# CPEN 355 Final Project

Heart Failure Prediction

**Ryan Manak**


Electrical Engineering
University of British Columbia
Canada
April 2024

# Contents

# List of Figures

# List of Tables

# Part 1

# Problem Definition

## 1.1 Introduction

Cardiovascular diseases (CVDs) are a significant global health concern, leading to numerous deaths annually. Heart failure, a common consequence of CVDs, requires early detection and management to improve patient outcomes and reduce mortality rates. Machine learning (ML) models offer a promising solution by using data to predict heart failure likelihood. This report explores an 11-feature dataset aimed at predicting heart disease, with the goal of developing an accurate ML model for early identification and management of individuals at risk, thus improving healthcare outcomes and reducing mortality associated with CVDs.

## 1.2 Dataset

This section describes the dataset, data visualization and data pre-processing with table 1.1 describing the features[1].

| **Feature** | **Description** | **Values** |
|---|---|---|
| Age | age of the patient | years |
| Sex | sex of the patient | M: Male, F: Female |
| ChestPainType | chest pain type | TA, ATA, NAP, ASY |
| RestingBP | resting blood pressure | mm Hg |
| Cholesterol | serum cholesterol | [mm/dl] |
| FastingBS | fasting blood sugar | (High) 1, (low) 0 |
| RestingECG | resting electrocardiogram results | Normal, ST, LVH |
| MaxHR | maximum heart rate achieved | between 60 and 202 |
| ExerciseAngina | exercise-induced angina | Y: Yes, N: No |
| Oldpeak | oldpeak = ST | Numeric value |
| ST_Slope | ST slope under exercise | Up, Flat, Down |
| HeartDisease | output class | 1: heart disease, 0: Normal |

Table 1.1: Description of Features in Heart Failure Prediction Dataset

Chest pain type, resting ECG, exercise angina and sex are all categorical features. To make these features easier to both visualized and processed with ML models

each categorical feature was mapped to an integer value. Below is an example of the mapping for ST Slope:

$$\text{Up} = 0, \text{Flat} = 1, \text{Down} = 2$$

With this update all features are visualized using a pair-plot seen in figure 1.1. Please note Orange corresponds to patients with heart disease and blue is patients without.

