Technical University of Crete
School of Electrical and Computer Engineering

Diploma Thesis

The Use of Machine Learning Algorithms in Predicting Patient Outcomes with Heart Failure

Author:

Stavros Flourentzou

Thesis Committee:

Professor Sotiris Ioannidis (Supervisor)

Professor Michail G. Lagoudakis

Assistant Professor Vasiliki Danilatou (EUC)

Chania, July 2025

Πολυτεχνείο Κρήτης
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικων Υπολογιστών

Διπλωματική Εργασία

Η Χρήση Αλγορίθμων Μηχανικής Μάθησης στην Πρόβλεψη της Έκβασης Ασθενών με Καρδιακή Ανεπάρκεια

Συγγραφέας:
Σταύρος Φλουρέντζου

Εξεταστική επιτροπή:
Καθηγητής Σωτήριος Ιωαννίδης (Επιβλέπων)
Καθηγητής Μιχαήλ Γ. Λαγουδάκης
Επίκουρη Καθηγήτρια Βασιλική Δανηλάτου (ΕΠΚ)

Χανιά, Ιούλιος 2025

Abstract

The weakening of the heart muscle is a major cause of illness and death all over the world, so there is an immediate need to have precise predictive tools, which can enable doctors to make well-informed decisions and help them to use healthcare resources appropriately. Recent advancements in artificial intelligence and machine learning have improved the capability to predict outcomes in heart failure patients by utilizing numerous and high-resolution healthcare datasets. The research of this diploma thesis considers the application of different types of machine learning algorithms (Logistic Regression, Support Vector Machine, Random Forest, Extreme Gradient Boosting, Light Gradient Boosting Machine, Tabular Transformer) to precisely predict critical outcomes, including intensive care unit (ICU) admission and mortality within the first month of hospitalization. The useful data that these models are using were extracted from a well-known medical database, Medical Information Mart for Intensive Care IV (MIMIC-IV). Experimental results demonstrate that the proposed models effectively predict ICU admissions with high accuracy and robust performance metrics. However, the prediction of mortality within one month after hospitalization demonstrates limited effectiveness due to significant class imbalance, leading to suboptimal performance in area under the curve (AUC) and accuracy. Despite applying class balancing techniques, the model struggles to accurately identify minority class instances. These findings underscore the challenges of class imbalance in prediction problems and the need for more advanced resampling or algorithmic approaches to improve predictive accuracy. The proposed models are designed to improve prognostic accuracy and identify high-risk patients, ultimately contributing to personalized treatment strategies and better healthcare management.

Περίληψη

Η αποδυνάμωση του μυοκαρδίου αποτελεί κύρια αιτία ασθενειών και θανάτων σε όλο τον κόσμο, οπότε υπάρχει άμεση ανάγκη για ακριβή εργαλεία πρόβλεψης που θα επιτρέπουν στους γιατρούς να λαμβάνουν τεκμηριωμένες αποφάσεις οι οποίες θα τους βοηθήσουν να χρησιμοποιήσουν τους πόρους της υγειονομικής περίθαλψης με ορθό τρόπο. Οι πρόσφατες εξελίξεις στην τεχνητή νοημοσύνη και τη μηχανική μάθηση έχουν βελτιώσει την ικανότητα πρόβλεψης αποτελεσμάτων σε ασθενείς με καρδιακή ανεπάρκεια με τη χρήση πολυπληθών και υψηλής ανάλυσης δεδομένων υγειονομικής περίθαλψης. Η έρευνα της παρούσας διπλωματικής εργασίας εξετάζει την εφαρμογή διαφορετικών τύπων αλγορίθμων μηχανικής μάθησης (Logistic Regression, Support Vector Machine, Random Forest, Extreme Gradient Boosting, Light Gradient Boosting Machine, Tabular Transformer) για την ακριβή πρόβλεψη κρίσιμων αποτελεσμάτων, συμπεριλαμβανομένης της εισαγωγής στη μονάδα εντατικής θεραπείας και της θνησιμότητας εντός του πρώτου μήνα νοσηλείας. Τα χρήσιμα δεδομένα που χρησιμοποιούν αυτά τα μοντέλα εξήχθησαν από μια γνωστή ιατρική βάση δεδομένων, τη ιατρική βάση δεδομένων για μονάδα εντατικής θεραπείας IV (MIMIC-IV). Τα πειραματικά αποτελέσματα καταδεικνύουν ότι τα προτεινόμενα μοντέλα προβλέπουν αποτελεσματικά τις εισαγωγές σε μονάδα εντατικής θεραπείας με υψηλή ακρίβεια και ισχυρές μετρήσεις απόδοσης. Ωστόσο, η πρόβλεψη της θνησιμότητας εντός ενός μηνός μετά τη νοσηλεία δεν είναι τόσο αποτελεσματική εξαιτίας της σημαντικής ανισορροπίας των κλάσεων, με αποτέλεσμα μη βέλτιστες επιδόσεις όσον αφορά το εμβαδόν κάτω από την καμπύλη (AUC) και την ακρίβεια. Παρά την εφαρμογή τεχνικών εξισορρόπησης κλάσεων, τα μοντέλα δυσκολεύονται να εντοπίσουν με ακρίβεια περιπτώσεις κλάσεων μειονοτήτων. Αυτά τα ευρήματα υπογραμμίζουν τις προκλήσεις της ανισορροπίας κλάσεων σε προβλήματα πρόβλεψεις και την ανάγκη για πιο προηγμένες προσεγγίσεις αναδειγματοληψίας ή αλγοριθμικές προσεγγίσεις για τη βελτίωση της προγνωστικής ακρίβειας.

Acknowledgements

I would like to thank everyone who has supported me throughout my years at the Technical University of Crete. Primarily, I want to thank Prof. Sotiris Ioannidis for giving me the opportunity to work with such professionals in their respective fields, Prof. Vasiliki Danilatou and Dr. Despoina Antonakaki, who guided me through every challenge and helped me stay on the right path. I would also like to thank Prof. Michail G. Lagoudakis for helping me understand the fundamentals of machine learning and AI. Last, but not least, I want to thank my family and friends for standing by my side through the ups and downs, and for their unwavering support over the years.

Contents

List of Tables

List of Figures

Abbreviations

The following abbreviations are used in this thesis:

ICU	Intensive Care Unit
MIMIC-IV	Medical Information Mart for Intensive Care IV
AUC	Area Under the Curve
AUC-ROC	Area Under the Curve - Receiver Operating Characteristic
SVM	Support Vector Machine
XGBoost	Extreme Gradient Boosting
LightGBM	Light Gradient Boosting Machine
TabTransformer	Tabular Transformer
SOFA	Sequential Organ Failure Assessment
SAPS II	Simplified Acute Physiology Score II
eICU-CRD	Electronic Intensive Care Unit Collaborative Research Database
AMI-CS	Acute Myocardial Infarction - Cardiogenic Shock
SHAP	SHapley Additive exPlanations
ICD-10	International Classification of Diseases, 10th Revision
NLP	Natural Language Processing
MLP	Multilayer Perceptron
SMOTE	Synthetic Minority Over-sampling Technique
Borderline-SMOTE	Borderline Synthetic Minority Over-sampling Technique
ADASYN	Adaptive Synthetic Sampling
SMOTE-Tomek	SMOTE combined with Tomek Links
SMOTE-ENN	SMOTE combined with Edited Nearest Neighbors
RF	Random Forest
LR	Logistic Regression

Chapter 1

Introduction - Motivation

Weakening of the heart muscle, the more generally used term heart failure, is among the leading causes of morbidity and mortality throughout the world, highlighting the importance of accurate and reliable prediction instruments.[6] Such prediction models are invaluable in assisting clinicians to make well-informed decisions and enabling more efficient use of health resources. Heart failure has reached epidemic proportions, with rising survival rates and a growing older population driving it and significantly elevating its overall incidence.

Advances in technology in recent times, particularly in artificial intelligence and machine learning, have opened up new possibilities for patient outcome prediction in heart failure based on complex, high-resolution clinical data. The current project examines the application of various machine learning algorithms to forecast critical outcomes such as Intensive Care Unit (ICU) admission and 30-day mortality after hospitalization. The ICU is a specialized hospital department where patients with critical health conditions receive continuous monitoring and advanced life support.

Predicting 30-day mortality after hospitalization helps clinicians identify high-risk patients and tailor treatment plans accordingly. Simultaneously, predicting ICU admission enables better allocation of healthcare resources and timely intensive care for those who need it most. Together, these two targets support improved patient outcomes and more efficient management of heart failure cases.

Models are trained on datasets from the public MIMIC-IV database, which has extensive records of ICU patients. Focusing on the convergence of epidemiological intelligence and computational horsepower, this research aims to support early detection and improved care for high-risk patients for heart failure—ultimately cracking one of the largest health puzzles of our era.

1.1 Problem Statement

Cardiac failure ranks among the world's top reasons for hospitalization and mortality with a tremendous health care burden. Early mortality prediction and ICU admission are important for maximizing patient management and resource allocation. With access to large data sets like MIMIC-IV, allows researchers to use electronic health records (EHRs) in developing prediction models using machine learning models.[7]

This thesis is concerned with MIMIC-IV dataset heart failure patients and attempts to develop two significant clinical event prediction models: mortality within the first month of hospitalization and ICU admission. Various machine learning models are employed, which fall under various model categories: linear models (Logistic Regression), support vector machines (SVM), tree-based models (Random Forest, Extreme Gradient Boosting, Light Gradient Boosting Machine), and tabular data neural network models, Tabular Transformer. These models are then tried out on the suitable performance metrics and validation procedures in anticipation of a consistent and stable predictive model. Afterwards, their results were compared in order to identify the most suitable model for this specific prediction task. Accuracy, area under the curve - receiver operating characteristic (AUC-ROC), and precision-recall were used to evaluate the models' performance, allowing a fair comparison. Through this analysis, the strengths and weaknesses of each algorithm in processing the clinical data to arrive at the best solution towards the problem at hand were determined.

Chapter 2

Related Work

A report on forecasting in-hospital mortality in acute heart failure patients was published in the Journal of Cardiothoracic and Vascular Anesthesia in 2024 [8]. In the publication, patient data were extracted from the MIMIC-IV database, with 5,114 ICU patients at a mean age of 72 years (55% male). Machine learning models applied for mortality prediction included Random Forest, XGBoost, LightGBM, and K-Nearest Neighbors, which were all evaluated using 5-fold cross-validation. Model performance was measured in terms of the area under the Receiver Operating Characteristic (ROC) curve (AUC). XGBoost provided the maximum performance of AUC 0.82 among the models, which also outperformed traditional ICU severity scores like Sequential Organ Failure Assessment (SOFA) and Simplified Acute Physiology Score II (SAPS II). Their aim was to determine whether machine learning could improve risk stratification in critically ill patients with heart failure.

Another retrospective study developed a mortality prediction model for cardiogenic shock patients from acute myocardial infarction, from the MIMIC-IV and Electronic Intensive Care Unit Collaborative Research Database (eICU-CRD) databases [9]. A total of 570 and 391 Acute Myocardial Infarction - Cardiogenic Shock (AMI-CS) patients were included with a mean age of 68.3 years. Various machine learning models were utilized, including Adaptive Boosting, XGBoost, Random Forest, and Logistic Regression. Model evaluation was carried out using AUC, and unexpectedly, the highest AUC of 0.869 was achieved by Logistic Regression. The authors stressed that even though ensemble techniques are powerful, simple models are capable of generating high predictive accuracy depending on the distribution of the data and the preprocessing.

A third study aimed to identify prognostic variables for the development of heart failure in patients with hypertension based on the XGBoost algorithm from the MIMIC-IV database [10]. The study identified 21 significant predictors, such as atrial fibrillation, renal dysfunction, anticoagulant use, and chronic obstructive pulmonary disease (COPD), to be highly associated with the development of heart failure. The focus was not only on prediction but also on unlocking clinically interpretable results, illustrating the explainability of the XGBoost model by feature importance scores.

Yet another study tackled clinical feature interpretation for heart failure patients through SHapley Additive exPlanations (SHAP) values on XGBoost outputs [11]. With a clinical dataset, authors identified the strongest features affecting mortality risk, i.e., age, temperature, heart rate, and respiratory rate. The paper emphasized that ensemble-based models such as XGBoost are not only accurate but also interpretable when SHAP is applied, which is desirable for clinical acceptance.

A recent paper proposed an interpretable deep learning model to predict in-hospital mortality for acute myocardial infarction patients using both MIMIC-IV and eICU datasets [12]. The model combined a Transformer-based model with intersample attention mechanisms and was trained on 39 clinical features. It achieved an AUC of 0.86, which outperformed traditional risk scores and baseline machine learning models. The authors emphasized the model interpretability facilitated by attention maps and post hoc analysis, guaranteeing transparency of clinical decision support systems.

A data-driven study from Veradigm PINNACLE outpatient registry linked to Symphony Health's Integrated Dataverse (IDV), evaluated machine learning potential for hospital readmission and worsening heart failure events (WHFEs) prediction [13]. The patient population in the dataset were those with heart failure with reduced ejection fraction (HFrEF). The models compared were Random Forest and XGBoost, both validated for 30-day, 90-day, and 365-day readmission risk. XGBoost had the best AUC for 30-day readmission (AUC = 0.595), whereas Random Forest had the best for 90-day readmission (AUC = 0.630). Although the AUCs were quite moderate, the study highlighted how

feature engineering, for instance, using Clinical Classifications Software (CCS) frequencies, can influence model performance in ambulatory care.

In a study [14], the authors investigated the limitations of using accuracy as a performance metric in binary classification tasks involving imbalanced datasets. They demonstrated that in cases where one class significantly outweighs the other, accuracy can give a false impression of model effectiveness. Instead, the study recommended the use of precision, recall, and F1-score as more reliable metrics. The authors particularly emphasized that recall is crucial when false negatives carry serious consequences, which is often the case in medical applications involving patient risk assessment.

These combined studies show that deep learning models such as XGBoost, LightGBM, Random Forest, SVM, and Logistic Regression are very common and high-performing classifiers of heart failure outcomes. As remarkable as performance with deep models such as Transformers has grown. Despite this ,tree-like models are still solid benchmarks, especially for structured clinical data. Incorporation of methods of feature explanation (such as SHAP) has made these models even stronger in terms of clinical feasibility.

