Technical University of Crete
School of Electrical and Computer Engineering

Data Mining from Medical Databases and Machine Learning Based Mortality Prediction for Venous Thromboembolism, Myocardial Infarction and Ischemic Stroke

Diploma Thesis by
Stylianos Nikolakakis

Thesis Committee
Associate Professor Sotiris Ioannidis (Supervisor)
Professor Michael Zervakis
Professor Michail G. Lagoudakis

Chania, November 2023

Πολυτεχνείο Κρήτης
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Εξόρυξη Δεδομένων από Ιατρικές Βάσεις Δεδομένων και Πρόβλεψη Θνησιμότητας βάσει Μηχανικής Μάθησης για Φλεβική Θρομβοεμβολή, Έμφραγμα του Μυοκαρδίου και Ισχαιμικό Εγκεφαλικό Επεισόδιο.

Διπλωματική Εργασία του
Στυλιανού Νικολακάκη

Επιτροπή Πτυχιακής
Αναπληρωτής Καθηγητής Σωτήριος Ιωαννίδης (Επιβλέπων)
Καθηγητής Μιχαήλ Ζερβάκης
Καθηγητής Μιχαήλ Γ. Λαγουδάκης

Χανιά, Νοέμβριος 2023

Abstract

Intensive Care Unit (ICU) patients with arterial or venous thrombosis suffer from high mortality rates. Mortality prediction in the ICU has been a major medical challenge, for which several scoring systems exist, but lack in specificity. This study focuses on three target groups, namely patients with thrombosis, ischemic stroke or myocardial infarction. The main goal is to develop and validate interpretable Machine Learning (ML) models to predict mortality, while exploiting all available data stored in the medical record. To this end, retrospective data from one freely accessible database, eICU, were used. Well-established ML algorithms were implemented utilizing automated and purposely built ML frameworks for addressing class imbalance. Prediction of early mortality showed excellent performance in all disease categories, in terms of the area under the receiver operating characteristic curve (AUC–ROC()): Venus Thromboembolism (VTE) 0.87, Myocardial Infarction (MI) 0.95, Ischemic Stroke (IS) 0.90. The predictive model of mortality developed from 4,385 VTE patients ended up with a signature of 475 features, 10,543 patients with myocardial infarction using 317 features and 4,326 patients with ischemic testing on 338 features. Our model outperformed traditional scoring systems in predicting mortality.

Περίληψη

Οι ασθενείς στη Μονάδα Εντατικής Θεραπείας (ΜΕΘ) που πάσχουν από αρτηριακή ή φλεβική θρόμβωση υποφέρουν από υψηλά ποσοστά θνητότητας. Η πρόβλεψη της θνησιμότητας στη ΜΕΘ είναι μια πρόκληση για την ιατρική, καθώς υπάρχουν διάφορα συστήματα και εργαλεία για αυτόν τον σκοπό. Ωστόσο, η ανεπάρκεια στην ειδικότητα αυτών των συστημάτων αποτελεί πρόβλημα. Η παρούσα μελέτη επικεντρώνεται σε τρεις κύριες ομάδες ασθενών, δηλαδή ασθενείς με θρόμβωση, ισχαιμικό εγκεφαλικό επεισόδιο ή καρδιακή προσβολή. Ο κύριος στόχος είναι η ανάπτυξη και επικύρωση ερμηνεύσιμων μοντέλων Μηχανικής Μάθησης (ΜΜ) για την πρόβλεψη θνησιμότητας, εκμεταλλευόμενα όλα τα διαθέσιμα δεδομένα που αποθηκεύονται στο ιατρικό φάκελο. Για το σκοπό αυτό, χρησιμοποιήθηκαν αναδρομικά δεδομένα από μία βάση δεδομένων που είναι ελεύθερα προσβάσιμη, την eICU. Εφαρμόστηκαν καθιερωμένοι αλγόριθμοι ΜΜ χρησιμοποιώντας αυτοματοποιημένα και ειδικά κατασκευασμένα πλαίσια ΜΜ για την αντιμετώπιση της ανισορροπίας κλάσεων. Η πρόβλεψη πρόωρης θνησιμότητας παρουσίασε εξαιρετική απόδοση σε όλες τις κατηγορίες νόσων, όσον αφορά την περιοχή κάτω από το καμπύλο χαρακτηριστικών λειτουργίας παραλαβής (AUC-ROC): Φλεβική Θρόμβωση 0.87, Έμφραγμα του Μυοκαρδίου 0.95 και Ισχαιμικό Εγκεφαλικό επεισόδιο 0.90. Το προγνωστικό μοντέλο θνησιμότητας που αναπτύχθηκε από 4,385 ασθενείς με Φλεβική Θρόμβωση περιλαμβάνει 475 χαρακτηριστικά, ενώ για τους 10,543 ασθενείς με καρδιακή προσβολή χρησιμοποιήθηκαν 317 χαρακτηριστικά και για τους 4,326 ασθενείς με Ισχαιμικό Εγκεφαλικό επεισόδιο, χρησιμοποιήθηκαν 338 χαρακτηριστικά. Το μοντέλο μας υπερτερεί στην πρόβλεψη θνησιμότητας σε σχέση με τα παραδοσιακά συστήματα σκοράρισης.

Acknowledgements

I would like to thank all of my professors and colleagues. Firstly, it has been an honor working with Prof. Sotiris Ioannidis for giving me the chance to cooperate with him. Many thanks also to Prof. Vasiliki Danilatou for guiding me and providing consultations, as well as Dr. Despoina Antonakaki and Dr. Christos Tzagkarakis for providing me with the appropriate resources and knowledge. I would also like to thank the rest of my thesis committee: Prof. Michalis Lagoudakis and Prof. Michalis Zervakis. Last, but not least, I would like to thank my family for supporting me all these years and my friends for always supporting in me.

Contents

List of Tables

List of Figures

Abbreviations

The following abbreviations are used in this thesis:

APACHE	Acute Physiology, Age, and Chronic Health Evaluation
AUC	Area Under Roc Curve
CI	Conference Interval
CV	Cross validation
DNN	Deep Neural Networks
eICU	Electronic Intensive Care Unit
GNB	Gaussian Naive Bayes Classifier
GBM	Gradient Boosting Machine
GLR	Generalized Linear Regression
HIPAA	Health Insurance Portability and Accountability Act
ICD	International Classification of Diseases
ICUs	Intensive Care Units
IS	Ischemic Stroke
KNN	K Nearest Neighbor
LOS	Length of Stay
LR	Logistic Regression
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MI	Myocardial Infarction
ML	Machine Learning
MSE	Mean Square Error
NIHSS	National Institutes of Health Stroke Scale
PCC	Pearson's Correlation Coefficient
PESI	Pulmonary Embolism Severity Index
RF	Random Forests
ROC	Receiver Operating Characteristic
SHAP	Shapley Additive Explanation
SMOTE	Synthetic Minority Oversampling Technique
sPESI	Simplified Pulmonary Embolism Severity Index
SVM	Support Vector Machine
VTE	Venous Thromboembolism
XGB	Extreme Gradient Boost

Chapter 1

Introduction - Motivation

Arterial and venous thrombosis is a major medical problem today as it accounts for most of the deaths worldwide [22]. Thrombosis is defined as the presence of clots in the circulatory system, either arterial vascular system with more common clinical presentations myocardial infarction, ischemic stroke or in the venous system, also known as venous thromboembolism (VTE), which in the form of deep venous thrombosis mainly in the lower extremities, can cause dangerous complications such as pulmonary embolism.

Thrombosis is the leading global cause of morbidity and mortality, since it has been estimated to account for 1 in 4 deaths worldwide in 2014 [30]. More importantly, there is a constant increase in its prevalence, due to the ageing of the population, the increased prevalence of chronic diseases, such as cancer and hospitalizations and more recently the COVID 19 pandemic [6].

Arterial thrombosis occurs mainly as a complication of atherosclerosis, which in turn has as its main causes chronic hypertension, diabetes mellitus, obesity, hypercholesterolemia and smoking. While in the case of venous thrombosis, it is mainly due to conditions where there is stasis - pooling of blood within the vessel such as prolonged immobility, chronic hospitalization, pregnancy, thrombophilia and infection with COVID-19.

Thrombosis has severe physical and psychological complications [18], such as post-traumatic stress disorder, post-thrombotic syndrome, recurrence and even death. Recurrence and mortality are significant especially during the first 12 months. The early identification of the outcome of the patients and/or possible complications e.g., mortality or recurrence is essential, since the process will impact decision making regarding treatment and possibly the survival of the patients.

1.1 Problem Statement

Thrombosis (arterial or venous) is a phenomenon that occupies the medical industry a lot, due to it being a dangerous state that can prove deadly. As of now, a wide range of data that belongs to patients hospitilized with thrombosis is available. This is mainly because of the rapid technological development that provides us with various equipment which render the gathering of data feasible. As a result, the topic of the survey is the mortality prognosis of patients hospitalized in intensive care units (ICUs). Predicting mortality is crucial for assessing severity of illness and adjudicating the value of novel treatments. The solution that we present for the above problem is the use of supervised machine learning (ML) models. The goal is for the required model training to provide a specific outcome that concerns the categorization of the patients, depending on what kind of information is associated with each patient. In case of the data size being vast, the performance of the models is improved. A variety of classifiers has been used for the prediction of the patients' mortality. Afterwards, their results were compared in order to track the optimal model for the specific problem. Having plenty of big data in our disposal, it is essential that they be taken advantage of so that they can prove useful in future alike situations. As it was previously stated, the ML models that use many data during their training, are capable of categorizing the requested attribute with greater precision. As of that, the presence of ML models for problems of similar texture is of great importance. ML is a fundamental technology used for the processing of data that exceed the ability of the human brain to understand. As a result, its use for medicine researches has been established.

Our approach to patient prognosis is based on the creation of a machine learning pipeline that accepts as input the appropriate data sets, containing patient characteristics. The classifiers were chosen to be used differ from each other in the principal that follows. The categories to which the classifiers belong are included in probabilistic models (Gaussian Naive Bayes and Logistic Regression), which use the feature space (Support Vector machine and K Nearest Neighbors) and finally the models whose algorithm is based on tree creation (Random Forest and Extreme Gradient Boost). The specific data of patients that concern us were extracted from a large database and with the appropriate processing the final data sets for the training of ML were synthesized. [29]

Chapter 2

Related Work

One study related to prognosing patient mortality diagnosed with pulmonary embolism using machine learning algorithms, has been presented recently at the ECS Congress 2021- The digital experience[15] For this publication, patient data from three different hospitals in the UK have been harvested. The data derives from 1554 patients, with a mean age of 65, (47% male). The machine learning models that were used for the mortality prognosis are Random Forests and Extreme Gradient Boost and logistic regression evaluated by 5-fold cross validation. To evaluate the performance of the models they took into account the area under Receiver operating characteristic (ROC) curve metric (AUC). Using Random Forests and Extreme Gradient Boost (XGB), the results given were 0.85 [95% Conference Interval (CI): 0.80 - 0.90] and 0.82 [95% CI: 0.77 - 0.87], whereas with the Logistic Regression model, 0.83 [95% CI: 0.78 - 0.88]. The aim of their research was to compare the results of the previous ML models between and the simplified Pulmonary Embolism Severity Index (sPESI) which is a prognostic score for patients with thrombosis. Their approach didn't take into consideration patients in the ICU.

Another similar retrospective study that used ML models to predict 30 day all-cause mortality in patients with VTE, was presented at Chest Annual Meeting 2020[23]. The data consisted of 101 characteristics from 439 patients that were hospitalized with pulmonary embolism with a mean age of 61 years. The characteristics were demographic, laboratory, clinical, echocardiographic, and computed tomography reports. The prognostic models that were implemented were XGB, Gradient boosting machine (GBM), Random Forests and Deep Neural Networks (DNN) and Generalized Linear Regression (GLR). In this study the Pulmonary Embolism severity index (PESI) as well as the sPESI score were used as reference points for the comparison of the models and the AUC for performance efficient evaluation. Xboost model has the best performance of 0.922(95% [CI]: 0.890-0.954 whereas GBM had

an AUC 0.911[0.875 - 0.947]), DNN 0.868 [0.833-0.903], GLM 0.865[0.816 - 0.914], RF 0.859 [0.843 - 0.875]. For PESI 0.805 [0.749 - 0.851], and sPESI 0.754[0.741 - 0.846]. PESI sPEsi are traditional medical scores that have been developed from statistical methods.

A survey dealing with the search for prognosis of mortality and morbidity of patients who had suffered a stroke in the Stroke Unit of a European Tertiary Hospital 3 months after their admission[16] was carried out in 6022 patients out of which 4922 had suffered an ischemic stroke, (mean age 71.9 $\pm$ 13.8 years) and 1100 intracerebral hemorrhage, (mean age 73.3 $\pm$ 13.1). Study results were generated using Random Forest by combining or isolating patient categories. In the patients with an ischemic episode, the AUC had a score of 0.909 $\pm$ 0.032, while for all patients the AUC was 0.904 $\pm$ 0.025. In the experiments, 68 features from various categories were used and the most important variables were National Institutes of Health Stroke Scale (NIHSS) at 24, 48 h and axillary temperature at admission.

