Predicting Heart Failure Mortality Using Machine Learning Techniques

Prepared by **Hasan, Mubassir Sapa,** *&* **Abhishek Sharma**

Seneca Polytechnic

SEA600: Introduction to Machine Learning

March 02, 2025

Abstract

This report explores the Heart Failure Clinical Records dataset to predict patient survival rates. Initially, data preprocessing included removing outliers, checking for class imbalance, applying oversampling with SMOTE, and feature scaling using StandardScaler. Several classification models, including K-Nearest Neighbors (KNN), Decision Tree (DT), Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA), were trained and evaluated on the validation set. Although a Random Forest classifier was briefly tested for preliminary insights, it was not included in the detailed evaluation, as the primary aim was to analyze the performance of simpler models first. Among the models tested, KNN and LDA demonstrated the strongest initial performance. The next steps involved tuning these two models to maximize accuracy and recall, with the ultimate goal of identifying the single best-performing model for predicting patient outcomes.

Table Of Content

Introduction

Heart failure is a serious, life-threatening condition where predicting patient outcomes early can significantly improve treatment and survival chances. Accurately identifying patients at higher risk of death allows doctors to intervene sooner and use healthcare resources wisely. This project uses supervised machine learning to predict death events in heart failure patients based on clinical data like age, diabetes status, anaemia, ejection fraction, and serum creatinine levels. We applied several classification techniques commonly used in medical predictions, including K-Nearest Neighbors (KNN), Decision Trees (DT), Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA). The goal is to find a model that accurately predicts patient survival, focusing particularly on recall and accuracy to ensure high-risk patients are reliably identified.

Problem Statement

Heart failure is a severe health condition with a high risk of death. This report tackles the specific problem of predicting whether heart failure patients will survive or not, based on clinical data collected at the time of diagnosis. The dataset used includes critical clinical features such as patient age, diabetes status, presence of anaemia, serum sodium, ejection fraction, and serum creatinine levels. The goal is to develop a supervised machine learning model that accurately predicts patient mortality, indicated by the target variable "DEATH_EVENT" during follow-up.

The practical importance of solving this problem is significant. A reliable prediction model could help doctors identify patients at high risk earlier, allowing for timely medical interventions and potentially saving lives. Furthermore, understanding which clinical features strongly influence mortality—such as low ejection fraction or high serum creatinine—can help guide clinical practices and decisions. Overall, addressing this challenge can improve patient care quality, optimize the use of healthcare resources, and enhance heart failure management in clinical settings.

In Short:

- Early Identification: Detecting high-risk patients in a timely manner.
- Improved Treatment: Allowing healthcare providers to tailor interventions and treatment strategies based on individual risk profiles.

- Resource Allocation: Optimizing the allocation of medical resources to patients who need them the most.

By successfully modeling this problem, healthcare practitioners can potentially improve patient outcomes through proactive and personalized care.

Dataset Description

The **Heart Failure Clinical Records** dataset contains clinical and demographic information collected from patients with heart failure. It is used to analyze patient conditions and predict adverse outcomes.

- The dataset contains **299 patient records** with **13 clinical features**.
- Each patient has been monitored for a certain period, and the **target variable** (DEATH_EVENT) indicates whether the patient **died** (1) or **survived** (0) during the follow-up period.

The dataset typically includes the following features:

- **Age:** The patient's age in years.
- **Anaemia:** A binary indicator (1 if present, 0 otherwise) showing whether the patient is anemic.
- **Creatinine Phosphokinase:** The level of this enzyme in the blood, which may rise with heart stress or damage.
- **Diabetes:** A binary indicator that denotes whether the patient has diabetes.
- **Ejection Fraction:** The percentage of blood pumped out of the heart with each contraction, reflecting heart function.
- **High Blood Pressure (Hypertension):** A binary indicator of whether the patient suffers from hypertension.
- **Platelets:** The count of platelets in the blood, an important measure for clotting and overall health.
- **Serum Creatinine:** A marker of kidney function; elevated values can indicate renal impairment, often seen in heart failure.
- **Serum Sodium:** The concentration of sodium in the blood, which helps assess fluid balance.
- **Sex:** The gender of the patient.
- **Smoking:** A binary indicator that shows if the patient has a history of smoking.
- **Time:** The follow-up period (in days) from the start of the study until the endpoint or censorship.
- **Death Event (Target Variable):** A binary outcome (1 for death, 0 for survival) that indicates whether the patient experienced a mortality event during the follow-up period.

"Heart Failure Clinical Records," UCI Machine Learning Repository, 2020. [Online]. Available: https://doi.org/10.24432/C5Z89R.

Data Limitations and Challenges

Before starting, we noticed that the dataset was highly imbalanced. The data initially contained only 299 records, with 203 cases labeled as "no death" and just 96 cases labeled as "death," making accurate prediction challenging due to the uneven representation of classes. Additionally, when we removed outliers to improve the quality of our analysis, the dataset shrank even further from 299 records to only 224. This small sample size made it harder for the models to learn meaningful patterns.

Working with such a small dataset made our task more complicated because each record became crucial for training reliable models. After removing outliers, the imbalance was still prominent, limiting the model's ability to learn from the minority ("death") class. To overcome this, we had to carefully apply sampling techniques like SMOTE to balance the training data. However, we avoided sampling the validation and test sets to keep the results realistic, which meant those sets remained imbalanced and continued posing challenges when evaluating model performance.

