**CIS 550: ADVANCED MACHINE LEARNING (SPRING 2025)**

**FINAL PROJECT SUMMARY**

**Hepatitis Disease Survival Prediction**

**Using Machine Learning Models**



**GROUP – 11**

**(16) Rakesh Kumar – 02188914**

**(17) John T Lally - 02010493**

**TABLE OF CONTENTS**

1. Introduction & Motivation
2. Background
3. Problem Statement & Objectives
4. Dataset Description
5. Methodology
   1. Environment and Tools
   2. Data Preprocessing
   3. Exploratory Data Analysis
   4. Modeling Approaches
   5. Evaluation Metrics
6. Results
7. Discussion
8. Limitations & Future Work
9. Conclusion
10. References

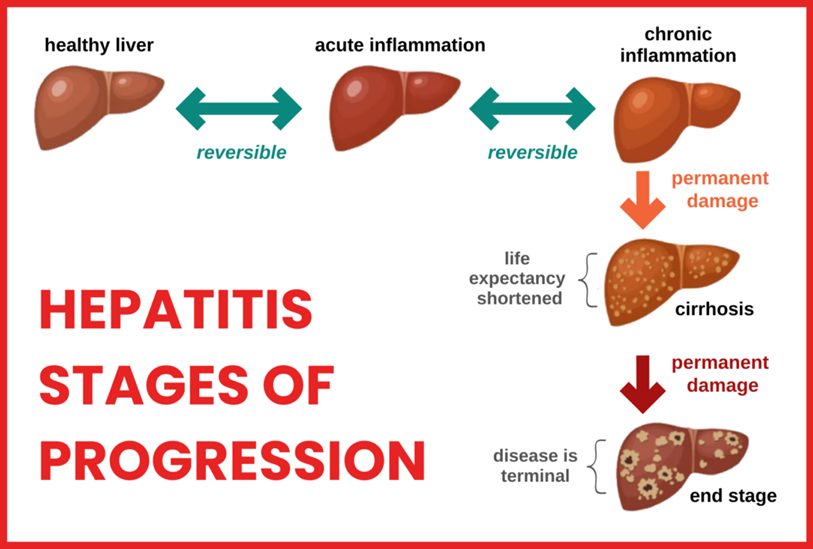
**INTRODUCTION AND MOTIVATION**

The liver is the body’s biochemical powerhouse: every drop of blood leaving the digestive track first flows through it for detoxification, nutrient processing, and metabolic regulation. When hepatitis, inflammation of the liver, strikes this vital organ can deteriorate rapidly, complicating clinical decision making. Clinicians triage hepatitis patients daily, but the decision to escalate care often hinges on fragmented thresholds or subjectivity. Machine learning thrives on this type of multivariate pattern finding.

The motivation for this project has two main focuses: clinical impact and learning value. The clinical impact of flagging high risk hepatitis patients early can trigger timely antiviral therapy, closer monitoring, and expedited transplant referral. The learning value is an end-to-end demonstration of tabular machine learning on a healthcare dataset, exhibiting every phase of the pipeline from data cleaning to model interpretation.

**BACKGROUND**

Hepatitis can arise from viral infection (types A through E), alcohol, drugs, and autoimmune attack. Viral hepatitis B and C affect around 350 million people worldwide. Disease trajectory ranges from asymptomatic to liver failure and death. Despite decades of research, there is no singular scalar score that perfectly stratifies risk across all hepatitis etiologies. Clinicians utilize tools such as MELD score, but these capture only a subset of laboratory information and are utilized primarily for transplant allocation. A data driven approach that integrates clinical flags with serum markers may reveal additional signal. Early identification of high-risk patients can guide timely interventions.



**PROBLEM STATEMENT**

The objective of this project is to build a machine learning classification model to predict whether a hepatitis patient is likely to survive (LIVE) or not (DIE), based on a range of clinical and laboratory features. This dataset-driven approach provides an opportunity to systematically apply and evaluate various ML algorithms in a healthcare-related context. While the model is for academic purposes, it helps demonstrate how data science techniques can be used to explore patterns and make basic predictions in medical datasets.

There are four specific objectives: build, compare, explain, and document. First, we build three distinct models: logistic regression, random forest, and XGBoost. Then, we compare performance using metrics including sensitivity, specificity, and AUC-ROC. Then, we explain the influential features to maintain clinician trust. Finally, we document the entire pipeline in a reproducible Jupyter notebook.