Another study that focuses on incidents regarding cerebral hemorrhage was published in Frontiers in Neurology, 20 January 2021 [25]. Their approach concerned 760 patients, from the MIMIC-III database, with mean age of 68.2 years and typical deviation of 15.5. Out of those patients, 383 passed away in the hospital, whereas the rest 377 survived. This experiment used Acute Physiology and Chronic Health Evaluation II(Apache II) score as a point of comparison. Hyperparemeter tuning was applied by using Grid-Search Cross validation and evaluate model identification performance. They focused on accuracy and receiver operating characteristic(ROC) curve. 72 variables within the first 24 h after ICU admission were used for the training of the model. For the survey's sake, 6 different ML algoriths(KNN, DTs, gs-Forest, AdaBoost, neural network and Random Forest) were put to use. Out of those algorithms, RF achieved the best performance with an AUC of 0.819. The performances of the rest of the models are the following: 0.725, KNN AUC:0.6, DT AUC: 0.617, NN AUC: 0.655, AdaBoost AUC:0.671. which ,as it was proven, outperformed APACHE II score.

A study that regarded short and long-term mortality prediction after an acute ST-elevation myocardial infarction (STEMI) in Asians was published in PLoS ONE journal 2021[5]. It studied 3 use cases , patients' mortality in the hospital, patients' mortality within 30 days and patients' mortality within a year. They focused on 6299 patients for their hospital model development. For the training of the models, 50 variables were considered. Mean age was 55.8 years (SD 11.5), survivals 961 (94.6%) and dead 338 (5.4%). 5417 were male. Various ML Algorithms, such as SVM, LR and RF, were used. The ML algorithms results for the patients at the hospital are: AUC: 0.88(0.846–0.910), for SVM, AUC: 0.87 (0.832–0.907) ,for RF and AUC: 0.89

(0.861–0.920) for LR. These results accomplish Outperformed Thrombolysis in Myocardial Infarction (TIMI) risk score

This study by Jun Ke et al. (2022) [20] investigated in-hospital mortality prediction models for patients with acute coronary syndrome. The study included 6,482 patients, and the in-hospital mortality rate was found to be 1.88%. The researchers employed logistic regression, gradient boosting decision tree, random forest, and support vector machine models to analyze the data. The performance of these models was evaluated based on the AUC metric. The main objective of this study was to develop accurate prediction models for in-hospital mortality in patients with acute coronary syndrome. The authors explored various machine learning models and assessed their performance based on their AUC scores. A higher AUC indicates better model performance in predicting the outcome of interest. The results revealed that all four models performed well, with AUCs ranging from 0.884 to 0.918.

In recent years, several studies have been conducted to explore the use of machine learning (ML) in predicting major adverse cardiac events (MACEs) in patients with acute myocardial infarction (AMI). One such study by Changhu Xiao et al., (2022) [36] aimed to assess the effectiveness of ML in predicting MACEs through a retrospective analysis. The study utilized a dataset of 500 patients who had undergone successful percutaneous coronary intervention for AMI. The researchers compared the predictive ability of six ML models to logistic regression (LR) analysis, which used 24 clinical variables. The study found that Killip classification, drug compliance, age, and creatinine and cholesterol levels were all independent predictors of MACEs. The random forest (RDF) model was identified as the best-performing model in predicting MACEs, with an accuracy rate of 0.734 and an area under the curve of 0.749. The study concluded that ML methods could be a promising tool for selecting optimal predictors and improving clinical outcomes in patients with AMI. These findings add to the growing body of literature that supports the potential of ML in predicting MACEs in patients with AMI.

Table 2.1: Summary of the related work

Reference	Proposed	Models	Datasets	Split	CV	Hyperparameter Tuning	Evaluation Metrics	Results
ECS Congress 2021	prediction mortality of patients with VTE	RF XGB	1.554 patients	70-30	5-fold CV	NO	AUC	RF best results AUC:0.85 vs sPESI score AUC:0.75
Chest Annual Meeting 2020	prediction early mortality (30-days) of patients with PE	GBM XGB DNN GLR	439 patients 101 features		5-fold CV		AUC	XGB best results AUC:0.92 vs PESI score AUC 0.8 vs sPESI score AUC:0.75
Scientific Reports volume 11(2021)	prediction mortality - morbidity of patients with stroke 3-months after admission	RF	6022 patients 65 features				AUC	RF best results AUC:0.9
Front. Neurol., 20 January 2021	Prediction Mortality of patients with Cerebral Hemorrhage	NN AdaBoost gcForest KNN DT RF	760 patients 72 features		10-fold CV	YES	AUC	RF best results AUC:0.82 vs APACHE II AUC:0.423
PLoS One. 2021 Aug	Prediction Mortality in hospital of patients with STEMI	SVM LR RF	6,299 patients 50 features	70-30	10-fold CV		AUC	AUC 0.88 vs TIMI score AUC:0.81
American Journal Emergency Medicine March 2022	Prediction Mortality in hospital of patients with ACS	LR GBDT RF SVM	6,482 patients 29 features	70-30		YES	AUC	LR AUC 0.88 GBDT AUC:0.92 RF AUC 0.91 SVM AUC 0.89
Journal Cardiovascular Development and Disease February 2022	Prediction Mortality in Patients with AMI	LR DT NB SVM RF GB	408 patients 41 features	60-40	5-fold CV		AUC Accurasy f1 score	LR AUC 0.72 DT AUC:0.66 NB AUC 0.73 SVM AUC 0.72 RF AUC 0.75 GB AUC 0.74
This Thesis	Prediction Mortality of patients with VTE - MI - IS	GNB LR KNN SVM RF XGB	VTE(4,385 patients 475 features) MI(9,656 patients 317 features) IS(3,866 patients 338 features	80-20 stratified	5-fold CV	YES using SMOTE	AUC Accuracy Specificity Sensitivity	Best Model Results: VTE(LR AUC:0.91) MI(XGB AUC:0.95) IS(XGB AUC:0.90)

Chapter 3

Data Source and Methodology

3.1 Data Source

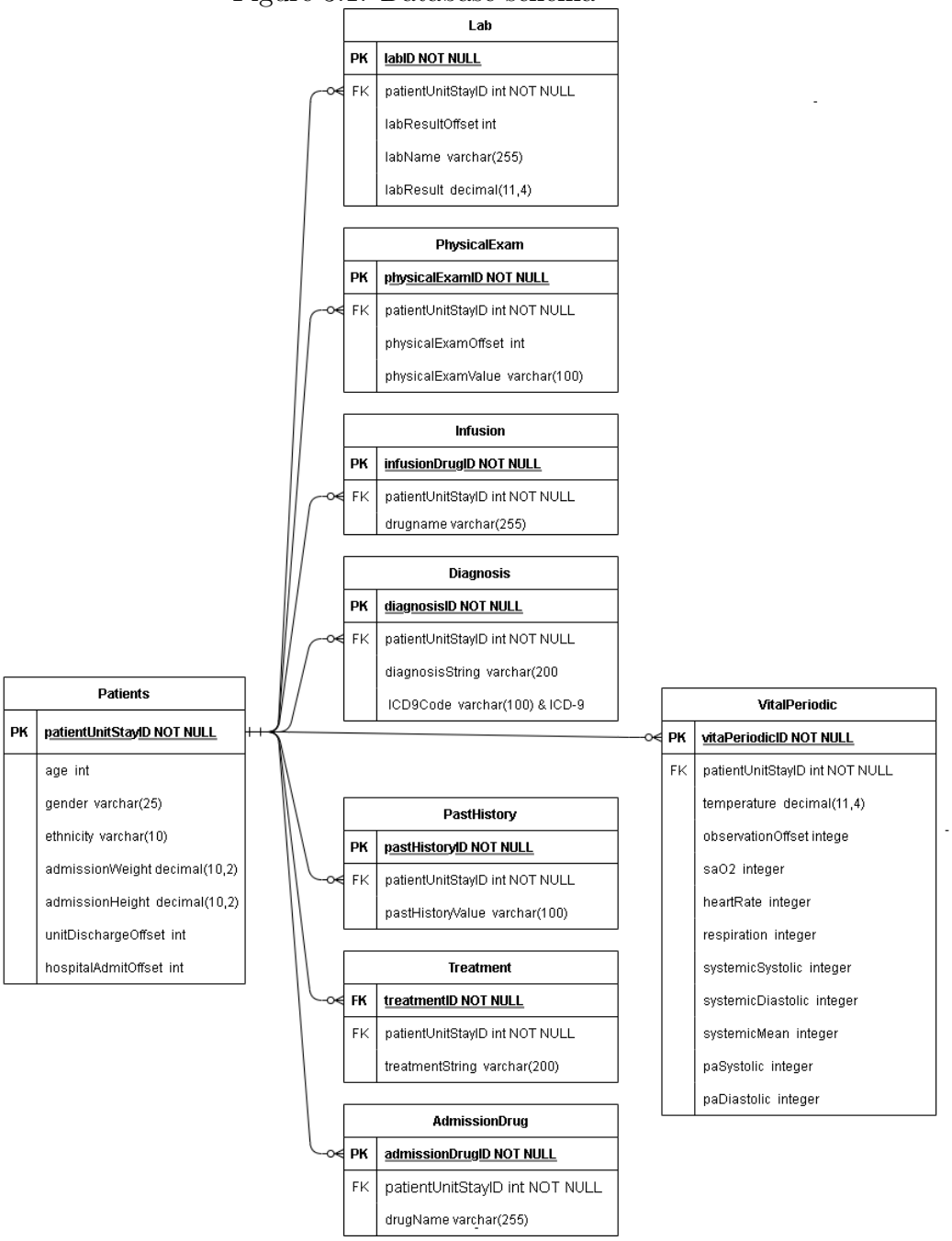
In the current study the data were harvested from Electronic Intensive Care Unit (eICU), a large multi-center critical care database for 200.859 admissions, 139,367 unique patients, admitted between 2014 and 2015, to ICUs monitored by eICU Programs across the United States, offered by Philips Healthcare collaborated with MIT Laboratory for Computational Physiology. The database contains vital sign measurements, care plan documentations, treatments information, diagnosis in formations, severity of illness measures, labs and more.This database was created in accordance with Health Insurance Portability and Accountability Act (HIPAA) standards and data access was approved by PhysioNet. Patient data were de-identified. [28]

3.2 Data Description

The eICU Database [28] provides us with various comma-separated value (CSV) files that describe 31 distinct tables, related to clinical information during the patients' hospitalizations. We decided to work on the following categories, the demographic, laboratory, vital data, treatments, diagnosis, infusions and various medical scores such as Acute Physiology, Age, and Chronic Health Evaluation (APACHE) score. Each csv file reconstructs a specific category and contains the unique value of each patient in ICU and the information that describe him. For each admission in the ICU, a "patientUnitStayID" key that shows whether the patient is generated. A patient could have been admitted to the ICU before. In our case, every admission is considered as a "new" patient because every time a patient returns to the ICU a new key is needed. A Reference is made below separately in every

table that has been used, as well as the attributes that we chose to use in our study. The structure of the database is shown in Figure 3.1 .

Figure 3.1: Database schema



Patient table (3.1) includes the demographic information of the patients like age or gender as well as the date on which they were admitted and discharged from the ICU.

Table 3.1: Patient Table that contains demographic data

Name	Datatype	Comment	Key
patientUnitStayID	int	surrogate key for ICU Stay	PK
age	int	patient's age	
gender	varchar(25)	patient's gender	
ethnicity	varchar(10)	patient's ethnicity	
admissionWeight	decimal(10,2)	patient's admission weight (kg)	
admissionHeight	decimal(10,2)	patient's admission height (cm)	
unitDischargeOffset	int	patient's discharged time from icu (min)	
hospitalAdmitOffset	int	patient's admitted time from icu (min)	

admissionDrug Table (3.2) describes the medication prescription of each patient, before they were admitted to the ICU. This table contains extra information about the drugs, for example the dosage and the date on which it was prescribed. In our research we took into account only the drug names.

Table 3.2: AdmissionDrug Table

Name	Datatype	Comment	Key
patientUnitStayID	int	surrogate key for ICU Stay	FK
drugName	varchar(255)	name of the selected admission drug	

Diagnosis table (3.3) is where the diagnosis of each patient is being recorded during their stay in the ICU. Moreover the ,International Classification of Diseases (ICD), ICD9 code, which represents the diagnosis code, is included which represents the diagnosis code. The focus area is the diagnosis of arterial and venous thrombosis. From the patients diagnosed with arterial thrombosis we chose to keep two categories. Those who have suffered with strokes and those with heart attacks. The ones with strokes are characterized by the ICD9 code (434,435) whereas the others by the ICD9 code (410, 411).

Table 3.3: Diagnosis Table

Name	Datatype	Comment	Key
patientUnitStayID	int	surrogate key for ICU Stay	FK
diagnosisString	varchar(200)	the full pathstring of the diagnosis	
ICD9Code	varchar(100)	ICD-9 code for the diagnosis	

Infusion Drug table (3.4) contains information about the infusion of drugs to the patients during their stay at the ICU, imported from the nursing flowsheet (entered either manually or interfaced from the hospital electronic health report system). We only kept the name of the drug used, the same way we went about utilizing the "admission drug" category.

Table 3.4: InfusionDrug Table

Name	Datatype	Comment	Key
patientUnitStayID	int	surrogate key for ICU Stay	FK
drugname	varchar(255)	picklist name of the infusion	

Treatment table (3.5) allows users to document specific active treatments for the patients in a structure format . From all the information given we only utilize the "treatmentString".

Table 3.5: Treatment Table

Name	Datatype	Comment	Key
patientUnitStayID	int	surrogate key for ICU Stay	FK
treatmentString	varchar(200)	the path of the treatment	

In the lab Table (3.6), we show all the laboratory tests that were conducted to each patient. They have been mapped to a standard set of measurements. In our case we keep the lab name and the lab result.