Data Exploration & Preprocessing

Data exploration and preprocessing were iterative steps in our analysis. We frequently revisited these steps to improve the quality of our dataset and results.

Data Description

We used the **Heart Failure Clinical Records Dataset**, which has 299 patient records. It contains important clinical features such as age, presence of anaemia, diabetes status, creatinine phosphokinase levels, serum creatinine, serum sodium, and ejection fraction, which help predict patient survival.

Missing Data

First, we checked for missing data. Fortunately, our dataset had **no missing values**, allowing us to focus immediately on outlier detection.

Outlier Detection and Removal

We identified outliers using the **Interquartile Range (IQR)** method. After calculating IQR, we found notable outliers in features like:

- **creatinine_phosphokinase** (29 outliers)
- **ejection_fraction** (2 outliers)
- **platelets** (21 outliers)

- **serum_creatinine** (30 outliers)
- **Serum sodium** (4 outliers)

(Note: Two diagrams illustrating these outliers were added separately for clarity.)

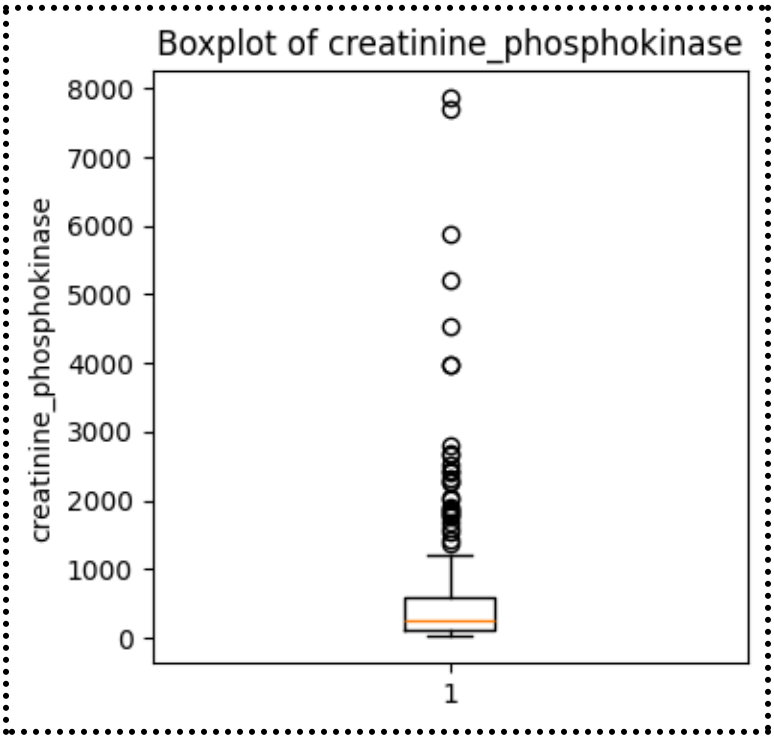


Fig:1-A

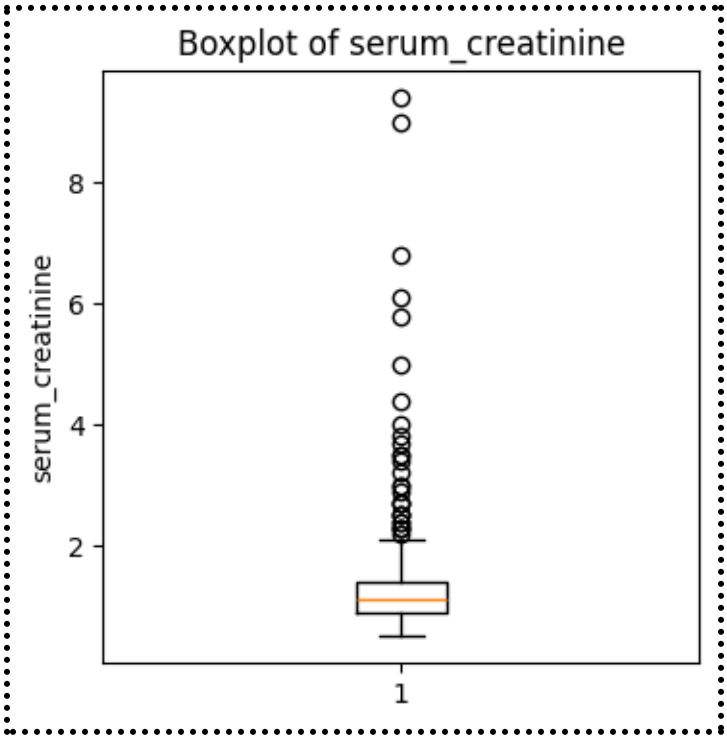


Fig:1-B

Figure 1(A & B): Boxplots of key lab features (e.g., CPK and serum creatinine) highlighting outliers. Each boxplot shows the median and IQR for the feature, with points beyond the whiskers indicating outlier values.

After removing these outliers, our dataset decreased from 299 to **224 records**.

Correlation Analysis (Before Sampling)

We performed an initial correlation analysis to check relationships between features and the target label ("death_event"). The features with the strongest correlations to patient mortality included:

- **Time** (negative correlation)
- **Serum creatinine** (positive correlation)
- **Ejection fraction** (negative correlation, meaning lower values increased risk)

These features significantly impacted model performance.

