Predicting One-Year Mortality for Patients with Chronic Diseases Using Administrative Data: A Machine Learning Approach to Chronic Hepatitis and Tuberculosis in Kazakhstan

Iliyar Arupzhanov, B. Eng. in ECE

Submitted in fulfilment of the requirements
for the degree of Master of Science
in Electrical and Computer Engineering

School of Engineering and Digital Sciences
Department of Electrical and Computer Engineering
Nazarbayev University

53 Kabanbay Batyr Avenue,
Astana, Kazakhstan, 010000

Supervisors: Amin Zollanvari, Abduzhappar Gaipov

April 2024

Declaration Form

DECLARATION

I hereby, declare that this manuscript, entitled “Predicting One-Year Mortality for Patients with Chronic Diseases Using Administrative Data: A Machine Learning Approach to Chronic Hepatitis and Tuberculosis in Kazakhstan”, is the result of my own work except for quotations and citations which have been duly acknowledged.

I also declare that, to the best of my knowledge and belief, it has not been previously or concurrently submitted, in whole or in part, for any other degree or diploma at Nazarbayev University or any other national or international institution.

Name: Iliyar Arupzhanov

Date: 05.04.2024

Abstract

Objectives: Chronic diseases pose a significant threat to global health, highlighting the need for innovative approaches to predict patient outcomes effectively. This study aims to predict the one-year mortality in patients with chronic viral hepatitis (CVH) and tuberculosis (TB) utilizing administrative data, which includes demographic information, comorbidities, diagnoses, and characteristics of service providers.

Methods: Clinical data collected from a nationwide database between January 2014 to December 2019 was analyzed with 82,700 CVH patients and 150,000 TB patients. The data were segmented into yearly cohorts to forecast mortality within one year based on information up to the end of the preceding year. We developed a machine learning platform utilizing six categories of models: linear, nearest neighbors, support vector machines, naïve Bayes, and ensemble methods (including gradient boosting, AdaBoost, and random forest). Feature importance was assessed through SHapley Additive exPlanations (SHAP) values.

Results: The year-specific models demonstrated an area under the receiver operating characteristic curve (AUC) between 0.74 and 0.83 on separate test sets. SHAP analysis showed that age, sex, type of hepatitis, and ethnicity are main predictors of one-year mortality for CVH patients. For TB patients, main predictors included age, type of TB, ethnicity, and duration of TB.

Conclusion: The results show that it is possible to construct accurate machine learning models using administrative health data for predicting one-year mortality in patients with CVH and TB. In future work, detailed laboratory and medical history data could be incorporated to improve performance. This integration can provide a helpful tool for healthcare workers to effectively manage and treat chronic diseases.

Table of Contents

Chapter 1 – Introduction

1.1 General

Chronic diseases such as diabetes, cardiovascular diseases (CVD), chronic viral hepatitis (CVH), and tuberculosis (TB) pose significant challenges to public health worlwide due to their high morbidity and mortality rates. Chronic diseases are important due to their long duration and high risk of death. Healthcare professionals work on improvement medical plans and raising public knowledge; however, mortality rates are still high. This underscores the complexity of proper managing and predicting the outcomes of chronic diseases.

CVH is a chronic disease commonly obtained due to infection with hepatitis B (HBV) and hepatitis C (HCV) viruses [1, 2]. About 300 million people, in 2019, suffered from chronic hepatitis B (CHB), and around 60 million people from chronic hepatitis C (CHC) according to the World Health Organization (WHO) [3]. In Kazakhstan particularly from 2014 to 2019, 82,700 patients were diagnosed with CHB or CHC [4]. CVH can lead to serious complications, such as liver cancer and chronic cirrhosis, which has high potential for death. Due to CVH and its corresponding complications 1.1 million people died in 2019 [3], which is significant number; therefore, the World Health Organization (WHO) aims to reduce mortality by 65% between 2015 to 2030 [5]. However, it is predicted that total number of deaths related to CHB and CHC could increase to 19 million by the year 2030 [5].

The situation with TB is more concerning [6]. Global Tuberculosis Report shows that there were 10.6 million cases of TB and 1.3 million deaths in 2022 indicating an impact [7]. In Kazakhstan 150,000 TB patients were recorded between 2014 and 2019 [8]. In 2022, TB caused 1.3 million deaths, which is a significant number [6]. WHO's End TB strategy aims to reduce TB-related deaths by 75% between 2015 and 2025; however, there are still challenges to meet this target [7]. Sakko et al. [8] using data from the Unified National Electronic Health

System (UNEHS) showed that all cause mortality rates related to TB increased from 8.4 to 15.2, per 100,000 individuals.

Considering these statistics, it is important to develop effective mortality prediction systems to help clinicians to tailor treatment plans and improve the survival of patients with chronic diseases. Computational prediction methods, particularly machine learning (ML) models, can be helpful in early detection and mortality prediction of diabetes. ML can be effective in the prediction tasks at the same time being less time-consuming than conventional detection methods and having almost zero cost of implementing. ML can be effectively utilized for predicting mortality in patients with chronic diseases. Several researchers developed machine learning models for mortality prediction tasks, which are discussed below.

1.2 Literature review

1.2.1 Prediction of hepatitis mortality

Several researchers [9-12] have used the University of California (UCI) dataset to construct a model for predicting mortality in patients with hepatitis. The UCI dataset included age, gender, ethnicity features, as well as physical examination findings, and laboratory test results from 155 individuals. Albogamy *et al.* [9] built four models, namely support vector machines (SVM), K-nearest neighbors (KNN), Naïve Bayes (NB) and bidirectional long/short-term memory (BiLSTM) networks to predict hepatitis mortality. BiLSTM model achieved the highest results with sensitivity of 0.93 and f-score of 0.93.

Bhargav *et al.* [10] evaluated four machine learning classifiers: SVM, NB, logistic regression (LR) and decision tree (DT). LR showed the best performance with sensitivity of 0.87 and f-score of 0.86. Yildirim [10] focused on predicting hepatitis mortality using the NB, C4.5 classifier and decision tables. NB achieved the highest performance with sensitivity of 0.9 and f-score of 0.75.

Obaido *et al.* [12] conducted a thorough study on the UCI hepatitis dataset. They evaluated seven machine learning classifiers, including NB, SVM, DT, LRR, random forest (RF), AdaBoost (ADB) and XGBoost (XGB). ADB showed an area under the curve (AUC) of 0.93 and sensitivity of 0.93, outperforming other models. This was the only research among the reviewed that reported the area AUC metric and performed model explainability using Shapley Additive eXplanations (SHAP) analysis. Authors found that high ascite levels, age, alkaline, and malaise levels were the most important predictors.

All reviewed studies evaluated ML models on a single dataset from UCI with small samples, which may limit the generalizability of the findings to various patient groups. Moreover, most of the studies did not report the AUC metric, which is independent of decision threshold, in comparison with other metrics. Furthermore, the focus on model explainability was largely absent except in the study by Obaido *et al.* [12]. This underscores a critical gap in understanding how these predictive models make their decisions, which is crucial for clinical application and trust in these systems.

1.2.2 Prediction of tuberculosis mortality

Recent studies [13-16] leveraging ML techniques have focused on enhancing the prognosis of TB by predicting treatment outcomes, particularly the risk of treatment failure.

Lino Ferreira da Silva Barros *et al.* [13] conducted a study to predict treatment failure outcomes (cured and died) of TB patients. The authors collected the dataset from the Brazilian health databases, consisting of 36,228 patients diagnosed and treated for TB between 2007 and 2018. The dataset after preprocessing contained 24,015 records with 38 variables, including laboratory test data, demographic data, comorbidities and medication information. The authors evaluated eight machine learning models: LRR, SVM, KNN, DT, RF, gradient boosting (GB),

multilayer perceptron (MLP) and ensemble of RF, GB and MLP. The GB classifier showed the best performance with an AUC of 0.965.

Peng *et al.* [14] employed ML models for early predicting treatment failure among patients with TB-diabetes comorbidity. The authors collected dataset of 429 patients with 69 features, including demographic data, comorbidities, laboratory test data and computer tomography (CT) images, from the Chongqing Public Health Medical Center from February 2019 to January 2021. Authors compared the performance of SVM, LRR, RF and XGB, with XGB showing the highest AUC of 0.928. Moreover, authors performed SHAP analysis to identify the most significant predictors of treatment failure in TB patients. Higher drug resistances, and high values of APTT, TT, and PDW laboratory test values were more associated with treatment failure.

Sauer *et al.* [15] evaluated the performance of LASSO, RF, SVM with linear and polynomial kernels in predicting cured cases of TB patients. Authors used dataset with 587 cases managed by the National Institute of Allergy and Infectious Diseases. LASSO outperformed other classifiers with an AUC of 0.72 and specificity of 0.96. Furthermore, the analysis revealed that drug sensitivity and imaging findings as critical predictors.

1.2 Aims & Objectives

The literature review reveals a gap in studies predicting mortality in patients with chronic diseases using only administrative data (including demographic data, comorbidities, diagnoses, and characteristics of service providers), which is notably straightforward and cost-effective to collect, making it appealing resource for large-scale health studies.

In this regard, we develop a ML platform to construct a model that predicts one-year mortality in patients with chronic diseases. Clinical data collected from the Kazakhstan Unified National Electronic Health System (UNEHS) [17] between January 2014 to December 2019

were used. The scope of this study is limited to hepatitis and tuberculosis due to availability of datasets for these diseases. Our findings show the effectiveness of developed ML platform in predicting one-year mortality for patients with hepatitis and TB in Kazakhstan. Additionally, we have performed SHAP analysis to identify the most important clinical variables in predicting one-year mortality.

Chapter 2 – Methodology

2.1 Data collection

We gathered data for this research from the UNEHS, which is managed by the Ministry of Health (MoH) and the Republican Center for Electronic Health (RCEH) in Kazakhstan. The UNEHS represents 65.1% of all medical organizations, mainly from the public sector [17].

The initial dataset derived from the UNEHS comprises 20,810,911 medical records from both the DRDP and DERI, covering the period of 2014 to 2019 [4]. From this extensive dataset, two disease-specific datasets were created for focused analysis.

For the hepatitis analysis, records of patients diagnosed with CVH were filtered using the International Classification of Diseases 10th Revision (ICD-10) codes: B18.1 (chronic viral hepatitis B without delta-agent) and B18.2 (chronic viral hepatitis C). After removing

duplicates, the hepatitis dataset consisted of 82,700 unique patients. Analysis of this group considered seven clinical variables, including age, sex, type of hepatitis, hepatitis duration, cirrhosis, and hospitalization, which are detailed in Table 2.1.

In a similar approach, the TB dataset was formed utilizing ICD-10 codes for tuberculosis, namely A15 (respiratory tuberculosis, bacteriologically and histologically confirmed) and A16 (respiratory tuberculosis, not confirmed bacteriologically, molecularly or histologically), resulting in a dataset of 149,122 individuals diagnosed with tuberculosis. Tuberculosis dataset contained nine clinical factors: sex, residence, age, type of tuberculosis, duration of tuberculosis, comorbidities including diabetes, hepatitis, and HIV, and frequency of hospital admissions, as outlined in Table 2.2.

Table 2.1: Description of clinical variables for hepatitis dataset

Variable	Type	Category/Units	Description
Type of Hepatitis	Categorical	Binary	Differentiates between Chronic hepatitis C or Chronic hepatitis B without delta function
Hepatitis Duration	Numeric	Years	Time elapsed from the hepatitis diagnosis until December 31st of the year preceding the prediction
Age	Numeric	Years	Patient's age when hepatitis was diagnosed
Sex	Categorical	Binary	Female or male
Ethnicity	Categorical	Ternary	Kazakhs, Russians, and others
Cirrhosis	Categorical	Binary	Indicates the presence of cirrhosis (yes /no). as a comorbidity

Hospitalization	Categorical	Binary	Records whether the patient underwent hospitalization (yes/no)

This study utilized secondary data from UNEHS, therefore the Nazarbayev University Institutional Review Ethics Committee (NU-IREC 490/18112021) waved the requirement for informed consent from the participants involved. This research followed the "Reporting of studies conducted using observational routinely collected health data" (RECORD) guidelines to ensure ethical and methodological integrity.