**DATASET DESCRIPTION**

The dataset used in this project is being taken from UC Irvine Machine learning Repository. The hepatitis dataset contains medical records of 155 patients diagnosed with hepatitis at a hospital in the 1980s, with the goal of predicting their survival status (LIVE or DIE). It consists of 20 attributes, including demographic features like age and sex, along with various clinical and biochemical indicators such as fatigue, liver condition, bilirubin levels, and prothrombin time. Most features are binary (yes/no), while a few are numerical and represent lab test results. The target variable is Class, indicating whether the patient survived.

This dataset provides a suitable foundation for applying classification algorithms and understanding the influence of different health indicators on patient outcomes. Data preprocessing steps are necessary to handle missing values and prepare the features for modeling.

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Attribute** | **Information** | **Type** |
| 1 | Class | Die, Live | Binary |
| 2 | Age | Number of years | Numeric |
| 3 | Sex | Male, Female | Binary |
| 4 | Steroid | No, Yes | Binary |
| 5 | Antivirals | No, Yes | Binary |
| 6 | Fatigue | No, Yes | Binary |
| 7 | Malaise | No, Yes | Binary |
| 8 | Anorexia | No, Yes | Binary |
| 9 | Liver Big | No, Yes | Binary |
| 10 | Liver Firm | No, Yes | Binary |
| 11 | Spleen Palpable | No, Yes | Binary |
| 12 | SPiders | No, Yes | Binary |
| 13 | Ascites | No, Yes | Binary |
| 14 | Varices | No, Yes | Binary |
| 15 | Bilirubin | Blood results | Numeric |
| 16 | ALK Phosphate | Blood results | Numeric |
| 17 | SGOT | Blood results | Numeric |
| 18 | Albumin | Blood results | Numeric |
| 19 | Protime | Blood results | Numeric |
| 20 | Histology | No, Yes | Binary |

**METHODOLOGY**

**COMPUTING ENVIRONMENT AND TOOLS**

|  |  |
| --- | --- |
| Category | Tool/Version |
| Language | Python 3.11 |
| Notebook | JupyterLab 4.0 |
| Machine Learning Libraries | scikit‑learn 1.5, xgboost 2.0, imbalanced‑learn 0.12 |
| Visualization | matplotlib 3.9, seaborn 0.13 |

**DATA PREPROCESSING:**

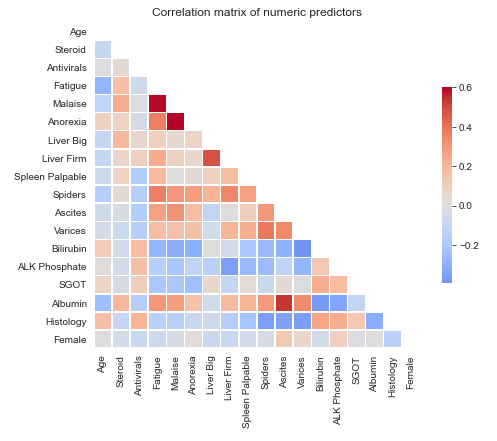
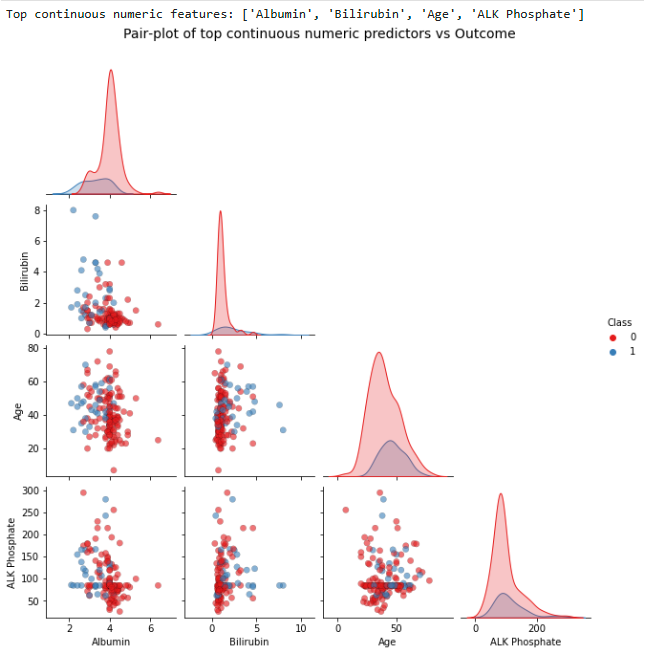
Several of the features have between 10-30% missing values. These were handled using either median or mode imputation, depending on the feature type. Categorical features had missing values imputed via mode, while numerical features had missing values imputed via median. The protime variable had the most missing values out of any feature, with nearly half of its values missing. Due to such a large number of missing values, this feature was dropped entirely. Each feature was type cast depending on the feature type. Categorical features were type cast as int8, while numerical features were type cast as float32. Categorical values were converted from 1 and 2 to 0 and 1 to keep things organized for the machine learning model. No scaling was used, as the tree models are unaffected and the logistic regression is robust with raw inputs. Finally, the cleaned frame was partitioned stratifying on the target into 80% training, 10% validation, and 10% test sets.