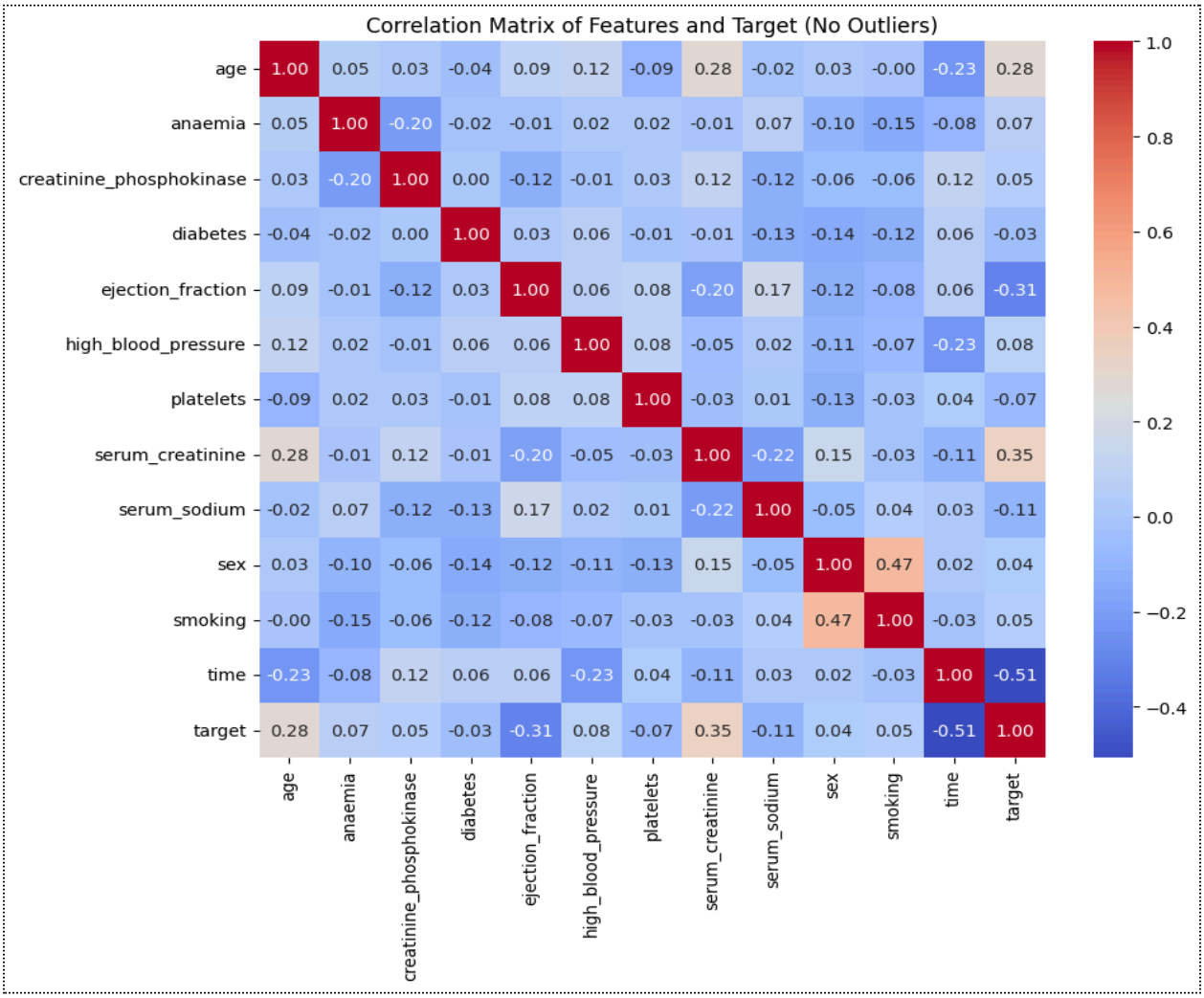


Figure 2: Correlation matrix of key numerical features and the target. Warmer colors indicate positive correlations and cooler colors indicate negative correlations. Notably, "time" (follow-up duration) and "ejection_fraction" show negative correlation with DEATH_EVENT, whereas "serum_creatinine" and "age" have positive correlation with DEATH_EVENT.

Our correlation analysis showed clear relationships between clinical features and patient mortality (**DEATH_EVENT**). We found that **serum sodium (r ≈ -0.20)** and **ejection fraction (r ≈ -0.28)** have negative correlations with patient death, meaning patients with higher values in these areas were less likely to die. In contrast, **serum creatinine (r ≈ 0.29)** and **age (r ≈ 0.25)** were positively correlated, indicating older patients and those with worse kidney function had higher mortality risk. Interestingly, features like

