

Heart Failure Survival Predictions

Sophie Lin

UCLA Department of Atmospheric and Oceanic Sciences

December 2025

1 Introduction

Heart failure is a chronic cardiovascular condition in which the heart is unable to sufficiently pump enough blood to supply the body, causing millions of deaths every year. It is typically associated with having diabetes, high blood pressure, or other heart conditions. Because of the many variables involved with this complex condition, predicting death from heart failure is difficult and requires vast data analysis. Recently, machine learning (ML) has been able to provide faster automated predictions for detecting relationships between these clinical features and patient outcomes. As the availability of electronic health records (EHRs) continues to grow, utilizing these computational approaches for predictions has become increasingly important to assist clinical decision making.

In this project, supervised classification machine learning models were applied to predict patient survival outcomes following a heart failure event given their clinical data from EHRs. Specifically, the ejection fraction and serum creatinine levels alone have been previously shown through feature ranking to be capable of predicting the patient's death event [1]. Five different types of ML models were assessed, namely logistic regression, support vector machine (SVM), random forest, decision tree, and multilayer perceptron (MLP), to examine which model would perform best in its ability to predict patient survival using only those two features.

2 Data

The dataset, "Heart Failure Clinical Records", used for this project contained medical data collected from 299 patients during their follow-up period and was obtained from the UC Irvine Machine Learning Repository [2]. Each patient record contained 13 clinical features commonly used in clinical evaluations of heart failure, listed in Table 1. These features included both demographic information and physiological measurements.

Variable	Description
Age	Age of patient
Anaemia	Decrease of hemoglobin or red blood cells
Creatinine phosphokinase	Level of the CPK enzyme in the blood
Diabetes	Whether or not patient had diabetes
Ejection fraction	Percentage of blood leaving the heart at each contraction
High blood pressure	Whether or not patient has hypertension
Platelets	Platelets in the blood
Serum creatinine	Level of serum creatinine in the blood
Serum sodium	Level of serum sodium in the blood
Sex	Woman or man
Smoking	Whether or not patient smokes
Time	Follow-up period
Death event	Whether or not patient died during the follow-up period

Table 1: Variables and their descriptions provided in the “Heart Failure Clinical Records” dataset

The dataset was read and stored as a PANDAS dataframe. Then, the data was visualized to examine the correlations between the various features and survival. Initial plots, such as pairwise scatterplots, boxplots, and histograms, were generated to identify which clinical measurements had stronger variability or noticeable trends that were interesting to further explore. It was noted that a previous study identified serum creatinine levels and ejection fraction as two key features that when used alone was enough to determine patient survival outcomes. The researchers of that study used feature ranking analysis to obtain that result [1]. This prior finding was considered when observing the plots to see whether similar patterns appeared.

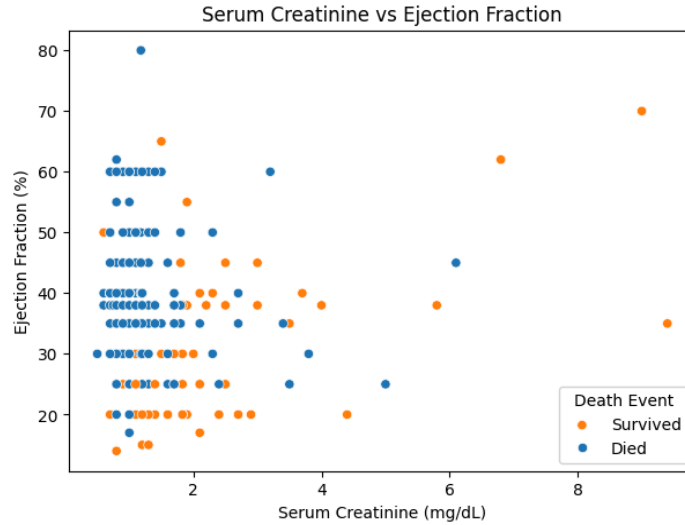


Figure 1: Scatter plot of the serum creatinine levels and ejection fraction of patients who survived (orange) vs died (blue)

