Liver Disease Prediction from Clinical Data: An Ensemble Classification Approach

Matías Mora (DNI: 49753972G)
Marc Cascant (DNI: 23896796A)
Universitat Politècnica de Catalunya, Barcelona, Spain

***Abstract*—This study presents a comprehensive machine learning pipeline for predicting liver disease from structured clinical and demographic data, developed in the context of a classification competition. The dataset comprises 579 patients with a strong class imbalance favoring liver disease cases. A modular framework was used to systematically combine preprocessing, resampling, and classification stages, through an exhaustive search that identifies the best configuration for each model. Various models—including linear, probabilistic, instance-based, and ensemble classifiers—were optimized and evaluated through stratified cross-validation using macro-averaged F1-score as the primary metric. Extensive experimentation revealed the importance of correlation handling, proper scaling, and careful resampling to mitigate bias and overfitting. The final model is a soft voting ensemble integrating four high-performing classifiers—LDA, QDA, KNN, and Logistic Regression—each embedded in a custom preprocessing pipeline. This model achieved strong validation results, with a macro F1-score of 0.71 and 0.69 on a train test split set. The project underscores the importance of modular preprocessing pipelines for ensuring reproducibility and maintaining a clean, transparent workflow. These pipelines allow each classifier to be paired with its most effective preprocessing configuration, simplifying experimentation and avoiding data leakage. It also highlights the practical trade-offs between model complexity, computational efficiency, and predictive performance. Ultimately, a custom ensemble strategy—built by combining diverse classifiers, each embedded in its own optimized preprocessing pipeline and aggregated via soft voting—proved to be the most effective solution for this real-world medical classification task.**

I. Introduction

The objective of this work is to develop a predictive model capable of determining whether a patient is affected by liver disease based on a set of clinical and demographic variables. The classification task is based on a real-world medical dataset containing records from patients in North-East Andhra Pradesh, India, and involves a binary outcome: identifying individuals as either healthy or suffering from liver disease.

The dataset comprises ten features per patient, including biochemical markers such as total and direct bilirubin, enzyme levels, and albumin ratios, as well as demographic information such as age and gender. The data is notably imbalanced, with a higher proportion of liver disease cases, posing an additional challenge for traditional classification algorithms.

This report presents a comprehensive pipeline for addressing this problem, including preprocessing, feature analysis, model selection, hyperparameter tuning, and performance evaluation. Multiple classification algorithms have been considered and compared, with particular attention paid to strategies for managing class imbalance and improving model generalization. All methods and processing steps are described in detail in the following sections of this report.

Model performance is assessed using the F1 score, a metric well-suited for imbalanced binary classification, as it captures a balance between precision and recall. The goal is to identify and optimize the model that achieves the highest possible F1 score on unseen data, ensuring both accuracy and robustness in predictive performance.

II. Exploratory Feature Analysis

A comprehensive exploratory data analysis (EDA) was conducted to examine the statistical properties, relationships, and potential issues in the dataset features. The objective of this analysis is to inform downstream preprocessing, feature selection, and modeling decisions by identifying sources of bias, redundancy, or noise.

A. Target Variable Distribution

The target variable is binary, with a notable imbalance between classes, as we mentioned. The proportion of class 0 significantly exceeds that of class 1 (71% and 29% respectively), raising concerns about model bias and reduced sensitivity in minority class detection.

B. Correlation Structure

To assess linear dependencies among the numerical features, we computed the Pearson correlation matrix (Figure 1). Several pairs of variables exhibit strong positive correlations, notably *SGOT* and *SGPT* ($r = 0.91$), *DB* and *TB* ($r = 0.85$), *ALB* and *TP* ($r = 0.72$), *A/G ratio* and *ALB* ($r = 0.72$), and *ALB* and *TB* ($r = 0.80$). These associations are physiologically consistent, as the variables measure related liver functions or protein levels. Nevertheless, such collinearity can introduce redundancy in tree-based models or instability in linear ones, reinforcing the need for regularization and embedded feature selection. Techniques to explicitly mitigate multicollinearity will be applied in subsequent steps.

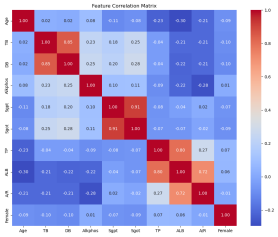

Fig. 1. Correlation matrix of selected numerical features.

