Computational Journal of Mathematical and Statistical Sciences
4(2), 664–696
DOI:10.21608/cjmss.2025.397747.1213
https://cjmss.journals.ekb.eg/

Predicting Clinical Outcomes in Liver Cirrhosis Using Machine Learning and Data Balancing Technique

Nurul Raihen[1,*], **Istiaq Hossain**[2], **Vinodh Chellamuthu**[3]

[1] Department of Mathematics and Statistics, The University of Toledo, Toledo, OH, 43606, USA; nurul.raihen@gmail.com.

[2] Department of Mathematics and Statistics, Stephen F. Austin State University, Nacogdoches, TX, 75965, USA; Md-Istiaq.Hossain@sfasu.edu.

[3] Department of Mathematics, Utah Tech University, Saint George, UT, 84770, USA; Vinodh.Chellamuthu@utahtech.edu.

* **Correspondence:** nraihen@fontbonne.edu

Abstract: Liver cirrhosis is a chronic and life-threatening disease that significantly impacts liver function and overall patient health. Early prediction of clinical outcomes in cirrhotic patients can aid in timely intervention and improved treatment planning. In this study, a dataset containing real-world clinical, biochemical, and demographic data from cirrhosis patients was used to develop predictive models for classifying patient outcomes into three categories: alive, deceased, and liver transplant. A total of fifteen machine learning algorithms were implemented under three scenarios: original dataset with all the rows dropped for missing values, the original dataset with standard data imputation, and a balanced dataset generated through data standardization and the SMOTE oversampling technique. SMOTE was applied to address class imbalance and improve the model's ability to learn from underrepresented outcomes. Experimental results indicate that the Extra Trees classifier achieved the highest predictive performance, with an accuracy of 85.00%, AUC 94.36%, and an F1 score 84.75% on this latter dataset. These findings underscore the importance of data balancing and model selection in improving outcome prediction in liver disease.

Keywords: Liver cirrhosis, SMOTE, Classifier, Machine learning, Logistic regression, Data balancing technique

Mathematics Subject Classification: 62H30, 60G25, 62J12.

Received: 25 June 2025; Revised: 2 October 2025; Accepted: 4 October 2025; Online: 12 October 2025.

1. Introduction

Among the many important organs in the human body, the liver is the largest solid organ situated in the upper right abdomen of our body. The liver plays a crucial role responsible for filtering blood, processing nutrients, and storing glycogen, vitamins, and other substances [1, 2]. Liver cirrhosis (also known as hepatic cirrhosis, chronic liver failure, or chronic hepatic failure, etc.) is a chronic condition of the liver in which the normal functioning tissue is replaced with scar tissue (fibrosis) and regenerative nodules [3]. Cirrhosis is known to be caused by several medical conditions and lifestyle choices such as but not limited to heavy drinking, metabolic dysfunction, extended heron usage, etc., and has also been attributed to obesity, high blood pressure, abnormal levels of cholesterol, type 2 diabetes, and metabolic syndrome [4, 5].

Early detection of liver cirrhosis can help greatly, especially when it comes to avoid irreversible damage to the organ, failure, or liver cancer, and thus saving lives, or at least a treatment plan can be devised as soon as possible [6]. However, early detection is challenging for cirrhosis as symptoms in include fatigue and weakness, loss of appetite, nausea, abdominal pain, jaundice, etc., which many of us may not consider as severe as liver cirrhosis, and an intense form of severity does not show up until the disease progresses to an irreversible stage. Machine learning (ML) is a branch of artificial intelligence which helps us find intrinsic associations between data and does so by systematically applying algorithms [7]. Various types of machine and deep learning models on different forms of datasets are applied to efficiently detect cirrhosis with a certain degree of ”success” metrics in mind (eg., accuracy, AUC, F1-score, etc.). A recent paper used A 1D convolutional neural network (CNN) (a type of artificial neural network) reported with prediction with an AUC of 0.90 (95% CI: $(0.75, 0.99)$) to predict liver cirrhosis from volatile organic compounds (VOC) [8]. Another study used machine learning models such as Support Vector Machine, Decision Tree, and Random Forest, with Random Forest giving the highest accuracy of 97% [9] on an open-access Liver Cirrhosis dataset. Other forms of datasets are also used that include imaging biomarkers, including conventional MRI and computer tomography (CT), but those are costly and time-consuming [10]. An alternative form of imaging biomarkers includes Ultrasound, which is also widely used due to being highly accurate, relatively inexpensive, and noninvasive [11].

Several machine learning studies have been conducted to predict liver disease progression and mortality, although relatively few have specifically utilized the Cirrhosis Patient Survival Prediction dataset from the UCI Machine Learning Repository. A notable application of this dataset was conducted by Cai Selvas Sala, who used algorithms such as K-Nearest Neighbors (KNN), Decision Trees, and Support Vector Machines (SVM) to predict patient survival outcomes [14]. The study provided exploratory insights into the importance of clinical features and offered baseline classification performance that can serve as a springboard for more advanced modeling approaches. In contrast, many other studies have used different liver disease datasets, such as the Indian Liver Patient Dataset (ILPD), which Anand Karna et al. explored using dimensionality reduction techniques like LDA, FA, t-SNE, and UMAP, achieving a maximum accuracy of 98.31% using Random Forest with 10-fold cross-validation [13]. Similarly, Mostafa and Hasan applied Support Vector Machines to classify blood donors and patients with hepatitis, fibrosis, and cirrhosis using imputation and PCA techniques, achieving a high accuracy of 98.23% [14].

Beyond these individual studies, systematic reviews and large-scale comparative analyses have

demonstrated the growing potential of machine learning for cirrhosis mortality prediction. For instance, Malik et al. reviewed several models, including Artificial Neural Networks (ANN), Gradient Boosting, and LightGBM, concluding that machine learning methods generally outperformed traditional scoring systems like MELD and Child-Pugh [15]. Notably, the ANN model by Cucchetti et al. achieved an AUROC of 0.96, while models like Gradient Boosting by Kanwal et al. and LightGBM by Simsek et al. achieved AUROCs of 0.81 and 0.85, respectively. Deep learning approaches have also gained attention, as shown in Guo et al.'s study where a Deep Neural Network (DNN) model reached AUROCs of 0.88, 0.86, and 0.85 for predicting 90−, 180−, and 365−day mortality, outperforming the MELD score [16]. Likewise, Kanwal et al. developed the Cirrhosis Mortality Model (CiMM), achieving an AUROC of 0.78 compared to MELD-Na's 0.67 [17]. Further, Simsek et al.'s LightGBM model consistently outperformed MELD-Na across time points [18], and Tsai et al.'s comparative analysis demonstrated the high performance when XGBoost is used in mortality prediction both in-hospital and ED mortality (area under the receiver operating characteristic curve: 0.866 and 0.861), respectively [19].

The literature review highlights that numerous intelligent systems and machine learning approaches have been proposed to predict health outcomes and support clinical decision-making across various medical domains. These advancements emphasize the growing importance of developing accurate predictive models that leverage clinical, demographic, and biochemical variables. Prior studies have demonstrated that artificial intelligence, particularly machine learning (ML) and deep learning (DL) methods, can significantly enhance the performance of diagnostic and prognostic tools in healthcare. However, relatively few studies have focused specifically on predicting clinical outcomes in cirrhosis datasets [20] and how preprocessing choices-imputation, standardization, and explicit imbalance correction-change performance. Moreover, [21] reports emphasize accuracy but provide limited parameter transparency for oversampling (e.g., SMOTE settings) and limited model interpretation for clinical insight. These gaps matter because class imbalance and preprocessing can strongly bias minority outcomes (notably the transplant class), and clinicians need both reliable discrimination and an explanation of which variables drive predictions.

To address this gap, the present study introduces a robust and data-driven modeling approach aimed at classifying cirrhosis outcomes into three categories: death, censored, and censored due to liver transplantation. A total of fifteen machine learning algorithms were applied across three data configurations, reporting full metrics, and make the imbalance handling explicit (SMOTE with sampling_strategy='auto', k_neighbors=5, fixed random_state). We then connect performance to practice by computing permutation importance and SHAP values for the top model (Extra Trees), thereby linking predictive gains (best AUC = 0.94; accuracy = 0.85) to clinically interpretable features. This design directly targets the above gaps and motivates our contribution as a practical, transparent baseline for tri-class outcome prediction in cirrhosis. This study not only compares the predictive performance of diverse ML models but also highlights the impact of data preprocessing strategies on outcome classification accuracy. The Experimental results revealed that ensemble-based classifiers, particularly the Extra Trees algorithm, outperformed other models across all evaluation metrics. To the best of our knowledge, no prior published study has reported this level of predictive performance on the cirrhosis dataset classification, indicating the potential of this method as a strong candidate for clinical decision support in liver disease management.

Objectives

- Predict tri-class outcomes: death, censored/alive, and censored due to liver transplant-for patients in the Mayo PBC cohort using clinical and biochemical variables.
- Benchmark a broad panel of machine-learning algorithms (15 models) across three preprocessing configurations to identify the best-performing approach for this task.
- Quantify the impact of preprocessing choices-mean imputation, one-hot encoding, standardization, and explicit imbalance correction via SMOTE-with transparent parameters.
- Use rigorous evaluation with 10-fold cross-validation and full metrics (accuracy, AUC, confusion matrices, ROC curves) to compare models.
- Provide model interpretability by computing permutation importance and SHAP values for the top model (Extra Trees) to link predictions to clinically meaningful features.
- Report the key finding that ensemble methods, especially Extra Trees under the standardized+SMOTE pipeline-achieve the strongest performance (accuracy = 0.85; AUC = 0.94), which was not observed in previous studies [20, 21].

The purpose of this study is to analyze clinical and biochemical variables to characterize and predict patient outcomes in detecting liver cirrhosis. One of the major goals of our study is to explore a reliable and more accurate algorithm compared to the already reported results in the community for Cirrhosis Patient Survival. The remainder of this paper is structured as follows. Section 2 outlines the methodology employed in this study, including an overview of the dataset's attributes, the performance evaluation metrics used, and a general description of the machine learning classifiers implemented. In Section 3, we describe the data source in detail, highlighting the numerical and categorical features, the presence of missing values, and the use of visualizations such as heatmaps and distribution plots to summarize the data characteristics in relation to the target variable, "Status". Section 4 presents the data preprocessing steps and the application of various machine learning models, reporting key performance metrics such as accuracy and AUC. This section also includes comparative bar charts to visualize model performance across datasets, focusing on accuracy and training time, as well as confusion matrices and AUC-ROC curves to further illustrate and compare the effectiveness of the best-performing models. Section 5 provides concluding remarks and summarizes the key findings of the study. Finally, Sections 6 and 7 present this study's limitations and offers some outline for future research, respectively.

2. Methodology

One of our central goals is to benchmark a wide range of machine-learning algorithms and identify models that not only outperform existing reports in the cirrhosis survival literature but also provide interpretable insights that are clinically meaningful. By evaluating multiple preprocessing pipelines and explicitly addressing challenges such as class imbalance, we seek to establish a transparent and reproducible baseline for tri-class outcome prediction—death, censored/alive, and liver transplant. This comprehensive evaluation is designed to fill a critical gap in the current literature and to move closer toward clinically deployable decision-support tools for cirrhosis patient survival. Toward this goal, the workflow of our methodology is represented in Figure 1, in which the dataset collection, preprocessing, model training, and performance evaluation are major parts.