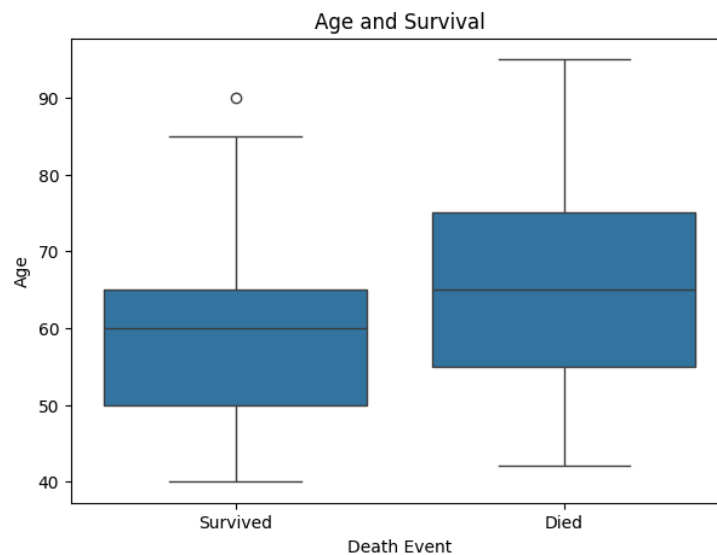


Figure 2: Boxplots comparing the age distribution of patients who survived (left) vs died (right)

Interestingly, the majority of the patients who died during their follow-up period clustered around having low serum creatinine levels and low to moderate ejection fraction percentage, as shown in the scatterplot of Figure 1. This showed the importance of the relationship between these two features compared to other features, such as age which had a weak association with survival based on the overlapping box plot distributions in Figure 2. Therefore, the variables this project focused on were ejection fraction and serum creatinine with the death event as the target variable for the machine learning models. The dataframe was filtered to have only ejection fraction and serum creatinine as the training features and the death

event as the target. For the death event, 0 represented survival whereas 1 represented death. This dataframe was then split for 80% of the data to be used as training and the other 20% as testing.

3 Models

The primary goal of the project was to evaluate the performance of various machine learning models to determine the best predictor of patient survival outcome. Supervised classification learning was used since the dataset contained binary survival outcomes as the target, allowing the models to learn the relationship between the input features and the target variable. The ML models tested were logistic regression, support vector machine, random forest, decision tree, and multilayer perceptron. These were all models known for being used in classification tasks. Using a scikit-learn Pipeline, the training features were normalized using a standard scaler before each model fitting to ensure consistent feature scales and enhance model performance.

3.1 Logistic Regression

Logistic regression is a linear model commonly used for binary classification, especially for small datasets, and therefore, the expectations for this model to perform well given the heart failure clinical record dataset was high. It works by modeling the probability that an input belongs to a particular class using the logistic or sigmoid function, which maps any input into a value between 0 and 1 to classify them. From the scikit-learn library, the logistic regression model was built using the LogisticRegression package. Because logistic regression relies on iterative optimization to estimate the model coefficients, the maximum iterations parameter, `max_iter`, was set to 2000 to ensure proper convergence of the solver.

3.2 Support Vector Machine

Support vector machines (SVM) classify data by finding the optimal hyperplane that separates classes in feature space by maximizing the distance between the hyperplane and the support vectors, or training points. SVMs utilize kernel functions, such as linear, polynomial, and radial basis function (RBF), to map the data. In this analysis, the SVM model was created using the SVC class from the scikit-learn library. The model was tested using default parameters, which incorporates the RBF kernel that measures the similarity score between two data points by calculating their Euclidean distance. This is often the default kernel for SVM algorithms since it can be applied to a wide range of problems.