C. Presence of Outliers

Boxplot inspection revealed the presence of significant outliers in several clinical measurements, particularly in liver-related variables such as *SGOT*, *SGPT*, *ALP*, *DB*, and *TB*. In contrast, variables such as *TP*, *ALB*, and the *A/G ratio* showed no apparent outliers, and *Age* also appeared well-behaved. While some of these extreme values may reflect genuine pathological conditions, they can exert disproportionate influence on scale-sensitive algorithms. To mitigate this, robust normalization strategies such as the Yeo-Johnson transformation were applied, which are effective for handling skewed and non-positive data. Although outlier detection will not be explicitly incorporated in the final model due to limited performance gains, as discussed later, different detection methods consistently identified between 60 and 70 outliers on average — a non-negligible number that highlights the potential for distortion in raw clinical data.

D. Distributional Properties and Normality

An inspection of the feature distributions revealed substantial deviations from normality in most variables, with marked right skewness and heavy tails. *TP* and *ALB* were the closest to normality, while others such as *SGOT*, *SGPT*, *ALP*, *DB*, and *TB* showed strong asymmetry. The distributions appear largely independent of the target class, making discrimination difficult; only *ALB* shows slight class separation. Additionally, class 0 tends to exhibit heavier tails. To mitigate these issues, normalization and scaling techniques were applied to improve distributional symmetry and model compatibility.

E. Feature Relevance and Ranking

Feature relevance was assessed using two distinct methods: a Random Forest–based importance ranking and the univariate ANOVA F-test via `SelectKBest`. The results differed notably between the two, highlighting that feature selection is sensitive to the evaluation criterion. However, certain variables emerged consistently among the top ranks, most notably *Alkaline Phosphatase* (*Alkphos*), *SGOT*, and *Total Bilirubin* (*TB*), all of which have well-established clinical relevance to liver function. These features will serve as a foundation for constructing focused, interpretable models in subsequent stages of the analysis.

III. Classification Methods

To address the binary classification task, a wide range of supervised learning algorithms were explored and evaluated. These include both individual classifiers and ensemble-based techniques, allowing for a comparative analysis of their predictive capabilities in the context of imbalanced medical data.

A. Preprocessing Strategy

The preprocessing stage was designed to be modular and flexible, allowing tailored pipelines to be combined with each model and resampling strategy. Each pipeline could include several optional steps, selected depending on model compatibility and relevance. This way, we could exhaustively find the best preprocessing for each model.

Correlation Handling. To address multicollinearity, we explored various strategies including pairwise regression techniques or divisions and feature engineering based on variable ratios. Dimensionality reduction via PCA was also evaluated but yielded weaker results and was discarded.

Normalization. Certain models, especially distance-based ones like KNN or SVM, are sensitive to the scale of input features. We tested pipelines with no normalization or with power transformations such as Yeo-Johnson, which can accommodate skewed distributions and zero values.

Scaling. After normalization, data were either left unchanged or scaled using MinMax or Robust scalers. RobustScaler was especially useful for mitigating the influence of outliers in numerical features.

Feature Selection. To reduce dimensionality and enhance generalization, we tested SelectKBest with various scoring functions. In some configurations, all features were retained; in others, only the most relevant subsets were kept.

These components were defined as pipeline steps and dynamically assembled to create candidate configurations for each model. This approach facilitated efficient exploration of preprocessing choices across models with diverse inductive biases.

B. Models

A wide range of classification algorithms was explored to evaluate their effectiveness on the liver disease dataset. These include both linear and nonlinear models, generative and discriminative approaches, and ensemble-based techniques. The goal was to compare their performance under consistent evaluation conditions and identify which are best suited to the characteristics of our data.

- **Linear Models:** Logistic Regression (LR) and Linear Discriminant Analysis (LDA) offer interpretability and computational efficiency, and are particularly effective under linear separability assumptions.
- **Quadratic and Probabilistic Models:** Quadratic Discriminant Analysis (QDA) and Gaussian Naive Bayes (NB) can model nonlinear class boundaries by allowing class-specific distributions. These models rely on strong assumptions about feature independence or homoscedasticity.
- **Kernel and Margin-Based Models:** Support Vector Machines (SVM) were evaluated using both standard and custom-designed kernels tailored to the feature space in

order to improve separation. Their performance is highly sensitive to scaling and class overlap.

- **Instance-Based Learning:** K-Nearest Neighbors (KNN) uses distance-based decision rules, which require proper normalization due to the influence of feature magnitude on proximity.
- **Tree-Based Models:** Standalone decision trees were considered for interpretability, but their main utility was as base learners within ensemble methods.
- **Regular Ensemble Methods:** Ensemble approaches such as Bagging (Random Forest, Extra Trees) and Boosting (AdaBoost, Gradient Boosting, XGBoost, HistGradientBoosting) were included for their robustness, ability to model complex interactions, and strong empirical performance across various pipelines.
- **Custom Ensemble Strategies:** In addition to standard ensembles, custom aggregation functions were implemented. These combine the top-performing individual models—selected based on validation performance—using manually assigned weights within soft voting or stacking frameworks. This allowed us to design ensembles tailored to the specific strengths of selected classifiers.