Table 3.6: Lab Table

Name	Datatype	Comment	Key
patientUnitStayID	int	surrogate key for ICU Stay	FK
labResultOffset	int	time period lab value was drawn	
labName	varchar(255)	the picklist name of the lab	
labResult	decimal(11,4)	the numeric value of the lab	

The pastHistory table (3.7) contains patient's pasthistory information and the physicalExam Table (3.8) physical exams.

Table 3.7: PastHistory Table

Name	Datatype	Comment	Key
patientUnitStayID	int	surrogate key for ICU Stay	FK
pastHistoryValue	varchar(100)	structured picklist of available past history items	

PhysicalExam table (3.8) contains the data displayed in the Constitutional Data field and the selection criteria for this data. The values for heart rate, blood pressure, temperature, respiratory rate and O2 sat include the 24 hour range as well as the current values.

Table 3.8: PhysicalExam Table

Name	Datatype	Comment	K
patientUnitStayID	int	surrogate key for ICU Stay	F
physicalExamOffset	int	minutes from unit admit time that the physical exam item was entered	
physicalExamValue	varchar(100)	Structured picklist of available of physical exam items	

Patients in the ICU are under constant monitoring and measurements of their vital indications are being recorded. These measurements are taken every minute in average and in the vitalPeriodic table (3.9) the intermediate values in a five minutes interval are being recorded.

Table 3.9: VitalPeriodic Table

Name	Datatype	Comment	Key
patientUnitStayID	integer	surrogate key for ICU Stay	FK
observationOffset	integer	number of minutes from unit admit time	
temperature	decimal(11,4)	patient's temperature value	
saO2	integer	patient's saO2 value	
heartRate	integer	patient's heartRate value	
respiration	integer	patient's respiration value	
systemicSystolic	integer	patient's systemicSystolic value	
systemicDiastolic	integer	patient's systemicDiastolic value	
systemicMean	integer	patient's systemicMean value	
paSystolic	integer	patient's paSystolic value	
paDiastolic	integer	patient's paDiastolic value	
paMean	integer	patient's paMean value	

3.3 Data Preprocessing

Data prepossessing is essential for the training of the ML models. The algorithms rely on the data required for the solution of a specific problem, in order to draw an outcome. The attributes, on which the data are recorded, are called features or characteristics. Some frequent problems that can be tracked in the data consist of missing values, the presence of noisy data and

other inconsistencies that can be observed. In order for the above problems to be resolved, data prepossessing is taking place for the smoother and more efficient functionality of the algorithms. For example, during the stage of their training, the algorithms use numeric data exclusively. That means that in cases of the existence of features containing string indexes, their values are replaced by either "0" or "1"(depending of whether they are included in the information or not).

The prior requirement for particular ML algorithms is the achievement of meaningful results and hence data transformation and data selection are applied for the performance's improvement. It is possible that interdependent characteristics exist in the dataset. This means that their information is associated , something which is unnecessary and makes training state more difficult

We then have to deal with their proper pre-processing for the creation of three data sets VTE, Myocardial infarction (MI) and Ischemic stroke (IS). In the order given these datasets describe patients with VTE, MI and IS respectively. The purpose of preprocessing is to create the appropriate features with the correct format so that the data sets are prepared to be compatible with the respective models. This process helps the models perform their operation without errors that render them unreliable for their performance.

An important feature extracted from table patient, is the length of stay (LOS), which reflects the time duration for which a patient was hospitalized in the intensive care unit. For the extraction, the hospitalAdmitOffset was subtracted from the unitDischargeOffset to get the los.

The eICU tables describe the patient's condition at the time they were admitted to the ICU. Tables (3.3) (3.5), (3.7) describe patient information from specific path attributes. Because these paths are complex and there are often more than one that contain the same information in a different sequence, we focused on identifying specific patterns in order to replace them with values that describe the path monosyllabic ally. The Tables (3.12-3.14) show the final form of the attributes after their replacement.

For laboratory, vital and physical exams data, the examinations that were done for each patient were grouped according to the time period in which they were performed. More specifically, the data were collected, the first 48 hours from the moment the patient had joined the ICU, and each examination was divided into 6 hour period. That means that we have the time values t ($t_1 = 0h - 6h, ..., t_8 = 42h - 48h$). For each time period t, we have created the characteristics based on the first measurement, the last measurement and the average of all the measurements from the examinations during this period.

A subset of medications was selected from the multitude of medications

listed for patients in the tables (admissionDrug, infusiondrug), as emphasis was placed on medications that play an essential role in treating patients rather than the wide range of medications. Table (3.11) describes the aggregated features after merging the tables (3.1-3.9).

Table 3.10: Demographic-clinical characteristics of patients from eICU Table

Characteristic	VTE	MI	SI
Overall patients	4.385 PE: 2.739(62.4%) DVT: 2,220 (50.6%)	10543	4326
Sex			
Female	2115	3835	2114
Male	2268	6707	2211
Uknown	1	-	1
Ethnicity			
Caucasian	3386	8327	3319
Afro-American	577	926	470
Native-American	15	34	14
Asian	36	167	67
Hispanic	165	371	219
Other	206	718	237
Average age (SD)	62.16(16.45)	66.01(13.35)	68.48(14.31)
Minimum age	15	16	19
Maximum age	90	90	90
Los (days)			
Average Los (SD)	11.12(11.9)	6.2(10.06)	8.01(15.03)
Median Los	7.13	3.73	5
Mortality			
Alive(%)	3838(87.5%)	9656(91.6%)	3866(89.4%)
Dead(%)	547(12.5%)	887(8.4%)	460(10.6%)

Table 3.11: Table that contains all features

Group	Description	Number of Features
Patient	Basic demographicinformation, LOS, discharge status	11
Diagnosis	Diagnoses documented during ICU stay	61
Lab	Laboratory tests	80
PhysicalExam	Vital signs	8
VitalPeriodic	Vital signs	10
AdmissionDrug	Medications taken prior to ICU admission	36
Infusion	Medications Transfusions Parenteral	30
Treatment	Medications	52
PastHistory	Past history of chronic diseases	111

3.3.1 Imputation

Missing data is a frequent problem in medical databases. In order to handle this situation we approached the missing data using imputation methods. Imputation is the process of missing data replacement with substituted values. Imputation preserves all cases by replacing missing data with an estimated value based on other available information. The most common methods for dealing with numerical missing data is the mean imputation, replacing missing values of a variable with the mean of known values for that variable. Mode imputation replaces missing values of a categorical variable with the mode of non-missing cases of that variable.

3.3.2 Correlation

Efficiency of machine learning models is determined from data which are used, during their training. At pre-processing stage, important steps must be taken in order to select the features that are necessary and as fewer as possible with the purpose of reducing the computation cost. Consequently we focus on correlation analysis.

Correlation analysis is a method which determines the strength of a relationship between two variables. There is the possibility that some attributes are linear correlated, thus we adopted Pearson's correlation coefficient (PCC) for detecting them. The PCC was applied on numerical attributes and is measured by the following mathematical equation.

$$\mathbf{r_{X,Y}} = \frac{\sum_{\mathbf{i=1}}^{\mathbf{N}}(\mathbf{x_i} - \bar{\mathbf{X}}_\mathbf{i})(\mathbf{y_i} - \bar{\mathbf{Y}}_\mathbf{i})}{\sum_{\mathbf{i=1}}^{\mathbf{N}}(\mathbf{x_i} - \bar{\mathbf{X}}_\mathbf{i})^\mathbf{2} \sum_{\mathbf{i=1}}^{\mathbf{N}}(\mathbf{y_i} - \bar{\mathbf{Y}}_\mathbf{i})^\mathbf{2}} \tag{3.1}$$

where:
$\mathbf{n}$ is the size of sample
$\mathbf{x_i}$ the i_{th} data values
$\mathbf{y_i}$ the i_{th} data values
$\bar{\mathbf{X}}, \bar{\mathbf{Y}}$ the average values of x,y

Values of PCC determine the association between two variables. If value (r) is close to zero that means there is a low association, otherwise the value close to $-\mathbf{1}$ or $+\mathbf{1}$ indicates a strong linear association between two variables. For applying PCC, the following requirements must be satisfied on variables (a) linear relationships, (b) independent, (c) normally distributed, (d) continuous random variables. [2]

3.3.3 Data Normalization

The efficiency of machine learning could be determined by the crowed of independents features (inputs and targets). This implies the use of different types of techniques to prepare those features to be suitable for the training model stage. Data Normalization or Feature scaling is a method which normalizing the range of independents features. Many machine learning algorithms at training stage use euclidean distance between two points for calculations. Thus, could cause a decrease in efficiency of the algorithms if a specific feature has wide range of values.

Data normalization is crucial to avoid the previous phenomenon. By using normalization we achieved to restrict this wide range to a particular range or decrease the original range of values, specifying where applicable. Summarizing the previous ones, the main goal of feature scaling is to change the numerical attributes or features of data to use a common scale without losing information. Normalization has plenty of techniques, couple of them are Z-score Normalization (Standardization), Min-Max Normalization (Rescaling) [19]

1.Standardization
Standardization is a technique which uses the mean and standard deviation for each feature individually. The produced data has zero mean and a unit standard deviation. The values are not restricted to a particular range but the spectrum is significantly reduced.

$$\mathbf{x'} = \frac{\mathbf{x_i} - \mu}{\sigma_\mathbf{j}} \tag{3.2}$$

where:

$\mathbf{x_i}$ is the i_{th} data values
μ is the mean of X
$\sigma_\mathbf{j}$ is the standard devition of j-th attribute

2.Rescaling
Min-Max Normalization in contrast with standardization, uses the minimum and maximum values of each features and rescales them to a new range of values. If a features has constant value, it can be removed because does not provide any information to the machine learning model.

$$\mathbf{x'} = \frac{\mathbf{x_i} - \mathbf{min}(\mathbf{x})}{\mathbf{max}(\mathbf{x}) - \mathbf{min}(\mathbf{x})} \tag{3.3}$$

where:
$\mathbf{min}(\mathbf{x})$ is the minimum value of x
$\mathbf{max}(\mathbf{x})$ is the maximum value of x

Chapter 4

Machine Learning and Methodology

4.1 Classification Methods

Machine learning is a procedure in which a machine is trained with data using a specific algorithm so that it gets more efficient with time. The goal behind that is for the machine to be capable of returning the most efficient results possible. Given that the data used are priorly categorized, machine learning methods are used for training the models. These kinds of methods are described as supervised learning algorithms.

Classification is defined as the problem of a new observation should be included. The machine is tasked with taking the right decision on the above problem. Binary classification is when the classes to be categorized are 2, whereas in Multi-class classification' case, the classes are more than 2. The ML algorithms tasked with categorizing the observations, are called Classifiers.

Binary classifier is used for our research, as the label of the samples which we want to be assigned regards the outcome of the patient's life. The models that we build include the Gaussian Naive Bayes Classifier, K Nearest Neighbor, Support Vector Machine, Logistic Regression, Random Forests and Extreme Gradient Boost.

4.1.1 Gaussian Naive Bayes

Naive Bayes classifier is a supervised learning algorithm that belongs to the probabilistic classifiers and is based on the Bayes theorem [1]. To state his theorem, Naive Bayes sets as a basic precondition that the features have a

strong independence between them. This algorithm has proven to be quite effective in applications such as test classification and a variety of medical diagnosis. [37, 34] The main use of the Naive Bayes algorithm is the creation of a probabilistic model, aiming to find the highest a posteriori probability. While having a feature vector x (with $x = x_1, ..., x_n$) as an entrance, it is checking to find the different classes that are requested C (C_1 through C_k) when the probability is maximized). Therefore for every class K the following probability is calculated:

$$\mathbf{P(C_k/x_1, ..., x_n) \propto P(C_K) * \prod_{i=1}^{n} P(x_i/C_k)} \tag{4.1}$$

The result of every probability $\mathbf{P(\frac{x_i}{k})}$ can be close to zero. While this relation develops, we can see that the result is very close to zero because of the multiplication process. This means that there is a chance of underflow which leads to values so close to zero that the computer cannot interpret them in order give useful results. To have this underflow issue resolved,

$$\mathbf{\hat{y} = argmax P(C_K) * \prod_{i=1}^{n} P(x_i/C_k)} \tag{4.2}$$

we convert the relation (3.4) in logarithmic scale so that the processes can be processed.

$$\mathbf{\hat{y} = argmax \ln P(C_K) * \prod_{i=1}^{n} P(x_i/C_k)} \tag{4.3}$$

$$\mathbf{\hat{y} = argmax \ln P(C_K) + \sum_{i=1}^{n} \ln P(x_i/C_k)} \tag{4.4}$$

where

$$\mathbf{P(x_i/C_k) = \frac{1}{\sqrt{2\pi\sigma C_k}} e^{\frac{-(x_i-\mu_k)^2}{2\sigma_k}}} \tag{4.5}$$

4.1.2 K Nearest Neighbors

K Nearest Neighbor (KNN) is a non-parametric and supervised algorithm that can be used for both classification and regression problems [12]. One important benefit of KNN is that it does not assume any particular distribution of the data in space. To categorize a sample, KNN uses the Euclidean distance between the specific observation and the data points:

$$D(x,p) = \sqrt{\sum_{i=1}^{n}(x_i - p_i)^2} \tag{4.6}$$

Here, x and p are two data points, and n is the number of dimensions or features of the data. KNN accepts the number of nearest neighboring data points to each observation, K, as an argument. In classification problems, KNN uses majority voting among the classes that compete for the observation, to determine the category to which the observation belongs, in case of a tie in the number of data points that represent a class [24].

