

CS 584 Project Final Report

Heart Transplant Outcome Prediction

Ashmita Gupta
Bhavana Dontamsetti
Neil Dhote

1 Introduction

Heart transplantation serves as a viable treatment option for patients with end-stage heart failure. However, survival rates post-transplantation are influenced by various factors, such as age, sex, and donor characteristics. In this project, we aim to analyze data from the Stanford Heart Transplant study to gain insight into the factors affecting the survival of heart transplant patients. Utilizing machine learning algorithms, we will develop a predictive model to determine the likelihood of a patient's survival following a heart transplant procedure.

The scarcity of organs available for transplant, with approximately 2,500 available annually compared to 60,000 potential recipients, necessitates maximizing the benefits derived from heart transplantation through improved recipient selection. Accurate estimation of heart transplant outcomes not only facilitates informed patient consent by enabling patients to better comprehend the associated risks and benefits, but also assists physicians in decision-making by evaluating patient-specific risks of the procedure, rather than relying on population-wide assessments. Consequently, accurate outcome prediction for heart transplantation is of utmost importance.

2 Problem Description

The primary goal of this project is to develop a predictive model for heart transplant patient survival, taking into account various patient and donor characteristics. The dataset encompasses information on 378 patients who underwent heart transplant procedures at Stanford University Hospital between 1968 and 1977, including variables such as patient age, year of acceptance for transplant, prior heart transplant, type of transplant (control or treatment), waiting time for transplant, and survival status and time. Through the application of machine learning algorithms, we aim to construct a model capable of predicting heart transplant patient survival based on these factors. This analysis will provide valuable insights to assist doctors and medical professionals in making informed decisions regarding heart transplant patients, ultimately leading to improved patient outcomes.

Predicting patient survival is crucial in clinical decision-making, as it enables physicians to effectively allocate resources and devise treatment strategies. By creating a predictive model derived from available patient data, clinicians can make well-informed decisions about patient care, optimize resource allocation, and enhance outcomes. Consequently, the objectives of this analysis include exploring the dataset, preprocessing the data, and developing a machine learning model that accurately predicts patient survival post-heart transplantation.

3 Brief Description of the Process and Each Variable in the Dataset

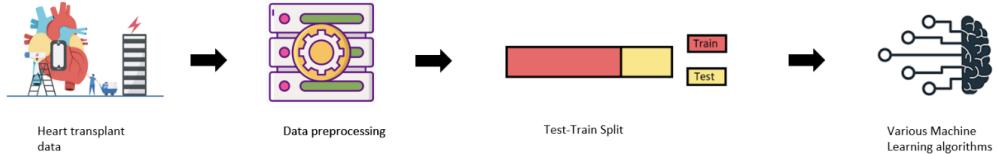


Figure 1: Evaluation of various machine learning algorithms

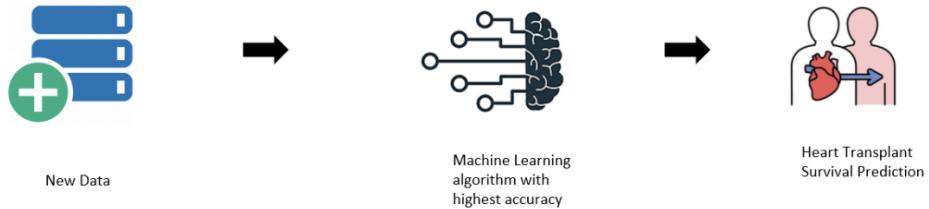


Figure 2: Apply algorithm with highest accuracy on new dataset to predict patient survival post transplantation

4 Data Sources

4.1 Dataset 1: Stanford University Hospital Data (Stanford)

id: unique identifier for each patient.
acceptyear: the year in which the patient was accepted into the transplant program.
age: age of the patient at the time of acceptance into the program.
survived: binary variable indicating whether the patient survived the transplant or not (died or alive).
survtime: the number of days between the transplant and the patient's death or last follow-up.
prior: binary variable indicating whether the patient had a prior transplant (no prior transplant or prior transplant).
transplant: categorical variable indicating the type of transplant (control or treatment).
wait: the number of days the patient was on the waiting list before being accepted into the transplant program.