2.2 Data preprocessing

The initial step in data preprocessing involved removing patient records with missing outcome data, either deceased or alive. The aim of our research is to identify who is dead or alive within the next year, which introduces the time aspect. However, the clinical data lacks precise timestamps, except for the availability of the clinical information up to prior year. To address this challenge, we divided each dataset into four distinct year-specific cohorts: 2016-, 2017-, 2018-, 2019-cohorts.

Table 2.2: Description of clinical variables for tuberculosis dataset

Variable	**Type**	**Category/Units**	**Description**
Type of Tuberculosis	Categorical	Binary	Differentiates between tuberculosis confirmed and not confirmed (bacteriologically and histologically)
Tuberculosis Duration	Numeric	Years	Time elapsed from the tuberculosis diagnosis until December 31st of the year preceding the prediction
Age	Numeric	Years	Patient's age when tuberculosis was diagnosed
Sex	Categorical	Binary	Female or male
Ethnicity	Categorical	Ternary	Kazakhs, Russians, and others

Residence	Categorical	Binary	
Diabetes	Categorical	Binary	Indicates the presence of diabetes (yes/no), as a comorbidity
Hepatitis	Categorical	Binary	Indicates the presence of hepatitis (yes/no), as a comorbidity
HIV	Categorical	Binary	Indicates the presence of HIV (yes /no), as a comorbidity
Number of hospitalizations	Numeric	Count	The total number of hospital admissions due to tuberculosis

Focusing on 2018-cohort as an illustrative example, the subcohort consisted of two patient groups:

1. Case group: Patients diagnosed with the disease before the start of 2018 and died within the same year. These patients correspond to case 1 as depicted in Figure 2.1.
2. Control group: Patients diagnosed before the beginning of 2018 but who remained alive throughout the year, similar to cases 2 and 4 in Figure 2.1.

Patients who died prior to the year of observation (case 3) and patients who were diagnosed within the year of observation (cases 5 and 6) were not included in subcohort. Only patients with available clinical information and who were alive up to the end of 2017 were included.

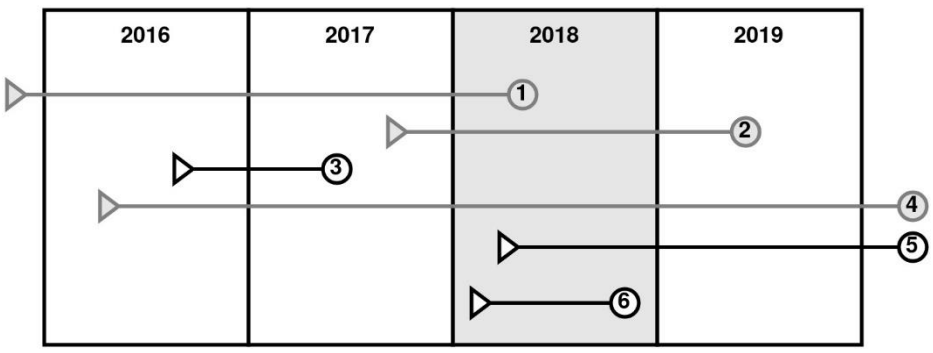

Figure 2.1: Description of subcohort selection

For the hepatitis dataset, the number of patients in each subcohort was as follows: 29,301 in 2016-cohort, 39,553 in 2017-cohort, 50,618 in 2018-cohort and 63,541 in 2019-cohort, respectively. A significant imbalance was noted between the numbers of deceased and survivors across the cohorts with ratios: 349:28,952 for 2016, 551:39,000 for 2017, 727:49,891 for 2018, and 783:62,758 for 2019.

For TB dataset, the number of patients in each subcohort were recorded as 69,925 for 2016, 86,670 for 2017, 101,612 for 2018, 112,703 for 2019. A significant imbalance in the data was noted, as evidenced by the death-to-survival ratios for each year: 1948:67,977 in 2016, 1982:84,688 in 2017, 2204:99,408 in 2018, and 2190:110,513 in 2019.

Each year-specific subcohort was divided with an 80/20 ratio into training and test sets using a stratified random split to keep the proportion of dead and alive in training and test sets the same as in the full cohort. For handling missing data, numeric variables were imputed using the median of the corresponding variables in the training data, while missing categorical variables were imputed using the most frequent category (mode). The training set is utilized to train and select the predictive model, while the test set is used to evaluate the best predictive model's performance.

2.3 Machine learning classifiers

Ten different classifiers were utilized in this study: linear models including logistic regression with L_2 ridge penalty (LRR) [18], support vector machines with linear kernel (SVM) [19], and perceptron (PER) [20]; Gaussian Naive Bayes (GNB) [20]; ensemble methods including random forest (RF) [21], XGBoost (XGB) [22], LightGBM (LGB) [23], gradient boosting with regression trees (GBRT) [24], and Adaboost with decision trees (ADB) [25]; K-nearest neighbors [20]. To have a better understanding of the classifiers using in this work, we provide a full detailed theoretical background below.

2.3.1 Gaussian Naïve Bayes

Naïve Bayes (NB) is a probabilistic machine learning algorithm used for classification tasks, based on the Bayes' theorem [20]. Bayes' theorem calculates the conditional (posterior) probability of an event, given prior knowledge about related events. In the context of classification, assuming a sample point of p features, $\mathbf{x} = [x_1, x_2, \dots, x_p]^T$, the Bayes' theorem can be used to calculate the probability that a sample belongs to class i for a given sample **x**:

$$P(Y = i|\mathbf{x}) = \frac{P(Y = i)P(\mathbf{x}|Y = i)}{P(\mathbf{x})} \quad (1)$$

where:

- $P(Y = i|\mathbf{x})$ is the posterior probability of class variable given **x**,
- $P(\mathbf{x}|Y = i)$ is the likelihood, which is the probability of **x** given class variable,
- $P(Y = i)$ is the prior probability of class i,
- $P(\mathbf{x})$ is the marginal probability of **x**.

In NB [20], it is assumed that features are independent of each other, given the class variable. Therefore, NB for a binary classification is represented as:

$$\psi(\mathbf{x}) = \begin{cases} 1 & if\ P(Y = 1)\prod_{j=1}^{p} P(x_j|Y = 1) > P(Y = 0)\prod_{j=1}^{p} P(x_j|Y = 0) \\ 0 & \end{cases} \quad (2)$$

In the Gaussian NB [20], we make an assumption that $P(x_j|Y = i)$ follows a Gaussian distribution:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{(x_j - \mu_y)^2}{2\sigma_y^2}} \quad (3)$$

where μ_y is a mean for class y and σ_y is a standard deviation for class y.

2.3.2 Logistic Regression

Logistic Regression (LR) is a learning technique used for binary classification tasks [18]. The concept of LR is to find a relationship between features and the probability of being in particular class by using the logistic function.

The logistic model expresses the log-odds, or logit, of the posterior probability $P(Y = 1|\mathbf{x})$ as a linear combination of the input features:

$$\text{logit}\big(P(Y = 1|\mathbf{x})\big) = \log\left(\frac{P(Y = 1|\mathbf{x})}{1 - P(Y = 1|\mathbf{x})}\right) = \mathbf{a}^T\mathbf{x} + b \tag{4}$$

In terms of the logistic sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, we have:

$$P(Y = 1|\mathbf{x}) = \frac{1}{1 + e^{-(\mathbf{a}^T\mathbf{x}+b)}} \tag{5}$$

which is then plugged in the form of Bayes classifier to construct the LR classifier:

$$\psi(\mathbf{x}) = \begin{cases} 1 & \text{if } \dfrac{1}{1 + e^{-(\mathbf{a}^T\mathbf{x}+b)}} > \dfrac{1}{2} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Equation (6) is equivalent to

$$\psi(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{a}^T\mathbf{x} + b > 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

In practice, **a** and b are unknown parameters which are estimated by an estimation algorithm, resulting in the estimates by $\hat{\mathbf{a}}$ and $\hat{b}$. $\psi(\mathbf{x})$ can then be defined as follows:

$$\psi(\mathbf{x}) = \begin{cases} 1 & \text{if } \hat{\mathbf{a}}^T\mathbf{x} + \hat{b} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

One common way for estimating the **a** and b is through maximum likelihood estimation (MLE). This method finds estimates $\hat{\mathbf{a}}$ and $\hat{b}$ that maximize the likelihood of observing the given labels in the dataset. Furthermore, this can be viewed as minimizing a loss function:

$$e(\boldsymbol{\beta}) = \sum_{j=1}^{n} \log\left(1 + e^{-y_j(\mathbf{a}^T\mathbf{x}_j + b)}\right) \tag{9}$$

After incorporating a regularization term to prevent overfitting, we obtain the following loss function:

$$e(\boldsymbol{\beta}) = C\sum_{j=1}^{n} \log\left(1 + e^{-y_j(\mathbf{a}^T\mathbf{x}_j + b)}\right) + \frac{1}{2}\left|\left|\boldsymbol{a}\right|\right|_2^2 \tag{10}$$

Here, C is a tuning hyperparameter controlling the trade-off between the loss term and $\left|\left|\boldsymbol{a}\right|\right|_2$ is l_2 norm of $\boldsymbol{a}$. Description and notations for LR were adapted from [26].

2.3.3 Support Vector Machines

Support Vector Machines (SVMs) are a powerful category of supervised learning algorithms utilized for both classification and regression tasks, though they are most commonly associated with classification [19]. SVMs aim to find the optimal separating hyperplane (decision boundary) that differentiates between classes in the feature space while maximizing the margin, which is defined as the shortest distance between the closest points of the classes (known as support vectors) to the hyperplane. This approach helps to ensure that the model is not only accurate on the training data but also generalizable to new, unseen data.

Given the training dataset $\mathbf{x}_i \in R^p$, consisting of n samples divided into two classes, and a corresponding vector label $y \in \{-1, 1\}^n$, where each y_i corresponds to the class of $\mathbf{x}_i$, we formulate the optimization problem for SVM with linear kernel [19] as follows:

$$\text{Minimize: } \frac{1}{2}\left|\left|w\right|\right|^2 + C\sum_{i=1}^{n} \xi_i \ , \tag{11}$$

subject to the constraints:

$$y_i(w^T\mathbf{x}_i + b) \geq 1 - \xi_i \tag{12}$$

$$\xi_i \geq 0, i = 1, \dots, n \tag{13}$$

where:

- w is the normal vector to the hyperplane.
- b is the bias term, adjusting the hyperplane's distance from the origin.
- ξ_i are slack variables to allow for margin violations; this accounts for data points that are either on the wrong side of the margin or the hyperplane, handling the non-linearly separable data.
- C is a regularization parameter that balances the trade-off between maximizing the margin and minimizing the classification error. A higher value of C places a greater penalty on misclassifications, leading to a decision boundary tightly fitted to the training data, while a lower C value encourages a wider margin and potentially more misclassifications in the training set.

2.3.4 Perceptron

Perceptron is one of the simplest types of artificial neural networks and a foundational model for understanding the field of machine learning [20]. It represents a linear classifier used for binary classification tasks.

