class is adequately represented in both training and validation phases. This approach enhances the reliability of performance metrics and the development of models that generalize well.

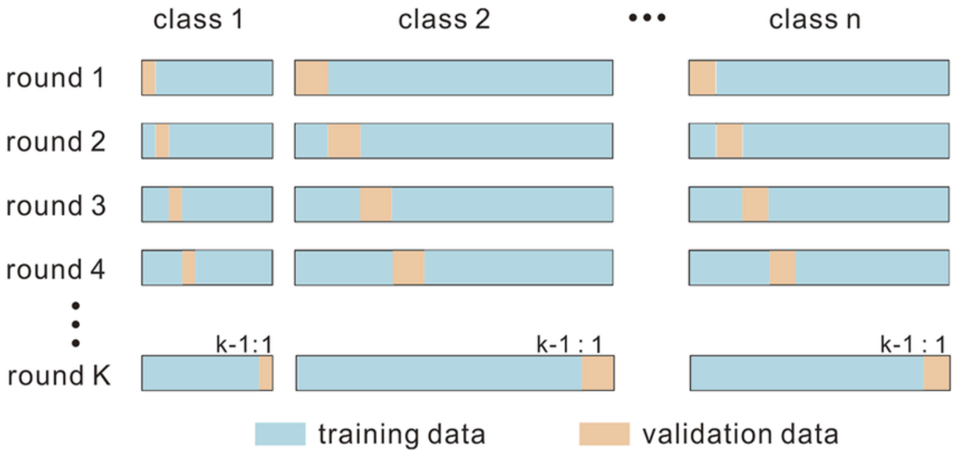


***Figure 9:** Schematic diagram of Stratified K-Fold Cross-validation [66].*

4.6 Hyperparameter tuning

Unlike model parameters, which are learned directly from the training data, hyperparameters are defined externally and play a significant role in model performance. Effective hyperparameter tuning is a crucial step in developing a robust ML model. Choosing appropriate hyperparameter values can lead to improvements in model performance, generalization, and computational efficiency.

Common strategies used for hyperparameter tuning are [67]:

- **Grid search**: this method involves an exhaustive search through a predefined set of hyperparameter values. While it is simple to implement, it can be computationally expensive, especially when dealing with many hyperparameters or large datasets.
- **Random search**: instead of evaluating all possible combinations, this approach explores random combinations of hyperparameters. This method is ideal when computational resources are limited.
- **Bayesian optimization**: this approach uses probabilistic models to estimate the performance of hyperparameter combinations and then selects the most promising options to evaluate. It is more efficient than grid or random search but it is more complex to implement.

5 DETAIL ENGINEERING

The following section provides a detailed overview of each stage of the project execution. It includes a structured explanation of the methods applied at each step, the final results obtained, and a discussion of the outcomes.

5.1 Programming environment

All coding for the project was carried out using the Python programming language because of its versatility, ease of use, and extensive ecosystem of data science libraries. Python is widely used in data science and ML due to its numerous open-source libraries that streamline the development of ML models.

Pandas and NumPy were used for data manipulation and numerical operations. For data visualization, Matplotlib and Seaborn were employed to generate plots and charts that supported Exploratory Data Analysis, feature selection, and model evaluation.

For the implementation of the various supervised ML models, the PyCaret library was used. PyCaret is a low-code ML library with an easy-to-use interface that simplifies and automates various ML workflows, facilitating efficient model development and experimentation [68]. Additionally, Scikit-learn was employed because of its wide array of tools that support algorithm implementation, model evaluation, and other essential ML tasks.

Script development was conducted in Jupyter Notebooks, providing an interactive coding environment for both writing and visualizing code. All project notebooks are available in a GitHub repository.

5.2 Data pre-processing

As described in Section 4.1, the data used in this study was obtained from the Hospital Clínic de Barcelona. The dataset consisted of 15 separate files, each containing different clinical and administrative information. Each file was first imported and then subjected to a series of data pre-processing steps to ensure the dataset was clean, consistent, and suitable for training ML models.

The initial step involved removing duplicate rows across all files to avoid redundancy. Missing values were assessed separately for each file. Given the large size of the datasets and the relatively low proportion of missing data, rows with missing values were removed using the *dropna()* function.

Due to the high number of columns in each file and the limited computational resources, only the most relevant features for diagnosis prediction were kept. Non-informative or redundant columns were dropped to manage dimensionality and focus on clinically meaningful features. Additionally, to enhance consistency and readability, some columns were also renamed across files.

In certain files, additional columns were generated to improve the dataset's predictive capacity. For example, an age column was computed using the patient's date of birth and the date of admission.

Furthermore, episode and care level durations were also calculated using the timestamps provided in the *episode_events.csv* and *care_level_events.csv* files, respectively.

For the *diagnostic_events.csv* file, only diagnoses that were not present on admission (poa = 0), were selected. This filtering was applied to focus on identifying new diagnoses developed during the hospital stay, rather than pre-existing conditions. Additionally, due to the large number of unique ICD-10-CM codes, it was necessary to group them into broader diagnostic categories to make the classification problem more manageable. Instead of predicting individual ICD-10-CM codes, diagnoses were grouped based on ICD-10-CM chapters, as shown in Table 2 [69].

Table 2: *ICD-10-CM chapters and corresponding code ranges.*

ICD-10-CM Chapter Name	Range
Certain infections and parasitic diseases	A00 to B99
Tumours (neoplasms)	C00 to D49
Diseases of the blood and blood-forming organs and disorders affecting the immunological mechanism	D50 to D89
Endocrine, nutritional, and metabolic diseases	E00 to E89
Mental and behavioural disorders	F01 to F99
Diseases of the nervous system	G00 to G99
Diseases of the eye and its appendages	H00 to H59
Diseases of the ear and the mastoid process	H60 to H95
Diseases of the circulatory system	I00 to I99
Diseases of the respiratory system	J00 to J99
Diseases of the digestive system	K00 to K95
Diseases of the skin and subcutaneous tissue	L00 to L99
Diseases of the musculoskeletal system and connective tissue	M00 to M99
Diseases of the genitourinary system	N00 to N99
Pregnancy, childbirth, and the postpartum period	O00 to O99
Certain conditions originating in the perinatal period	P00 to P96
Congenital malformations, deformities, and chromosomic anomalies	Q00 to Q99
Abnormal symptoms, signs, and test results not otherwise classified	R00 to R99
Injuries, poisonings, and other consequences of external causes	S00 to T88
Codes for special purposes (ex: COVID-19)	U00 to U99
External causes of morbidity	V00 to Y99
Factors influencing health status and contact with health services	Z00 to Z99

Reducing the number of classes to predict offered several advantages, including a more manageable number of classes for modeling, making it easier to train models, reducing the risk of overfitting, and enhancing interpretability. Moreover, grouping diagnoses into broader chapters provided a more balanced class distribution and improved model generalization.

After pre-processing, relevant features from each file were merged into a single unified dataset using the patient NHC and episode reference identifiers. The resulting dataset contained 1045984 rows and 135 columns. Table 3 shows the distribution of diagnosis counts across the ICD-10-CM chapters, which was useful for identifying any class imbalances.

Subsequently, an Exploratory Data Analysis (EDA) was performed to better understand the final dataset structure. This included examining data distributions, identifying data types, and detecting potential imbalances or biases. The resulting plots are provided in Annex B.

Table 3: *Distribution of diagnoses counts across the ICD-10-CM chapters.*

ICD-10-CM Chapter Name	Count	Percentage
Factors influencing health status and contact with health services	211968	20.26%
Diseases of the genitourinary system	111154	10.63%
Certain infections and parasitic diseases	108634	10.39%
Diseases of the digestive system	93312	8.92%
Abnormal symptoms, signs, and test results not otherwise classified	82998	7.93%
Injuries, poisonings, and other consequences of external causes	74850	7.16%
Diseases of the respiratory system	59964	5.73%
Tumours (neoplasms)	55096	5.27%
Diseases of the blood and blood-forming organs and disorders affecting the immunological mechanism	48742	4.66%
External causes of morbidity	46363	4.43%
Diseases of the circulatory system	40953	3.92%
Endocrine, nutritional, and metabolic diseases	33772	3.23%
Congenital malformations, deformities, and chromosomic anomalies	29363	2.81%
Mental and behavioural disorders	16418	1.57%
Diseases of the nervous system	12700	1.21%
Diseases of the musculoskeletal system and connective tissue	8305	0.79%
Diseases of the skin and subcutaneous tissue	6767	0.65%
Codes for special purposes (ex: COVID-19)	3515	0.34%
Diseases of the eye and its appendages	585	0.06%
Pregnancy, childbirth, and the postpartum period	525	0.05%

5.3 Feature selection

Before performing feature selection, all categorical variables were encoded using appropriate techniques. Specifically, ordinal variables were encoded using label encoding via the *LabelEncoder* function in *sklearn.preprocessing*, ensuring that the natural order was maintained. As for nominal categorical variables, one-hot encoding was applied using the *OneHotEncoder* function to avoid introducing any ordinal relationships.

Following the encoding step, feature selection was carried out to identify the most relevant features for the classification tasks and generate different subsets for model evaluation. As discussed in Section 4.3, there are various methods for feature selection, for this project, two approaches were used. The first method involved tree-based feature importance, while the second used univariate statistical selection through the ANOVA F-test.

For the tree-based method, an ensemble of decision trees was constructed using the *ExtraTreesClassifier* class from *sklearn*. This algorithm builds an ensemble of 100 randomized trees, each trained on random subsets of the data which helps improve generalization and reduce overfitting. Once the model was trained, feature importance scores were extracted using the *feature_importances_* attribute. These importance scores measures each feature's contribution to reducing impurity in the classification trees. The top 20 features, ranked from most to least important, are presented in Figure 10.

The second method applied was a univariate feature selection using the ANOVA F-test. In this method, each feature was individually evaluated for its statistical significance in relation to the target diagnosis variable. The *SelectKBest* function, using the F-score metric, selected the top 20 features with the highest discriminatory power, as shown in Figure 11.

Both methods produced ranked lists of important features. Detailed results and visualizations for both methods can be found in Annex C.

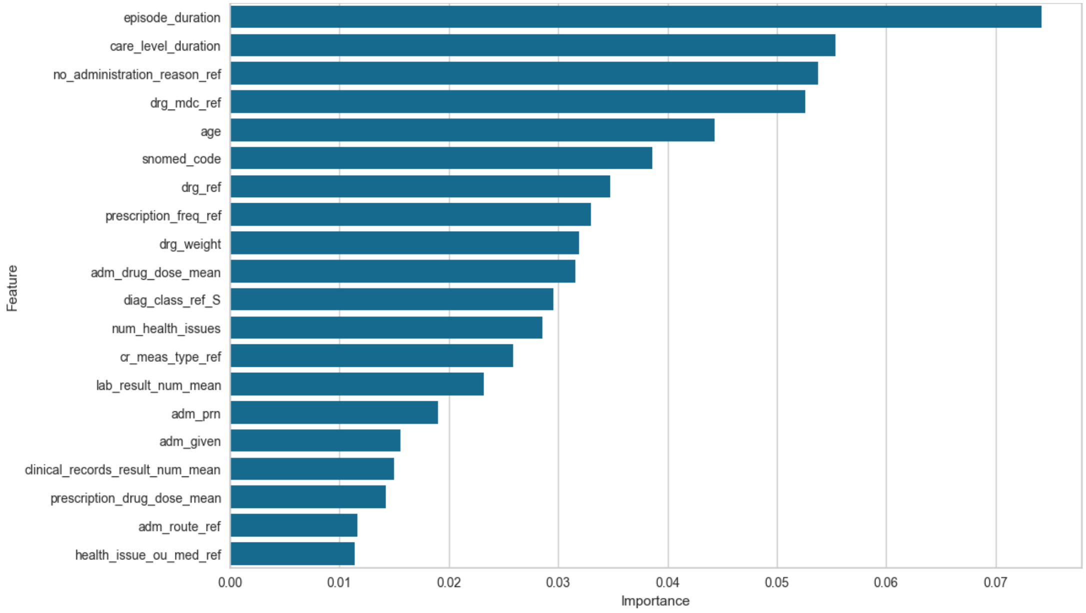


Figure 10: *Top 20 most important features based on Decision Trees.*

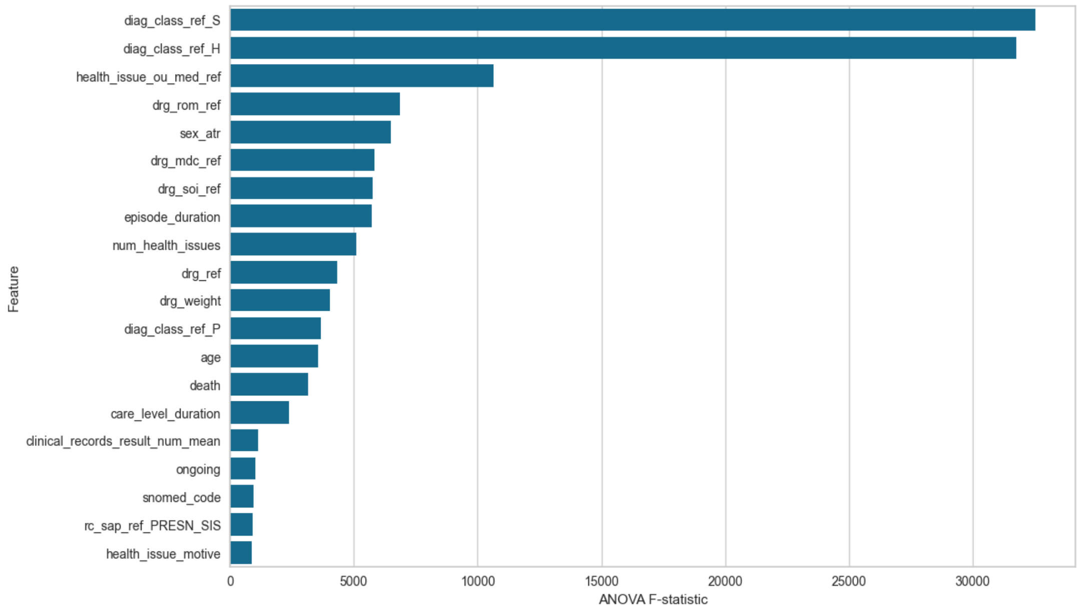


Figure 11: *Top 20 most important features based on ANOVA F-test.*

5.3.1 Definition of the subsets

To optimise classification performance and evaluate the impact of different groups of features, several subsets were generated by combining various input variables. Each subset represents a specific selection of features, based on the results of the feature selection methods described earlier.

Table 4 provides an overview of the different subsets generated along with the number of variables in each of them. A comprehensive list of all variables included in each subset can be found in Annex D.

Table 4: *Description of the different subsets and the num.*

Subset	Description	Number of variables
Subset 1	Dataset with all the features	134
Subset 2	Dataset with the top 20 featured based on decision trees	20
Subset 3	Dataset with the top 10 featured based on decision trees	10
Subset 4	Dataset with the top 20 featured based on ANOVA test	20
Subset 5	Dataset with the top 10 featured based on ANOVA test	10
Subset 6	Dataset with only the features that appear in both the top 20 from decision trees and ANOVA test	11
Subset 7	Dataset with all the features from the top 20 of both decision trees and ANOVA test.	29

5.4 Supervised Machine Learning model selection

After defining the different subsets, the next step was to perform model selection for each subset. As outlined in Section 4.4, a range of supervised ML models were considered. To identify the model that delivered the best performance for each subset, a comparative analysis of the different models was conducted using PyCaret, a library that automates various ML workflows, enabling efficient model development, comparison, and tuning.

The model selection process began with the use of the *setup()* function, which initializes the experiment within PyCaret and establishes the transformation pipeline according to the parameters provided. During this step, the data is also split into training (70%) and testing (30%) sets.

Subsequently, the *compare_models()* function from the Pycaret library was employed to train and evaluate the selected estimators. This function performs a 10-fold stratified cross-validation, providing a robust estimate of the model's performance while preserving the class distribution in each fold, which is an important consideration when handling imbalanced datasets.

The output is a ranked table of models with their corresponding average performance metrics across folds. The following tables summarize the comparative performance results of various ML models evaluated on the different subsets.

Table 5: *Performance of various ML models on Subset 1.*

Model	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
DT	0.6602	0.8137	0.6602	0.6603	0.6602	0.6238	0.6238
XGBoost	0.4619	0.9076	0.4619	0.4709	0.4639	0.4085	0.4089
KNN	0.3046	0.7567	0.3046	0.3132	0.3053	0.2318	0.2323
MLP	0.2026	0.5000	0.2026	0.0411	0.0683	0.0000	0.0000
LR	0.2005	0.0000	0.2005	0.0409	0.0679	-0.0019	-0.0136
SVM	0.0685	0.0000	0.0685	0.0107	0.0136	0.0000	0.0007

Table 6: *Performance of various ML models on Subset 2.*

Model	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
DT	0.6627	0.8175	0.6627	0.6632	0.6629	0.6267	0.6267
XGBoost	0.4538	0.9063	0.4538	0.4662	0.4569	0.4003	0.4008
KNN	0.3280	0.7748	0.3280	0.3359	0.3287	0.2575	0.2580
MLP	0.2026	0.5000	0.2026	0.0411	0.0683	0.0000	0.0000
LR	0.2005	0.0000	0.2005	0.0409	0.0679	-0.0019	-0.0137
SVM	0.1186	0.0000	0.1186	0.0176	0.0301	0.0000	-0.0002

Table 7: *Performance of various ML models on Subset 3.*

Model	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
XGBoost	0.2823	0.8387	0.2823	0.2833	0.2769	0.2107	0.2114
RF	0.2710	0.8038	0.2710	0.2688	0.2688	0.1926	0.1928
DT	0.2707	0.7866	0.2707	0.2737	0.2693	0.1937	0.1941
KNN	0.2596	0.7178	0.2596	0.2694	0.2608	0.1830	0.1836
MLP	0.2026	0.5000	0.2026	0.0411	0.0683	0.0000	0.0000
LR	0.2003	0.0000	0.2003	0.0411	0.0679	-0.0020	-0.0137
SVM	0.0803	0.0000	0.0803	0.0121	0.0188	0.0001	-0.0002

Table 8: *Performance of various ML models on Subset 4.*

Model	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
DT	0.4429	0.9052	0.4429	0.4431	0.4427	0.3840	0.3841
RF	0.4429	0.9056	0.4429	0.4417	0.4420	0.3837	0.3837
XGBoost	0.4282	0.9024	0.4282	0.4428	0.4323	0.3725	0.3731
KNN	0.3288	0.7588	0.3288	0.3411	0.3306	0.2585	0.2593
MLP	0.2026	0.5000	0.2026	0.0411	0.0683	0.0000	0.0000
LR	0.2008	0.0000	0.2008	0.0410	0.0680	-0.0016	-0.0128
SVM	0.0806	0.0000	0.0806	0.0114	0.0194	0.0002	0.0004

Table 9: *Performance of various ML models on Subset 5.*

Model	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
RF	0.4014	0.8985	0.4014	0.4455	0.4004	0.3450	0.3478
XGBoost	0.4014	0.8985	0.4014	0.4407	0.3975	0.3445	0.3474
DT	0.4013	0.8985	0.4013	0.4458	0.4006	0.3453	0.3481
KNN	0.3960	0.7831	0.3960	0.4836	0.4070	0.3385	0.3437
MLP	0.3869	0.8854	0.3869	0.4818	0.3779	0.3219	0.3287
LR	0.2588	0.0000	0.2588	0.1901	0.1916	0.1249	0.1409
SVM	0.2009	0.0000	0.2009	0.2018	0.1600	0.1068	0.1206

***Table 10:** Performance of various ML models on Subset 6.*

Model	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
DT	0.4422	0.9050	0.4422	0.4424	0.4420	0.3832	0.3833
RF	0.4422	0.9054	0.4422	0.4410	0.4413	0.3829	0.3829
XGBoost	0.4268	0.9021	0.4268	0.4413	0.4311	0.3710	0.3716
KNN	0.4099	0.8032	0.4099	0.4242	0.4137	0.3479	0.3486
MLP	0.2026	0.5000	0.2026	0.0411	0.0683	0.0000	0.0000
LR	0.2004	0.0000	0.2004	0.0409	0.0679	-0.0020	-0.0151
SVM	0.0980	0.0000	0.0980	0.0155	0.0261	0.0002	0.0004

***Table 11:** Performance of various ML models on Subset 7.*

Model	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
DT	0.6644	0.8185	0.6644	0.6648	0.6646	0.6286	0.6286
XGBoost	0.4555	0.9065	0.4555	0.4675	0.4584	0.4002	0.4027
KNN	0.3360	0.7811	0.3360	0.3448	0.3369	0.2662	0.2668
MLP	0.2026	0.5000	0.2026	0.0411	0.0683	0.0000	0.0000
LR	0.2007	0.0000	0.2007	0.0410	0.0679	-0.0017	-0.0122
SVM	0.1186	0.0000	0.1186	0.0176	0.0301	0.0000	-0.0022

After evaluating the performance of the various ML models, the best model for each subset was selected based on overall performance metrics. In general, ensemble-based models such as decision trees, random forest, and XGBoost were the top performers, this is most likely because of their ability to capture complex non-linear relationships and interactions within the data.

The final selected models for each subset are summarized in Table 12.

***Table 12:** Final model selected for each subset.*

Subset	Selected model
Subset 1	Decision trees
Subset 2	Decision trees
Subset 3	XGBoost
Subset 4	Decision trees
Subset 5	Random Forest
Subset 6	Decision trees
Subset 7	Decision trees

5.5 Hyperparameter tuning

Following the initial training and evaluation of the best performing model for each data subset, a hyperparameter tuning process was conducted to identify the best combination of hyperparameter values that maximize model performance. This tuning was performed using PyCaret's *tune_model()* function, which by default employs *RandomGridSearch*. This method efficiently explores a wide range of hyperparameter combinations by sampling randomly from specified distributions. However, in cases where the default random grid search did not lead to performance improvements, a more targeted and exhaustive tuning approach was conducted using *GridSearchCV* from *sklearn*. This allowed the evaluation of specific hyperparameter combinations based on a custom-defined parameter grid.

For decision tree models, the key hyperparameters considered during tuning included:

- ***criterion***: it determines the function used to evaluate the quality of a split.
- ***max_depth***: limits the maximum depth of the tree to prevent overfitting.
- ***min_samples_leaf***: specifies the minimum number of samples required to be present at a leaf node.
- ***min_samples_split***: sets the minimum number of samples needed to split an internal node.

As for random forest, the key parameters included:

- ***criterion***: it works similarly to the one used in decision trees.
- ***max_depth***: limits tree depth to reduce overfitting.
- ***n_estimators***: is the number of trees in the forest. Increasing this value generally improves model performance and stability but has a higher computational cost.

Finally, for XGBoost models, the primary hyperparameters tuned were:

- ***learning_rate***: also known as eta, controls the step size at each boosting iteration.
- ***max_depth***: influences the complexity of each individual tree.
- ***n_estimators***: defines the number of boosting rounds.

The following tables compare the performance metrics of the selected models for each subset before and after hyperparameter tuning.

Table 13: *Performance of DT model on Subset 1 before and after hyperparameter tuning.*

	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
Before	0.6602	0.8137	0.6602	0.6603	0.6602	0.6238	0.6238
After	0.6602	0.8137	0.6602	0.6603	0.6602	0.6238	0.6238
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

***Table 14:** Performance of DT model on Subset 2 before and after hyperparameter tuning.*

	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
Before	0.6627	0.8175	0.6627	0.6632	0.6629	0.6267	0.6267
After	0.6694	0.8213	0.6694	0.6699	0.6696	0.6341	0.6341
	+0.0067	+0.0038	+0.0067	+0.0067	+0.0067	+0.0074	+0.0074

***Table 15:** Performance of XGBoost model on Subset 3 before and after hyperparameter tuning.*

	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
Before	0.2823	0.8387	0.2823	0.2833	0.2769	0.2107	0.2114
After	0.2875	0.8398	0.2875	0.2888	0.2827	0.2156	0.2164
	+0.0052	+0.0011	+0.0052	+0.0055	+0.058	+0.0049	+0.0050

***Table 16:** Performance of DT model on Subset 4 before and after hyperparameter tuning.*

	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
Before	0.4429	0.9052	0.4429	0.4431	0.4427	0.3840	0.3841
After	0.4429	0.9052	0.4429	0.4432	0.4427	0.3840	0.3841
	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000

***Table 17:** Performance of RF model on Subset 5 before and after hyperparameter tuning.*

	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
Before	0.4014	0.8985	0.4014	0.4455	0.4004	0.3450	0.3478
After	0.4015	0.8985	0.4015	0.4463	0.4002	0.3449	0.3476
	+0.0001	0.0000	+0.0001	+0.0008	-0.0002	-0.0001	-0.0002

***Table 18:** Performance of DT model on Subset 6 before and after hyperparameter tuning.*

	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
Before	0.4422	0.9050	0.4422	0.4424	0.4420	0.3832	0.3833
After	0.4422	0.9050	0.4422	0.4424	0.4420	0.3832	0.3833
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 19: *Performance of DT model on Subset 7 before and after hyperparameter tuning.*

	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
Before	0.6644	0.8185	0.6644	0.6648	0.6646	0.6286	0.6286
After	0.6698	0.8215	0.6704	0.6704	0.6700	0.6346	0.6346
	+0.0054	+0.0030	+0.0060	+0.0056	+0.0054	+0.0060	+0.0060

As shown in the performance comparison tables, the models trained on Subsets 2, 3, and 7 demonstrated the most significant improvements after hyperparameter tuning. This suggest that these subsets contained feature combinations particularly sensitive to parameter optimization, allowing the models to better capture underlying patterns in the data. In contrast, the remaining subsets showed only small improvements, indicating that either the default hyperparameters were already nearly optimal or that the feature combinations were less complex, and thus offering limited room for improvement.

