A Showdown of Classification Models to Predict Heart Disease Risk

Sona Shaukath (T00710785)

DASC 5420: Theoretical Machine Learning
Faculty of Science
Thompson Rivers University
15 April, 2023

Abstract

Heart Disease is a health issue resolved with early identification, prevention and treatment. This study is a stepping stone to classify heart disease risk based on other variables, Body Mass Index (BMI), Smoking, Physical Health, Mental Health, and Sex, to name a few. This study aids in predicting heart disease based on the five classification models. In this study, comparing five classification models, namely, K-Nearest Neighbours (KNN), Support Vector Machine (SVM), Random Forest, Decision Tree and Naive Bayes, with the help of "F1-Score", helps to identify which classification model works best for the data. This imperative study is commencing further study on this data to understand, analyse and prevent Heart Disease Risk based on available variables.

Introduction

The early detection and prevention of heart disease play a vital role in treating cardiovascular disease. Being a significant public health issue, research and project in this direction are a step closer to a better analysis of the disease. In this fast pace technological world, Machine learning techniques come forward as one of the best methods to identify and predict heart diseases. To further dwell on this, classification methods shone in the field of health care in order to diagnose several medical conditions.

This study uses classification methods to predict heart disease risk based on several factors, including BMI (Body Mass Index), Physical Health, Mental Health, General Health, Sex, Age, and many more factors. The sole purpose of this imperative study is to apply five classification models: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, Decision Tree and Naive Bayes, to predict the heart disease risk using the evaluation metrics F1-score. In addition, this study also aids in identifying which classification methods work best to predict heart disease risk.

Data

The data set used for this study is a derived from the dataset from the CDC and is a major part of the Behavioral Risk Factor Surveillance System (BRFSS), which conducts annual telephone surveys to gather data on the health status of U.S. residents. The original dataset consists of 401,958 rows and 279 columns. From this large dataset, a smaller dataset was derived with 319,795 rows and 18 columns in Kaggle [1]. This dataset from Kaggle was my source for this study.
From 18 columns in this dataset, “HeartDisease” is the response variable which is a categorical variable with Yes or No values. The other 17 variables are considered as predictor variables which aids in identifying the best classification model to predict the heart risk. The other variables include:

1. “BMI” (Body Mass Index) – continuous variable with 1491 unique values.
2. “Smoking” – Categorical variable with 2 unique values “Yes” or “No” which denoted whether a person smokes or not.
3. “AlcoholDrinking” – Categorical variable with 2 unique values “Yes” or “No” which denotes whether the person drink alcohol or not.

4. "Stroke" – Categorical variable with 2 unique values "Yes" or "No" which implies whether a person had a stroke or not.
5. "PhysicalHealth" – variable with 30 unique values.
6. "MentalHealth" – variable with 30 unique values.
7. "DiffWalking" – Categorical variable with 2 unique values "Yes" or "No" which implies whether the person has a difference in walking or not.
8. "Sex" – Categorical variable with 2 unique values "Male" or "Female".
9. "AgeCategory" –variable which is denoted in buckets of age.
10. "Race" – Categorical variable with 6 unique values.
11. "Diabetic" – Categorical variable with 5 unique values "No", "No, borderline diabetes", "Yes(during pregnancy)", and "Yes"
12. "PhysicalActivity" – Categorical variable with 2 unique values "Yes" or "No" which implies whether the person does physical activities or not.
13. "GenHealth" – Categorical variable with 5 unique values "Poor", "Fair", "Good", "Very Good" and "Excellent".
14. "SleepTime" –variable which is denotes the sleeping time of a person.
15. "Asthma" – Categorical variable with 2 unique values "Yes" or "No" which implies whether the person has Asthma or not.
16. "KidneyDisease" – Categorical variable with 2 unique values "Yes" or "No" which implies whether the person has/had kidney disease or not.
17. "SkinCancer" – Categorical variable with 2 unique values "Yes" or "No" which implies whether the person has/had skin cancer or not.