Mathematically, the perceptron function is defined as follows [21]:

$$g(x) = f(\mathbf{w}^T\mathbf{x} + b) \tag{14}$$

where:

- $\mathbf{x}$ is an input feature vector, $\mathbf{w} = [w_1, \dots, w_n]$ denotes the weight vector, b is the bias weight.
- f is a nonlinear function known as activation function.

Typically, one choice of $f(\cdot)$ is a step function:

$$f(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \tag{15}$$

A perceptron is trained iteratively by updating the weights and bias based on the errors made by model in prediction. Based on the (15), we use $y_{true}, y_{pred} \epsilon \{0,1\}$ to match the choice of activation function. The weights for each instance are updated as follows [27]:

$$w_{new} = w_{old} + \eta(y_{true} - y_{pred})\mathbf{x} \tag{16}$$

and the bias is updated as:

$$b_{new} = b_{old} + \eta(y_{true} - y_{pred}) \tag{17}$$

where η is a learning rate.

2.3.5 K-nearest neighbors

K-nearest neighbors (KNN) is one of the earliest and the most basic machine learning algorithms used for classification problems [20]. KNN predicts a particular test instance based on the majority class of its k-nearest neighbors from the training observations. Euclidean distance is a common measure used to determine how close two points are.

KNN is formulated as follows:

$$\psi(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{i=1}^{k} \frac{1}{k} I_{\{y_{(i)}(\mathbf{x})=1\}} > \sum_{i=1}^{k} \frac{1}{k} I_{\{y_{(i)}(\mathbf{x})=0\}} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

where:

- k is the number of nearest neighbors defined.
- $\mathbf{x}$ is a feature vector of test observation.
- $y_{(i)}(\mathbf{x})$ denotes the class label of i^{th} nearest observation to $\mathbf{x}$.
- I is an indicator function, with $I = 1$ when $y_{(i)}(\mathbf{x}) = 1$, and $I = 0$ otherwise.

2.3.6 Random Forest

Random Forest algorithm is an ensemble learning technique that builds upon the concept of **b**ootstrap **agg**regat**ing** (bagging) of decision trees as base learners and introducing a randomness into the feature selection process for splits in each tree [21]. In the standard training of decision trees, the algorithm evaluates *all* features at each split point and chooses the one that maximizes the impurity drop, such as Gini impurity or information gain. This process continues recursively until meeting the split-stopping criteria like a minimum number of samples per node or maximum tree depth, to prevent overfitting and ensure computational efficiency.

RFs modify this approach by considering only a subset of randomly chosen d features, instead of considering all p features, where $d \leq p$. The optimal split is split is then determined on this subset, aiming to maximize the impurity drop. Common choices for d include $\sqrt{p}$ or $\log_2 p$ to balance exploring feature space and maintaining the trees' diversity.

The RF algorithm can be summarized as follows (the description and notations are adapted from [26]):

1. Create bootstrap samples: $\mathbf{S}^*_{tr,j}, j = 1, \ldots, B$
2. For each bootstrap sample, train a decision tree classifier denoted by ψ^*_j, where:
    a. To find the best split at each node, randomly pick $d \leq p$ features and maximize the impurity drop over these features.
    b. Grow the tree until one of the split-stopping criteria is met.
3. Perform the prediction for **x** as:

$$\psi_{RF}(\mathbf{x}) = \underset{i \in \{0,1\}}{\operatorname{argmax}} \sum_{j=1}^{B} I(\psi^*_j(\mathbf{x}) = i) \tag{19}$$

2.3.7 AdaBoost

AdaBoost, short for Adaptive Boosting, is an ensemble learning model designed to enhance the performance of weak learning algorithms [25]. AdaBoost is based on sequentially training a series of weak learners, each focusing on the errors of its predecessors, iteratively improving the overall performance of the model. Initially, all observations in the training data being assigned equal weights. During the training process, AdaBoost update these weights by increasing weights for observations that were misclassified by the previous learner, while decreasing weights for those that were correctly classified. AdaBoost ensures that subsequent learners focus more on the difficult cases that were challenging to previous learners. Through this iterative process of reweighting and retraining, AdaBoost assembles a strong classifier from a collection of simpler, weaker models.

Algorithm and its equations for implementing AdaBoost for a binary classification task are adapted from [26] and represented as follows:

1. Initialization: Assign an initial weight $w_j = \frac{1}{n}$ to each observation $j = 1,2, \dots, n$
2. For each iteration $m = 1$ to M (where M is the number of base models):
    a. Train a weak classifier $\psi_m(\mathbf{x})$ using the weighted training data.
    b. Determine error rate $\hat{\epsilon}_m$ of the weak classifier $\psi_m(\mathbf{x})$, which is the weighted sum of all misclassified points:

$$\hat{\epsilon}_m = \frac{\sum_{j=1}^{n} w_j I_{\{y_j \neq \psi_m(\mathbf{x}_j)\}}}{\sum_{j=1}^{n} w_j}, \tag{20}$$

    where I is an indicator function, with $I = 1$ when predicted label $\psi_m(\mathbf{x}_j)$ does not match true label y_j, and $I = 0$ otherwise.

    c. Calculate the confidence coefficient α_m for the classifier:

$$\alpha_m = \eta \log\left(\frac{1-\hat{\epsilon}_m}{\hat{\epsilon}_m}\right) \tag{21}$$

where η represents the learning rate, a tuning parameter that controls the influence of each classifier.

d. Update the weights for each observation $j = 1,2,\dots,n$:

$$w_j \leftarrow w_j e^{\alpha_m I_{\{y_j \neq \psi_m(\mathbf{x}_j)\}}} \tag{22}$$

3. Construct the final boosted classifier $\psi_{boost}(\mathbf{x})$ by aggregating the weak classifiers, weighted by their confidence α_m values:

$$\psi_{boost}(\mathbf{x}) = \underset{i\in\{0,1\}}{\operatorname{argmax}} \sum_{m=1}^{M} \alpha_m I_{\{\psi_m(\mathbf{x})=i\}} \tag{23}$$

2.3.8 Gradient Boosting with Regression Trees

Gradient Boosting with Regression Trees is a powerful machine learning technique for both regression and classification problems, which produces a prediction model in the form of an ensemble of weak models, particularly regression trees [24]. The algorithm starts with a base model, and iteratively improves the model by sequentially fitting trees to the residuals of the previous iterations.

Algorithm for implementing GBRT and its equations for a c-class classification task are adapted from [26] and are represented as follows:

1. Initialization: $F_{k0}(\mathbf{x}) = 0, \;\; k = 0,\dots,c-1$
2. For each iteration $m = 1$ to M, where M is the total number of boosting iterations (trees to be created):
    a. Compute class-specific probabilities:

$$p_{km}(\mathbf{x}) = \frac{e^{F_{k(m-1)}(\mathbf{x})}}{\sum_{k=0}^{c-1} e^{F_{k(m-1)}(\mathbf{x})}} \tag{24}$$

b. For each $k = 0, \ldots, c-1$:

   i. For each observation $i = 1, \ldots, n$, compute pseudo-residuals:

$$r_{ikm} = I_{\{y_i=k\}} - p_{km}(\mathbf{x}_i) \tag{26}$$

   ii. Fit a regression tree to the r_{ikm} values, creating terminal regions R_{jkm}, for $j = 1, \ldots, J$.

   iii. For each terminal region $j = 1, \ldots, J$ compute:

$$\gamma_{jkm} = \frac{c-1}{c} \frac{\sum_{\mathbf{x}_i \in R_{jkm}} r_{ikm}}{\sum_{\mathbf{x}_i \in R_{jkm}} |r_{ikm}|(1-|r_{ikm}|)} \tag{27}$$

c. Update model:

$$F_{km}(\mathbf{x}) = F_{k(m-1)}(\mathbf{x}) + \gamma_{jkm} I(\mathbf{x} \in R_{jkm}), \tag{28}$$

where $I(x \in R_{jm})$ is an indicator function with $I = 1$ when **x** falls into the j-th terminal region of m-th tree and $I = 0$ otherwise.

3. The final model $F_m(\mathbf{x})$ is passed through a logistic function to convert the output into a probability:

$$p_k(\mathbf{x}) = \frac{e^{F_{kM}(\mathbf{x})}}{\sum_{k=0}^{c-1} e^{F_{kM}(\mathbf{x})}} \tag{29}$$

The predicted class is then determined by $\hat{y} = \underset{k}{\operatorname{argmin}}\, p_k(\mathbf{x})$.

2.3.9 XGBoost

XGBoost, an optimized gradient boosting algorithm, known for its efficiency, flexibility, and portability [22]. Presentation and formulation of XGBoost is adapted from [26]:

XGBoost optimizes the gradient boosting machine learning algorithm by incorporating a regularized objective function that balances model performance with complexity to avoid overfitting:

$$h_m = \underset{h \epsilon \Phi}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, F_{m-1}(\mathbf{x}_i) + h\,(\mathbf{x}_i)) + \Omega(h) \tag{30}$$

where:

- L is a loss function quantifying the discrepancy between the actual y_i and its estimate at iteration m.
- $\Omega(h)$ represents the complexity measure of the tree h, penalizing the complexity of the trees.
- Φ represents the space of all regression trees.

XGBoost employs a second-order Taylor approximation to efficiently optimize the objective function:

$$L\big(y_i, F_{m-1}(\mathbf{x}_i) + h(\mathbf{x}_i)\big) \approx L\big(y_i, F_{m-1}(\mathbf{x}_i)\big) + g_{i,1} h(\mathbf{x}_i) + \frac{1}{2} g_{i,2} h(\mathbf{x}_i)^2 \tag{31}$$

where:

$$g_{i,1} = \left[\frac{\partial L\big(y_i, F_{m-1}(\mathbf{x}_i)\big)}{\partial F(\mathbf{x}_i)}\right] \tag{32}$$

$$g_{i,2} = \left[\frac{\partial^2 L\big(y_i, F_{m-1}(\mathbf{x}_i)\big)}{\partial^2 F(\mathbf{x}_i)}\right] \tag{33}$$

With $L\big(y_i, F_{m-1}(\mathbf{x}_i)\big)$ being independent of h, it is treated as constant term and can be removed, simplifying to:

$$h_m = \underset{h\epsilon\Phi}{\operatorname{argmin}} \sum_{i=1}^{n} \left[g_{i,1} h(\mathbf{x}_i) + \frac{1}{2} g_{i,2} h(\mathbf{x}_i)^2 \right] + \Omega(h) \tag{34}$$

Defining $I_j = \{i | \mathbf{x}_i \in R_j\}$ as the instance set in leaf j and J_h, as the number of leaves in tree h, we express the equation as:

$$h_m = \underset{h\epsilon\Phi}{\operatorname{argmin}} \sum_{j=1}^{J_h} \left[\left(\sum_{i\in I_j} g_{i,1} \right) b_j + \frac{1}{2} \left(\sum_{i\in I_j} g_{i,2} \right) b_j^2 \right] + \Omega(h) \tag{35}$$

Define $\Omega(h)$ as originally proposed in [22]:

$$\Omega(h) = \gamma J_h + \frac{1}{2} \lambda \sum_{j=1}^{J_h} b_j^2 \tag{36}$$

where γ and λ are parameters tuning the model. This regularization includes a term penalizing the number of leaves and a shrinkage effect on the leaf scores.