Table 20 presents the best performing hyperparameter values identified for each subset.

Table 20: *Best hyperparameter values for each subset.*

Subset	Best hyperparameters

Subset 1	criterion = ’entropy’
	max_depth = None
	min_samples_leaf = 1
	min_samples_split = 2

Subset 2	criterion = ‘log_loss’
	max_depth = None
	min_samples_leaf = 1
	min_samples_split = 2

Subset 3	colsample_bytree = 0.9
	learning_rate = 0.15
	max_depth = 7
	min_child_weight = 3
	n_estimators = 290

Subset 4	criterion = ’entropy’
	max_depth = None
	min_samples_leaf = 1
	min_samples_split = 2

Subset 5	criterion = gini
	max_depth = 15
	n_estimators = 300

Subset 6	criterion = ’entropy’
	max_depth = None
	min_samples_leaf = 1
	min_samples_split = 2

Subset 7	criterion = ‘log_loss’
	max_depth = None
	min_samples_leaf = 1
	min_samples_split = 2

5.6 Model testing

After optimizing and tuning the hyperparameters for each model, the final step was to evaluate the model’s performance on the unseen test set. This step provides a realistic estimate of how the model would perform in a real-world setting.

To carry out this step, the *predict_model()* function in PyCaret was used. This function applies the final tuned model to the previously split test set and computes key performance metrics. These metrics provide a comprehensive view of the model’s ability to correctly classify patient diagnoses across multiple classes.

Table 21 presents the final performance metrics for each subset, organized by best overall performance.

Table 21: *Performance results on the test set. Ranked by best overall performance.*

	Accuracy	AUC	Recall	Precision	F1	Kappa	MCC
Subset 1	0.6895	0.8303	0.6895	0.6898	0.6896	0.6564	0.6564
Subset 7	0.6889	0.8327	0.6889	0.6895	0.6891	0.6556	0.6556
Subset 2	0.6875	0.8320	0.6875	0.6881	0.6878	0.6541	0.6541
Subset 6	0.4446	0.9054	0,4446	0.4433	0.4437	0.3856	0.3856
Subset 4	0.4453	0.9056	0.4453	0.4441	0.4444	0.3864	0.3865
Subset 5	0.4032	0.8985	0.4032	0.4500	0.3991	0.3457	0.3484
Subset 3	0.2879	0.8398	0.2879	0.2901	0.2834	0.2164	0.2171

5.7 Results and discussion

After evaluating the final models on the test set, a clear performance distinction can be observed across the different subsets. Subsets 1, 7, and 2 show the best overall performance, achieving an average accuracy of approximately 68.9%, with similar recall, precision, and F1-scores. These subsets also achieved the highest Cohen's Kappa and Matthew's Correlation Coefficient (MCC) scores, indicating strong agreement beyond chance and balanced performance across multiple classes. These results suggest that the feature combinations in Subsets 1, 7, and 2 provide a well-balanced and informative representation of the patient data, enabling the models to generalize effectively on unseen cases.

Subsets 6, 4, and 5 achieved significantly lower performance, with accuracy values around 40% to 44%. However, they recorded very high AUC values, indicating that while the model was able to rank classes well, its final classification thresholds may not have been optimal, possibly due to class imbalance. This discrepancy between AUC and classification metrics suggest the potential benefit of threshold calibration or cost-sensitive learning in future work.

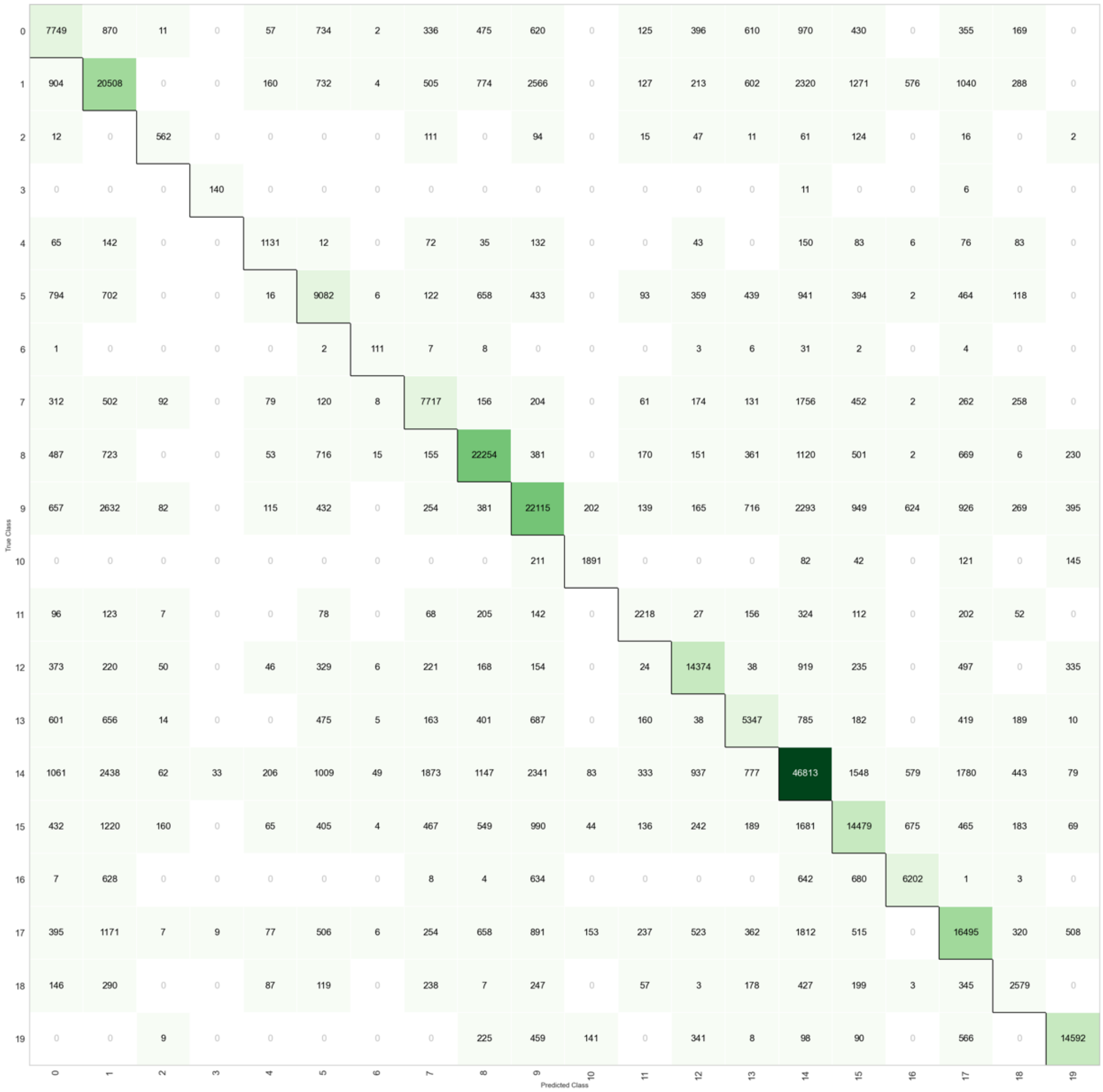
Subset 3, despite requiring XGBoost, one of the most complex models, has shown the lowest performance metrics. This poor performance indicates that the feature selection of this subset. Did not provide enough discriminatory power, or that the complexity of the model may have led to overfitting during training and poor generalization.

To further analyse the results, several plots were generated, including confusion matrices, classification reports, and feature importance visualizations. Together, they provide a clearer understanding of which classes are most accurately predicted, where misclassifications occur, and which features contribute most to the predictions.

An analysis of the confusion matrix (Figure 12) and the classification report (Figures 13) for Subset 1 reveals significant variation in the model's predictive performance across different classes. Specifically, certain classes like Class 14 and Class 8 exhibit high precision and recall, indicating strong predictive reliability. In contrast, other classes, like Class 2 and Class 3, are frequently misclassified. This suggest that the model struggles to learn their distinct features.

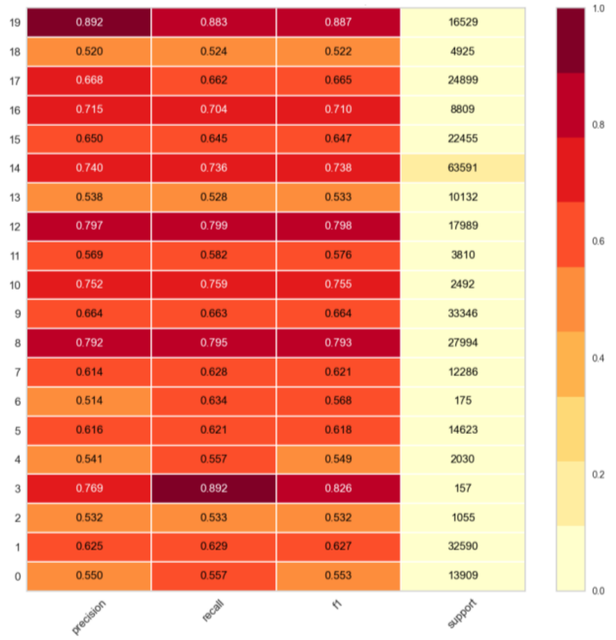
This discrepancy is not coincidence, instead, it reflects a clear correlation between class distribution and model performance. As shown in Table 22, classes with a higher number of samples tend to achieve better classification outcomes, whereas classes with fewer samples are more susceptible to error. This imbalance introduces bias into the model, making it more likely to favour majority classes.

For this reason, addressing class imbalance during data pre-processing is essential. Future improvements could include the use of resampling methods such as Synthetic Minority Over-sampling Technique (SMOTE), generating synthetic data, or incorporating class-weighted loss functions during training. These methods can help the model learn meaningful patterns across all classes and improving overall performance.


True Class
Predicted Class

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	7749	870	11	0	57	734	2	336	475	620	0	125	396	610	970	430	0	355	169	0
1	904	20508	0	0	160	732	4	505	774	2566	0	127	213	602	2320	1271	576	1040	288	0
2	12	0	562	0	0	0	0	111	0	94	0	15	47	11	61	124	0	16	0	2
3	0	0	0	140	0	0	0	0	0	0	0	0	0	0	11	0	0	6	0	0
4	65	142	0	0	1131	12	0	72	35	132	0	0	43	0	150	83	6	76	83	0
5	794	702	0	0	16	9082	6	122	658	433	0	93	359	439	941	394	2	464	118	0
6	1	0	0	0	0	2	111	7	8	0	0	0	3	6	31	2	0	4	0	0
7	312	502	92	0	79	120	8	7717	156	204	0	61	174	131	1756	452	2	262	258	0
8	487	723	0	0	53	716	15	155	22254	381	0	170	151	361	1120	501	2	669	6	230
9	657	2632	82	0	115	432	0	254	381	22115	202	139	165	716	2293	949	624	926	269	395
10	0	0	0	0	0	0	0	0	0	211	1891	0	0	0	82	42	0	121	0	145
11	96	123	7	0	0	78	0	68	205	142	0	2218	27	156	324	112	0	202	52	0
12	373	220	50	0	46	329	6	221	168	154	0	24	14374	38	919	235	0	497	0	335
13	601	656	14	0	0	475	5	163	401	687	0	160	38	5347	785	182	0	419	189	10
14	1061	2438	62	33	206	1009	49	1873	1147	2341	83	333	937	777	46813	1548	579	1780	443	79
15	432	1220	160	0	65	405	4	467	549	990	44	136	242	189	1681	14479	675	465	183	69
16	7	628	0	0	0	0	0	8	4	634	0	0	0	0	642	680	6202	1	3	0
17	395	1171	7	9	77	506	6	254	658	891	153	237	523	362	1812	515	0	16495	320	508
18	146	290	0	0	87	119	0	238	7	247	0	57	3	178	427	199	3	345	2579	0
19	0	0	9	0	0	0	0	0	225	459	141	0	341	8	98	90	0	566	0	14592

***Figure 12:** Confusion matrix for DT model of Subset 1.*


1.0
0.8
0.6
0.4
0.2
0.0

	precision	recall	f1	support
19	0.892	0.883	0.887	16529
18	0.520	0.524	0.522	4925
17	0.668	0.662	0.665	24899
16	0.715	0.704	0.710	8809
15	0.650	0.645	0.647	22455
14	0.740	0.736	0.738	63591
13	0.538	0.528	0.533	10132
12	0.797	0.799	0.798	17989
11	0.569	0.582	0.576	3810
10	0.752	0.759	0.755	2492
9	0.664	0.663	0.664	33346
8	0.792	0.795	0.793	27994
7	0.614	0.628	0.621	12286
6	0.514	0.634	0.568	175
5	0.616	0.621	0.618	14623
4	0.541	0.557	0.549	2030
3	0.769	0.892	0.826	157
2	0.532	0.533	0.532	1055
1	0.625	0.629	0.627	32590
0	0.550	0.557	0.553	13909

***Figure 13:** Classification report for DT model of Subset 1.*

***Table 22:** Model assigned class numbers and corresponding ICD-10-CM chapter.*

Class	ICD-10-CM Chapter Name	Count	Percentage
14	Factors influencing health status and contact with health services	211968	20.26%
9	Diseases of the genitourinary system	111154	10.63%
1	Certain infections and parasitic diseases	108634	10.39%
8	Diseases of the digestive system	93312	8.92%
17	Abnormal symptoms, signs, and test results not otherwise classified	82998	7.93%
15	Injuries, poisonings, and other consequences of external causes	74850	7.16%
12	Diseases of the respiratory system	59964	5.73%
19	Tumours (neoplasms)	55096	5.27%
5	Diseases of the blood and blood-forming organs and disorders affecting the immunological mechanism	48742	4.66%
0	External causes of morbidity	46363	4.43%
7	Diseases of the circulatory system	40953	3.92%
13	Endocrine, nutritional, and metabolic diseases	33772	3.23%
16	Congenital malformations, deformities, and chromosomic anomalies	29363	2.81%
18	Mental and behavioural disorders	16418	1.57%
11	Diseases of the nervous system	12700	1.21%
10	Diseases of the musculoskeletal system and connective tissue	8305	0.79%
4	Diseases of the skin and subcutaneous tissue	6767	0.65%
2	Codes for special purposes (ex: COVID-19)	3515	0.34%
6	Diseases of the eye and its appendages	585	0.06%
3	Pregnancy, childbirth, and the postpartum period	525	0.05%

The main objective of this project was to investigate whether health problems coded in SNOMED CT (*snomed_code* variable) can effectively serve as predictors for discharge diagnoses coded in ICD-10-CM. Additionally, the project also aimed to identify the most important input features to predict discharge diagnoses.

To explore this, features importance plots from the models obtained from Subsets 1, 7 and 2 were computed. These plots provide insight into the relative contribution of each variable to the predictive performance of the trained models. For additional performance plots across all subsets, please refer to Annex E.

Figure 14 shows the top 10 most important features from the model trained on Subset 1. As we can see the *snomed_code* variable ranked 9th, out of a total of 135 variables in the subset. In comparison, Figure 15, which corresponds to Subset 2, shows that *snomed_code* ranked 12th. Lastly, Figure 16, which represents Subset 7, places snomed_code at 15th in importance. This consistency suggest that while *snomed_code* is not one of the top predictors, it consistently appears across all subsets, indicating moderate importance. Although it is not the most influential predictor on its own, it still provides valuable information for predicting final diagnoses and performs best when combined with other clinical and demographic features.

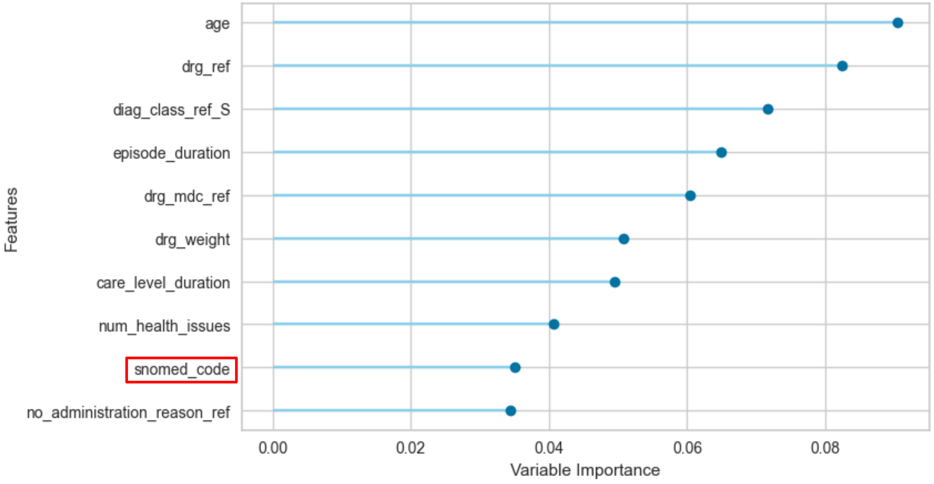


Figure 14: *Top 10 most important features from DT model on Subset 1.*

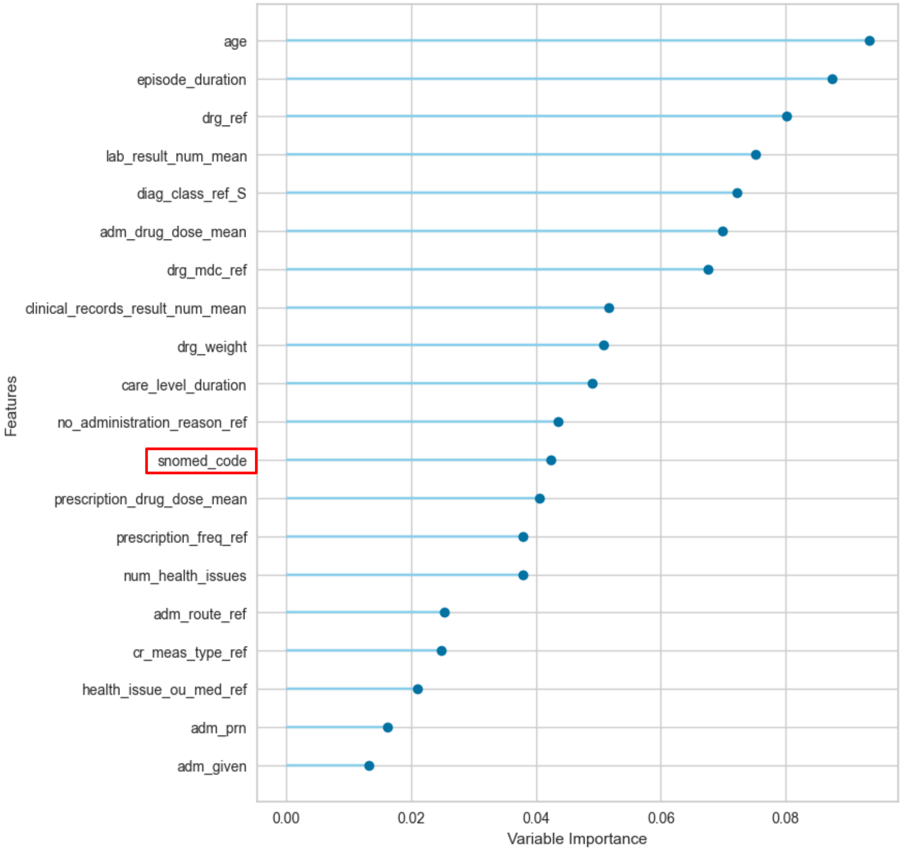


Figure 15: *Feature importance from DT model on Subset 2.*

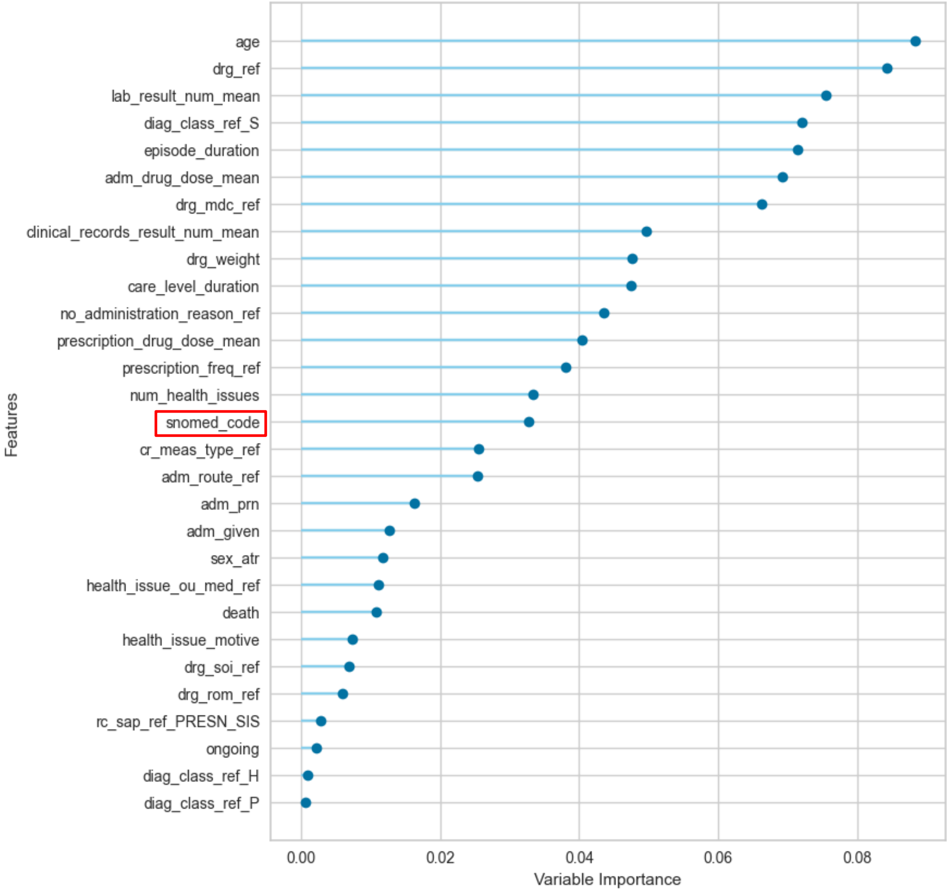


***Figure 16:** Feature importance from DT model on Subset 7.*

From the feature importance plots, we can see that several variables were repeatedly ranked highly across all three models. This suggest their strong relevance in predicting discharge diagnoses:

- **age**: could reflect age-related comorbidities and disease patterns.
- **drg_ref**: represents Diagnosis Related Group reference, which are clinically grouped conditions used mainly for billing and reimbursement purposes.
- **episode_duration**: may correlate with illness severity or complexity of treatment.
- **diag_class_ref_S**: diagnosis classification level.
- **lab_result_num_mean**: average lab test results.
- **adm_drug_dose_mean**: average drug administration dose.
- **drg_mdc_ref**: Major Diagnostic Category (MDC)
- **care_level_duration**: length of the care level.

Some factors that could explain why snomed_code did not emerge as one of the top predictors for ICD-10-CM discharge diagnoses are:

- **Granularity and mapping challenges**: as explained in Section 2, SNOMED CT codes are highly granular and capture detailed clinical information. However, the target variable, corresponds to a broader diagnostic category. The inherent complexity of mapping detailed SNOMED CT concepts to generalized ICD-10 codes introduces limitations.

- **Variation in coding practices**: in the clinical setting, healthcare professionals have not consistently recorded health problems in SNOMED CT unless required. As a result, there is a bias, there are diagnostics with more complete SNOMED CT coding.
- **Lack of standardized use among clinicians**: many users, are not yet fully trained or incentivised to systematically document health problems using SNOMED CT. This results in underreporting or inconsistent coding, which reduces the completeness and reliability of the variable across the dataset.

5.7.1 Limitations

This section highlights the primary limitations encountered during the project. Acknowledging these challenges is important in order to effectively inform and direct future research efforts.

The first challenge encountered is the dataset size and complexity. Handling data from 15 different files, each containing different types of clinical information, required significant effort in terms of cleaning, processing, and merging. Clinical datasets are inherently messy, often containing incomplete records, and variables that are difficult to interpret without expert knowledge. Additionally, healthcare data is subject to a wide range of biases, including missing data, errors in coding, and discrepancies between clinical observations and final diagnoses. Despite rigorous pre-processing, some noise and inconsistency likely remained in the data.