4.1.3 Support Vector Machine

Support Vector Machine (SVM)[11] is one of the most popular supervised learning algorithms. It is used for Regression problems but mainly for binary classification problems. The samples are recorded in the n-dimensional space, where n is specified by the number of the traits in their whole

The goal of the SVM lies in the finding of a hyper-plane which distinctly classifies the data points, in order to separate the 2 classes in such a way that the margin(ie. distance between data points of both classes) takes its maximum value.

The equation of the hyper plane is shown below:

$$\mathbf{f(x) = w^T x + b} \tag{4.7}$$

If a linear separation does not exist between our data, it is necessary that they undergo conversion in order for them to be shifted from their initial space to a higher dimension feature space. SVM uses numerous mathematical functions(kernel functions) for the conversion of the traits. Some of the basic kernels include sigmoid ,rbf ,linear and polynomial.

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

4.1.4 Logistic Regression

Logistic Regression (LR)[13] is a natural probabilistic algorithm, like GNB. LR belongs to the supervised learning algorithms and it is mainly used for binary classification problems. The type of the independent variables(X) can be either categorical or numerical, whereas the dependent variables(Y) are comprised of just categorical values. In binary classification's case, the Y variables are binomial distributed ($Y : \Omega \rightarrow 0, 1$), meaning P(Y=1/X) true and P(Y=0/X) false [35].

In order for a linear decision boundary to be created, LR applies the logit function, also known as sigmoid function, which maps the probabilities (ranging from 0 to 1) to the set of all real numbers, R. The logit function is defined as described bellow:

$$\ln \frac{\mathbf{p}}{\mathbf{1-p}} = \mathbf{z} \tag{4.8}$$

where:
$\mathbf{p}$ depicts the success probability
$\frac{\mathbf{p}}{\mathbf{1-p}}$ is defined as odds and depicts the degree of the success probability
$\mathbf{z} = \beta_0 + \beta_{1x} + ... + \beta_{\mu x}$
$\beta_\mathbf{i}$ are the regression coefficients, computed via maximum likelihood

If we further expand the relationship (4.8), then it is concluded that:

$$\mathbf{p_i} = \frac{\mathbf{e^z}}{\mathbf{1 + e^z}} \tag{4.9}$$

or equivalent

$$\mathbf{p_i} = \frac{\mathbf{1}}{\mathbf{1 + e^{-z}}} \tag{4.10}$$

4.1.5 Random Forest Classifier

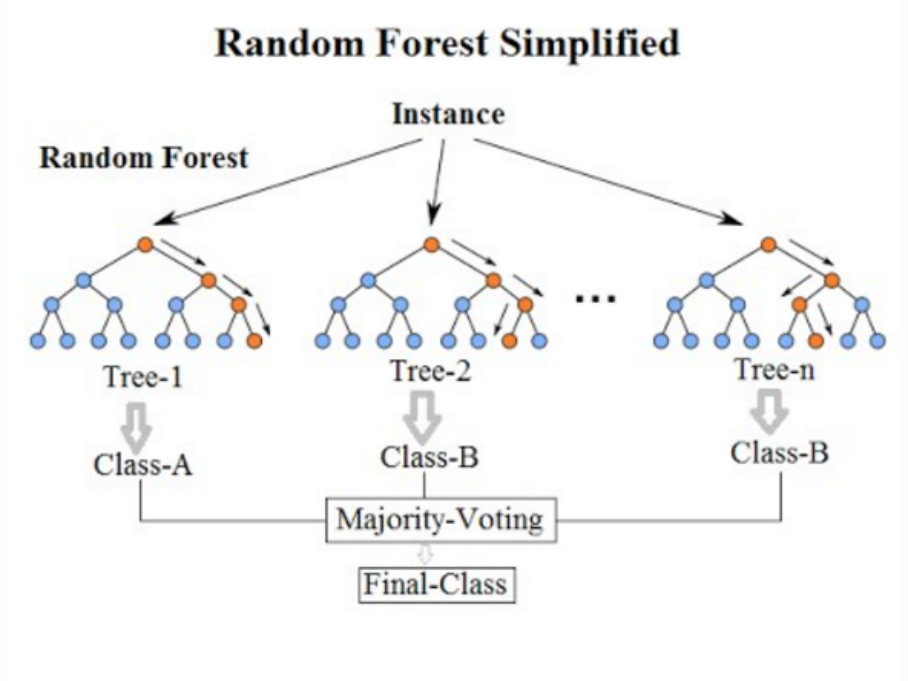
Random Forest[7] proposed by Leo Breiman is a classifier that creates decision trees or classifier trees. Random Forest creates an ensemble of decision trees. Each tree is trained on a random subset of features and samples from the training set. Each one of those gives an outcome prediction for the characteristic for which the search is carried out, with an aim to find the category to which it belongs to. For the classification tasks, the assignment of the charecteristic's label, comes up from the class that was chosen by most of the decision trees.

The creation of the tree classifiers is based on the algorithm called bagging or bootstrap aggregation. Bagging algorithm it selects a random subset

of both features and samples with replacement to create each tree with M characteristics. To continue with a random number is being chosen m<< M out of M characteristics and the best split gets to be used in the node. Afterwards, every tree develops at the greater degree possible. Throughout the duration of the construction of the Random Forest, number m stays the same. After that, the value of the classify sample is being chosen with the help of the majority voting (figure 4.1).

Those data that weren't chosen are named out of bag and they form nearly one-third of the original data set. Out-of-bag samples are used as a validation set to estimate the performance of the Random Forest model during training.

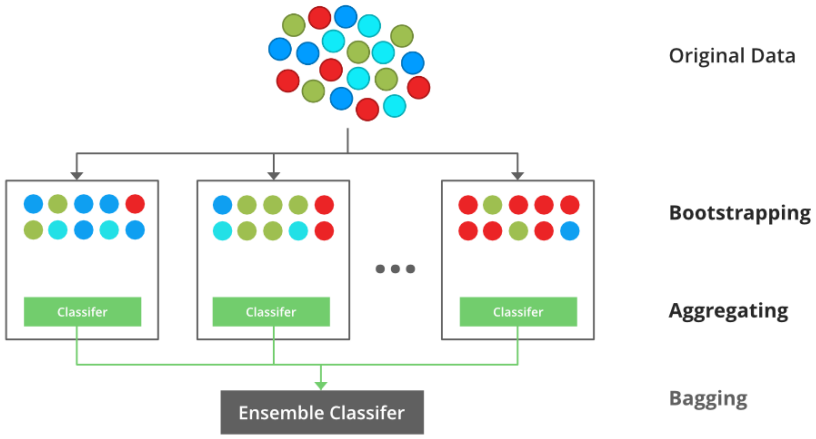
Figure 4.1: Random Forest diagram



4.1.6 Extreme Gradient Boost Classifier

The Extreme Gradient Boosting (XGB)[10] algorithm is based on the logic of ensemble learning, more specifically Boosting. XGB is an evolution of Gradient Boosting, which uses the Gradient Descent optimization method to minimize errors on a collection of weak models that are generated serially.

Figure 4.2: XGB Bugging diagram



According to the ensemble learning method, a collection of weak classifiers can form the basis for creating a strong classifier. A weak classifier is defined as any classifier whose performance is even slightly better than that of random selection. Unlike the Bagging method, in which weak classifiers are generated in parallel to arrive at decisions that will participate in voting to select the dominant one, the Boosting method generates weak classifiers serially. In the case of XGB the weak classifiers are in the general case decision trees, but generally there is also the option of being one of Tree, DART, Linear or Tweedie Regression. Each weak classifier is assigned a weight that is related to its prediction accuracy. Also, after each iteration a weight is assigned to each instance as well. In the case of XGB, weak classifiers are generally decision trees, but there is also the option of using Tree, DART, Linear, or Tweedie Regression. Each weak classifier is assigned a weight that is related to its prediction accuracy. Additionally, after each iteration, a weight is assigned to each instance. If the instance is not classified correctly, its weight is increased. The selection of snapshots to participate in each iteration depends on the weight of each snapshot. In this way, there are more iterations of the process with the instances that were misclassified by the previous weak classifiers, in an attempt by the model to persist in solving the more difficult cases. After adding each new weak classifier, the weights of each weak classifier and each instance are recalculated (see figure 4.1).

4.2 Validation and Machine Learning Evaluation

4.2.1 Metrics

A basic set of performance measurements is often used for the evaluation of the efficiency of machine learning algorithms. The kind of metrics that will be used for the evaluation of the algorithms is based on what kind of problems we face. If we face classification models, then the basic metrics that are used consist of accuracy, precision, recall, F1-score and area under the ROC curve. In case of the models belonging to the regression category, either the mean square error (MSE), the mean absolute error (MAE) or the R^2 is used.

In our case, we face machine learning models that approach medical predictions by classifying (for eg. predict which patients will leave or die), and in medical binary classification matters, in general, the models are evaluated via the use of the area under the Reciever operating characteristics curve (ROC), confusion matrix, accuracy, sensitivity and specificity.

Confusion matrix

This matrix is comprised of the results regarding the predictions that the machine learning models provide us with. It is a 2X2 matrix which includes the above piece of information:
a) the number of true positive
b) the number of false positive
c) the number of true negative
d) the number of false negative
By using the confusion matrix, we can calculate the bellow metrics.

Accuracy

Accuracy indicates how correctly our model predicts, is defined by the following formula.

$$\textbf{Accuracy} = \frac{\textbf{TP} + \textbf{TN}}{\textbf{TP} + \textbf{FP} + \textbf{TN} + \textbf{FN}} \tag{4.11}$$

ROC curve

A ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

True Positive Rate False Positive Rate True Positive Rate (TPR) is a synonym for recall and is therefore defined as follows:

$$\mathbf{TPR} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}} \tag{4.12}$$

False Positive Rate (FPR) is defined as follows:

$$\mathbf{FPR} = \frac{\mathbf{FP}}{\mathbf{FP} + \mathbf{TN}} \tag{4.13}$$

An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the threshold classifies more items as positive, thus increasing both False Positives and True Positives.

Area Under the ROC curve (AUC)

AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1). AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example.

Sensitivity

Sensitivity is a measure of how well a machine learning model can detect positive instances. It is also known as the true positive rate (TPR) or recall.

$$\mathbf{Sensitivity} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}} \tag{4.14}$$

Specificity

Specificity measures the proportion of true negatives that are correctly identified by the model.

$$\mathbf{Specificity} = \frac{\mathbf{TN}}{\mathbf{TN} + \mathbf{FP}} \tag{4.15}$$

It is the percentage of the model's right predictions

4.2.2 Resampling

A common phenomenon observed in medical data and in many different situations is that the amount of observations in the represented classes is not similar, so that the frequency that one class displays is a lot higher another.

For example, in our case we have to deal with a binary classification problem which means that the number of patients that have died after they were admitted to the ICU is much smaller than the amount of patients that are still alive. Therefore there is an imbalance amongs the two classes.

Most of the machine learning models work more efficiently in the case of an imbalance amongst the multitude of classes in our data. In the opposite case, because a majority class is a lot higher then the minority class the decision boundary that is implemented in the chosen models, will be significantly affected. For this reason there's a big chance of a misclassification in the minority class to come up which will then lead us to lower percentages of accuracy.

Our goal for the minimization of the imbalance class problem is achieved by applying resampling techniques to restore the balance of the different samples that belong to different classes. One of these techniques, called oversample, is the creation of new samples and it contributes to the increase of samples of the minority class. Moreover there is also the down sample technique that can make that aims to the minimization of the majority class. The resampling is being implemented on data to prepare them for model training so that the model has as much accuracy as possible.

The method that was chosen for our study is called synthetic minority oversampling technique (SMOTE)[9]. This method is based on the creation of synthetic samples along the line segments to the future space, using the initial data of the minority class and subsequently the K nearest neighbor method is being applied so that the samples are being selected in random manner which will increase the minority class.

4.2.3 Cross Validation

When we construct a machine learning model, it is very common to come across an over fitting problem. That happens because the model has been training extensively on training data (noise) which has a negative impact on its performance when it is given new data to process. With the cross validation [31] technique we can mitigate this phenomenon.

Cross validation (CV) is a resampling technique that is applied to statistical learning methods and it is very useful when the data we have to deal with are limited. There are various ways to do cross validation on data with the main goal of determining the performance of the model on unseen data.

The cross validation technique we used is K-fold cross validation which separates the data into k-folds (k-groups) approximately equal in size in a random manner. After the k-folds have been created, the first fold is being recognized as the validation set and the following ones are characterized as

training data sets. Subsequently, a fit model is constructed and the mean square error (MSE) is calculated in the held-out fold (validation set). This procedure is repeated K times and each time the next fold is treated as the validation set. The total assessment of the k-fold CV is calculated as the mean value of the MSE for every repetition. In the case that we are called to implement classification models we use different metrics to evaluate the model. More specifically we use the Area under the roc curve against the MSE, therefore to evaluate the k-fold cross validation we calculate the mean value of Area under the roc curve for all the repetitions.

4.3 Hyperparameters Tuning

The aim of Hyperparameter Tuning is the performance's improvement of the ML models, we were led to the calculation of their hyperparameters, during the construction phase of the model. The hyperparameters to be tuned must be experimented by combining them using a range of values for each of them in order to find which combination is the best. Thus, the primary goal is the optimization of the hyperparameters that control the learning process. Each model has a different construction therefore, it has its own hyperparameters. The following shows which hyperparameters were tuned for each model.