Each model was evaluated in combination with relevant resampling strategies and preprocessing pipelines (as described previously). Hyperparameters were tuned using randomized search, and performance was assessed using F_1-score as the primary metric, as detailed in the following sections.

C. Imbalance treatment

Given the strong class imbalance in the dataset, various strategies were applied to mitigate bias toward the majority class. These included both resampling techniques and algorithm-specific mechanisms.[1]

Resampling. For each model, we evaluated several resamplers (undersamplers and oversamplers) to observe their impact on performance. The methods included Tomek Links, SMOTENC (with default and reduced neighborhood size), SMOTETomek, NearMiss, and RandomOverSampler. Additionally, models were also trained without resampling to serve as a baseline. These were integrated into the preprocessing pipelines and evaluated systematically across model configurations.

Class Weighting. Where supported (e.g., in Logistic Regression, SVM, Decision Trees, and Gradient Boosting), we used `class_weight="balanced"` to assign inverse-frequency weights to classes. This approach adjusts the model's objective function to penalize misclassification of the minority class more heavily.

Overall, our classification approach integrates a comprehensive set of tools, combining systematic preprocessing, a wide range of modeling strategies—from generative (Naive Bayes, LDA) to discriminative (Logistic Regression, SVM), and from simple learners to advanced ensemble methods—and targeted techniques to address class imbalance. By evaluating these components in a modular and comparative framework, we aim to determine which combinations are most effective for the liver disease prediction task. This is done in light of the dataset's intrinsic challenges, including small sample size, mixed numerical and categorical features, and class imbalance.

[1]Threshold tuning was tested but discarded due to overfitting, as detailed in Section V.

IV. Evaluation Metrics

To assess model performance under class imbalance, we prioritized the F_1-score as our main evaluation metric. It balances precision and recall and is especially informative when accuracy may be misleading due to majority-class bias. It is defined as:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1}$$

Precision and recall themselves measure the rate of false positives and false negatives, respectively:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

Given the medical context, high recall (catching true cases) is crucial, while precision prevents unnecessary false alarms. Additional metrics—accuracy, class-wise F_1, and support—were also tracked for completeness.

For the best models, we further computed ROC-AUC and Precision–Recall curves, as well as other benchmarks like the confusion matrix, although F_1 remained our primary selection criterion. Learning curves were used during development to diagnose under- or overfitting when needed.

All metrics were computed via cross-validation to ensure unbiased evaluation across configurations. This consistent setup supports the experimental comparisons described next.

V. Experiments

The experimental framework was designed to systematically evaluate classification performance across combinations of models, resampling strategies, and preprocessing pipelines. Each model was paired with compatible resamplers and preprocessing configurations, as described in earlier sections. These components were integrated into complete pipelines using `ImbPipeline`, ensuring modularity, consistency, and compatibility with cross-validation procedures. A schematic of the pipeline sequence is shown in Figure 2.

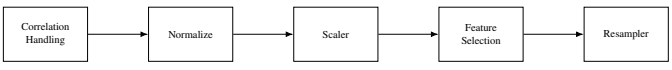

Fig. 2. Pipeline structure used for preprocessing and resampling.

The number of configurations tested per model varied depending on their complexity and training time. In the most exhaustive cases, up to $6 \times 2 \times 3 \times 4 \times 7 = 1008$ combinations were explored, corresponding respectively to the number of variants at each pipeline stage as shown in Figure 2. Simpler

models were evaluated more extensively, while computationally expensive models, such as ensemble trees, were tested with a reduced configuration set.

For each model configuration (as in pipeline), a tailored hyperparameter search space was defined and optimized using `RandomizedSearchCV` with 25 iterations and 5-fold stratified cross-validation. This strategy provided a good balance between computational efficiency and exploratory coverage. Earlier experiments using `GridSearchCV` and 50 iterations were computationally intensive with marginal gains, leading to the adoption of randomized search.

All evaluations were performed through cross-validation in the entire training set, as internal validation proved to be more robust than a hold-out approach during the initial testing phases. Random seeds were fixed where possible to ensure reproducibility. Each configuration was assessed using a consistent set of metrics—accuracy, precision, recall, and class-wise F_1—with F_1-score used as the primary optimization criterion due to class imbalance.