As it is clear from this Figure, the first step involves data collection and preparation of the cirrhosis dataset. The dataset used in this study includes real clinical and laboratory data from patients diag-

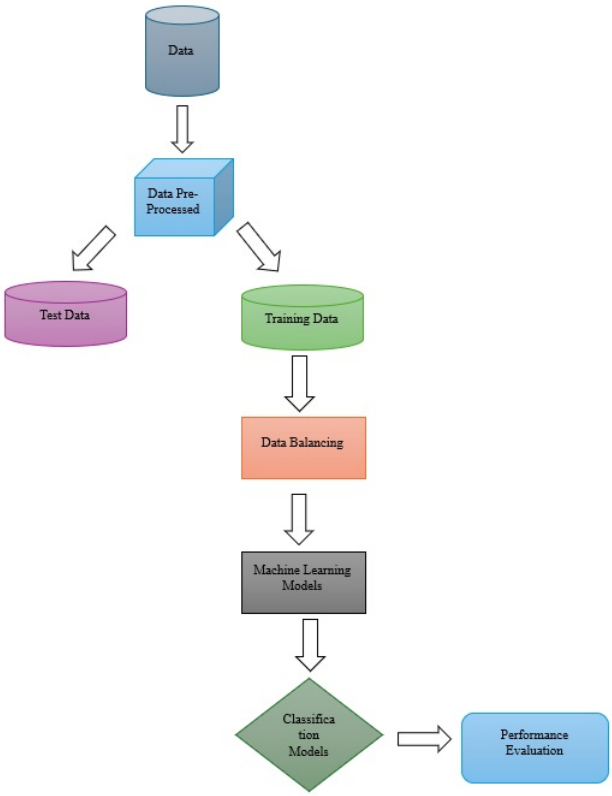


Figure 1. Research Methodology

nosed with liver cirrhosis. The dependent variable, which serves as the output of the model, is the patient's clinical outcome, referred to as Status, and categorized into three classes: alive, deceased, and liver transplant. The objective of this modeling effort is to accurately predict patient outcomes based on various input features, including biochemical markers, hematological parameters, and clinical indicators. Given the diversity and number of input variables, modeling the complex relationships between these variables and the outcome presents a challenge. More detailed information regarding the dataset and the variables used in this study is provided in the following sections. In the following, the data mining process was carried out to identify meaningful and hidden patterns within the cirrhosis dataset. To enable effective analysis, categorical variables such as sex, drug treatment, and clinical indicators (e.g., ascites, hepatomegaly, spiders) were converted into numerical form. The dataset was then split using a 10-fold cross-validation approach, dividing the data into training and testing sets, and the average performance across folds was considered in the evaluation. Since the target variable Status is imbalanced, meaning the number of samples in each class (alive, deceased, and transplant) differs, balancing methods were applied to improve model accuracy and generalization. In machine learning, both undersampling and oversampling techniques are commonly used to address such an imbalance. In this study, oversampling was employed to increase the representation of the minority class. This involved replicating or synthetically generating additional samples from the underrepresented outcome category to ensure the model receives balanced exposure during training. In this study, the SMOTE (Synthetic Minority Over-sampling Technique) method was employed as one of the oversampling strategies to address the imbalance in the Status classes. SMOTE increases the number of minority class samples

by generating synthetic instances rather than duplicating existing ones. This technique operates in the feature space, where artificial samples are created through interpolation between a real minority class sample and its nearest neighbors [22]. Specifically, for each minority class sample, the distance to one or more of its k-nearest neighbors is calculated. A random number between 0 and 1 is then multiplied by the difference between the feature vectors of the sample and its neighbor. The resulting value is added to the original sample to generate a new, synthetic data point. This procedure effectively creates new samples along the line segments joining the minority sample and its neighbors in the multidimensional feature space. By repeating this process across all features and minority samples, SMOTE generates a balanced dataset that improves the model's ability to learn patterns from underrepresented classes. The results obtained from both the original dataset and the SMOTE-balanced dataset were analyzed separately to compare the performance of the applied models. This comparison helped assess the impact of class balancing on model effectiveness. Various machine learning algorithms were implemented and evaluated across both datasets. In this study, model performance was assessed using evaluation metrics including training time, accuracy, and the F1 score. These metrics were selected to capture both the predictive power and computational efficiency of each model, and their values were calculated based on standard performance formulas. The confusion matrix is used to determine these values. Confusion matrices are tables used to describe the performance of a classification model on a set of test data that is already well-known. There are 4 variables that make up the confusion matrix. True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are the four possible outcomes. The format of the confusion matrix is displayed in Table 1.

Table 1. Names and descriptions of dataset attributes.

	Predictive Positive	**Predictive Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

To perform an accurate evaluation of the machine learning classifiers, important measures were collected from the confusion matrix. In addition to the correct classification rate or accuracy, other metrics such as True Positive Rate (TPR), False Positive Rate (FPR), Precision, Recall, F1 score, and ROC area were used to evaluate the machine learning classifiers. These metrics were used to evaluate the classifiers' performance. Our evaluation metrics for the classifiers included:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN},$$
$$\text{Precision} = \frac{TP}{TP+FP},$$
$$\text{Recall} = \frac{TP}{TP+FN},$$
$$\text{F1-score} = \frac{2TP}{2TP+FP+FN},$$

It should be noted that the training time index is not expressed in seconds; instead, it is a unitless and relative measure. The model with the longest training duration is assigned a value of 1, and the training time for all other models is calculated as a ratio relative to this maximum value, resulting in values less than 1 for faster models [23]. The absolute training times on our system (Intel i7 processor, 16 GB RAM) ranged approximately from 0.03 seconds (fastest model) to 1.2 seconds (slowest model). While hardware-dependent, these values demonstrate that all models are computationally feasible for

research and potential clinical applications.

2.1. Description of the main classifiers based on the ML

In this subsection, the main algorithms based on ML used in this study are briefly described.

2.1.1. Extra Trees (ET) Classifier

The Extra Trees (ET) Classifier is an ensemble learning algorithm used for classification tasks, similar to the Random Forest Classifier (RFC). It works by constructing multiple decision trees where each tree is trained on a random subset of features and split points, which introduces additional randomness into the process [24]. This randomization reduces the risk of overfitting and improves the model's generalization to unseen data. Each tree in the ensemble provides a prediction, and the final output is determined by aggregating the predictions from all the trees in the forest. While the training phase of the Extra Trees algorithm is relatively fast, the model's prediction speed may decrease as the number of trees increases, since more trees generally lead to more accurate, yet slower predictions. The ET Classifier is particularly well-suited for high-dimensional data, handling many features effectively, and is commonly used as a benchmark for comparing the performance of other classification algorithms [25].

2.1.2. Random Forest (RF) Classifier

The Random Forest (RF) Classifier is an ensemble learning technique used to address classification problems by building a collection of decision trees. Each tree is trained using a random subset of the features and the data, which ensures that the model doesn't become overly dependent on any particular feature or data point, helping to reduce overfitting. Once trained, each tree in the forest makes a prediction, and the final result is determined by combining the outputs of all the trees, typically through a majority vote. This process of combining multiple trees helps to improve the overall accuracy of the model. While the training phase is relatively fast, the prediction phase can slow down as the number of trees increases, since more trees generally result in a more accurate, but computationally intensive, model. The Random Forest classifier is especially effective when dealing with high-dimensional data and datasets with many features, making it a popular choice for tasks where other classification algorithms may struggle [26].

2.1.3. Gradient Boosting (GB) Classifier

The Gradient Boosting (GB) Classifier is a powerful ensemble learning technique used for classification tasks. Unlike methods like Random Forest, which build multiple trees independently, Gradient Boosting builds decision trees sequentially. In this approach, each subsequent tree attempts to correct the errors made by the previous trees. The algorithm assigns higher weights to the data points that were misclassified by earlier trees, focusing on improving the model's performance for these harder-to-classify instances [27]. After all the trees are built, the final prediction is obtained by combining the results of all the trees, typically by summing their weighted predictions. One of the key advantages of Gradient Boosting is its ability to generate highly accurate models, as it builds on the errors of previous models, gradually improving performance. However, this comes at the cost of longer training times, as the sequential nature of the algorithm requires building one tree at a time. Despite this, Gradient Boosting remains a popular choice for many classification tasks due to its strong performance,

especially in situations where precision is critical [28]. The model also provides flexibility in terms of regularization, which can help to prevent overfitting when applied to complex datasets with many features.

2.1.4. Light Gradient Boosting Machine (LGBM) Classifier

The Light Gradient Boosting Machine (LGBM) Classifier is an optimized version of the traditional Gradient Boosting (GB) algorithm, designed to be more efficient while maintaining high predictive accuracy. LGBM differs from traditional GB methods in its approach to building decision trees. It uses a technique called leaf-wise growth, where the algorithm selects the leaf with the largest reduction in loss to split at each step, rather than following a level-wise growth strategy used in many other gradient boosting models [29]. This strategy allows LGBM to build deeper trees, which can capture more complex patterns in the data. Additionally, LGBM leverages histogram-based algorithms, which enable it to speed up the training process by binning continuous values into discrete bins and reducing the computational complexity. Despite its improved efficiency, LGBM still produces highly accurate models, especially in large datasets with many features. Its flexibility also allows for fine-tuning various hyperparameters to optimize performance and reduce overfitting. While LGBM is faster and more scalable than other boosting methods, it may still suffer from overfitting if not carefully tuned [30]. Overall, LGBM has become a popular choice for classification tasks, particularly in competitive machine learning environments, due to its speed, scalability, and high performance on large datasets.

2.1.5. Extreme gradient boosting (XGB) Classifier

The Extreme Gradient Boosting (XGB) Classifier is a highly efficient and scalable implementation of the gradient boosting framework, designed to improve the performance and speed of traditional gradient boosting methods. XGB builds decision trees sequentially, where each tree attempts to correct the errors made by the previous ones, focusing on instances that were misclassified or had higher residuals. One of the key innovations of XGB is its use of a second-order approximation of the loss function, which incorporates both the first and second derivatives of the loss to guide the optimization process, resulting in faster convergence and better performance on complex datasets. Additionally, XGB incorporates regularization techniques, such as L1 (Lasso) and L2 (Ridge) regularization, which help prevent overfitting and improve generalization by controlling the complexity of the model [31]. The model also utilizes a histogram-based method for faster training on large datasets, reducing memory consumption and computational time. These features, combined with XGB's ability to handle missing data, parallel processing, and distributed computing, have made it a popular choice for classification tasks in machine learning competitions and real-world applications. Despite its high performance, XGB requires careful hyperparameter tuning to avoid overfitting, especially on small datasets.

2.1.6. Quadratic Discriminant Analysis (QDA)

Quadratic Discriminant Analysis (QDA) is a classification technique used to model the relationship between predictor variables and a categorical target variable, especially when the decision boundaries between classes are non-linear. Unlike Linear Discriminant Analysis (LDA), which assumes that the classes share the same covariance matrix, QDA allows each class to have its own covariance matrix, making it more flexible when dealing with complex datasets where class distributions may differ sig-

nificantly in shape [32]. In QDA, the decision boundary between classes is quadratic rather than linear, as it involves a second-order polynomial function of the predictors. This results in a more flexible and accurate model when the assumption of equal covariance matrices in LDA is violated. However, this increased flexibility comes at the cost of needing more data to estimate the separate covariance matrices for each class, which can lead to overfitting in cases with limited data. Despite this, QDA can perform well in situations where the class distributions are well-separated and the assumption of multivariate normality holds. QDA is often used in applications such as face recognition, medical diagnostics, and text classification, where class-specific variances need to be taken into account to improve prediction accuracy [33].

2.1.7. AdaBoost (ADA) Classifier

AdaBoost (Adaptive Boosting) is an ensemble learning technique used for classification tasks that combines the predictions of multiple weak learners, typically decision trees, to create a stronger, more accurate model. The key idea behind AdaBoost is to iteratively train a sequence of classifiers, where each new classifier focuses on correcting the errors made by the previous ones. During each iteration, the algorithm assigns higher weights to the misclassified instances, thereby forcing the next classifier to focus more on the harder-to-classify examples [34]. The final prediction is made by taking a weighted majority vote from all the classifiers, where each classifier's vote is weighted by its accuracy. One of the advantages of AdaBoost is that it is relatively simple to implement and can significantly improve the performance of weak classifiers. However, AdaBoost is sensitive to noisy data and outliers, as incorrect classifications are weighted more heavily during the boosting process, which can lead to overfitting in certain cases. Despite this, AdaBoost is widely used in a variety of applications such as image recognition, text classification, and medical diagnostics due to its effectiveness and efficiency in creating highly accurate models with a relatively low computational cost [35].

2.1.8. Logistic Regression (LR)

Logistic Regression (LR) is a widely used statistical method for binary classification tasks, where the goal is to predict the probability that an instance belongs to one of two classes. Unlike linear regression, which models the relationship between the predictors and the target variable as a straight line, logistic regression applies the logistic function (also known as the sigmoid function) to map the linear combination of input features to a value between 0 and 1. This makes it suitable for predicting probabilities that can be interpreted as the likelihood of an instance belonging to a particular class [36]. The model estimates the parameters using maximum likelihood estimation, where the likelihood function measures how well the model fits the observed data. Logistic Regression is efficient, easy to interpret, and works well for problems with linearly separable classes. However, it may struggle with complex relationships between the features and the target variable, as it assumes a linear decision boundary. Despite this limitation, LR is often used as a benchmark model for classification tasks and can be extended to multi-class problems using techniques such as one-vs-all or softmax regression [37]. It is widely applied in fields such as medicine, finance, and social sciences, where the output of the model represents the probability of an event occurring, such as disease diagnosis or customer churn.

2.1.9. K-Nearest Neighbors (KNN) Classifier

The K-Nearest Neighbors (KNN) Classifier is a simple, yet powerful, algorithm used for both classification and regression tasks. The core idea behind KNN is that the prediction for a given instance is based on the majority class (for classification) or the average value (for regression) of its K closest neighbors in the feature space. To determine the proximity of neighbors, KNN typically uses distance metrics such as Euclidean distance, though other metrics like Manhattan or Minkowski distance can also be used depending on the application [38]. The number of neighbors, K, is a key hyperparameter that determines the model's complexity and sensitivity to noise-small values of K can lead to overfitting, while large values can make the model too simple, potentially underfitting the data. One of the main advantages of KNN is its simplicity and ease of implementation, making it a popular choice for small to medium-sized datasets where the decision boundary between classes is not linear. However, KNN can be computationally expensive during both the training and prediction phases, particularly as the size of the dataset increases, since the algorithm requires computing the distance to every point in the training set for each prediction [39]. KNN is widely used in applications such as image recognition, recommendation systems, and anomaly detection due to its intuitive nature and ability to adapt to complex data structures.