Integrating the regularization term into the objective function (34), results in:

$$h_m = \underset{h\epsilon\Phi}{\operatorname{argmin}}\, obj_m, \tag{37}$$

where:

$$obj_m = \sum_{j=1}^{J_h} \left[\left(\sum_{i\in I_j} g_{i,1} \right) b_j + \frac{1}{2} \left(\sum_{i\in I_j} g_{i,2} + \lambda \right) b_j^2 \right] + \gamma J_h \tag{38}$$

We can compute the optimal values of scores for a fixed structure with J_h leaves as:

$$b_j^* = -\frac{\sum_{i\in I_j} g_{i,1}}{\sum_{i\in I_j} g_{i,2} + \lambda}, j = 1, \dots, J_h \tag{39}$$

yielding the objective function value:

$$obj_m^* = -\frac{1}{2} \sum_{j=1}^{J_h} \frac{\left(\sum_{i\in I_j} g_{i,1} \right)^2}{\sum_{i\in I_j} g_{i,2} + \lambda} + \gamma J_h \tag{40}$$

Given the challenge of enumerating all possible tree structures, XGBoost adopts a greedy algorithm that starts from a single leaf and iteratively grows the tree. It evaluates the quality of splits by the resulting loss reduction, calculated as:

$$obj_m^* - obj_{m,\text{split}}^* = \frac{1}{2}\left(\frac{\left(\sum_{i\in I_L} g_{i,1}\right)^2}{\sum_{i\in I_L} g_{i,2}+\lambda} + \frac{\left(\sum_{i\in I_R} g_{i,1}\right)^2}{\sum_{i\in I_R} g_{i,2}+\lambda} - \frac{\left(\sum_{i\in I_j} g_{i,1}\right)^2}{\sum_{i\in I_j} g_{i,2}+\lambda}\right) - \gamma \quad (41)$$

where I_L and I_R are instance sets of left and right nodes after the split.

2.4 Machine learning platform

The optimal selection of yearly-specific predictive models was achieved using a grid search with stratified 5-fold cross-validation (5-fold CV). Grid search is a widely used technique in machine learning for hyperparameter optimization. Grid search in combination with stratified 5-fold CV aims to identify the optimal hyperparameters and select the best model from a space of hyperparameters by evaluating model performance using cross-validation on training set. Table 2.3 shows the hyperparameter values that were utilized during the model selection phase for these classifiers.

Key steps undertaken for conducting grid search with stratified 5-fold CV include:

1. The training set is split into 5 equally sized folds using a stratified random split, maintaining a consistent ratio of class labels across each fold to mitigate the effects of data imbalance (see Figure 3.2).
2. In each iteration, one fold is held-out as the validation set, and the remaining four folds are combined to form training set.

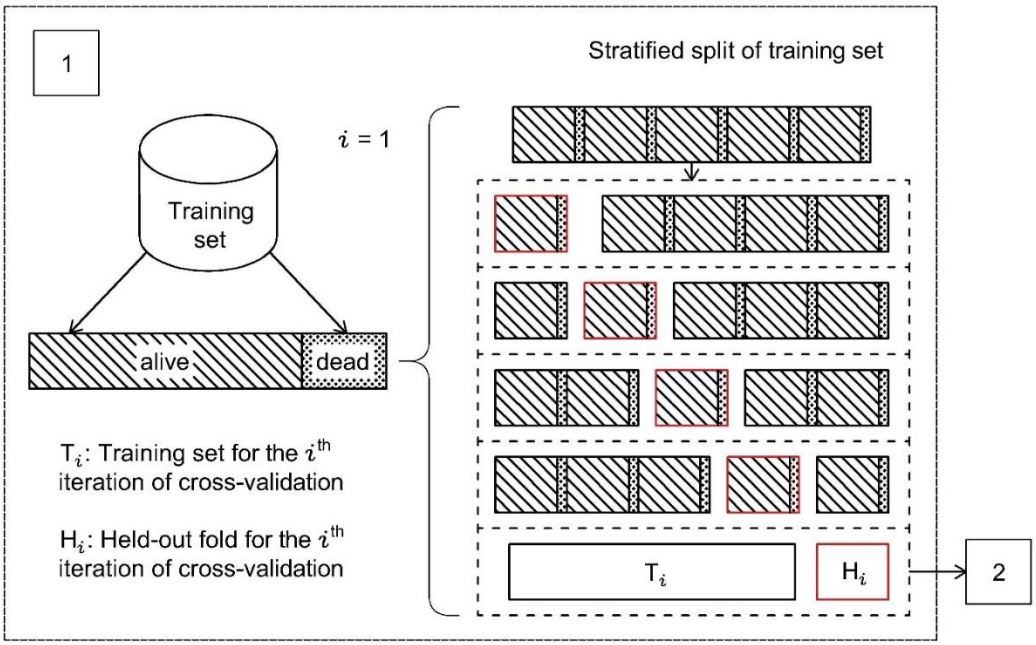

Figure 3.2: Stratified split of training set (first block of grid search with 5-fold CV)

3. Prior to model training, feature scaling, particularly, standardization is applied based on the training data (the combined four folds), ensuring uniformity (see Figure 3.3).
4. Models are trained on the training set using a predefined set of hyperparameters and evaluated on the validation set. In our approach, the area under the ROC curve (AUC) is the chosen metrics for evaluation, due to its independence of any specific decision threshold (see Figure 3.3).

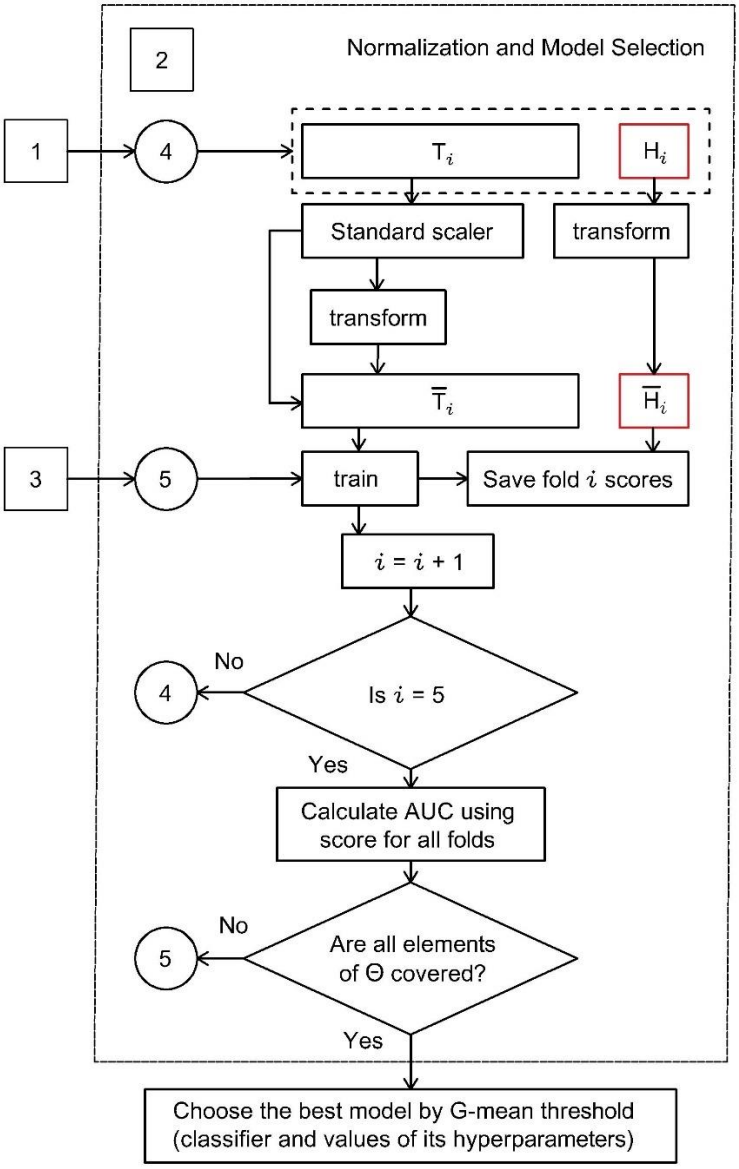

Figure 3.3: Normalization and model selection (second block of grid search with 5-fold CV)

5. This cycle is repeated for each fold and for every possible combination of hyperparameters within the hyperparameter space (Figure 3.5). As a result, each hyperparameter combination trained and validated 5 times, each time on different fold.

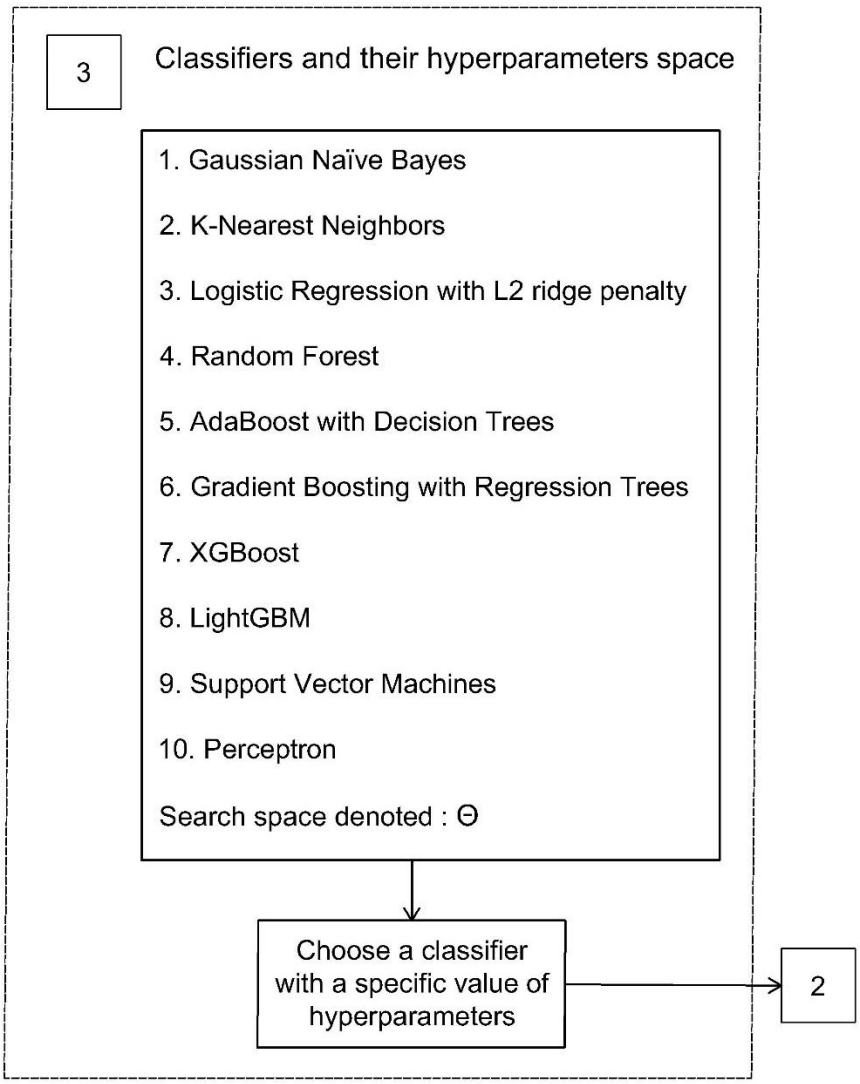

Figure 3.4: Classifiers and their hyperparameters space (third block of grid search with 5-fold CV)

6. After all iterations are completed, classifier with hyperparameter combination the highest average AUC across all 5 folds is selected as the best predictive model.

The year-specific classifier selected during grid search is retrained using the optimal hyperparameters on the complete training set after normalization to obtain final year-specific predictive model.

2.5 Performance evaluation metrics

The traditional accuracy, which simply calculates the proportion of correctly classified instances, can be misleading, potentially overestimating performance of predictive model when dealing with imbalanced data. To overcome this challenge, balanced accuracy (BA), precision, specificity, sensitivity, geometric mean of sensitivity and specificity (G-mean) as well as AUC, were used to evaluate the final year-specific classifier on the test set. These metrics are defined as following:

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

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

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

$$BA = \frac{specificity + sensitivity}{2}$$

$$Gmean = \sqrt{specificity * sensitivity}$$

where TP, FP, TN, and FN are number of true positives, false positives, true negatives, and false negatives, respectively.