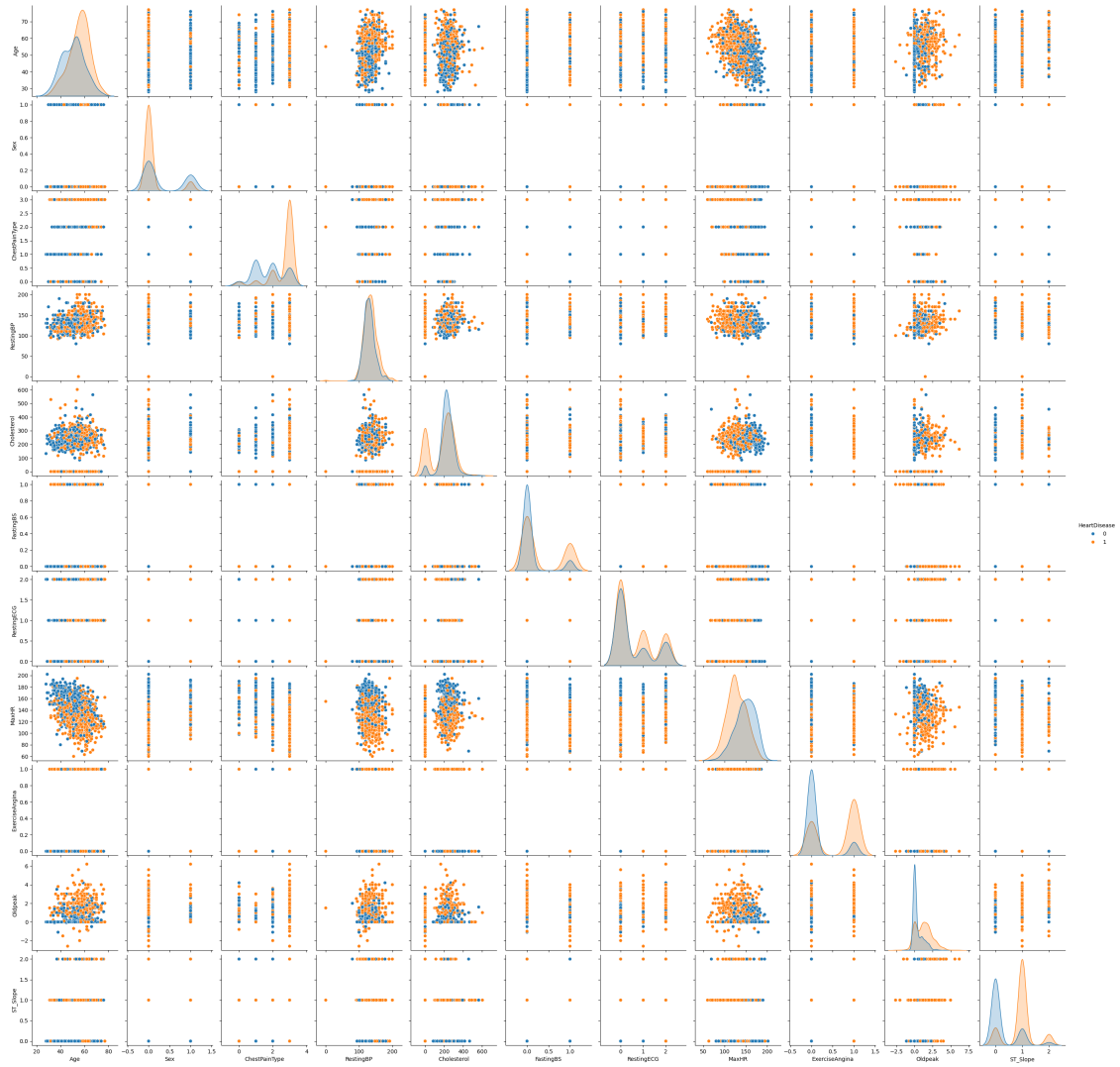


Figure 1.1: Full Dataset Pair-plot

First inspection shows ST-slope, oldpeak, exercise angina, and max heart rate show visible separation between patients with and without heart disease. Figure 1.2 shows a pairplot of only these features.

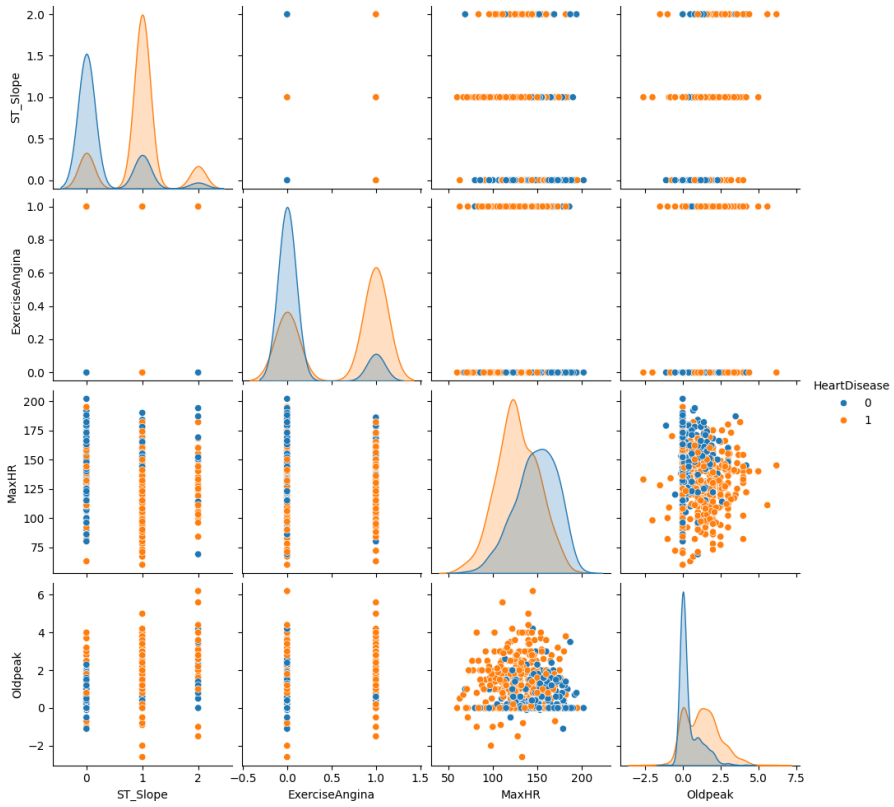


Figure 1.2: Data-subset Pair-plot

Further inspection of these features can be performed later if the models tested do not effectively classify patients with and without heart disease.