2.1.10. Naive Bayes (NB)

Naive Bayes (NB) is a probabilistic classifier based on Bayes' Theorem, which provides a simple yet effective method for classifying data, especially in high-dimensional spaces. The algorithm operates under the assumption of conditional independence, meaning it assumes that the features used for prediction are independent of each other given the class label. This "naive" assumption significantly simplifies the calculation of the posterior probability for each class, allowing the model to classify new instances by computing the likelihood of each class based on the observed feature values. Despite its simplistic assumption, Naive Bayes often performs surprisingly well, particularly when the independence assumption roughly holds or when the relationships between features are weak. The model works by estimating the class probabilities using the frequency of the features in the training data, and it applies Bayes' Theorem to combine these probabilities to determine the most likely class for a new instance. One of the main advantages of Naive Bayes is its computational efficiency, as it requires only a small amount of data to train and can make predictions very quickly. However, the model's performance can degrade if the independence assumption is not valid, as it overlooks any potential correlations between features. Despite this limitation, Naive Bayes is widely used in text classification, spam filtering, and sentiment analysis, where the independence assumption often holds reasonably well [40].

2.1.11. Decision Tree (DT) Classifier

The Decision Tree (DT) Classifier is a widely used machine learning algorithm that models data using a tree-like structure, where each internal node represents a decision based on one of the input features, and each leaf node corresponds to a class label or a predicted value. The algorithm recursively splits the data at each node based on the feature that provides the best separation, typically using metrics like Gini impurity or entropy to determine the optimal split at each step [41]. This process continues until the tree reaches a predefined stopping criterion, such as a maximum depth or a minimum

number of samples in a leaf node. Decision Trees are easy to interpret, as they clearly show how decisions are made based on feature values, making them particularly useful in applications where model transparency is important. However, Decision Trees are prone to overfitting, especially when they are allowed to grow too deep and capture noise in the data. This can lead to poor generalization on unseen data. To mitigate overfitting, techniques like pruning or ensemble methods, such as Random Forests, are often applied. Despite these limitations, Decision Trees remain popular for classification tasks, including medical diagnoses, customer segmentation, and fraud detection, due to their simplicity, interpretability, and effectiveness when properly tuned [42].

2.1.12. Ridge (RIDGE) Classifier

The Ridge (RIDGE) Classifier is a regularized version of linear regression used for classification tasks. It is based on the concept of linear models but incorporates L2 regularization, also known as ridge regularization, to prevent overfitting, especially when dealing with multicollinearity or when the number of features exceeds the number of data points. The algorithm minimizes the residual sum of squares, as in ordinary least squares regression, but also adds a penalty term proportional to the square of the coefficients of the model. This penalty term helps to shrink the coefficients, reducing their magnitude and preventing the model from fitting the noise in the data [43]. By controlling the size of the coefficients, Ridge regression improves the model's generalization ability, making it less sensitive to small fluctuations in the training data. The regularization strength is controlled by a hyperparameter, typically denoted as alpha, which determines the importance of the penalty term. While Ridge regression works well in situations where features are highly correlated, it is less effective for feature selection because it tends to shrink coefficients rather than set them exactly to zero, unlike Lasso regression. The Ridge Classifier is widely used in scenarios with high-dimensional datasets, such as text classification and gene expression analysis, where regularization is necessary to avoid overfitting and improve model robustness [44].

2.1.13. Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a classification technique used to find a linear combination of features that best separates two or more classes. The goal of LDA is to maximize the distance between the means of different classes while minimizing the variation within each class. This is achieved by calculating the between-class and within-class scatter matrices and finding a projection that maximizes the ratio of these two scatter matrices. The result is a set of discriminant functions that can be used to classify new instances by projecting them onto a lower-dimensional space where the classes are as separable as possible. LDA assumes that the data from each class is normally distributed with the same covariance matrix, which is a key limitation when the classes have different covariance structures. Despite this assumption, LDA is highly effective when the classes are well-separated and the assumptions hold, and it is computationally efficient. One of the main advantages of LDA over other classification methods is that it performs well even with relatively small datasets and is less prone to overfitting. LDA is widely used in applications such as face recognition, medical diagnostics, and speech recognition, where the goal is to reduce the dimensionality of the data while maintaining class separability [45].

2.1.14. SVM- Linear Kernel (SVM)

Support Vector Machine with a Linear Kernel (SVM) is a powerful classification algorithm that finds the optimal hyperplane that separates data points from different classes in a high-dimensional feature space. The primary objective of SVM is to maximize the margin between the closest data points of each class, known as support vectors, and the hyperplane. A larger margin is associated with better generalization and classification performance on unseen data [46]. In the case of a linear kernel, the decision boundary is a straight line (or a hyperplane in higher dimensions), which is suitable when the data is linearly separable. The linear SVM formulation involves solving an optimization problem to identify the hyperplane that maximizes the margin, subject to the constraint that all data points are correctly classified, or are within a tolerance defined by a regularization parameter, C. The linear kernel SVM is often used in problems where the decision boundary between classes is relatively simple, and the data does not require more complex transformations. One of the key advantages of the linear SVM is its efficiency and ability to handle high-dimensional data effectively, such as text classification or gene expression data, where the number of features can be much larger than the number of data points [47]. However, SVM with a linear kernel may not perform well if the data is not linearly separable, in which case, non-linear kernels may be more appropriate.

2.1.15. Dummy (DUMMY) Classifier

The Dummy (DUMMY) Classifier is a simple baseline algorithm used in classification tasks, primarily for comparison with more sophisticated models. Rather than using the input features to make predictions, the Dummy Classifier makes predictions based on predefined strategies that do not involve any learning from the data. The most common strategies include predicting the most frequent class (most-frequent), predicting random class labels (uniform), or using the class distribution in the training set to generate predictions (stratified). These methods provide a baseline against which the performance of more complex classifiers can be evaluated. While the Dummy Classifier does not offer predictive power on its own, it serves as an important tool for assessing whether a more sophisticated model has truly learned meaningful patterns from the data or is merely overfitting noise. The algorithm is fast and computationally inexpensive, but its performance is typically poor when compared to more advanced models. The Dummy Classifier is particularly useful in situations where there is a need to establish a baseline for the accuracy or error rate of a model in imbalanced datasets or simple classification problems [48]. Despite its simplicity, the Dummy Classifier offers valuable insights into model evaluation, especially in the early stages of model development or when benchmarking algorithms.

3. Dataset and research variables

The dataset analyzed in this study originates from a Mayo Clinic investigation into primary biliary cirrhosis (PBC), conducted between 1974 and 1984 [49]. PBC is a chronic liver disease characterized by progressive scarring, often leading to cirrhosis and liver failure [49]. This dataset includes detailed information on 418 patients and focuses on predicting survival outcomes through 17 clinical features. The survival outcomes are classified into three categories: D (Death) for patients who passed away, C (Censored) for those alive at the time of the study's conclusion, and CL (Censored due to Liver Transplantation) for patients who received a liver transplant. The data provides valuable insights into

the interplay of clinical, demographic, and treatment variables, serving as a robust foundation for predictive modeling in healthcare. A full description of the study protocol, including methodology and clinical relevance, is available in the study by Dickson et al. [1989]. The features in the dataset include 17 clinical and demographic variables, such as age, gender, and liver function indicators. Examples of clinical variables include the presence of ascites, hepatomegaly, and vascular spiders, as well as edema status, categorized as "No Edema", "Edema Resolved by Diuretics", or "Persistent Edema Despite Diuretics". Drug treatment information, specifically the administration of D-penicillamine or placebo, is also recorded. Data preprocessing included imputing missing values with the mean for numeric variables and applying one-hot encoding to categorical variables.

Table 2 displays the first five rows of key numerical variables from the cirrhosis dataset, illustrating the variability in liver function and metabolic markers across patients. Bilirubin levels range from 1.1 to 14.5 mg/dL, reflecting differing degrees of liver excretory impairment. Albumin levels (2.54 – 4.14 g/dL) and prothrombin times (10.3 – 12.2 seconds) indicate variation in hepatic synthetic function, while elevated SGOT and Alk_Phos values in some patients suggest active hepatocellular and biliary injury. Cholesterol and triglyceride levels show metabolic diversity, and platelet counts range from 136 to 221 $\times 10^3/\mu$L, with lower counts hinting at possible portal hypertension. Copper variability and extreme Alk_Phos levels further emphasize the heterogeneity of disease severity. Overall, this subset underscores the clinical complexity and multidimensional nature of cirrhosis progression.

3.1. Overview of Numerical and Categorical Features

Table 2. Sample Numerical Data from Cirrhosis Dataset (first 5 rows)

Age	Bilirubin	Cholesterol	Albumin	Copper	Alk_Phos	SGOT	Tryglicerides	Platelets	Prothrombin
21464	14.5	261.0	2.60	156.0	1718.0	137.95	172.0	190.0	12.2
20617	1.1	302.0	4.14	54.0	7394.8	113.52	88.0	221.0	10.6
25594	1.4	176.0	3.48	210.0	516.0	96.10	55.0	151.0	12.0
19994	1.8	244.0	2.54	64.0	6121.8	60.63	92.0	183.0	10.3
13918	3.4	279.0	3.53	143.0	671.0	113.15	72.0	136.0	10.9

Table 3 presents the first five entries of key categorical variables from the cirrhosis dataset, including treatment type, clinical signs, and disease progression indicators. The Drug variable distinguishes between patients who received D-penicillamine (coded as 1) and those who received a placebo (coded as 2). Among these samples, most patients received the active treatment. The Sex variable is coded numerically (1 = male, 2 = female), with males predominating in this subset. Clinical features such as Ascites, Hepatomegaly, Spiders, and Edema are ordinal or binary indicators of disease symptoms. For example, higher values for Edema (up to 3) may denote more severe fluid accumulation. Stage, ranging from 1 to 4, indicates fibrosis severity, and Status represents patient grouping based on clinical outcomes or survival progression. Notably, patients with Stage 4 fibrosis appear more frequently in the CL status classes (e.g., Status 3), suggesting a link between advanced disease and clinical severity. Table 4 summarizes the features with missing values in the cirrhosis dataset, with the highest missingness observed in Triglycerides (136 cases), Cholesterol (134), and Copper (108). Several clinical variables, such as Drug, Ascites, Hepatomegaly, and Spiders, each have 106 missing entries. In contrast, essential features like Status, Sex, and Age are complete. Figure 2 shows these patterns using a missing data heatmap, where yellow marks indicate missing entries. The pattern shows that missing-

Table 3. Sample Categorical Data from Cirrhosis Dataset (first 5 rows)

Drug	Sex	Ascites	Hepatomegaly	Spiders	Edema	Stage	Status
1	1	2	2	2	3	4	3
1	1	1	2	2	1	3	1
1	2	1	1	1	2	4	3
1	1	1	2	2	2	4	3
2	1	1	2	2	1	3	2

ness is variable-specific and not row-wise, suggesting that data loss is systematic and localized. This structure supports the use of feature-level imputation methods, such as multiple imputation or iterative modeling, to address incomplete entries without discarding valuable patient records.

Table 4. Missing Data Summary for Cirrhosis Dataset

Feature	Missing Values
Drug	106
Ascites	106
Hepatomegaly	106
Spiders	106
Cholesterol	134
Copper	108
Alk_Phos	106
SGOT	106
Tryglicerides	136
Platelets	11
Prothrombin	2
Stage	6

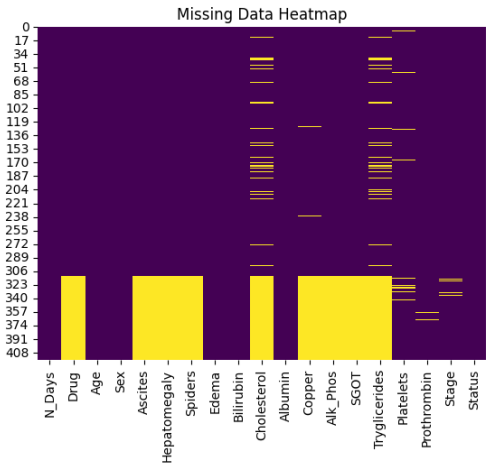


Figure 2. Missing Data Heatmap for Cirrhosis Dataset

Statistical summaries of these features, as well as their distributions, are presented in Table 5, enabling deeper insights into the dataset's structure.

Table 5 also summarizes key numerical features in the cirrhosis dataset, highlighting substantial variability across clinical and biochemical measures. Follow-up time (N_Days) spans from 41 to 4,795 days, and patient age ranges from 9,598 to 28,650 days, reflecting a broad demographic. Markers of liver function, such as Bilirubin and Albumin—show distinct profiles, with bilirubin exhibiting right-skewness (max: 28.0 mg/dL) and albumin values being more tightly distributed. Elevated variability is also evident in Cholesterol, Triglycerides, Copper, and especially Alk_Phos, which ranges up to 13,862 IU/L, suggesting the presence of extreme liver or biliary pathology in some patients. Enzymes like SGOT and hematological parameters like Platelets and Prothrombin show wide ranges as well, reinforcing the clinical heterogeneity of the cohort. These statistics underscore the importance of robust preprocessing and stratified analysis for reliable modeling.