Another significant challenge was the imbalance of diagnostic categories in the dataset. Some ICD-10-CM chapters were heavily represented, while others appeared infrequently. This imbalance can lead ML models to favour majority classes and reducing sensitivity to less frequent diagnoses. Although multiclass classification metrics such as AUC and F1-score were used, class imbalance likely affected overall generalizability and may have contributed to biased predictions.

Finally, processing and analysing high-dimensional healthcare data, especially during pre-processing, model training and hyperparameter tuning, was computationally intensive. While PyCaret streamlined much of the workflow, the underlying algorithms, particularly ensemble methods like random forest and XGBoost, still demanded substantial memory and processing time. These limitations restricted the number of experiments that could be conducted, for example, during hyperparameter grid search, potentially narrowing the optimization of the model's performance.

6 EXECUTION SCHEDULE

6.1 Work Breakdown Structure

The Work Breakdown Structure (WBS) is a project management tool that breaks down a project into smaller, more manageable components. It provides a structured overview of the fundamental elements required for a successful execution of the project. In this case, the WBS is divided into four main sections: project preparation, data pre-processing, Machine Learning models, and project report. Each of these sections is further divided into specific tasks to provide a detailed understanding of the project workflow. Figure 17 illustrates the activities included in each of the main sections. A detailed description of these individual tasks, along with their estimated durations, is provided below.

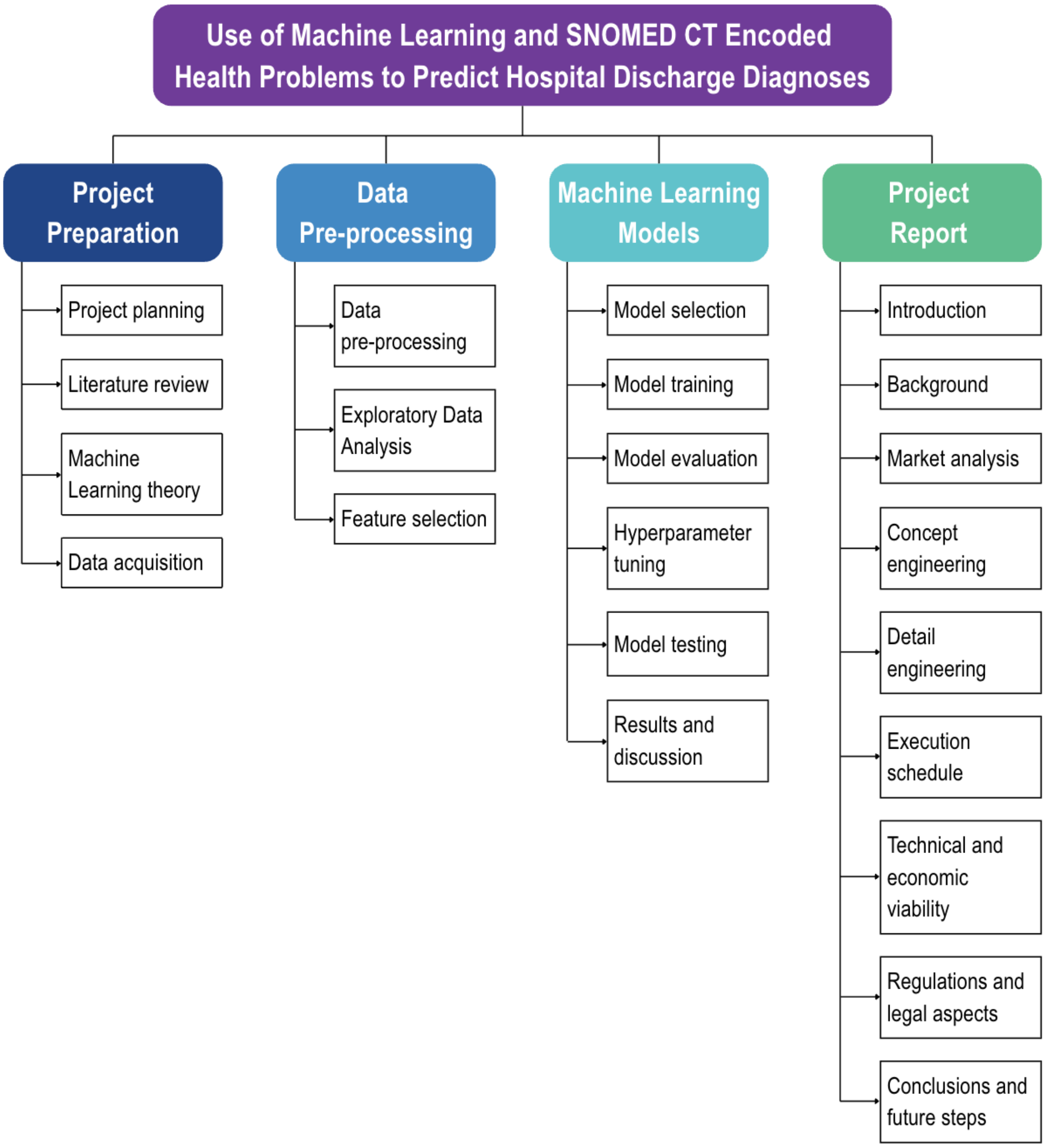


Figure 17: *Work Breakdown Structure (WBS) diagram of the project.*

6.1.1 WBS dictionary

Table 23: *WBS dictionary for "Project Preparation" stage.*

1	Project Preparation	
1.1	**Project planning**	Duration: 7 days
During this phase, the key activities required to complete the project are identified, and a clear and structured work methodology is established. With the help of the tutors of the project, the project's goals and scope are also defined. These goals have to be realistic, specific, and achievable within the given timeframe and resource constraints.		
1.2	**Literature review**	Duration: 14 days
A comprehensive literature review is conducted to gather relevant information and insights about the project. This includes performing background research to understand the theorical foundations of the project, researching previous works, and analysing the current state of the art in the field. Alongside this, a market analysis is performed to explore current trends, potential applications, and future opportunities related to the project. To manage and organize all the consulted sources efficiently, the reference management software Mendeley was used.		
1.3	**Machine Learning theory**	Duration: 14 days
Reviewing the theoretical background of Machine Learning algorithms relevant to the project by researching various ML models and studying their fundamental concepts and principles.		
1.4	**Data acquisition**	Duration: 58 days
Ask the project's director for the data and analyse and understand its structure and content. It involves reviewing the data format, identifying key features, and consulting with the tutor to clarify the meaning of various columns and how to properly handle them during data pre-processing.		

Table 24: *WBS dictionary for "Data Pre-processing" stage.*

2	Data Pre-processing	
2.1	**Data pre-processing**	Duration: 61 days
Preparing the data for analysis. This step includes, identifying missing values applying normalization or scaling techniques if necessary to avoid introducing inaccuracies or bias. The goal is to ensure the final dataset is clean, consistent, and ready for analysis and model training.		
2.2	**Exploratory Data Analysis**	Duration: 3 days
Conducting an Exploratory Data Analysis (EDA) to understand the main characteristics of the dataset and examine how each variable behaves. This step involves applying data visualization techniques to identify trends, relationships, and potential correlations.		
2.3	**Feature selection**	Duration: 21 days
Identifying and selecting the most relevant features that contribute to the predicting the final diagnosis. This step involves applying different feature importance techniques to eliminate or reduce irrelevant columns.		

Table 25: *WBS dictionary for "Machine Learning Models" stage.*

3	Machine Learning Models	
3.1	**Model selection**	Duration: 7 days
Identifying and selecting the most appropriate ML models. This step involves comparing different models and their performances. This helps determine which model gives best results keeping in mind the objectives of the project.		
3.2	**Model training**	Duration: 9 days
Training the selected model using the training set of the dataset. This step allows the algorithm to learn the patterns and relationships between the input features and the target variable.		
3.3	**Model evaluation**	Duration: 5 days
Asses the performance of the model using different performance metrics such as accuracy, AUC, recall, precision, or F1 score. This step helps assess how well the model performs.		
3.4	**Hyperparameter tuning**	Duration: 7 days
Optimizing the model's predictive performance and results by adjusting the hyperparameters through different techniques such as random search or grid search. The objective of this step is to find the best combination of parameters that improve the model's performance.		
3.5	**Model testing**	Duration: 7 days
Evaluate the final model on a testing set to evaluate its real-world performance. This provides an unbiased assessment of how well the model generalized to unseen data and confirms the robustness of the model.		
3.6	**Results and discussion**	Duration: 7 days
Present and summarize the model's results, highlighting the key findings and performance outcomes. This step also provide an analysis of the results by discussing the limitations of the project, interpret the implications of the results, and reflect on what could be improved.		

Table 26: *WBS dictionary for "Project Report" stage.*

4	Project Report	
4.1	**Introduction**	Duration: 7 days
Write the introduction section of the project by describing the motivation behind the project, defining the clear objectives and scope, and provide an overview of the methodology used to carry out the project.		
4.2	**Background**	Duration: 14 days
Overview of the theorical foundations necessary to understand the context of the project. It involves summarizing key concepts and developments related to the project as well as identifying current challenges, and limitations of ML in predicting discharge diagnoses.		

<table>
<tr><td><b>4.3</b></td><td><b>Market analysis</b></td><td>Duration: 7 days</td></tr>
<tr><td colspan="3">Analyse the healthcare market sector by identifying the target market and potential customers. This section also involves a discussion of future perspectives, emerging trends, and opportunities that could arise in this sector.</td></tr>
<tr><td><b>4.4</b></td><td><b>Concept engineering</b></td><td>Duration: 19 days</td></tr>
<tr><td colspan="3">Description and evaluation of the different methods that could be used to achieve the project objectives. This section includes outlining different approaches considered and explaining the reasoning behind the chosen method.</td></tr>
<tr><td><b>4.5</b></td><td><b>Detailed engineering</b></td><td>Duration: 28 days</td></tr>
<tr><td colspan="3">Describe the practical implementation of the project, detailing the steps taken during the project, such as data handling, feature selection, model selection, model training, model evaluation, hyperparameter tuning, model testing, and the generation of results.</td></tr>
<tr><td><b>4.6</b></td><td><b>Execution schedule</b></td><td>Duration: 7 days</td></tr>
<tr><td colspan="3">Develop an execution schedule that includes a PERT diagram to identify critical activities that must not be delayed, and a GANTT diagram to keep track of the activities that need to be completed throughout the project.</td></tr>
<tr><td><b>4.7</b></td><td><b>Technical and economic viability</b></td><td>Duration: 4 days</td></tr>
<tr><td colspan="3">Assess the project’s technical and economic viability. This step includes the development of a SWOT analysis to identify strengths, weaknesses, opportunities, and threats, as well as an evaluation of the project’s costs.</td></tr>
<tr><td><b>4.8</b></td><td><b>Regulations and legal aspects</b></td><td>Duration: 3 days</td></tr>
<tr><td colspan="3">Review relevant regulations, standards, and legal considerations that may affect the project. This step also aims to identify any potential legal challenges associated with the project.</td></tr>
<tr><td><b>4.9</b></td><td><b>Conclusions and future steps</b></td><td>Duration: 7 days</td></tr>
<tr><td colspan="3">Write and summarize the key findings and outcomes of the project. This section discusses the lessons learned, limitations encountered, and proposed possible future steps or work to further improve the project.</td></tr>
</table>

6.2 Program Evaluation and Review Technique

The Program Evaluation and Review Technique (PERT) is a tool used in project management designed to analyse and map out the tasks needed to complete a project. In Table 27, the list of all project activities, dependencies, and the estimated duration is represented. Based on this information, a PERT chart is generated (Figure 18), where each task is represented by an arrow, and the connecting points, also known as nodes, indicate key project milestones. The top number in each node is its ID, while the bottom numbers represent time metrics: the left number is the earliest possible start time (t early), and the right number, is the latest acceptable finish time (t last) for preceding tasks without causing project delays. The critical path, highlighted in purple, refers to the set of tasks where the margin for delay is zero, meaning that the earliest start and the latest finish time is the same. Any delay in these tasks will result in a delay in the entire project.

Table 27: *Activity table for the PERT diagram.*

ID	Activity name	Dependencies	Duration (days)
A	Project planning	-	7
B	Literature review	A	14
C	Machine Learning theory	B	14
D	Data acquisition	-	58
E	Data pre-processing	D	61
F	Exploratory Data Analysis	E	3
G	Feature selection	F	21
H	Model selection	G	7
I	Model training	H	9
J	Model evaluation	I	5
K	Hyperparameter tuning	J	7
L	Model testing	K	7
M	Results and discussion	L	7
N	Introduction	A	7
O	Background	B	14
P	Market analysis	B	7
Q	Concept engineering	C	19
R	Detail engineering	Q	28
S	Execution schedule	A	7
T	Technical and economic viability	R	4
U	Regulations and legal aspects	R	3
V	Conclusions and future steps	M, T, U	7

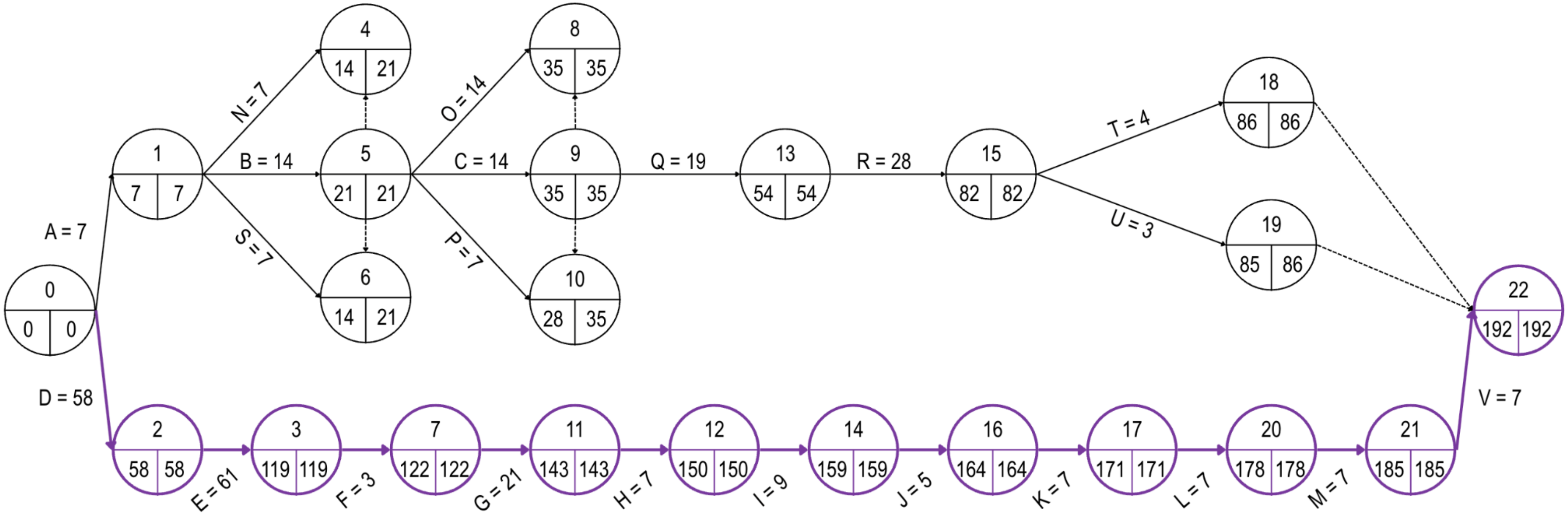


Figure 18: *PERT diagram of the project.*

6.3 GANTT diagram

A GANTT diagram is a visual project management tool that outlines the timeline of tasks and milestones involved in completing a project. It shows the start and end dates for each activity involved.

The project took place from October 2024 to May 2025. The first months were dedicated to bibliographic research and a review of ML theory. After acquiring the data, the focus of the project shifted towards developing the model. As it can be seen in Figure 19, a significant portion of that time was dedicated to signal pre-processing, reflecting its crucial role in ensuring the success of subsequent steps.

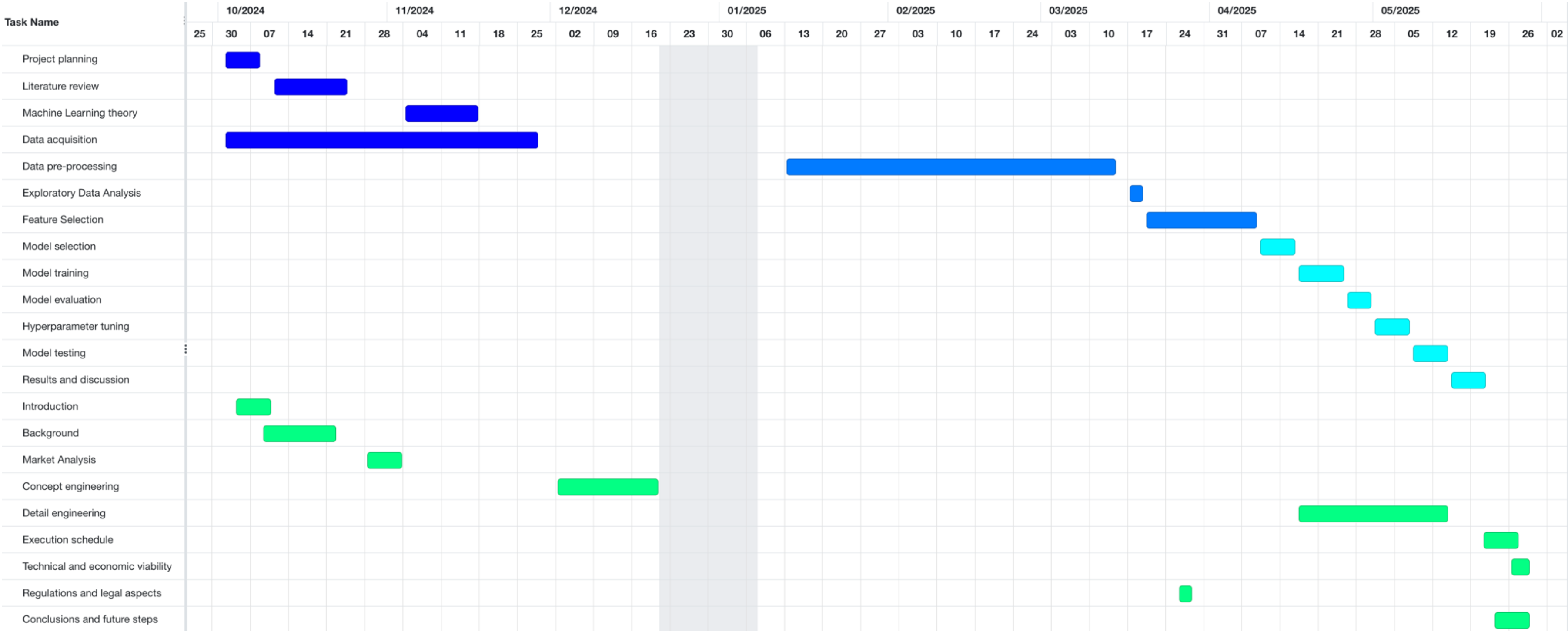


Figure 19: *GANTT diagram of the project.*

7 TECHNICAL VIABILITY

To evaluate the technical viability of the project, a SWOT analysis is conducted (Table 28). This approach helps to identify the strengths, weaknesses, opportunities, and threats related to the project's technical aspects, allowing for an assessment of both internal and external factors that may impact its success.

By analysing the strengths, we aim to emphasize the project's technical expertise, valuable assets, and unique resources that provide a strategic advantage over competitors. Identifying these strengths allow us to understand what differentiates the project and contributes to its success.

On the other hand, identifying weaknesses allow us to uncover internal challenges and resources limitations that may hinder the project's development or performance. Early recognition of these limitations allows for focused improvements to prevent possible setbacks.

In the opportunities section, external trends, market changes and developments, and technological innovations that the project can capitalize on to enhance growth are examined. This analysis helps position the project to take advantage of emerging possibilities.

Finally, the threats evaluation addresses external risks, such as competitive pressures, regulatory changes, or technological disruptions, which could undermine the project's technical feasibility. Acknowledging these threats support strategic planning to reduce their potential impact.

Table 28: *SWOT analysis of the project.*

Strengths	Weaknesses
- The dataset is large and contains multiple diverse features. - Knowledge on ML and Python. - Use of automated libraries like PyCaret that facilitate model development and optimization. - Uncover complex patterns in clinical data.	- Imbalance and largeness of the dataset. - Complexity constraints. - Limited computational resources - Limited personal experience. - Limited interpretability
Opportunities	**Threats**
- Growing market for diagnosis prediction. - Advancements in AI and ML present opportunities to enhance accuracy and efficiency of predictive models. - Integration with clinical workflows.	- Data privacy and security. - Compliance with legal and ethical standards. - Regulatory approval. - Bias and fairness.

8 ECONOMIC VIABILITY

The economic viability of the project is evaluated by examining the three main components required for its successful execution: data, technical resources, and human resources.

The dataset used in this study was provided by the Hospital Clínic de Barcelona, so there was no data acquisition cost. However, maintaining access to such clinical data typically involves administrative efforts and potential expenses related to data governance, privacy compliance, and security measures.

As for technical resources, the project was carried out using a personal computer. The computer used required sufficient processing power and memory to handle data pre-processing, model training, and evaluation. Using open-source software libraries such as PyCaret and Scikit-learn helped minimize software licensing costs. However, advanced ML workflows, especially with larger datasets or more complex models, may require investment in high-performance computing resources or cloud services, which could increase operational costs.

Finally, regarding the human resources, the project team consisted of the principal researcher, me, and the supervising tutor and project director. The human resources were estimated according to the salary of a Biomedical Engineer graduate salary.

Table 29 shows an estimation of the project costs.

Table 29: *Estimation of the project costs.*

	Description	Quantity	Estimated cost
Data	Data acquisition	1	0 €
Technical resources	Personal computer	1	800 €
	Visual Studio Code	1	0 €
Human resources	Biomedical engineer	1 (400 h)	8.40 €/hour
	Project manager	1 (8 months)	2000 €/month
		TOTAL	**20160 €**

9 REGULATIONS AND LEGAL ASPECTS

The implementation of ML in healthcare require careful consideration of various legal, ethical, and regulatory frameworks, especially when working with sensitive clinical data. This section outlines the regulatory challenges that must be considered.

9.1 Data protection and patient privacy

The dataset used in this study consist of real patient data obtained from the Hospital Clínic de Barcelona. As such, strict adherence to data protection regulations was essential. This study was approved by the Ethical comity of the hospital (see Annex A) and all patient identifiable information was removed or anonymized before data processing to ensure privacy. Additionally, access to the dataset was restricted to authorized individuals involved in the project.

9.2 Ethical considerations

Data was used solely for research and model development, with no clinical decisions or interventions based on the predictions. However, the models used in the project learn from the input data and, as a result, may also reflect any inherent biases present within that data.

No direct interaction with patients or medical interventions occurred during the study, so no additional ethical approval was required. However, future applications of these models in a real-world clinical setting would require approval from a clinical ethics board.

9.3 Medical device regulation

The models generated in this project are intended solely for research purposes. However, if this was to be applied into a clinical decision support system, several regulatory and legal aspects would need to be addressed. Under the European Medical Device Regulation (MDR) (EU) 2017/745, any software designed to process, analyse, generate, or modify medical information must comply with rigorous standards to ensure safety, performance, and alignment with its intended medical use [70].

AI-driven diagnostic tools may be classified as medical device software, requiring CE marking and formal validation. Additionally, clear policies must be established to define accountability for decisions made with AI support, especially in the cases of misdiagnosis. Finally, the use of AI in clinical environments require a certain level of transparency and explainability to meet both ethical standards and professional guidelines.

10 CONCLUSIONS AND FUTURE STEPS

This project aimed to explore the relationship between SNOMED CT encoded health problems and discharge diagnoses coded in ICD-10-CM. Using real clinical data from the Hospital Clínic de Barcelona, several supervised ML models were trained and evaluated across different subsets, achieving promising results. The best performing models achieved accuracies close to 69%, with high consistency across other metrics such as AUC, recall, and precision. These finding suggest that health problems are not only correlated with final diagnoses but can also serve as valuable inputs in data-driven clinical decision support systems.

Feature importance analysis across subsets revealed that variables such as age, DRG, episode duration, lab results, and drug dosage consistently contributed to prediction accuracy. These insights determine that demographic data, treatment duration, and ongoing patient monitoring are crucial in coding final diagnoses.

This study demonstrated the potential of ML to support diagnostic decision-making and highlighted how it can offer decision support tools that could help improve diagnostic accuracy, resource allocation, and overall care quality in hospital environments.