Finally, before modeling, the full dataset is split into test and train datasets using Scikit-Learn's train-test-split method, which randomizes the data set. Eighty percent of the dataset was allocated for training, while the remaining 20% was designated for testing. The total size of the training dataset is 734 elements, with the test dataset comprising 184 elements.

# Part 2

# Methods and Experiments

## 2.1 Introduction

This section delves into the different models used for classification. Each model was tested by initially varying its hyperparameters and comparing the accuracy of predictions for the test and train datasets. This information is then used to provide insights into the range that should be used for a grid search with k-fold cross-validation to find the final model.

## 2.2 Random Forest Model

### 2.2.1 Method

Table 2.1 describes the hyper-parameters used in the random forest model [2].

| **Name** | **Description** |
|---|---|
| Criterion | Criteria for Split |
| Number of Estimators | Number of trees created |
| Max Features | Max features of each Tree |
| Max Depth | Max depth of each tree |
| Alpha | Cost Complexity parameter for pruning |

Table 2.1: Random Forest Hyperparameters

### 2.2.2 Experiment

Gini-impurity and Shannon information gain, (via entropy or log-loss) are compared as splitting criterion [3]. Additionally cost complexity pruning is performed via tuning alpha [3] :

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}|$$

Where $|\tilde{T}|$ is the number of terminal node in $T$ and $R$ is defined as the miss-classification rate [3].

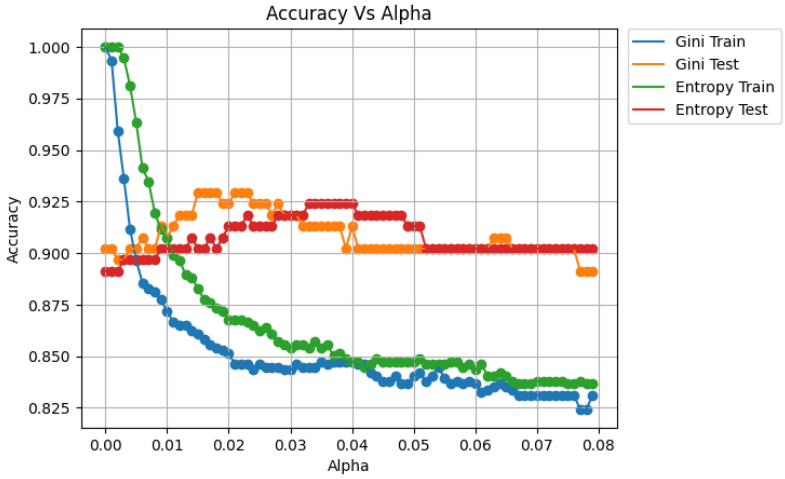


Figure 2.1: Random Forest Alpha Sweep

Figure 2.1 shows a sweep of alpha for both Gini and Log-loss splitting criteria. No significant difference is discernible between log-loss and Gini splitting criteria. We clearly see over-fitting to the training data with no pruning, conversely alpha values above 0.3 show under-fitting of the training data.