3.3 Random Forest

Random forest is an ensemble method that builds multiple decision trees and aggregates their predictions with a technique called bootstrap aggregation (bagging). It reduces overfitting and often achieves high accuracy for classification tasks. For this project, the model was built with the RandomForestClassifier from scikit-learn. However, when building and testing this model with 200 estimators for the small dataset size, it had a relatively poor performance with an accuracy of 70% in comparison with the first two types of models created. To obtain a more reliable model, k-fold cross-validation was applied to evaluate multiple random forest configurations across different training splits ranging from 1 to 5. K-fold cross-validation is a method used to split the training dataset into subsets known as folds to train the model, helping improve accuracy by ensuring the model is able to generalize to new data. At the end, the best model is selected and saved.

K-fold	Accuracy
1	0.792
2	0.833
3	0.646
4	0.729
5	0.702

Table 2: K-fold cross-validation accuracy results

After utilizing the k-fold cross-validation for the random forest, the best model from the second fold was saved as it had an accuracy of 83.3% seen in Table 2. This model with the highest validation accuracy across the folds was then selected as the final classifier and used for testing on the dataset.

3.4 Decision Tree

Decision trees split the feature space into regions based on feature thresholds, but they may overfit the data if not pruned. The decision tree model was implemented in this project using scikit-learn's DecisionTreeClassifier, which allowed for tuning of parameters such as maximum depth, minimum samples per split, and splitting criteria. To reduce the risk of overfitting on the small dataset size of 299, a parameter grid search on the maximum depth was conducted, testing the accuracy of the decision tree model using varying values of max depth ranging from 1 to 5.

Max depth	Accuracy
1	0.750
2	0.817
3	0.783
4	0.767
5	0.733

Table 3: Max depth grid search accuracy results

As shown in Table 3, the parameter grid search resulted in a max depth of 2, yielding an accuracy of 81.7%, higher than any other tested depth. This was likely due to the small size of the dataset as trees with more depth would tend to memorize the samples instead of learning generalizable decisions. Therefore, the selected max depth of 2 was the parameter kept for the final decision tree model training and fitting.

3.5 Multilayer Perceptron

The multilayer perceptron (MLP) is a type of neural network that is composed of an input layer, one or more hidden layers, and an output layer, where each layer consists of multiple interconnected neurons and requires feature scaling for optimal performance. This type of model uses a backpropagation algorithm to minimize the loss function. The MLP model in this project was implemented using scikit-learn's package called MLPClassifier, and multiple configurations of the hidden layer size were tested to examine which model architecture would result in the best performance. In the end, a single hidden layer with 150 neurons, represented by the notation (150,) in the parameters, achieved the highest accuracy.

4 Results

When comparing the results, the MLP and decision tree models had the highest accuracy of 81.7% in comparison with the other models. In addition to accuracy scores, each model's sensitivity, the true positive rate, and specificity, the true negative rate, were also calculated to evaluate each model's efficiency in determining true survival outcomes versus false classifications. The highest sensitivity was achieved by the decision tree model, resulting in a score of 76.2%. There were three models that yielded the highest specificity score of 92.3% among the rest, namely the logistic regression, SVM, and MLP models. All of the model performance statistics for accuracy, sensitivity, and specificity are provided in Table 4.

Model	Accuracy	Sensitivity	Specificity
Logistic Regression	0.733	0.381	0.923
SVM	0.767	0.476	0.923
Random Forest	0.700	0.619	0.744
Random with Cross-Validation	0.767	0.667	0.821
Decision Tree	0.817	0.762	0.846
MLP	0.817	0.619	0.923

Table 4: Summary statistics for every model tested in the project

The confusion matrices plotted below in Figure 3 provide a visual representation of the sensitivity and specificity scores, revealing the exact number of true positives, false positives, true negatives, and false negatives classified among each model. The logistic regression, SVM, and MLP models had the highest amount of true positives, while the decision tree model had the highest amount of true negatives identified.