4.2 Dataset 2: Journal of the American Statistical Association Data (JASA)

birth.dt: birth date of patient
accept.dt: acceptance into program
tx.date: transplant date
fu.date: end of followup
fustat: dead or alive
surgery: prior bypass surgery
age: age (in years)
futime: followup time
wait.time: time before transplant
transplant: transplant indicator
mismatch: mismatch score
hla.a2: particular type of mismatch
mscore: another mismatch score
reject: rejection occurred

Note: We dropped birth.dt, mismatch, hla.a2, mscore, reject from Jasa dataset because these are not strongly associated with the target variable (survived) and added id and survival time column. Also, we have renamed the columns in the JASA dataset for the process of merging.

4.3 Dataset 3: Heart Dataset From R Survival Package (Heart)

start, stop: entry and exit time

event: survived or not

age: age of the patient at the time of acceptance into the program

year: year of acceptance (in years after 1 Nov 1967)

surgery: prior bypass surgery (yes = 1)

transplant: received transplant (yes = 1)

id: patient id

5 Key Observations from Exploratory Data Analysis Before Modeling

- **Observation 1:** The range of the age variable is from 0 to 73, with a mean of 45.9 years and standard deviation of 12.3 years.
- **Observation 2:** The range of the wait variable is from 0 to 365, with a mean of 73.8 days and standard deviation of 81.6 days. This implies that the patient has to wait for an average of 74 days to receive a transplant
- **Observation 3:** For the categorical variable (transplant), the summary statistics include the frequency of each category (control or treatment). In this dataset, there are 59 patients in the control group and 44 patients in the treatment group. Also, the below plot implies that survival is dependent on the transplant treatment and the ones who got treatment has much higher survival rate:

5.1 Boxplot for Age

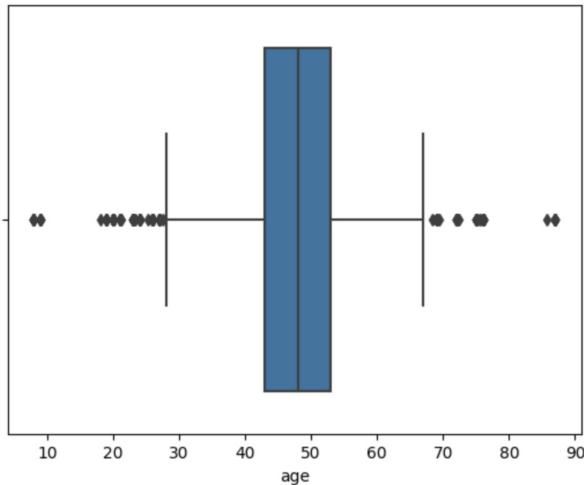


Figure 3: Here, boxplot for age in the dataset shows that the median age of patients is around 47 years. There are a few outliers on both ends of the age spectrum, with the youngest patient being around 12 years old and the oldest being around 83 years old. We can see that the majority of patients were in their 40s or early 50s.

5.2 Correlation Matrix

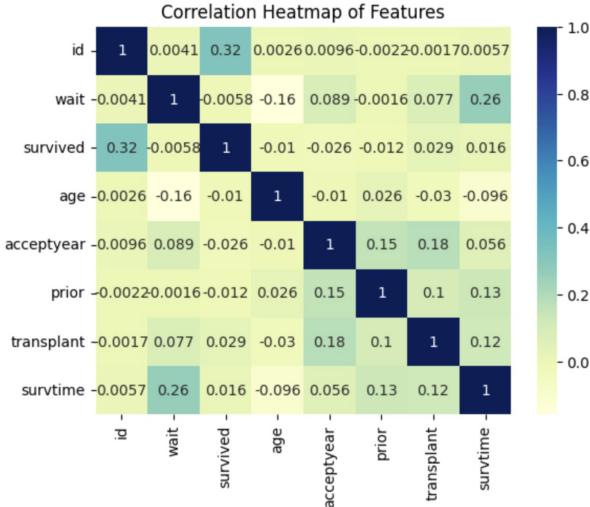


Figure 4: Here, the plot shows the correlation coefficients between each pair of variables in the dataset. We can see that there is a moderate positive correlation between age and survival time.