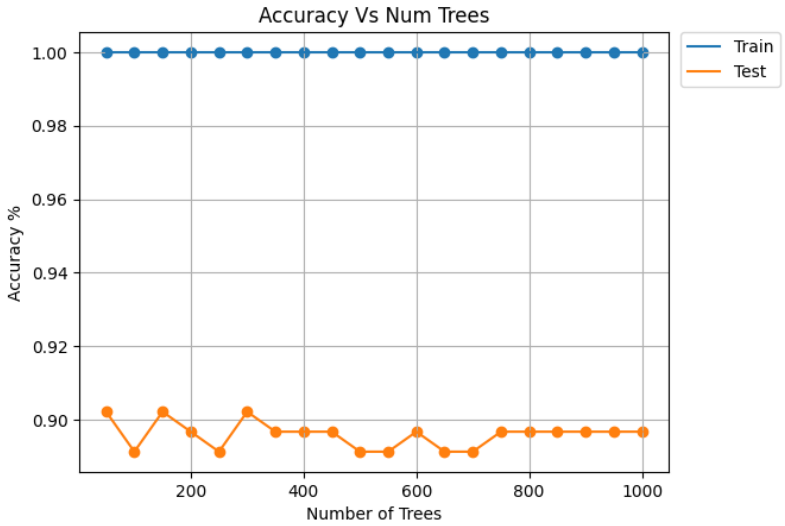


Figure 2.2: Random Forest Tree Sweep

Figure 2.2 shows a sweep of number of trees used in the random forest classifier. We clearly see no benefit to accuracy in the test data beyond 350 classifiers, as we are not using any cross-validation for this test we also see training accuracy to over-fit.

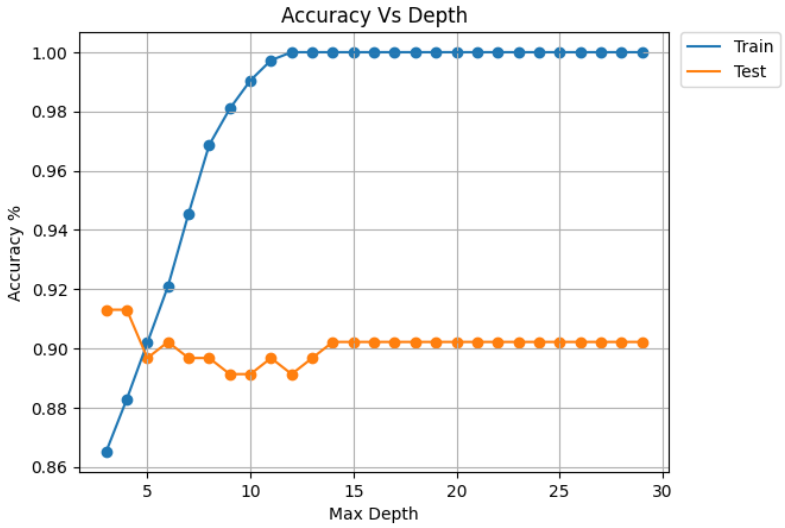


Figure 2.3: Random Forest Tree Depth Sweep

Figure 2.3 showed under fitting on the training data for trees with a max depth below 15 while depths above 20 showed no benefit to the test data accuracy.

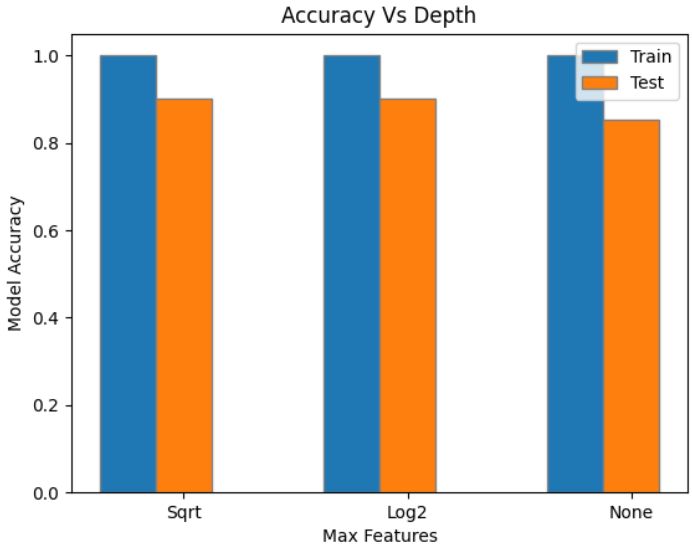


Figure 2.4: Random Forest Tree Max Features Sweep

Finally figure 2.4 showed no significant difference between between Log2(m) features and sqrt(m) features with no limit clearly performing worse. Now with the insights gained above a grid search was performed with 10 folds for k-fold cross validation. The follow space was searched:

- Alpha Ranging From 0.001 to 0.005
- Max Depth From 4 to 20

The number of trees was set to 350, the max number of features was set to sqrt(m), and the splitting criteria were configured for Gini. After completing the

search, it was performed multiple times with finer increments of hyperparameters, and the best hyperparameters are summarized in Table 2.2. The performance of the model is summarized in table 2.3.

| **Name** | **Description** |
|---|---|
| Criterion | Gini |
| Number of Estimators | 350 |
| Max Features | sqrt(m) |
| Max Depth | 10 |
| Alpha | 0.003 |

Table 2.2: Optimal Random Forest Hyperparameters

| **Dataset** | **Accuracy** |
|---|---|
| Training Data | 94.4141 |
| Testing Data | 90.2173 |
| Full Dataset | 93.5729 |

Table 2.3: Random Forest Accuracy

## 2.3 SVM Models

### 2.3.1 Method

Table 2.4 describes the hyper-parameters used for SVM classifiers. Note that C is only sweept for the linear kernel and is left at 1 for all other non-linear kernels [2].

| **Name** | **Description** |
|---|---|
| Kernel | Criteria for Split |
| Gamma | Kernel Coefficient for non-linear Kernels |
| C | Regularization Parameter |

Table 2.4: SVM Classifier Hyperparameters

Each feature is also normalized to not give excess weight to features with a large scale, such as max heart rate. The formula below shows the operation performed to make a normalized version of both the train and test dataset.

$$X_N = \frac{X - \text{mean}(X)}{\text{std}(X)}$$

### 2.3.2 Experiment

A test is performed for each kernel as they will all require unique hyper-parameter tuning. Figure 2.5 shows variation of gamma for the Gaussian kernel. The figure clearly shows over-fitting for values above $10^0$ and under fitting for values less than $10^{-3}$.