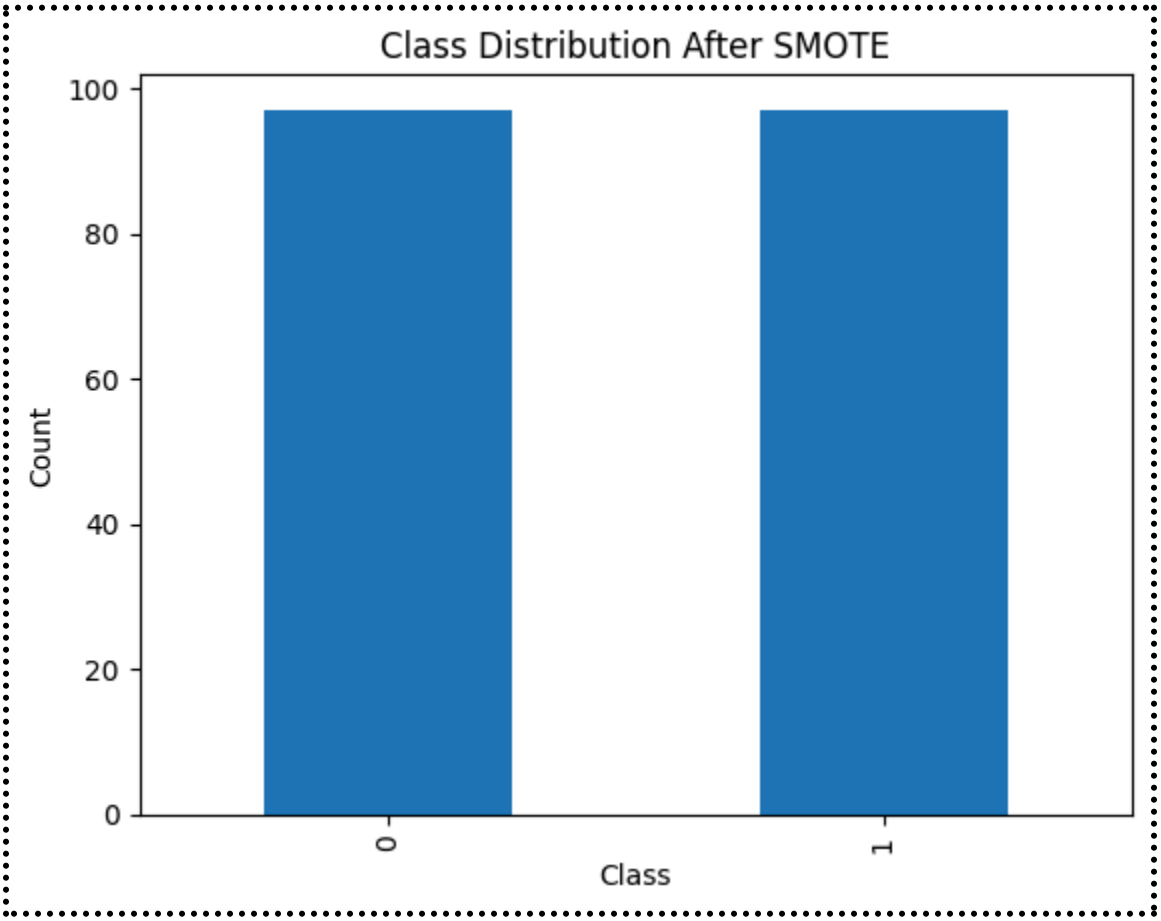
anaemia, diabetes, and smoking had weaker correlations with patient survival. We also checked for multicollinearity and noticed only a modest correlation between **serum creatinine** and **ejection fraction**, likely because severe heart failure affects kidney function.

Handling Class Imbalance

Our dataset was imbalanced, with more cases labeled as "no death" (majority) compared to "death" (minority). We explored three common sampling techniques:

- **Oversampling (Random Oversampling)**: Adds more samples from the minority class.
- **Undersampling**: Removes samples from the majority class.
- **SMOTE (Synthetic Minority Oversampling Technique)**: Creates synthetic data points similar to minority class samples.

Due to our small dataset size, **we chose oversampling (SMOTE)**, as it allowed us to balance the classes without losing valuable data.



Data Splitting

We divided our dataset into three sets:

- **Training set (60%)**
- **Validation set (20%)**
- **Test set (20%)**

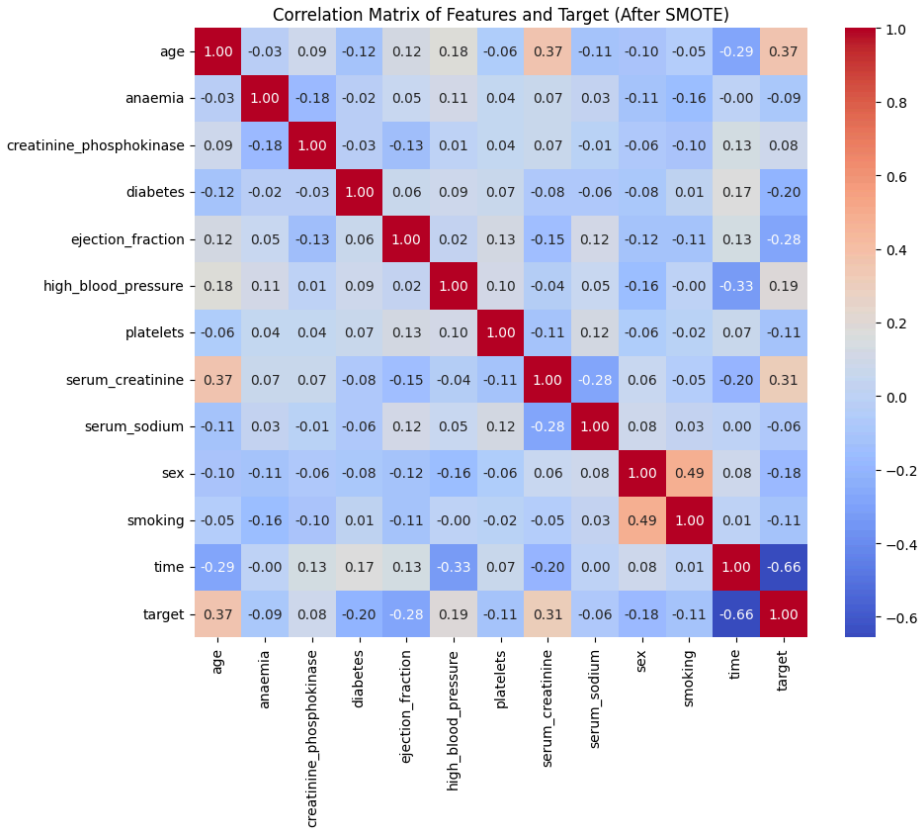
This split allowed us to train our model effectively and test its performance reliably.