Several techniques were evaluated but excluded from the final pipeline due to poor generalization. These include threshold tuning—prone to overfitting on training folds—and outlier removal based on z-score or IQR, which resulted in information loss and weaker performance, particularly before scaling or robust models.

To facilitate interpretation, model evaluations were carried out both individually and by grouped categories—such as algorithm family, resampling method, and preprocessing configuration. This enabled us to assess the relative impact of each design choice and to identify patterns of consistently strong performance across different classes of models.

These results were used to construct both per-group rankings—within each model family (differentiating by preprocessing strategy and resampling method) —and a unified global ranking based primarily on macro F_1-score. This dual evaluation allowed for both intra-group comparison and overall model selection. The top-performing configuration from the global ranking was ultimately chosen for final analysis and interpretation.

A. Model Configurations and Experimental Variants

Table I summarizes the tested configurations per model group. ITuses abbreviated tags to summarize preprocessing configurations: **Resamp. Basic** refers to standard resampling techniques such as Tomek Links, SMOTE, SMOTETomek, and Random Oversampling; **Corr. General** includes domain-informed correlation handling such as computing biochemical ratios (e.g., DB/TB, Sgot/Sgpt) or regression-based residuals to reduce collinearity; **Corr. Simple** applies only a subset of these (typically 2-variable ratios); and **Corr. Identity** skips correlation correction entirely. **Select. General** includes simple selectors like `SelectKBest`, while **Select. RF/XGB** uses model-based selection with Random Forests or XGBoost, and **Identity** indicates no feature selection.

Model Group	Models	Resamp.	Corr.	Select.
Linear Models	Logistic, SVM	Basic	General	General
Discriminant Analysis	LDA, QDA	Basic	General	LDA: Gen., QDA: —
Probabilistic Models	NB	Basic	General	General
Instance-Based	KNN	Full	General	General
Tree-Based	DT, RF, ET	DT: B, others: —	Simple	RF / Gen.
Boosting Trees	GB, AdaB, HGB	Basic	Simple	Identity
XGBoost Variants	XGB (Std., Min.)	XGB	General	XGB
Bagging Methods	Bagging (DT, KNN, NB)	SMOTETomek None	DT: Simple, others: General	DT: RF, others: Gen.

TABLE I
GROUPED MODELS WITH ASSOCIATED RESAMPLING METHODS, CORRELATION TECHNIQUES, AND FEATURE SELECTION STRATEGIES.

VI. RESULTS

In this section, we provide a concise analysis of the most relevant model explored in the experimental phase. To select it, for each standard classifier, we report its best macro F1-score obtained via cross-validation, highlight its main strengths and limitations, and specify the preprocessing and resampling combination that led to its optimal performance.

This analysis is intended to identify not only which model perform best individually, but also under what conditions it excels, offering insights into its robustness, sensitivity to class imbalance, and interaction with specific feature transformations.

The final selected model is a `VotingClassifier` using soft voting, composed of the four best-performing individual classifiers:

- **QDA** without resampling, Yeo-Jonhson, MinMax, all features
- **LDA** with Tomek Links, Yeo-Johnson, RobustScaler, best 5
- **KNN** with SMOTETomek, Yeo-Johnson, RobustScaler, best 5
- **Logistic Regression** with Random Oversampling, no normalization, RobustScaler, all features

Each classifier was embedded in its own dedicated preprocessing pipeline and trained using optimized hyperparameters found through randomized search. The ensemble applied a soft-voting strategy, assigning equal weights to each of the four classifiers (weights = [2, 2, 2, 2]). Thanks to the modular pipeline structure, each model retained its own optimal correlation handling, normalization, feature selection, and sampling strategy—enhancing diversity and ensuring strong individual performance within the ensemble.

The ensemble achieved the following average metrics across validation folds:

TABLE II
PERFORMANCE METRICS OF THE FINAL VOTINGCLASSIFIER ENSEMBLE.

Model	Accuracy	Precision	Recall	F1 Class 0	F1 Class 1	F1
Voting Ensemble	0.741	0.533	0.727	0.805	0.615	0.710

Firstly, we may say that the configuration of the best model showcases the critical role of preprocessing in achieving strong results. Careful choices in normalization, scaling, feature selection, and resampling directly influenced each classifier's performance, proving that our exhaustive pipeline selection is indeed effective.