5.3 Density Plot for Survival Time

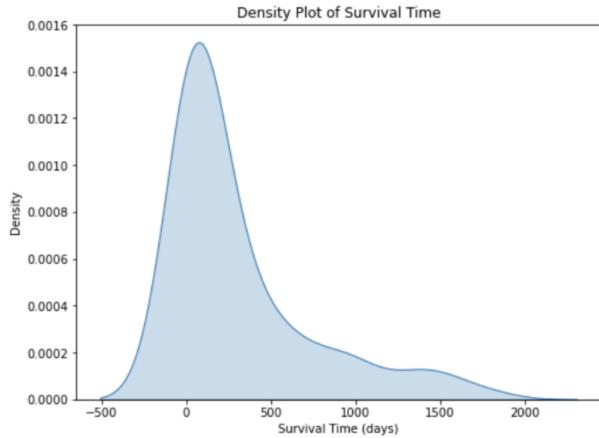


Figure 5: Here, the plot indicates that the majority of patients survived for less than 2 years after receiving a heart transplant. There is a long tail on the right-hand side of the plot, indicating that some patients survived for much longer periods. The plot is slightly skewed to the right. Overall, the plot suggests that the survival time variable may be a useful predictor for the survival of heart transplant patients.

6 Methods

6.1 Data Augmentation (SMOTE)

We faced data challenges because data in this domain is quite sensitive and it's not easily available. Hence, we collected data from different sources and did data augmentation to get a good amount of data for training the model. In this project, we utilized the Synthetic Minority Over-sampling Technique (SMOTE) for data augmentation to address the issue of limited data points in our dataset. SMOTE is an effective method for generating synthetic samples by interpolating between existing data points, increasing the size of the dataset while preserving its original characteristics. This approach is particularly useful for handling imbalanced datasets, as it creates synthetic samples that balance the distribution of classes, ultimately enhancing the performance of machine learning models. After applying SMOTE to our dataset, the number of data points increased to 5,400, providing a more substantial foundation for training and testing our predictive model, which in turn led to more accurate and reliable outcomes in heart transplant patient survival prediction.

6.2 Testing/Training Split

The data was divided into separate training and testing datasets to ensure a reliable evaluation of our predictive model. We utilized two data augmented (SMOTE) datasets, the Stanford dataset and the JASA dataset, for training purposes, providing a diverse foundation for our model. To assess the performance of our trained model, we employed a third dataset, the Heart dataset, exclusively for testing, enabling us to evaluate the model's effectiveness and generalizability on previously unseen data.

6.3 Hyperparameter Tuning (Grid Search)

Grid Search was employed for hyperparameter tuning to optimize the performance of our four selected machine learning models: Logistic Regression, Support Vector Machines (SVM), Random Forest, and Naive Bayes. Grid Search systematically evaluates a specified range of hyperparameter values, identifying the combination that yields the best model performance. This approach allowed us to fine-tune our machine learning models, ensuring that each model achieved the highest possible accuracy and generalizability in predicting heart transplant patient survival.

7 Models

7.1 Logistic Regression

Logistic Regression models the relationship between a binary dependent variable and one or more independent variables. In the context of our project, Logistic Regression was employed to predict the probability of patient survival based on the available patient and donor characteristics. This method's simplicity and interpretability make it an attractive choice for our project, as it allows for straightforward examination of the relationships between the input variables and survival outcomes.

7.2 Support Vector Machines (SVM)

Support Vector Machines (SVM) aim to find the optimal separating hyperplane between two classes, maximizing the margin between them. In our project, we utilized SVMs to classify patients into survival and non-survival groups based on their characteristics. The SVMs model is known for its ability to handle high-dimensional data and its robustness against overfitting, making it a suitable candidate for analyzing complex relationships between multiple variables and predicting patient outcomes.

7.3 Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees and combines their predictions to improve the overall model performance. For our project, we used Random Forest to predict patient survival, taking advantage of its ability to capture complex interactions between variables and its inherent resistance to overfitting. By aggregating the predictions of numerous decision trees, the Random Forest model reduces the risk of errors and biases, leading to more accurate and reliable predictions of heart transplant patient outcomes.