Extreme Gradient boost hyperparameters:
eta:$[0.1, 0.15, 0.2]$
maxdepth:$[5, 6, 8, 10, 12, 15]$
min_child_weight:$[1, 3, 5, 7]$
gamma:$[0.0, 0.1, 0.2, 0.3, 0.4]$
colsample_bytree:$[0.3, 0.4, 0.5, 0.7]$

Random Forests hyperparameters
number of trees:$[50, 100, 150, 200, 250, 300, 350, 400]$
maxdepth:$[5, 6, 7, 8, 10, 12, 15]$
min_samples_split:$[2, 5, 10]$
min_sampels_leaf:$[1, 3, 4]$

Support Vector Machine hyperparameters:
C: $[0.1, 1, 10, 100]$
gamma: $[0.0001, 0.001, 0.01, 0.1, 1, 10]$

K Nearest Neighbor hyperparameters:
K:$[3 - 21]$

Logistic Regression hyperparameters:
C: $[100, 10, 1.0, 0.1, 0.01]$

Gaussian Naive Bayes hyperparameters:
var smoothing: $[10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}, 10^{-9}, 10^{-10}, 10^{-11}, 10^{-12}, 10^{-13}, 10^{-1}$

4.3.1 Grid-Search Cross Validation

For the tuning of the hypermeters, the most efficient algorithm that can be applied is Gridsearch. Every hyperparameter of each model, accompanied by the different values, are registered in a domain grid. The function of the gridsearch is based on the trial of all the possible combinations of the domain grid, in order to detect the combination which contains the best score in a specific performance metric (i.e. AUCROC).

The technique of the Gridsearch is practically executed during the performing of the K-fold cross validation in the training data and validation data and it applies grid search to every model that is being created. Therefore, it aims at the location of the most effective combination of hyperparameters for which the model achieves its best performance.

This specific algorithm can prove to be exhaustive for huge grids, because there is a plethora of different combinations that the algorithm is being called to execute for big domain grids. This contributes to the time of execution being quite a lot. For this reason GridSearchCV was used in models that consist of a small amount of hyperparameters, like KNN, SVC, LR and GNB.

Scikit-learn class model-selection provides us with the GridSearchCV method with the following listed parameters:

1. Estimator: a scikit-learn model
2. Param_grid: A dictionary with parameter names as keys and lists of parameter values.
3. Scoring: The performance measure.
4. Cv: An integer that is the number of folds for K-fold cross-validation.

4.3.2 Bayesian-Search Cross Validation

Bayesian optimization [3] can be applied to seek the global optimum of expensive black-box where the functions are computationally expensive to find the extrema. It can be applied to functions whose expressions do not need to be completed in a finite number of operations. It can also be used for functions which are expensive to calculate, when the function is non-convex, or the derivatives are difficult to evaluate. Bayesian optimization is practically limited to optimizing upon less than 20 parameters. Bayesian Optimization has been established to Bayes theorem,

$$P(A/B) = P(B/A) * P(A)/P(B) \tag{4.16}$$

The above relation is simplified as:

$$P(A/B) \propto P(B/A) * P(A) \tag{4.17}$$

Where $P(A/B)$ is posterior probability, $P(B/A)$ is likelihood and $P(A)$ is prior probability Calculation of the normalizing value $P(B)$ is dismissed and describe the conditional probability as a proportional quantity. There is no interest in calculation for a specific conditional probability, but instead in optimizing a quantity.

The core idea of Basyesian optimization is occupied by the above formula. The purpose of Bayesian Optimization is to combine the sample information with the prior distribution of function's f(x) to obtain the posterior of the function. This information is used for finding where the function f(x) is maximized according to a criterion that is represented by a utility function u known as acquisition function. Acquisition function is used to determine the next sample point that maximize the expected utility.

1: For n=1,2,$\cdots$

2: Find x_n by optimizing the acquisition function u over function f:

$x_n = \arg\max u(x|D_{1:n-1})$

3: Sample the objective function: $y_n = f(x_n)$

4: Augment the data $D_{1:n} = \{D_{1:n-1}, (x_n, y_n)\}$ and update the posterior of function f.

5: End for.

Chapter 5

Machine Learning Pipeline

The use of a Machine Learning (ML) pipeline is essential in resolving our problem because it allows us to experiment with various techniques on the data to achieve the best model performance possible. An ML pipeline is an end-to-end process that organizes the flow of data into models and presents the prediction outcomes as output. It consists of three main stages, including pre-processing, learning evaluation, and prediction. The pipeline is made up of multiple modules, each with its own functionality that plays a critical role in achieving the desired final outcome. The ML pipeline offers a great deal of flexibility, allowing for modification and adaptation to suit different data types and modeling needs. Figure 5.1 illustrates the workflow of the constructed ML pipeline. We relied on the tools provided by sklearn [27] to implement the modules, while pandas [26], a Python library, was used for data processing. The pre-processing stage involves cleaning and preparing the data, including handling missing values, scaling, and normalization. The learning evaluation stage focuses on evaluating the performance of the models and selecting the best model based on the evaluation metrics. The prediction stage involves applying the selected model to new data to generate predictions.

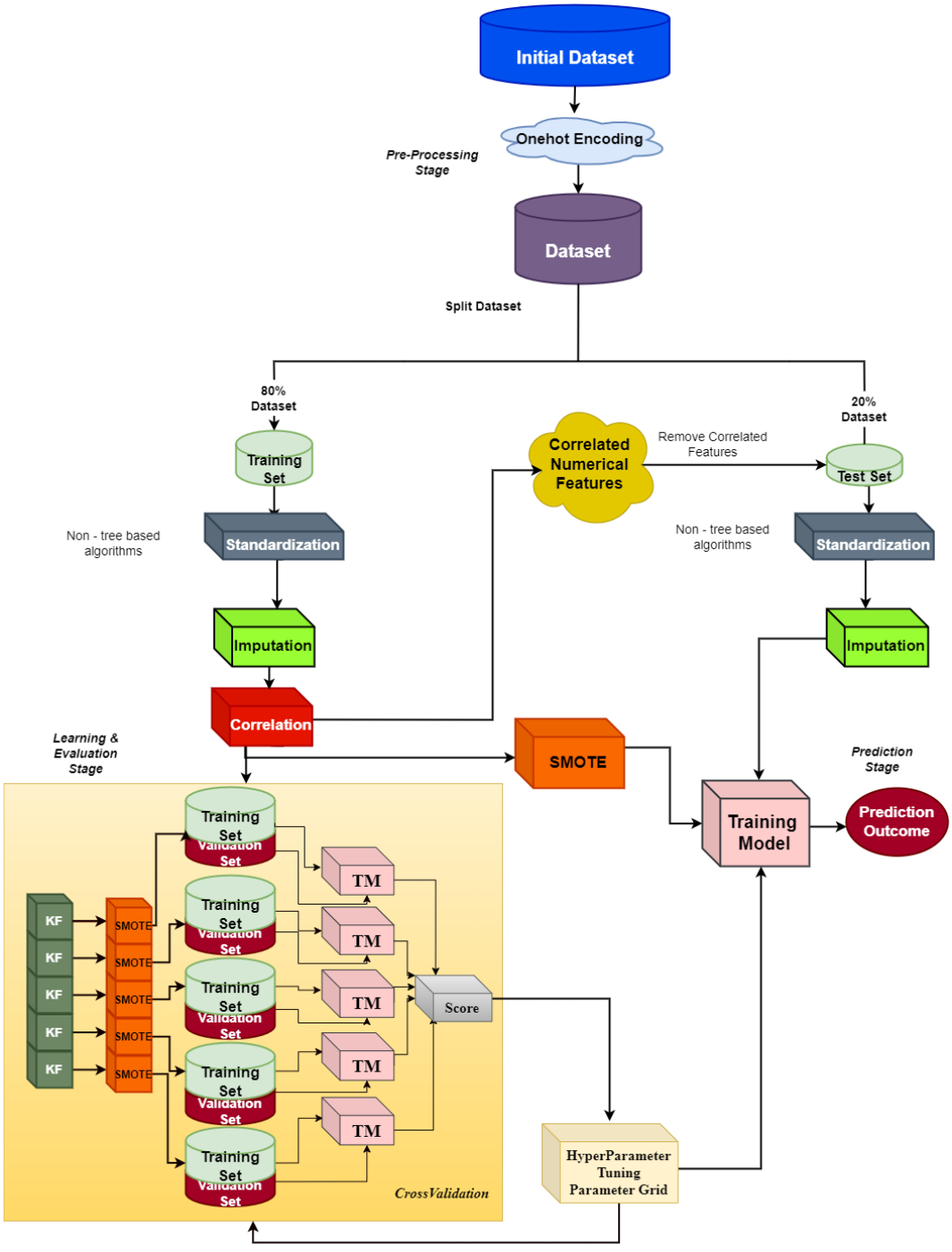


Figure 5.1: ML Pipeline

5.1 Preprocessing Stage

Preprocessing is the first stage of the machine learning pipeline. The initial data receives its first processing with the categorization of its information. More specifically the features that have categorical information, are transformed into numerical features (one hot encoding). A key factor in learning machine learning is the correct isolation between the sets intended for learning the model (training set) and the sets intended for its evaluation (test set). Therefore, each processing of the data must be done separately for each of these two separate sets. The isolated sets received the following processes: standardization for non-tree based algorithms, imputation for filling in the missing values as well as correlation for the correlated features that are unnecessary and burden the machine learning models.

5.1.1 One-hot encoding

Input data contain mixed data types (numerical measurements and text fields). One-hot encoding was applied for their management and for categorizing a categorical feature into the n possible values of the n numerical features. Each feature represented with the respective feature has a value of 1, else it has value of 0. The fact that the categorical features are converted into numerical ones, leads to the increasing of the model dimensionality. Pandas.get_dummies() was used for applying one-hot encoding. It accepts the categorical features as a parameter and creates features in numerical form.

5.1.2 Split dataset

The dataset's split constitutes the next pipeline step. The training set consists of 80% of the total dataset , while the test set of the rest 20%. Training set is used for fitting ML models. On the other hand, traning set is used in the end and it provides us with the final prediction outcomes. The chosen function for dividing the dataset into two subsets, is the train_test_split() from the sklearn module. Separation is achieved by using stratified as with Stratified Sampling ensures each group receives the proper representation within the sample. When the population can be partitioned into homogeneous subgroups, this technique gives a more accurate estimate of model parameters than random sampling..

5.1.3 Standardization

Since the main dataset was partitioned, normalization needs to be applied into the two subsets. The form of normalization used is standardization. Standardization was not applied in all the experiments, but only in the GNB, LR, SVM and KNN algorithms, i.e. in the algorithms that do not apply a tree implementation. StandardScaler() from scikit learn, is the tool for transforming the data.

5.1.4 Correlation

Preprocessing's next stage regards the finding of correlated features. Their tracking is rendered possible through the corr() method, from pandas module. It accepts the dataframe as an input and returns the correlation matrix that contains all the correlations among the features. Correlation matix values range in the [-1,1] interval. The threshold was chosen to be equal to 0.9. This means that each feature with correlation greater than 0.9 or less than -0.9 , is removed from the dataset. The related features found in the training set are also deleted from the test set.

5.1.5 Imputation

The existence of missing values is apparent in some features, as the real information that exists is negligible. To that end, we added a threshold whose role is to specify the allowing missing values that each feature can possess. After a number of tries, the value of the threshold was chosen as 0.7. In case of a feature consisting of a single value, it is removed as it does not contribute to the forming of the outcome.

Following the check of the above requirements for the missing values, mean imputation for the numerical features is applied. Empty cells are replaced with the mean value of all the values that have been registered in each respective features, whereas in categorical features' case, mode imputation is taking place and the values with the highest appearance rate in the feature to which they belong , are registered in the empty values. Towards achieving a balanced ratio between the two classes, the Synthetic Minority Oversampling Technique (SMOTE) [9] is adopted. I For imputation, Autoimpute, which is a python package for analysis and implementations of imputation methods, is used. More specifficaly, the MiceImputer method from Autoimpute method was applied. It passes through data multiple times and iteratively optimizes imputations in each column.

5.2 Learning & Evaluation Stage

In the above process, the classifier repeatedly tests different combinations of hyperparameters by performing 5-fold cross-validation on them. Once all the possible combinations have been tested, the results are compared to identify the best outcome provided by the specific classifier. The hyperparameters that produced the best result are then used in the final training of the model.

To implement this process, two modules of skopt are used: GridSearchCV and BayesianSearchCV. GridSearchCV is used for SVM, KNN, GNB, and LR classifiers, while BayesianSearchCV is used for XGB and Random Forest classifiers. The reason for using different methods is that RF and XGB have a large number of hyperparameters that make the GridSearch method impractical due to the large number of possible combinations. Bayesian Search, on the other hand, uses probabilistic models to optimize the search process, reducing the computational time and cost.

During the GridSearchCV and BayesianSearchCV processing, SMOTE is applied to each 5-fold to balance the classes 0 and 1. This helps to ensure that the model is not biased towards one particular class and provides more accurate predictions.

5.3 Prediction Stage

The last stage of the pipeline involves predicting the classifiers based on the best hyperparameters selected from the learning and evaluation stage. To ensure that the prediction model is accurate, SMOTE is applied on the training set to generate new samples and balance the number of instances in both classes (0 and 1). This helps to improve the model's ability to detect the minority class by creating synthetic samples of the minority class. After generating the new samples, the classifier is trained using the new modified training set and the hyperparameters selected from the learning and evaluation stage. Once the training is complete, the final prediction is made on the test set using the trained classifier. The prediction accuracy is evaluated using various metrics such as AUC, sensitivity, specificity, precision, and recall, which help to determine the effectiveness of the prediction model.