Chapter 3

Data Source and Methodology

3.1 Data Source

The data of this work is Medical Information Mart for Intensive Care IV (MIMIC-IV), a large multi-center intensive care database established by the MIT Lab for Computational Physiology. There are about 65,794 ICU stays of 53,150 individual patients admitted into the intensive care unit (ICU) over the period between 2008 and 2019 in Boston's Beth Israel Deaconess Medical Center.The database contains rich clinical information such as demographics, vital signs, lab values, medications, procedures, diagnoses, and clinical notes. MIMIC-IV is applied widely in clinical research and enables the development of advanced models for patient outcome prediction and decision support. [15]

3.2 Data Description

In this study, we focused on a range of heart failure-related conditions, identified in the MIMIC-IV database using International Classification of Diseases, 10th Revision (ICD-10) diagnostic codes. Patients diagnosed with these conditions (Table 3.1) were selected as the study population. We extracted relevant clinical information from various tables, including demographic details (e.g., age, gender, ethnicity) and laboratory measurements. Notably, laboratory data were limited to the first six hours following ICU admission to better reflect the patients' early clinical status and ensure consistency in the analysis. Below is the table listing the selected heart failure diseases.

Table 3.1: Heart Failure Diseases

ICD Code	Description
I110	Hypertensive heart disease with heart failure
I130	Hypertensive heart and chronic kidney disease with heart failure and stage 1 through stage 4 chronic kidney disease, or unspecified chronic kidney disease
I132	Hypertensive heart and chronic kidney disease with heart failure and with stage 5 chronic kidney disease, or end stage renal disease
I5020	Unspecified systolic (congestive) heart failure
I5021	Acute systolic (congestive) heart failure
I5022	Chronic systolic (congestive) heart failure
I5023	Acute on chronic systolic (congestive) heart failure
I5030	Unspecified diastolic (congestive) heart failure
I5031	Acute diastolic (congestive) heart failure
I5032	Chronic diastolic (congestive) heart failure
I5033	Acute on chronic diastolic (congestive) heart failure
I5040	Unspecified combined systolic (congestive) and diastolic (congestive) heart failure
I5041	Acute combined systolic (congestive) and diastolic (congestive) heart failure
I5042	Chronic combined systolic (congestive) and diastolic (congestive) heart failure
I5043	Acute on chronic combined systolic (congestive) and diastolic (congestive) heart failure
I50810	Right heart failure, unspecified
I50811	Acute right heart failure
I50813	Acute on chronic right heart failure
I50814	Right heart failure due to left heart failure
I5082	Biventricular heart failure
I5083	High output heart failure
I5089	Other heart failure
I509	Heart failure, unspecified

The table above outlines the heart failure criteria used to define the study population. Based on the selected diagnoses, 11,864 patients with an average age of 70.60 years were included. Two prediction tasks were formulated: (1) ICU admission, and (2) 30-day mortality following hospitalization. Among the

patients, 4,209 (35.48%) were admitted to the ICU, and 1,256 (10.59%) died within the first month. The dataset included 4,396 females and 5,095 males, with slightly higher ICU admission and mortality rates observed in males. These statistics offer a general overview of the population and form the foundation for the predictive modeling that follows.

Table 3.2: Demographic characteristics

	Patients	Average age	Sex	
			Female	Male
	11864	70.60	4396	5095
ICU Admissions				
ICU admitted	4209 (35.48%)	70.48	1822 (33.00%)	2387 (37.64%)
ICU not admitted	7655 (64.52%)	70.68	3700 (67.00%)	3955 (62.36%)
Mortality first month				
Dead	1256 (10.59%)	76.33	557 (10.09%)	699 (11.91%)
Alive	10608 (89.41%)	69.93	4965 (89.91%)	5643 (88.98%)

3.3 Feature Selection

The features selected for this study include demographic data as well as laboratory measurements from the patients presented earlier. The selection was based on correlation analysis, aiming to identify the features most relevant to the prediction task of this study. The selected features were then appropriately preprocessed to be compatible with the machine learning models developed in the context of this research.

Table 3.3: Demographic Features with description

Feature Names	**Description**
subject_id	Is a unique identifier which specifies an individual patient
icd_code_1	The first disease (the International Coding Definitions (ICD) code.)
icd_code_2	The secondary disease (the International Coding Definitions (ICD) code.)
total_minutes_in_hosp	Total minutes in hospital
admission_location	Provides information about the location of the patient prior to arriving at the hospital
race	Race
marital_status	Marital status
admission_type	Is useful for classifying the urgency of the admission
gender	Gender
anchor_age	Shifted age for privacy reasons
Height (Inches)	Height
Weight (Lbs)	Weight
diastolic	Diastolic blood pressure
systolic	Systolic blood pressure
BMI (kg/m2)	Body Mass Index
hospital_entries	How many times does the patient enter the hospital

Table 3.4: Lab Features labels

Lab Features - Labels	
Alveolar-arterial Gradient	Bicarbonate, Urine
Base Excess	Creatinine Clearance
Calculated Bicarbonate, Whole Blood	Creatinine, Serum
Carboxyhemoglobin	Length of Urine Collection
Chloride, Whole Blood	Total Collection Time
Free Calcium	Urine Creatinine
Glucose	Urine Volume, Total
Hematocrit, Calculated	Mesothelial Cell (Ascites,Hematology)
Lactate	Other(Cerebrospinal Fluid, Hematology)
Oxygen Saturation	Plasma
pCO2	CD3 %
pH	CD3 Absolute Count
pO2	CD5 %
Potassium, Whole Blood	CD5 Absolute Count
Albumin, Ascites	Heparin, LMW
25-OH Vitamin D	Hypersegmented Neutrophils
Acetaminophen	Atypical Lymphocytes (Other Body Fluid,Hematology)
Alanine Aminotransferase (ALT)	Bands
Alkaline Phosphatase	Basophils
Anti-DGP (IgA/IgG)	Eosinophils(Other Body Fluid, Hematology)
Beta-2 Microglobulin	Mesothelial cells(Other Body Fluid,Hematology)
CA-125	NRBC
Calcium, Total	Atypical Lymphocytes (Pleural,Hematology)
Carcinoembyronic Antigen (CEA)	Other(Ascites,Hematology)
Chloride	Plasma Cells
Cyclosporin	WBC Casts
Ethanol	HIV 1 Viral Load

Phenytoin, Free	H. pylori IgG Ab Value
Phenytoin, Percent Free	VZV IgG Ab Value
Phosphate	Lactate Dehydrogenase, CSF
Rapamycin	UTX10
Salicylate	dRVVT - Screen
Thyroglobulin	SCT - Screen
Amylase, Body Fluid	Eosinophils (Cerebrospinal Fluid, Hematology)
Bilirubin, Total, Body Fluid	Macrophage
24 hr Protein	Other(Pleural,Hematology)

3.4 Preprocessing

Preprocessing the data is important before training the machine learning models because it significantly impacts the performance, accuracy, and generalization ability of the models. It means transforming raw data into clean data in a structured form that is ready to be analyzed. In the course of this study, some preprocessing techniques were applied, including correlation analysis for feature selection and retention of the most informative features, imputation to fill missing values, standardization to normalize numerical features into a uniform range, and ordinal encoding to convert categorical variables into a numerical form that most machine learning algorithms accept. Techniques such as oversampling or undersampling were considered to ensure a more balanced distribution of the target variable. All preprocessing steps except for correlation analysis and imputation were applied independently within each cross-validation fold to prevent data leakage and ensure a fair evaluation. Proper preprocessing not only improves data quality but also increases model training efficiency and predictive power.[16]

3.4.1 Correlation

Knowing how the variables relate to each other is one way of seeing patterns or relations which are likely to have effects on outcomes. If there is more than one feature, whether or not one or more are likely to follow each other and affect each other can guide follow-up analysis or

modeling. Correlation analysis is used as a statistical technique for estimating the strength and direction of linear associations among continuous variables to meet this requirement. Pearson correlation coefficient is used in this study to measure the degree of linear relationship between two continuous variables. The coefficient's value will be between -1 and 1. A coefficient value of 1 indicates strong positive relationship, and a coefficient value of -1 indicates strong negative relationship. The value of 0 close to it indicates very little or no linear relationship. This method helps in the creation of informative relationships, preventing redundancy in features, and enhancing the interpretability of models to be built.[17]

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

r = correlation coefficient

x_i = values of the x-variable in a sample

$\bar{x}$ = mean of the values of the x-variable

y_i = values of the y-variable in a sample

$\bar{y}$ = mean of the values of the y-variable

3.4.2 Imputation

Missing values are a pervasive problem in datasets found in practice and can negatively impact the validity and performance of statistical analyses and machine learning algorithms. Imputation methods are used to make inferences and substitute missing values with potential alternatives. Depending on the missingness mechanism and type of data, simple methods such as mean or median imputation for numerical values.The most frequently occurring category was used to impute missing values in categorical variables, as it preserves the original distribution and does not affect numerical aggregates such as the mean.

Proper imputation preserves data integrity and reduces subsequent analysis bias. [18]

3.4.3 Standardization

Standardization is a preprocessing technique used to normalize all numerical features onto a comparable scale, usually having a mean of zero and unit standard deviation. It is especially important for machine learning models based on distance measurements or linear transformations, such as Support Vector Machines and neural networks. By standardizing the features, the model prevents any bias toward features with larger value ranges, making sure that all features have an equal contribution to the learning process.[19]

3.4.4 Ordinal encoding

In order to be used in machine learning, the categorical features need to be transformed into numeric representation using encoding techniques. Ordinal encoding is one of the techniques whereby each category is assigned an integer that reflects the natural order of the categories. This transformation allows algorithms to handle categorical data effectively and preserve meaningful relationships between categories. Ordinal encoding is especially useful when the categorical features have a natural ordering, such as rankings or sizes.[35]

3.4.5 Class Imbalance

The employed data in this study are class-imbalanced, with one class significantly outweighing the other in terms of instance count. This imbalance can lead to model bias against the minority class, reducing its ability to accurately identify those instances. To address this issue, appropriate balancing techniques were applied to improve training quality and model performance. Specifically, class weighting methods were used in tree-based models that support this option, while for the other models, the

Adaptive Synthetic Sampling (ADASYN) oversampling technique was employed.[20]

3.4.6 Feature Engineering

Demographic and clinical features were selected based on domain knowledge to support meaningful input to the models. For example, features like admission type, number of hospital visits, and ICD codes for primary and secondary diagnoses were considered clinically relevant for predicting patient outcomes. This selection process incorporated clinical understanding to improve the dataset’s representation and the model's ability to learn from significant patient attributes.

Chapter 4

Classification Algorithms and Evaluation Framework

4.1 Classification Methods

Machine learning is a method of artificial intelligence that allows for computers to learn from information and make choices directly without explicit programming. Its general category is supervised learning, where the model is trained using labeled data to predict accurately. Under its scope, classification is among the primary activities, where the aim is to predict which class a new data point will fall into. A binary classifier is a model that performs binary classification, i.e., it predicts a value between two classes, such as "yes" and "no".

Classification models can be grouped into different categories based on their underlying approach, such as tree-based models like Random Forest, Light Gradient Boosting Machine, Extreme Gradient Boosting , linear models like Logistic Regression, Support Vector Machine, and deep learning models such as Tabular Transformer. In our research, we utilize representative methods from each of these categories to evaluate and compare their performance in binary classification tasks.

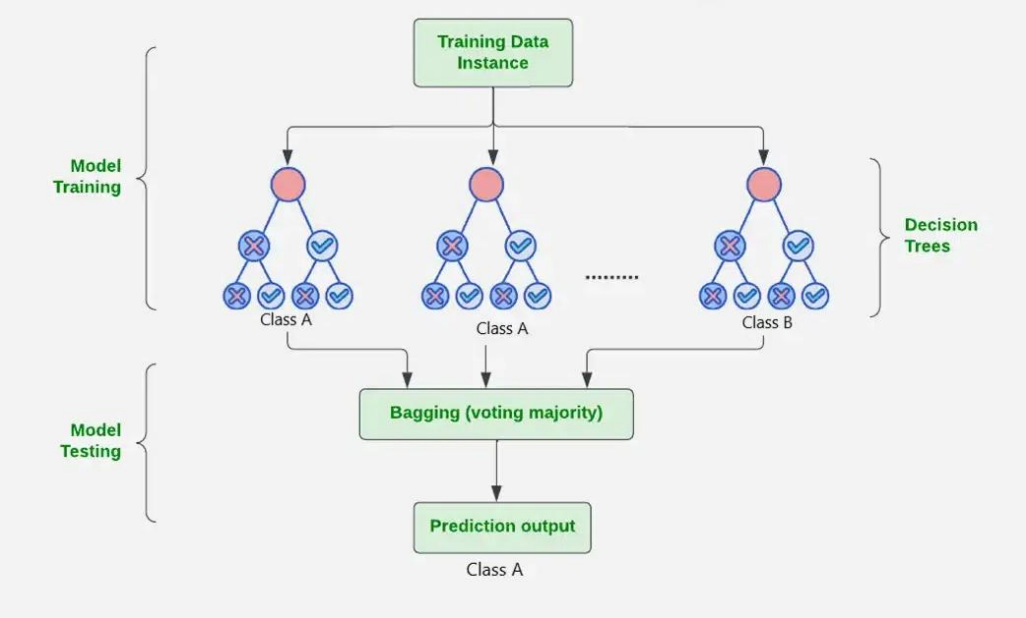
4.1.1 Random Forest

Random Forest is one of the most popular and robust supervised machine learning algorithms, widely chosen for its interpretability and ability to handle complex datasets with good accuracy in both classification and regression tasks. It was initially suggested by Leo Breiman in 2001 [21] with decision trees and ensemble method bagging (bootstrap aggregating). The overall idea of Random Forest is that a number of weak or simple models such as decision trees can be aggregated to give a significantly stronger and robust predictive model.

Random Forest creates many decision trees during training. Each tree is trained on some other random subset of the available dataset, and at each decision in a tree, a random subset of the features is used for the purpose of selecting the best split. This dual randomization—first in data sampling and second in features—encourages trees to be diverse and avoids overfitting, which is generally an issue with single decision trees.

For prediction, the algorithm uses a voting mechanism (majority vote for classification) or averaging (mean for regression) across all the trees in the forest. This collective decision-making leads to higher accuracy and lower variance compared to the use of a single decision tree. Random Forest is also less prone to noise in the data and works reasonably well even with missing or incomplete data.[22]

Figure 4.1: Random Forest diagram



Source: https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/

4.1.2 Extreme Gradient Boosting

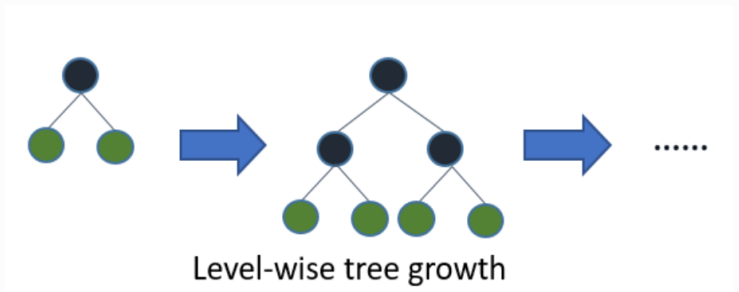
Extreme Gradient Boosting (XGBoost) is a scalable and performance-optimized gradient boosting algorithm, favored for its high predictive accuracy and efficiency in handling large and complex datasets. XGBoost, introduced by Tianqi Chen and Carlos Guestrin in 2016 [25], is well-known for its efficiency

and success in machine learning competitions as well as in practical deployment. XGBoost builds an ensemble of decision trees sequentially, where each tree tries to correct the residuals (errors) of the previous ones. The sequential error correction helps in reducing bias and improving predictive capability.