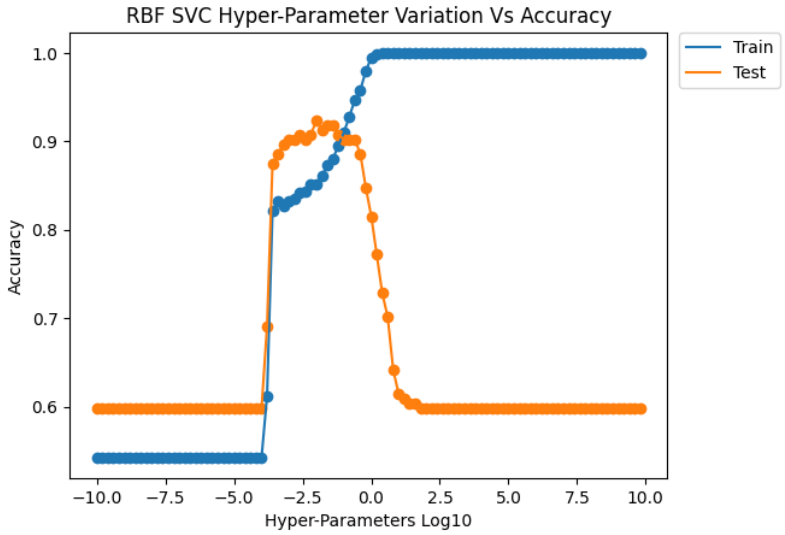


Figure 2.5: SVM Gaussian Kernel Hyperparameter Variation

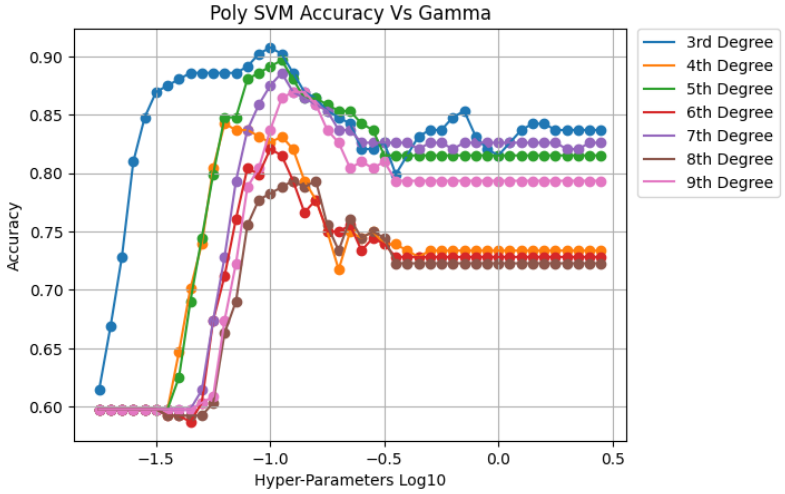


Figure 2.6: SVM Polynomial Kernel Hyperparameter Variation

Figure 2.6 shows the prediction accuracy on test data for varying degrees of polynomial kernels and varying gamma. The third-degree polynomial is shown to be generally a better performer for a wider variation of gamma and will be used for the following k-fold sweep.

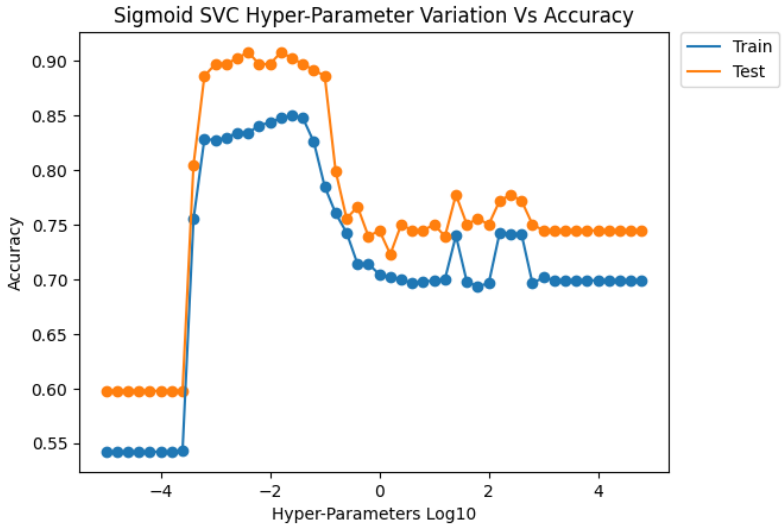


Figure 2.7: SVM Sigmoid Kernel Hyperparameter Variation

Figure 2.7 shows variation of gamma for the Sigmoid kernel. The figure clearly shows over-fitting for values above $10^0$ and under fitting for values less than $10^{-4}$.

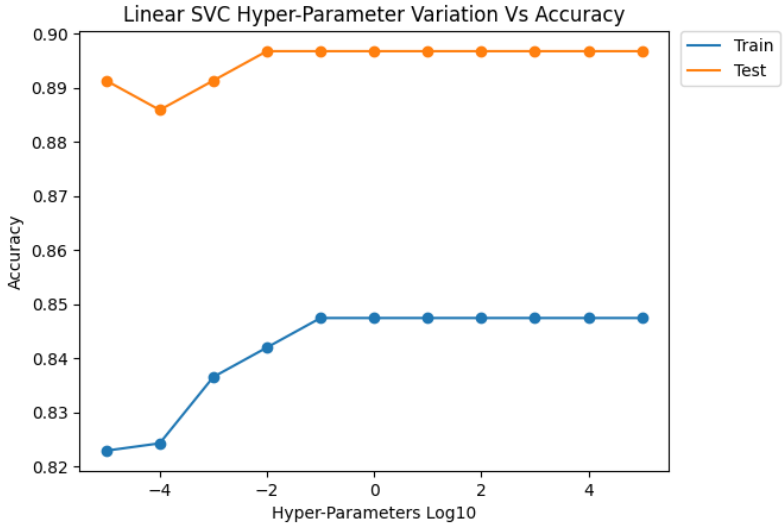


Figure 2.8: SVM Linear Kernel Hyperparameter Variation

Figure 2.8 shows variation of gamma for the linear kernel. This kernel will not be tested further with k-folds as it clearly performs worse than all other non-linear kernels and the previous random forest model.