Despite the promising results, several areas for improvement were identified. The imbalanced distribution of classes led to challenges in model sensitivity. Future models could implement techniques like SMOTE, or class weighting to better handle imbalanced data. As for interpretability, introducing explainability tools such as SHAP or LIME would make them more interpretable to clinical users and increase their practical applicability. Furthermore, the use of generative AI models, particularly Large Language Models (LLMs), could significantly improve the prediction of discharge diagnoses coded in ICD-10 based on health problems initially coded in SNOMED-CT at the beginning of the care process. Unlike traditional ML approaches, which often rely on statistical correlations and may fail to capture deeper semantic relationships, LLMs possess a more advanced ability to model clinical progression and the conceptual connections between symptoms, syndromes, and formal diagnoses. This enables more realistic and clinically coherent interferences, facilitating the consolidation of care trajectories from early observations to structured diagnoses, even when those concepts do not share explicit semantic or hierarchical structures in the source terminologies. Moreover, this approach may help address the challenge of mapping between SNOMED-CT and ICD-10, where relationships are many-to-many or lack formal correspondences altogether. Instead of relying on rigid evidence mapping, LLMs can interpret from contextual patterns in data how a SNOMED-CT coded problem, may correspond to an ICD-10 coded diagnosis. This is made possible by the semantic proximity and clinical plausibility derived from large scale patterns in text or structured data, opening the door to more flexible and intelligent terminology bridging.

11 REFERENCES

1. Wang, W., Ferrari, D., Haddon-Hill, G., & Curcin, V. (2023). Electronic Health Records as Source of Research Data. *Neuromethods*, *197*, 331–354. https://doi.org/10.1007/978-1-0716-3195-9_11

2. *Electronic Health Records | CMS*. (n.d.). Retrieved April 25, 2025, from https://www.cms.gov/priorities/key-initiatives/e-health/records

3. Weed, L. L. (1968). Medical Records That Guide and Teach. *New England Journal of Medicine*, *278*(11), 593–600. https://doi.org/10.1056/NEJM196803142781105,

4. Jacobs, L. (2009). Interview with Lawrence Weed, MD— The Father of the Problem-Oriented Medical Record Looks Ahead. *The Permanente Journal*, *13*(3), 84. https://doi.org/10.7812/TPP/09-068

5. *Health and Well-Being*. (n.d.). Retrieved April 25, 2025, from https://www.who.int/data/gho/data/major-themes/health-and-well-being

6. Balogh, E. P., Miller, B. T., Ball, J. R., Care, C. on D. E. in H., Services, B. on H. C., Medicine, I. of, & The National Academies of Sciences, E. and M. (2015). *Technology and Tools in the Diagnostic Process*. https://www.ncbi.nlm.nih.gov/books/NBK338590/

7. Simons, S. M. J., Cillessen, F. H. J. M., & Hazelzet, J. A. (2016). Determinants of a successful problem list to support the implementation of the problem-oriented medical record according to recent literature. *BMC Medical Informatics and Decision Making*, *16*(1), 102. https://doi.org/10.1186/S12911-016-0341-0

8. National Library of Medicine, U. (2013). *Mapping SNOMED CT to ICD-10-CM Technical Specifications*. http://www.ihtsdo.org/our-standards/licensing/.

9. *What is SNOMED CT | SNOMED International*. (n.d.). Retrieved April 25, 2025, from https://www.snomed.org/what-is-snomed-ct

10. *Overview of SNOMED CT*. (n.d.). Retrieved April 25, 2025, from https://www.nlm.nih.gov/healthit/snomedct/snomed_overview.html

11. *2. SNOMED CT Benefits - SNOMED CT Starter Guide - SNOMED Confluence*. (n.d.). Retrieved April 25, 2025, from https://confluence.ihtsdotools.org/display/DOCSTART/2.+SNOMED+CT+Benefits

12. Isaradech, N., & Khumrin, P. (2021). Auto-mapping Clinical Documents to ICD-10 using SNOMED-CT. *AMIA Summits on Translational Science Proceedings*, *2021*, 296. https://pmc.ncbi.nlm.nih.gov/articles/PMC8378640/

13. Scichilone, R., & Giannangelo, K. (2013). *WHO-FIC INFORMATION SHEET International Classification of Diseases (ICD) and Standard Clinical Reference Terminologies: A 21 st Century Informatics Solution*. http://www.who.int/classifications/en/].

14. *Importance of ICD*. (n.d.). Retrieved April 25, 2025, from https://www.who.int/standards/classifications/frequently-asked-questions/importance-of-icd

15. *ICD-10-CM | Classification of Diseases, Functioning, and Disability | CDC*. (n.d.). Retrieved April 25, 2025, from https://www.cdc.gov/nchs/icd/icd-10-cm/index.html

16. Mahbubani, K., Georgiades, F., Goh, E. L., Chidambaram, S., Sivakumaran, P., Rawson, T., Ray, S., Hudovsky, A., & Gill, D. (2018). Clinician-directed improvement in the accuracy of hospital clinical coding. *Future Healthcare Journal*, *5*(1), 47. https://doi.org/10.7861/FUTUREHOSP.5-1-47

17. National Library of Medicine, U. (2013). *Mapping SNOMED CT to ICD-10-CM Technical Specifications*. http://www.ihtsdo.org/our-standards/licensing/.

18. Amisha, Malik, P., Pathania, M., & Rathaur, V. (2019). Overview of artificial intelligence in medicine. *Journal of Family Medicine and Primary Care*, *8*(7), 2328. https://doi.org/10.4103/JFMPC.JFMPC_440_19,

19. Chiu, H. Y., Chao, H. S., & Chen, Y. M. (2022). Application of Artificial Intelligence in Lung Cancer. *Cancers*, *14*(6), 1370. https://doi.org/10.3390/CANCERS14061370

20. Bzdok, D., Krzywinski, M., & Altman, N. (2017). Machine learning: a primer. *Nature Methods 2017 14:12*. https://www.nature.com/articles/nmeth.4526

21. Habehh, H., & Gohel, S. (2021). Machine Learning in Healthcare. *Current Genomics*, *22*(4), 291. https://doi.org/10.2174/1389202922666210705124359

22. Adlung, L., Cohen, Y., Mor, U., & Elinav, E. (2021). Machine learning in clinical decision making. *Med*, *2*(6), 642–665. https://doi.org/10.1016/J.MEDJ.2021.04.006/ASSET/2B8CA7C0-7021-4409-B5CA-634DAA17D471/MAIN.ASSETS/GR1.JPG

23. Mohanty, S. D., Lekan, D., McCoy, T. P., Jenkins, M., & Manda, P. (2022). Machine learning for predicting readmission risk among the frail: Explainable AI for healthcare. *Patterns*, *3*(1), 100395. https://doi.org/10.1016/J.PATTER.2021.100395

24. Johnson, K. B., Wei, W. Q., Weeraratne, D., Frisse, M. E., Misulis, K., Rhee, K., Zhao, J., & Snowdon, J. L. (2020). Precision Medicine, AI, and the Future of Personalized Health Care. *Clinical and Translational Science*, *14*(1), 86. https://doi.org/10.1111/CTS.12884

25. Wu, Y., Li, L., Xin, B., Hu, Q., Dong, X., & Li, Z. (2023). Application of machine learning in personalized medicine. *Intelligent Pharmacy*, *1*(3), 152–156. https://doi.org/10.1016/J.IPHA.2023.06.004

26. van der Velden, B. H. M., Kuijf, H. J., Gilhuijs, K. G. A., & Viergever, M. A. (2022). Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. *Medical Image Analysis*, *79*. https://doi.org/10.1016/j.media.2022.102470

27. Ghnemat, R., Alodibat, S., & Abu Al-Haija, Q. (2023). Explainable Artificial Intelligence (XAI) for Deep Learning Based Medical Imaging Classification. *Journal of Imaging*, *9*(9). https://doi.org/10.3390/JIMAGING9090177,

28. Arora, A., Alderman, J. E., Palmer, J., Ganapathi, S., Laws, E., McCradden, M. D., Oakden-Rayner, L., Pfohl, S. R., Ghassemi, M., McKay, F., Treanor, D., Rostamzadeh, N., Mateen, B., Gath, J., Adebajo, A. O., Kuku, S., Matin, R., Heller, K., Sapey, E., … Liu, X. (2023). The value of standards for health datasets in artificial intelligence-based applications. *Nature Medicine*, *29*(11), 2929. https://doi.org/10.1038/S41591-023-02608-W

29. Sadilek, A., Liu, L., Nguyen, D., Kamruzzaman, M., Serghiou, S., Rader, B., Ingerman, A., Mellem, S., Kairouz, P., Nsoesie, E. O., MacFarlane, J., Vullikanti, A., Marathe, M., Eastham, P., Brownstein, J. S., Arcas, B. A. y., Howell, M. D., & Hernandez, J. (2021). Privacy-first health research with federated learning. *Npj Digital Medicine*, *4*(1), 1–8. https://doi.org/10.1038/S41746-021-00489-2;SUBJMETA=308,639,692,705;KWRD=MATHEMATICS+AND+COMPUTING,MEDICAL+RESEARCH

30. Ghassemi, M., Naumann, T., Schulam, P., Beam, A. L., Chen, I. Y., & Ranganath, R. (2020). A Review of Challenges and Opportunities in Machine Learning for Health. *AMIA Summits on Translational Science Proceedings*, *2020*, 191. https://pmc.ncbi.nlm.nih.gov/articles/PMC7233077/

31. Lin, C., Hsu, C. J., Lou, Y. S., Yeh, S. J., Lee, C. C., Su, S. L., & Chen, H. C. (2017). Artificial Intelligence Learning Semantics via External Resources for Classifying Diagnosis Codes in Discharge Notes. *Journal of Medical Internet Research*, *19*(11), e380. https://doi.org/10.2196/JMIR.8344

32. Park, D. J., Park, M. W., Lee, H., Kim, Y. J., Kim, Y., & Park, Y. H. (2021). Development of machine learning model for diagnostic disease prediction based on laboratory tests. *Scientific Reports 2021 11:1*, *11*(1), 1–11. https://doi.org/10.1038/s41598-021-87171-5

33. *Healthcare Services Market Size & Opportunities Report, 2033*. (n.d.). Retrieved April 25, 2025, from https://www.businessresearchinsights.com/market-reports/healthcare-services-market-117601

34. Bitkina, O. V., Park, J., & Kim, H. K. (2023). Application of artificial intelligence in medical technologies: A systematic review of main trends. *Digital Health*, *9*, 20552076231189332. https://doi.org/10.1177/20552076231189331

35. Bitkina, O. V., Park, J., & Kim, H. K. (2023). Application of artificial intelligence in medical technologies: A systematic review of main trends. *Digital Health*, *9*, 20552076231189332. https://doi.org/10.1177/20552076231189331

36. Roos, N. P., Wennberg, J. E., & McPherson, K. (1988). Using diagnosis-related groups for studying variations in hospital admissions. *Health Care Financing Review*, *9*(4), 53. https://pmc.ncbi.nlm.nih.gov/articles/PMC4192882/

37. *ML | Handling Missing Values | GeeksforGeeks*. (n.d.). Retrieved April 26, 2025, from https://www.geeksforgeeks.org/ml-handling-missing-values/

38. *Categorical Data Encoding Techniques in Machine Learning | GeeksforGeeks*. (n.d.). Retrieved April 26, 2025, from https://www.geeksforgeeks.org/categorical-data-encoding-techniques-in-machine-learning/

39. *Data Normalization in Data Mining | GeeksforGeeks*. (n.d.). Retrieved April 26, 2025, from https://www.geeksforgeeks.org/data-normalization-in-data-mining/

40. *Understanding Feature Importance and Visualization of Tree Models | GeeksforGeeks*. (n.d.). Retrieved April 26, 2025, from https://www.geeksforgeeks.org/understanding-feature-importance-and-visualization-of-tree-models/

41. Kumar, M., & Rath, S. K. (2016). Feature Selection and Classification of Microarray Data using Machine Learning Techniques. *Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology: Systems and Applications*, 213–242. https://doi.org/10.1016/B978-0-12-804203-8.00015-8

42. Schober, P., & Schwarte, L. A. (2018). Correlation coefficients: Appropriate use and interpretation. *Anesthesia and Analgesia*, *126*(5), 1763–1768. https://doi.org/10.1213/ANE.0000000000002864

43. Fritz, M., & Berger, P. D. (2015). Will anybody buy? Logistic regression. *Improving the User Experience Through Practical Data Analytics*, 271–304. https://doi.org/10.1016/B978-0-12-800635-1.00011-2

44. Kleinbaum, D. G., & Klein, M. (n.d.). *Logistic Regression A Self-Learning Text Third Edition*. Retrieved May 28, 2025, from http://www.springer.com/series/2848

45. Cortes, C., Vapnik, V., & Saitta, L. (1995). Support-vector networks. *Machine Learning 1995 20:3*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

46. Liu, Z., & Xu, H. (2014). Kernel parameter selection for support vector machine classification. *Journal of Algorithms and Computational Technology*, *8*(2), 163–177. https://doi.org/10.1260/1748-3018.8.2.163

47. Messaoud, S., Bradai, A., Bukhari, S. H. R., Quang, P. T. A., Ahmed, O. Ben, & Atri, M. (2020). A survey on machine learning in Internet of Things: Algorithms, strategies, and applications. *Internet of Things*, *12*, 100314. https://doi.org/10.1016/J.IOT.2020.100314

48. Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, *1*(1), 81–106. https://doi.org/10.1023/A:1022643204877/METRICS

49. Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324/METRICS

50. Khan, M. Y., Qayoom, A., Nizami, M. S., Siddiqui, M. S., Wasi, S., & Raazi, S. M. K. U. R. (2021). Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive

Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques. *Complexity*, *2021*. https://doi.org/10.1155/2021/2553199

51. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Https://Doi.Org/10.1214/Aos/1013203451*, *29*(5), 1189–1232. https://doi.org/10.1214/AOS/1013203451

52. Chen, T., & Guestrin, C. (n.d.). *XGBoost: A Scalable Tree Boosting System*. https://doi.org/10.1145/2939672.2939785

53. Parveen, N., Gupta, M., Kasireddy, S., Ansari, M. S. H., & Ahmed, M. N. (2024). ECG based one-dimensional residual deep convolutional auto-encoder model for heart disease classification. *Multimedia Tools and Applications*, *83*(25), 66107–66133. https://doi.org/10.1007/S11042-023-18009-7

54. Cover, T. M., & Hart, P. E. (1967). Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

55. Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/NATURE14539;SUBJMETA=117,639,705;KWRD=COMPUTER+SCIENCE,MATHEMATICS+AND+COMPUTING

56. Choi, R. Y., Coyner, A. S., Kalpathy-Cramer, J., Chiang, M. F., & Peter Campbell, J. (2020). Introduction to Machine Learning, Neural Networks, and Deep Learning. *Translational Vision Science & Technology*, *9*(2), 14–14. https://doi.org/10.1167/TVST.9.2.14

57. *Understanding the Perceptron: A Foundation for Machine Learning Concepts*. (n.d.). Retrieved May 28, 2025, from https://www.lucentinnovation.com/blogs/technology-posts/understanding-the-perceptron

58. Chan, K. Y., Abu-Salih, B., Qaddoura, R., Al-Zoubi, A. M., Palade, V., Pham, D. S., Ser, J. Del, & Muhammad, K. (2023). Deep neural networks in the cloud: Review, applications, challenges and research directions. *Neurocomputing*, *545*, 126327. https://doi.org/10.1016/J.NEUCOM.2023.126327

59. Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics*, *17*(1), 168–192. https://doi.org/10.1016/J.ACI.2018.08.003/FULL/PDF

60. *What Is a Confusion Matrix and How Do You Interpret It?* (n.d.). Retrieved May 29, 2025, from https://plat.ai/blog/confusion-matrix-in-machine-learning/

61. Ahsan, M. M., Luna, S. A., & Siddique, Z. (2022). Machine-Learning-Based Disease Diagnosis: A Comprehensive Review. *Healthcare*, *10*(3), 541. https://doi.org/10.3390/HEALTHCARE10030541

62. McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia Medica*, *22*(3), 276. https://doi.org/10.11613/bm.2012.031

63. Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1), 1–13. https://doi.org/10.1186/S12864-019-6413-7/TABLES/5

64. Alsalem, M. A., Zaidan, A. A., Zaidan, B. B., Hashim, M., Madhloom, H. T., Azeez, N. D., & Alsyisuf, S. (2018). A review of the automated detection and classification of acute leukaemia: Coherent taxonomy, datasets, validation and performance measurements, motivation, open challenges and recommendations. *Computer Methods and Programs in Biomedicine*, *158*, 93–112. https://doi.org/10.1016/J.CMPB.2018.02.005

65. T R, M., V, V. K., V, D. K., Geman, O., Margala, M., & Guduri, M. (2023). The stratified K-folds cross-validation and class-balancing methods with high-performance ensemble classifiers for breast cancer classification. *Healthcare Analytics*, *4*, 100247. https://doi.org/10.1016/J.HEALTH.2023.100247

66. Duan, X. (2023). Automatic identification of conodont species using fine-grained convolutional neural networks. *Frontiers in Earth Science*, *10*. https://doi.org/10.3389/FEART.2022.1046327

67. Bergstra, J., Ca, J. B., & Ca, Y. B. (2012). Random Search for Hyper-Parameter Optimization Yoshua Bengio. *Journal of Machine Learning Research*, *13*, 281–305. http://scikit-learn.sourceforge.net.

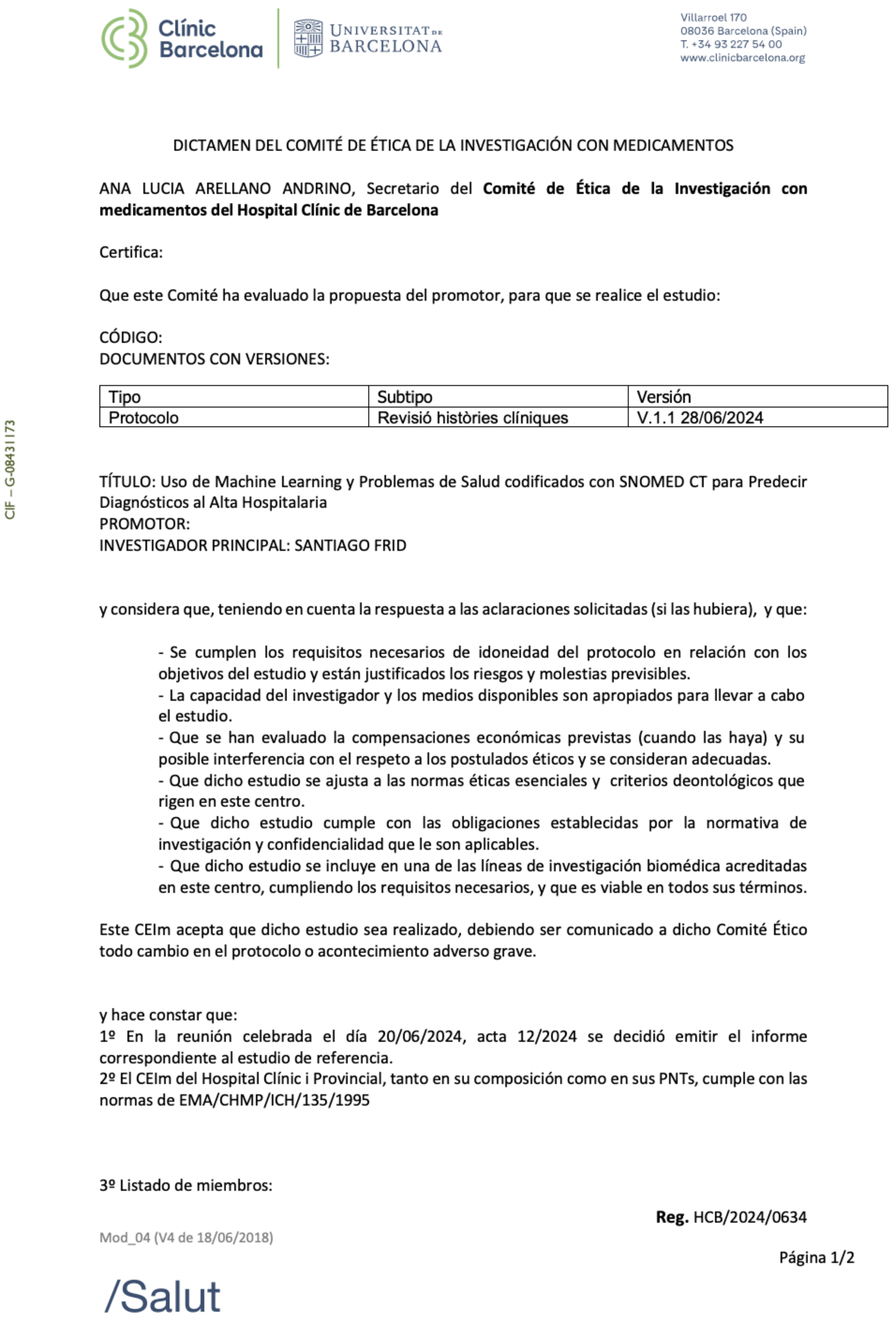
68. *PyCaret 3.0 | Docs*. (n.d.). Retrieved May 20, 2025, from https://pycaret.gitbook.io/docs

69. *Manual de Codifi cación*. (n.d.).

70. *Regulation - 2017/745 - EN - Medical Device Regulation - EUR-Lex*. (n.d.). Retrieved May 30, 2025, from https://eur-lex.europa.eu/eli/reg/2017/745/oj/eng

12 ANNEXES

ANNEX A. CEIm Approval



Clínic Barcelona | UNIVERSITAT DE BARCELONA

Villarroel 170
08036 Barcelona (Spain)
T. +34 93 227 54 00
www.clinicbarcelona.org

CIF – G-08431173

DICTAMEN DEL COMITÉ DE ÉTICA DE LA INVESTIGACIÓN CON MEDICAMENTOS

ANA LUCIA ARELLANO ANDRINO, Secretario del **Comité de Ética de la Investigación con medicamentos del Hospital Clínic de Barcelona**

Certifica:

Que este Comité ha evaluado la propuesta del promotor, para que se realice el estudio:

CÓDIGO:
DOCUMENTOS CON VERSIONES:

Tipo	Subtipo	Versión
Protocolo	Revisió històries clíniques	V.1.1 28/06/2024

TÍTULO: Uso de Machine Learning y Problemas de Salud codificados con SNOMED CT para Predecir Diagnósticos al Alta Hospitalaria
PROMOTOR:
INVESTIGADOR PRINCIPAL: SANTIAGO FRID

y considera que, teniendo en cuenta la respuesta a las aclaraciones solicitadas (si las hubiera), y que:

- Se cumplen los requisitos necesarios de idoneidad del protocolo en relación con los objetivos del estudio y están justificados los riesgos y molestias previsibles.
- La capacidad del investigador y los medios disponibles son apropiados para llevar a cabo el estudio.
- Que se han evaluado la compensaciones económicas previstas (cuando las haya) y su posible interferencia con el respeto a los postulados éticos y se consideran adecuadas.
- Que dicho estudio se ajusta a las normas éticas esenciales y criterios deontológicos que rigen en este centro.
- Que dicho estudio cumple con las obligaciones establecidas por la normativa de investigación y confidencialidad que le son aplicables.
- Que dicho estudio se incluye en una de las líneas de investigación biomédica acreditadas en este centro, cumpliendo los requisitos necesarios, y que es viable en todos sus términos.