Table 2.3: Search space of hyperparameters for model selection using grid search with cross-validation

Classifiers	Hyperparameter	Candidate Hyperparameter Space
Logistic Regression	penalty	L_2

(LRR)	regularization parameter C	0.001, 0.01, 0.1, 1, 10, 100
Perceptron (PER)	alpha	0.0001, 0.001, 0.01
	penalty	L_2, L_1, None
Gaussian Naive Bayes (GNB)	-	-
Random Forest (RF)	number of estimators	1, 5, 10, 50, 100
	maximum depth	5, 10, 20, 50
	maximum features	'log2', 'sqrt',
XGBoost (XGB)	maximum depth	5, 15, 100
	number of estimators	10, 100, 200, 500
	learning rate	0.001, 0.01, 0.1
LightGBM (LGB)	number of leaves	4, 8, 16
	number of estimators	10, 100, 200, 500
	learning rate	0.001, 0.01, 0.1
Gradient Boosting with Regression Trees (GBRT)	number of estimators	10, 100, 200, 500
	learning rate	0.001, 0.01, 0.1
AdaBoost (ADB)	number of estimators	10, 100, 200, 500
	learning rate	0.01, 0.1, 1
K-Nearest Neighbors (KNN)	number of neighbours	3, 5
Support Vector Machines (SVM)	penalty	L_2
	kernel	linear
	regularization parameter C	0.1, 0.5, 1, 5

2.6 Handling imbalanced classification

Imbalanced classification is a significant challenge in machine learning, particularly in medical datasets, where one class (mortality cases) is significantly underrepresented compared

to another (survival cases) [28]. This imbalance can lead models to perform poorly, especially in identifying less prevalent but important cases. Addressing this imbalance is crucial for improving the accuracy and effectiveness of predictive models in healthcare.

To address this issue, researchers have developed various strategies, which can be broadly categorized into data-level, algorithmic-level, ensemble methods and cost-sensitive learning, and ensemble techniques. These methods primarily aim to reduce the bias towards the majority class, there improving the model's ability to correctly predict the minority class instances [29]. Among these, data-level methods are especially important for adjusting the class distribution during the data preprocessing stage. Haixiang *et al.* [30] made a thorough analysis of methods handling data imbalance, and discovered that a majority of studies use resampling methods to address data imbalance, showing the importance of these techniques.

The most basic data-level methods are Random undersampling (RUS) and random oversampling (ROS). The former is implemented by ranndomly removing samples from the majority class in order to balance the class distribution [31]. In constrast, the latter is used to achieve balanced class distribution by generating additional samples in the minority class [31]. However, these approaches do not demonstrate any assumptions regarding the data and might ineffectively handle the complexity of issues presented in datasets used in medical fields. Therefore, there is a need for more advanced methods.

The Synthetic Minority Over-sampling Technique (SMOTE) is an advanced oversampling technique which is used to address a class imbalance [32]. It generates synthetic data points of the minority class by interpolating among existing minority class instances in feature space in contrast to operating in data space. For a given minority class sample x_i and its nearest neighbors x_{nn} synthetic samples are generated as follows:

$$x_{new} = x_i + \lambda(x_{nn} - x_i)$$

where λ is a random number within the range $[0,1]$. The choice of neighbors and the number of synthetic samples generated depend on the required level of over-sampling. If the amount of over-sampling needed is 300%, only 3 neighbors from the $k = 5$ nearest neighbors are randomly chosen.

Tomek Links, developed by Tomek [33] as an extension to Condensed Nearest Neighbor [34] methods, is an effective undersampling technique to refine the decision boundaries in imbalanced datasets. A Tomek Link is defined between a pair of instances from different classes, x_i and x_j, if there exists no instance x_z such that $d(x_i, x_z) < d(x_i, x_j)$ or $d(x_j, x_z) < d(x_i, x_j)$, where $d(\cdot)$ denotes the Euclidean distance. The primary strategy of this undersampling technique involves the removal instances from the majority class, which has Tomek Links, assuming these to be either noise or border cases. This results in a more defined, cleaner separation between classes, facilitating improved classifier accuracy and generalization. This technique is particularly valuable in preprocessing for imbalanced datasets, where the focus is on refining the class distribution and enhancing the learning algorithm's ability to distinguish between classes effectively.

In this study, sampling techniques were integrated as a part of grid search with stratified 5-fold CV.

2.7 SHapley Additive exPlanations

Our approach included conducting a SHapley Additive exPlanations [35] (SHAP) analysis to achieve two main objectives: firstly, to assess the individual significance of each feature in predicting mortality; and secondly, to understand how each feature influences the direction of the prediction. SHAP values were computed for each year-specific classifier selected during the model selection phase.

Chapter 3 – Results

This chapter presents the experimental findings from our study on hepatitis and tuberculosis datasets. We explore the performance of various classifiers under different conditions, comparing their efficacy without sampling techniques, with the application of SMOTE, and with the implementation of Tomek Links.

3.1 Hepatitis cohort findings

3.1.1 Prediction performance without sampling techniques

Following the methodology described previously, we identified the best year-specific models using a grid search with 5-fold CV. Table 3.1 presents the mean and standard deviation of the AUC estimates for each classifier in grid across the folds. Notably, the RF achieved the highest AUC for the years 2016 and 2018, and the LRR achieved the highest AUC for the 2017 and 2018 years. Subsequent evaluation of these optimal year-specific models on their corresponding year-specific test sets is summarized in Table 3.2.

3.2.2 Prediction performance with SMOTE

Following the assessment of predictive performance without sampling techniques, this subsection examines the impact of training models with SMOTE technique. As shown in Table 11, the application of SMOTE technique did not improve the model performance. Although the best year-specific classifiers maintained the same AUC for the years 2016 and 2017, there was notable decrease in AUC scores for the years 2018 and 2019.

Moreover, the performance evaluation on the year-specific test sets, as summarized in Table 6, further supports this observation. Despite the application of SMOTE sampling technique, the performance metrics did not exhibit significant improvement compared to the models trained without sampling techniques.

Table 3.1: AUC estimates (mean ± standard deviation) for each classifier over 5 folds of 5-fold cross-validation obtained on the yearly-specific training sets for hepatitis cohort without sampling

Classifier	AUC			
	2016	2017	2018	2019
LRR	0.793 ± 0.032	**0.789 ± 0.038**	0.773 ± 0.020	**0.785 ± 0.017**
SVM	0.586 ± 0.075	0.590 ± 0.121	0.610 ± 0.066	0.565 ± 0.103
PER	0.650 ± 0.101	0.588 ± 0.102	0.621 ± 0.090	0.635 ± 0.062
GNB	0.777 ± 0.026	0.778 ± 0.039	0.760 ± 0.021	0.776 ± 0.018
KNN	0.555 ± 0.006	0.547 ± 0.011	0.550 ± 0.008	0.556 ± 0.019
RF	**0.796 ± 0.024**	0.782 ± 0.029	**0.782 ± 0.018**	0.781 ± 0.013
XGB	0.781 ± 0.031	0.775 ± 0.032	0.774 ± 0.020	0.774 ± 0.018
ADB	0.795 ± 0.027	0.783 ± 0.034	0.776 ± 0.018	0.784 ± 0.019
LGB	0.792 ± 0.025	0.786 ± 0.029	0.781 ± 0.017	0.782 ± 0.018
GBRT	0.786 ± 0.028	0.788 ± 0.017	0.776 ± 0.017	0.779 ± 0.018

Table 3.2: Performance evaluation of the optimal year-specific classifier estimated on the corresponding test sets of hepatitis cohort without sampling

Classifier	Balanced Accuracy	AUC	Specificity	Sensitivity	Precision	G-mean
2016 cohort (RF)	0.689	0.749	0.807	0.571	0.035	0.679
2017 cohort (LRR)	0.705	0.778	0.709	0.700	0.033	0.704
2018 cohort (RF)	0.671	0.762	0.749	0.593	0.033	0.666
2019 cohort (LRR)	0.745	0.821	0.693	0.796	0.031	0.742

3.1.3 Prediction performance with Tomek links

Our final analysis within the hepatitis cohort evaluated the impact of undersampling by removing Tomek links. We observed similar patterns of model performance in training without sampling techniques, with RF achieving the highest AUC values for the years 2016 and 2018, and LRR achieving the highest AUC values for the 2017 and 2018 years, as detailed in Table 3.5. The detailed outcomes are shown in Table 3.5, with the performance evaluation on test sets provided subsequently in Table 3.6.

Performance evaluation on the test sets, shown in Table 3.6, indicated a slight improvement in AUC performance for models trained with Tomek links.

Table 3.3: AUC estimates (mean ± standard deviation) for each classifier over 5 folds of 5-fold cross-validation obtained on the yearly-specific training sets for hepatitis cohort with SMOTE

Classifier	AUC			
	2016	2017	2018	2019
LRR	0.789 ± 0.040	**0.788 ± 0.009**	**0.768 ± 0.019**	**0.785 ± 0.013**
SVM	0.785 ± 0.042	0.785 ± 0.008	0.784 ± 0.038	0.780 ± 0.012
PER	0.758 ± 0.045	0.756 ± 0.022	0.710 ± 0.018	0.766 ± 0.019
GNB	0.762 ± 0.041	0.772 ± 0.016	0.745 ± 0.017	0.776 ± 0.008
KNN	0.607 ± 0.017	0.594 ± 0.037	0.595 ± 0.015	0.608 ± 0.006
RF	0.776 ± 0.038	0.768 ± 0.004	0.759 ± 0.016	0.773 ± 0.014
XGB	0.778 ± 0.040	0.774 ± 0.002	0.745 ± 0.018	0.770 ± 0.017
ADB	**0.790 ± 0.044**	0.776 ± 0.006	0.764 ± 0.016	0.781 ± 0.013
LGB	0.782 ± 0.044	0.775 ± 0.006	0.768 ± 0.019	0.781 ± 0.013
GBRT	0.777 ± 0.041	0.773 ± 0.003	0.745 ± 0.018	0.768 ± 0.016

Table 3.4: Performance evaluation of the optimal year-specific classifier estimated on the corresponding test sets for hepatitis cohort with SMOTE

Classifier	Balanced Accuracy	AUC	Specificity	Sensitivity	Precision	G-mean
2016 cohort (ADB)	0.691	0.763	0.696	0.686	0.027	0.690
2017 cohort (LRR)	0.705	0.777	0.696	0.709	0.033	0.703
2018 cohort (LRR)	0.661	0.746	0.727	0.593	0.031	0.656
2019 cohort (LRR)	0.745	0.821	0.693	0.796	0.031	0.742

Table 3.5: AUC estimates (mean ± standard deviation) for each classifier over 5 folds of 5-fold cross-validation obtained on the yearly-specific training sets for hepatitis cohort with Tomek links

Classifier	AUC			
	2016	2017	2018	2019
LRR	0.792 ± 0.019	**0.790 ± 0.032**	0.770 ± 0.018	**0.785 ± 0.027**
SVM	0.581 ± 0.059	0.596 ± 0.069	0.591 ± 0.042	0.593 ± 0.049
PER	0.591 ± 0.070	0.639 ± 0.110	0.597 ± 0.065	0.567 ± 0.087
GNB	0.770 ± 0.024	0.778 ± 0.034	0.755 ± 0.016	0.773 ± 0.032
KNN	0.545 ± 0.021	0.570 ± 0.023	0.555 ± 0.011	0.559 ± 0.012
RF	**0.796 ± 0.022**	0.784 ± 0.033	**0.784 ± 0.012**	0.782 ± 0.027
XGB	0.791 ± 0.020	0.784 ± 0.034	0.775 ± 0.013	0.777 ± 0.024
ADB	0.795 ± 0.022	0.782 ± 0.034	0.774 ± 0.017	0.784 ± 0.027
LGB	0.793 ± 0.017	0.784 ± 0.032	0.783 ± 0.012	0.783 ± 0.025
GBRT	0.793 ± 0.018	0.784 ± 0.036	0.778 ± 0.015	0.780 ± 0.023

Table 3.6: Performance evaluation of the optimal year-specific classifier estimated on the corresponding test sets of hepatitis cohort with Tomek links

Classifier	Balanced Accuracy	AUC	Specificity	Sensitivity	Precision	G-mean
2016 cohort	0.682	0.753	0.821	0.543	0.035	0.668
2017 cohort	0.705	0.778	0.709	0.700	0.033	0.704
2018 cohort	0.681	0.764	0.726	0.634	0.033	0.678
2019 cohort	0.741	0.822	0.701	0.777	0.032	0.739

3.1.4 Discussion of hepatitis cohort findings

Experimental results showed that classifiers trained with Tomek links showed the highest performances in terms of AUC compared to those trained without sampling and with SMOTE. Table 3.5 shows that classifier developed for each specific year achieved an AUC in the range of 0.78 to 0.8, which is considered ‘fair’ (close to ‘good’) for diagnostic tests [36]. Although the application of Tomek links slightly improved the performance of the models, precision metric was not improved.