**EXPLORATORY DATA ANALYSIS**

The exploratory data analysis (EDA) was performed after the data preprocessing to explore the dataset and spot initial patterns that may be useful during later steps. The outcome distribution of the target class was approximately 90% of patients who lived (0) and 10% of patients died (1).

Key patterns emerged, as can be seen below in the selected pair plot on the left and correlation matrix on the right. The four numeric features of albumin, bilirubin, age, and ALK phosphate had the strongest correlation with class and were plotted in the pair plot, with red representing alive (0) and blue representing dead (1).

Some of the key numeric separators were albumin which was lower in non-survivors and bilirubin which was higher in non-survivors. For the clinical flags, ascietes and spider angiomas were both more prevalent among deaths. The features were found to be independent, with no extreme multicollinearity among any of the lab measures. These insights confirm that both labs and clinical signs carry strong predictive signal.

A pair-plot and correlation matrix of numeric predictors

**MODELING APPROACHES**

Three class-balanced models were chosen for this machine learning application: XGBoost (XGB), Logistic Regression (LR), and Random Forest (RF). The XGBoost (XGB) model is a gradient‑boosted trees tuned for AUC, excels at subtle non‑linear patterns. Logistic Regression (LR) serves as the interpretable baseline, easy to deploy and explain to clinicians. The Random Forest (RF) model is an ensemble of 100 bagged trees, robust to noise and outliers while capturing non‑linear interactions.

The notebook first partitioned the cleaned dataframe stratified on the outcome into 80% training, 10% validation, 10% test. Because only 10% of patients in the corpus died, the LR and RF models were told to apply class weights inversely proportional to class frequency, ensuring the minority class was not ignored. XGBoost was kept deliberately shallow (17 boosted trees) to avoid over-fitting on the tiny sample.

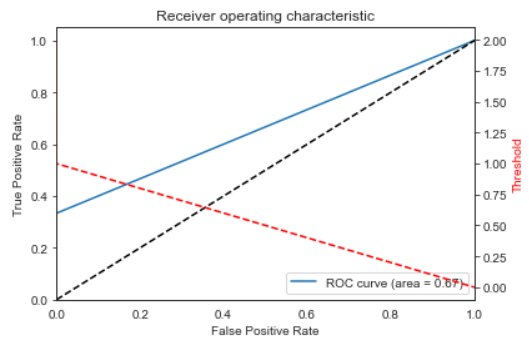
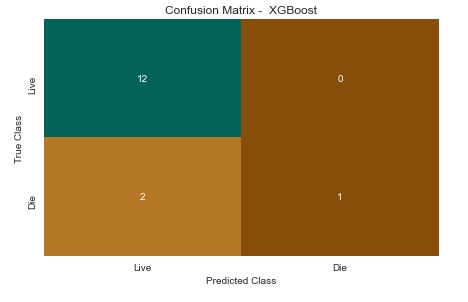
All three models were trained on the 19 retained predictors. After probability prediction, each model converted probabilities to hard labels with a common threshold of 0.25. That lower-than-usual cut-off was chosen so that potential deaths were favored over false reassurance in a clinical triage setting.

**EVALUATION METRICS**

To assess models, we used a variety of evaluation metrics. These include accuracy for overall correctness, sensitivity (recall) to catch true non‑survivors, specificity to avoid over‑triaging healthy patients, precision and NPV to gauge alert reliability, false positive/negative rates to quantify error costs, and AUC‑ROC for threshold‑independent discrimination. Sensitivity is given the greatest weight in interpretation, since failing to identify a patient who is likely to die represents a more serious clinical error than issuing an extra alert for a survivor. AUC-ROC is still valuable even though sensitivity is prioritized to capture the trade off at all thresholds, which is useful during model tuning.