Este CEIm acepta que dicho estudio sea realizado, debiendo ser comunicado a dicho Comité Ético todo cambio en el protocolo o acontecimiento adverso grave.

y hace constar que:
1º En la reunión celebrada el día 20/06/2024, acta 12/2024 se decidió emitir el informe correspondiente al estudio de referencia.
2º El CEIm del Hospital Clínic i Provincial, tanto en su composición como en sus PNTs, cumple con las normas de EMA/CHMP/ICH/135/1995

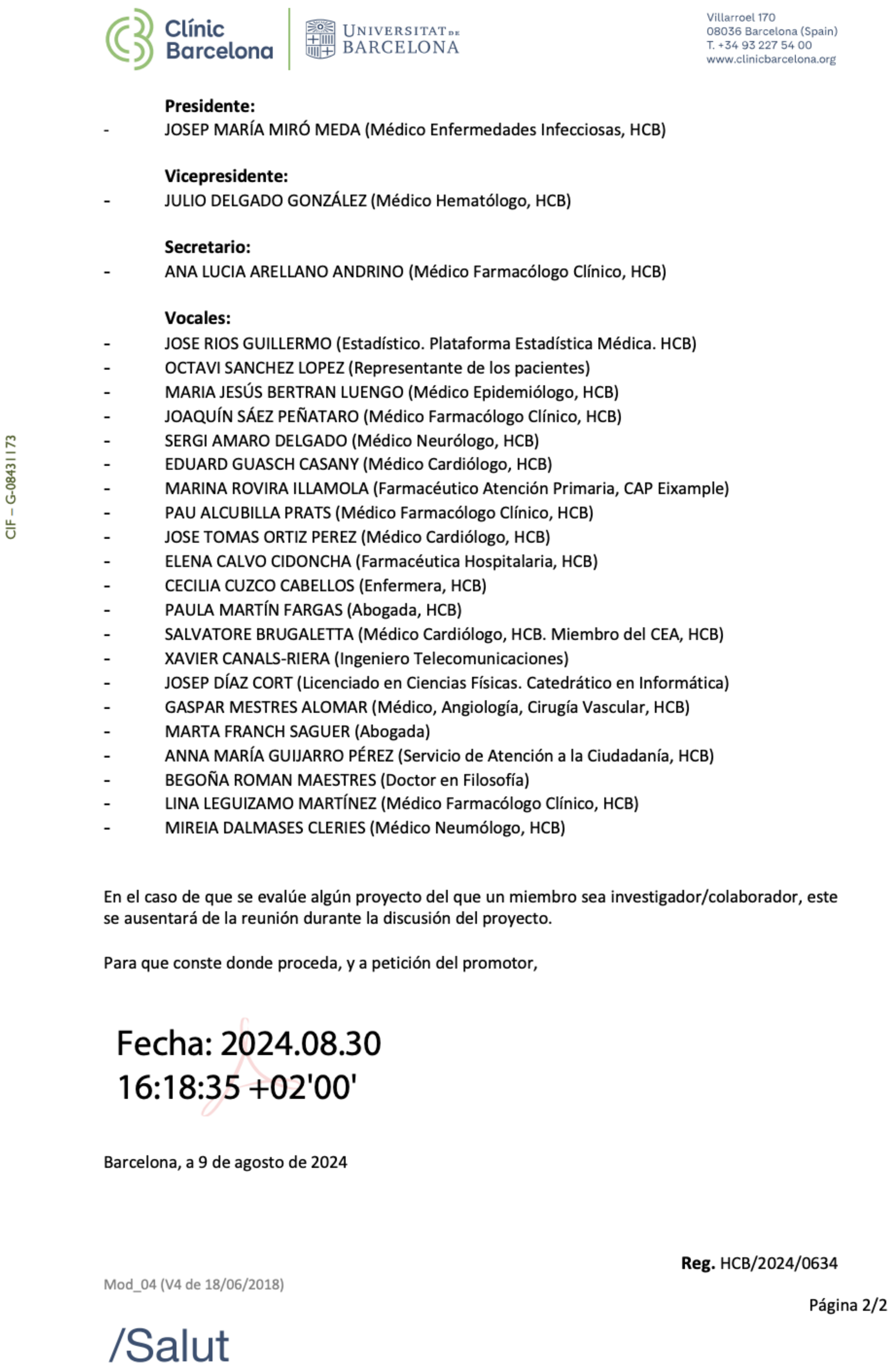
3º Listado de miembros:

Reg. HCB/2024/0634

Mod_04 (V4 de 18/06/2018)

Página 1/2

/Salut



Clínic Barcelona | UNIVERSITAT DE BARCELONA

Villarroel 170
08036 Barcelona (Spain)
T. +34 93 227 54 00
www.clinicbarcelona.org

CIF – G-08431173

Presidente:

- JOSEP MARÍA MIRÓ MEDA (Médico Enfermedades Infecciosas, HCB)

Vicepresidente:

- JULIO DELGADO GONZÁLEZ (Médico Hematólogo, HCB)

Secretario:

- ANA LUCIA ARELLANO ANDRINO (Médico Farmacólogo Clínico, HCB)

Vocales:

- JOSE RIOS GUILLERMO (Estadístico. Plataforma Estadística Médica. HCB)
- OCTAVI SANCHEZ LOPEZ (Representante de los pacientes)
- MARIA JESÚS BERTRAN LUENGO (Médico Epidemiólogo, HCB)
- JOAQUÍN SÁEZ PEÑATARO (Médico Farmacólogo Clínico, HCB)
- SERGI AMARO DELGADO (Médico Neurólogo, HCB)
- EDUARD GUASCH CASANY (Médico Cardiólogo, HCB)
- MARINA ROVIRA ILLAMOLA (Farmacéutico Atención Primaria, CAP Eixample)
- PAU ALCUBILLA PRATS (Médico Farmacólogo Clínico, HCB)
- JOSE TOMAS ORTIZ PEREZ (Médico Cardiólogo, HCB)
- ELENA CALVO CIDONCHA (Farmacéutica Hospitalaria, HCB)
- CECILIA CUZCO CABELLOS (Enfermera, HCB)
- PAULA MARTÍN FARGAS (Abogada, HCB)
- SALVATORE BRUGALETTA (Médico Cardiólogo, HCB. Miembro del CEA, HCB)
- XAVIER CANALS-RIERA (Ingeniero Telecomunicaciones)
- JOSEP DÍAZ CORT (Licenciado en Ciencias Físicas. Catedrático en Informática)
- GASPAR MESTRES ALOMAR (Médico, Angiología, Cirugía Vascular, HCB)
- MARTA FRANCH SAGUER (Abogada)
- ANNA MARÍA GUIJARRO PÉREZ (Servicio de Atención a la Ciudadanía, HCB)
- BEGOÑA ROMAN MAESTRES (Doctor en Filosofía)
- LINA LEGUIZAMO MARTÍNEZ (Médico Farmacólogo Clínico, HCB)
- MIREIA DALMASES CLERIES (Médico Neumólogo, HCB)

En el caso de que se evalúe algún proyecto del que un miembro sea investigador/colaborador, este se ausentará de la reunión durante la discusión del proyecto.

Para que conste donde proceda, y a petición del promotor,

Fecha: 2024.08.30
16:18:35 +02'00'

Barcelona, a 9 de agosto de 2024

Reg. HCB/2024/0634

Mod_04 (V4 de 18/06/2018)

Página 2/2

/Salut

ANNEX B. Exploratory Data Analysis

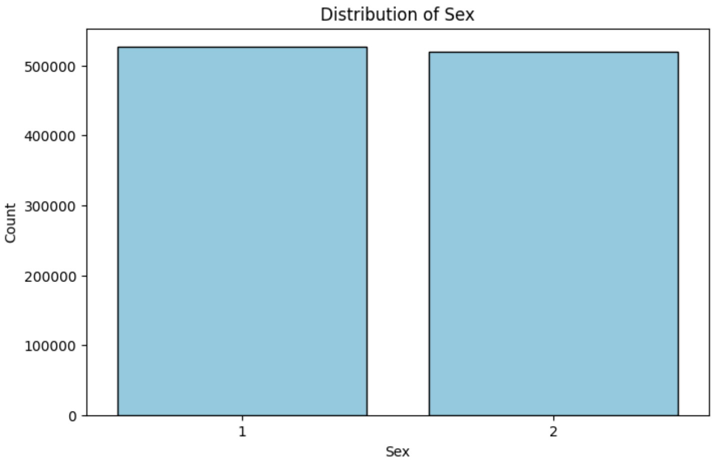


Figure B.1: *Distribution of sex of the dataset.*

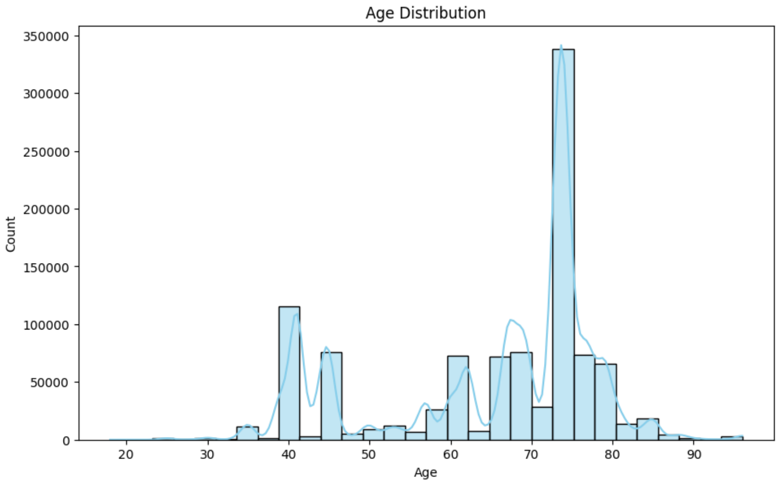


Figure B.2: *Age distribution of the dataset.*

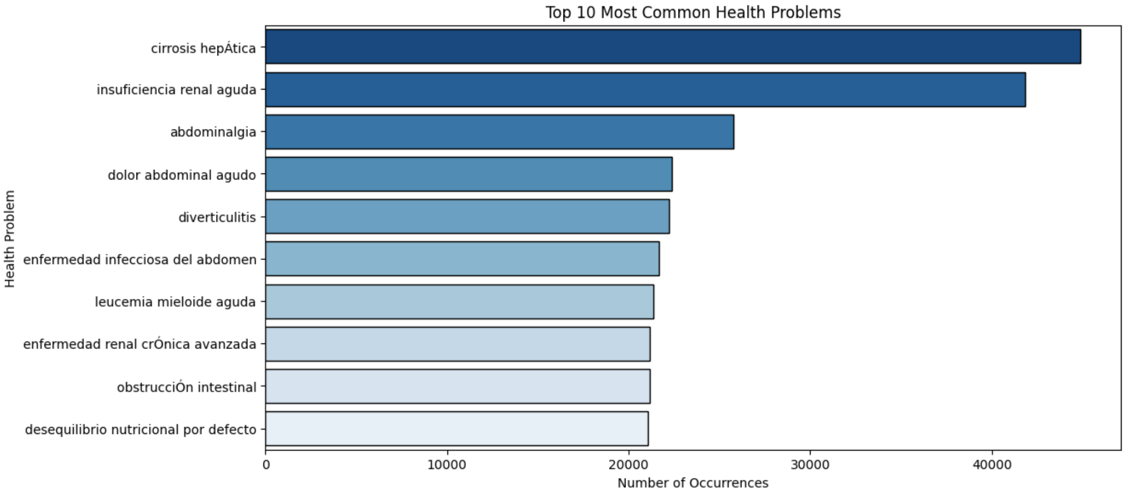


Figure B.3: *Top 10 most common health problems in the dataset.*

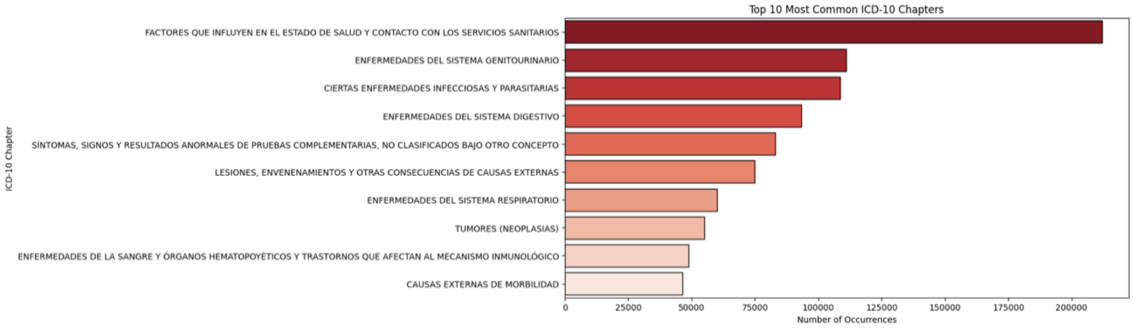


Figure B.4: *Top 10 most common ICD-10 chapters in the dataset.*

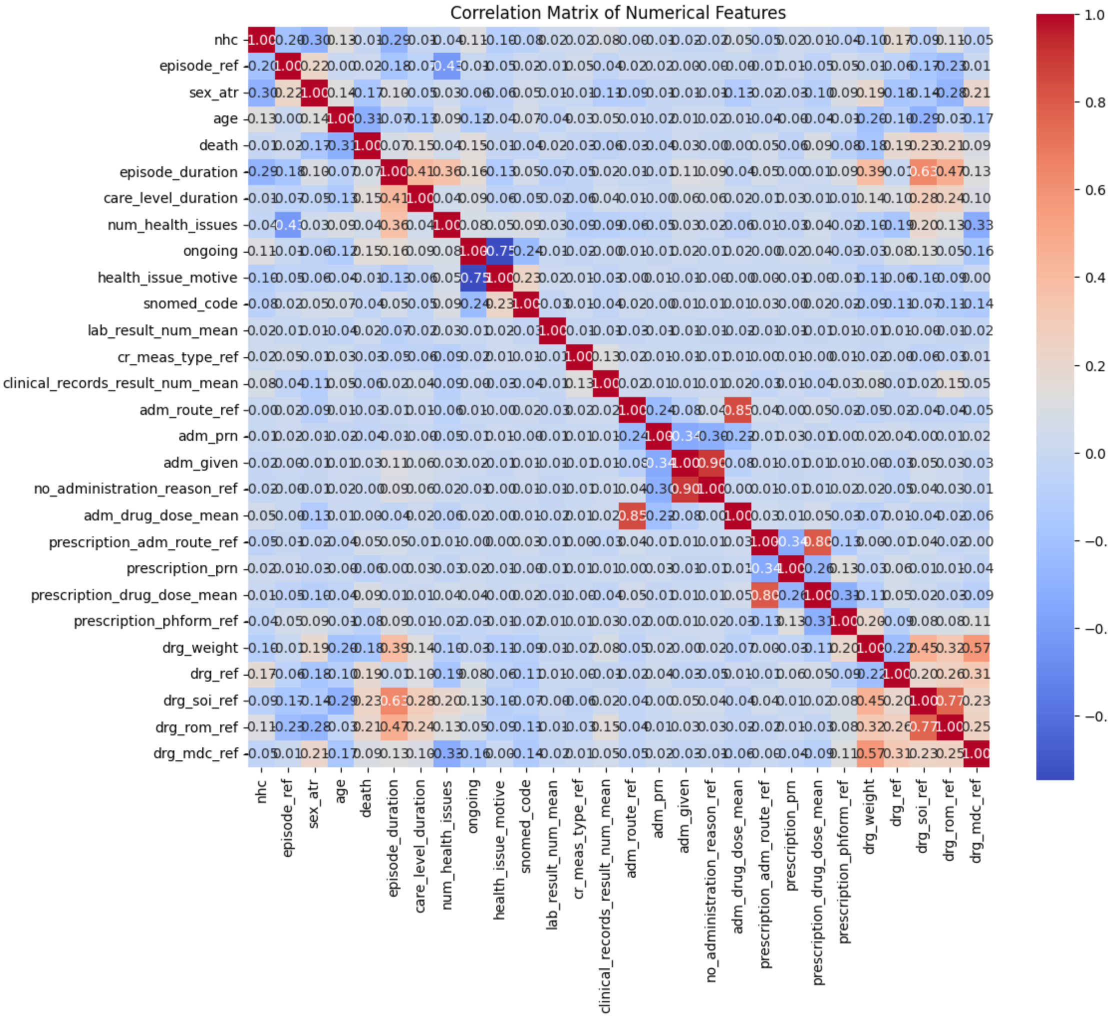


Figure B.5: *Correlation matrix of the numerical features of the dataset.*

ANNEX C. Feature importance ranking plots

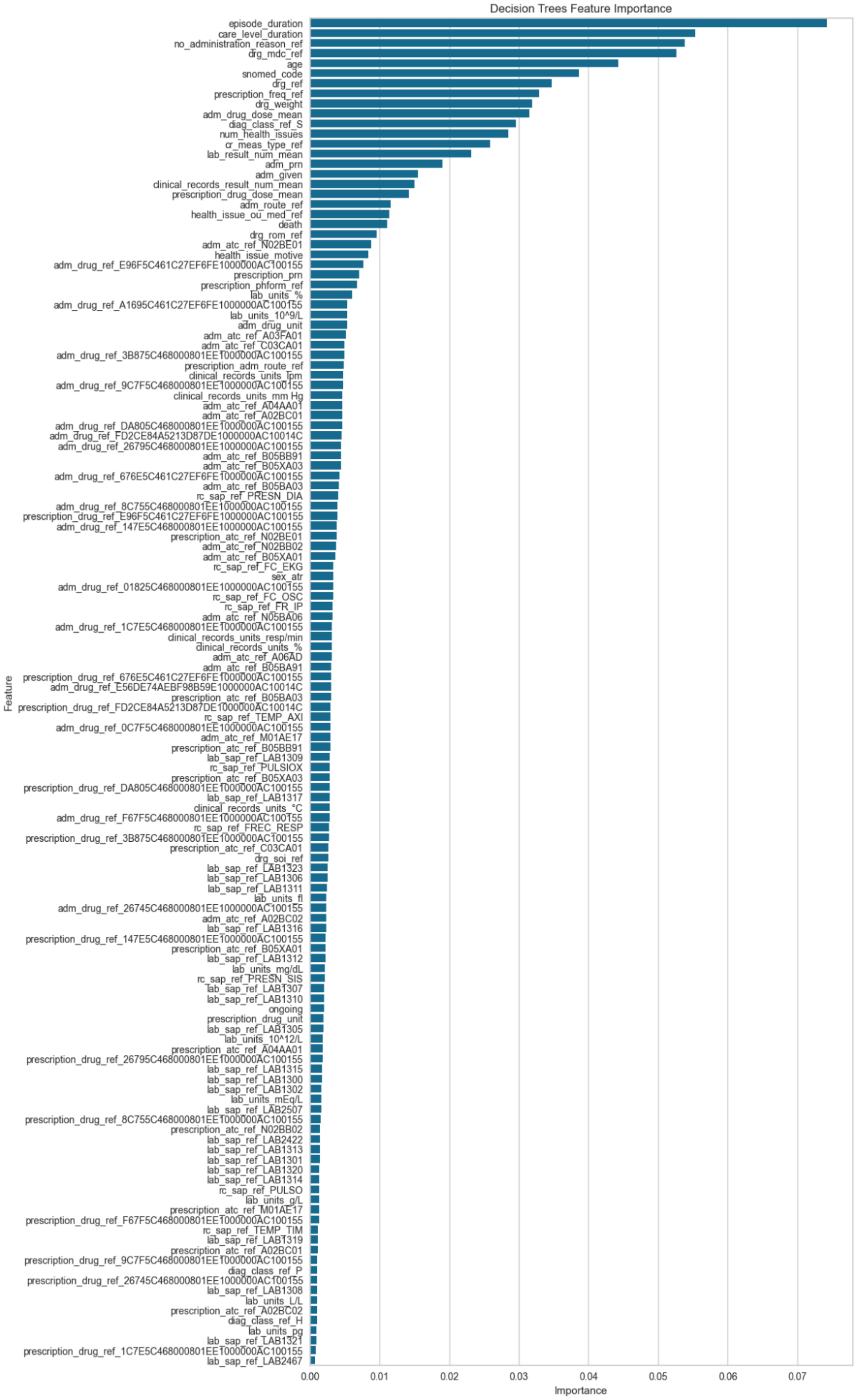


Figure C.1: *Rankig of feature importance by Decision Trees.*

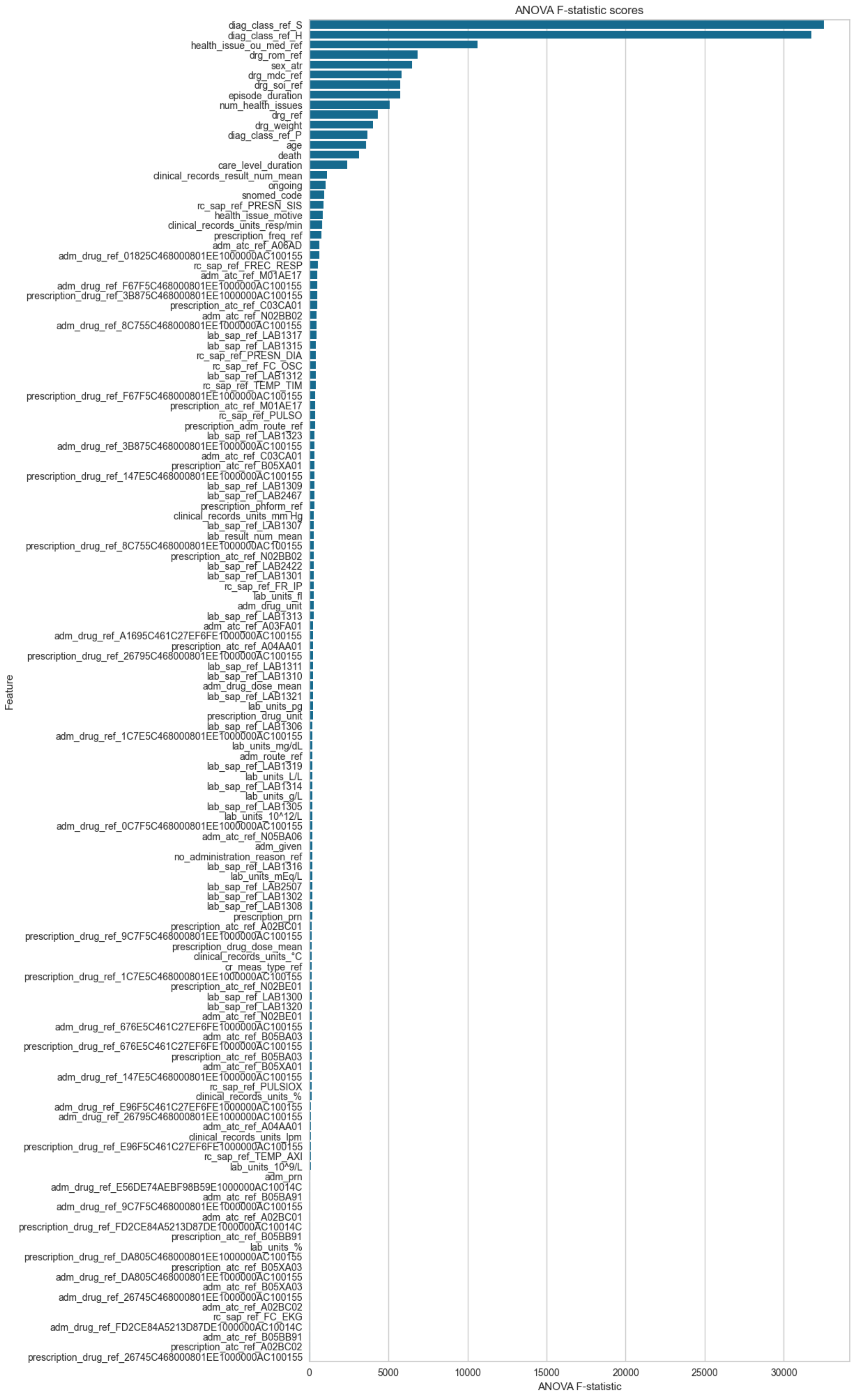
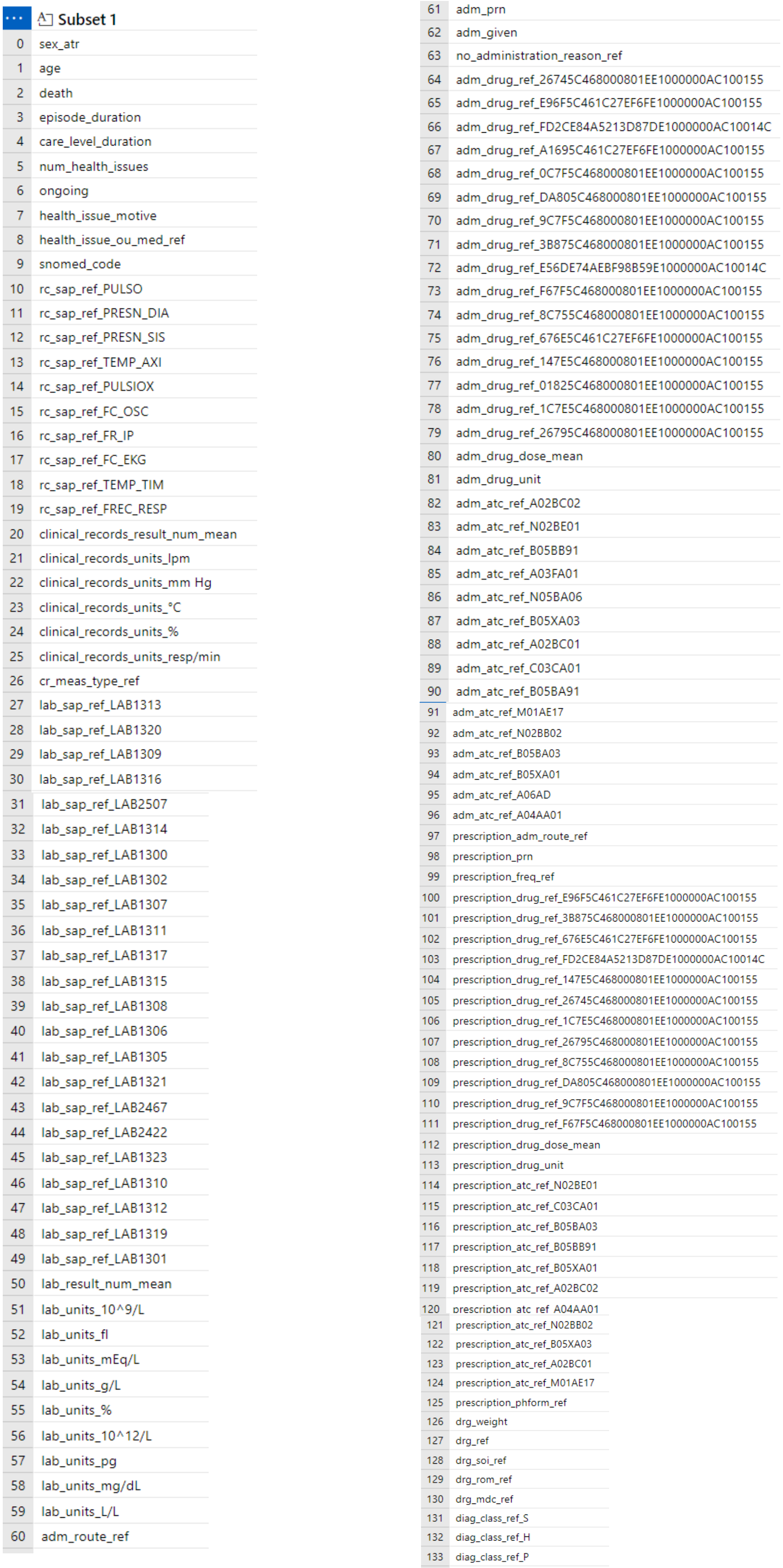