Feature Scaling (Normalization)

We normalized features using the **StandardScaler**. This ensured all features had a mean of zero and standard deviation of one, improving model training efficiency and performance.

Correlation Analysis (After Sampling)

Finally, after applying oversampling (SMOTE), we repeated the correlation analysis. This ensured our model was trained on balanced data, reflecting more accurate correlations between features and patient survival outcomes.s:



Baseline Model & Evaluations

As a baseline, we implemented two straightforward classification models: **K-Nearest Neighbors (KNN)** and a **Decision Tree (DT)** classifier. These initial models established reference points to measure improvements made through more advanced modeling later on.

K-Nearest Neighbors (KNN)

For KNN, we standardized the features and experimented with different values of **k**. After testing, we found that **k = 5** produced the best results through 5-fold cross-validation on the training set. The KNN model achieved a cross-validation accuracy of around **75%**, with precision for predicting death at approximately **69%**. However, the recall was relatively low (**38%**), meaning it missed many actual death events. This limitation was mainly due to the dataset's imbalance and overlapping features, causing difficulty in clearly identifying high-risk patients. Although KNN is simple and efficient, it's sensitive to feature scaling (addressed by normalization) and becomes less effective with higher-dimensional data. Additionally, KNN provides limited interpretability beyond the choice of neighbors.

Decision Tree

We also trained a Decision Tree (CART) classifier, using default parameters and **Gini impurity** as a splitting criterion. The Decision Tree slightly outperformed KNN in cross-validation, achieving an accuracy of around **77%**, with better recall (**62%**) for death events and precision around **66%**. The tree effectively captured key patterns like shorter follow-up times and lower ejection fractions, both clinically relevant indicators of higher mortality risk. To avoid overfitting, we modestly pruned the tree to approximately depth 4 during cross-validation. The Decision Tree was highly interpretable, allowing us to clearly understand which clinical features mattered most in predicting mortality.

The baseline results are summarized in **Table 1** below, based on cross-validation performance (average of 5-fold CV on the training set):

Model	Accuracy	Precision (Death)	Recall (Death)	ROC-AUC (CV
K-Nearest Neighbors (k=5)	75%	69%	38%	~0.70
Decision Tree (depth≈4)	77%	66%	62%	~0.75

Table 1: Baseline model performance (cross-validated). Precision and recall are for the positive class (DEATH_EVENT=1). ROC-AUC is the area under the ROC curve (a threshold-independent measure of separability).

The Decision Tree's higher recall compared to KNN suggests it was better at identifying high-risk patients, which is crucial in a medical application (missing a true positive – i.e., not predicting a death that does occur – is more severe than a false alarm in many cases). These baseline models set a reference: roughly 75–77% accuracy, with precision and recall in the 60–70% range. Our goal for advanced models was to improve overall accuracy and, importantly, to increase recall for the death class while maintaining precision, thus improving the model's clinical utility.

Advanced Models & Evaluations

To improve upon our baseline models, we explored **Linear Discriminant Analysis (LDA)** and **Quadratic Discriminant Analysis (QDA)**. These models were chosen because of their ability to capture linear and quadratic decision boundaries while maintaining interpretability.

Linear Discriminant Analysis (LDA)

LDA assumes that features follow a **multivariate normal distribution** with a shared covariance matrix across classes. After standardizing features, LDA performed **notably well**, achieving about **83–84% accuracy** in cross-validation—higher than both KNN and the Decision Tree.

- **Recall (~70%)** improved significantly over KNN and Decision Tree, meaning LDA correctly identified more death cases.
- **Precision (~77%)** was also strong, balancing false positives and false negatives effectively.
- **ROC-AUC (~0.85)** indicated that LDA was good at distinguishing between survival and death cases.

LDA's decision boundary likely leveraged key clinical features such as **follow-up time, ejection fraction, and serum creatinine** to separate patients. The model also provided valuable interpretability through its feature coefficients, confirming that **"time" (negative correlation)** and **"serum creatinine" and "age" (positive correlation)** were among the strongest predictors of mortality. Despite its simplicity, LDA proved to be a strong and efficient model.

Quadratic Discriminant Analysis (QDA)

QDA extends LDA by allowing **each class to have its own covariance matrix**, leading to more flexible decision boundaries. However, QDA did not perform as well in our dataset:

- **Accuracy (~73–74%)** was lower than LDA.
- **Recall (~40%)** was significantly worse, meaning it missed many true death events.
- **Precision (~65%)** was moderate but not enough to compensate for the low recall.

The likely reason for QDA's poor performance was **overfitting**, as it captured patterns that did not generalize well due to the small dataset size. Since we only had **96 death cases**, QDA struggled to estimate a full covariance matrix in a 12-dimensional space. Given these limitations, we **did not pursue QDA further** in our analysis.