A critical feature of XGBoost is the use of a regularized objective function for controlling model complexity and preventing overfitting. Specifically, it combines the training loss with a regularization term (based on the number of leaves and the L2 norm of leaf weights), which favors simpler models with good generalization. XGBoost also incorporates techniques like shrinkage (learning rate) and column subsampling, both of which render it robust and generalize well. Moreover, XGBoost uses a level-wise tree growth strategy, where all the leaves at the same depth are split before moving deeper. This level-wise approach tends to produce more balanced trees and reduces the risk of overfitting, making it more stable—especially for smaller datasets.

Implementation-wise, XGBoost also supports parallel processing, which is much faster than traditional gradient boosting methods. It does this through a novel strategy of finding the best split points for trees using histogram-based approximation. On top of that, XGBoost is also very flexible and supports tree and linear booster models, missing value handling, early stopping, and GPU acceleration, making it suitable for a wide variety of tasks from classification to ranking. [24]

Figure 4.2: XGBoost diagram



Source: https://datascience.stackexchange.com/questions/26699/decision-trees-leaf-wise-best-first-and-level-wise-tree-traverse

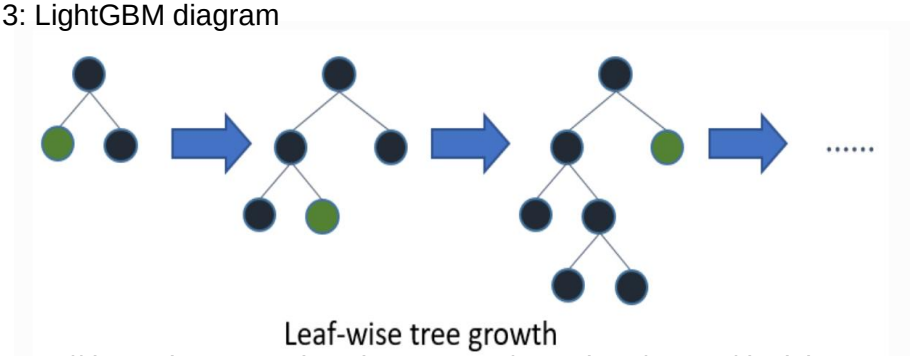
4.1.3 Light Gradient Boosting

Light Gradient Boosting Machine (LightGBM) is an extremely rapid gradient boosting framework developed by Microsoft [23], chosen for its ability to deliver high prediction performance while significantly reducing training time and memory usage. As with other boosting algorithms, LightGBM builds an ensemble of decision trees sequentially where the following tree attempts to correct the errors of the preceding ones. The novelty of LightGBM is that it is computationally efficient and scalable, therefore can handle high-dimensional data as well as large data.

Among the core innovations of LightGBM is the use of Gradient-based One-Side Sampling (GOSS). The approach selects high gradient value points—those hardest for the model to forecast—and then samples at random from the rest of the points, in effect focusing on the most beneficial samples. In addition, Exclusive Feature Bundling (EFB) is utilized in an effort to reduce the number of features by bundling mutually exclusive features, thereby drastically reducing dimensionality and enhancing efficiency without sacrificing accuracy.

In contrast to other boosting libraries such as XGBoost, LightGBM constructs trees leaf-wise (best-first), rather than level-wise. This means that it always splits the leaf that gives the highest reduction in loss. It starts with raw data, applies techniques like GOSS and EFB to speed up training and reduce dimensionality, and trains trees sequentially, each one correcting the errors of the previous. The goal is fast, memory-efficient training with high predictive accuracy. [24]

Figure 4.3: LightGBM diagram



Source: https://datascience.stackexchange.com/questions/26699/decision-trees-leaf-wise-best-first-and-level-wise-tree-traverse

4.1.4 Logistic Regression

Logistic Regression is a fundamental statistical method for binary classification tasks, modeling the probability of class membership when the target variable is limited to two outcomes. Linear regression, on the other hand, produces continuous values, whereas logistic regression output is an estimation of the probability an input point would be in a given class. This is achieved through the use of the logistic (sigmoid) function, which maps any real-valued number to a value between 0 and 1. The model works by performing a linear

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$$

logit function

combination of the input features and subsequently applying the outcome to the sigmoid function [26]. In mathematical terms, this can be expressed as:

Apply the exponential function to both sides to eliminate the logarithm and follow the general mathematics:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)}} \quad \Rightarrow \quad \sigma(z) = \frac{1}{1 + e^{-z}}$$

logistic function derivation

sigmoid function

In this case, the β terms are the model coefficients, which are learned from the data on training by an operation like maximum likelihood estimation. The output is interpreted as the probability of the input belonging to class 1, and a threshold (usually 0.5) is applied to create a final prediction.

Logistic regression is highly respected for its interpretability, simplicity, and power, especially if the independent variables and target have a rough linear relationship. It handles linearly separable data well and is a satisfactory default choice for a wide range of classification problems. Although, it can struggle with

complex or non-linear data structures unless supplemented with interaction terms or kernel tricks. [27]

4.1.5 Support Vector Machine

Support Vector Machines (SVM) is a supervised machine learning algorithm for classification and regression [28] . Its core idea is to find the optimal hyperplane that best separates two or more classes by maximizing the margin between them, improving model robustness and generalization. One of the most prominent arguments of SVM is that it performs exceptionally well even in high-dimensional space, i.e., data with many features. Moreover, by employing kernel functions, SVM can handle non-linear classifying problems by transforming the data to a higher-dimensional space where linear separability is simpler. The most widely used kernels are linear, sigmoid, polynomial, and RBF (Radial Basis Function).

$$\textbf{RBF Kernel: } k(x,y) = e^{\frac{(x-y)^2}{2\sigma^2}}$$
$$\textbf{Sigmoid Kernel: } k(x,y) = \tanh \gamma(x^T y) + r$$
$$\textbf{Polynomial Kernel: } k(x,y) = (\gamma x^T y + r)^d$$
$$\textbf{Linear Kernel: } k(x,y) = x^T y$$

Though efficient, SVM proves to be inefficient if used for classifying very large datasets since its computational complexity is exponential. It is still, nevertheless, a favorite and trustworthy approach to binary classification problems and is being employed intensively in fields of image classification, disease diagnosis, and spam filtering.

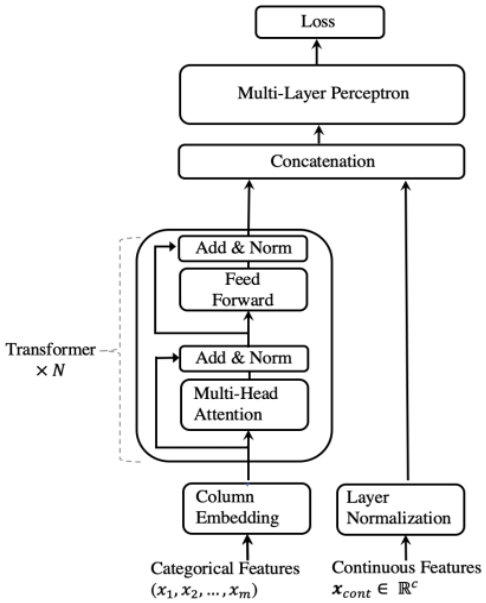
4.1.6 Tabular Transformer

Tabular Transformer is a cutting-edge deep learning model specifically designed for structured/tabular data, which are the most common data types in real-world applications such as healthcare, finance, and business intelligence.

It is chosen for its ability to capture complex feature interactions and improve prediction accuracy. A majority of typical deep learning models like Multi-Layer Perceptrons (MLPs) usually perform less optimally than gradient boosting machines like XGBoost when dealing with tabular data, primarily due to the fact that they have poor categorical variable management and the ability to capture complicated feature interactions.

TabTransformer, introduced by Huang et al. (2020) [29] , resolves these deficiencies through leveraging Transformer models, originally developed for Natural Language Processing (NLP), to perform self-attention-based embedding of categorical features. These context-sensitive embeddings achieve between-variable interactions for categorical variables and merge them with continuous features and input into an MLP for predictions. TabTransformer supports transfer learning, allowing the model to generalize well even with little labeled data. This makes it most valuable in small dataset domains or high-dimensional categorical variables.

Figure 4.4: The architecture of TabTransformer



Source: https://arxiv.org/abs/2012.06678

4.2 Validation and Machine Learning Evaluation

Validation and evaluation in machine learning are critical processes to ensure that a model generalizes to new, unseen data. Validation involves splitting the dataset into subsets (e.g., train/validation/test) in order to estimate the model's performance on data it has not seen during training. Common techniques include k-fold cross-validation, which gives more accurate performance estimates. Metrics of measurement are based on a proper measure of accuracy such as precision, recall, F1-score, or AUC depending on the type of problem being classification, regression, etc. In our application, which involves binary classification for a medical prediction task with class imbalance, certain specialized metrics are required. Simple accuracy is too naive and misleading, and emphasis is placed upon the confusion matrix, sensitivity, specificity, and precision. In addition, Precision-Recall curve and ROC-AUC are crucial to measure the quality of how the model marks the most critical cases, particularly crucial in health applications when false diagnoses will have life-changing consequences.

Confusion matrix

A confusion matrix is a tool used to evaluate classification models by comparing actual vs. predicted labels. It contains:

- True Positives (TP) – Correctly predicted positives
- True Negatives (TN) – Correctly predicted negatives
- False Positives (FP) – Incorrectly predicted positives (Type I error)
- False Negatives (FN) – Incorrectly predicted negatives (Type II error)

From the confusion matrix, we derive the following metrics:

Accuracy

The proportion of total correct predictions (both positive and negative) out of all predictions.

$$\mathbf{Accuracy} = \frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{FP} + \mathbf{TN} + \mathbf{FN}}$$

Precision

Out of all predicted positives, how many were actually positive — useful for reducing false positives.

$$Precision = \frac{TP}{TP + FP}$$

Recall / Sensitivity

Of all actual positives, how many the model correctly found — important for not missing real positives.

$$Recall = \frac{TP}{TP + FN}$$

Specificity

Of all actual negatives, how many were correctly predicted — helps avoid false positives.

$$\mathbf{Specificity} = \frac{\mathbf{TN}}{\mathbf{TN} + \mathbf{FP}}$$

False Positive Rate

Of all real negatives, how many were wrongly predicted as positive.

$$\mathbf{FPR} = \frac{\mathbf{FP}}{\mathbf{FP} + \mathbf{TN}}$$

ROC-AUC & Precision-Recall Curve

ROC-AUC Curve: Plots Recall (TPR) vs False Positive Rate (FPR) at all thresholds. It shows how well the model separates the two classes. Useful when classes are balanced and you care about both positives and negatives.

Precision-Recall Curve: Plots Precision vs Recall (TPR) at all thresholds. It shows the trade-off between correctly finding positives and being accurate when predicting positives.Useful when positive class is rare (imbalanced data), like disease detection.

4.2.1 Resampling

Class imbalance is common in clinical datasets, as severe outcomes like ICU admission or death occur less frequently than non-serious ones. In this study, multiple resampling techniques were tested to mitigate this issue, including Synthetic Minority Over-sampling Technique (SMOTE),Borderline Synthetic Minority Over-sampling Technique (Borderline-SMOTE), Adaptive Synthetic Sampling (ADASYN), SMOTE combined with Tomek Links (SMOTE-Tomek), SMOTE combined with Edited Nearest Neighbors (SMOTE-ENN), and the ScalePosWeight method [5]. Specifically, ADASYN generates new samples for the minority class based on the difficulty of classification, in contrast SMOTE creates synthetic samples between neighboring minority class instances. Borderline-SMOTE focuses on "borderline" samples near the decision boundary. SMOTE-Tomek combines oversampling with the removal of Tomek links (pairs of close samples from different classes), and SMOTE-ENN combines oversampling with the removal of noisy samples using the Edited Nearest Neighbours method [36]. Finally, ScalePosWeight adjusts the weight of the minority class without altering the dataset itself [37]. Each technique was applied to improve the model’s ability to detect rare but clinically important outcomes. A comparative evaluation was conducted, and the detailed performance of each method is discussed in the results section, emphasizing their practical impact in medical prediction tasks. These techniques have been widely adopted in imbalanced learning, especially in healthcare applications where identifying minority class cases is critical.[20]

4.2.2 Cross Validation

Several studies have reported that standard cross-validation can lead to overoptimistic performance estimates, particularly when hyperparameter tuning and performance evaluation are performed on the same data folds. To address this issue, our work employs a more strict validation strategy. Hyperparameter tuning is conducted within an inner stratified cross-validation loop. The final model's performance is then evaluated on an independent, held-out test set that remains completely unseen during both training and model selection. This approach effectively replicates the structure of nested cross-validation and enables an unbiased estimation of generalization performance, in line with recommendations from the literature [38]. In particular, stratified K-fold cross-validation was used where the data were divided into k subsets (folds) such that the class distribution within each fold was preserved. The model was trained and tested k times, using different folds as the validation set each time and the other k–1 folds as the training set. This method has the advantage that the majority and minority classes will have an equal proportion in all the splits, which is desirable for the situation with imbalanced medical data. By averaging the fold evaluation scores, the results reflect more accurately and the performance over genuinely unseen data as anticipated, which is absolutely essential in applications of high-stakes such as healthcare.[30]

4.3 Hyperparameters Tuning

Hyperparameter tuning is the adjustment of the hyperparameters of a machine learning model which are not trained on the data, such as learning rate, batch size, etc. They strongly affect the model's performance and ability to generalize. Correct choice of hyperparameters can significantly improve the accuracy of predictions, whereas a wrong choice can lead to underfitting or overfitting. Tuning systematically is therefore of highest importance to construct trustable models.