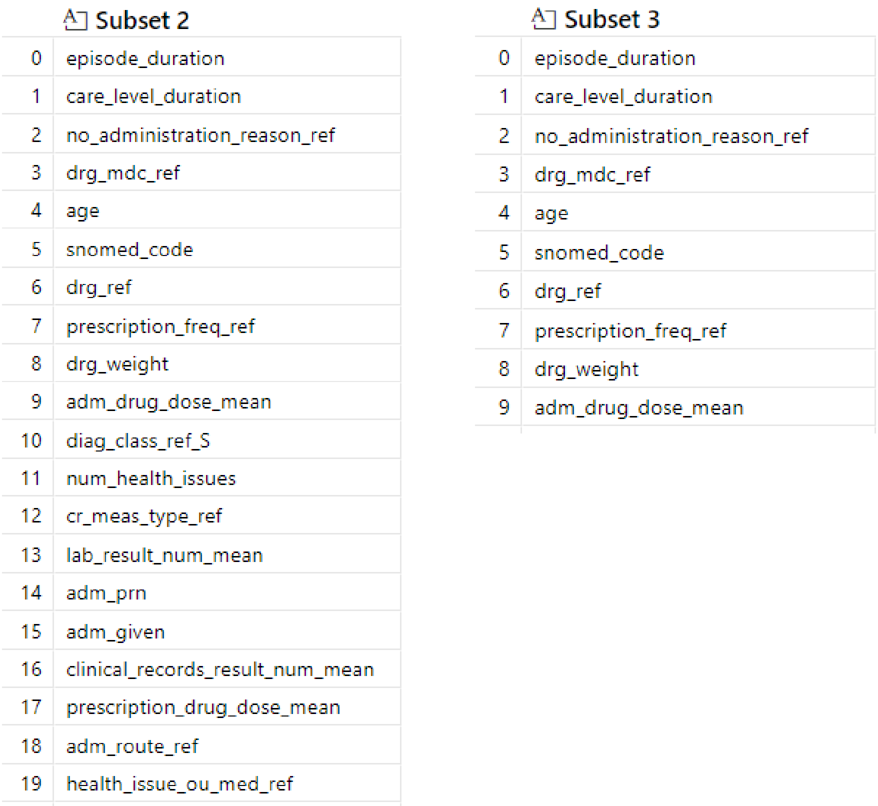
Figure C.2: *Ranking of feature importance by ANOVA F-test.*

ANNEX D. List of variables for each subset



···	Subset 1
0	sex_atr
1	age
2	death
3	episode_duration
4	care_level_duration
5	num_health_issues
6	ongoing
7	health_issue_motive
8	health_issue_ou_med_ref
9	snomed_code
10	rc_sap_ref_PULSO
11	rc_sap_ref_PRESN_DIA
12	rc_sap_ref_PRESN_SIS
13	rc_sap_ref_TEMP_AXI
14	rc_sap_ref_PULSIOX
15	rc_sap_ref_FC_OSC
16	rc_sap_ref_FR_IP
17	rc_sap_ref_FC_EKG
18	rc_sap_ref_TEMP_TIM
19	rc_sap_ref_FREC_RESP
20	clinical_records_result_num_mean
21	clinical_records_units_lpm
22	clinical_records_units_mm Hg
23	clinical_records_units_°C
24	clinical_records_units_%
25	clinical_records_units_resp/min
26	cr_meas_type_ref
27	lab_sap_ref_LAB1313
28	lab_sap_ref_LAB1320
29	lab_sap_ref_LAB1309
30	lab_sap_ref_LAB1316
31	lab_sap_ref_LAB2507
32	lab_sap_ref_LAB1314
33	lab_sap_ref_LAB1300
34	lab_sap_ref_LAB1302
35	lab_sap_ref_LAB1307
36	lab_sap_ref_LAB1311
37	lab_sap_ref_LAB1317
38	lab_sap_ref_LAB1315
39	lab_sap_ref_LAB1308
40	lab_sap_ref_LAB1306
41	lab_sap_ref_LAB1305
42	lab_sap_ref_LAB1321
43	lab_sap_ref_LAB2467
44	lab_sap_ref_LAB2422
45	lab_sap_ref_LAB1323
46	lab_sap_ref_LAB1310
47	lab_sap_ref_LAB1312
48	lab_sap_ref_LAB1319
49	lab_sap_ref_LAB1301
50	lab_result_num_mean
51	lab_units_10^9/L
52	lab_units_fl
53	lab_units_mEq/L
54	lab_units_g/L
55	lab_units_%
56	lab_units_10^12/L
57	lab_units_pg
58	lab_units_mg/dL
59	lab_units_L/L
60	adm_route_ref
61	adm_prn
62	adm_given
63	no_administration_reason_ref
64	adm_drug_ref_26745C468000801EE1000000AC100155
65	adm_drug_ref_E96F5C461C27EF6FE1000000AC100155
66	adm_drug_ref_FD2CE84A5213D87DE1000000AC10014C
67	adm_drug_ref_A1695C461C27EF6FE1000000AC100155
68	adm_drug_ref_0C7F5C468000801EE1000000AC100155
69	adm_drug_ref_DA805C468000801EE1000000AC100155
70	adm_drug_ref_9C7F5C468000801EE1000000AC100155
71	adm_drug_ref_3B875C468000801EE1000000AC100155
72	adm_drug_ref_E56DE74AEBF98B59E1000000AC10014C
73	adm_drug_ref_F67F5C468000801EE1000000AC100155
74	adm_drug_ref_8C755C468000801EE1000000AC100155
75	adm_drug_ref_676E5C461C27EF6FE1000000AC100155
76	adm_drug_ref_147E5C468000801EE1000000AC100155
77	adm_drug_ref_01825C468000801EE1000000AC100155
78	adm_drug_ref_1C7E5C468000801EE1000000AC100155
79	adm_drug_ref_26795C468000801EE1000000AC100155
80	adm_drug_dose_mean
81	adm_drug_unit
82	adm_atc_ref_A02BC02
83	adm_atc_ref_N02BE01
84	adm_atc_ref_B05BB91
85	adm_atc_ref_A03FA01
86	adm_atc_ref_N05BA06
87	adm_atc_ref_B05XA03
88	adm_atc_ref_A02BC01
89	adm_atc_ref_C03CA01
90	adm_atc_ref_B05BA91
91	adm_atc_ref_M01AE17
92	adm_atc_ref_N02BB02
93	adm_atc_ref_B05BA03
94	adm_atc_ref_B05XA01
95	adm_atc_ref_A06AD
96	adm_atc_ref_A04AA01
97	prescription_adm_route_ref
98	prescription_prn
99	prescription_freq_ref
100	prescription_drug_ref_E96F5C461C27EF6FE1000000AC100155
101	prescription_drug_ref_3B875C468000801EE1000000AC100155
102	prescription_drug_ref_676E5C461C27EF6FE1000000AC100155
103	prescription_drug_ref_FD2CE84A5213D87DE1000000AC10014C
104	prescription_drug_ref_147E5C468000801EE1000000AC100155
105	prescription_drug_ref_26745C468000801EE1000000AC100155
106	prescription_drug_ref_1C7E5C468000801EE1000000AC100155
107	prescription_drug_ref_26795C468000801EE1000000AC100155
108	prescription_drug_ref_8C755C468000801EE1000000AC100155
109	prescription_drug_ref_DA805C468000801EE1000000AC100155
110	prescription_drug_ref_9C7F5C468000801EE1000000AC100155
111	prescription_drug_ref_F67F5C468000801EE1000000AC100155
112	prescription_drug_dose_mean
113	prescription_drug_unit
114	prescription_atc_ref_N02BE01
115	prescription_atc_ref_C03CA01
116	prescription_atc_ref_B05BA03
117	prescription_atc_ref_B05BB91
118	prescription_atc_ref_B05XA01
119	prescription_atc_ref_A02BC02
120	prescription atc ref A04AA01
121	prescription_atc_ref_N02BB02
122	prescription_atc_ref_B05XA03
123	prescription_atc_ref_A02BC01
124	prescription_atc_ref_M01AE17
125	prescription_phform_ref
126	drg_weight
127	drg_ref
128	drg_soi_ref
129	drg_rom_ref
130	drg_mdc_ref
131	diag_class_ref_S
132	diag_class_ref_H
133	diag_class_ref_P

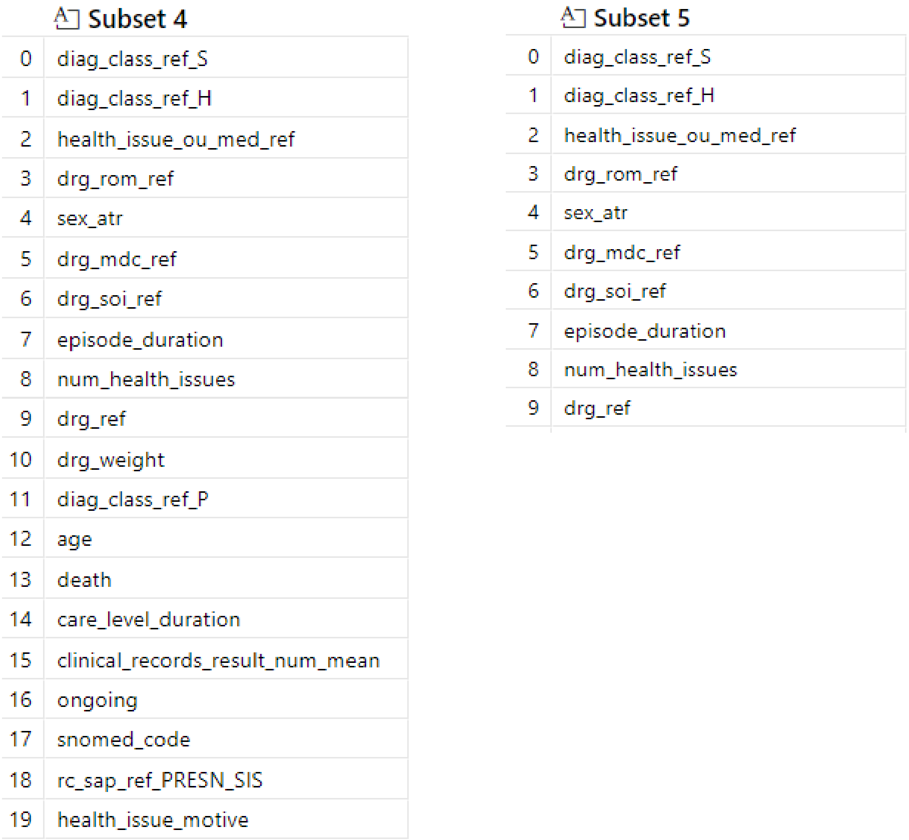
***Figure D.1:** List of variable names for Subset 1.*



	Subset 2
0	episode_duration
1	care_level_duration
2	no_administration_reason_ref
3	drg_mdc_ref
4	age
5	snomed_code
6	drg_ref
7	prescription_freq_ref
8	drg_weight
9	adm_drug_dose_mean
10	diag_class_ref_S
11	num_health_issues
12	cr_meas_type_ref
13	lab_result_num_mean
14	adm_prn
15	adm_given
16	clinical_records_result_num_mean
17	prescription_drug_dose_mean
18	adm_route_ref
19	health_issue_ou_med_ref

	Subset 3
0	episode_duration
1	care_level_duration
2	no_administration_reason_ref
3	drg_mdc_ref
4	age
5	snomed_code
6	drg_ref
7	prescription_freq_ref
8	drg_weight
9	adm_drug_dose_mean

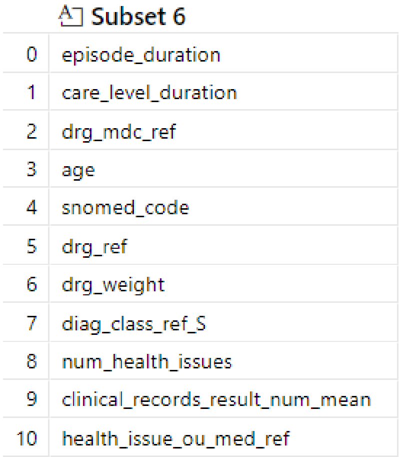
Figure D.2: *List of variable names for Subset 2 (left) and Subset 3 (right).*



	Subset 4
0	diag_class_ref_S
1	diag_class_ref_H
2	health_issue_ou_med_ref
3	drg_rom_ref
4	sex_atr
5	drg_mdc_ref
6	drg_soi_ref
7	episode_duration
8	num_health_issues
9	drg_ref
10	drg_weight
11	diag_class_ref_P
12	age
13	death
14	care_level_duration
15	clinical_records_result_num_mean
16	ongoing
17	snomed_code
18	rc_sap_ref_PRESN_SIS
19	health_issue_motive

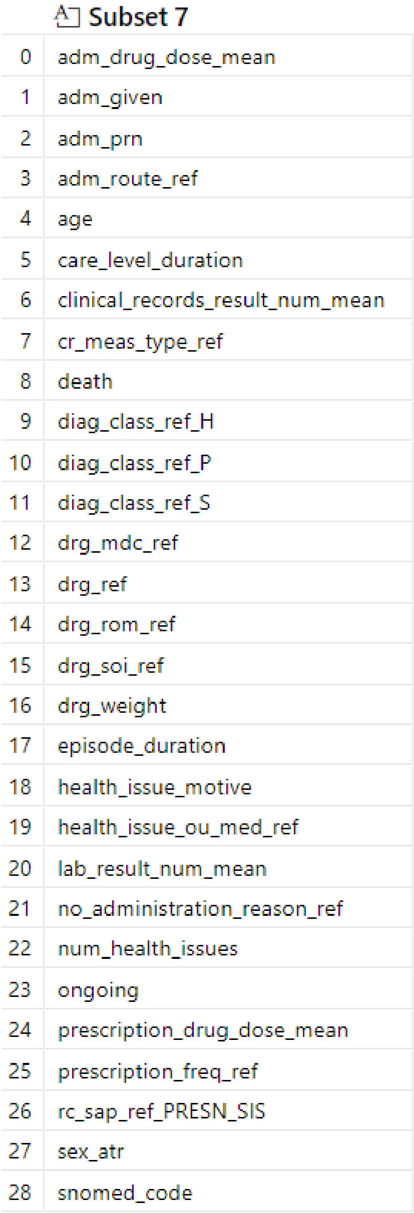
	Subset 5
0	diag_class_ref_S
1	diag_class_ref_H
2	health_issue_ou_med_ref
3	drg_rom_ref
4	sex_atr
5	drg_mdc_ref
6	drg_soi_ref
7	episode_duration
8	num_health_issues
9	drg_ref

Figure D.3: *List of variable names for Subset 4 (left) and Subset 5 (right).*



	Subset 6
0	episode_duration
1	care_level_duration
2	drg_mdc_ref
3	age
4	snomed_code
5	drg_ref
6	drg_weight
7	diag_class_ref_S
8	num_health_issues
9	clinical_records_result_num_mean
10	health_issue_ou_med_ref

Figure D.4: *List of variable names for Subset 6.*



	Subset 7
0	adm_drug_dose_mean
1	adm_given
2	adm_prn
3	adm_route_ref
4	age
5	care_level_duration
6	clinical_records_result_num_mean
7	cr_meas_type_ref
8	death
9	diag_class_ref_H
10	diag_class_ref_P
11	diag_class_ref_S
12	drg_mdc_ref
13	drg_ref
14	drg_rom_ref
15	drg_soi_ref
16	drg_weight
17	episode_duration
18	health_issue_motive
19	health_issue_ou_med_ref
20	lab_result_num_mean
21	no_administration_reason_ref
22	num_health_issues
23	ongoing
24	prescription_drug_dose_mean
25	prescription_freq_ref
26	rc_sap_ref_PRESN_SIS
27	sex_atr
28	snomed_code