3.2 Tuberculosis cohort findings

3.2.1 Prediction performance without sampling techniques

Similar to the hepatitis cohort, we identified the best year-specific models for each classifier by employing a grid search combined with 5-fold CV. The GBRT and XGB classifiers showed the best performance in terms of AUC across different subcohorts. The detailed AUC estimates for each classifier are summarized in Table 3.7.

Following the cross-validation, the best year-specific classifiers were further evaluated on their respective year-specific test sets, as summarized in Table 3.8.

Table 3.7: AUC estimates (mean ± standard deviation) for each classifier over 5 folds of 5-fold cross-validation obtained on the yearly-specific training sets for tuberculosis cohort without sampling

Classifier	AUC			
	2016	2017	2018	2019
LRR	0.786 ± 0.009	0.790 ± 0.009	0.805 ± 0.010	0.815 ± 0.004
SVM	0.586 ± 0.075	0.601 ± 0.008	0.636 ± 0.100	0.622 ± 0.056
PER	0.658 ± 0.102	0.743 ± 0.016	0.651 ± 0.064	0.696 ± 0.078
GNB	0.776 ± 0.011	0.779 ± 0.012	0.780 ± 0.012	0.795 ± 0.005
KNN	0.608 ± 0.009	0.591 ± 0.037	0.584 ± 0.009	0.588 ± 0.012
RF	0.814 ± 0.011	0.802 ± 0.011	0.814 ± 0.014	0.829 ± 0.005
XGB	0.818 ± 0.009	**0.811 ± 0.011**	**0.817 ± 0.015**	**0.831 ± 0.005**
ADB	0.809 ± 0.011	0.806 ± 0.014	0.813 ± 0.015	0.827 ± 0.005
LGB	0.819 ± 0.007	0.808 ± 0.011	0.814 ± 0.014	0.830 ± 0.005
GBRT	**0.820 ± 0.011**	0.809 ± 0.013	0.815 ± 0.015	0.829 ± 0.005

Table 3.8: Performance evaluation of the optimal year-specific classifier estimated on the corresponding test sets of tuberculosis cohort without sampling

Classifier	Balanced Accuracy	AUC	Specificity	Sensitivity	Precision	G-mean
2016 cohort (GBRT)	0.736	0.749	0.786	0.687	0.084	0.734
2017 cohort (XGB)	0.716	0.799	0.656	0.777	0.051	0.714
2018 cohort (XGB)	0.741	0.820	0.744	0.739	0.060	0.742
2019 cohort (XGB)	0.760	0.839	0.739	0.781	0.056	0.759

3.2.2 Prediction performance with SMOTE

Following the assessment of predictive performance without sampling techniques, this subsection examines the impact of training models with SMOTE technique. As shown in Table 3.9, the application of SMOTE technique did not improve the model performance. Although the best year-specific classifiers maintained the same AUC for the years 2016 and 2017, there was notable decrease in AUC scores for the years 2018 and 2019.

Moreover, the performance evaluation on the year-specific test sets, as summarized in Table 3.10, further supports this observation. Despite the application of SMOTE sampling technique, the performance metrics did not exhibit significant improvement compared to the models trained without sampling techniques.

3.2.3 Prediction performance with Tomek links

Following the assessment of predictive performance with SMOTE, this subsection examines the impact of training models with Tomek links technique. The results are presented in Table 3.11, showing the mean and standard deviation of the AUC estimates for each classifier across the folds. Notably, the LGB and GBRT classifiers achieved the highest AUC in their respective subcohorts.

3.2.4 Discussion of tuberculosis cohort findings

Similarly to hepatitis cohort, experimental results showed that classifiers trained with Tomek links showed the highest performances in terms of AUC compared to those trained without sampling and with SMOTE. Table 3.11 shows that classifier developed for each specific year achieved an AUC in the range of 0.815 to 0.830, which is considered ‘good’ for diagnostic tests [36]. Although the application of Tomek links slightly improved the performance of the models, precision metric was not improved.

Table 3.9: AUC estimates (mean ± standard deviation) for each classifier over 5 folds of 5-fold cross-validation obtained on the yearly-specific training sets for tuberculosis cohort with SMOTE

Classifier	AUC			
	2016	2017	2018	2019
LRR	0.796 ± 0.004	0.788 ± 0.009	0.804 ± 0.012	0.815 ± 0.015
SVM	0.785 ± 0.042	0.785 ± 0.008	0.784 ± 0.038	0.780 ± 0.012
PER	0.758 ± 0.045	0.763 ± 0.016	0.756 ± 0.040	0.766 ± 0.019
GNB	0.775 ± 0.010	0.776 ± 0.012	0.781 ± 0.012	0.796 ± 0.018
KNN	0.643 ± 0.017	0.594 ± 0.037	0.628 ± 0.017	0.636 ± 0.007
RF	0.808 ± 0.007	0.805 ± 0.011	0.796 ± 0.011	0.810 ± 0.014
XGB	**0.818 ± 0.007**	**0.810 ± 0.011**	0.802 ± 0.007	**0.819 ± 0.013**
ADB	0.809 ± 0.006	0.804 ± 0.014	**0.805 ± 0.012**	0.818 ± 0.011
LGB	0.817 ± 0.005	0.809 ± 0.011	0.804 ± 0.009	0.818 ± 0.011
GBRT	0.816 ± 0.004	0.809 ± 0.013	0.799 ± 0.008	0.817 ± 0.012

Table 3.10: Performance evaluation of the optimal year-specific classifier estimated on the corresponding test sets for tuberculosis cohort with SMOTE

Classifier	Balanced Accuracy	AUC	Specificity	Sensitivity	Precision	G-mean
2016 cohort (XGB)	0.732	0.821	0.763	0.702	0.079	0.733
2017 cohort (XGB)	0.715	0.796	0.651	0.780	0.049	0.713
2018 cohort (ADB)	0.736	0.809	0.670	0.802	0.051	0.733
2019 cohort (XGB)	0.743	0.829	0.731	0.755	0.053	0.743

Table 3.11: AUC estimates (mean ± standard deviation) for each classifier over 5 folds of 5-fold cross-validation obtained on the yearly-specific training sets for tuberculosis cohort with Tomek links

Classifier	AUC			
	2016	2017	2018	2019
LRR	0.791 ± 0.010	0.798 ± 0.010	0.805 ± 0.014	0.815 ± 0.005
SVM	0.581 ± 0.059	0.596 ± 0.069	0.591 ± 0.042	0.593 ± 0.049
PER	0.686 ± 0.059	0.641 ± 0.071	0.700 ± 0.078	0.657 ± 0.108
GNB	0.772 ± 0.009	0.776 ± 0.010	0.779 ± 0.012	0.795 ± 0.010
KNN	0.622 ± 0.009	0.607 ± 0.011	0.584 ± 0.013	0.593 ± 0.010
RF	0.821 ± 0.012	0.818 ± 0.009	0.811 ± 0.009	0.827 ± 0.004
XGB	0.827 ± 0.012	0.822 ± 0.012	0.815 ± 0.007	0.829 ± 0.007
ADB	0.817 ± 0.013	0.814 ± 0.012	0.813 ± 0.011	0.827 ± 0.007
LGB	**0.830 ± 0.012**	0.823 ± 0.011	**0.816 ± 0.010**	**0.830 ± 0.005**
GBRT	0.829 ± 0.011	**0.824 ± 0.013**	0.814 ± 0.008	0.780 ± 0.023

Table 3.12: Performance evaluation of the optimal year-specific classifier estimated on the corresponding test sets of tuberculosis cohort with Tomek links

Classifier	Balanced Accuracy	AUC	Specificity	Sensitivity	Precision	G-mean
2016 cohort (LGB)	0.743	0.828	0.744	0.743	0.077	0.744
2017 cohort (GBRT)	0.725	0.807	0.766	0.684	0.064	0.724
2018 cohort (LGB)	0.739	0.819	0.729	0.748	0.058	0.739
2019 cohort (LGB)	0.759	0.838	0.736	0.783	0.055	0.759

3.3 SHAP analysis

3.3.1 Hepatitis cohort

For the hepatitis disease, we calculated SHAP values for the RF classifier for the 2016 and 2018 cohorts, and the LRR classifier for the 2017 and 2019 cohorts. It is important to highlight that we analyzed SHAP values for all clinical variables within the training dataset, as feature selection was not performed. To summarize the SHAP values from all cohorts, we calculated the average of the mean absolute SHAP values (AMAS) for each variable. The variables were then ranked based on their AMAS values: age, sex, hepatitis type, ethnicity, hepatitis duration, cirrhosis, and hospitalization times, with their respective AMAS values being 0.697, 0.298, 0.121, 0.099, 0.084, 0.039, and 0.020. The results highlight age, sex, hepatitis type, and ethnicity as the top factors influencing the prediction of one-year mortality.

SHAP analysis demonstrated that older age, male sex, chronic hepatitis C and ethnic minorities associated with higher risk of mortality. Multiple studies have demonstrated a correlation between older age and a higher mortality in chronic viral hepatitis patients [37-40]. In [40], the authors performed a nationwide study on mortality associated with HCV and HBV based on the data from the Italian National Cause of Death Register. The results demonstrated that patients in the older age categories have higher rates of mortality than those in the younger age group. Similarly, research conducted on a nationwide register-based cohort in Denmark revealed that individuals with HBV at an older age have a higher risk of mortality than younger patients [38].

Several papers have looked into the connection between sex and hepatitis mortality. A population-based cohort study in France indicated that male HBV patients have higher all-cause and HBV-related mortality rates compared to female patients [41]. Findings of this study

correspond with this study, which determined that males with CVH have a higher risk of mortality than females with the same condition.

An extensive 13-year population-based investigation carried out in Asia indicated that individuals with HCV had a significantly higher risk of cardiovascular events and overall death than those with HBV [42]. Study from Switzerland also found that patients with HCV have higher mortality risk than patients with HBV [43]. This research found a similar pattern, which supports these conclusions.

An additional subject that draws attention is the connection between ethnicity and death among patients. Several studies examined the prevalence and mortality of hepatitis in developed nations while accounting for ethnic differences [44-46]. A study based on the Chronic Hepatitis Cohort Study (CHeCS) in the United States detected that African Americans had the highest mortality rates, 26% higher than white patients. In contrast, Asian American/American Indian/Pacific Islander (AAPI) patients had the lowest death rates [46]. Previous studies [47-48] based on the UNEHS database found a considerable difference in death rates between ethnic groups, which is consistent with our conclusion.