Table 6 provides an overview of the categorical variables included in the cirrhosis dataset. Binary clinical features such as Sex, Ascites, Hepatomegaly, and Spiders are encoded as having two unique

Table 5. Summary Statistics for Numerical Data

Feature	Count	Mean	Std	Min	25%	50%	75%	Max
N_Days	418	1917.78	1104.67	41.00	1092.75	1730.00	2613.50	4795.00
Age	418	18533.35	3815.85	9598.00	15644.50	18628.00	21272.50	28650.00
Bilirubin	418	3.22	4.41	0.30	0.80	1.40	3.40	28.00
Cholesterol	418	369.51	191.07	120.00	273.00	369.51	369.51	1775.00
Albumin	418	3.50	0.42	1.96	3.24	3.53	3.77	4.64
Copper	418	97.65	73.70	4.00	51.25	97.65	100.75	588.00
Alk_Phos	418	1982.66	1848.44	289.00	1016.25	1717.00	1982.66	13862.40
SGOT	418	122.56	48.97	26.35	91.00	122.55	135.75	457.25
Tryglicerides	418	124.70	53.48	33.00	95.00	124.70	127.75	598.00
Platelets	418	257.02	97.02	62.00	190.00	257.02	315.50	721.00
Prothrombin	418	10.73	1.02	9.00	10.00	10.60	11.10	18.00

values, typically representing the presence or absence of the condition. Edema is a three-level ordinal variable indicating the severity of fluid accumulation (None, Slight, Severe), while Drug distinguishes between patients treated with D-penicillamine and those given a placebo. Two key outcome-related variables, Stage and Status, are categorical but carry ordered significance. Stage ranges from 1 to 4, reflecting fibrosis progression, while Status captures clinical outcome groups across three levels.

Table 6. Summary of Categorical Variables

Feature	Unique Values
Sex	2 (Male, Female)
Ascites	2 (Yes, No)
Hepatomegaly	2 (Yes, No)
Spiders	2 (Yes, No)
Edema	3 (None, Slight, Severe)
Drug	2 (D-penicillamine, Placebo)
Stage	4 (1, 2, 3, 4)
Status	3 (1, 2, 3)

The bar chart in Figure 3 shows the distribution of the Status variable, representing clinical outcomes of patients in the cirrhosis dataset. Most patients were classified as Status 1 (death), followed by a substantial number in Status 3 (censored due to liver transplantation). A notably smaller group was categorized as Status 2 (censored). This imbalance highlights the predominance of surviving and deceased patients in the cohort, with relatively few undergoing transplantation. Such a distribution is important to consider when modeling outcomes or applying classification algorithms, as it may influence class representation and prediction accuracy.

The pie chart on the left of Figure 4 illustrates the distribution of patient age, measured in days. The majority of patients (64.4%) fall within the 10,000 to 20,000-day range (approximately 27 – 55 years), while 35.4% are aged 20,000 to 30,000 days (approximately 55 – 82 years). A very small fraction (0.2%) is under 10,000 days old (less than 27 years), indicating that the cohort is predominantly

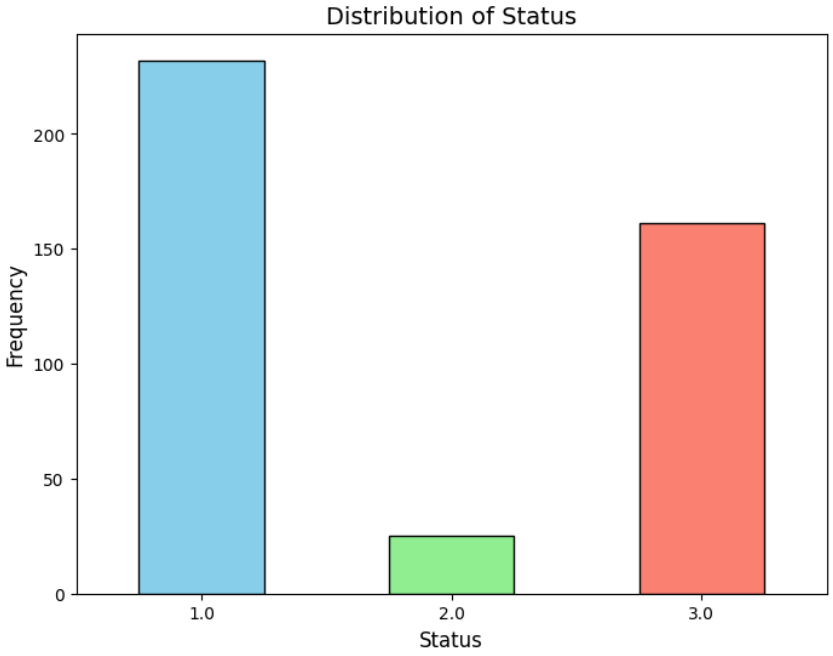


Figure 3. Distribution of Status: 1 represents "death", 2 represents "censored", and 3 represents "censored due to liver transplantation"

middle-aged to older adults, consistent with the typical demographic affected by chronic liver disease. On the right, the gender distribution shows a strong skew toward male patients, who comprise 89.5% of the dataset, while female patients account for only 10.5%. This gender imbalance may reflect the natural epidemiology of cirrhosis in this population or biases in cohort selection. Such distributions are important to acknowledge when interpreting model outcomes or generalizing findings, particularly in studies related to sex-specific risk factors and treatment response.

3.2. Categorical Variable Distributions by Patient Status

Figure 5 illustrates the categorical distributions of Stage, Drug, and Hepatomegaly by Status classes within the cirrhosis dataset. These plots reveal key clinical patterns across disease severity categories. The first bar diagram displays the distribution of Stage by Status, showing that patients classified under Status 3 (denoted in green) are predominantly concentrated in Stage 4, which represents the most advanced disease stage. In contrast, Status 1 patients (blue) are more frequent in Stage 2 and Stage 3, suggesting that lower status values correlate with less advanced fibrosis. Status 2 (orange), representing an intermediate group, appears relatively balanced across all stages but in noticeably smaller proportions. This pattern implies a possible progressive transition of clinical status across fibrosis stages. The second bar diagram visualizes the Drug distribution by Status, comparing patients who received D-penicillamine (Drug = 1) and placebo (Drug = 2). Notably, a higher number of Status 1 patients received D-penicillamine compared to Status 3, which might suggest early therapeutic intervention in less severe cases. Conversely, the placebo group has a relatively even spread between Status 1 and Status 3, suggesting potential non-response or control group assignment in more advanced cases. Status 2 patients again represent a minor proportion across both treatment groups. The third

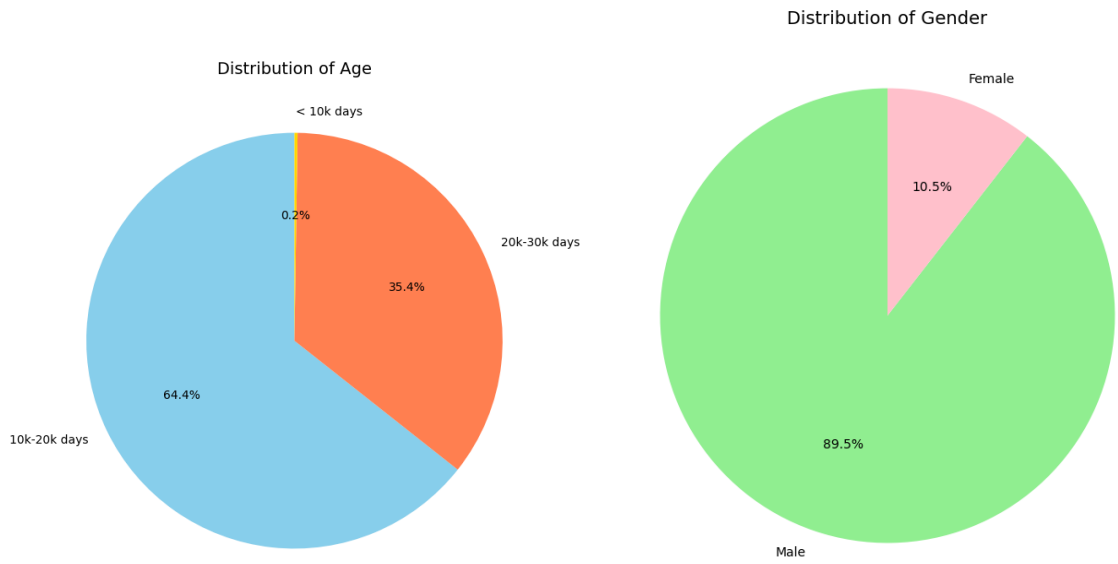


Figure 4. Distribution of Age and Gender

bar diagram illustrates the distribution of Hepatomegaly by Status, showing that Hepatomegaly = 2 (presence of liver enlargement) is dominant in both Status 1 and Status 3 groups. This result may indicate that hepatomegaly alone does not sufficiently discriminate between clinical statuses, as it is prevalent across both early and advanced cases. However, the lower frequency of Hepatomegaly = 1 (absence) in Status 3 supports its association with more advanced disease. Together, these distributions highlight clinically relevant distinctions and overlaps among cirrhosis-related variables across patient status categories.

3.3. Analysis of correlations between Numerical Features

Figure 6 presents a Pearson correlation heatmap that quantitatively illustrates the linear relationships between numerical variables within the cirrhosis dataset. Pearson correlation coefficients range from -1 to $+1$, where: $+1$ indicates a perfect positive linear relationship, -1 indicates a perfect negative linear relationship, and 0 indicates no linear relationship. Only numerical features were included to ensure the statistical validity of the correlation matrix, which reveals biologically and clinically consistent patterns among liver function markers. Notably, serum bilirubin demonstrated moderate positive correlations with both prothrombin time ($r = 0.43$) and SGOT ($r = 0.39$), suggesting that as hepatocellular injury progresses, reflected by elevated bilirubin, there is a corresponding delay in clotting efficiency and an increase in hepatocellular enzyme release. Bilirubin also showed a positive correlation with serum copper levels ($r = 0.41$), likely reflecting impaired biliary excretion of copper in cholestatic states. Conversely, bilirubin exhibited a negative correlation with albumin ($r = -0.31$), supporting the association between worsening liver function and decreased hepatic protein synthesis. In parallel, albumin and prothrombin time were also negatively correlated ($r = -0.20$), reinforcing the link between declining synthetic function and extended clotting times in cirrhotic patients. Platelet count showed an inverse correlation with bilirubin ($r = -0.26$), likely due to hypersplenism secondary to portal hypertension, a hallmark of advanced liver disease. A mild correlation was observed

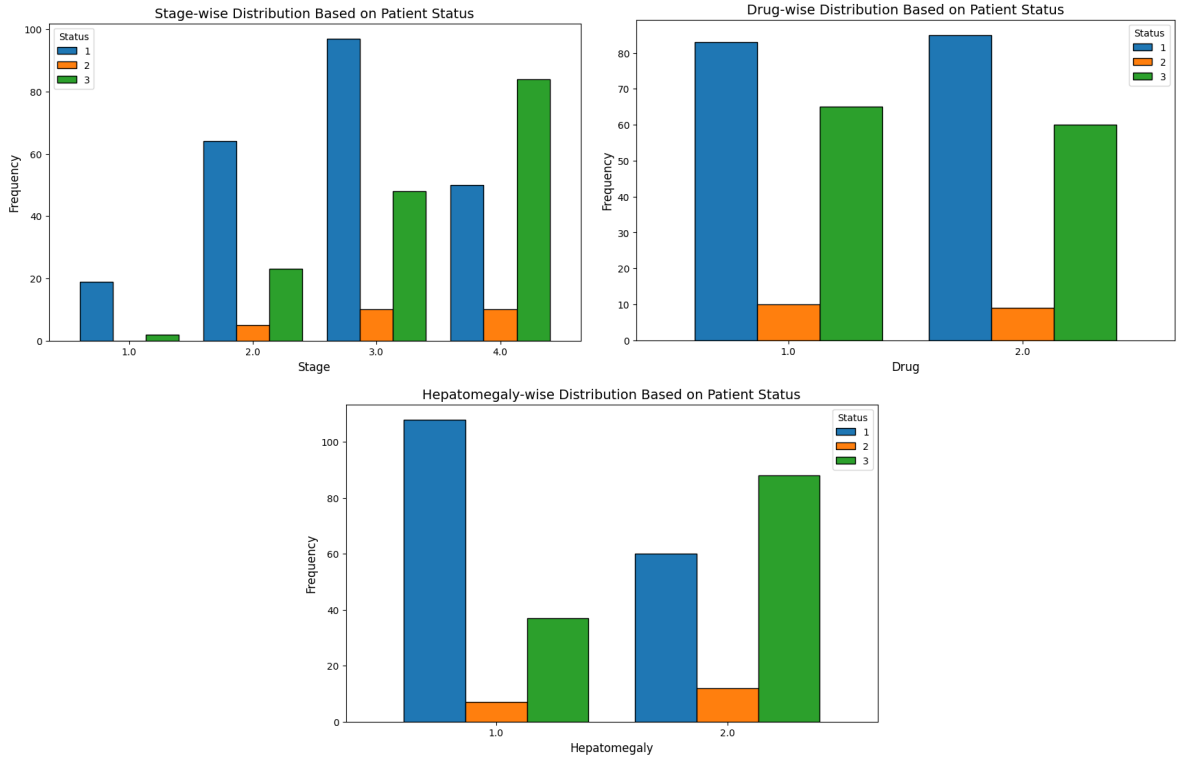


Figure 5. Distribution of Stage, Drug, and Hepatomegaly by Status

between triglycerides and SGOT ($r = 0.27$), potentially indicative of metabolic alterations accompanying hepatic inflammation or steatosis. Other variables, such as alkaline phosphatase, cholesterol, and triglycerides, exhibited relatively weak or negligible correlations with most features, suggesting these may vary independently or be influenced by extrinsic factors such as diet, medications, or comorbid metabolic conditions. Overall, the heatmap reveals clinically coherent interdependencies among liver injury markers, synthetic function indicators, and hematological parameters. These insights not only validate known psychophysiological mechanisms in cirrhosis but also provide a foundation for multivariate modeling and biomarker-driven disease stratification.