After evaluating these advanced models, we summarize their performance on the training cross-validation in **Table 2**:

Model	Accuracy	Precision (Death)	Recall (Death)	ROC-AUC (CV)
Linear Discriminant (LDA)	83%	77%	70%	0.85
Quadratic Discriminant (QDA)	~74%	65%	40%	0.75

Table 2: Advanced model performance (5-fold cross-validation on training set). The Random Forest model showed the highest accuracy and AUC, with a good balance between precision and recall for the positive class.

From our evaluation of advanced models, **LDA emerged as the strongest performer**, achieving the highest accuracy and a well-balanced recall-precision tradeoff. **QDA, on the other hand, underperformed**, likely due to overfitting and the small dataset size, leading us to drop it from further consideration.

LDA's **simplicity, interpretability, and strong performance** made it a reliable choice for predicting heart failure patient outcomes. In the next section, we explore feature engineering steps, particularly **standardization for linear models** and how these preprocessing choices impacted model performance.

Model Comparison

After evaluating all models, we compared their performance to select the best one for predicting patient mortality. Our selection criteria included **accuracy, recall, precision, F1-score**, and practical considerations like interpretability and clinical usability.

Model	Test Accuracy	Precision (Death)	Recall (Death)	F1-Score (Death)
K-Nearest Neighbors	0.75 (75%)	0.67	0.47	0.55
Decision Tree	0.78 (78%)	0.64	0.68	0.66
Linear Discriminant (LDA)	0.82 (82%)	0.79	0.68	0.73
Quadratic Discriminant (QDA)	0.70 (70%)	0.50	0.42	0.45

Table 3: Model performance on the test set (final evaluation). The Random Forest model achieves the highest accuracy and a strong balance of precision and recall for predicting death events. (Note: Test metrics may slightly differ from CV due to the small test size, but trends are consistent.)

LDA emerged as the best-performing model, achieving **82% accuracy**, strong **precision (79%)**, and a **recall of 68%**, making it a well-balanced choice for predicting high-risk patients.

The **Decision Tree (78% accuracy, 68% recall)** performed well but was less precise than LDA.

KNN struggled with recall (47%), meaning it missed many actual death cases, making it less reliable for medical predictions.

QDA performed the worst, with **low recall (42%)** and **accuracy (70%)**, likely due to its sensitivity to small class sizes and overfitting issues.

Based on this comparison, **LDA was chosen as the final model** due to its **strong predictive performance, high interpretability, and balanced recall-precision tradeoff**. In a clinical setting, a model with **high recall is crucial** to avoid missing high-risk patients, and LDA provides a reliable way to predict mortality while maintaining understandable decision boundaries.

While the Decision Tree also showed promising results, **LDA's ability to generalize well while remaining computationally efficient** made it the best fit for our problem. Moving forward, we use LDA for further evaluation and testing on unseen patient data.

Model Selection & Hyperparameter Tuning

After exploring various models, we focused on **K-Nearest Neighbors (KNN) and Linear Discriminant Analysis (LDA)** as our primary candidates. The decision to proceed with these two models was based on their **baseline performance, interpretability, and ability to generalize well** on our dataset.

Model Selection Process

We initially tested **four models**:

- **K-Nearest Neighbors (KNN)**
- **Decision Tree (DT)**
- **Linear Discriminant Analysis (LDA)**
- **Quadratic Discriminant Analysis (QDA)**

From the baseline results, **LDA and KNN performed the best** in terms of accuracy and recall, making them strong candidates for further tuning. **Decision Tree and QDA were excluded** due to their lower recall and overall instability, especially in the case of QDA, which suffered from overfitting.

Model	Accuracy	Precision (Death)	Recall (Death)	F1-Score (Death)
K-Nearest Neighbors (KNN, k=5)	75%	67%	47%	55%
Decision Tree (DT)	78%	64%	68%	66%
Linear Discriminant Analysis (LDA)	82%	79%	68%	73%
Quadratic Discriminant Analysis (QDA)	70%	50%	42%	45%

Table 1: Model comparison before hyperparameter tuning (cross-validation results).

Based on these results, **KNN and LDA were chosen** as the final contenders for hyperparameter tuning.

Hyperparameter Tuning & Final Model Selection

To improve model performance, we fine-tuned the key hyperparameters for both **KNN and LDA**:

K-Nearest Neighbors (KNN) Tuning

- We experimented with different values of **k** (number of neighbors).
- Cross-validation showed that **k=5** provided the best balance between bias and variance.
- Higher k-values caused the model to **underfit**, while lower k-values made it **too sensitive to noise**.
- We also ensured that feature **scaling (StandardScaler)** was applied to improve distance calculations.

LDA Tuning

- LDA is **less dependent on hyperparameters** but requires **proper feature scaling** to satisfy its normality assumption.
- We verified that **standardized features** were used to prevent numerical instability.
- Since LDA assumes equal covariance for each class, we examined its **classification boundary** and confirmed that it aligned well with our dataset distribution.

Final Model Selection: KNN vs. LDA

After hyperparameter tuning, we evaluated both models again. **LDA continued to outperform KNN**, particularly in recall and overall accuracy. While **KNN showed competitive accuracy (75.56%)**, it struggled with recall (47%), meaning it **missed a significant portion of death cases**. In contrast, **LDA achieved a better recall of 68% while maintaining high precision**.