Table 4.1: Hyperparameter tuning for prediction of ICU admission

Hyperparameter	TabTransformer	LightGBM	XGBoost	Random Forest
Learning rate (lr)	log-uniform [0.0001–0.01]	[0.01, 0.05, 0.1]	[0.01, 0.05, 0.1]	-
Number of estimators	-	integer [100–300]	[100, 300, 500]	integer [50–500]
Max depth	-	[7,9,-1] (-1=no limit)	[3,5,6]	integer [3–15]
Dropout	continuous [0.2–0.6]	-	-	-
Batch size	integer [32–128]	-	-	-
Number of leaves	-	[15,31,63]	-	-
Subsample	-	-	[0.7,0.8,1.0]	-
Colsample by tree	-	-	[0.7,0.8,1.0]	-
Min samples split	-	-	-	integer [2–10]
Min samples leaf	-	-	-	integer [1–5]
Max leaf nodes	-	-	-	integer [10–1000]
Layers	[128–64, 256–128, 512–256,512–256–128]	-	-	-

Table 4.2: Hyperparameter tuning for prediction of ICU admission (SVM, LR)

Hyperparameter	SVM	Logistic Regression
C (regularization)	log-uniform [0.01–10]	[0.001, 0.005, 0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 1, 5, 10]
Gamma	log-uniform [0.00001–0.01]	-
Kernel	one of ['rbf', 'sigmoid']	-

Table 4.3: Hyperparameter tuning for 30 day mortality rate prediction

Hyperparameter	**TabTransformer**	**LightGBM**	**XGBoost**	**Random Forest**
Learning rate (lr)	log-uniform [0.0001–0.01]	log-uniform [0.0001–0.1]	log-uniform [0.0001–0.1]	-
Number of estimators	-	integer [500–1100]	Integer [50–1000]	integer [50–150]
Max depth	-	integer [4–14]	integer [2–6]	integer [3–8]
Layers	[256–128, 512–256, 512–256–128]	-	-	-
Dropout	continuous [0.4–0.5]	-	-	-
Batch size	[64, 128, 256]	-	-	-
Number of leaves	-	integer [20–100]	-	-
Min child samples	-	integer [30–50]	-	-
Min child weight	-	-	integer [3–15]	-
reg_alpha (L1)	-	log-uniform [0.01–3]	log-uniform [5–60]	-
reg_lambda (L2)	-	log-uniform [0.01–3]	log-uniform [5–60]	-
Colsample by tree	-	continuous [0.3–0.7]	continuous [0.5–0.9]	-
Subsample	-	-	continuous [0.5–0.9]	-
Gamma	-	-	continuous [3–12]	-
Min samples split	-	-	-	integer [10–25]
Min samples leaf	-	-	-	integer [5–12]
Max features	-	-	-	continuous [0.2–0.5]
Max samples	-	-	-	continuous [0.5–0.8]

Table 4.4: Hyperparameter tuning for 30 day mortality rate prediction (SVM, LR)

Hyperparameter	SVM	Logistic Regression
C	log-uniform [0.001–10]	[0.001, 0.005, 0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 1, 5, 10]
Gamma	log-uniform [0.00001–0.1]	-
Kernel	one of ['rbf', 'sigmoid']	-

In this study, an extensive and modern hyperparameter tuning process was conducted, with Bayesian optimization being used in most cases to efficiently explore the search space and identify optimal configurations. Grid search was applied only for Logistic Regression, as its simplicity allowed for an exhaustive yet manageable search. For the second prediction task, a larger set of hyperparameters was considered due to the increased class imbalance in the data, which required the inclusion of regularization techniques (L1 and L2) to prevent overfitting and improve the models' generalization ability.

4.3.1 Bayesian Search Cross-Validation

Bayesian Search Cross-Validation is a high-powered hyperparameter optimization technique that builds a probabilistic model of the objective function and subsequently employs it to select the most suitable hyperparameter combinations to attempt. In contrast to regular grid or random search, Bayesian optimization allows the balance between exploration and exploitation as it learns from iterations, which renders it much more efficient, especially in high-dimensional or costly search spaces. With each step, it updates its hypothesis about the objective function from history and chooses the next set of hyperparameters. Studies have shown that this method works better than other search strategies in the majority of machine learning tasks [31] , and therefore it is a suitable choice for hyperparameter tuning of complex models in the present study.

Bayesian optimization is based on Bayes' theorem, which provides a mathematical way to update the probability of a hypothesis based on new evidence. The core formula is:

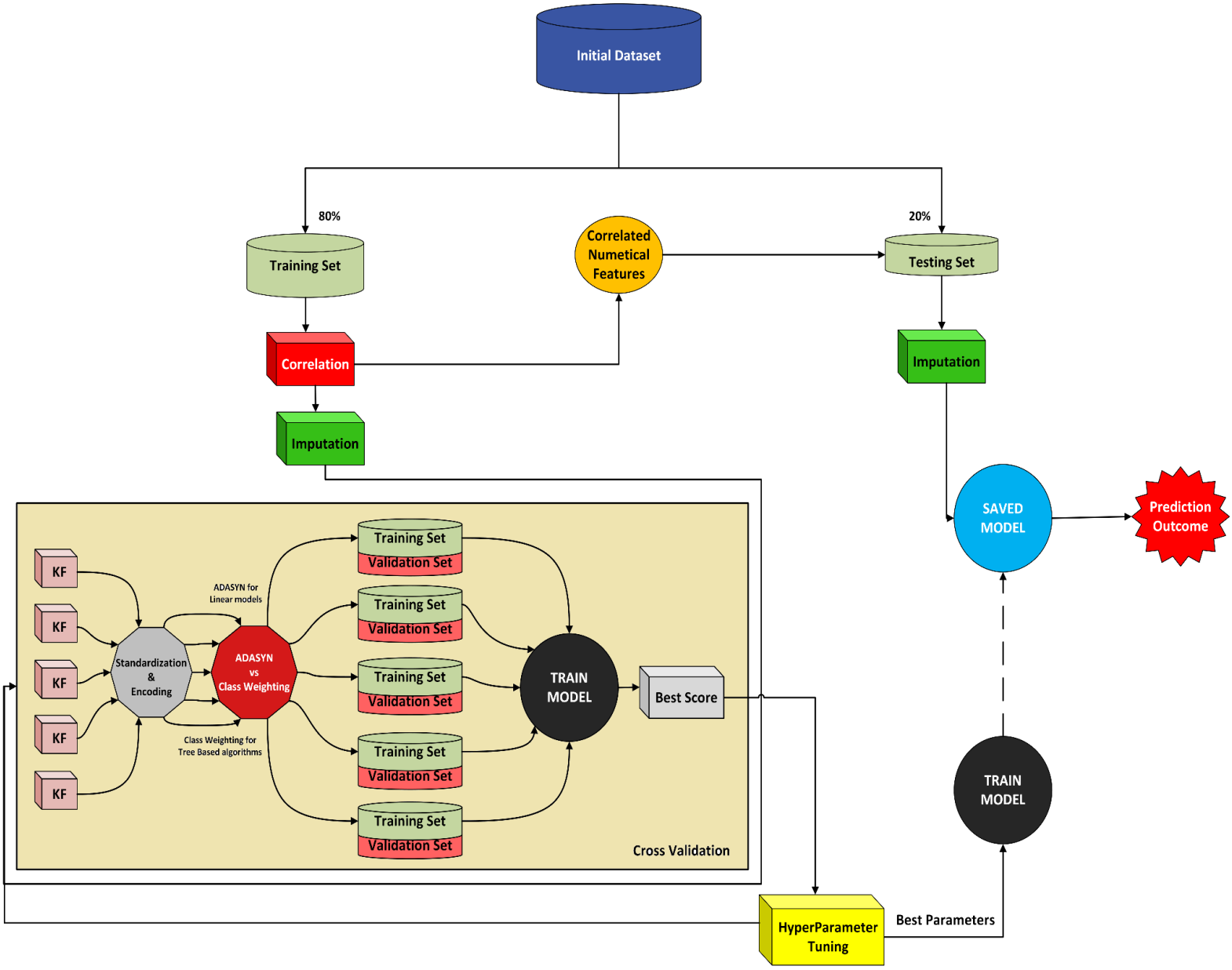
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \text{ if } P(B) \neq 0.$$

Chapter 5

Machine Learning Pipeline

Using a well-structured machine learning pipeline, as depicted below figure, is crucial for handling complex datasets and ensuring consistent model performance. The division into training and testing sets at the outset allows it to be feasible to have an unbiased ultimate evaluation [32]. Operations such as correlation filtering and imputation play a crucial role in data quality and dimensionality reduction, which affect model precision [33]. The application of ADASYN or class weighting addresses this imbalance, which is common in medical datasets, by generating synthetic samples adaptively or adjusting the model's focus on minority classes. This improves both recall and accuracy, enhancing the model's ability to correctly identify rare but critical cases. Standardization and encoding are utilized to condition data appropriately for multiple model types (linear vs. tree-based models). K-Fold cross-validation offers reliable estimation of performance through variance reduction by minimizing data split effects. Hyperparameter optimization automates searching for the optimal model parameters, overcoming manual trial-and-error and generalizing better. Finally, saving and deploying the trained model provides reproducibility and enables real-time results in predictions, a critical operation within clinical practice where reliability matters a great deal [34].

Figure 5.1 : Machine Learning pipeline



5.1 Preprocessing Stage

Preprocessing is a fundamental operation in any machine learning pipeline since it directly affects the quality and performance of the resulting model. Raw data is normally noisy, contains missing values, inconsistencies, and irrelevant features that may confuse learning algorithms and compromise predictive power. Before model training, the dataset is split into training and test sets, and each set is independently preprocessed. By preprocessing through standardisation, imputation, encoding, and data transformation, models are ensured to learn from structured and meaningful input, and thus derive more generalisable and consistent results. Preprocessing quality often exerts a greater influence on performance than algorithm selection.

5.1.1 Split Dataset

The initial step in the machine learning pipeline was to split the dataset into a training set and a test set in a ratio of 80-20. It is really important to observe how well the model will be doing on unseen data. The training set (80%) is utilized to train the model, and the test set (20%) is kept separate and only used to confirm the performance after training. Both sets were written to separate files and used separately in later stages of the pipeline in order to have a clean separation and prevent any data leakage. Also applied the stratify parameter to ensure that the class distribution remains consistent across both sets, which is crucial when working with imbalanced classes.

5.1.2 Correlation

The next stage in the pipeline was correlation analysis, which was performed before imputation. This decision was made because many features had extremely high rates of missing values (up to 98%) and imputing them before assessing their usefulness would introduce unnecessary computational cost

and potential noise. By running correlation analysis first, I was able to identify and remove highly redundant features early on, especially those with excessive missing values that would be unreliable to impute. This step ensured that only meaningful, non-duplicated features were passed to the imputation stage, improving both efficiency and model robustness. The analysis was conducted only on the training set to avoid data leakage. I checked how similar the numerical features are by calculating their correlation. To avoid checking the same pairs twice, I looked only at the upper triangle of the correlation table (everything above the diagonal), which compares all features against all others at once. Features with a correlation above 0.9 were flagged for removal. The same features were also removed from the test set to avoid evaluating the model on unseen columns and ensure consistency across both datasets.

5.1.3 Imputation

To handle missing data and ensure that the machine learning models can be trained effectively, numerical and categorical features were imputed. Missing values can negatively affect model performance or even prevent some algorithms from functioning effectively. For numeric columns, I imputed missing values with the mean value since it is a simple and efficient way of maintaining the overall distribution of data without suffering from bias towards outliers. For categorical columns, I imputed missing values with the most frequent category since it maintains the data distribution. The imputation was performed using the same fitted format on both the training and test sets, first fitting the imputer on the training data and then applying the same transformation to the test set to maintain consistency and prevent data leakage.

5.1.4 Standardization/Encoding

After handling missing values, I proceeded with standardization of numerical features and encoding of categorical features to prepare the data for model training. Numerical columns were standardized using StandardScaler(), which transforms the data to have a mean of 0 and a standard deviation of 1. This

step is important for models that are sensitive to the scale of input features, ensuring that all numerical variables contribute equally. For categorical columns, I used OrdinalEncoder() to convert category labels into integer values. Ordinal encoding was used as a simple and efficient method to transform categorical values into numerical format suitable for model input. Both the standardization and encoding steps were fitted on the training data within each k-fold and then applied to the corresponding validation fold using the same parameters, ensuring consistency and preventing data leakage.

5.1.5 Oversampling

One important procedure following data preprocessing and before model training was the resolution of the issue of class imbalance which is a critical problem in medical prediction tasks where minority class instances are exceedingly important. For more details, see Section 6.2.1. To address this, diverse methodologies were applied depending on the model type. Class weighting was used with tree-based classifiers, assigning higher importance to minority class instances during training. In contrast, for simpler models such as Support Vector Machines and Logistic Regression, the ADASYN (Adaptive Synthetic Sampling) technique was used to synthetically generate minority class examples. Both of these were used within each fold during cross-validation to ensure that the models were exposed to balanced data during learning but kept the validity and independence of sets for validation. This setting maintained the criteria for equitable learning on all the folds and assisted in more stable and generalizable performance.

5.2 Learning & Evaluation Stage

The learning and evaluation process is the most critical stage in any machine learning process because it significantly influences the final performance and overall generalization of the models. Bayesian Search Cross-Validation was utilized to identify the optimal combination of hyperparameters and enhance the accuracy of the models. This advanced technique was utilized because it efficiently explores the hyperparameter space by finding a balance between exploration and exploitation, using data from past evaluations to guide the search. Its probabilistic nature makes it particularly well-suited for high-complexity models with large or continuous parameter spaces.

Logistic regression alone was an exception, where a simpler grid search technique was employed due to the minimal hyperparameters and simplicity of the model. Both cases used hyperparameter tuning through the use of 5-fold cross-validation. Training input data were split evenly into five portions, four of which were utilized for training and one for validation per iteration. This method guaranteed that each combination of hyperparameters was tested reliably and consistently, avoiding overfitting and helping to select configurations that generalize well to unseen data.

Once the hyperparameter search was complete, the best-performing parameters, based on the average validation metrics across all folds, were selected for each model. These final models, trained on the full training set using the optimal hyperparameters, were then evaluated on the independent test set to provide an unbiased estimate of their generalization performance.

To further understand how the models arrived at their predictions, SHAP was employed for model interpretability. SHAP was chosen because it is a model-agnostic method that offers both local and global interpretability, allowing us to analyze the contribution of each feature to individual predictions and overall model behavior.

5.3 Prediction Stage

The final stage of the pipeline was to generate predictions on the unseen test set using the optimized models. After training each model with the best hyperparameter configuration discovered through cross-validation, predictions were generated on the test data that had been held completely separate from the training and validation process. This allowed for an unbiased and fair evaluation of model performance. The predicted probabilities and class labels were then used to compute the final evaluation metrics, allowing for a reliable assessment of each model's ability to generalize to new and unseen data.