7.4 Naive Bayes

Naive Bayes is a probabilistic machine learning algorithm based on the Bayes' theorem, which assumes independence between input variables. As part of our project, we applied Naive Bayes to predict patient survival by estimating the conditional probabilities of each input variable given the survival outcome. Despite its simplifying assumption of variable independence, Naive Bayes has demonstrated effectiveness in various classification tasks and can provide a fast and efficient solution for predicting heart transplant patient survival.

8 Results

8.1 Logistic Regression

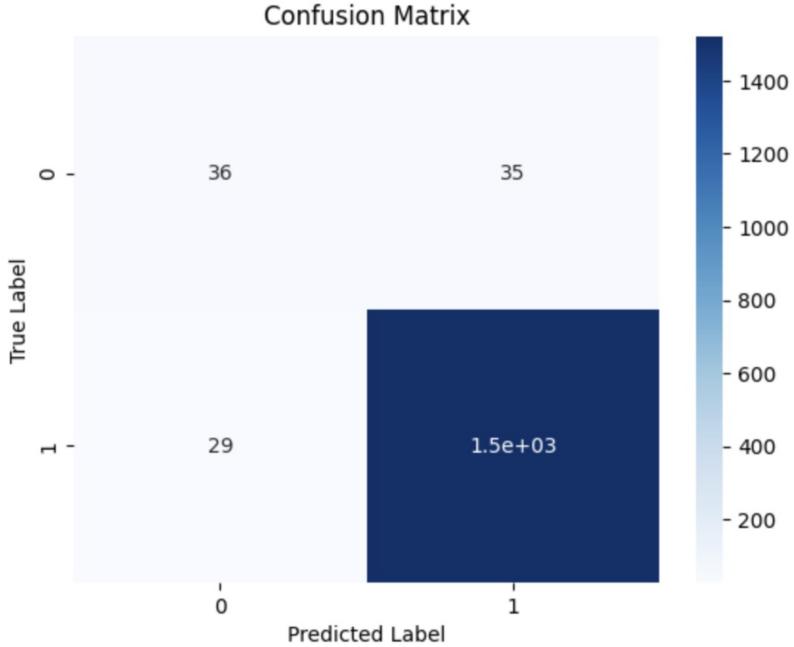


Figure 6: Confusion Matrix for Logistic Regression

```
Training Accuracy using Logistic regression: 0.9701058201058201
Validation set Accuracy using Logistic regression:: 0.9598765432098766
Testing Accuracy using Logistic regression:: 0.563953488372093
```

Figure 7: Accuracies for Logistic Regression

```
Precision: 0.977491961414791
Recall: 0.9812782440284055
```

Figure 8: Precision and Recall for Logistic Regression

Given the training accuracy of 0.9701, the logistic regression model was able to correctly predict the class labels for 97.01% of the examples in the training set. The validation set accuracy of 0.9599 indicates that the model was able to generalize well on examples that it had not seen during training. However, the testing accuracy of 0.5639 indicates that the model did not perform as well on the unseen test examples. This could be due to overfitting to the training data or not having enough data to generalize well on the test set. A precision of 0.9775 indicates that out of all the examples that the model predicted as positive, 97.75% of them were actually positive. A recall of 0.9813 indicates that the model correctly identified 98.13% of all the actual positive examples.