Figure D.5: *List of variable names for Subset 7.*

ANNEX E. Performance plots across all subsets

Subset 1

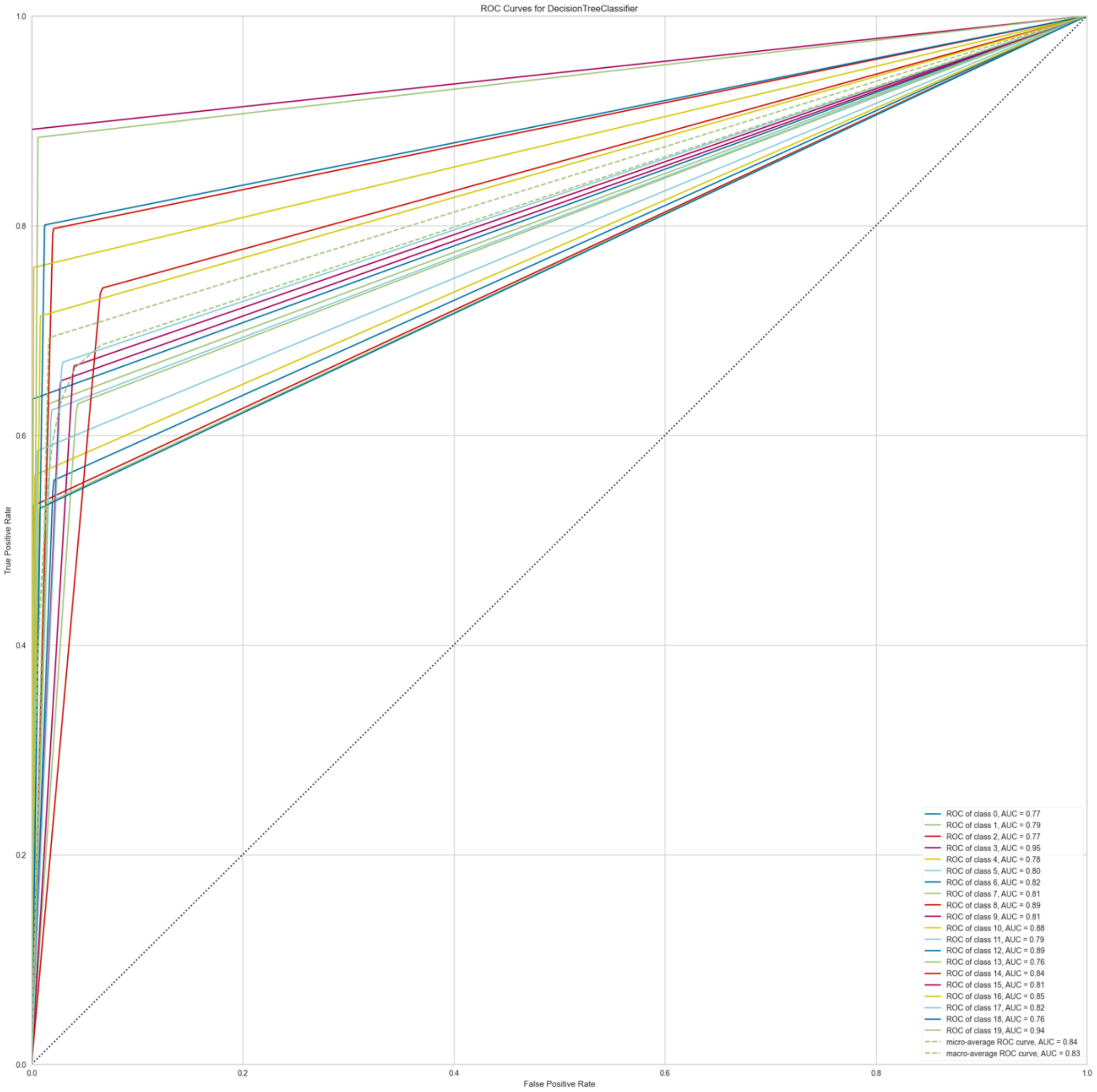


Figure E.1: *AUC plot for Subset 1.*

Subset 2

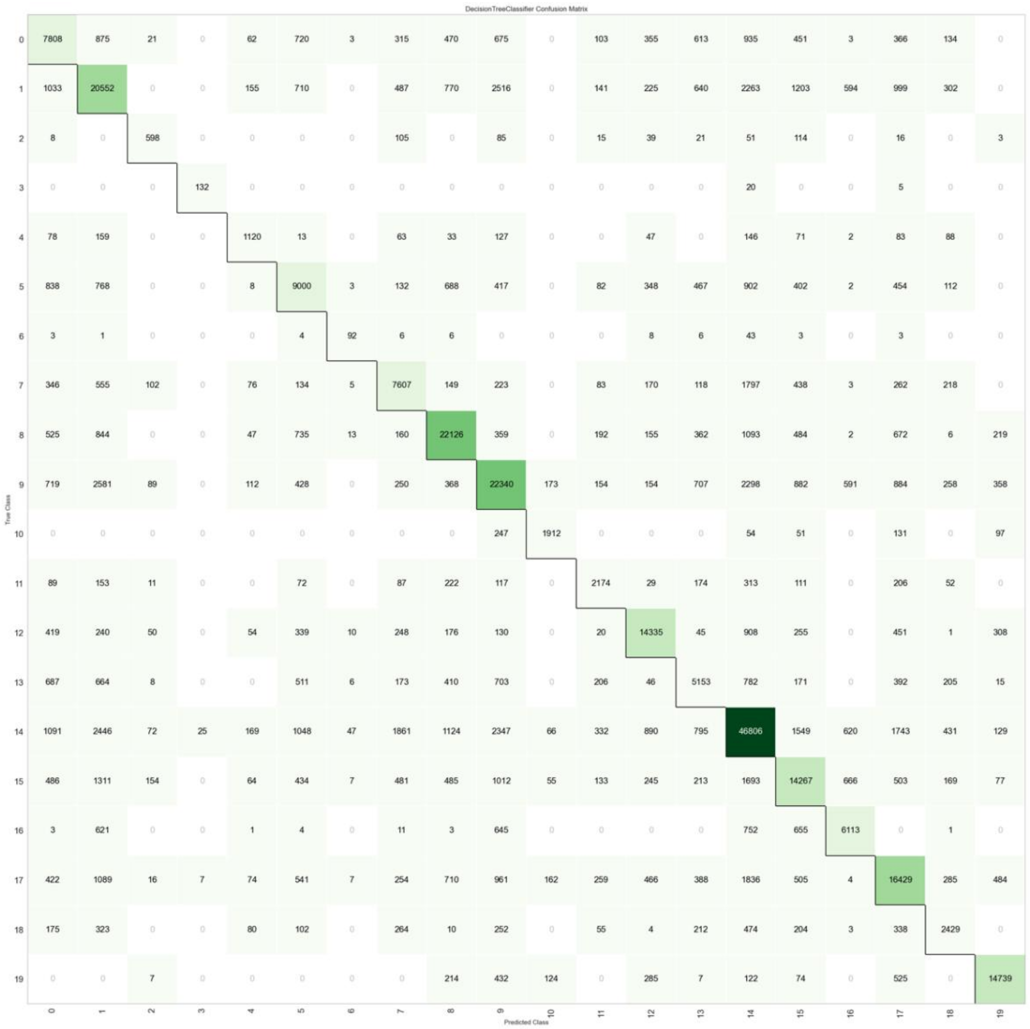


Figure E.2: *Confusion matrix for Subset 2.*

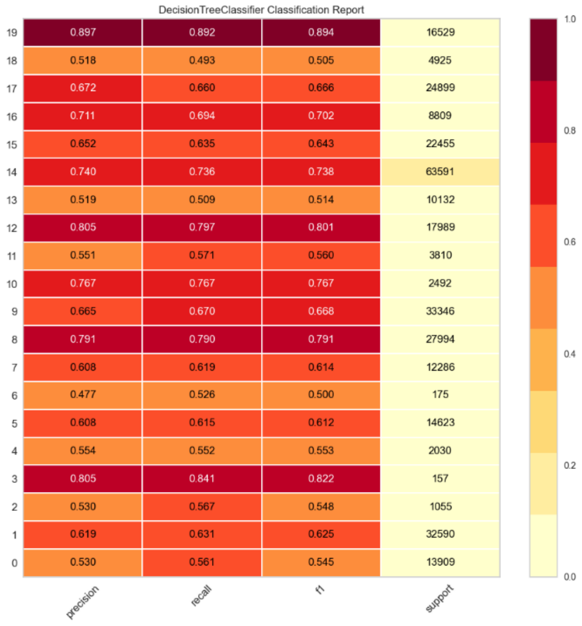


Figure E.3: *Classification report for Subset 2.*

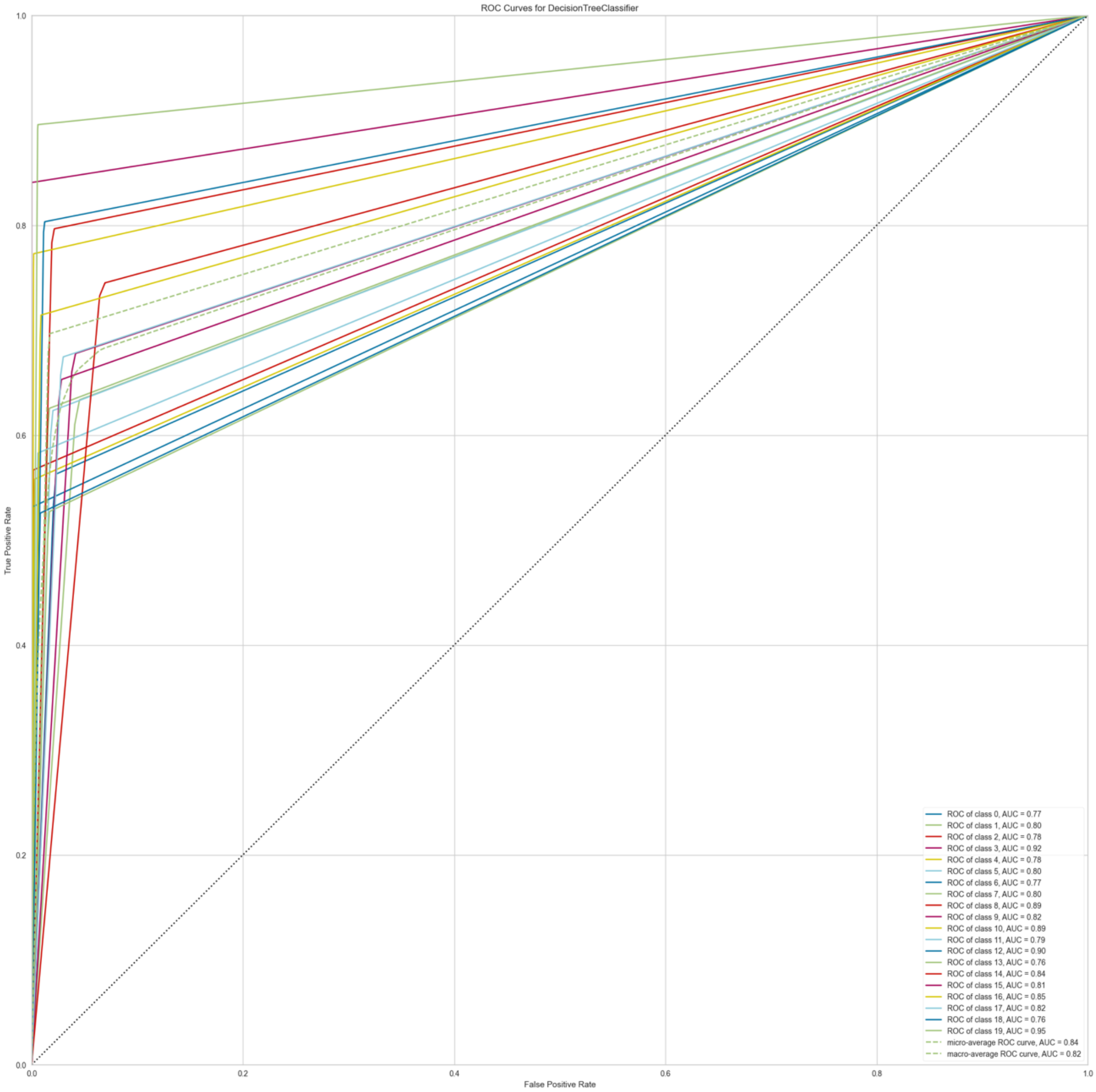
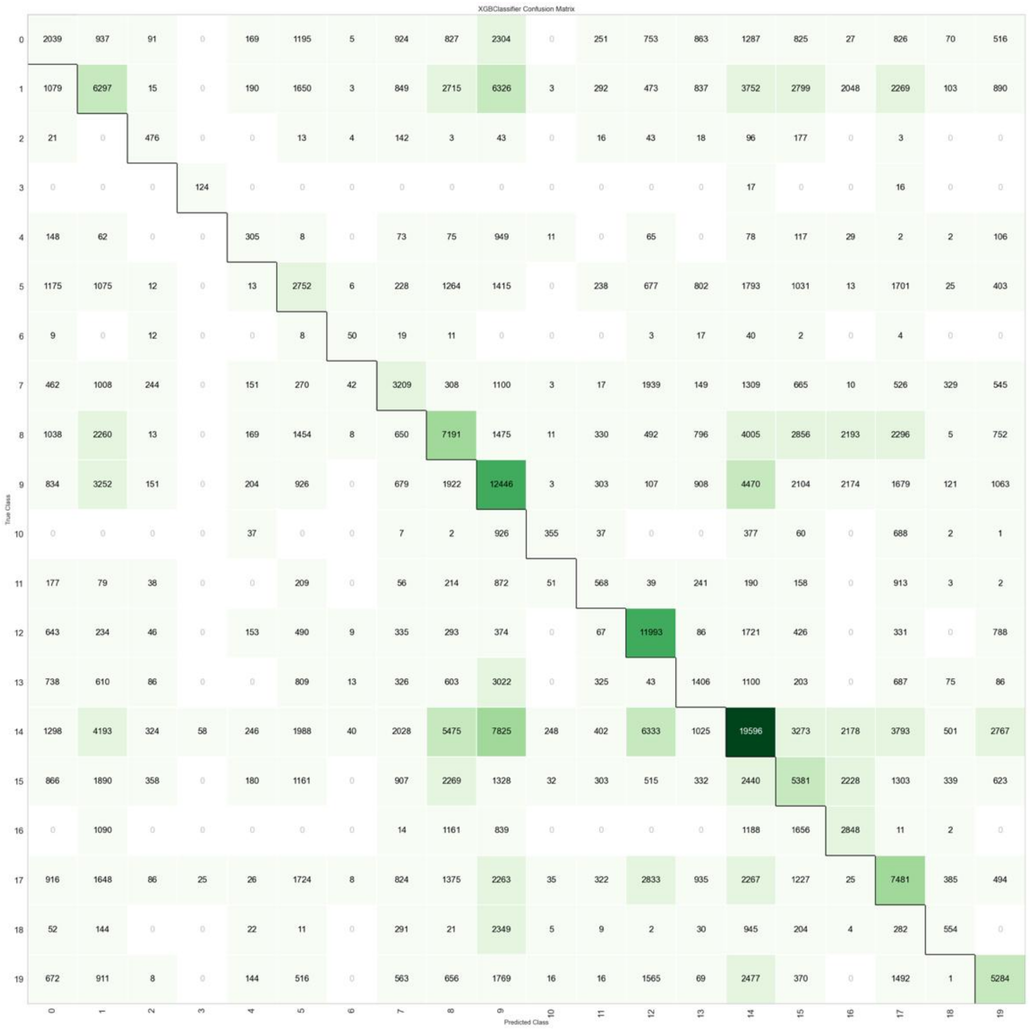


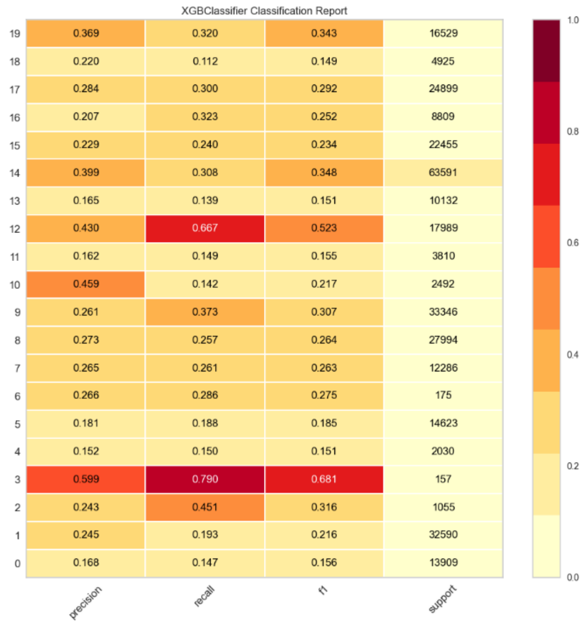
Figure E.4: *AUC plot for Subset 2.*

Subset 3



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	2039	937	91	0	169	1195	5	924	827	2304	0	251	753	863	1287	825	27	826	70	516
1	1079	6297	15	0	190	1650	3	849	2715	6326	3	292	473	837	3752	2799	2048	2269	103	890
2	21	0	476	0	0	13	4	142	3	43	0	16	43	18	96	177	0	3	0	0
3	0	0	0	124	0	0	0	0	0	0	0	0	0	0	17	0	0	16	0	0
4	148	62	0	0	305	8	0	73	75	949	11	0	65	0	78	117	29	2	2	106
5	1175	1075	12	0	13	2752	6	228	1264	1415	0	238	677	802	1793	1031	13	1701	25	403
6	9	0	12	0	0	8	50	19	11	0	0	0	3	17	40	2	0	4	0	0
7	462	1008	244	0	151	270	42	3209	308	1100	3	17	1939	149	1309	665	10	526	329	545
8	1038	2260	13	0	169	1454	8	650	7191	1475	11	330	492	796	4005	2856	2193	2296	5	752
9	834	3252	151	0	204	926	0	679	1922	12446	3	303	107	908	4470	2104	2174	1679	121	1063
10	0	0	0	0	37	0	0	7	2	926	355	37	0	0	377	60	0	688	2	1
11	177	79	38	0	0	209	0	56	214	872	51	568	39	241	190	158	0	913	3	2
12	643	234	46	0	153	490	9	335	293	374	0	67	11993	86	1721	426	0	331	0	788
13	738	610	86	0	0	809	13	326	603	3022	0	325	43	1406	1100	203	0	687	75	86
14	1298	4193	324	58	246	1988	40	2028	5475	7825	248	402	6333	1025	19596	3273	2178	3793	501	2767
15	866	1890	358	0	180	1161	0	907	2269	1328	32	303	515	332	2440	5381	2228	1303	339	623
16	0	1090	0	0	0	0	0	14	1161	839	0	0	0	0	1188	1656	2848	11	2	0
17	916	1648	86	25	26	1724	8	824	1375	2263	35	322	2833	935	2267	1227	25	7481	385	494
18	52	144	0	0	22	11	0	291	21	2349	5	9	2	30	945	204	4	282	554	0
19	672	911	8	0	144	516	0	563	656	1769	16	16	1565	69	2477	370	0	1492	1	5284

Figure E.5: *Confusion matrix for Subset 3.*



	precision	recall	f1	support
19	0.369	0.320	0.343	16529
18	0.220	0.112	0.149	4925
17	0.284	0.300	0.292	24899
16	0.207	0.323	0.252	8809
15	0.229	0.240	0.234	22455
14	0.399	0.308	0.348	63591
13	0.165	0.139	0.151	10132
12	0.430	0.667	0.523	17989
11	0.162	0.149	0.155	3810
10	0.459	0.142	0.217	2492
9	0.261	0.373	0.307	33346
8	0.273	0.257	0.264	27994
7	0.265	0.261	0.263	12286
6	0.266	0.286	0.275	175
5	0.181	0.188	0.185	14623
4	0.152	0.150	0.151	2030
3	0.599	0.790	0.681	157
2	0.243	0.451	0.316	1055
1	0.245	0.193	0.216	32590
0	0.168	0.147	0.156	13909