Chapter 6

Results

6.1 ICU Admission Prediction for Heart Failure Patients

The data used for this prediction, ICU admission prediction for heart failure patients, consists of 11,864 patients with a mean age of 70.60 years. Among them, 4,209 (35.48%) were admitted to the ICU due to heart failure. Table 6.1 presents the demographic information. It should be noted that ages were calculated using adjusted birth years with a consistent offset, preserving the distribution's integrity while ensuring data anonymization. We extracted significant features, including demographic and laboratory ones. The total number of features used was 88.

Table 6.1: Demographic characteristics for ICU

	Patients	Average age	Sex	
			Female	Male
	11864	70.60	4396	5095
ICU Admissions				
ICU admitted	4209 (35.48%)	70.48	1822 (33%)	2387 (37.64%)
ICU not admitted	7655 (64.52%)	70.68	3700 (67%)	3955 (62.36%)

Results for ICU Admissions

Table 6.2 Evaluation metrics for ICU admissions

Models	Accuracy	Sensitivity	Specificity	AUC
TabTransformer	0.81 (±0.02)	0.75(±0.03)	0.85(±0.02)	0.88(±0.01)
LightGBM	0.87(±0.01)	0.80(±0.02)	0.91(±0.02)	0.94(±0.02)
XGBoost	0.87(±0.01)	0.80(±0.01)	0.91(±0.01)	0.94(±0.02)
Random Forest	0.85(±0.01)	0.78(±0.01)	0.89(±0.02)	0.92(±0.02)
SVM	0.77(±0.02)	0.78(±0.02)	0.77(±0.01)	0.81(±0.01)
Logistic Regression	0.77(±0.02)	0.78(±0.01)	0.77(±0.02)	0.86(±0.02)

The results indicate that LightGBM and XGBoost achieve the best overall performance for predicting ICU admissions, with high scores across all metrics—accuracy (0.87), sensitivity (0.80), specificity (0.91), and AUC (0.94). This suggests that these models strike an effective balance between correctly identifying patients who need ICU care and minimizing false positives. TabTransformer also performs competitively, though slightly lower. In contrast, SVM and Logistic Regression show noticeably weaker performance across all metrics, indicating they are less suitable for this task.

Figure 6.1 : AUC-ROC curve of ICU admissions

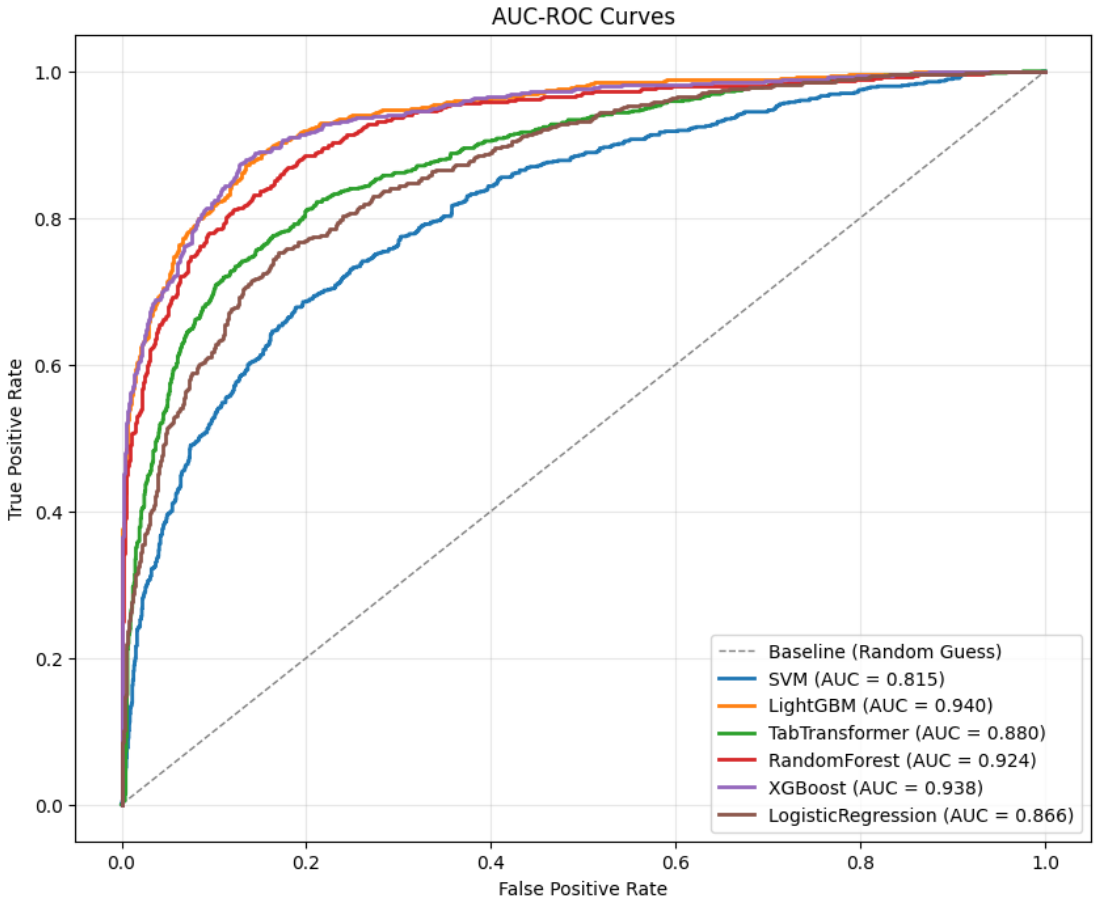


Figure 6.2 : Precision - Recall curve of ICU admissions

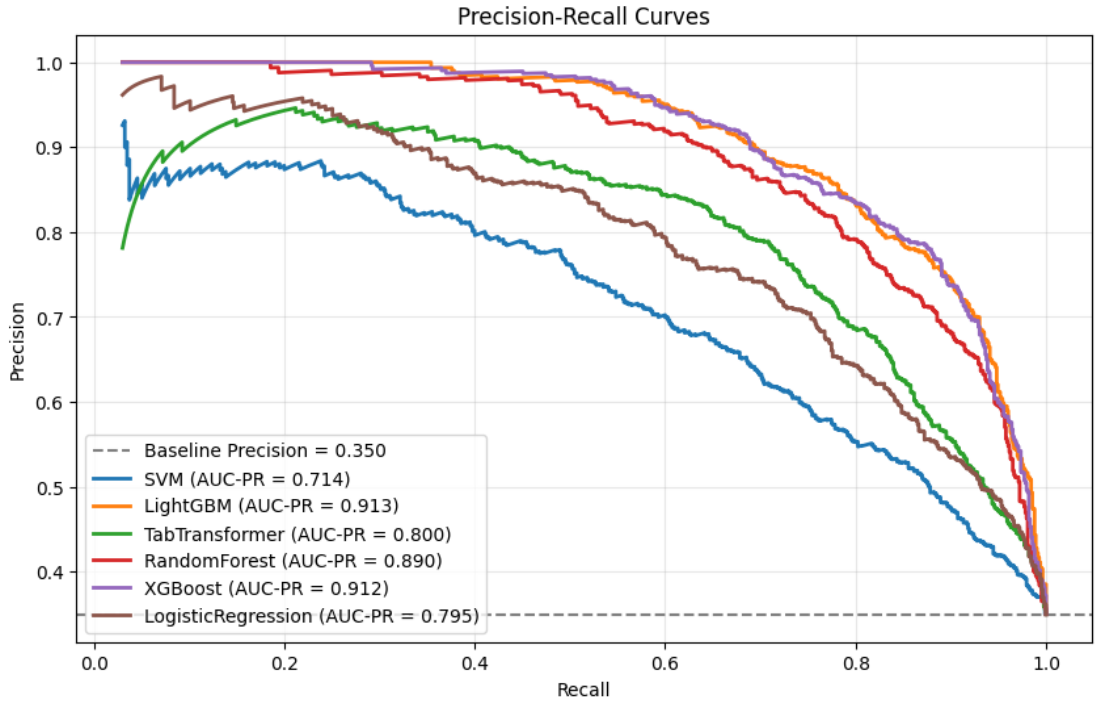


Table 6.3: Hyperparameters ICU admissions

XGBoost	LightGBM	RF	TabTransformer	LR	SVM
eta: 0.05	eta: 0.05	n_estimators: 500	eta: 0.00016	C: 0.3	C: 10
Max_depth: 5	Max_depth: -1	Max_depth: 15	Layers: 256-128	Solver : saga	Gamma: 0.0036
n_estimators: 300	n_estimators: 121	Min_sample_leaf: 1	Batch_size: 111		Kernel: rbf
Subsample: 0.8	Num_leaves: 31	Min_sample_split: 10	Dropout: 0.278		
Colsample_bytree: 0.7		Max_leaf_nodes: 1000			

Most Important Features:

I extracted the 10 most important features of the models Light Gradient Boosting Machine (LightGBM), Extreme Gradient Boosting (XGBoost) and TabTransformer using the method of the SHAP values.

Figure 6.3 : ICU admissions most important Features of LightGBM

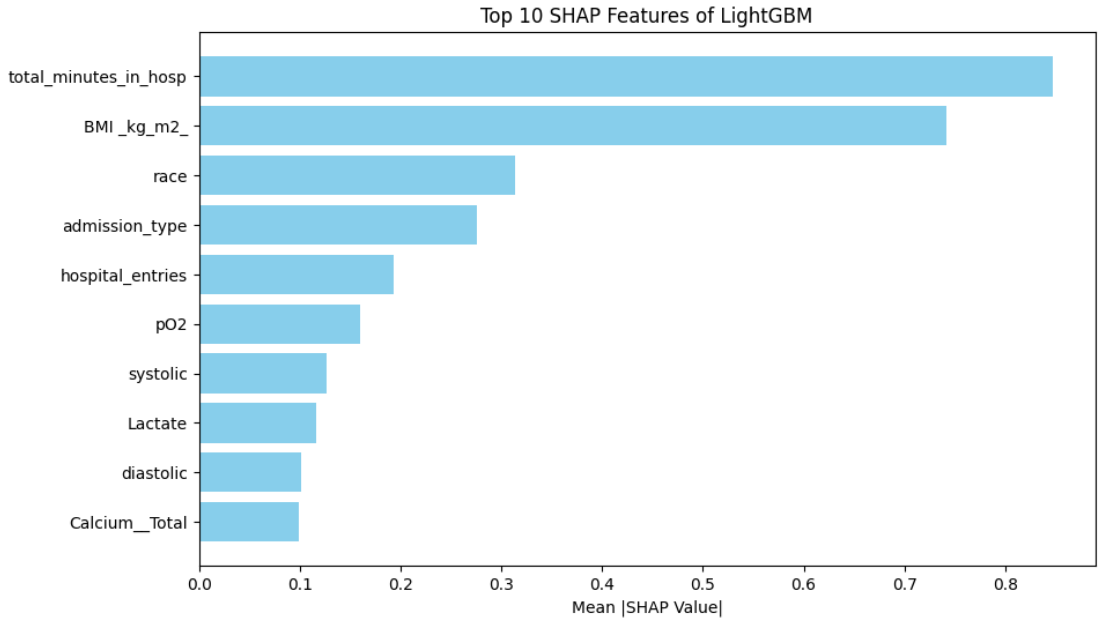


Figure 6.4 : ICU admissions most important Features of XGBoost

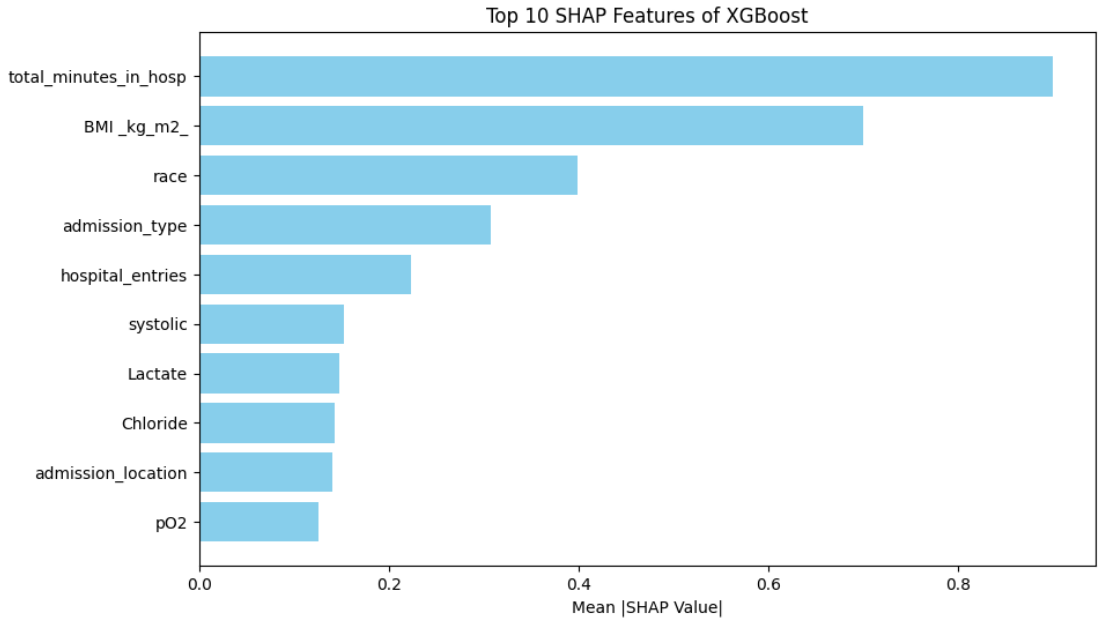
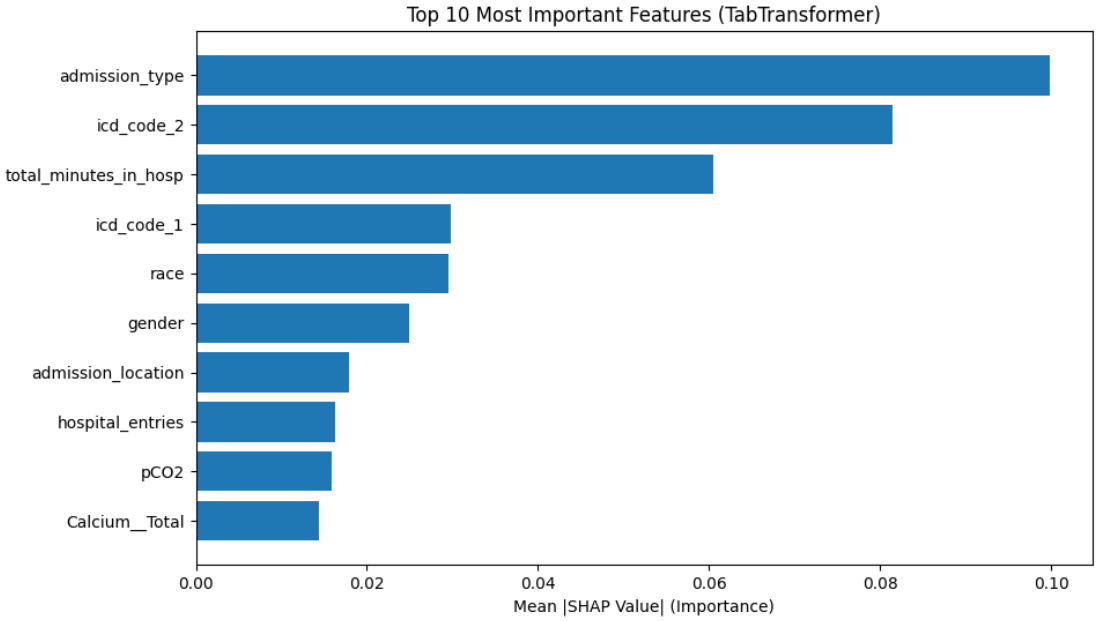


Figure 6.5 : ICU admissions most important Features of TabTransformer



6.2 30 Day Mortality Prediction in Patients with Heart Failure

The data used for this prediction, mortality within the first month of hospitalisation of patients with heart failure, consists of 11,864 patients with a mean age of 70.60 years. Among them, 1256 (10.59%) died within the first month of entering the hospital due to heart failure. Table 6.4 presents the demographic information. It should be noted that ages were calculated using adjusted birth years with a consistent offset, preserving the distribution's integrity while ensuring data anonymization. We extracted significant features, including demographic and laboratory ones. The total number of features used was 88.