8.2 Decision Tree

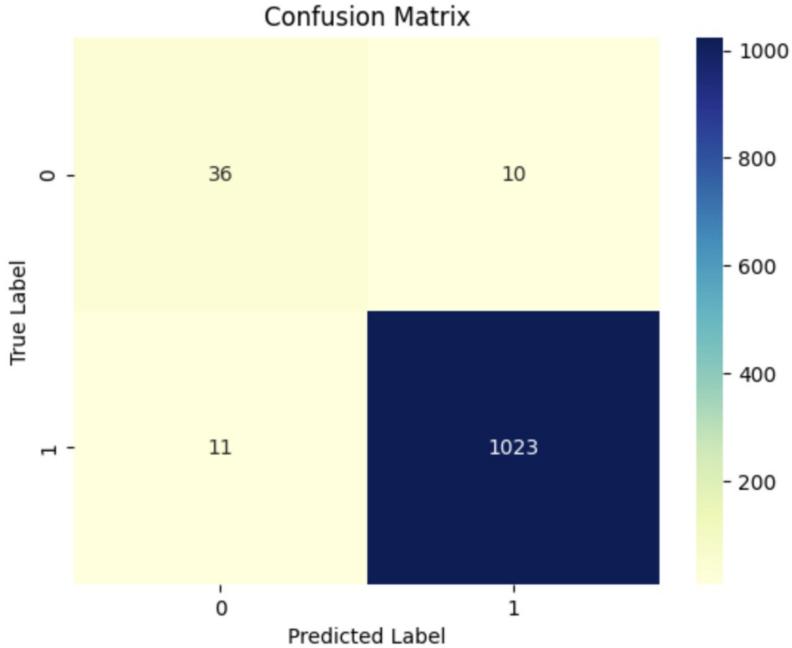


Figure 9: Confusion Matrix for Decision Tree

```
Best parameters found: {'max_depth': 7, 'min_samples_leaf': 4, 'min_samples_split': 5}
Training Accuracy using Decision Tree: 0.992824074074074
Validation set Accuracy using Decision Tree: 0.9805555555555555
Testing Accuracy using Decision Tree: 0.6627906976744186
```

Figure 10: Accuracies for Decision Tree

```
Precision: 0.9903194578896418
Recall: 0.9893617021276596
```

Figure 11: Precision and Recall for Decision Tree

The best parameters found for the Decision Tree model are 'max_depth': 7, 'min_samples_leaf': 4, 'min_samples_split': 5. This means that the model was trained with a maximum depth of 7, a minimum number of samples required to be at a leaf node of 4, and a minimum number of samples required to split an internal node of 5. The training accuracy of 0.9928 indicates that the Decision Tree model was able to correctly predict the class labels for 99.28% of the examples in the training set. The validation set accuracy of 0.9806 indicates that the model was able to generalize well on examples that it had not seen during training. The testing accuracy of 0.6628 indicates that the model did not perform well on the unseen test examples. This could be due to overfitting to the training data or not having enough data to generalize well on the test set. A precision of 0.9903 indicates that out of all the examples that the model predicted as positive, 99.03% of them were actually positive. A recall of 0.9894 indicates that the model correctly identified 98.94% of all the actual positive examples.

8.3 Random Forest

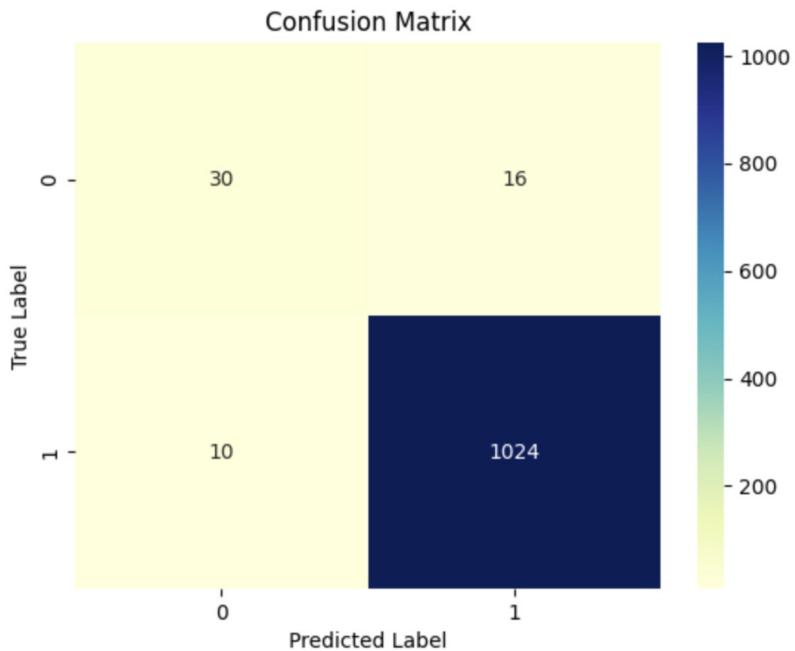


Figure 12: Confusion Matrix for Random Forest

```
Best parameters found: {'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 150}
Training Accuracy using Random Forest: 0.9990740740740741
Validation set Accuracy using Random Forest: 0.975925925925926
Testing Accuracy using Random Forest: 0.8372093023255814
```