Chapter 6

Results

6.1 Prediction Mortality of ICU patients with Venous Thromboembolism

The data for the prediction of the mortality of the patients with venous thrombosis regarded 4,385 overall patients with mean age equal to 60.6 years.(SD±12.9 years) of these patients, 2,739 of these patients were diagnosed with pulmonary embolism and 2,220 with deep vein thrombosis. Out of these patients, 547 people died (12.5%) , whereas 3838 managed to survive(88.5%). 2,350 variables were gathered and created from the eICU database. Table 6.1 contains the demographic and clinical information during the prepossessing stage, 475 features out of them were chosen to be used for training the models. According to results, as shown in Fig. 6.1 and table 6.2, Logistic Regression provides the best scores with an AUC of 0.87 CI:0.87-0.89 with hyperparameter C value 0.1 after hyperparameter tuning and using the liblinear solver. XGB follows with AUC 0.82 CI:0.80-0.83 with hyperparameters values eta 0.15, maxdepth 12, gamma 0.2. Random Forest AUC 0.81 CI:0.80-0.82, GNB AUC 0.81 CI 0.79-0.83 and the worst Learner KNN AUC 0.69 CI 0.67-0.71. Table 6.3 contains the detected hyperparameters for each algorithm. For algorithms with tree implementation,(Fig. 6.2, 6.3) display the features importances, motor and day1verbal for XGB and random forest respectively,

Characteristic	VTE
Overall patients	4.385 PE: 2.739(62.4%) DVT: 2,220 (50.6%)
Sex	
Female	2115
Male	2268
Uknown	1
Ethnicity	
Caucasian	3386
Afro-American	577
Native-American	15
Asian	36
Hispanic	165
Other	206
Average age (SD)	62.16(16.45)
Minimum age	15
Maximum age	90
Los (days)	
Average Los (SD)	11.12(11.9)
Median Los	7.13
Mortality	
Alive(%)	3838(87.5%)
Dead(%)	547(12.5%)

Table 6.1: Demographic-clinical characteristics of patients with VTE

Table 6.2: Evaluation For Classifiers VTE

Models	Accuracy	Sensitivity	Specificity	AUC
XGB	0,84±0,15	0.94±0.01	0,36±0,03	0,82±0,03
SVM	0.72±0.02	0.93±0.01	0.23±0.1	0,71±0.2
RF	0,76±0,15	0,93±0,01	0,35±0,01	0,81±0,1
LR	0,83±0,01	0.95±0,01	0.41±0,25	0.87±0.02
GNB	0,82±0,01	0.94±0.01	0.36±0,03	0,81±0,02
KNN	0,42±0,01	0,95±0.02	0,16±0,05	0,69±0,02

Figure 6.1: VTE AUC-ROC curve VTE

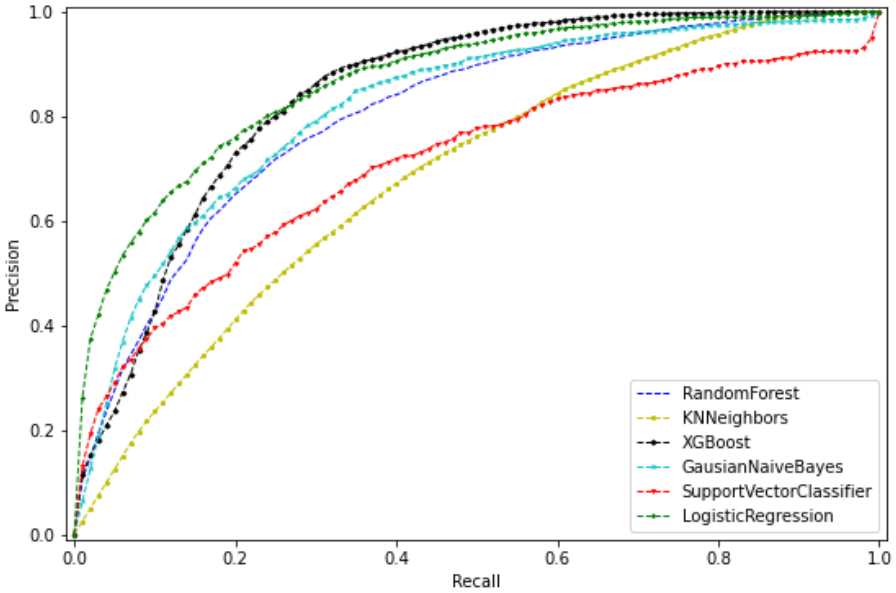


Table 6.3: Hyperparameters VTE

XGB	RF	LR	SVM	GNB	KNN
eta: 0.15	n_estimators: 400	C: 0,1	C: 10	var_smoothing: 0.01	n_neighbors: 20
maxdepth: 12	maxdepth: 15	solver: liblinear	kernel: sigmoid		
min_child_weight: 3	min_samples_leaf: 1				
gamma: 0,2	min_child_split: 5				
colsample_bytree: 0,7	criterion: gini				

Figure 6.2: Most important Features XGB

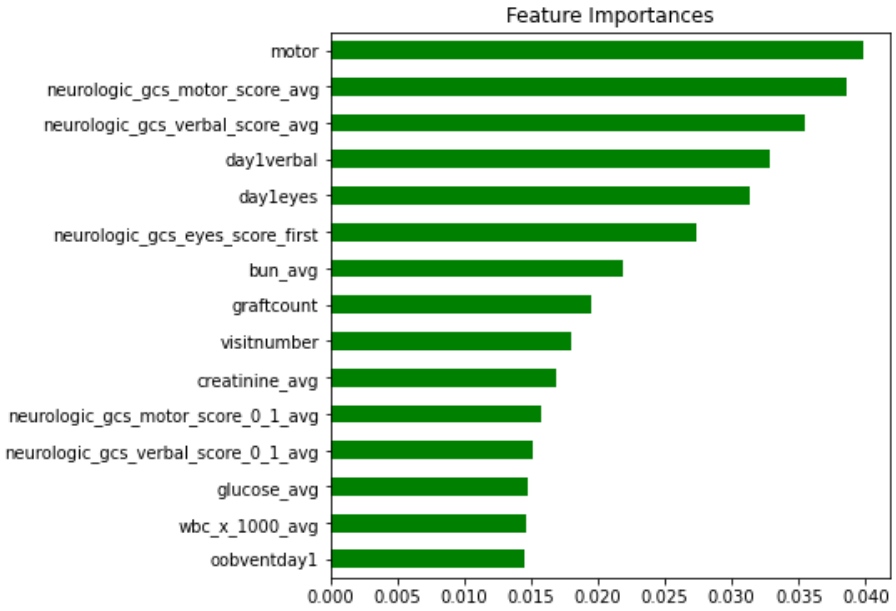
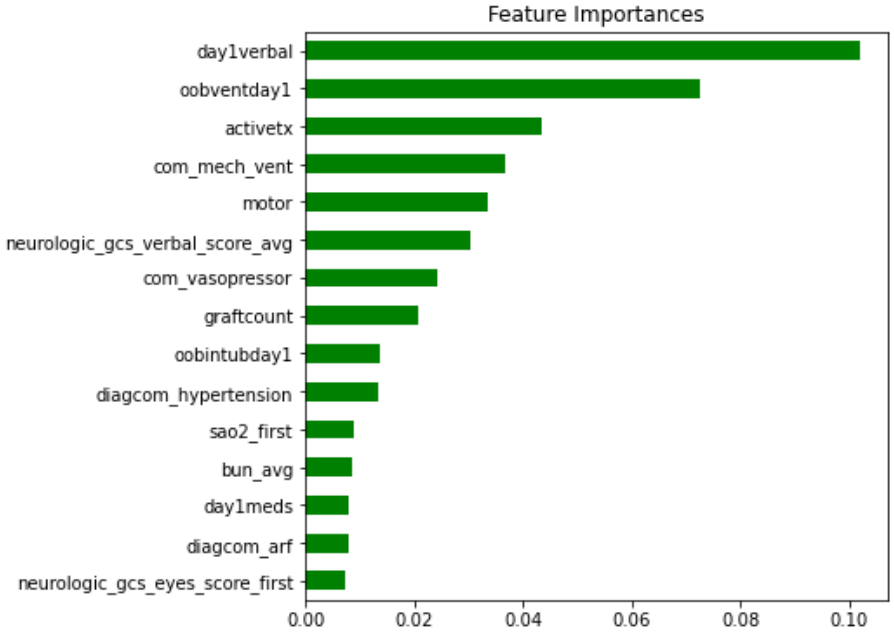


Figure 6.3: Most important Features Random Forest



6.2 Prediction Mortality of ICU patients with Myocardial Infarction

The study included a total of 10,542 patients who were diagnosed with myocardial infarction Table 6.4. The median age was 66 years with SD±13.35, 6707 of the patients were female and 3835 male, 91.6% survivors and 8.4% non-survivors. After processing the data, 317 features took part in conducting the experiments. In this particular experiment, the best metrics are provided by XGB with AUC 0.95 CI 0.94-0.96 (hyperparameters eta: maxdepth:) and then SVM-RF with AUC 0.93 CI:0.92-0.94, LR AUC 0.91 CI:0.90-0.92, GNB AUC 0.88 CI:0.87-0.89 and finally KNN with AUC 0.87 CI:0.86-0.88 as shown in table 6.5 and figure 6.4. Table 6.6 shows in detail all hyperparameters calculated from the Bayesian Search and Grid Search. Figures [6.5, 6.6] contains the most important features for XGB and Random Forest.

Characteristic	myocardial
Overall patients	10543
Sex	
Female	3835
Male	6707
Uknown	-
Ethnicity	
Caucasian	8327
Afro-American	926
Native-American	34
Asian	167
Hispanic	371
Other	718
Average age (SD)	66.01(13.35)
Minimum age	16
Maximum age	90
Los (days)	
Average Los (SD)	6.2(10.06)
Median Los	3.73
Mortality	
Alive(%)	9656(91.6%)
Dead(%)	887(8.4%)

Table 6.4: Demographic-clinical characteristics of patients with myocardial infarction

Table 6.5: Evaluation Classifiers myocardial infarction

Models	Accuracy	Sensitivity	Specificity	AUC
XGB	0.95±0.01	0.96±0.01	0.75±0.04	0.95±0.01
RF	0.92±0.02	0.96±0.01	0.54±0.08	0.93±0.01
SVM	0.93±0.01	0.96±0.01	0.6±0.02	0.93±0.01
LR	0.9±0.02	0.97±0.01	0.43±0.02	0.91±0.01
GNB	0.85±0.01	0.97±0.01	0.33±0.02	0.88±0.01
KNN	0.7±0.02	0.98±0.01	0.2±0.01	0.87±0.01

Figure 6.4: Myocardial infarction AUCROC curve

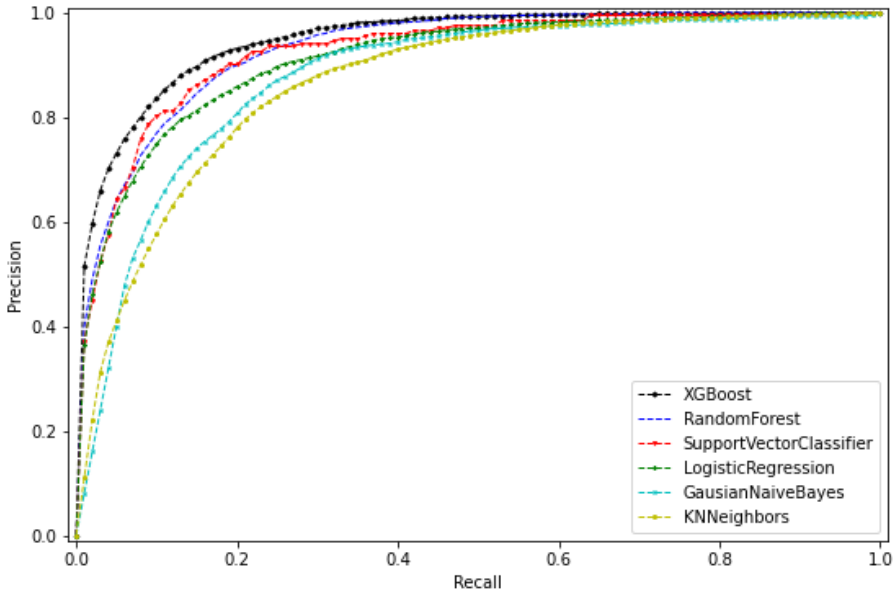


Table 6.6: Hyperparameters myocardial infarction

XGB	RF	LR	SVM	GNB	KNN
eta: 0.15	n_estimators: 400	C: 0,01	C: 1	var_smoothing: 0.01	n_neighbors: 20
maxdepth: 8	maxdepth: 15	solver: liblinear	kernel: rbf		
min_child_weight: 1	min_samples_leaf: 1				
gamma: 0,1	min_child_split: 5				
colsample_bytree: 0,4	criterion: entropy				