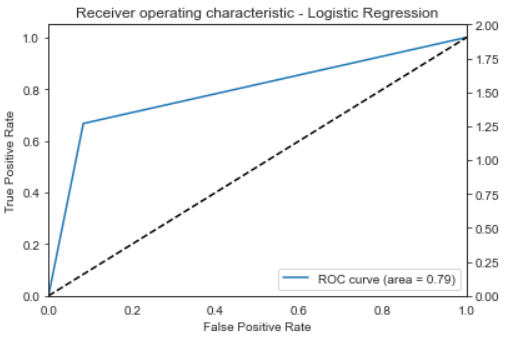
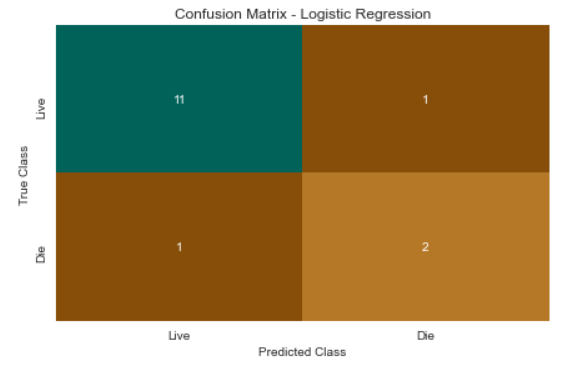
**RESULTS**

All three models therefore deliver the same headline accuracy, but they trade recall and false-alarm rates in different ways, an outcome that underscores the danger of using accuracy alone on imbalanced clinical data. Random Forest delivered the best overall performance with an AUC of 0.96, balancing sensitivity (66.7%) and specificity (91.7%). XGBoost was more conservative, with a specificity of 100%, but it sacrificed sensitivity (33.3%), missing two‑thirds of non-survivors. Logistic Regression matched Random Forest's sensitivity and specificity at the 25% threshold, offering simple interpretability but lower AUC (0.81). A 25% cutoff thus avoids missing sick patients while keeping extra work-ups manageable. Random Forest likely outperformed XGBoost due to the small dataset size and instability of boosting methods on the very small and imbalanced datasets. These models are compared in the charts below.

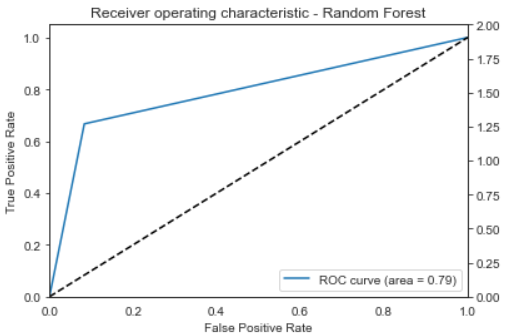
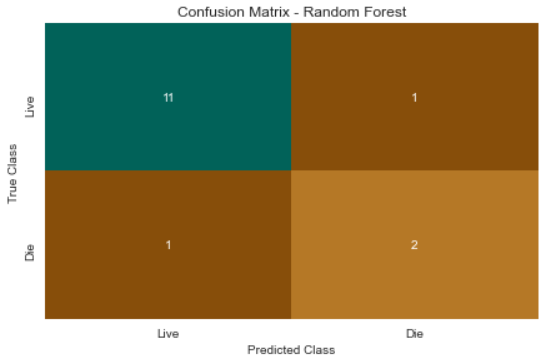


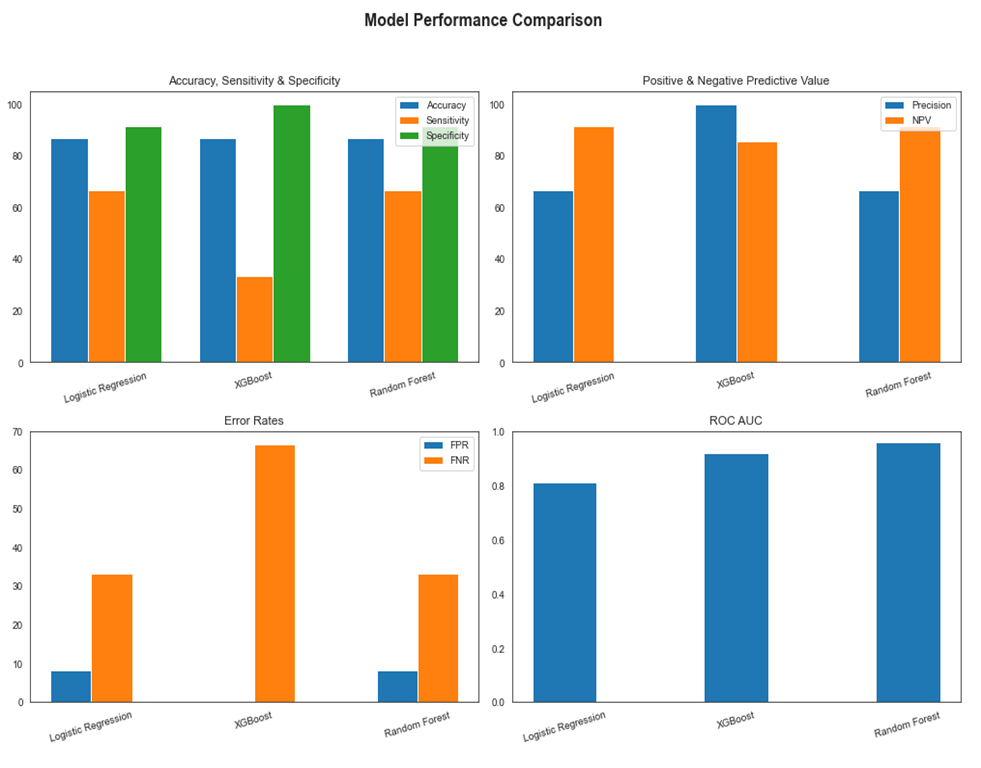
Confusion Matrix and AUC-ROC plot of XGBoost