3.3.2 Tuberculosis cohort

Following the similar approach, we performed SHAP analysis for the tuberculosis disease cohorts for selected classifiers during the model selection phase, specifically for the LGB classifier for the 2016, 2018 and 2019 cohorts, and the GBRT for the 2017 cohort.

The analysis ranked the variables by their AMAS values: age, type of TB, ethnicity, duration of TB, number of hospitalizations, diabetes, residence, HIV and hepatitis with their respective AMAS values being 0.753, 0.221, 0.215, 0.203, 0.168, 0.163, 0.038, 0.017, 0.007 and 0.003. The results highlight age, type of TB, ethnicity and duration of TB, as the top factors influencing the prediction of one-year mortality.

SHAP analysis demonstrated that older age, respiratory tuberculosis bacteriologically confirmed, ethnic minorities, longer duration and male sex associated with higher risk of mortality in TB patients.

A consistent finding across multiple studies is the correlation between older age and increased mortality among TB patients [49-52]. The research by Anne Christine Nordholm *et al.* [51] conducted based on the Danish population-based cohort showed that age>50 results in higher mortality for TB patients. Similarly, findings by Wang *et al.* [52] highlighted that older age associated with high risk of all-cause mortality in TB patients.

Sex has also emerged as a significant factor of TB mortality with male patients exhibiting higher risk of mortality [53-56]. The study by Choi *et al.* [55] on mortality of TB patients in Republic of Korea provided evidence that male TB patients had higher mortality than their female counterparts. A retrospective cohort study [56] of Taiwanese patients also showed that males exhibited higher hazards of all-cause mortality, compared to female patients.

Several studies evaluated the connection between ethnicity and mortality in TB patients. A study based on the data from the Estonian Tuberculosis Registry [57] showed that non-Estonian ethnicity contributed to higher mortality in TB patients. An retrospective population-based cohort study [58] of TB patients in Brazil also investigated the effect of race categories on mortality. Study showed that browns had higher risk of death among other categories. This study on TB patients aligns with previous studies [47-48] based on the UNEHS database, which is consistent with our conclusion.

This comprehensive analysis emphasizes the importance of demographic and disease-specific variables, notably age and the specific characteristics of the disease, in mortality prediction. The consistent identification of these factors across different cohorts and diseases underscores their critical role in the predictive modeling of healthcare outcome.

Chapter 4 – Redefining the definition of traditional performance metrics

In the context of binary classification, conventional evaluation metrics such as True Positive Rate (TPR), True Negative Rate (TNR), and Positive Predictive Value (PPV) play a pivotal role in assessing the performance of classification models. These metrics, defined as follows, are instrumental in quantifying the model's ability to accurately classify instances into positive (P) or negative (N) categories:

- **True Positive Rate (TPR)**, also referred to as sensitivity, quantifies the likelihood that a model correctly classifies an instance as positive when it indeed belongs to the positive class. Mathematically, it is expressed as $\mathrm{TPR} = P(s(\mathbf{X}) > t | \mathbf{X} \in \mathrm{P})$, where $s(\mathbf{X})$ denotes the score or probability assigned by the model to instance $\mathbf{X}$, and t represents the classification threshold.
- **True Positive Rate (TPR)**, known as specificity, measures the probability that a model accurately classifies an instance as negative when it is truly negative. Mathematically, it is expressed as $\mathrm{TNR} = P(s(\mathbf{X}) \leq t | \mathbf{X} \in \mathrm{N})$.
- **Positive Predictive Value (PPV)**, or precision, indicates the probability that an instance classified by the model as positive is indeed positive, defined by $\mathrm{PPV} = P(\mathbf{X} \in \mathrm{P} | s(\mathbf{X}) > t)$.
- F_1 score, harmonic mean of PPV and TPR, defined as:

$$F_1 = 2 \times \frac{PPV \times TPR}{PPV + TPR}$$

Traditionally, these metrics depend on the application of a single threshold, t, to the model's output scores to distinguish between positive and negative classifications. While this single thresholding approach is straightforward, it may not always provide the necessary

flexibility for specific decision-making, particularly in contexts where the consequences of false positives and false negatives diverge significantly.

This study introduces an innovative approach to classification evaluation through the adoption of **double thresholding**. Unlike the traditional method, double thresholding utilizes two thresholds, t_{low} and t_{high} to define two distinct regions for classification decisions:

1. **Positive classification region:** An instance is classified as positive if its model score, $s(\mathbf{X})$, falls between t_{low} and t_{high}, i.e., $t_{low} < s(\mathbf{X}) < t_{high}$.
2. **Negative classification region**: An instance is classified as negative if its score lies outside the aforementioned range, either below t_{low} or above t_{high}.

To demonstrate that our approach enhances the model's performance and effectively navigates the trade-off between sensitivity and specificity, we conducted experiments using a synthetically generated 2D dataset. The dataset was specifically designed with instances of the positive class generated from a unimodal normal distribution, defined by a singular mean and standard deviation. This ensured that the positive instances fell within a specific interval. Conversely, instances of the negative class were produced using a bimodal distribution, characterized by two distinct means and a shared standard deviation, positioned outside the specified interval. The distribution of the generated synthetic dataset is depicted in Figure 5.1.

Subsequently, a logistic regression model was trained utilizing the standard training process. The process for identifying the optimal single decision threshold involved iterating the threshold value from 0 to 1 in increments of 0.001, while calculating the F_1 score on the training dataset. The threshold yielding the highest F_1 score on the training dataset was chosen. Moreover, the model underwent fine-tuning across a spectrum of threshold pairs to discover the combination that maximized the F_1 score on the training dataset. The performance of three models— the baseline model, the model with single thresholding, and the model with double

thresholding—was then evaluated on the test set. The results obtained from this evaluation are presented in Table 5.1:

Table 5.1 Results of experiment with double thresholding

Model	Accuracy	Specificity	Sensitivity	Precision	F_1 score
Baseline	0.804	1.0	0.0	0.0	0.0
Single thresholding	0.548	0.446	0.963	0.298	0.455
Double thresholding	0.823	0.812	0.884	0.535	0.667

As illustrated in the Table 5.1, in our synthetic dataset analysis, the double thresholding method outperformed both the default classifier settings and single threshold optimization across several key metrics, notably increasing the F_1 score significantly. This improvement highlights the potential of double thresholding to achieve a more balanced performance, particularly in enhancing recall without a corresponding unacceptable decrease in overall accuracy. This method is invaluable in applications where the cost of misclassification is asymmetric, such as medical diagnoses, fraud detection, and other sensitive areas.

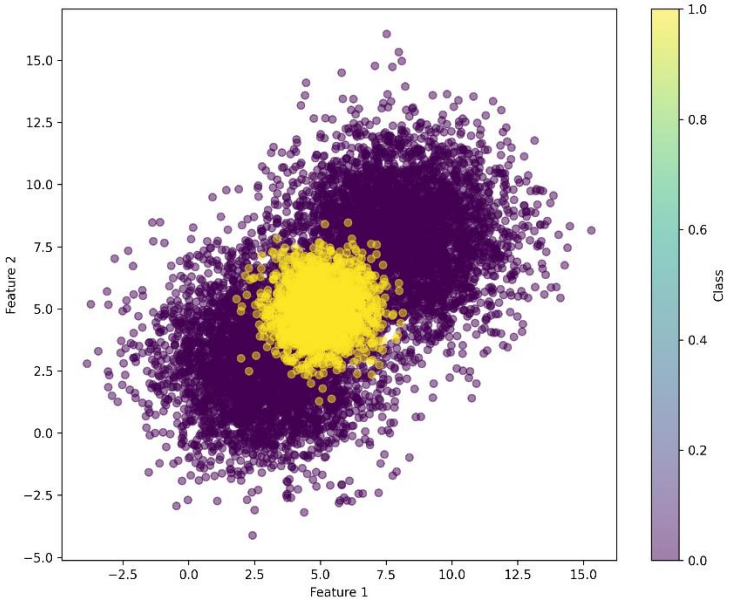

Figure 5.1: Visualization of the synthetic dataset

Chapter 5 – Conclusion and future works

In this study, an advanced machine learning platform was developed to predict one-year mortality in CVH and TB patients based on administrative health records. The strength of this work is that the data is collected from a population-based registry and for a sufficiently long time that enables constructing classifiers based on a true random sample of the population. Although our investigation utilized a relatively limited number of administrative features, the constructed classifiers achieved an AUC in the range from 0.74 to 0.83, rated as ‘fair’ and approaching ‘good’, according to standard diagnostic test metrics. These AUC results indicate a considerable achievement in predicting one-year mortality in CVH and TB patients using solely administrative health data. Another SMOTE and Tomek links were further applied to improve the performance of the models; however, Tomek links only slightly improved the AUC results whereas SMOTE has no effect on the performance.

The study identified that age, sex, type of hepatitis and ethnicity were important predictors of mortality for patients with hepatitis, consistent with existing research on the subject. Moreover, the study identified that important clinical variables for predicting TB mortality are age, type of TB, ethnicity and duration of TB. These findings have significant implications, potentially leading to better tailored treatment plans and approaches for managing hepatitis and TB in various healthcare environments. Moreover, these results could also help public health campaigns and encourage the adoption of healthier lifestyles to prevent mortality from hepatitis and TB.

This study has several limitations. From a clinical perspective, our study does not include laboratory data and detailed medical histories of the patients. Moreover, important information regarding the precise timestamp attached to each comorbidity, as well as anthropometric indices (BMI), and alcohol use, were not considered. Including these details

could potentially enhance the performance of similar predictive models built in the future. However, incorporating such data would incur additional costs.

From the standpoint of machine learning, this study lacks a feature selection stage. Although, this stage is less crucial in our current study due to the limited number of features and the large sample size, the inclusion of clinical notes or laboratory data could introduce additional features. In such a scenario, it would be generally expected to have a feature selection stage to mitigate the challenges posed by the curse of dimensionality in pattern recognition [20] (also referred to as the peaking phenomenon [59]). These aspects will be the subject of our future investigations.

Future research should focus on combining administrative data with key comorbidities, laboratory data, body measurements, and patients' medical history to create more accurate and robust predictive models, further enhancing patient care and treatment results. Moreover, as the introducing new features would require including of feature selection stage.

References

[1] C. L. Lai, V. Ratziu, M.-F. Yuen, and T. Poynard, "Viral hepatitis B," *The Lancet*, vol. 362, no. 9401, pp. 2089–2094, Dec. 2003. doi:10.1016/s0140-6736(03)15108-2

[2] T. Poynard, M.-F. Yuen, V. Ratzin, and C. L. Lai, "Viral hepatitis C," *The Lancet*, vol. 362, no. 9401, pp. 2095–2100, Dec. 2003. doi:10.1016/s0140-6736(03)15109-4

[3] "Global progress report on HIV, viral hepatitis and sexually transmitted infections, 2021," World Health Organization, https://www.who.int/publications/i/item/9789240027077 (accessed Jun. 10, 2023).

[4] A. Ashimkhanova et al., "Epidemiological characteristics of chronic viral hepatitis in Kazakhstan: Data from Unified Nationwide Electronic Healthcare System 2014–2019," *Infection and Drug Resistance*, vol. Volume 15, pp. 3333–3346, Jun. 2022. doi:10.2147/idr.s363609

[5] "Combating hepatitis B and C to reach elimination by 2030: Advocacy brief," World Health Organization, https://apps.who.int/iris/handle/10665/206453 (accessed Jun. 10, 2023).