Table 6.4: Demographic characteristics for 30 Day mortality prediction

	Patients	Average age	Sex	
			Female	Male
	11864	70.60	4396	5095
Mortality first month				
Dead	1256 (10.59%)	76.33	557 (10.09%)	699 (11.91%)
Alive	10608 (89.41%)	69.93	4965 (89.91%)	5643 (88.98%)

Results for 30 Day mortality prediction:

Table 6.5 Evaluation metrics for 30 Day mortality prediction

Models	Accuracy	Sensitivity	Specificity	AUC
TabTransformer	0.70(±0.02)	0.77(±0.03)	0.69(±0.01)	0.80(±0.01)
LightGBM	0.82(±0.01)	0.60(±0.03)	0.78(±0.01)	0.82(±0.02)
XGBoost	0.78(±0.02)	0.74(±0.01)	0.78(±0.02)	0.83(±0.01)
Random Forest	0.81(±0.01)	0.57(±0.01)	0.84(±0.01)	0.80(±0.02)
SVM	0.79(±0.01)	0.51(±0.02)	0.82(±0.01)	0.73(±0.02)
Logistic Regression	0.81(±0.01)	0.63(±0.01)	0.83(±0.02)	0.82(±0.02)

In the context of predicting 30 Day mortality prediction, XGBoost demonstrates the best overall balance, achieving high sensitivity (0.74), good specificity (0.78), and the highest AUC (0.83). LightGBM and Logistic Regression show strong accuracy (0.82 and 0.81, respectively), but their lower sensitivity values indicate that they may miss a significant number of true positive cases that are critical in such cases. The TabTransformer achieves the highest sensitivity (0.77), making it effective at identifying patients at risk of death, though at the cost of lower accuracy and specificity. On the other hand, SVM presents the weakest performance across most metrics, particularly sensitivity and AUC, suggesting it is less suitable for this task.

Figure 6.6 : AUC-ROC curve of 30 Day mortality prediction:

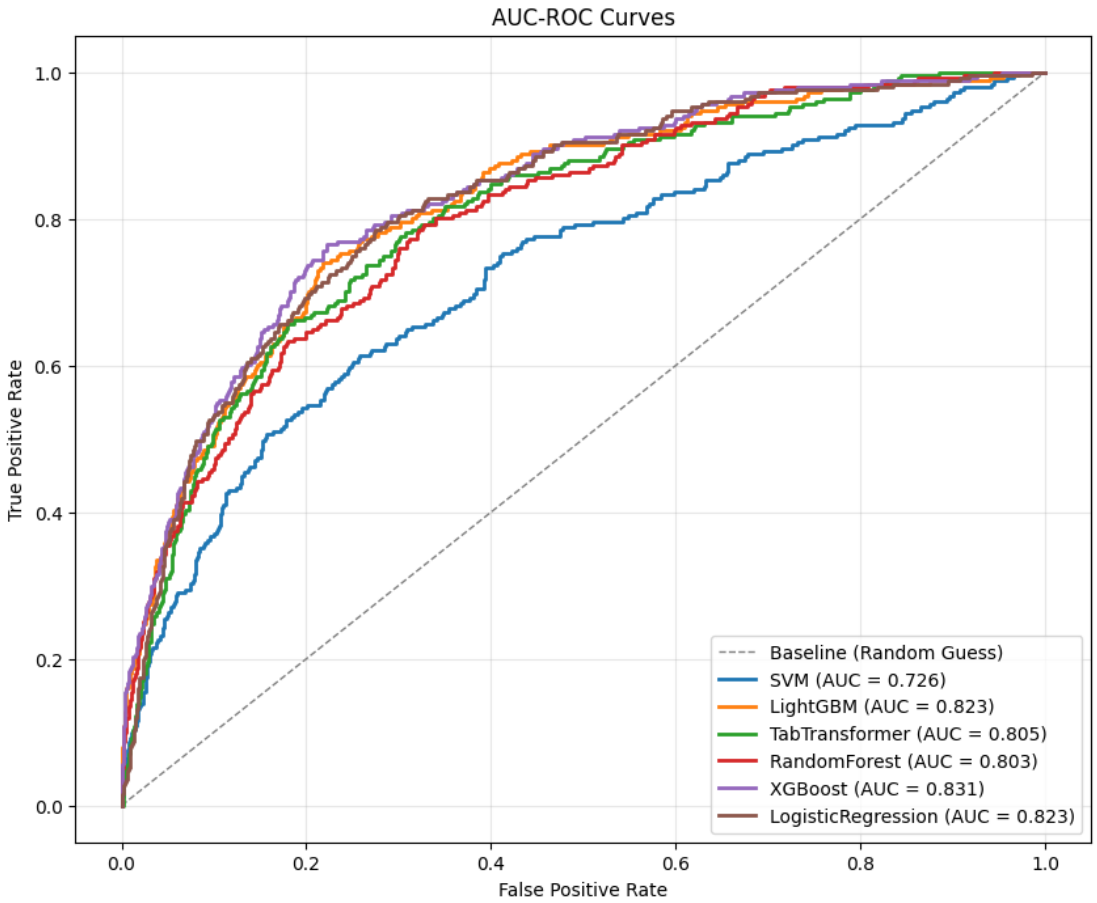
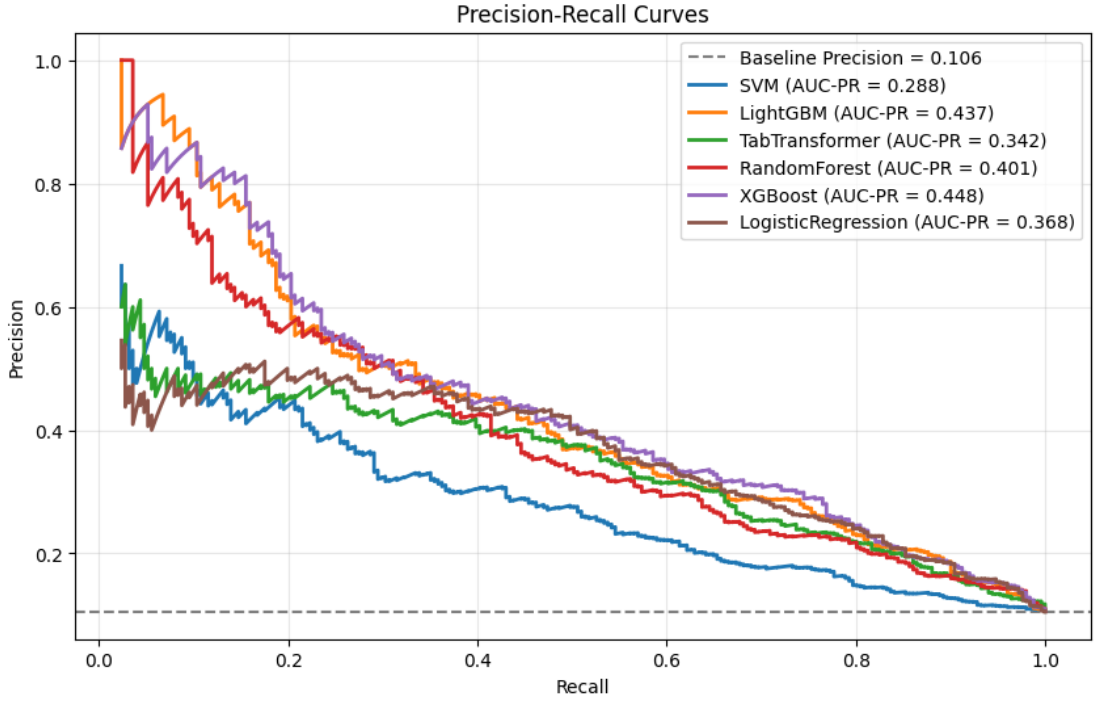


Figure 6.7 : Precision - Recall curve of 30 Day mortality prediction



6.2.1 Precision-Recall curve and oversampling-undersampling methods

Initially, performance in precision and recall was suboptimal due to class imbalance, which negatively affected the overall quality of the models. Since accuracy is misleading in such cases, the precision-recall curve was chosen as the primary evaluation metric, as it is generally more suitable and reliable for imbalanced classification problems. This is also supported in the literature, where researchers emphasize that metrics such as precision, recall, and F1-score are far more meaningful than accuracy when dealing with imbalance datasets.[1]

To address the imbalance, I explored various oversampling and undersampling techniques, including ADASYN, SMOTE, Borderline-SMOTE, SMOTE-Tomek, SMOTE-ENN, as well as the ScalePosWeight technique. [5] These methods were tested on the training set using models such as XGBoost, Logistic Regression, and SVM, and were evaluated using precision-recall curves. Based on the results of the precision-recall curves, the ADASYN technique with a sampling strategy of 0.5 was selected for the Logistic Regression and SVM models, as it provided the best balance between precision and recall. The sampling strategy defines the desired ratio between the minority and majority classes after oversampling. In contrast, for the other tree-based models (such as XGBoost), the class weighting technique using the scale_pos_weight parameter was more effective.

Interestingly, despite the fact that precision-recall curves are generally recommended for imbalanced datasets,the ROC curve appeared more stable and informative across models. This behavior is actually expected and is also discussed in the paper *"The receiver operating characteristic (ROC) curve accurately assesses imbalanced data*sets" [2]. The authors explain that precision-recall curves are more sensitive to class imbalance, since precision is directly influenced by the number of false positives, which tends to be higher when the positive class is rare. As a result, the lower shape of the precision-recall curve in my results is a natural outcome of the dataset's imbalance and not necessarily a sign of poor model performance. Nevertheless, the precision-

recall curve remains the most appropriate and informative metric in this context, as it focuses specifically on the minority class, which is the main interest in this classification task. Therefore, both ROC and precision-recall curves were analyzed, with greater emphasis placed on precision-recall due to the nature of the problem.

Figure 6.8: Precision - Recall curves for various oversampling Methods (XGBoost)

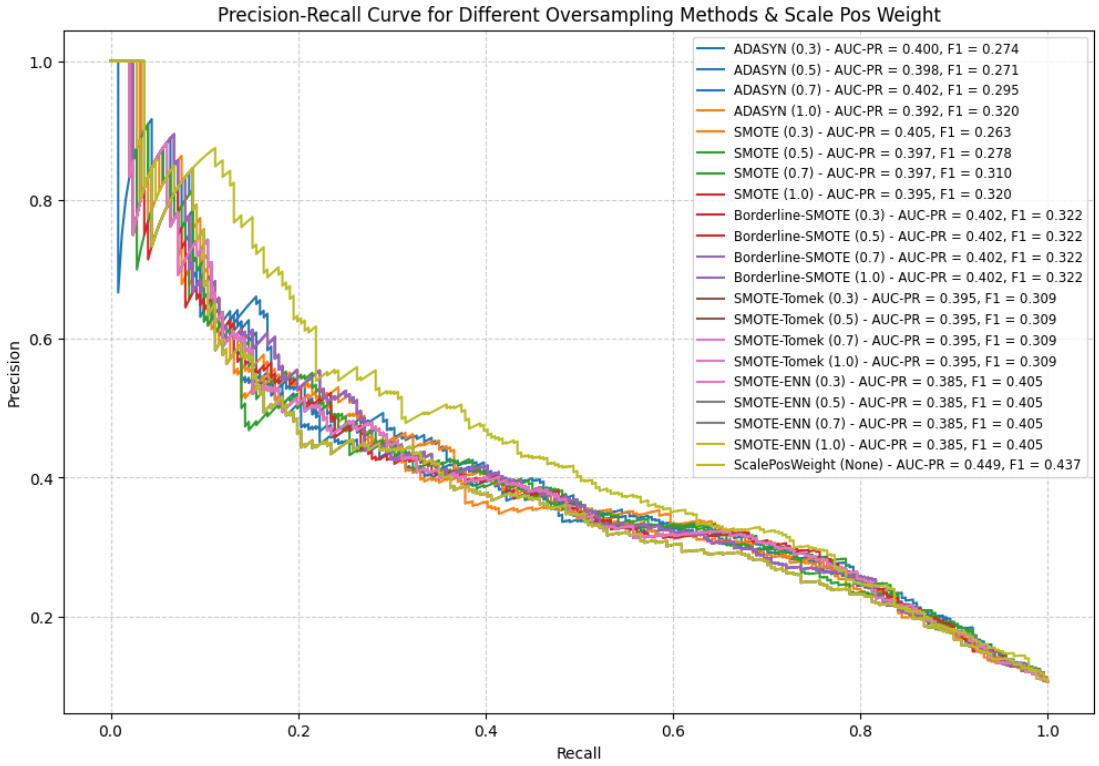


Figure 6.9: Precision - Recall curves for various oversampling Methods (SVM)