4. Results and Discussion

This section is dedicated to reporting the results of our investigation on correctly classifying and displaying various metrics for our liver cirrhosis dataset, comparing model accuracies and training times of the ten best models, and then comparing model performance on correctly classifying only the best model in each case. We are going to work on the following three different preprocessed datasets.

1. Simple pre-processing: We drop every row of our dataset that has at least one missing value. We apply no data balancing technique in this case. Even though we used the PyCaret library in Python for automated machine learning tasks, including data preprocessing, imputation, model selection, and performance comparison, since our dataset already has rows with missing values that were removed before using PyCaret, the data imputation that is done automatically was not applied in this case.

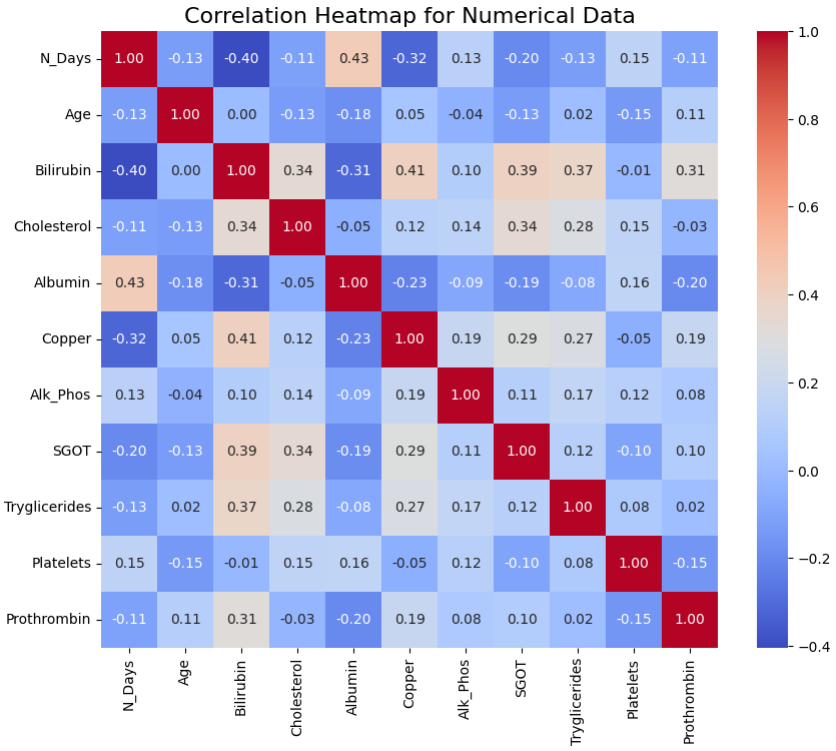


Figure 6. Correlation of Numerical Features

2. Data imputations through PyCaret: In this case, we consider the full dataset (that initially contains missing values in it) and impute the numerical features by the column mean (average) and categorical features by column mode (the most frequent value of a column) through PyCaret's automated data imputation process. In this case, we did not apply any data balancing technique.
3. Data standardization and SMOTE balancing: We performed data standardization by using the scikit-learn 'StandardScaler' feature to standardize each sample by removing the mean and scaling to unit variance of each feature. We also considered balancing data by performing Synthetic Minority Oversampling Technique (SMOTE), a statistical method to balance a dataset by generating new minority class samples. The next paragraph gives a summary of what a typical SMOTE application looks like.

Imbalanced classification refers to the task of developing predictive models on datasets where one class (typically the class of interest) is significantly underrepresented compared to others. This imbalance poses a challenge because many standard machine learning algorithms tend to favor the majority class, resulting in poor predictive performance on the minority class, which is often the class of greatest importance (e.g., in fraud detection or medical diagnosis). A common strategy to address this issue is to oversample the minority class. While the simplest approach is to duplicate existing minority class examples, such duplication does not introduce any new information and may lead to overfitting. A more effective alternative is the Synthetic Minority Oversampling Technique (SMOTE), which generates new synthetic samples by interpolating between existing examples in the minority class. SMOTE acts as a form of data augmentation, helping models better learn the decision boundary and improving generalization to unseen minority class instances. For example, suppose our dataset contains two fea-

tures and we have only two samples in the minority class: $(2, 3)$ and $(4, 5)$. SMOTE would create a new synthetic point by interpolating between them, for instance, a point like $(3.2, 4.1)$, which lies along the line segment connecting the two original points. By repeating this process across the dataset, SMOTE increases the representation of the minority class in a more meaningful and diverse way. This helps machine learning models learn a more balanced decision boundary, leading to improved generalization and performance in the underrepresented class.

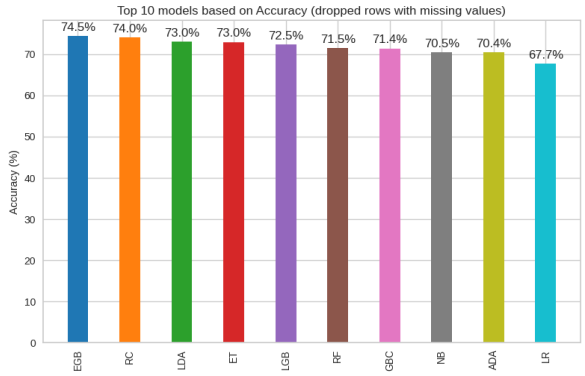
The next few subsections summarize our findings of various model performances based on different metrics. The accuracies and training times are our main comparison metrics for calling these models the best model, where the datasets are from the three different pipelines mentioned above.

4.1. Performance with Simple Preprocessing

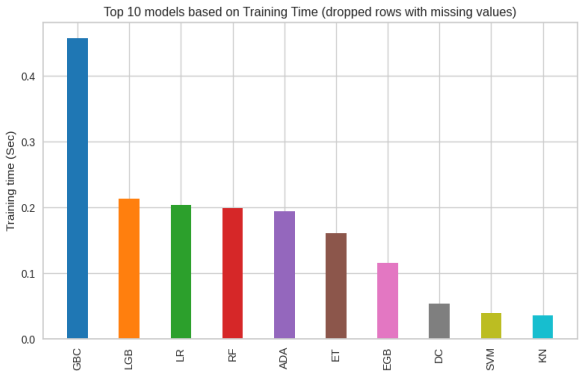
The best-performing model for the simple preprocessed data (Table 7 and Figure 7) was Extreme Gradient Boosting (EGB), achieving the highest accuracy (0.7447) and AUC (0.8673). It also demonstrated strong performance across other metrics including F1 score (0.7308), Kappa (0.5143), and MCC (0.5244). Ridge Classifier (RC) and Linear Discriminant Analysis (LDA) also performed competitively in terms of accuracy (0.7403 and 0.7303, respectively), and were among the most computationally efficient, with training times under 0.03 seconds. Extra Trees (ET) and LightGBM (LGB) showed robust AUC values (0.8406 and 0.8334), indicating good classification potential. On the lower end, models like Decision Tree (DT), K-Nearest Neighbors (KN), and Quadratic Discriminant Analysis (QDA) underperformed across most metrics, while the Dummy Classifier (DC) showed no learning capability, as expected. Training times varied significantly, with some models (e.g., Logistic Regression (LR) and Gradient Boosting Classifier (GBC)) being notably slower.

Table 7. Performance of Various Models with Simple Preprocessing

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
EGB	0.7447	0.8673	0.7447	0.7337	0.7308	0.5143	0.5244	0.158
RC	0.7403	0.4338	0.7403	0.6983	0.7128	0.4932	0.5078	0.030
LDA	0.7303	0.8174	0.7303	0.7139	0.7136	0.4868	0.5006	0.033
ET	0.7297	0.8406	0.7297	0.6939	0.7042	0.4785	0.4931	0.185
LGB	0.7245	0.8334	0.7245	0.6876	0.7028	0.4745	0.4813	0.339
GBC	0.7195	0.8466	0.7195	0.6986	0.6988	0.4679	0.4857	0.367
RF	0.7145	0.8524	0.7145	0.6738	0.6898	0.4498	0.4595	0.239
NB	0.7047	0.8065	0.7047	0.7086	0.6948	0.4456	0.4570	0.030
ADA	0.7042	0.7776	0.7042	0.7067	0.6879	0.4519	0.4776	0.116
LR	0.6876	0.7709	0.6876	0.6575	0.6673	0.4057	0.4157	1.231
SVM	0.6679	0.4721	0.6679	0.6718	0.6257	0.3532	0.3979	0.038
DT	0.6247	0.6776	0.6247	0.6545	0.6276	0.3353	0.3445	0.032
KN	0.6008	0.6998	0.6008	0.5774	0.5771	0.2254	0.2401	0.047
QDA	0.5968	0.6951	0.5968	0.6116	0.5535	0.2408	0.2723	0.032
DC	0.5339	0.5000	0.5339	0.2858	0.3721	0.0000	0.0000	0.028



(a) Top 10 models based on Accuracy (dropped rows with missing values)



(b) Top 10 models based on Training Time (dropped rows with missing values

Figure 7. Figures (a) and (b) display the comparison bar charts for accuracy scores and training time for the top 10 models in descending order.

4.2. Performance with Data Imputation Through PyCaret Preprocessing

Table 8 and Figure 8 presents the performance of various classification models after applying data imputation prior to using PyCaret for preprocessing and model training. Among all models evaluated, the Gradient Boosting Classifier (GBC) achieved the highest accuracy of 0.7907 and the top F1-score of 0.7729, indicating superior predictive performance. Although the AUC value for GBC was recorded as zero, likely due to a reporting or calculation issue, it demonstrated the best overall classification effectiveness based on other metrics, including precision, recall, Kappa, and MCC.

The Extra Gradient Boosting (EGB) and Random Forest (RF) models also performed strongly, each with an accuracy of approximately 0.763 and competitive F1-scores and MCC values. Conversely, models like the Dummy Classifier (DC), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) underperformed across most metrics, suggesting limited suitability for the dataset under the current preprocessing approach.

4.3. Performance with Data Standardization and SMOTE

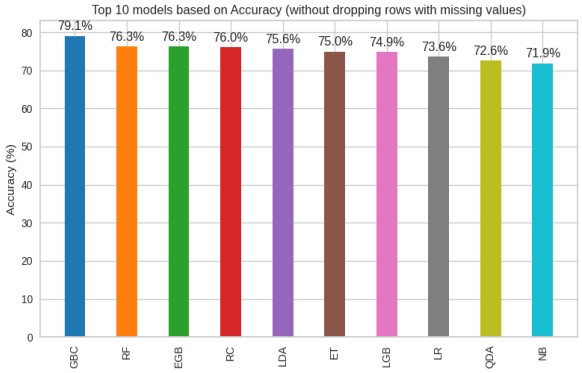
Table 9 and Figure 9 summarize the performance metrics of classification models after combining data imputation and manual feature engineering before applying PyCaret. The Extra Trees Classifier (ET) emerged as the best-performing model with the highest accuracy of 0.8500, the top AUC score of 0.9436, and strong F1, Kappa, and MCC scores, indicating robust generalization and balanced performance. Other strong performers include the Random Forest (RF), Gradient Boosting Classifier (GBC), and LightGBM (LGB), all showing high accuracy (above 0.80), competitive F1-scores, and consistent agreement across all evaluation metrics.