Figure 13: Accuracies for Random Forest

```
Precision: 0.9846153846153847
Recall: 0.9903288201160542
```

Figure 14: Precision and Recall for Random Forest

The best parameters found for the Random Forest model are 'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 150. This means that the model was trained with a maximum depth of 10, a minimum number of samples required to be at a leaf node of 1, a minimum number of samples required to split an internal node of 2, and 150 decision trees. The training accuracy of 0.9991 indicates that the Random Forest model was able to correctly predict the class labels for 99.91% of the examples in the training set. The validation set accuracy of 0.9759 indicates that the model was able to generalize well on examples that it had not seen during training. The testing accuracy of 0.8372 indicates that the model performed reasonably well on the unseen test examples. A precision of 0.9846 indicates that out of all the examples that the model predicted as positive, 98.46% of them were actually positive. A recall of 0.9903 indicates that the model correctly identified 99.03% of all the actual positive examples.

8.4 Naive Bayes

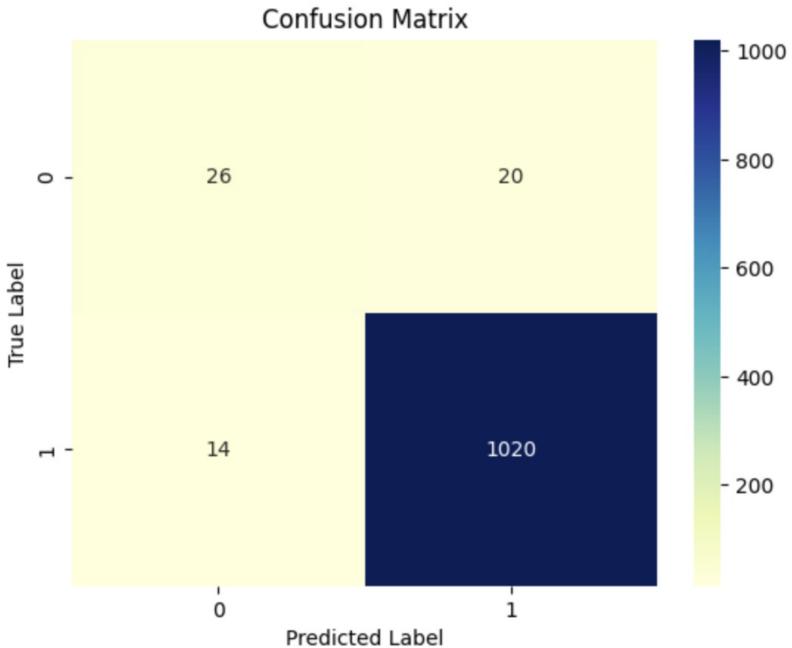


Figure 15: Confusion Matrix for Naive Bayes

```
Training Accuracy using Naives Bayes: 0.9773148148148149
Validation set Accuracy using Naives Bayes: 0.9685185185185186
Testing Accuracy using Naives Bayes: 0.5581395348837209
```

Figure 16: Accuracies for Naive Bayes

```
Precision: 0.9807692307692307
Recall: 0.9864603481624759
```

Figure 17: Precision and Recall for Naive Bayes

The training accuracy of 0.9773 indicates that the Naive Bayes model was able to correctly predict the class labels for 97.73% of the examples in the training set. The validation set accuracy of 0.9685 indicates that the model was able to generalize well on examples that it had not seen during training. However, the testing accuracy of 0.5581 indicates that the model did not perform well on the unseen test examples. This could be due to overfitting to the training data or not having enough data to generalize well on the test set. A precision of 0.9808 indicates that out of all the examples that the model predicted as positive, 98.08% of them were actually positive. A recall of 0.9865 indicates that the model correctly identified 98.65% of all the actual positive examples.