Now a grid search is performed for the gaussian, sigmoid and third order polynomial kernels with 10 folds for k-fold cross validation. The follow space was searched:

- Gamma Ranging From $10^{-4}$ to 10

Table 2.5 clearly shows the Gaussian kernel to be the best performer with a gamma of 0.1584.

| **Model** | **Training Dataset** | **Test Dataset** | **Full Dataset** |
|---|---|---|---|
| Gaussian | 92.7792 | 90.2173 | 92.1568 |
| Sigmoid | 84.8773 | 90.7608 | 85.7298 |
| Polynomial | 87.7384 | 88.5869 | 87.7995 |

Table 2.5: SVM Accuracy Various Kernels

# Part 3

# Results Analysis

## 3.1 Results

Based on the previous sections, we determined that the Gaussian kernel for SVM and the Random Forest classifier were the best predictors. Figures 3.1 to 3.4 display the confusion matrices for each classifier, covering both the test dataset and the full dataset.

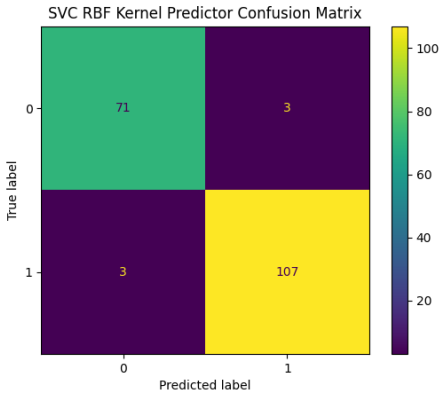


Figure 3.1: Test Data

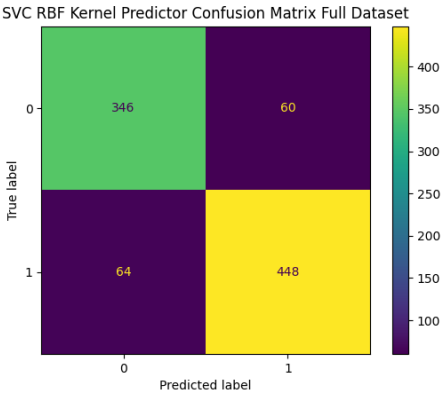


Figure 3.2: Full Dataset

SVM Confusion Matrices

We observe that the SVM model exhibits a lower false positive and false negative rate for the test data, whereas the random forest model emerges as the better predictor overall for the full dataset.

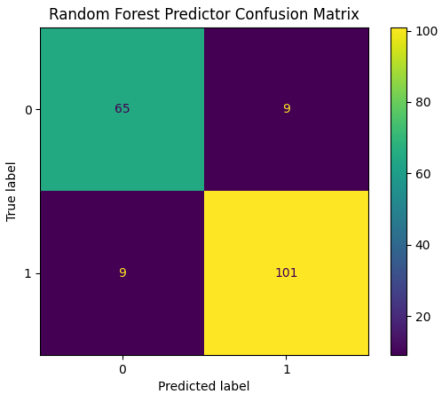


Figure 3.3: Test Data

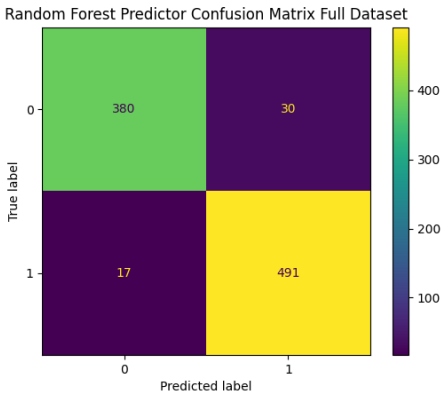


Figure 3.4: Full Dataset

Random Forest Confusion Matrices

## 3.2 Conclusion

Since the random forest classifier performs better overall for the main dataset, it is the recommended option for early prediction of heart disease. In future research, exploring feature importance and techniques like PCA could help reduce feature flexibility. Additionally, testing more complex models such as various neural networks and ensembles of multiple ML models could yield valuable insights and potentially result in more performant classifiers.

# Bibliography

[1] Kaggle, “Heart failure prediction dataset,”

[2] X. Li, “Cpen 355 lecture notes,”

[3] sklearn, “Randomforestclassifier,”