Logistic Regression:

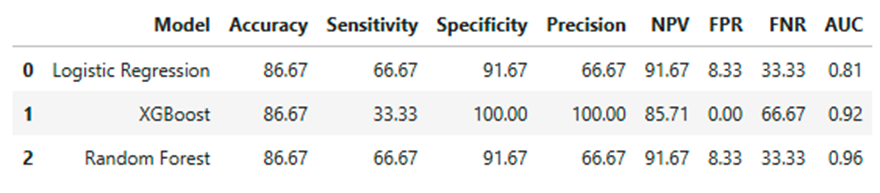


RANDOM FOREST





Charts of Model Performance Comparison



Comparison of Model Performance Metrics

**DISCUSSION**

The experiment demonstrates that careful class weighting in relatively simple models can outperform more sophisticated boosting when data are scarce and imbalanced. Logistic regression, despite its linear form, matches the random-forest ensemble in both discrimination and error profile, providing a highly interpretable alternative that clinicians can audit with coefficient inspection. XGBoost, by contrast, illustrates how a model that optimizes global AUC may still adopt a threshold-specific operating point that clinicians find unacceptable, in this case by sacrificing two-thirds of the high-risk patients to eliminate all false alarms. Threshold tuning is critical in clinical settings, where the cost of false negatives outweighs that of false positives. Future deployment would benefit from dynamic threshold adjustment based on clinician input and workflow needs.

**LIMITATIONS AND FUTURE WORK**

This dataset presented several limitations reducing its effectiveness and impact. The sample size was quite small, consisting of only 155 total patient records, with a test class of only 15 (10%). The medical records only came from a single hospital. The records are a single static snapshot of each patient’s health.

This opens the opportunity for future work in this area of hepatitis research. To address the small sample size, a larger multi-hospital dataset can be utilized. There are different machine learning methods that can be utilized to address the class imbalances found in the dataset. Finally, longitudinal health trends can be integrated into the study to enable a better understanding of patient trajectory over time.

**CONCLUSION**

This project compared three machine learning models: XGBoost, Logistic Regression, and Random Forest. All utilized class balanced loss to account for the class imbalance between patients who lived and patients who died. At the 25% threshold, logistic regression and random forest both caught 2 of 3 deaths, while flagging one healthy patient, while XGB flagged zero healthy patients, but only caught 1 of 3 deaths. This 25% threshold helps balance sensitivity versus workload by avoiding missing sick patients while limiting extra medical workups for healthy patients. Random forest and logistic regression models caught the most at-risk hepatitis patients with few false alarms. Overall, this machine learning model gives doctors an automated tool to improve early risk detection in hepatitis care and ultimately improve patient outcomes.

**REFERENCES**

"Hepatitis." UCI Machine Learning Repository, 1983, https://doi.org/10.24432/C5Q59J.

National Library of Medicine. “Hepatitis.” *MedlinePlus*, medlineplus.gov/hepatitis.html. Accessed 15 Apr. 2025.

Rosly, Rosaida et al. “Analyzing performance of classifiers for medical datasets.” International journal of engineering and technology 7 (2018): 136.

World Health Organization: WHO. Hepatitis. 11 Mar. 2020, www.who.int/health-topics/hepatitis. Accessed 15 Apr. 2025.