4.4. Overall comparison on model performance

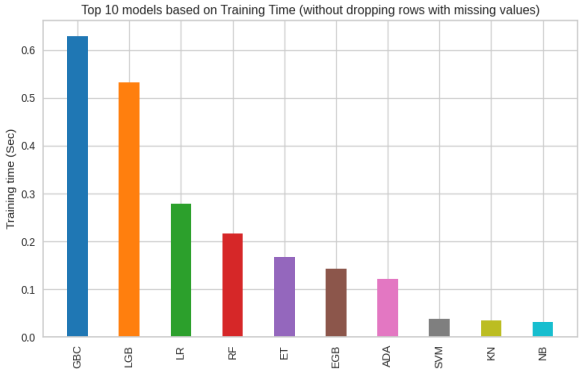
Across all three experimental setups, simple preprocessing (SP), data imputation through PyCaret (DITP), and standardization with SMOTE (SS), the performance of classification models improved noticeably when appropriate preprocessing techniques were applied, as shown in Figure 10. In the SP setup, Extreme Gradient Boosting (EGB) achieved the highest accuracy (0.7447), while Ridge

Table 8. Performance of Various Models with Data Imputation prior to PyCaret Preprocessing

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
GBC	0.7839	0.8680	0.7839	0.7520	0.7645	0.5780	0.5843	0.399
EGB	0.7701	0.8693	0.7701	0.7667	0.7623	0.5620	0.5690	0.144
RF	0.7669	0.8663	0.7669	0.7264	0.7401	0.5400	0.5520	0.222
ET	0.7634	0.8456	0.7634	0.7326	0.7415	0.5360	0.5470	0.175
RC	0.7531	0.7921	0.7531	0.7171	0.7288	0.5104	0.5222	0.027
LDA	0.7494	0.8298	0.7494	0.7389	0.7366	0.5127	0.5217	0.035
LGB	0.7493	0.8575	0.7493	0.7375	0.7386	0.5179	0.5235	1.008
LR	0.7425	0.7987	0.7425	0.7274	0.7281	0.4991	0.5083	0.864
QDA	0.7324	0.7884	0.7324	0.6933	0.7065	0.4681	0.4804	0.031
ADA	0.7226	0.7975	0.7226	0.7253	0.7178	0.4841	0.4901	0.114
NB	0.7187	0.8254	0.7187	0.7178	0.7068	0.4555	0.4676	0.029
DT	0.6838	0.7147	0.6838	0.7001	0.6852	0.4178	0.4235	0.032
KN	0.6605	0.6981	0.6605	0.6368	0.6274	0.3101	0.3314	0.039
SVM	0.6268	0.7664	0.6268	0.5944	0.5663	0.2475	0.2817	0.038
DC	0.5547	0.5000	0.5547	0.3077	0.3959	0.0000	0.0000	0.025



(a) Top 10 models based on Accuracy (without dropping rows with missing values



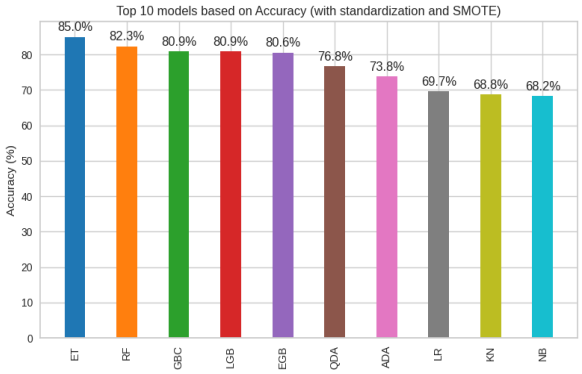
(b) Top 10 models based on Training Time (without dropping rows with missing values

Figure 8. Figures (a) and (b) display the comparison accuracy scores and training time bar charts with data imputations before PyCaret Preprocessing for the top 10 models in descending order.

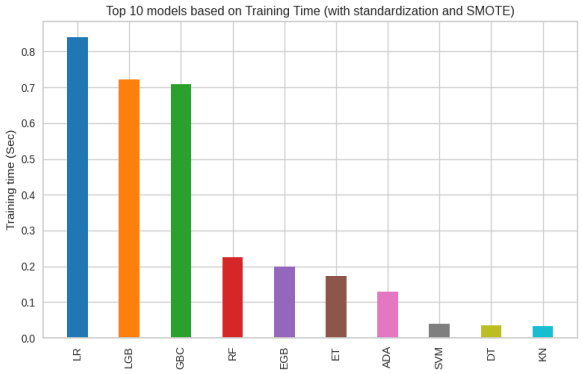
Classifier (RC) and Linear Discriminant Analysis (LDA) offered strong results with minimal training time ($\approx$ 0.03 seconds each). After applying data imputation, Gradient Boosting Classifier (GBC) slightly outperformed others in accuracy (0.7839), while EGB, Random Forest (RF), and LightGBM (LGB) also maintained competitive performance. In this case, RC and LDA again had the fastest training times ($\approx$ 0.03 seconds), highlighting their computational efficiency. In the final setup with additional preprocessing (SS), Extra Trees (ET) achieved the highest accuracy (0.85), followed by RF and GBC, reinforcing the strength of ensemble-based approaches. Training times in the SS setup were

Table 9. Model Performance after Imputation and Manual Feature Engineering before Applying PyCaret

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
ET	0.8500	0.9436	0.8500	0.8535	0.8475	0.7748	0.7785	0.169
RF	0.8235	0.9321	0.8235	0.8286	0.8215	0.7351	0.7388	0.217
GBC	0.8088	0.9234	0.8088	0.8197	0.8037	0.7126	0.7204	0.482
LGB	0.8088	0.9311	0.8088	0.8109	0.8077	0.7131	0.7149	0.952
EGB	0.8059	0.9295	0.8059	0.8139	0.8035	0.7085	0.7132	0.193
QDA	0.7676	0.8968	0.7676	0.7945	0.7660	0.6518	0.6628	0.028
ADA	0.7382	0.8592	0.7382	0.7517	0.7347	0.6067	0.6142	0.112
LR	0.6971	0.8304	0.6971	0.7134	0.6915	0.5444	0.5527	0.033
KN	0.6882	0.8729	0.6882	0.7202	0.6714	0.5315	0.5589	0.031
NB	0.6824	0.8462	0.6824	0.7067	0.6749	0.5233	0.5354	0.028
DT	0.6765	0.7565	0.6765	0.6813	0.6711	0.5139	0.5196	0.030
RC	0.6735	0.7854	0.6735	0.6915	0.6675	0.5091	0.5192	0.029
LDA	0.6676	0.8271	0.6676	0.6864	0.6605	0.5002	0.5115	0.029
SVM	0.5941	0.7502	0.5941	0.6193	0.5912	0.3908	0.3993	0.033
DC	0.3235	0.5000	0.3235	0.1047	0.1582	0.0000	0.0000	0.025



(a) Top 10 models based on Accuracy (with standardization and SMOTE)



(b) Top 10 models based on Training Time (with standardization and SMOTE)

Figure 9. Figures (a) and (b) display the comparison of accuracies and training time (in descending order) bar charts of the top 10 models with data standardization and the SMOTE technique.

still lowest for RC and LDA, but ensemble methods such as EGB and LGB required substantially longer times (≈0.95–1.0 seconds).

Overall, these results demonstrate that preprocessing steps, especially data imputation and feature balancing, significantly enhance model performance, and that ensemble learning methods (ET, RF, GBC, EGB, LGB) consistently deliver the strongest accuracies, while linear methods (RC, LDA) remain highly efficient in terms of training time.

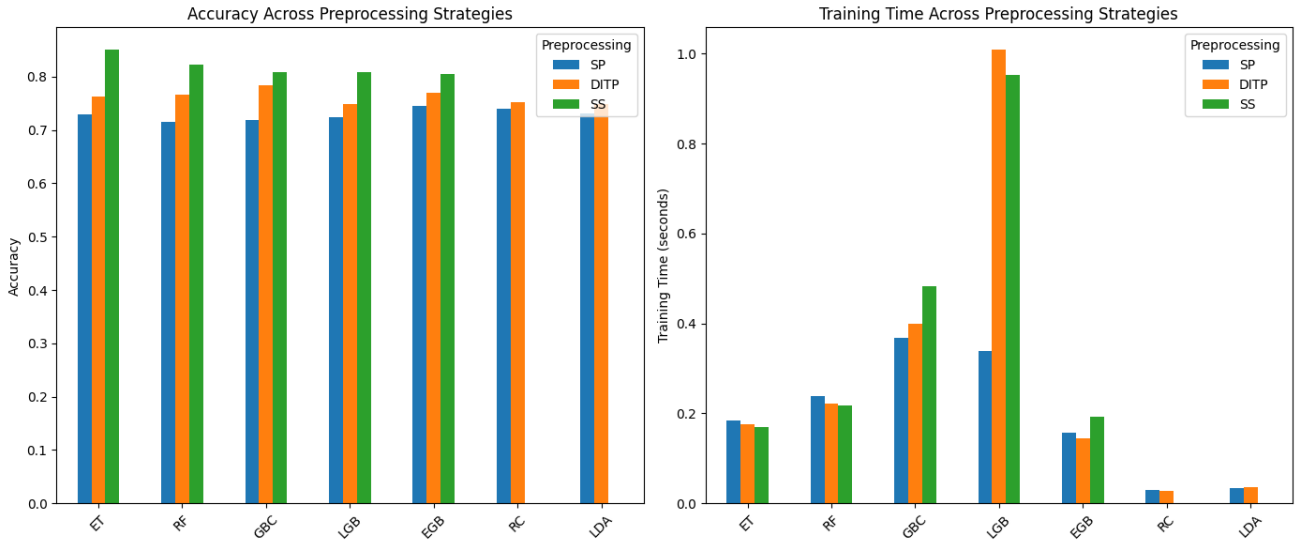


Figure 10. Comparison of model accuracies and training times across preprocessing pipelines.

4.5. Class distribution before and after SMOTE

In our case, because all features are numeric after preprocessing and imputation, plain SMOTE is sufficient to generate synthetic samples without introducing invalid or inconsistent values. This simplifies the workflow while ensuring that the resampled dataset remains suitable for downstream modeling in PyCaret. The SMOTE parameter settings, and the class distributions before and after oversampling are shown in Tables 10, 11, and 12, respectively.

Table 10. SMOTE Parameter Settings

Parameter	**Description / Value**
sampling_strategy	'auto': Raises all minority classes up to the majority class count.
k_neighbors	5: Each synthetic sample is generated using 5 nearest neighbors.
random_state	42: Ensures reproducibility of results.
n_jobs	None: Single-threaded execution.

Table 11. Class distribution BEFORE oversampling (train)

Status	Count	Percentage
1	232	55.50%
2	25	5.98%
3	161	38.52%

Table 12. Class distribution AFTER oversampling (train)

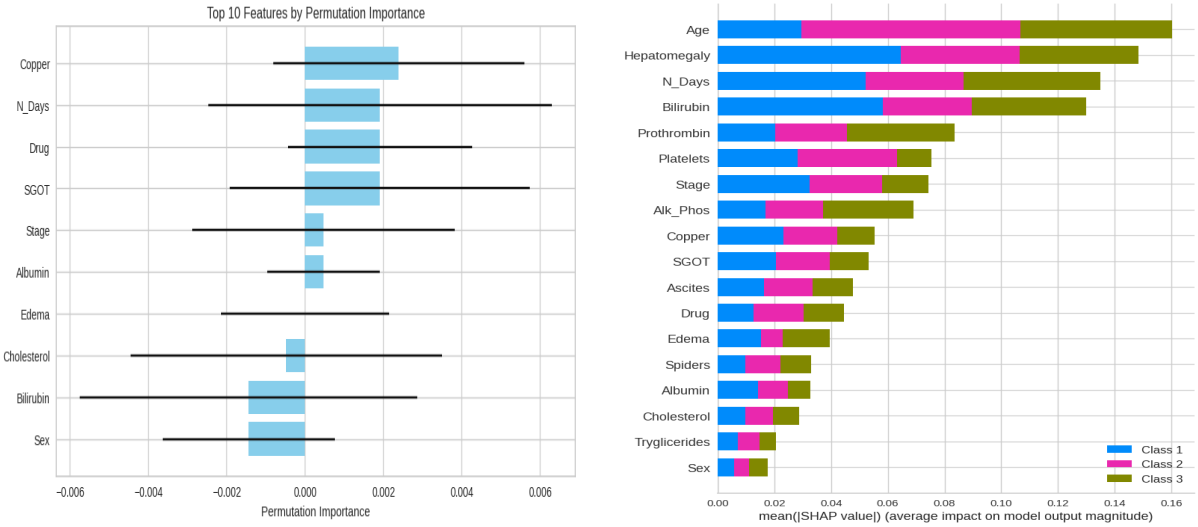
Status	Count	Percentage
1	232	33.33%
2	232	33.33%
3	232	33.33%

4.6. *Permutation Importance and SHAP values*

Understanding which features most significantly influence a model's predictions is crucial for interpretability and gaining insights into the underlying data relationships. To this end, feature importance analysis was conducted using the trained Extra Trees classifier for predicting cirrhosis status (Classes 1, 2, and 3). This analysis employed two complementary methods: Permutation Importance and SHAP (Shapley Additive exPlanations) values. The model was trained on data that had undergone imputation of missing values, standardization, and SMOTE oversampling to address class imbalance.

The Permutation Importance plot (Figure 11a) quantifies a feature's importance by measuring the decrease in model performance when its values are randomly shuffled. This analysis indicates that 'Copper', 'N_Days', 'Drug', and 'SGOT'are among the most important features for this model. On the other hand, the SHAP summary plots (Figure 11b) provide a deeper understanding by illustrating the impact of each feature on the model's output for individual instances, aggregated to show overall feature influence across the dataset. This plot emphasizes the significant influence of features such as 'Age', 'Hepatomegaly', 'N_Days', 'Bilirubin', and 'Prothrombin'on the model's predictions and reveals how different feature values contribute to predicting each class.

Together, these Permutation Importance and SHAP analyses offer a comprehensive view of the key features driving the Extra Trees model's predictions for cirrhosis status, providing valuable insights into the most relevant clinical and demographic factors.



(a) Permutation Importance of Top Features **(b)** Shap Bar Plot of Feature Importance

Figure 11. Figures (a) and (b) summarize feature importance using Permutation Importance and SHAP values for the Extra Trees model predicting Cirrhosis Status (Classes 1, 2, and 3).