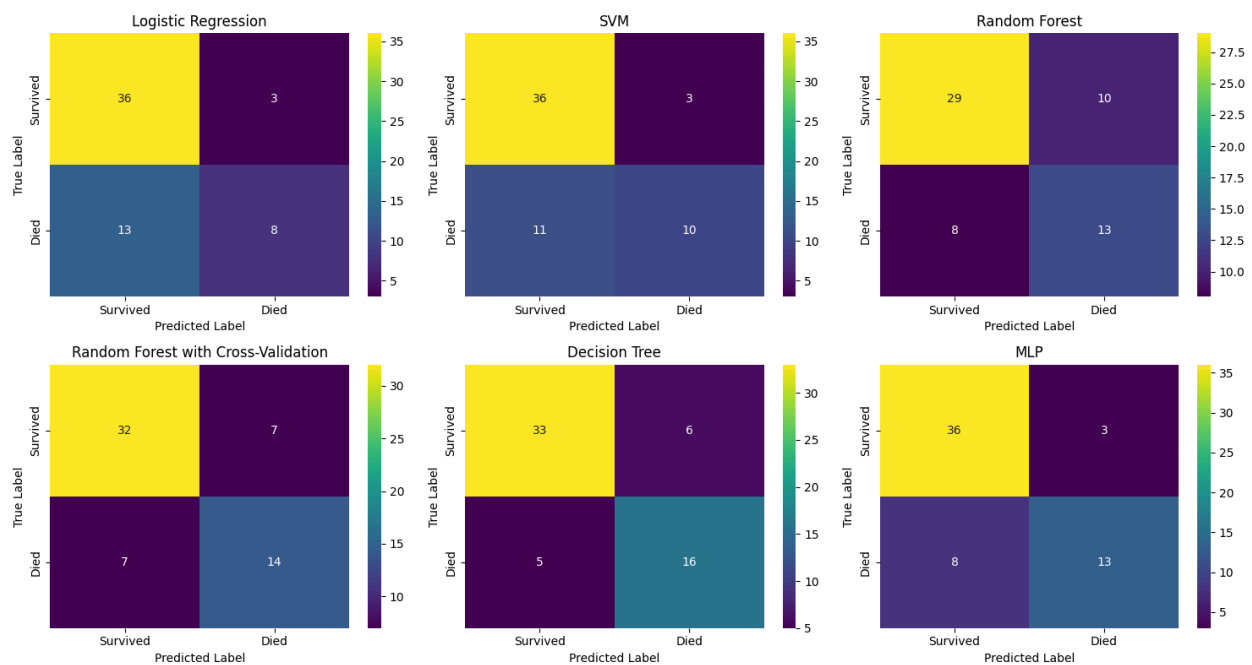


Figure 3: Grid of confusion matrices for every model

ROC curves and area under the curve (AUC) scores were also generated for each model to evaluate their ability to distinguish between the two survival outcomes. The curves observed in Figure 4 do not exhibit smoothness and are rather very jagged in shape since the dataset used

for machine learning was small. The standard random forest model without the k-fold cross-validation produced the largest AUC score of 0.83, but the random forest model with cross-validation, decision tree model, and MLP model achieved a very similar AUC score of 0.82.