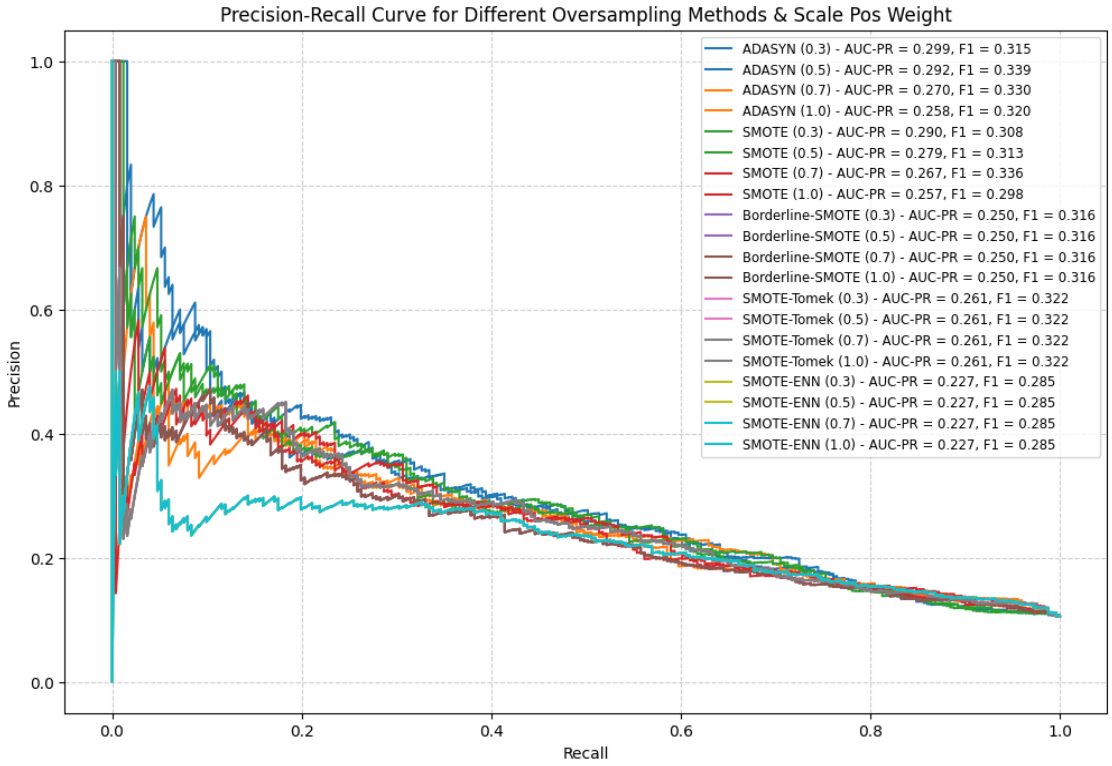


Figure 6.10: Precision - Recall curves for different oversampling Methods (Logistic Regression)

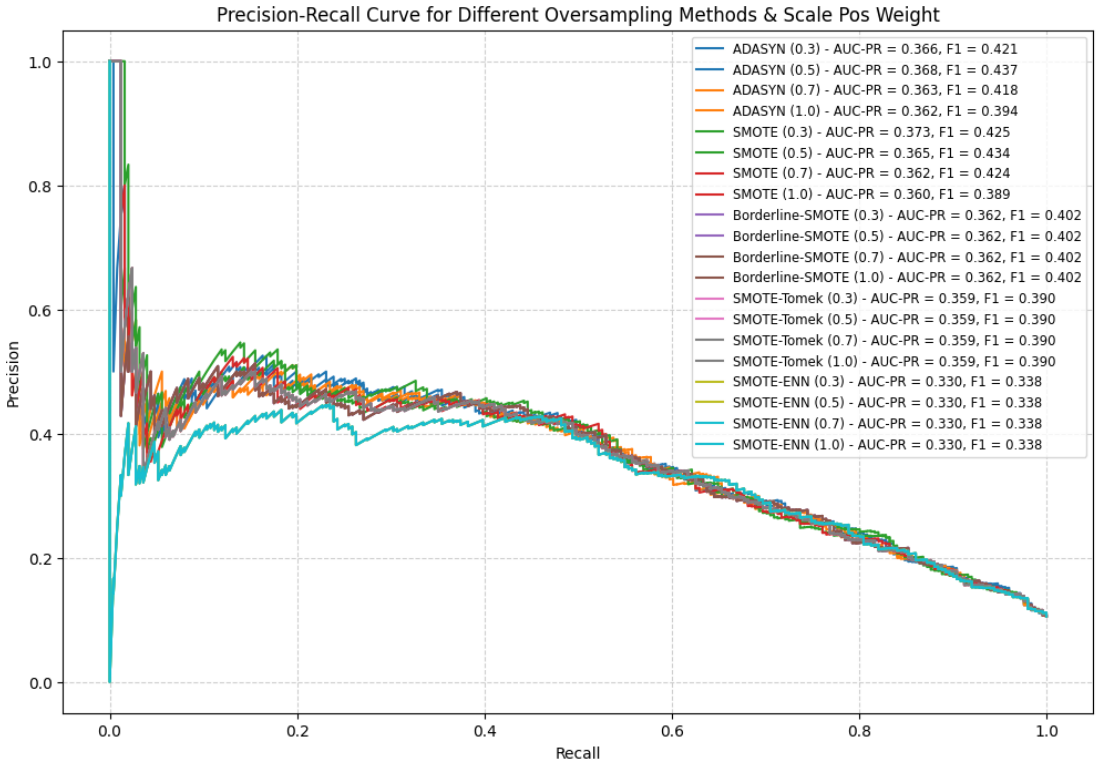


Table 6.6: Hyperparameters for 30 Day mortality prediction

XGBoost	LightGBM	RF	TabTransformers	LR	SVM
eta: 0.1	eta: 0.057	n_estimators: 120	eta: 0.00068	C: 0.3	C: 1.45
Max_depth: 6	Max_depth: 9	Max_depth: 7	Layers: 512-256-128	Solver: Liblinear	Gamma : 0.003
n_estimators: 1500	n_estimators: 727	Min_sample_leaf: 8	Batch_size: 256		Kernel: rbf
Subsample: 0.8	Num_leaves: 24	Min_sample_split: 15	Dropout: 0.47		
Colsample_bytree: 0.8	Colsample_bytree: 0.51	Max_samples: 0.5			
Gamma: 3	Min_child_samples: 47	Max_features : 0.2			
Min_child_weight: 3	Reg_alpha: 0.13				
Reg_alpha: 5	Reg_lamda: 0.0012				
Reg_lamda: 5					

Most Important Features:

Figure 6.11 : 30 Day mortality prediction most important Features of LightGBM

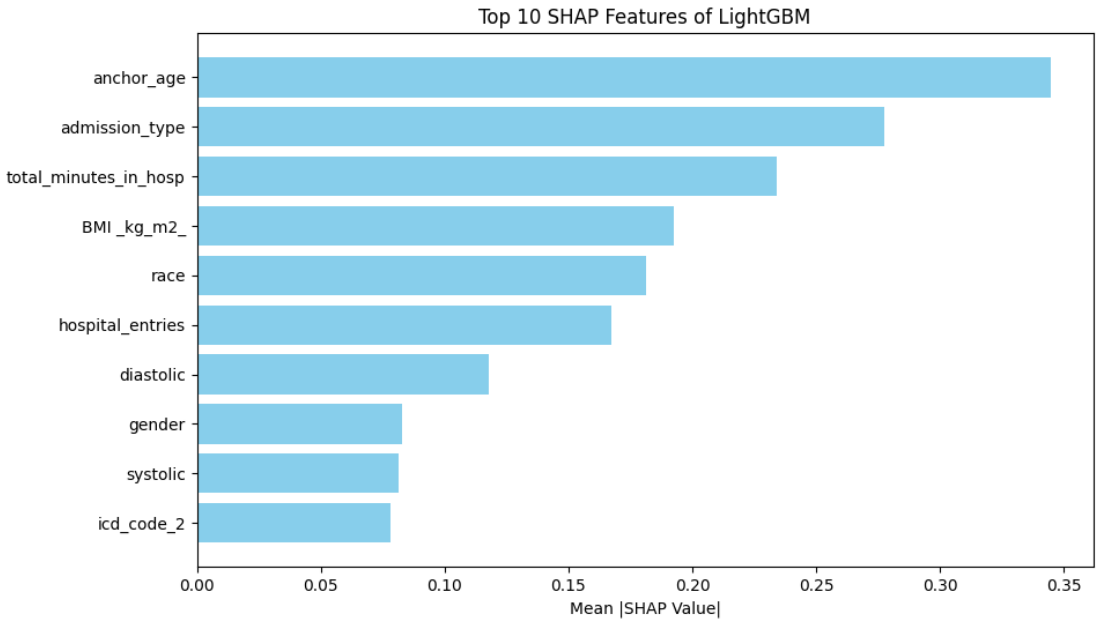


Figure 6.12 : 30 Day mortality prediction most important Features of XGBoost

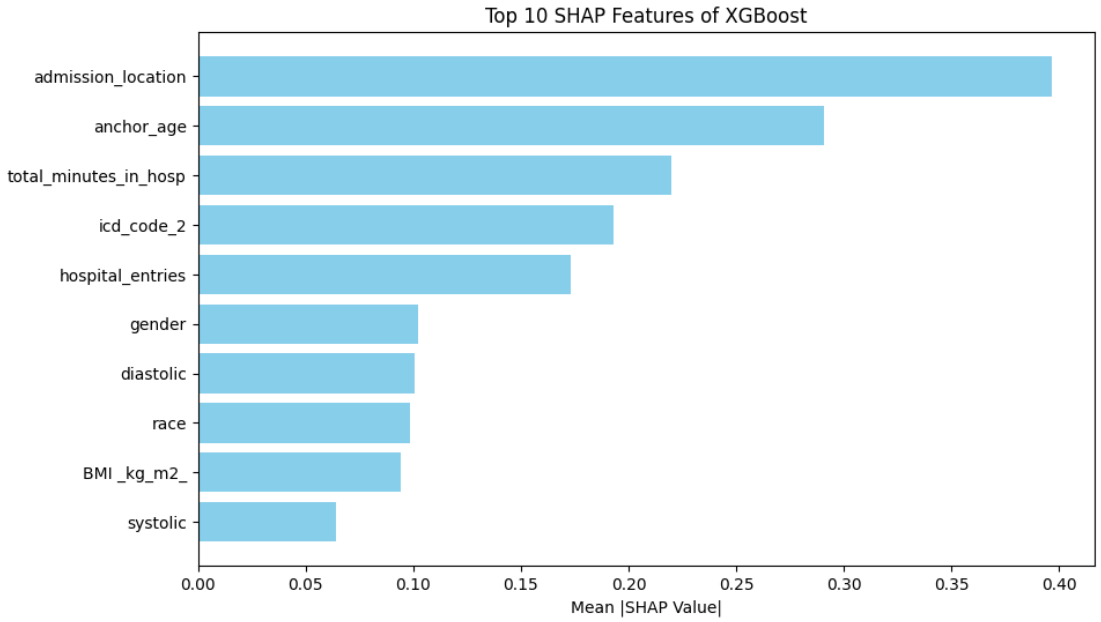
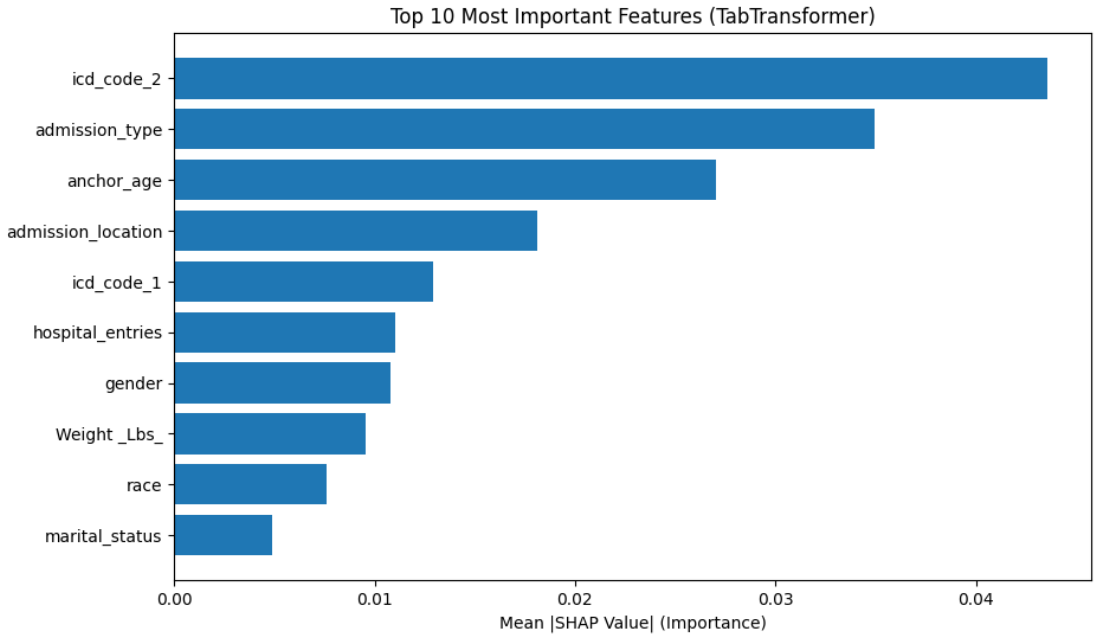


Figure 6.13 : 30 Day mortality prediction most important Features of TabTransformer



Chapter 7

Discussion

Tree-based models such as XGBoost and LightGBM consistently outperformed other algorithms across the prediction tasks, demonstrating superior accuracy, robustness, and efficiency. The main contribution of this paper is the exploration of numerous demographic and laboratory characteristics in electronic health records.The characteristics were investigated to determine their role in predicting two main outcomes: (a) ICU admission and (b) hospital death in the first month. The study utilized the publicly available MIMIC-IV database and included patients with diverse cardiovascular diagnoses, with a special emphasis on different classes of heart failure.

The medical data present in this study were complex, unstructured, and often incomplete, and presented huge challenges for analysis. In order to prepare data for machine learning algorithms, a thorough preprocessing phase was required. This involved labor-intensive operations such as imputation of missing values, removal of outliers and correlation between features to improve data quality and consistency.[3] Correlation analysis was applied in choosing redundant or irrelevant features and helped feature selection to ensure optimum model performance.[4] Although time-consuming, this exercise was essential in building a clean and trustworthy dataset.

In this study, several machine learning models were employed to compare different classification approaches. Linear models such as Logistic Regression, tree models such as Random Forest, XGBoost, and LightGBM, and more complex architectures such as Support Vector Machines (SVM) and the deep learning-based TabTransformer. Temporal sequence models like Long Short-Term Memory (LSTM) networks and Transformers are crucial for Health Event Records (HER) because they can effectively capture the time-dependent patterns and relationships within sequential health data. These models process

data points in order, allowing them to learn how past events influence future outcomes, which improves prediction accuracy in tasks involving Electronic Health Records (EHR) or Health Event Records (HER). All the models were carefully optimized through hyperparameter tuning with selection of solvers, kernel functions, learning rates, and regularization strategies, depending on the algorithm. The aim was to compare their relative strengths and suitability for the two prediction tasks. This variety of models gave a good comparison and allowed us to capture linear as well as non-linear relationships in the data. Although the process of tuning was computationally intense, it created models that not only were accurate but also extremely reliable.