9 Comparison of Models

Model	Training Set Accuracy	Validation Set Accuracy	Test Set Accuracy	Precision	Recall
Decision Tree	99%	98%	66.20%	0.990319458	0.989361702
Naïve Bayes	97.70%	96.80%	55.80%	0.980769231	0.986460348
Logistic Regression	97%	95.90%	56.40%	0.977491961	0.981278244
Random Forest	99.90%	97.5	84%	0.984615385	0.99032882

Figure 18: Comparison Table of the 4 Models Used

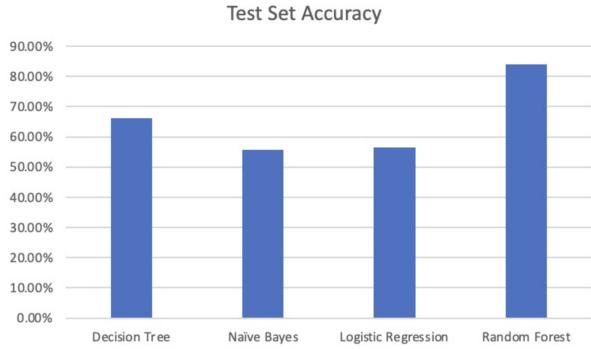


Figure 19: Comparison Chart of the 4 Models Used

Based on the above results, the Random Forest model has the highest test set accuracy of 84%, followed by the Decision Tree model with 66.20% accuracy. The Naïve Bayes and Logistic Regression models have lower test set accuracy of 55.80% and 56.40% respectively. In terms of precision, the Random Forest model has the highest value of 0.9846, meaning that it correctly identifies 98.46% of all the positive examples. The Naïve Bayes model follows with a precision of 0.9808. In terms of recall, the Decision Tree model has the highest value of 0.9894, meaning that it correctly identifies 98.94% of all the actual positive examples. The Naïve Bayes model follows with a recall of 0.9865. Overall, the Random Forest model seems to be the best model among the four, based on its higher accuracy and precision values. However, the Decision Tree model performs well in terms of recall and could be a good option if correctly tuned to avoid overfitting. We have also built a web application in which we can predict the heart transplantation outcome based on the inputs the user gives in the input textboxes.

10 Conclusion and Future Work

This project has made notable progress in predicting patient survival after heart transplants. Using machine learning models like Logistic Regression, SVM, Random Forest, and Naïve Bayes, we built a predictive model to help healthcare professionals make better decisions. This project highlights the value of data augmentation, diverse training datasets, hyperparameter tuning, and employing various machine learning models. Through this project, we gained insights into data preprocessing, model optimization, and the significance of accurate outcome prediction in clinical settings. These learnings can be applied to improve resource allocation and treatment planning. Future work could extend this project to other organ transplants and leverage advancements in machine learning and artificial intelligence to refine the predictive capabilities of our models.

11 Source Code

<https://github.com/neildhote/OrganOracle>

12 References

1. <https://optn.transplant.hrsa.gov/data/view-data-reports/national-data/>
2. <https://www.nature.com/articles/s41598-018-21417-7>
3. [https://www.jhltonline.org/article/S1053-2498\(20\)30675-6/fulltext](https://www.jhltonline.org/article/S1053-2498(20)30675-6/fulltext)
4. <https://ieeexplore.ieee.org/document/8036766>
5. <https://www.sciencedirect.com/science/article/abs/pii/S0167923620301184>

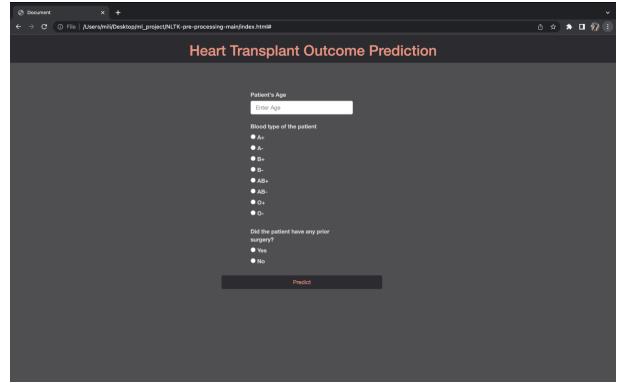


Figure 20: Web Application