These results indicate that the ensemble not only achieved good overall accuracy but also maintained balance between precision and recall. In particular, the relatively high recall (0.73) for the positive class (patients with liver disease) is crucial in this context, as failing to identify affected individuals could have serious consequences. The class-wise F1-scores reveal stronger performance on the majority class, but with reasonable trade-offs in class 1 performance, consistent with our optimization strategy.

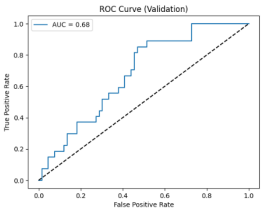

Fig. 3. ROC curve of the final ensemble model (AUC = 0.68).

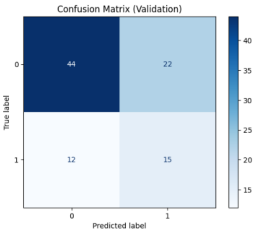

Fig. 4. Confusion matrix of the final model on validation data.

The confusion matrix (Figure 4) shows that the model correctly predicted 15 of the 27 patients with liver disease, while maintaining a relatively low number of false negatives (12). Although 22 healthy individuals were misclassified as diseased, this reflects a cautious bias in favor of recall — a design choice aligned with the clinical motivation. The ROC curve (Figure 3) shows an AUC of 0.68, suggesting moderate ability to separate the classes across thresholds.

Note that these diagnostic plots were generated from a final train-validation split, created exclusively for visualization purposes after model selection. All reported metrics, however, stem from cross-validation to ensure unbiased performance estimates.

Overall, the ensemble outperformed most individual models, confirming the benefit of combining multiple classifiers with diverse behaviors. This model was submitted as the final candidate, given its robustness, modularity, and strong validation metrics across the target metrics. However, it's worth noting that both XGBoost and Logistic Regression stood out as particularly effective "lightweight" models. Their simplicity, speed, and relatively high validation scores make them attractive options in scenarios where computational efficiency is a priority.

VII. Conclusions

This project has provided a comprehensive and hands-on opportunity to explore the full machine learning pipeline in a medical classification context. From data exploration and preprocessing to model training, tuning, and final evaluation, each stage contributed key insights into the practical challenges of building robust predictive systems. In particular, we deepened our understanding of how different types of classifiers behave under varying preprocessing and resampling strategies, and how sensitive results can be to seemingly minor design decisions.

A major lesson learned was the importance of appropriate data treatment and the prevention of information leakage. Early experiments revealed how improper application of transformations or resampling outside the cross-validation loop could lead to inflated scores and misleading conclusions. This underscored the necessity of carefully designed pipelines that respect the independence of validation folds and isolate data-driven steps. To address this, we relied heavily on a modular and standardized pipeline structure, which proved critical not only for preventing leakage but also for enabling reproducibility, clarity, and flexibility when experimenting with different configurations. This design choice made it possible to interchange preprocessing techniques, samplers, and classifiers with minimal overhead, thus streamlining the entire workflow.

As experimentation grew in scale and complexity, we also encountered the inherent trade-off between exhaustive optimization and computational feasibility. Some models, especially tree-based ensembles, demanded significant runtime, leading us to prioritize smarter exploration strategies such as randomized hyperparameter search. Balancing model variety with execution time became a key consideration.

Notably, ensemble methods such as voting and stacking classifiers consistently delivered the best results. By combining diverse models trained on complementary configurations, these ensembles demonstrated improved generalization and stability compared to individual classifiers. The final selected model—a soft voting ensemble of LDA, KNN, Logistic Regression, and QDA—embodied this strategy and achieved a macro F_1-score of 0.71 on validation and 0.69 on the competition test set, confirming its robustness.

Overall, this work illustrates that successful predictive modeling goes beyond selecting a strong algorithm. It requires thoughtful data preprocessing, rigorous validation, a well-structured and modular pipeline, awareness of potential pitfalls like imbalance and leakage, and sometimes sacrificing theoretical optimality in favor of practical feasibility. These are lessons that extend well beyond this task and into broader real-world data science applications.

References

[1] M. Lichman, *UCI Machine Learning Repository: Liver Disorder Data Set*, University of California, Irvine, 2013.

[2] M. A. Kuzhippallil, C. Joseph, and A. Kannan, "Comparative analysis of machine learning techniques for Indian liver disease patients," in *Proc. 6th Int. Conf. on Advanced Computing and Communication Systems (ICACCS)*, 2020, pp. 778–782.

[3] S. Kulkarni, C. J. Joshua, T. Vaichole, S. Mohan, and C. Iwendi, "Ensemble machine learning algorithms for detecting liver disease with enhanced preprocessing," *Diagnostics*, vol. 15, no. 6, 2023, Art. ID 1447.