Figure 6.5: Most important Features XGB Myocardial infaction

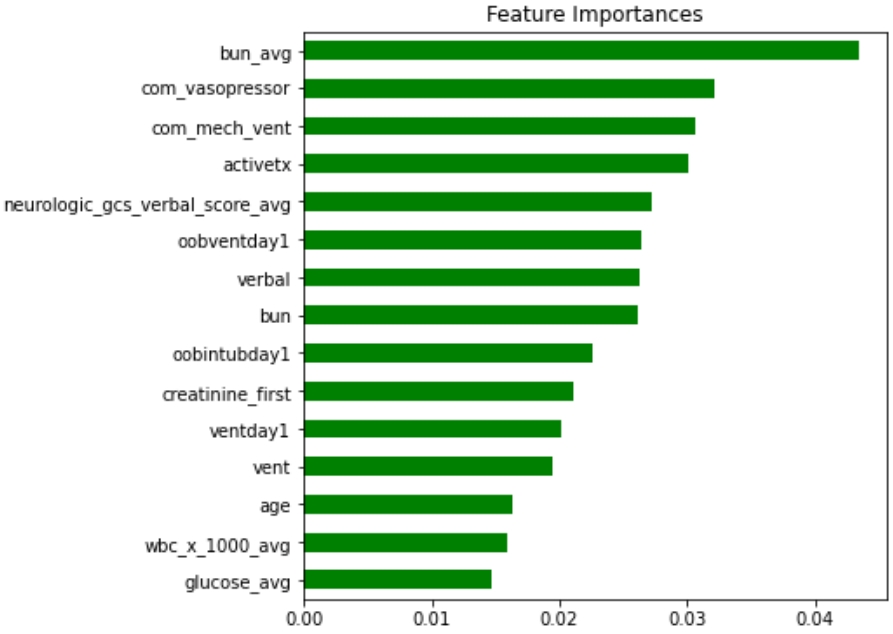
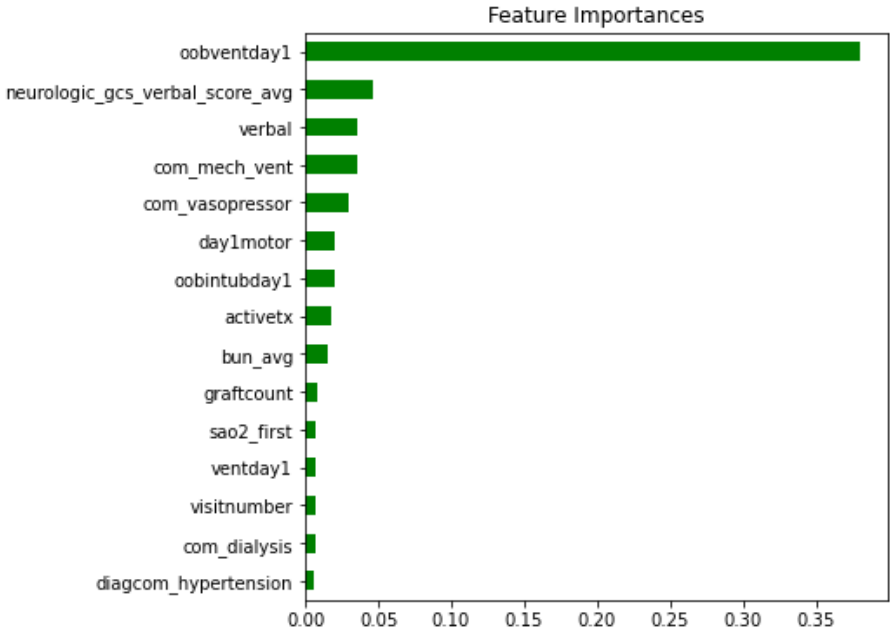


Figure 6.6: Most important Features Random Forest Myocardial infarction



6.3 Prediction Mortality of ICU patients with Ischemic Stroke

The current study analyzed a total of 4,326 patients who were admitted with a diagnosis of ischemic stroke, among whom 3,866 (89.4%) died. The mean age of the patients was 68.48 years with a standard deviation of 14.31 years. The demographic characteristics of the patients are presented in Table 6.7. A total of 338 features were used for training the classifiers in order to predict mortality. Among the different classifiers that were tested, XGB provided the best results with an AUC of 0.90 and a confidence interval of 0.89-0.91, as shown in Table 6.8 and Figure 6.7. The hyperparameter values for each classifier are provided in Table 6.9. Further analysis was conducted to identify the most important features for RF and XGB, which are presented in Figures 6.9 and 6.8, respectively. These findings highlight the potential of XGB as a powerful tool for predicting mortality in patients with ischemic stroke, using a large number of features extracted from the electronic health records.

Characteristic	IS
Overall patients	4326
Sex	
Female	2114
Male	2211
Uknown	1
Ethnicity	
Caucasian	3319
Afro-American	470
Native-American	14
Asian	67
Hispanic	219
Other	237
Average age (SD)	68.48(14.31)
Minimum age	19
Maximum age	90
Los (days)	
Average Los (SD)	8.01(15.03)
Median Los	5
Mortality	
Alive(%)	3866(89.4%)
Dead(%)	460(10.6%)

Table 6.7: Demographic-clinical characteristics of patients with IS

Table 6.8: Evaluation Classifiers Stroke

Models	Accuracy	Specificity	Sensitivity	AUC
XGB	0.91±0.01	0.94±0.01	0.6±0.04	0.9±0.01
RF	0.88±0.02	0.94±0.01	0.48±0.06	0.89±0.02
LR	0.85±0.01	0.95±0.01	0.4±0.03	0.87±0.01
GNB	0.79±0.01	0.96±0.01	0.31±0.01	0.83±0.01
KNN	0.63±0.04	0.97±0.01	0.2±0.02	0.79±0.01
SVM	0.71±0.02	0.94±0.01	0.21±0.02	0.71±0.02

Table 6.9: Hyperparameters Stroke

XGB	RF	LR	SVM	GNB	KNN
eta: 0.1	n_estimators: 350	C: 0,01	C: 10	var_smoothing: 0.01	n_neighbors 10
maxdepth: 10	maxdepth: 15	solver: newton-cg	kernel: sigmoid		
min_child_weight: 3	min_samples_leaf: 1				
gamma: 0,2	min_child_split: 2				
colsample_bytree: 0,5	criterion: entropy				

Figure 6.7: IS AUCROC curve

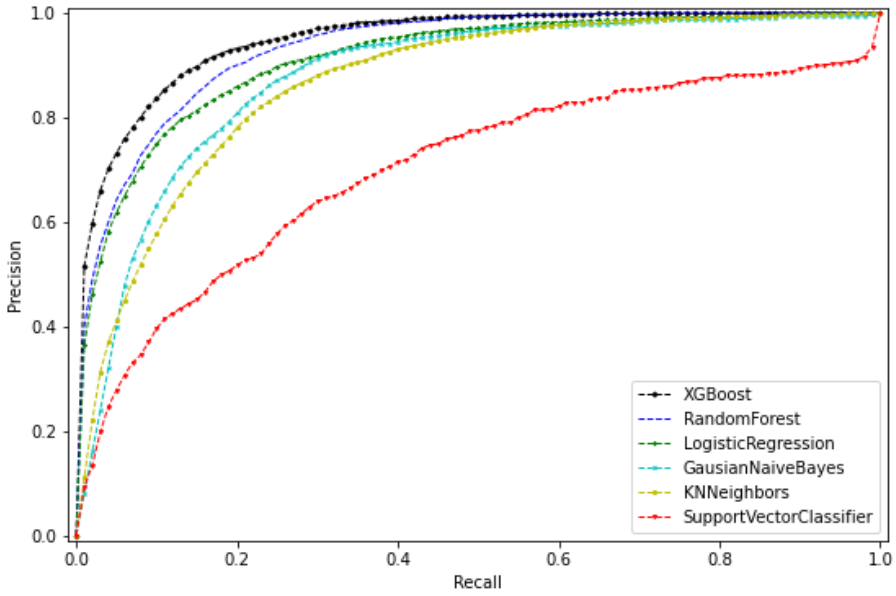


Figure 6.8: Most important Features XGB IS

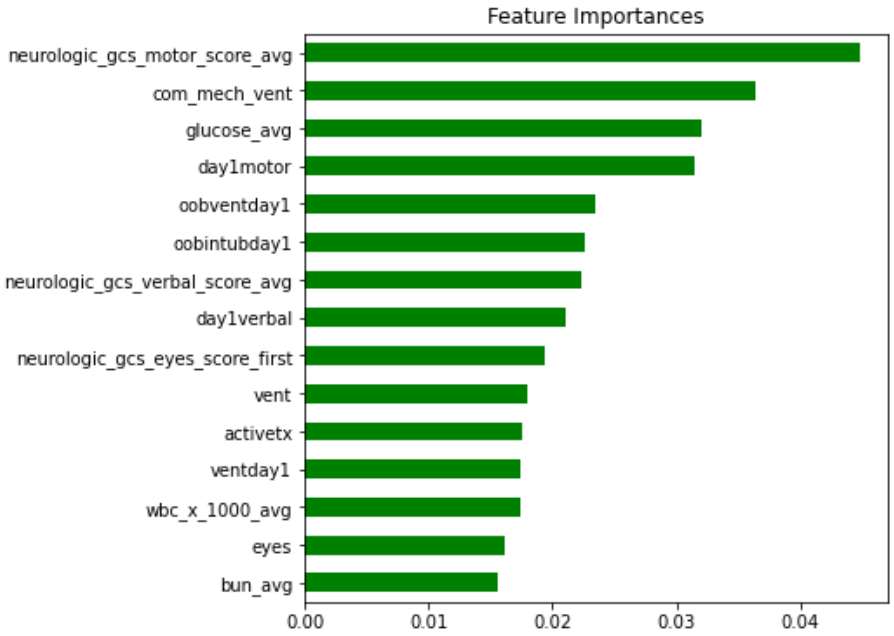
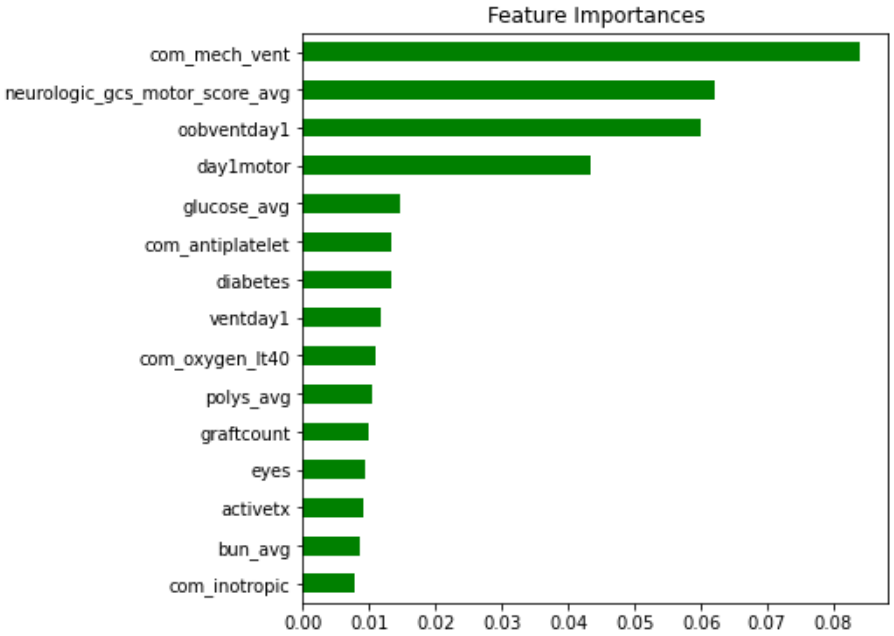


Figure 6.9: Most important Features Random Forest IS



Chapter 7

Discussion

The primary contribution of this work is the investigation of a vast number of clinicolaboratory features stored in the electronic health record. These features were grouped into meaningful sets such as vital signs, labs, medications, and procedures, and were timestamped to examine their impact on the prediction of mortality in ICU patients. This research utilized an open-access large healthcare dataset, eICU, to extract three homogeneous population cohorts based on their diagnoses, including venous thromboembolism, myocardial infarction, and ischemic stroke.

The study dealt with medical data that was intricate, disorderly, and frequently had gaps in its information, making it arduous to handle. In order to guarantee that the data was appropriate for utilization in machine learning algorithms, it had to undergo a thorough cleaning and preprocessing stage. This procedure necessitated a considerable amount of time and effort; however, it led to the development of a high-quality dataset, which could be utilized to construct precise prediction models. The process of cleaning and preprocessing the medical data was a crucial step in the study. It involved various techniques such as imputing missing values, handling outliers, and transforming variables to improve the accuracy of the data. The dataset needed to be prepared in a manner that would allow machine learning algorithms to process and comprehend it effectively. Despite the challenges posed by the intricate and messy nature of the medical data, the efforts invested in cleaning and preprocessing were worthwhile. The resulting high-quality dataset facilitated the construction of precise prediction models. Accurate prediction models have significant implications in the medical field, where they can be used to make informed decisions and improve patient outcomes.

To develop the machine learning pipeline, the above features were accepted as arguments, and the features were modified to the desired state to test various ML algorithms for the final comparison between the produced

results. State-of-the-art ML algorithms, including GNB, KNN, LR, SVM, RF, and XGB, were compared. Additionally, the class imbalance problem in medical datasets was addressed, and interpretable models were developed to identify clinically meaningful predictive signatures. The developed model outperformed traditional clinical scores in predicting mortality. A multidimensional time series data-driven research approach was used to identify machine learning algorithms with the highest predictive performance. Overall, this research offers valuable insights into the use of machine learning algorithms to predict mortality in ICU patients and highlights the importance of utilizing complex medical data to develop accurate prediction models.