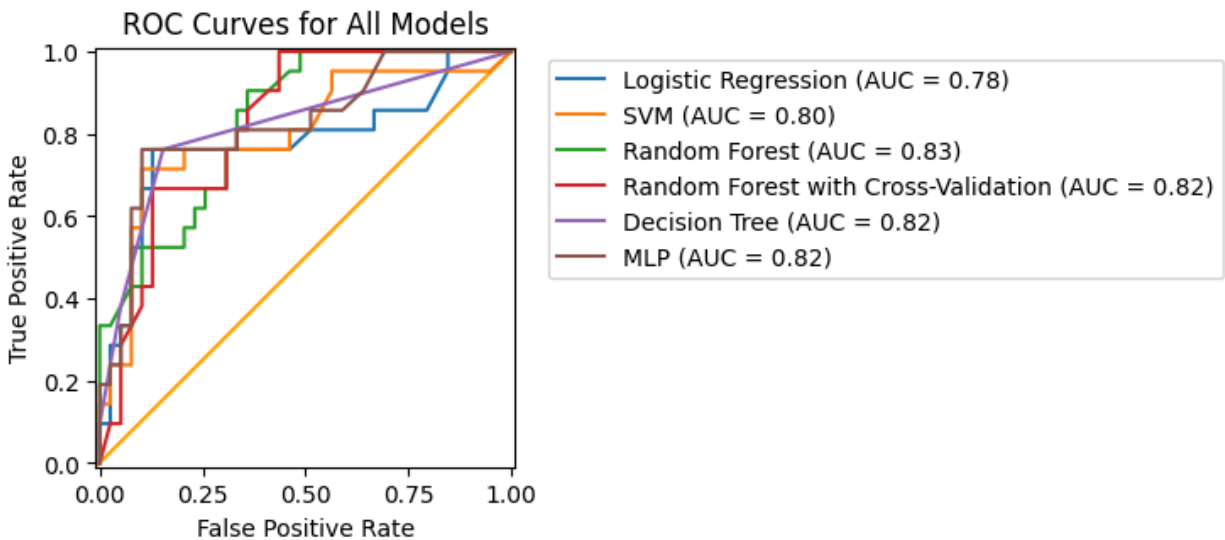


Figure 4: ROC curves with AUC scores for every model

5 Discussion

From the results, machine learning models have proven to provide meaningful predictions of patient survival outcomes. Although the MLP and decision tree models both achieved the highest accuracy, the decision tree demonstrated stronger sensitivity, making it more suitable for situations where detecting true positives carries more weight. On the other hand, the MLP model balanced sensitivity and specificity overall and maintained high performance across all metrics. Because the target variable of the project was to predict patient survival outcomes, there needed to be a balance between sensitivity and specificity. Especially in a medical context, sensitivity is important in order to correctly identify survival while specificity prevents the mislabeling of survivors, which could lead to inaccurate interpretations of patient risk. This need for a well-rounded performance made the MLP classifier the best overall model.

ROC curve analysis further confirmed that all models were capable of using two features alone for predictions, with the standard random forest without cross-validation achieving the largest AUC of 0.83 and other models, cross-validated random forest, decision tree, and MLP, were close behind at an AUC of 0.82. Having the AUC scores very close to 1 meant the models were capable of making the right classification for each patient most of the time. Even the models that were used with minimal hyperparameter tuning or used with default settings, such as the logistic regression and SVM in this project, obtained accuracy scores greater than 70%.

Incorporating parameter optimization improved prediction performance, as demonstrated by how the random forest model without cross-validation had 70% accuracy but, with the additional cross-validation, increased to 76.7% accuracy. Overall, these findings indicate that machine learning models could capture the relevant patterns in serum creatinine and ejection fraction to predict patient survival effectively, supporting their use in clinical settings.

6 Conclusion

The overall goal of the project was to apply and investigate machine learning models in a binary classification problem of identifying patient survival using a small set of features, determining which type of model resulted in the best performance. This study demonstrated that indeed these machine learning models are capable of providing reliable predictions of heart failure patient survival, and serum creatinine and ejection fraction alone were sufficient enough to achieve strong model performance, supporting prior findings that these features are highly informative for survival outcomes [1]. Among the models tested in this project, the decision tree and MLP models obtained the highest accuracy scores, while further sensitivity and specificity analyses highlighted their differing strengths in identifying patient survival versus death outcomes. These results suggest that machine learning could potentially serve as a support tool for clinical decision making, enabling faster assessment of patient survival. Future work could expand the feature set, include larger patient datasets, and explore more advanced architectures or hyperparameter tuning to potentially improve model performance and predictive accuracy.

7 References

1. Davide Chicco, Giuseppe Jurman: "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 (2020).
<https://doi.org/10.1186/s12911-020-1023-5>
2. "Heart Failure Clinical Records." UCI Machine Learning Repository, 2020,
<https://doi.org/10.24432/C5Z89R>.