4.7. *Confusion matrices and Receiver operating characteristic (ROC) curves for best performing models*

In this section, we compare the best-performing model (based on the highest accuracies) in each of three different scenarios: simple data pre-processing, data imputations through PyCaret, and with data standardization and the SMOTE technique applied to the dataset. The medical test results, often

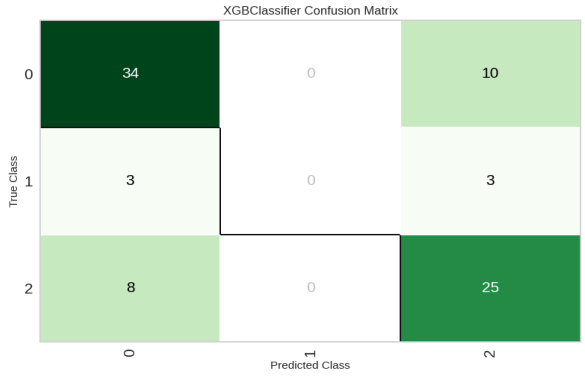
obtained as continuous values may require a process of conversion and better interpretation to determine the presence or absence of a disease. One of the ways this can be done is by using the receiver operating characteristic (ROC) curve [50]. In the context of liver cirrhosis detection using machine learning models, the Receiver Operating Characteristic (ROC) curve serves as a vital diagnostic tool for evaluating the classification performance of the models. The ROC curve is generated by plotting the True Positive Rate (sensitivity) against the False Positive Rate (1 - specificity) across a range of classification thresholds. This curve visually represents the model's ability to distinguish between patients with and without liver cirrhosis. A key quantitative measure derived from the ROC curve is the Area Under the Curve (AUC), commonly referred to as AUC-ROC. The AUC provides a single scalar value summarizing the model's discriminative power across all possible thresholds. An AUC of 0.5 suggests that the model performs no better than random guessing, while an AUC of 1.0 indicates perfect discrimination between cirrhosis and non-cirrhosis cases. In clinical applications such as cirrhosis detection, a higher AUC signifies a model's greater effectiveness in correctly identifying diseased versus non-diseased individuals, which is crucial for early diagnosis and timely intervention. Given the potential class imbalance and the high stakes of misclassifications in medical diagnosis, the AUC-ROC is particularly appropriate for evaluating model performance. It enables robust comparison between different classifiers, independent of the choice of threshold or underlying class distribution, thus offering a reliable criterion for selecting the most clinically useful model. Figure 12 illustrates the performance of the XGBoost classifier before and after hyperparameter tuning. The tuned model (Figures (c) and (d)) demonstrates clearer class separation in the confusion matrix and improved ROC curves compared to the untuned version, indicating more balanced predictions and higher discriminative ability across all outcome classes.

The Gradient Boosting classifier shows clear improvements after tuning, with fewer misclassifications and higher class-specific AUC values. As illustrated in Figure 13, the ROC curves demonstrate stronger discriminative power across all classes, confirming that hyperparameter tuning enhances predictive performance. These results highlight the value of systematic tuning for optimizing clinical prediction models.

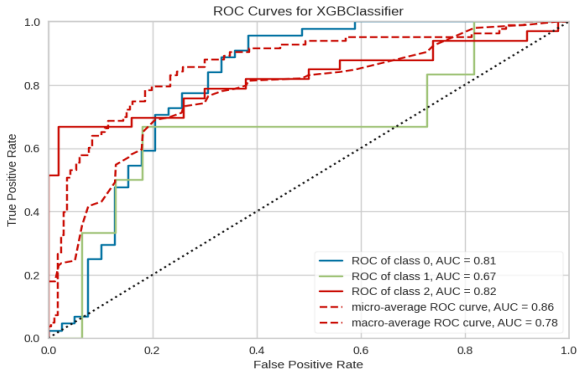
The Extra Trees classifier demonstrates strong predictive performance after tuning, with high accuracy and balanced classification across all three outcomes. We report key hyperparameters of the best-performing Extra Trees model, including n_estimators, max_depth, max_features, bootstrap, min_samples_split/leaf, and random state for transparency and reproducibility in Table 13. In addition, as shown in Figure 14, both the confusion matrix and the ROC curve highlight consistent improvements compared to the untuned case, achieving AUC values above 0.93. These results confirm that ensemble methods, particularly Extra Trees, provide reliable and clinically meaningful discrimination.

Table 13. Key Hyperparameters of the Best Extra Trees Model (with SMOTE + Standardization)

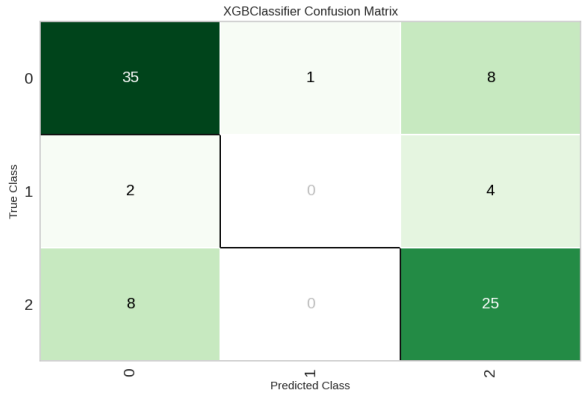
Parameter	**Value**	**Parameter**	**Value**
n_estimators	100	max_depth	None
max_features	'sqrt'	bootstrap	False
min_samples_split	2	min_samples_leaf	1
random_state	42	criterion	'gini'



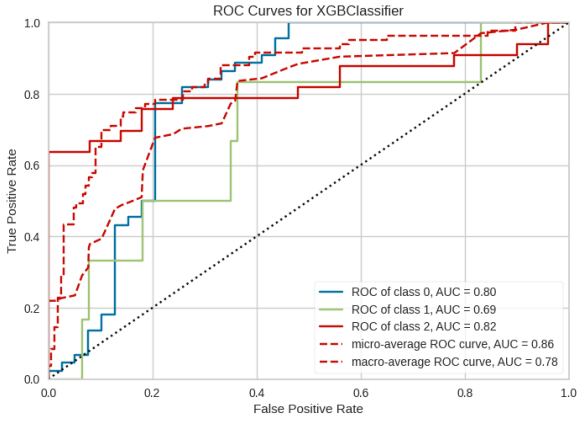
(a) Confusion Matrix (untuned)



(b) ROC curve (untuned)



(c) Confusion Matrix (tuned)

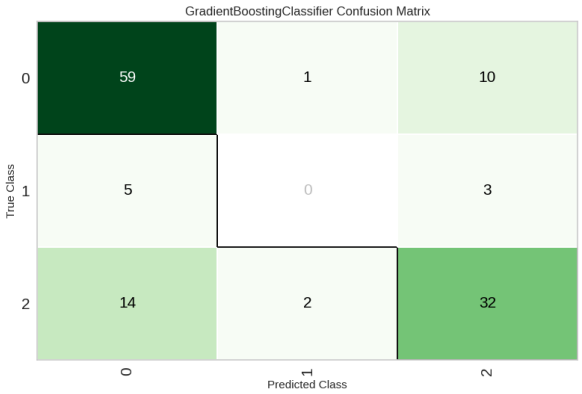


(d) ROC curve (tuned)

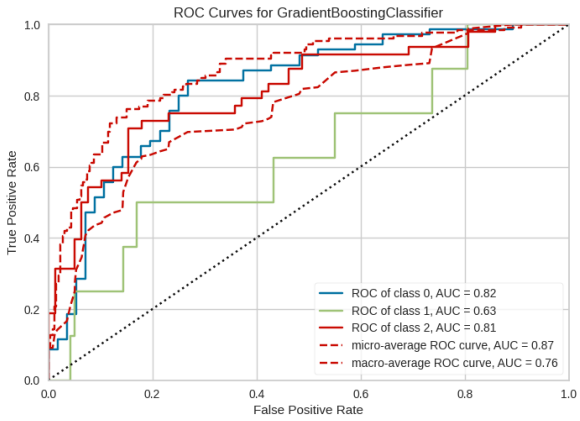
Figure 12. Figures (a) and (c) present the confusion matrices for the untuned and tuned models, respectively, while Figures (b) and (d) illustrate the corresponding ROC curves.

5. Conclusion

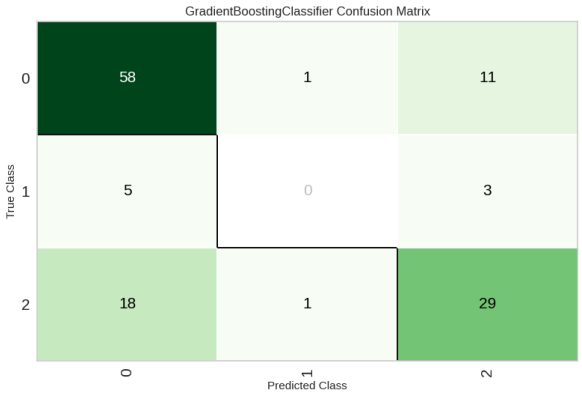
Liver cirrhosis is a progressive and potentially fatal disease that impairs the liver's ability to perform vital physiological functions. Accurate prediction of patient outcomes in cirrhosis is essential for effective clinical decision-making and disease management. This study aims to develop predictive models that classify patients based on clinical outcomes, categorized as death, censored, and censored due to liver transplantation, using a dataset derived from real patient records. The dataset includes both numerical and categorical variables, such as liver enzyme levels, blood markers, and clinical signs like ascites and hepatomegaly. A total of fifteen machine learning models were evaluated using three approaches: the original dataset with rows containing missing values removed, one on the same dataset after standard data imputation, and the other on a balanced version of the data created using the SMOTE method. SMOTE generates synthetic samples of the minority classes to address the skewed class distribution. The Experimental results showed that the ensemble-based algorithm, the ET classifier, yielded the highest classification accuracy when trained on the SMOTE-balanced dataset. These findings highlight that both the choice of machine learning model and the preprocessing approach substantially influence the predictive power in cirrhosis outcome classification, revealing distinct outcomes



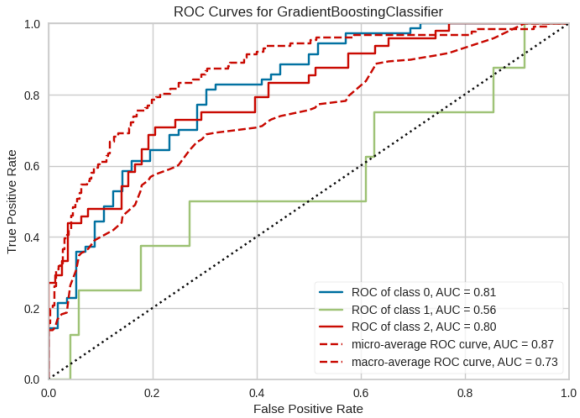
(a) Confusion Matrix (untuned)



(b) ROC curve (untuned)

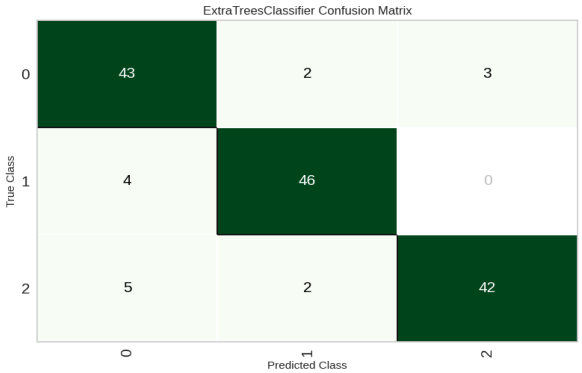


(c) Confusion Matrix (tuned)

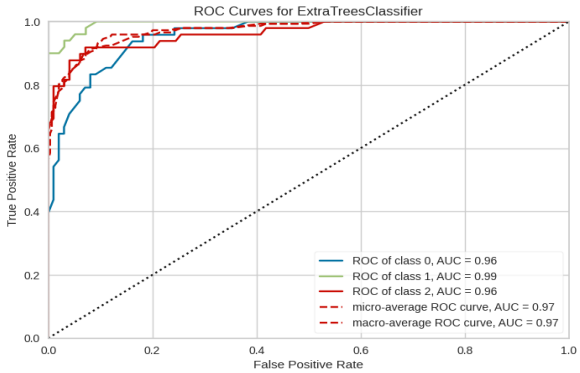


(d) ROC curve (tuned)

Figure 13. Figures (a) and (c) present the confusion matrices for the untuned and tuned models, respectively, while Figures (b) and (d) illustrate the corresponding ROC curves.



(a) Confusion Matrix (tuned/untuned)



(b) ROC curve (tuned/untuned)

Figure 14. Figures (a) and (b) display the confusion matrix and the ROC curve for both unturned and tuned cases, respectively.

not observed in previous studies [20, 21]. Training models on the original dataset led to biased pre-

dictions favoring the majority class (Status 1), whereas balancing significantly improved performance across all evaluation metrics, including accuracy, F1-score, and ROC-AUC. This study highlights the importance of data preprocessing and balancing techniques in improving predictive performance for clinically imbalanced health datasets.

6. Limitations of the Study

This study has several limitations that should be considered when interpreting the results. First, our analysis was restricted to the Mayo PBC cohort, which, while widely used and clinically validated, may not generalize to other populations or more recent cirrhosis cohorts. Second, the dataset is relatively small (418 patients) with an imbalanced class distribution, particularly for the transplant outcome. Although we applied SMOTE and reported parameters transparently, synthetic oversampling may not perfectly represent real-world clinical variability. Third, our models were trained on a fixed set of clinical and biochemical variables available in the dataset, meaning additional prognostic biomarkers or imaging features could further improve prediction accuracy but were not available for inclusion. Fourth, while we incorporated interpretability methods (permutation importance and SHAP) for the top-performing model, these techniques still provide correlational rather than causal explanations. Finally, we used cross-validation for evaluation, but external validation on an independent dataset was not possible, which limits direct clinical translation of our findings. Another limitation is that we did not compare our models directly with established clinical scoring systems such as MELD or Child–Pugh, since not all required variables were available in the dataset. Despite these limitations, the study provides a transparent benchmark for tri-class outcome prediction in cirrhosis, highlights the importance of preprocessing decisions, and offers a framework that can be extended to larger and more diverse datasets in future research.