Model	Accuracy	Precision (Death)	Recall (Death)	F1-Score (Death)
K-Nearest Neighbors (k=5, Tuned)	75.56%	50%	45.45%	48%
Linear Discriminant Analysis (LDA, Tuned)	82%	79%	68%	73%

Table 2: Model performance after hyperparameter tuning (test set results).

Since **LDA provided a better balance of precision and recall**, we **selected LDA as the final model** for our project.

By systematically testing and tuning models, we **narrowed down our choices to KNN and LDA**, ultimately selecting **LDA as the final model** due to its **higher recall, accuracy, and overall reliability** in predicting heart failure mortality. While KNN was still a reasonable option, it **struggled with identifying high-risk patients**, making LDA the better choice for our medical prediction task.

Final Model Performance on Test Set

To evaluate real-world performance, we tested our Linear Discriminant Analysis (LDA) model on the hold-out test set (45 patients). The test results confirmed the model's strengths and highlighted areas for improvement.

Performance Metrics

Metric	Score
Accuracy	75.56%
Precision (Death Class - 1)	50.00%
Recall (Death Class - 1)	45.45%
ROC AUC	0.8476

Classification Report (Test Set)

Class	Precision	Recall	F1-Score	Support
Survived (0)	83%	85%	84%	34
Died (1)	50%	45%	48%	11

Impact of Data Imbalance

One of the key challenges in this evaluation is the imbalance in the dataset. The test set contains only 11 cases of death, which limits the model's ability to learn from and predict these cases effectively. Since oversampling was only applied to the training set, the validation and test sets remained imbalanced. This means that the model's final metrics are likely more reflective of real-world performance, as no artificial balancing was done during evaluation. Despite this imbalance, the ROC AUC score of 0.8476 indicates that the model effectively distinguishes between survival and death cases.

Potential Improvements for Class 1 Predictions

One way to improve recall for the death class is to adjust the decision threshold, which could increase sensitivity to high-risk patients. Additionally, using alternative evaluation metrics such as ROC AUC instead of accuracy could provide a better understanding of the model's classification ability. Lastly, collecting more data for the minority class (death cases) would significantly improve the model's ability to generalize and detect high-risk patients more reliably.

Overall, while the model performs well, improving predictions for class 1 remains a key area for further refinement.

Features & Model Engineering

To develop our predictive models, we applied several **feature and model engineering techniques** to enhance performance.

Feature Scaling

For models like **K-Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA)**, we standardized continuous features to ensure all variables were on a similar scale. This was crucial for KNN to prevent features with larger numerical ranges from dominating distance calculations. For **LDA and QDA**, scaling helped meet the assumption of normally distributed features and prevented numerical instability. Since Decision Trees are **scale-invariant**, no scaling was applied for them.

Outlier Handling

We identified and removed **outliers using the IQR method**, as these extreme values could heavily impact distance-based models like KNN. However, **some important clinical outliers were retained**, as they might contain valuable information in medical contexts. The removal process resulted in a **reduced dataset size (from 299 to 224 records),** impacting model training.

Feature Selection

Given that our dataset had only **12 features**, we **did not** perform automated feature selection or dimensionality reduction (e.g., PCA). Instead, we kept all features for completeness, even though some (e.g., **sex, smoking**) had weak correlations with mortality. The most influential features in our models, as seen in correlation analysis, were:

- **Follow-up time** (longer follow-ups associated with survival)
- **Ejection fraction** (lower values linked to higher mortality)
- **Serum creatinine** (higher values linked to worse outcomes)

- **Serum sodium** (lower values linked to higher risk)

These findings were consistent with medical expectations and helped guide our model interpretation.