For mortality prediction, all the models performed modestly, with AUC between 0.73 and 0.83. XGBoost had the highest AUC (0.83) with very good discriminative power, followed closely by Logistic Regression, LightGBM, and Random Forest at 0.82. TabTransformer showed high recall (0.77), likely due to its ability to learn rich feature embeddings that better capture the distinct properties of tabular data, which favors detecting more true positives. Recall is very important for mortality prediction, as failing to identify patients at high risk (false negatives) can lead to missed opportunities for early treatment. Nonetheless, it needs to be noted that the accuracy of some models would have been affected by class imbalance in mortality data, where the number of patients who died was much lower compared to patients surviving. Such an imbalance will make models favor the majority class, lowering recall or increasing false negatives. For instance, SVM performed the poorest on this task, with the lowest AUC (0.73) and recall (0.51), most likely because of its vulnerability to class imbalance distributions. Therefore, models with higher recall are more clinically valuable, even if they compromise slightly on other metrics.

In ICU admission prediction, the models performed significantly better. LightGBM and XGBoost had the highest AUC scores (0.94), with Random Forest (0.92) and TabTransformer (0.88) coming next. These models had good recall-precision balance, with LightGBM showing high recall (0.80) and specificity (0.91). TabTransformer was also competitive with recall (0.75) being

particularly important in a clinical setting where failure to capture important cases could be lethal. On the other hand, SVM and Logistic Regression were lower across all the metrics for this task, with AUCs of 0.81 and 0.86, respectively.

Compared to existing literature, our results demonstrate highly competitive and in some cases superior performance in predicting both ICU admission and in-hospital mortality for patients with heart failure. For example, in a recent 2024 study that focused on acute heart failure patients from MIMIC-IV, the best-performing model, XGBoost, achieved an AUC of 0.82 for mortality prediction [8]. In our case, XGBoost outperformed this result with an AUC of 0.83, likely due to more aggressive feature selection, imputation strategies, and inclusion of distinct heart failure subtypes. Similarly, although Logistic Regression reached an AUC of 0.869 in a cardiogenic shock cohort [9], our model with Logistic Regression achieved an AUC of 0.82, which is within reasonable range given our broader and more heterogeneous population of cardiovascular patients.

In terms of ICU admission prediction, our model achieved even higher performance, with XGBoost and LightGBM both reaching AUC scores of 0.94, exceeding the results in comparable studies where AUCs were rarely reported above 0.90. This indicates that our data preprocessing pipeline and model tuning contributed to robust generalization across different ML architectures. Furthermore, despite the fact that other studies have highlighted the importance of model interpretability through SHAP [11], our use of TabTransformer complements these efforts by offering recall-oriented predictions, which are critical in clinical environments. Lastly, unlike many previous works that emphasized only AUC or accuracy, we report and analyze precision, recall, and class imbalance explicitly aligning with recommendations in [14] for more comprehensive evaluation in medical classification tasks.

Overall, tree-based models delivered strong and balanced performance in both prediction tasks. TabTransformer showed promise as it manifested recall-oriented behavior, which is valuable in high-stakes clinical applications. These results underscore the need to evaluate models with appropriate metrics and

select the right algorithm based on the clinical objective — whether to prioritize early detection (recall), prevent false positives (precision), or offer overall discriminative power (AUC).

The study has several limitations. First, the medical data used were complex, unstructured, and incomplete, requiring extensive preprocessing that may have impacted the accuracy of the results. Additionally, class imbalance, especially in 30 day mortality rate prediction, likely affected some models' ability to correctly identify high-risk cases. Finally, the study relied only on the MIMIC-IV database, which limits the generalizability of the findings to other populations or clinical settings.

Chapter 8

Conclusions

The conducted research may serve as a proof-of-concept study for forecasting in-hospital 30-day mortality and ICU admission among cardiovascular disease patients, including various types of heart failure. The results are clinically valid and promising because the most relevant predictive features identified by the models align with established medical knowledge. The validation indicated that machine learning models are capable of providing accurate classification results, especially in the intensive care environment. Of specific interest were tree-based models such as XGBoost and Random Forest, which performed consistently well on both prediction tasks and are thus viable candidates for possible inclusion in future real-world clinical decision support systems. Moreover, these models show great promise for use in real-time clinical settings, helping doctors continuously monitor patients in intensive care, identify emergencies at an early stage, modify treatments instantly, and predict risks as they happen to support better decision-making.

Chapter 9

Future Work

Future work could include widening the scope of predictive models being employed, including deep learning-based models that have been tailored specifically to tabular health data. Further, incorporating temporal sequence and longitudinal data may also be beneficial for predicting outcomes for heart failure more precisely, based on monitoring the evolution of a patient's state over time. Examining methods for interpretability, e.g., SHAP, would also provide more transparency to model choices, a crucial element towards clinical uptake. Finally, their evaluation on real-world data or real-world hospital environments would be a key step towards generalizability and deployment in the field.

Bibliography

1. R. Mohammed, J. Rawashdeh, and M. Abdullah, “Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results,” in *2020 11th International Conference on Information and Communication Systems (ICICS)*, IEEE, Apr. 2020, pp. 243–248. Accessed: Mar. 27, 2025. [Online]. Available: https://doi.org/10.1109/icics49469.2020.239556
2. E. Richardson, R. Trevizani, J. A. Greenbaum, H. Carter, M. Nielsen, and B. Peters, “The receiver operating characteristic curve accurately assesses imbalanced datasets,” *Patterns*, vol. 5, no. 6, p. 100994, Jun. 2024, doi: 10.1016/j.patter.2024.100994.
3. K. Maharana, S. Mondal, and B. Nemade, “A review: Data pre-processing and data augmentation techniques,” *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.
4. P. Schober, C. Boer, and L. A. Schwarte, “Correlation Coefficients: Appropriate Use and Interpretation,” *Anesthesia Analgesia*, vol. 126, no. 5, pp. 1763–1768, May 2018, doi: 10.1213/ane.0000000000002864.
5. M. H. Kotb and R. Ming, “Comparing SMOTE Family Techniques in Predicting Insurance Premium Defaulting using Machine Learning Models,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 9, 2021, doi: 10.14569/ijacsa.2021.0120970.
6. V. L. Roger, “Epidemiology of Heart Failure,” *Circulation Research*, Aug. 2013.
7. R. C. Deo, “Machine Learning in Medicine,” *Circulation*, Nov. 2015.
8. J. Li, Y. Sun, J. Ren, Y. Wu, and Z. He, “Machine learning for in-hospital mortality prediction in critically ill patients with acute heart failure: A retrospective analysis based on MIMIC -Ⅳdatabases,” Research Square Platform LLC, Jan. 2024. Accessed: Apr. 06, 2025. [Online]. Available: https://doi.org/10.21203/rs.3.rs-3834698/v1
9. Q. Zhang, L. Xu, Z. Xie, W. He, and X. Huang, “Machine learning-based prediction of mortality in acute myocardial infarction with cardiogenic

shock," *Frontiers in Cardiovascular Medicine*, vol. 11, Oct. 2024, doi: 10.3389/fcvm.2024.1402503.

10. J. Jung, D. Kim, and I. Hwang, "Exploring Predictive Factors for Heart Failure Progression in Hypertensive Patients Based on Medical diagnosis Data from the MIMIC-IV Database," MDPI AG, Apr. 2024. Accessed: Apr. 06, 2025. [Online]. Available: https://doi.org/10.20944/preprints202404.1771.v1
11. J. Chen, L. Yang, J. Han, L. Wang, T. Wu, and D. Zhao, "Interpretable Machine Learning Models Using Peripheral Immune Cells to Predict 90-Day Readmission or Mortality in Acute Heart Failure Patients," *Clinical and Applied Thrombosis/Hemostasis*, vol. 30, Jan. 2024, doi: 10.1177/10760296241259784.
12. P. Xie *et al.*, "Development and Validation of an Explainable Deep Learning Model to Predict In-Hospital Mortality for Patients With Acute Myocardial Infarction: Algorithm Development and Validation Study," *Journal of Medical Internet Research*, vol. 26, p. e49848, May 2024, doi: 10.2196/49848.
13. B. Ru *et al.*, "Comparison of Machine Learning Algorithms for Predicting Hospital Readmissions and Worsening Heart Failure Events in Patients With Heart Failure With Reduced Ejection Fraction: Modeling Study," *JMIR Formative Research*, vol. 7, p. e41775, Apr. 2023, doi: 10.2196/41775.
14. T. Saito and M. Rehmsmeier, "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets," *PLOS ONE*, vol. 10, no. 3, p. e0118432, Mar. 2015, doi: 10.1371/journal.pone.0118432.
15. A. Johnson, L. Bulgarelli, T. Pollard, S. Horng, L. A. Celi, and R. Mark, "MIMIC-IV." Accessed: Apr. 06, 2025. [Online]. Available: https://physionet.org/content/mimiciv/2.1/
16. K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.

17. J. Benesty, J. Chen, Y. Huang, and I. Cohen, “Pearson Correlation Coefficient,” in *Springer Topics in Signal Processing*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1–4. Accessed: Apr. 13,2025.Available:https://doi.org/10.1007/978-3-642-00296-0_5
18. E. Acuña and C. Rodriguez, “The Treatment of Missing Values and its Effect on Classifier Accuracy,” in *Classification, Clustering, and Data Mining Applications*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 639–647. Accessed: Apr. 13, 2025. [Online]. Available: https://doi.org/10.1007/978-3-642-17103-1_60
19. S. Vinay, “STANDARDIZATION IN MACHINE LEARNING,” Delhi Technological University . Accessed: Apr. 13, 2025. [Online]. Available: https://www.researchgate.net/publication/349869617_STANDARDIZATION_IN_MACHINE_LEARNING
20. H. He and E. A. Garcia, “Learning from Imbalanced Data,” IEEE (Institute of Electrical and Electronics Engineers). Accessed: Apr. 13, 2025.[Online].Available:https://www.researchgate.net/publication/224541268_Learning_from_Imbalanced_Data
21. L. Breiman, A. Cutler, A. Liaw, and M. Wiener, “randomForest: Breiman and Cutlers Random Forests for Classification and Regression,” *CRAN: Contributed Packages*, Apr. 2002, doi: 10.32614/cran.package.randomforest.
22. GeeksforGeeks, “Random Forest Algorithm in Machine Learning,” *GeeksforGeeks*, Feb. 22, 2024. [Online]. Available: https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/
23. G. Ke *et al.*, “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” Microsoft Research. Accessed: Apr. 22, 2025. [Online]. Available: https://www.microsoft.com/en-us/research/publication/lightgbm-a-highly-efficient-gradient-boosting-decision-tree/
24. “Features — LightGBM 4.6.0.99 documentation.” Accessed: Apr. 22, 2025.[Online].Available:https://lightgbm.readthedocs.io/en/latest/Features.html
25. T. Chen and C. Guestrin, “XGBoost,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data*

Mining, New York, NY, USA: ACM, Aug. 2016, pp. 785–794. Accessed: Apr.22,2025.Available:https://doi.org/10.1145/2939672.2939785

26. S. Dayananda, “Logistic Regression - Sandun Dayananda,” *Medium*, Jul. 22, 2023. Accessed: Apr. 22, 2025. [Online]. Available: https://sandundayananda.medium.com/logistic-regression-55512384851b
27. M. P. LaValley, “Logistic Regression,” *Circulation*, vol. 117, no. 18, pp. 2395–2399, May 2008, doi: 10.1161/circulationaha.106.682658.
28. C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/bf00994018.
29. X. Huang, A. Khetan, M. Cvitkovic, and Z. Karnin, “TabTransformer: Tabular Data Modeling Using Contextual Embeddings,” arXiv.org. Accessed: Apr. 25, 2025. [Online]. Available: https://arxiv.org/abs/2012.06678
30. S. Widodo, H. Brawijaya, and S. Samudi, “Stratified K-fold cross validation optimization on machine learning for prediction,” *Sinkron*, vol. 7, no. 4, pp. 2407–2414, Oct. 2022, doi: 10.33395/sinkron.v7i4.11792.
31. J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms,” *Advances in Neural Information Processing Systems*, vol. 25.
32. R. Kohavi, “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection,” unknown. Accessed: May 05, 2025. [Online]. Available: https://www.researchgate.net/publication/2352264_A_Study_of_Cross-Validation_and_Bootstrap_for_Accuracy_Estimation_and_Model_Selection
33. C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, “A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data,” *Frontiers in Energy Research*, vol. 9, Mar. 2021, doi: 10.3389/fenrg.2021.652801.
34. “Hidden Technical Debt in Machine Learning Systems.” Accessed: May 05,2025.[Online].Available:https://research.google/pubs/hidden-technical-debt-in-machine-learning-systems

35. “OrdinalEncoder,” scikit-learn. Accessed: May 25, 2025. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html
36. I. Dey and V. Pratap, “A Comparative Study of SMOTE, Borderline-SMOTE, and ADASYN Oversampling Techniques using Different Classifiers,” in *2023 3rd International Conference on Smart Data Intelligence (ICSMDI)*, IEEE, Mar. 2023, pp. 294–302. Accessed: May27,2025.[Online].Available:https://doi.org/10.1109/icsmdi57622.2023.00060
37. G. Velarde *et al.*, “Tree boosting methods for balanced and imbalanced classification and their robustness over time in risk assessment,” *Intelligent Systems with Applications*, vol. 22, p. 200354, Jun. 2024, doi: 10.1016/j.iswa.2024.200354.
38. S. Varma and R. Simon, “Bias in error estimation when using cross-validation for model selection,” *BMC Bioinformatics*, vol. 7, no. 1, Feb. 2006, doi: 10.1186/1471-2105-7-91.
39. Y. Zheng and V. Stodden, “The Idealized Machine Learning Pipeline (IMLP) for Advancing Reproducibility in Machine Learning,” in *Proceedings of the 2nd ACM Conference on Reproducibility and Replicability*, New York, NY, USA: ACM, Jun. 2024, pp. 110–120. Accessed:May28,2025.[Online].Available:https://doi.org/10.1145/3641525.3663630
40. M. J. Bdair, “Enhancing Machine Learning Workflows: A Comprehensive Study of Machine Learning Pipelines,”
41. https://datascience.stackexchange.com/questions/26699/decision-trees-leaf-wise-best-first-and-level-wise-tree-traverse