7. Future Research

Building on this study, several avenues of future work can be pursued. First, validating the models on larger and more recent cirrhosis cohorts from diverse populations will strengthen generalizability and ensure broader clinical applicability. Second, integrating additional variables such as genomics, proteomics, imaging biomarkers, and lifestyle factors could improve predictive performance and provide deeper insights into patient heterogeneity. Third, more advanced imbalance handling techniques beyond SMOTE-such as adaptive synthetic sampling or cost-sensitive learning-should be explored to further reduce bias in minority outcome prediction. Fourth, extending beyond conventional machine-learning classifiers, deep learning architectures (e.g., recurrent or graph-based models) may capture nonlinear temporal dynamics more effectively, particularly when longitudinal follow-up data are available. Fifth, external validation using independent cohorts and prospective clinical trials will be critical to confirm the robustness and clinical utility of the models. Because this study relied on a single dataset (the Mayo PBC cohort), the generalizability of our findings to other populations, healthcare systems, and contemporary cirrhosis cohorts remains uncertain. Future research should therefore prioritize external benchmarking to strengthen clinical relevance. Finally, deployment-oriented studies are needed, including the development of user-friendly decision-support systems that can be integrated into clinical workflows. Such tools should not only be validated for

technical accuracy but also assessed for their real-world impact on patient care. In particular, future work should benchmark machine-learning models against established clinical scores such as MELD and Child-Pugh to ensure interpretability and alignment with existing clinical practice.

Funding Statement: There are no funders to report for this submission.
Acknowledgments: The authors sincerely thank the two anonymous reviewers for their valuable and constructive feedback, which has significantly contributed to improving the quality of this paper.
Conflict of Interest: The authors hereby declare that there is no conflict of interest in the publication of this paper.
Author's contributions: M.N.R.-conceptualization, methodology, software development, validation, formal analysis, investigation, writing—original draft, and writing—review & editing. M.I.H.-conceptualization, validation, investigation, and writing—review & editing. V.C.-conceptualization, formal analysis, and writing—review & editing. All authors have read and approved the final version of the manuscript.
Availability of data and materials: All data supporting the findings of this study are included within the article. The Python codes used for data preprocessing, model training, and evaluation are available from the authors upon request.

References

1. Elias, H., & Bengelsdorf, H. (1952). The structure of the liver of vertebrates. Cells Tissues Organs, 14(4), 297-337.
2. Abdel-Misih, S. R., & Bloomston, M. (2010). Liver anatomy. Surgical Clinics, 90(4), 643-653.
3. Tsochatzis, E. A., Bosch, J., & Burroughs, A. K. (2014). Liver cirrhosis. The Lancet, 383(9930), 1749-1761.
4. Huang, D. Q., Wong, V. W., Rinella, M. E., Boursier, J., Lazarus, J. V., Yki-Järvinen, H., & Loomba, R. (2025). Metabolic dysfunction-associated steatotic liver disease in adults. Nature Reviews Disease Primers, 11(1), 14.
5. Gan, C., Yuan, Y., Shen, H., Gao, J., Kong, X., Che, Z., ... & Xiao, J. (2025). Liver diseases: epidemiology, causes, trends and predictions. Signal Transduction and Targeted Therapy, 10(1), 33.
6. Khan, R. A., Luo, Y., & Wu, F. X. (2022). Machine learning based liver disease diagnosis: A systematic review. Neurocomputing, 468, 492-509.
7. Raihen, M. N., & Akter, S. (2024). Comparative assessment of several effective machine learning classification methods for maternal health risk. Computational Journal of Mathematical and Statistical Sciences, 3(1), 161-176.
8. Wieczorek, M., Weston, A., Ledenko, M., Thomas, J. N., Carter, R., & Patel, T. (2022). A deep learning approach for detecting liver cirrhosis from volatolomic analysis of exhaled breath. Frontiers in Medicine, 9, 992703.
9. Hanif, I., & Khan, M. M. (2022, October). Liver cirrhosis prediction using machine learning approaches. In 2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 0028-0034). IEEE.

10. Maryam, A. H., Sultan, L. R., Sagreiya, H., Cary, T. W., Karmacharya, M. B., & Sehgal, C. M. (2022, October). Machine learning improves early detection of liver fibrosis by quantitative ultrasound radiomics. In 2022 IEEE International Ultrasonics Symposium (IUS) (pp. 1-4). IEEE.

11. Cadranel, J. F., Rufat, P., & Degos, F. (2000). Practices of liver biopsy in France: results of a prospective nationwide survey. Hepatology, 32(3), 477-481.

12. Goldberg, D., Mantero, A., Kaplan, D., Delgado, C., John, B., Nuchovich, N., ... & Reese, P. P. (2022). Accurate long-term prediction of death for patients with cirrhosis. Hepatology, 76(3), 700-711.

13. Karna, A., Khan, N., Rauniyar, R., & Shambharkar, P. G. (2024). Unified dimensionality reduction techniques in chronic liver disease detection. arXiv preprint arXiv:2412.21156.

14. Rehman, A. U., Butt, W. H., Ali, T. M., Javaid, S., Almufareh, M. F., Humayun, M., ... & Shaheen, M. (2024). A Machine Learning-Based Framework for Accurate and Early Diagnosis of Liver Diseases: A Comprehensive Study on Feature Selection, Data Imbalance, and Algorithmic Performance. International Journal of Intelligent Systems, 2024(1), 6111312.

15. Malik, S., Frey, L. J., & Qureshi, K. (2025). Evaluating the predictive power of machine learning in cirrhosis mortality: a systematic review. Journal of Medical Artificial Intelligence, 8.

16. Guo, A., Mazumder, N. R., Ladner, D. P., & Foraker, R. E. (2021). Predicting mortality among patients with liver cirrhosis in electronic health records with machine learning. PloS one, 16(8), e0256428.

17. Kanwal, F., Taylor, T. J., Kramer, J. R., Cao, Y., Smith, D., Gifford, A. L., ... & Asch, S. M. (2020). Development, validation, and evaluation of a simple machine learning model to predict cirrhosis mortality. JAMA network open, 3(11), e2023780-e2023780.

18. Raihen, M. N., Begum, S., Akter, S., & Sardar, M. N. (2025). Leveraging Data Mining for Inference and Prediction in Lung Cancer Research. Computational Journal of Mathematical and Statistical Sciences, 4(1), 139-161.

19. Tsai, S. C., Lin, C. H., Chu, C. C., Lo, H. Y., Ng, C. J., Hsu, C. C., & Chen, S. Y. (2024). Machine learning models for predicting mortality in patients with cirrhosis and acute upper gastrointestinal bleeding at an emergency department: A retrospective cohort study. Diagnostics, 14(17), 1919.

20. Yamaganti, R., Nair, A. A., Reddy, V. K., & Botika, T. (2024, March). Cirrhosis Patient Survival Prediction Analysis Using ML Algorithms. In International Conference on Artificial Intelligence and Smart Energy (pp. 120-137). Cham: Springer Nature Switzerland.

21. Sousa, R., Passos, M., Almeida, M., Ribeiro, M., & Peixoto, H. (2024). Harnessing data mining to predict survival outcomes in patients with hepatic cirrhosis. Procedia Computer Science, 238, 938-943.

22. Liu, J. (2022). Importance-SMOTE: a synthetic minority oversampling method for noisy imbalanced data. Soft Computing, 26(3), 1141-1163.

23. Narkhede, S. (2018). Understanding auc-roc curve. Towards data science, 26(1), 220-227.

24. Himel, G. M. S., & Islam, M. M. (2025). A Smart Intelligence System for Hen Breed and Disease classification using Extra Tree classifier-based Ensemble Technique. Journal of Electrical Systems and Information Technology, 12(1), 2.

25. Hussain, I., Qureshi, M., Ismail, M., Iftikhar, H., Zywiołek, J., & López-Gonzales, J. L. (2024). Optimal features selection in the high dimensional data based on robust technique: Application to different health database. Heliyon, 10(17).

26. Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.

27. Ahmad, F., Alasskar, A., Samui, P., & Asteris, P. G. (2025). Machine learning-based graphical user interface for predicting high-performance concrete compressive strength: comparative analysis of gradient boosting machine, random forest, and deep neural network models. Frontiers of Structural and Civil Engineering, 19(7), 1075-1090.

28. Ferreira, P., Martins, E., Silva, J., & Teixeira, P. (2025, April). Feature Selection and XGBoost for Enhanced Intrusion Detection: A Comparative Study Across Benchmark Datasets. In 2025 13th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.

29. Wang, S. S., Yan, J., & Geng, H. (2025). Prediction of the ROP based on GA-LightGBM and drilling data. Geosystem Engineering, 28(1), 12-30.

30. Noviandy, T. R., Maulana, A., Irvanizam, I., Idroes, G. M., Maulydia, N. B., Tallei, T. E., ... & Idroes, R. (2025). Interpretable machine learning approach to predict Hepatitis C virus NS5B inhibitor activity using voting-based LightGBM and SHAP. Intelligent Systems with Applications, 25, 200481.

31. Muhammad, A. C., & Sari, R. F. (2025, July). Extreme Gradient Boosting with XAI Feature Importance for Energy Prediction. In 2025 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT) (pp. 491-497). IEEE.

32. McLachlan, G. J. (2005). Discriminant analysis and statistical pattern recognition. John Wiley & Sons.

33. Almetwally, E. M., Elbatal, I., Elgarhy, M., & Kamel, A. R. (2025). Implications of machine learning techniques for prediction of motor health disorders in Saudi Arabia. Alexandria Engineering Journal, 127, 1193-1208.

34. Asbai, N., Bounazou, H., & Zitouni, S. (2025). A novel approach to deriving adaboost classifier weights using squared loss function for overlapping speech detection. Multimedia Tools and Applications, 1-28.

35. Kumar, V., Shukla, S., & Gyanchandani, M. (2025). Using Machine Learning-Based AdaBoost Algorithm. Data Science and Applications: Proceedings of ICDSA 2024, Volume 2, 1264, 161.

36. Kaur, S., Gupta, S., Gupta, D., Juneja, S., Nauman, A., Khan, M., ... & Mallik, S. (2025). High-accuracy lung disease classification via logistic regression and advanced feature extraction techniques. Egyptian Informatics Journal, 29, 100596.

37. Dey, D., Haque, M. S., Islam, M. M., Aishi, U. I., Shammy, S. S., Mayen, M. S. A., ... & Uddin, M. J. (2025). The proper application of logistic regression model in complex survey data: a systematic review. BMC Medical Research Methodology, 25(1), 15.

38. Amer, A. A., Ravana, S. D., & Habeeb, R. A. A. (2025). Effective k-nearest neighbor models for data classification enhancement. Journal of Big Data, 12(1), 86.

39. Mintogo, L. B., Nkou, E. D. D., & Nkiet, G. M. (2025). Nonparametric estimation of sliced inverse regression by the *k*-nearest neighbors kernel method. arXiv preprint arXiv:2505.19359.

40. Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).

41. Chen, H., Lin, Z., Chen, Z., Jian, J., & Liu, C. (2025). Adaptive DWA algorithm with decision tree classifier for dynamic planning in USV navigation. Ocean Engineering, 321, 120328.

42. Achari, A. P. S. K., & Sugumar, R. (2025, March). Performance analysis and determination of accuracy using machine learning techniques for decision tree and RNN. In AIP Conference Proceedings (Vol. 3252, No. 1, p. 020008). AIP Publishing LLC.

43. Huang, W., Cai, Y., & Zhang, G. (2025). Battery degradation analysis through sparse ridge regression.

44. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B: Statistical Methodology, 58(1), 267-288.

45. Zhao, S., Zhang, B., Yang, J., Zhou, J., & Xu, Y. (2024). Linear discriminant analysis. Nature Reviews Methods Primers, 4(1), 70.

46. Patle, A., & Chouhan, D. S. (2013, January). SVM kernel functions for classification. In 2013 International conference on advances in technology and engineering (ICATE) (pp. 1-9). IEEE.

47. Ray, K. K., Kumari, A., Kumar, S., Machavaram, R., Shekh, I., Deshmukh, S. M., & Tadge, P. (2025). Guava Leaf Disease Detection Using Support Vector Machine (SVM). Smart Agricultural Technology, 101190.

48. Jordanov, I., Petrov, N., & Petrozziello, A. (2018). Classifiers accuracy improvement based on missing data imputation. Journal of Artificial Intelligence and Soft Computing Research, 8(1), 31-48.

49. Fleming, T. R., & Harrington, D. P. (2013). Counting processes and survival analysis. John Wiley & Sons.

50. Nahm, F. S. (2022). Receiver operating characteristic curve: overview and practical use for clinicians. Korean journal of anesthesiology, 75(1), 25-36.