In this research, a comprehensive range of classifiers were meticulously selected to represent various category implementations. The selection process involved considering a diverse set of probabilistic models such as GNB and LR, as well as tree models like XGB and RF. Furthermore, models that utilize N-dimensional space implementations, such as KNN and SVM, were also included. The classifiers were evaluated by calculating hyperparameters and selecting appropriate kernels, solvers, and criteria to enhance the accuracy of the assessment. Each algorithm was assessed based on its individual strengths, which were proportional to each experiment. The selection of multiple classifiers was imperative in ensuring that the study was robust and comprehensive. Each classifier brings a unique set of advantages to the table, which can be leveraged to improve the accuracy and efficiency of the predictions made by machine learning models. The probabilistic models such as GNB and LR are commonly used for classification tasks, especially when the data is highly skewed. Tree models like XGB and RF, on the other hand, are useful for handling non-linear relationships between features. Meanwhile, models that utilize N-dimensional space implementations, such as KNN and SVM, are useful for identifying patterns in high-dimensional data. The evaluation process involved identifying the optimal hyperparameters for each classifier and selecting suitable kernels, solvers, and criteria to enhance the accuracy of the assessment. This process was time-consuming and required a significant amount of computational resources. However, the efforts invested in selecting and evaluating the classifiers resulted in a robust and reliable dataset that could be used to construct precise prediction models.

To ensure the reliability of the results, each classifier was tested 15 times separately, and the mean value was calculated for each experiment. The evaluation and comparison between them were based on specific scores, including specificity, sensitivity, and AUC. Overall, the predictive models we developed showed good performance in predicting mortality of patients with VTE, MI, and IS. The AUC values for all three models were above 0.8, indicating good discrimination. The LR model performed well for VTE, with an AUC of

0.83 and high sensitivity of 0.95. However, the specificity was relatively low at 0.41, which means that the model may have a higher false positive rate. This could be due to the complexity of the VTE disease process, which involves multiple risk factors and comorbidities. For IS, the XGB model had the best performance, with an AUC of 0.95 and high accuracy, sensitivity, and specificity. This suggests that the model was able to identify important predictors of mortality for IS patients and capture their interactions effectively. The XGB model also performed well for MI, with an AUC of 0.9 and high sensitivity and accuracy. However, the specificity was lower than that of the IS model, which may reflect the heterogeneity of MI patients and the difficulty in identifying specific risk factors. Further validation using external datasets and prospective studies is needed to confirm their usefulness in clinical practice.

Head to head comparisons of the various studies in ICU mortality prediction are difficult, since the various studies have different inclusion and exclusion criteria, different types of studied features, and various definitions of mortality. Our study targeted three specific groups of patients, patients with venous thromboembolism, patients with myocardial infarction and patients with ischemic stroke. All diagnostic groups are high-risk patients, with a substantial risk of ICU admission and mortality. Mortality prediction models for ICU patients with thrombosis, myocardial or ischemic stroke that are based on ML algorithms and use a large amount of clinical and laboratory data, structured and unstructured, are almost completely absent in the literature. Moreover, traditional scoring systems are not specific for these three diseases. To the best of our knowledge, only one publication on a relatively small number of patients with venous thromboembolism has been published [32]. Similarly to Runnan et al., we compared state-of-the-art ML algorithms with traditional scores, and we achieved comparable performance and identified similar predictive features[14].

Some limitations of this study should be considered. First of all, the study was based on retrospective data. Since the data were collected in the past, it is possible that many medical practices have changed over time. Second, the selection of the studied diagnostic groups was based solely on ICD-9 codes and DRG codes [8], and not on imaging studies. Third, time series data were processed in specific time stamps, which increased significantly the dimensionality of the data. Moreover, we observed that labs and vital signs in both datasets were infrequently reported in the first 48 h, thus leading to a dramatically increased number of missing data on the various time stamps. Fourth, a direct comparison of our model with the only PE-specific score, PESI, was not possible, since this is not included in the datasets[14].

Chapter 8

Conclusions

The presented research could be used as a proof of concept study that could be further validated in prospective or more recent datasets. Prediction of in-hospital mortality in patients with VTE, MI or IS is highly feasible. The results of this study are promising and, most importantly, interpretable, since the predictive features included in the model were clinically meaningful. The evaluation results proved that machine learning models are able to produce accurate classifcation result for patients' mortality. specially XGB and RF models performed consistently throughout all the experiments.

Chapter 9

Future Work

One of the primary goals of our future work is to directly compare our models with the PESI score and in a prospective cohort study. Inclusion of more features, such as genetic information and imaging studies, would be ideal and would probably improve the predictive performance. We could also focus on features extracted on the day of discharge to predict other outcomes, such as ICU readmission. Our future vision is to develop an intelligent ML-based system that is continuously updated with new clinical events and detailed information of the current clinical status of the patient, which could be a useful assistant for the physician and their clinical decision-making. To this end, the use of deep learning models, such as long short-term memory (LSTM) [33] for importing time series data in high-frequency datasets, and neural networks [4] could probably achieve better generalization performance with a significantly lower error rate. Shapley additive explanation (SHAP) analysis could be used to explain the output of our predictive model [21]. Handling of the high imbalance ratio of the datasets could be performed with other advanced resampling methods, such as Generative Adversarial Networks [17] [14].

Bibliography

[1] reprinted in biometrika 45 296-315 1958 thomas bayes 1702-1761 http://www.cs.monash.edu.au/~lloyd/tildeimages/people/bayes/index.html ([about bayes]).

[2] Pearson's correlation coefficient. pages 1090–1091, 2008.

[3] Practical bayesian optimization of machine learning algorithms. pages 2951–2959, 2012. Copyright: Copyright 2013 Elsevier B.V., All rights reserved.; 26th Annual Conference on Neural Information Processing Systems 2012, NIPS 2012 ; Conference date: 03-12-2012 Through 06-12-2012.

[4] L. Ali and S.A.C. Bukhari. An approach based on mutually informed neural networks to optimize the generalization capabilities of decision support systems developed for heart failure prediction. *IRBM*, 42(5):345–352, October 2021.

[5] Firdaus Aziz, Sorayya Malek, Khairul Shafiq Ibrahim, Raja Ezman Raja Shariff, Wan Azman Wan Ahmad, Rosli Mohd Ali, Kien Ting Liu, Gunavathy Selvaraj, and Sazzli Kasim. Short- and long-term mortality prediction after an acute ST-elevation myocardial infarction (STEMI) in asians: A machine learning approach. *PLOS ONE*, 16(8):e0254894, August 2021.

[6] Kochawan Boonyawat, Pichika Chantrathammachart, Pawin Numthavaj, Nithita Nanthatanti, Sithakom Phusanti, Angsana Phuphuakrat, Pimjai Niparuck, and Pantep Angchaisuksiri. Incidence of thromboembolism in patients with COVID-19: a systematic review and meta-analysis. *Thrombosis Journal*, 18(1), November 2020.

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

[8] R. Busse, A. Geissler, A. Aaviksoo, F. Cots, U. Hakkinen, C. Kobel, C. Mateus, Z. Or, J. O'Reilly, L. Serden, A. Street, S. S. Tan, and

W. Quentin. Diagnosis related groups in europe: moving towards transparency, efficiency, and quality in hospitals? *BMJ*, 346(jun07 3):f3197–f3197, June 2013.

[9] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002.

[10] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[12] T. Cover and P. Hart. Nearest neighbor pattern classification. 13:21–27, 1967.

[13] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.

[14] Vasiliki Danilatou, Stylianos Nikolakakis, Despoina Antonakaki, Christos Tzagkarakis, Dimitrios Mavroidis, Theodoros Kostoulas, and Sotirios Ioannidis. Outcome prediction in critically-ill patients with venous thromboembolism and/or cancer using machine learning algorithms: External validation and comparison with scoring systems. *International Journal of Molecular Sciences*, 23(13), 2022.

[15] W El-Bouri, A Sanders, GYH Lip, and BBC-VTE Investigators. Machine learning prediction of mortality in venous thromboembolism patients: the birmingham black country venous thromboembolism (bbc-vte) cohort. *European Heart Journal*, 42(Supplement_1):ehab724–3059, 2021.

[16] Carlos Fernandez-Lozano, Pablo Hervella, Virginia Mato-Abad, Manuel Rodríguez-Yáñez, Sonia Suárez-Garaboa, Iria López-Dequidt, Ana Estany-Gestal, Tomás Sobrino, Francisco Campos, José Castillo, Santiago Rodríguez-Yáñez, and Ramón Iglesias-Rey. Random forest-based prediction of stroke outcome. *Scientific Reports*, 11(1), May 2021.

[17] Ghadeer Ghosheh, Jin Li, and Tingting Zhu. A review of generative adversarial networks for electronic health records: applications, evaluation measures and data sources. 2022.

[18] Claire M. Hull, Dévan Rajendran, and Arturo Fernandez Barnes. Deep vein thrombosis and pulmonary embolism in a mountain guide: Awareness, diagnostic challenges, and management considerations at altitude. *Wilderness &amp Environmental Medicine*, 27(1):100–106, March 2016.

[19] T. Jayalakshmi and A. Santhakumaran. Statistical normalization and back propagationfor classification. *International Journal of Computer Theory and Engineering*, pages 89–93, 2011.

[20] Jun Ke, Yiwei Chen, Xiaoping Wang, Zhiyong Wu, Qiongyao Zhang, Yangpeng Lian, and Feng Chen. Machine learning-based in-hospital mortality prediction models for patients with acute coronary syndrome. *The American Journal of Emergency Medicine*, 53:127–134, March 2022.

[21] Nguyen Quoc Khanh Le, Quang Hien Kha, Van Hiep Nguyen, Yung-Chieh Chen, Sho-Jen Cheng, and Cheng-Yu Chen. Machine learning-based radiomics signatures for EGFR and KRAS mutations prediction in non-small-cell lung cancer. *International Journal of Molecular Sciences*, 22(17):9254, August 2021.

[22] Rafael Lozano, Mohsen Naghavi, Kyle Foreman, Stephen Lim, Kenji Shibuya, Victor Aboyans, Jerry Abraham, Timothy Adair, Rakesh Aggarwal, and Stephanie Y Ahn et al. Global and regional mortality from 235 causes of death for 20 age groups in 1990 and 2010: a systematic analysis for the global burden of disease study 2010. *The Lancet*, 380(9859):2095–2128, December 2012.

[23] Ahmad Manshad, Oguz Akbilgic, Yevgeniy Brailovsky, Dalila Masic, Alexandru Marginean, Sorcha Allen, Katerina Porcaro, Shannon Kuhrau, Karim Merchant, Nathalie Antonios, Lucas Chan, Stephen Morris, Ibrahim Chowdhury, Jeremiah Haines, Jawed Fareed, and Amir Darki. Machine learning-based prediction of 30-day all-cause mortality in patients hospitalized with acute pulmonary embolism. *Chest*, 158(4):A2213–A2214, October 2020.

[24] Antonio Mucherino, Petraq Papajorgji, and Panos M Pardalos. *Data mining in agriculture*, volume 34. Springer Science & Business Media, 2009.

[25] Ximing Nie, Yuan Cai, Jingyi Liu, Xiran Liu, Jiahui Zhao, Zhonghua Yang, Miao Wen, and Liping Liu. Mortality prediction in cerebral hemorrhage patients using machine learning algorithms in intensive care units. *Frontiers in Neurology*, 11, January 2021.

[26] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[28] Tom J. Pollard, Alistair E. W. Johnson, Jesse D. Raffa, Leo A. Celi, Roger G. Mark, and Omar Badawi. The eICU collaborative research database, a freely available multi-center database for critical care research. *Scientific Data*, 5(1), sep 2018.

[29] Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. Machine learning in medicine. *N. Engl. J. Med.*, 380(14):1347–1358, April 2019.

[30] G.E. Raskob, P. Angchaisuksiri, A.N. Blanco, H. Buller, A. Gallus, B.J. Hunt, E.M. Hylek, A. Kakkar, S.V. Konstantinides, M. McCumber, Y. Ozaki, A. Wendelboe, and J.I. Weitz. Thrombosis. *Arteriosclerosis, Thrombosis, and Vascular Biology*, 34(11):2363–2371, November 2014.

[31] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. pages 532–538, 2009.

[32] Runnan Shen, Ming Gao, Yangu Tao, Qinchang Chen, Guitao Wu, Xushun Guo, Zuqi Xia, Guochang You, Zilin Hong, and Kai Huang. Prognostic nomogram for 30-day mortality of deep vein thrombosis patients in intensive care unit. *BMC Cardiovascular Disorders*, 21(1), January 2021.

[33] Hans-Christian Thorsen-Meyer, Annelaura B Nielsen, Anna P Nielsen, Benjamin Skov Kaas-Hansen, Palle Toft, Jens Schierbeck, Thomas Strøm, Piotr J Chmura, Marc Heimann, Lars Dybdahl, Lasse Spangsege, Patrick Hulsen, Kirstine Belling, Søren Brunak, and Anders Perner. Dynamic and explainable machine learning prediction of mortality in patients in the intensive care unit: a retrospective study of high-frequency data in electronic patient records. *The Lancet Digital Health*, 2(4):e179–e191, April 2020.

[34] Georgios Tzanos, Christoforos Kachris, and Dimitrios Soudris. Hardware acceleration on gaussian naive bayes machine learning algorithm. pages 1–5, 2019.

[35] Raymond E Wright. Logistic regression. 1995.

[36] Changhu Xiao, Yuan Guo, Kaixuan Zhao, Sha Liu, Nongyue He, Yi He, Shuhong Guo, and Zhu Chen. Prognostic value of machine learning in patients with acute myocardial infarction. *Journal of Cardiovascular Development and Disease*, 9(2):56, February 2022.

[37] Harry Zhang. The optimality of naive bayes. *AA*, 1(2):3, 2004.