Figure E.6: *Classification report for Subset 3.*

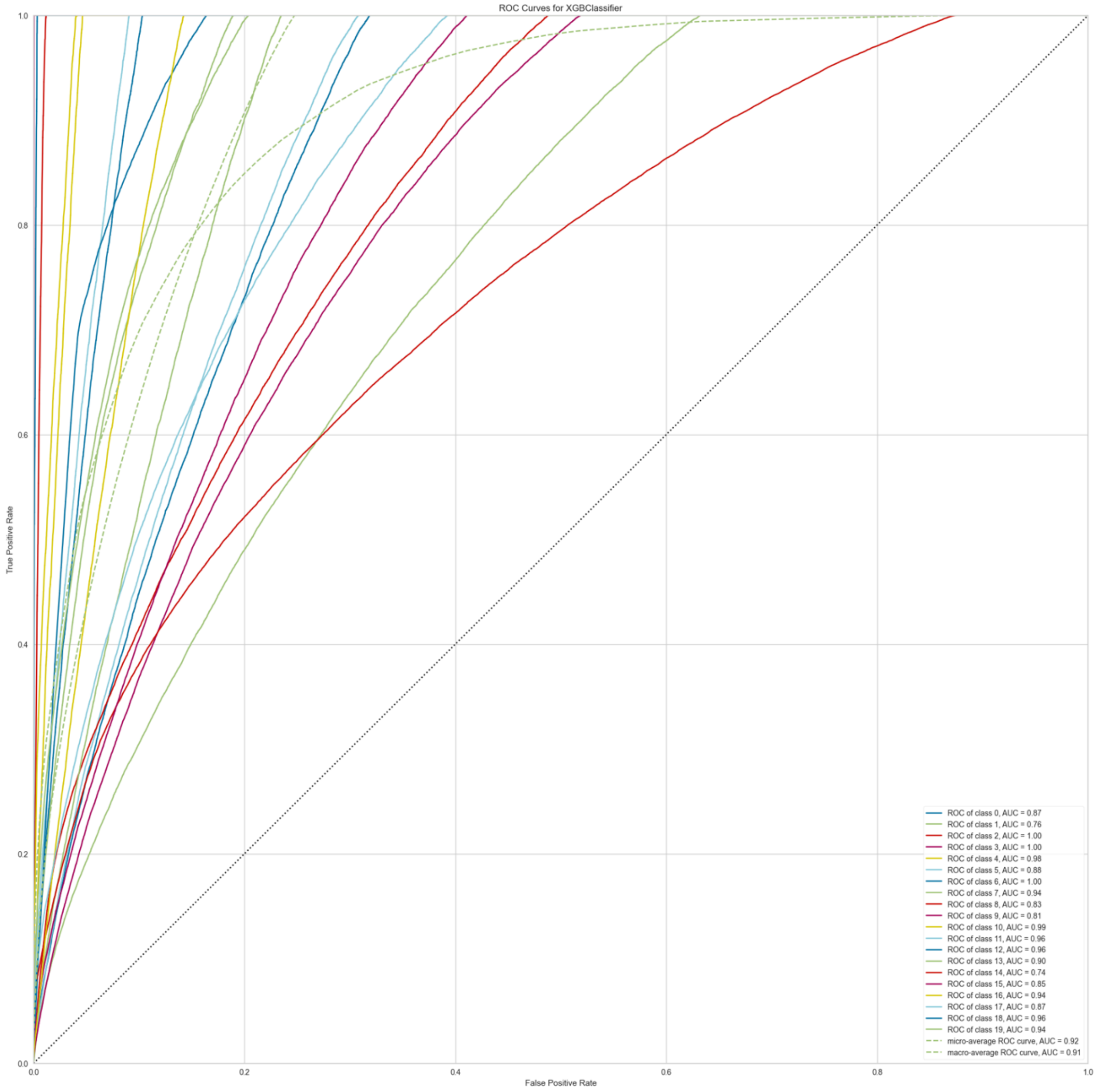


Figure E.7: *AUC plot for Subset 3.*

Subset 4

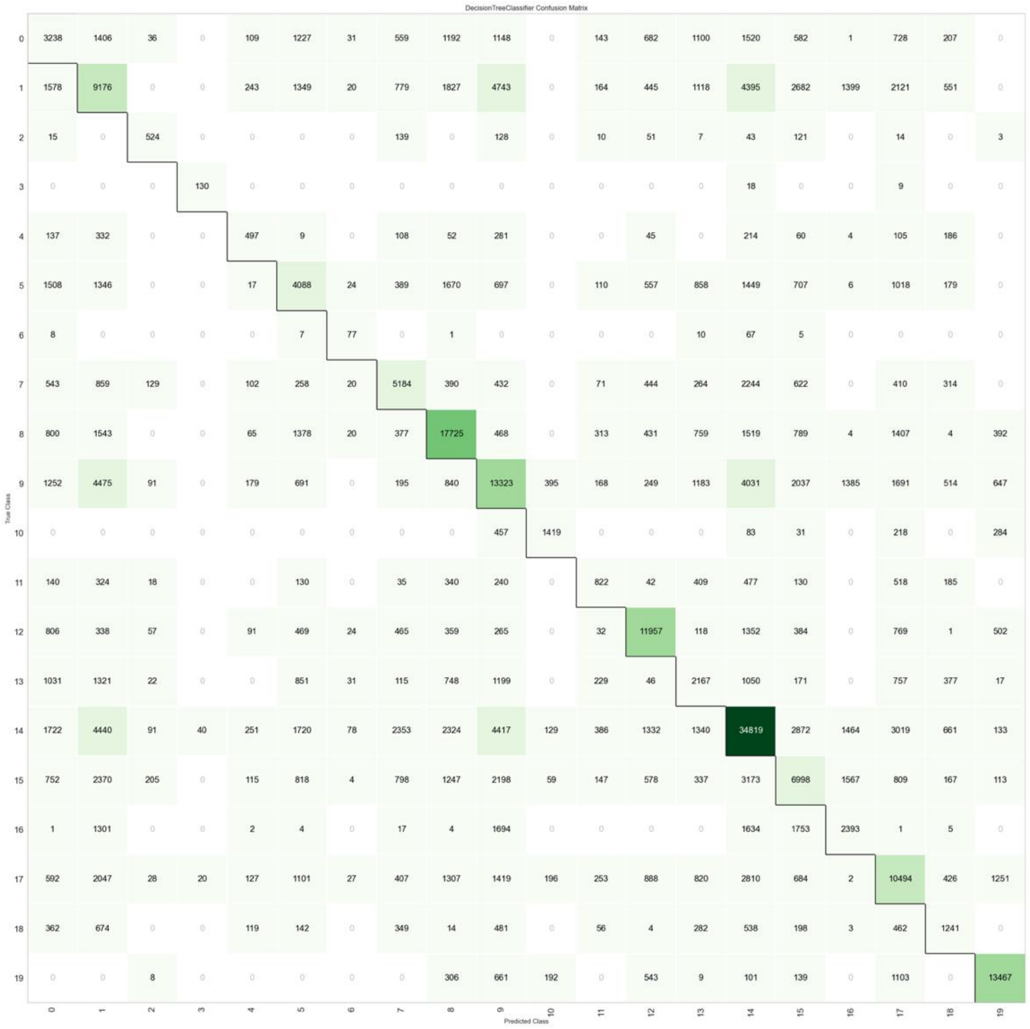


Figure E.8: *Confusion matrix for Subset 4.*

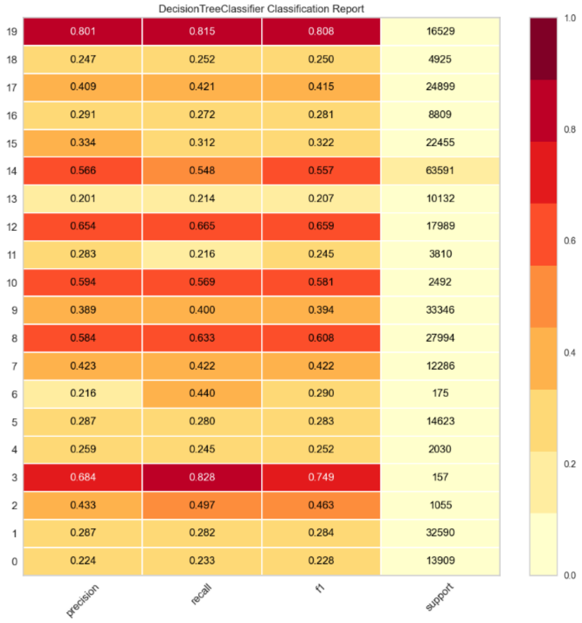


Figure E.9: *Classification report for Subset 4.*

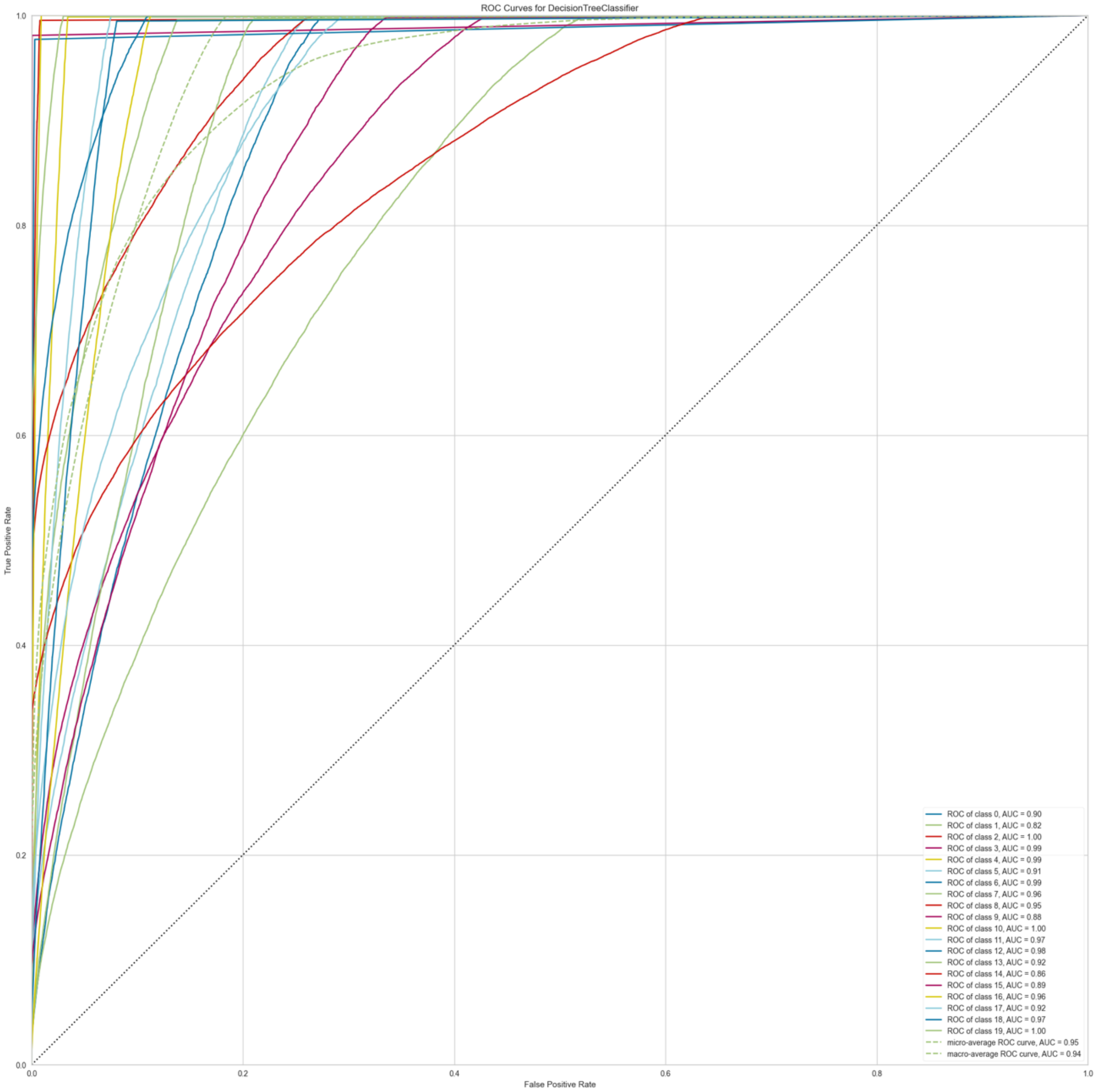


Figure E.10: *AUC plot for Subset 4.*

Subset 5

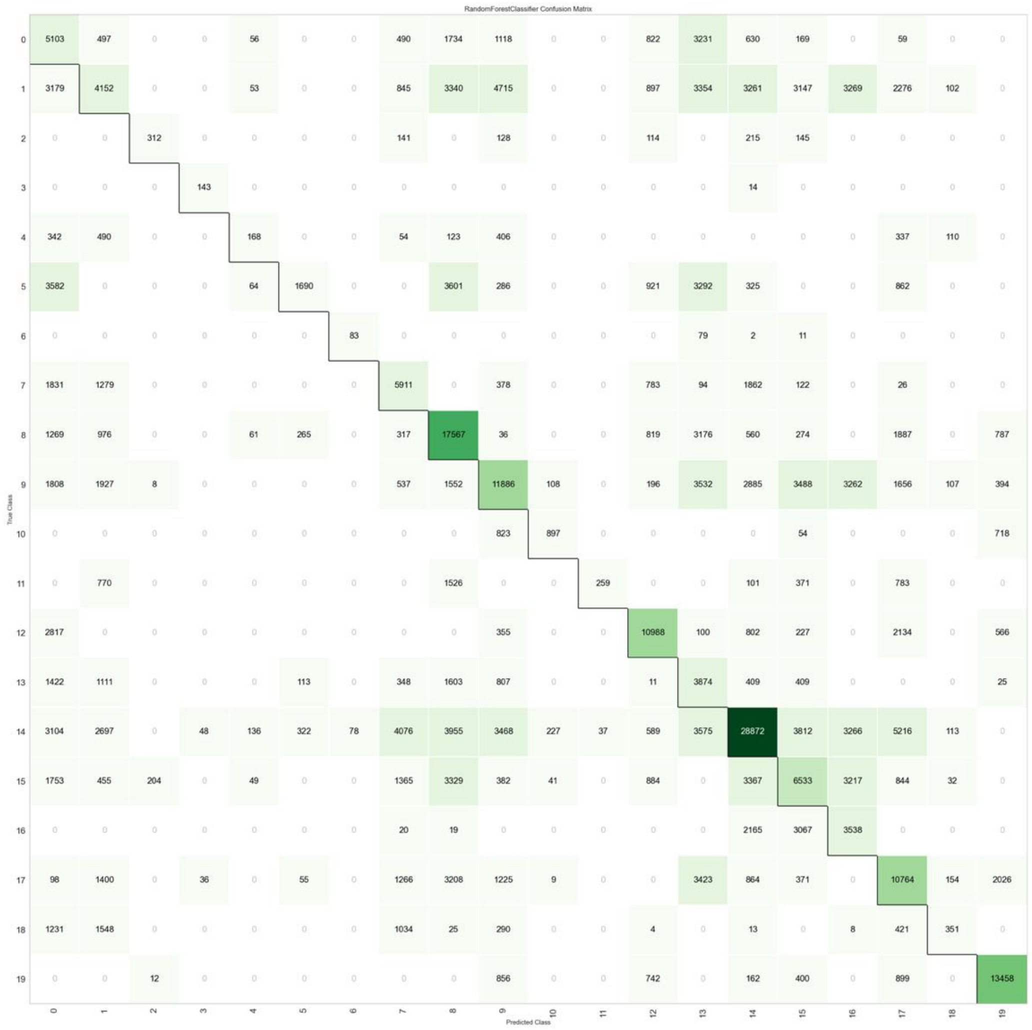


Figure E.11: *Confusion matrix for Subset 5.*

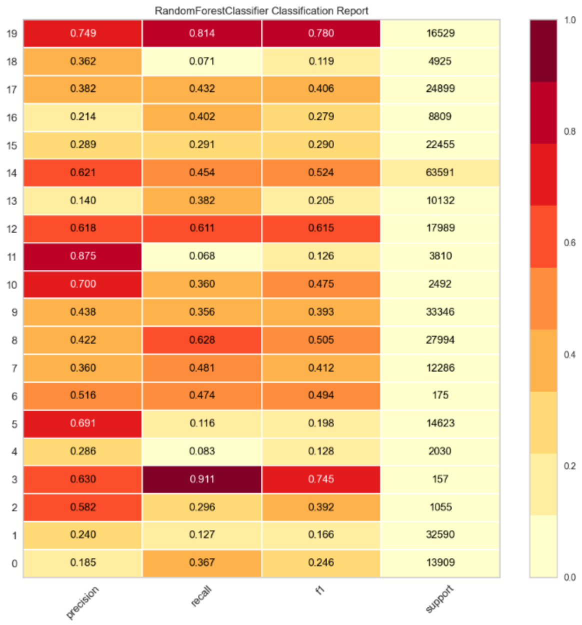


Figure E.12: *Classification report for Subset 5.*

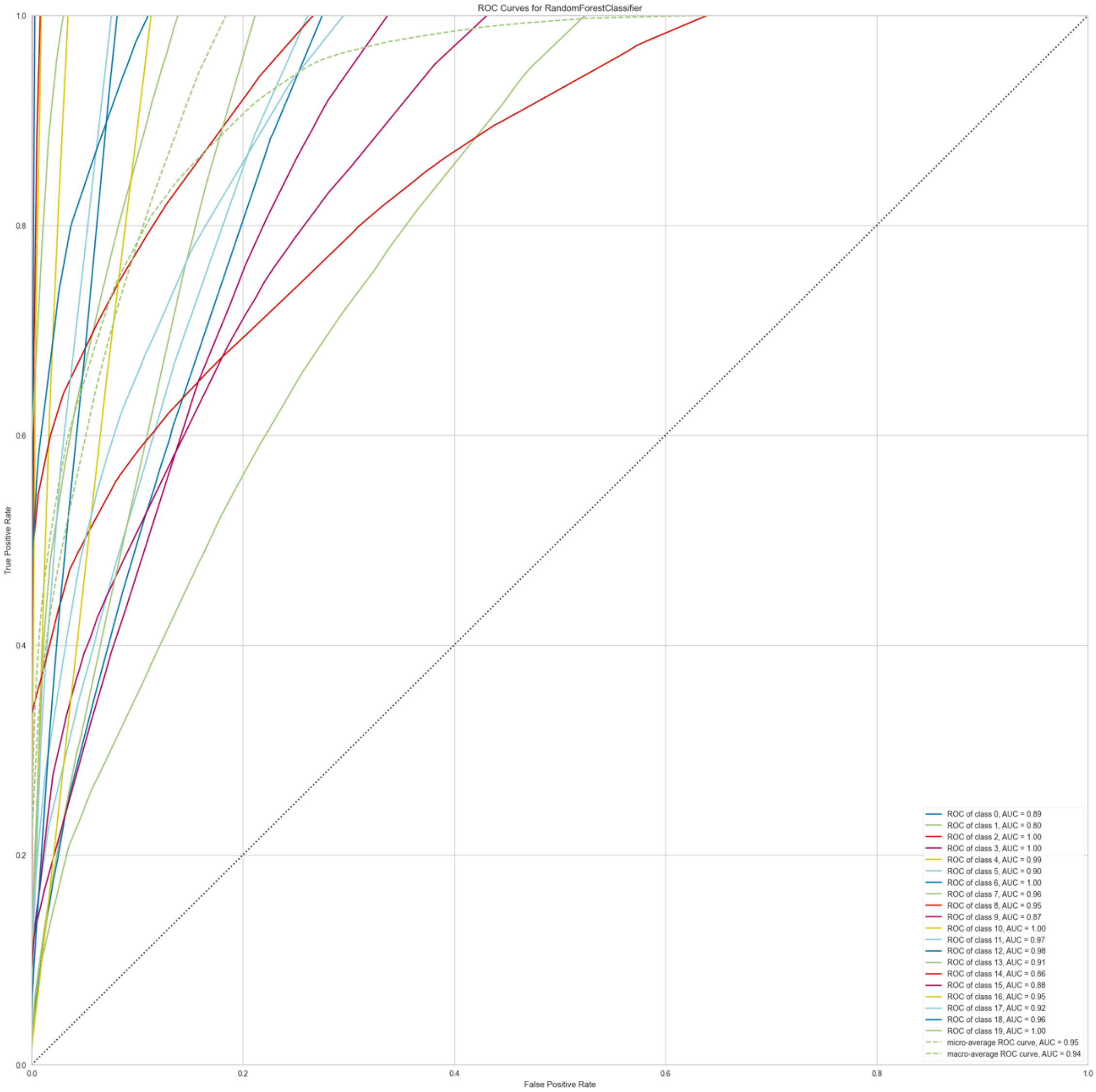


Figure E.13: *AUC plot for Subset 5.*

Subset 6

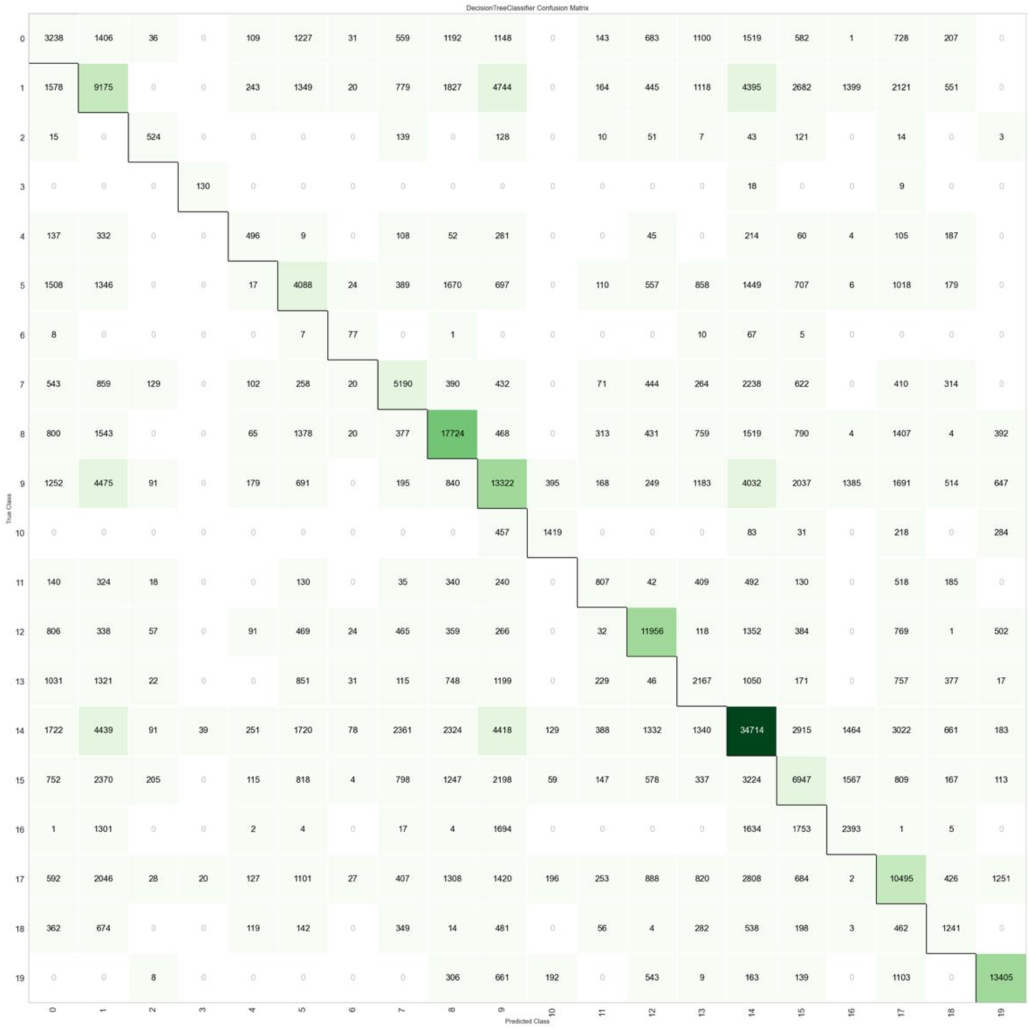


Figure E.14: *Confusion matrix for Subset 6.*

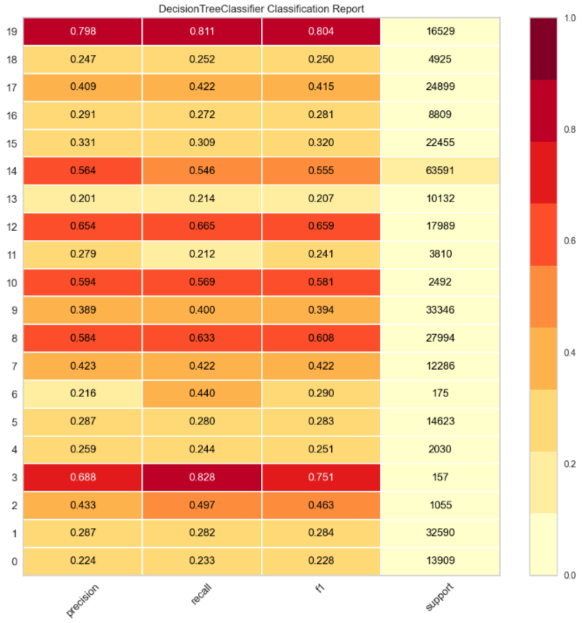


Figure E.15: *Classification report for Subset 6.*

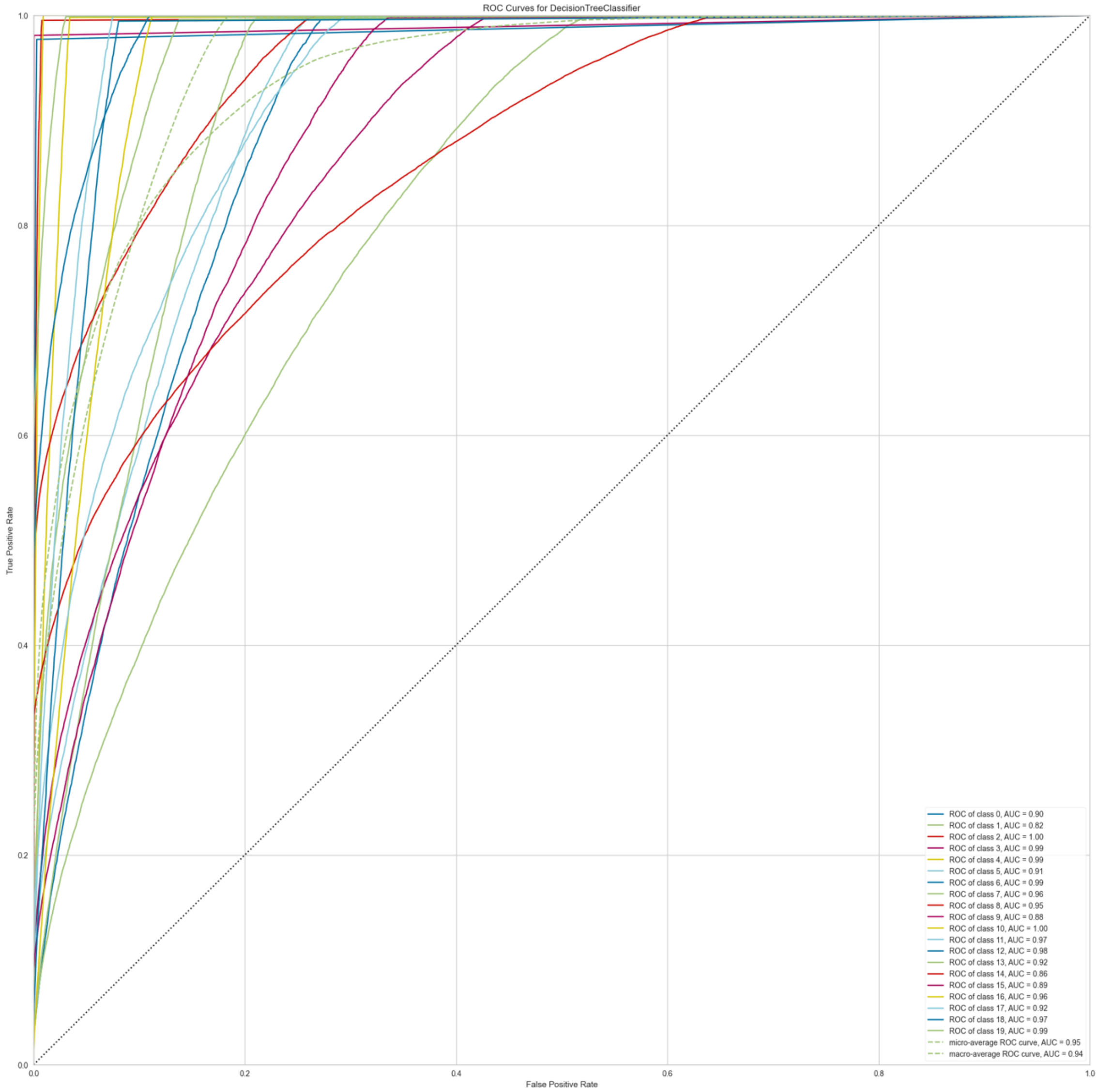


***Figure E.16:** AUC plot for Subset 6.*

Subset 7

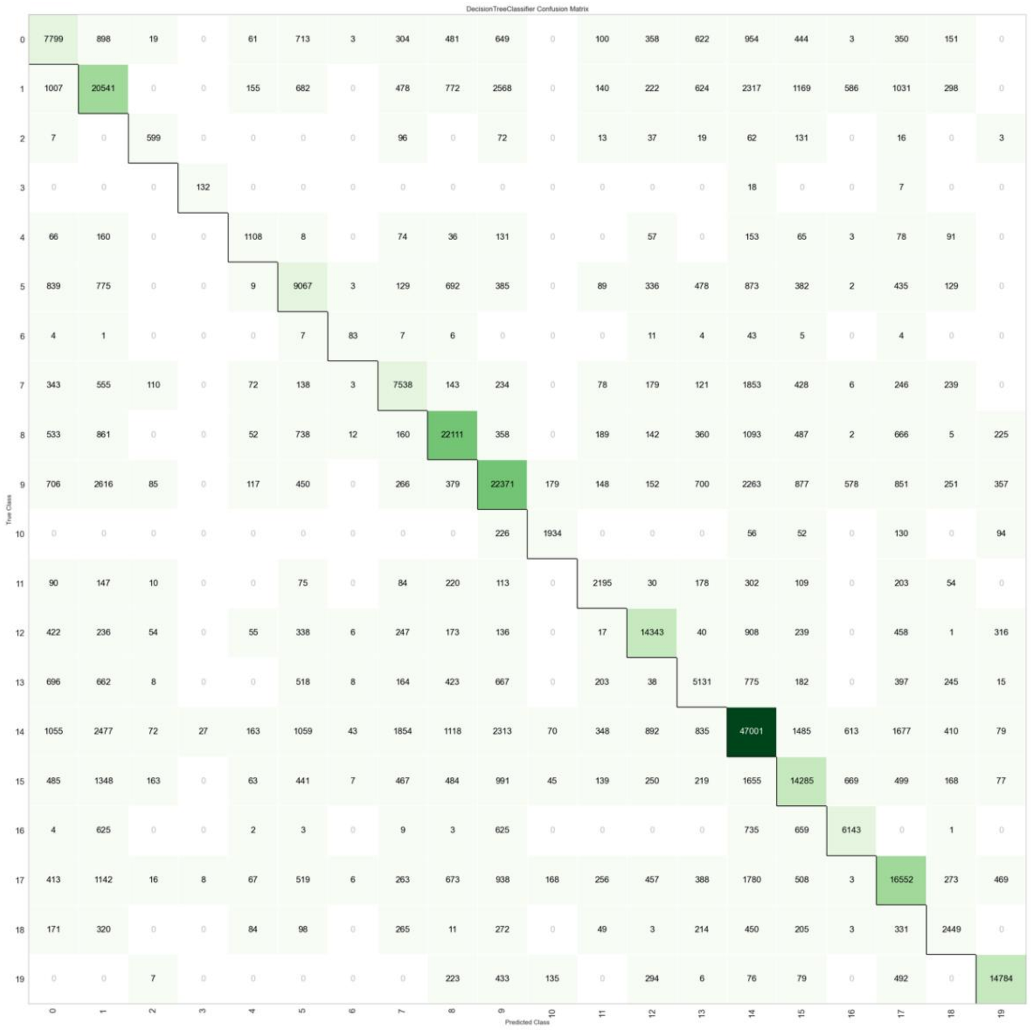


Figure E.17: *Confusion matrix for Subset 7.*

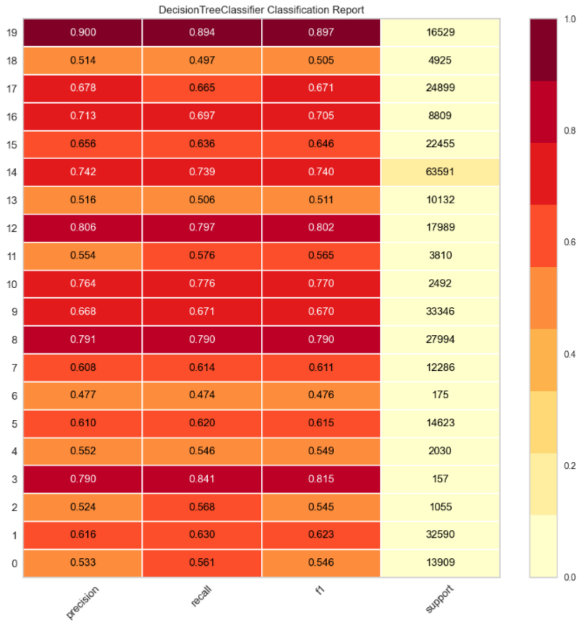


Figure E.18: *Classification report for Subset 7.*

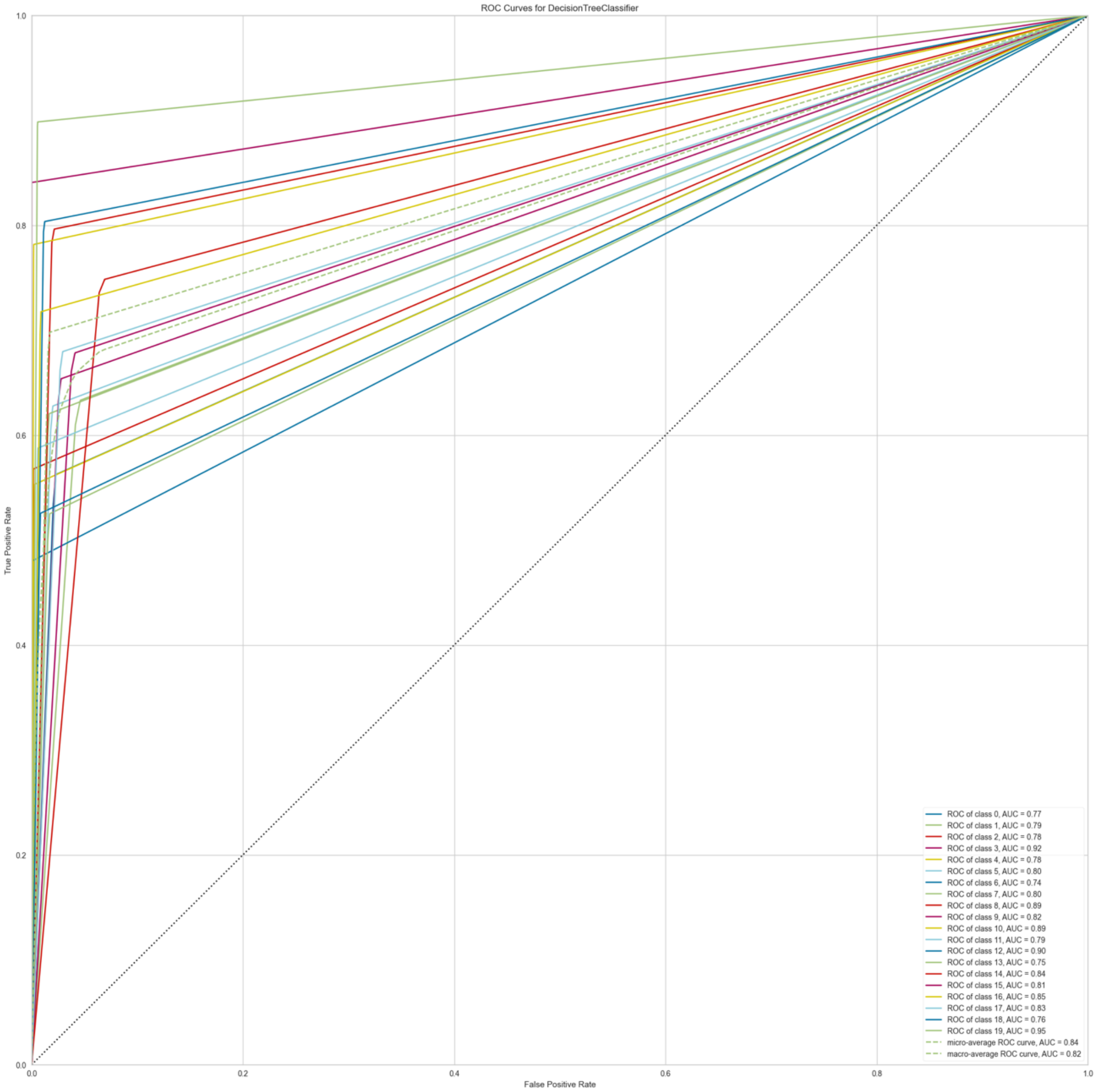


Figure E.19: *AUC plot for Subset 7.*