[6] C. Dye, "Global Epidemiology of Tuberculosis," *The Lancet*, vol. 367, no. 9514, pp. 938–940, Mar. 2006. doi:10.1016/s0140-6736(06)68384-0

[7] "Global tuberculosis report 2023," World Health Organization, https://www.who.int/teams/global-tuberculosis-programme/tb-reports/global-tuberculosis-report-2023 (accessed Apr. 4, 2024).

[8] Y. Sakko et al., "Epidemiology of tuberculosis in Kazakhstan: Data from the Unified National Electronic Healthcare System 2014–2019," *BMJ Open*, vol. 13, no. 10, Oct. 2023. doi:10.1136/bmjopen-2023-074208

[9] F. R. Albogamy et al., "Decision support system for predicting survivability of hepatitis patients," *Frontiers in Public Health*, vol. 10, Apr. 2022. doi:10.3389/fpubh.2022.862497

[10] K.S. Bhargav et al., "Application of machine learning classification algorithms on hepatitis dataset," *International Journal of Applied Engineering Research*, vol. 13, no. 12732-12737 (2018). https://www.ripublication.com/ijaer18/ijaerv13n16_45.pdf

[11] P. Yildirim, "Filter based feature selection methods for prediction of risks in hepatitis disease," *International Journal of Machine Learning and Computing*, vol. 5, no. 4, pp. 258–263, Aug. 2015. doi:10.7763/ijmlc.2015.v5.517

[12] G. Obaido et al., "An interpretable machine learning approach for hepatitis B diagnosis," *Applied Sciences*, vol. 12, no. 21, p. 11127, Nov. 2022. doi:10.3390/app122111127

[13] M. H. Lino Ferreira da Silva Barros et al., “Benchmarking machine learning models to assist in the prognosis of tuberculosis,” *Informatics*, vol. 8, no. 2, p. 27, Apr. 2021. doi:10.3390/informatics8020027

[14] A. Peng et al., “Explainable machine learning for early predicting treatment failure risk among patients with TB-diabetes comorbidity,” *Scientific Reports*, vol. 14, no. 1, Mar. 2024. doi:10.1038/s41598-024-57446-8

[15] C. M. Sauer et al., “Feature selection and prediction of treatment failure in tuberculosis,” *PLOS ONE*, vol. 13, no. 11, Nov. 2018. doi:10.1371/journal.pone.0207491

[16] M. Asad, A. Mahmood, and M. Usman, “A machine learning-based framework for predicting treatment failure in tuberculosis: A case study of six countries,” *Tuberculosis*, vol. 123, p. 101944, Jul. 2020. doi:10.1016/j.tube.2020.101944

[17] A. Gusmanov et al., “Review of the research databases on population-based registries of Unified Electronic Healthcare System of kazakhstan (UNEHS): Possibilities and limitations for epidemiological research and real-world evidence,” *International Journal of Medical Informatics*, vol. 170, p. 104950, Feb. 2023. doi:10.1016/j.ijmedinf.2022.104950

[18] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, 2009.

[19] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995. doi:10.1007/bf00994018

[20] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley, 2001.

[21] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. doi:10.1023/a:1010933404324

[22] T. Chen and C. Guestrin, “XGBoost,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 2016. doi:10.1145/2939672.2939785

[23] G. Ke. et al.: LightGBM: A Highly Efficient Gradient Boosting Decision Tree, in: I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Adv Neural Inf Process Syst, Curran Associates, Inc., 2017. https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[24] J. H. Friedman, “Greedy function approximation: A gradient boosting machine.,” *The Annals of Statistics*, vol. 29, no. 5, Oct. 2001. doi:10.1214/aos/1013203451

[25] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, Aug. 1997. doi:10.1006/jcss.1997.1504

[26] A. Zollanvari, *Machine Learning with Python: Theory and Implementation*. Springer Nature, 2023.

[27] C. M. Bishop, *Pattern Recognition and Machine Learning.* Springer New York, 2006.

[28] J. Liu et al., “Mortality prediction based on imbalanced high-dimensional ICU big data,” *Computers in Industry*, vol. 98, pp. 218–225, Jun. 2018. doi:10.1016/j.compind.2018.01.017

[29] Haibo He and E. A. Garcia, “Learning from Imbalanced Data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009. doi:10.1109/tkde.2008.239

[30] G. Haixiang et al., “Learning from class-imbalanced data: Review of methods and applications,” *Expert Systems with Applications*, vol. 73, pp. 220–239, May 2017. doi:10.1016/j.eswa.2016.12.035

[31] G. E. Batista, R. C. Prati, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data,” *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, Jun. 2004. doi:10.1145/1007730.1007735

[32] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique", *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321-357, 2002.

[33] I. Tomek, "Two modifications of CNN", *IEEE Trans. Syst. Man Cybern.*, 1976.

[34] P. E. Hart, "The Condensed Nearest Neighbor," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515–516, 1968.

[35] Lundberg, S.M., Allen, P.G., Lee, S.-I.: A Unified Approach to Interpreting Model Predictions, 2017. https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[36] J. M. Pines, C. R. Carpenter, A. S. Raja, and J. D. Schuur, *Evidence-based emergency care*, Nov. 2012. doi:10.1002/9781118482117

[37] H. B. El-Serag, J. Kramer, Z. Duan, and F. Kanwal, “Epidemiology and outcomes of hepatitis C infection in elderly us veterans,” *Journal of Viral Hepatitis*, vol. 23, no. 9, pp. 687–696, Apr. 2016. doi:10.1111/jvh.12533

[38] S. Bollerup et al., “Mortality and cause of death in persons with chronic hepatitis B virus infection versus healthy persons from the general population in Denmark,” *Journal of Viral Hepatitis*, vol. 29, no. 9, pp. 727–736, Jun. 2022. doi:10.1111/jvh.13713

[39] M. Alavi et al., “Mortality trends among people with hepatitis B and C: A population-based linkage study, 1993-2012,” *BMC Infectious Diseases*, vol. 18, no. 1, May 2018. doi:10.1186/s12879-018-3110-0

[40] U. Fedeli, E. Grande, F. Grippo, and L. Frova, “Mortality associated with hepatitis C and hepatitis B virus infection: A nationwide study on multiple causes of Death Data,” *World Journal of Gastroenterology*, vol. 23, no. 10, p. 1866, 2017. doi:10.3748/wjg.v23.i10.1866

[41] C. Montuclard et al., “Causes of death in people with chronic HBV infection: A population-based Cohort Study,” *Journal of Hepatology*, vol. 62, no. 6, pp. 1265–1271, Jun. 2015. doi:10.1016/j.jhep.2015.01.020

[42] V. C.-C. Wu et al., “Comparison of cardiovascular outcomes and all-cause mortality in patients with chronic hepatitis B and C: A 13-year nationwide population-based study in Asia,” *Atherosclerosis*, vol. 269, pp. 178–184, Feb. 2018. doi:10.1016/j.atherosclerosis.2018.01.007

[43] O. Keiser et al., “Trends in Hepatitis c-related mortality in Switzerland,” *Journal of Viral Hepatitis*, vol. 25, no. 2, pp. 152–160, Nov. 2017. doi:10.1111/jvh.12803

[44] B. Emmanuel, M. D. Shardell, L. Tracy, S. Kottilil, and S. S. El-Kamary, “Racial disparity in all-cause mortality among hepatitis C virus-infected individuals in a general US population, Nhanes III,” *Journal of Viral Hepatitis*, vol. 24, no. 5, pp. 380–388, Dec. 2016. doi:10.1111/jvh.12656

[45] D. Bixler et al., “Mortality among patients with chronic hepatitis B infection: The chronic hepatitis cohort study (checs),” *Clinical Infectious Diseases*, vol. 68, no. 6, pp. 956–963, Jul. 2018. doi:10.1093/cid/ciy598

[46] M. Lu et al., “Trends in cirrhosis and mortality by age, sex, race, and antiviral treatment status among US chronic hepatitis B patients (2006-2016),” *Journal of Clinical Gastroenterology*, vol. 56, no. 3, pp. 273–279, Mar. 2021. doi:10.1097/mcg.0000000000001522

[47] S. Yerdessov et al., “Epidemiological characteristics and climatic variability of viral meningitis in Kazakhstan, 2014–2019,” *Frontiers in Public Health*, vol. 10, Jan. 2023. doi:10.3389/fpubh.2022.1041135

[48] A. Midlenko et al., “Prevalence, incidence, and mortality rates of breast cancer in Kazakhstan: Data from the Unified National Electronic Health System, 2014–2019,” *Frontiers in Public Health*, vol. 11, Apr. 2023. doi:10.3389/fpubh.2023.1132742

[49] A.-S. Christensen, P. H. Andersen, N. Obel, A. B. Andersen, and Roed, “Long-term mortality in patients with pulmonary and extrapulmonary tuberculosis: A Danish nationwide Cohort Study,” *Clinical Epidemiology*, p. 405, Nov. 2014. doi:10.2147/clep.s65331

[50] K. Romanowski et al., “Long-term all-cause mortality in people treated for tuberculosis: A systematic review and meta-analysis,” *The Lancet Infectious Diseases*, vol. 19, no. 10, pp. 1129–1137, Oct. 2019. doi:10.1016/s1473-3099(19)30309-3

[51] A. C. Nordholm et al., “Mortality, risk factors, and causes of death among people with tuberculosis in Denmark, 1990-2018,” *International Journal of Infectious Diseases*, vol. 130, pp. 76–82, May 2023. doi:10.1016/j.ijid.2023.02.024

[52] Y. Wang et al., “Changes in tuberculosis burden and its associated risk factors in Guizhou Province of China during 2006–2020: An observational study,” *BMC Public Health*, vol. 24, no. 1, Feb. 2024. doi:10.1186/s12889-024-18023-w

[53] J.-Y. Feng et al., “Gender differences in treatment outcomes of tuberculosis patients in Taiwan: A prospective observational study,” *Clinical Microbiology and Infection*, vol. 18, no. 9, Sep. 2012. doi:10.1111/j.1469-0691.2012.03931.x

[54] M.-E. Jimenez-Corona, “Gender differentials of pulmonary tuberculosis transmission and reactivation in an endemic area,” *Thorax*, vol. 61, no. 4, pp. 348–353, Apr. 2006. doi:10.1136/thx.2005.049452

[55] H. Choi et al., “Long-term mortality of post-tuberculous survivors in Korea: A population-based longitudinal study,” 10.02 - *Tuberculosis and non-tuberculous mycobacterial diseases*, Sep. 2022. doi:10.1183/13993003.congress-2022.396

[56] V. Chidambaram et al., “Male sex is associated with worse microbiological and clinical outcomes following tuberculosis treatment: A retrospective cohort study, a systematic review of the literature, and meta-analysis,” *Clinical Infectious Diseases*, vol. 73, no. 9, pp. 1580–1588, Jun. 2021. doi:10.1093/cid/ciab527

[57] K. Blöndal, K. Rahu, A. Altraja, P. Viiklepp, and M. Rahu, “Overall and cause-specific mortality among patients with tuberculosis and multidrug-resistant tuberculosis,” *The International Journal of Tuberculosis and Lung Disease*, vol. 17, no. 7, pp. 961–968, Jul. 2013. doi:10.5588/ijtld.12.0946

[58] P. V. Viana et al., “Factors associated with death in patients with tuberculosis in Brazil: Competing risks analysis,” *PLOS ONE*, vol. 15, no. 10, Oct. 2020. doi:10.1371/journal.pone.0240090

[59] A. Zollanvari, A. P. James, and R. Sameni, “A theoretical analysis of the peaking phenomenon in classification,” *Journal of Classification*, vol. 37, no. 2, pp. 421–434, Jul. 2019. doi:10.1007/s00357-019-09327-3