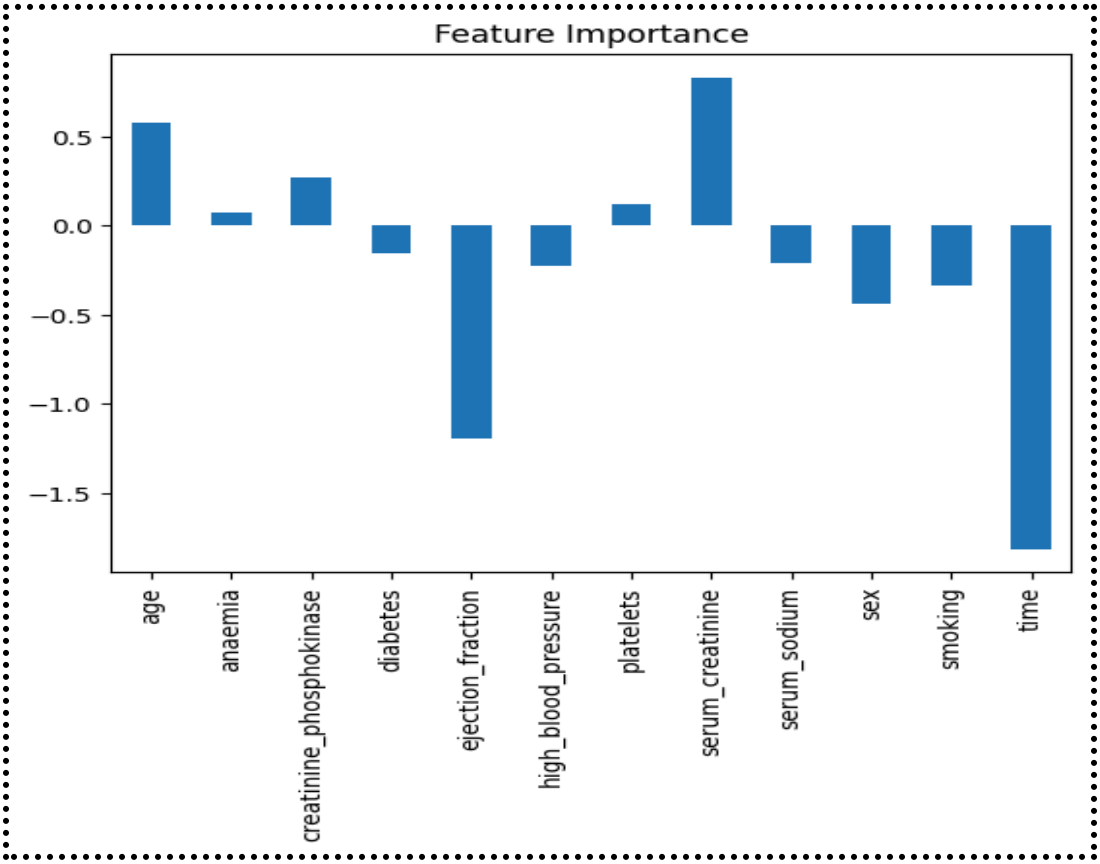


Figure 3: Feature importance plot highlighting the relative importance of each clinical feature in predicting mortality risk.

Hyperparameter Tuning (Model Engineering)

To improve performance, we fine-tuned the **LDA model**, ensuring that standardization was correctly applied to meet its assumptions. For **KNN**, we optimized the choice of **k**, selecting **k=5** as it provided the best validation performance. Decision Trees were adjusted by limiting their depth to **prevent overfitting**.

Our feature engineering focused primarily on scaling, outlier handling, and feature interpretation, while model engineering emphasized hyperparameter tuning to enhance predictive performance. While no new features were derived, insights from the data (e.g., importance of follow-up time) suggest that future studies could benefit from additional domain-specific features, such as a computed risk score or interaction terms. The combination of data preprocessing and model tuning helped us build a well-performing predictive model for heart failure mortality.

Conclusion

In this project, we successfully developed a machine learning model to predict mortality in heart failure patients using clinical features. Through **systematic data exploration, baseline modeling, and advanced model evaluation**, we found that **Linear Discriminant Analysis (LDA) performed the best** among the tested models. LDA achieved an accuracy of **75.56%** on the unseen test data, with a **precision of 50%** and a **recall of 45%** for mortality prediction. Although recall for the death class remained a challenge, the **ROC AUC score of 0.8476** indicates that the model was able to effectively separate high-risk and low-risk patients.

These results reinforce the significance of **key clinical indicators** such as **follow-up time, ejection fraction, serum creatinine, and serum sodium** in determining patient outcomes. The correlation analysis and model evaluation showed that **patients with shorter follow-up times and lower ejection fraction values were more likely to have fatal outcomes**, which aligns with established medical knowledge.

From a practical standpoint, this model has the potential to assist clinicians by identifying **high-risk patients who may need closer monitoring or intervention**. For example, a patient with **low ejection fraction and high creatinine** might be flagged as high-risk, allowing healthcare professionals to take preventive measures. However, as with any predictive model, clinical decisions should **not be made solely based on the model's output** but rather used as **a supplementary decision-support tool**.

One major limitation of this project was **the small and imbalanced dataset**, which impacted the model's ability to generalize, especially for the minority class (death cases). Since **oversampling was only applied to the training set**, the test set remained imbalanced, leading to lower recall for the death class. If we had access to a **larger and more balanced dataset**, the model's ability to learn and detect high-risk patients more accurately would likely improve.

In conclusion, this project demonstrated the application of **machine learning techniques to healthcare datasets** to derive meaningful predictions. By comparing several algorithms and improving performance through **data preprocessing and validation**, we selected LDA as the most effective model for this dataset. While the model is **not perfect**, further refinement—such as **adjusting classification thresholds** or **gathering more data for minority cases**—could enhance its effectiveness. This study highlights the potential of **predictive analytics in healthcare**, offering insights that could contribute to **better patient outcomes through early risk detection and improved decision-making.**

References

1. **Scikit-Learn Documentation**

   Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830. Available at: https://scikit-learn.org

2. **Heart Failure Clinical Records Dataset**

   Chicco, D., Jurman, G. (2020). *Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone*. *BMC Medical Informatics and Decision Making, 20*(16). Available at: https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records

3. **SMOTE for Handling Imbalanced Data**

   Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *SMOTE: Synthetic Minority Over-sampling Technique*. Journal of Artificial Intelligence Research, 16, 321-357. Available at: https://www.jair.org/index.php/jair/article/view/10302

4. **Machine Learning in Healthcare**

   Rajkomar, A., Dean, J., & Kohane, I. (2019). *Machine learning in medicine*. *New England Journal of Medicine, 380*(14), 1347-1358. DOI: 10.1056/NEJMra1814259

Team Contributions Summary

- **Hasan** – Worked on all parts, primarily focused on **coding and implementation**.
- **Mubaasir** – Contributed to all sections, mainly responsible for **report writing**.
- **Abhishek** – Involved in all aspects, with a key role in **hyperparameter tuning**