Due to low computing power on my personal computer, I initially sampled 10000 rows for my study so that the models work properly. This was done because when I tried to fit the models on the available dataset, it was loading but the answers where not getting printed. After further research on this, I came to an understanding that this is because of the sample size and then decided to reduce the size.

Exploratory Data Analysis

The GitHub link for the R Code, .csv file is https://github.com/sonashaukath22/DASC-5420-Final-Project.

Data Pre-processing

1. *Removal of Null Values:* The null values were removed from the dataset using the "na.omit" function in R. This aids in not considering the rows that has null values.
2. *Conversion of variable with String Values into Numeric Values:*
   2.1. The variables which contained string values were converted to numeric values. The variables "HeartDisease", "Smoking", "AlcoholDrinking", "DiffWalking", "PhysicalActivity", "Asthma" , "KidneyDisease", and "SkinCancer" with values "Yes" or "No" were converted to "0" for "No" and "1" for "Yes".
   2.2. The variable "Sex" was converted to "1" for "Male" and "0" for "Female".
   2.3. "SleepTime" variable was converted to categorical variable by taking the values greater than the median of the values in "SleepTime" to "1" and the rest to "0".

2.4. The variable “GenHealth” was converted into numeric values with “0” for “Poor”, “1” for “Fair”, “2” for “Good”, “3” for “Very good” and “4” for “Excellent”.

2.5. “Diabetic” was converted into numeric values with “0” foe “No”, “1” for “No, borderline diabetes”, “2” for “Yes (during pregnancy)” and “3” for “Yes”.

3. *Checking for outliers:* The variable “BMI” is the only continuous variable due to which a boxplot was plotted to check whether the variable has outliers. Figure 1 shows that the variable “BMI” has outliers.

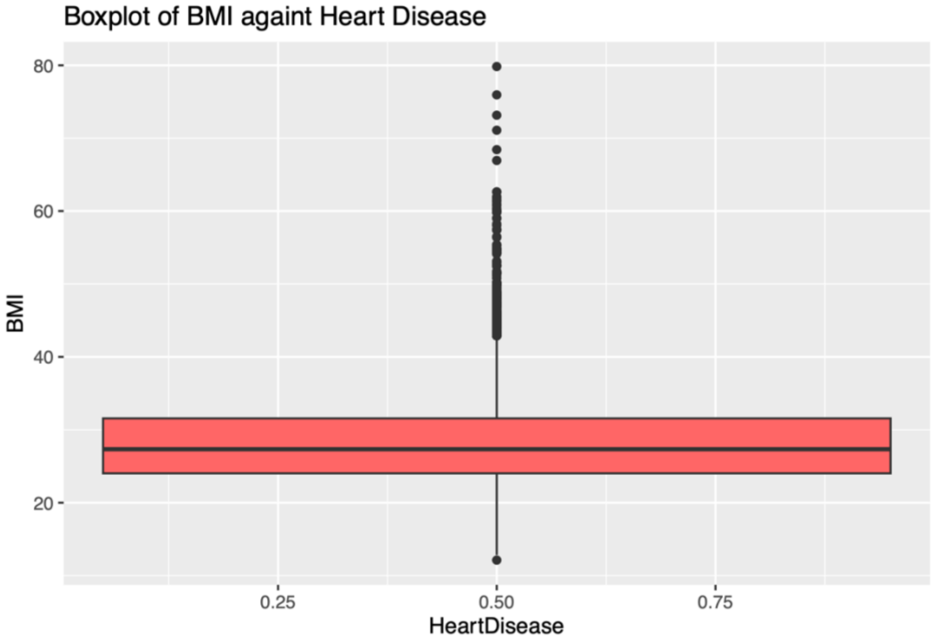


Figure 1. Boxplot of BMI against Heart Disease (Before Removal of Outliers)

The IQR (Interquartile Range) is one of the very commonly used method to remove outliers [2]. This involves the calculation of difference between the third and first quartile of the dataset and multiplying this difference with a constant (here 1.5) to calculate a threshold. The datapoints that fall outside this calculated threshold are outliers which are removed. Figure 2 shows the boxplot of BMI after the removal of outliers.

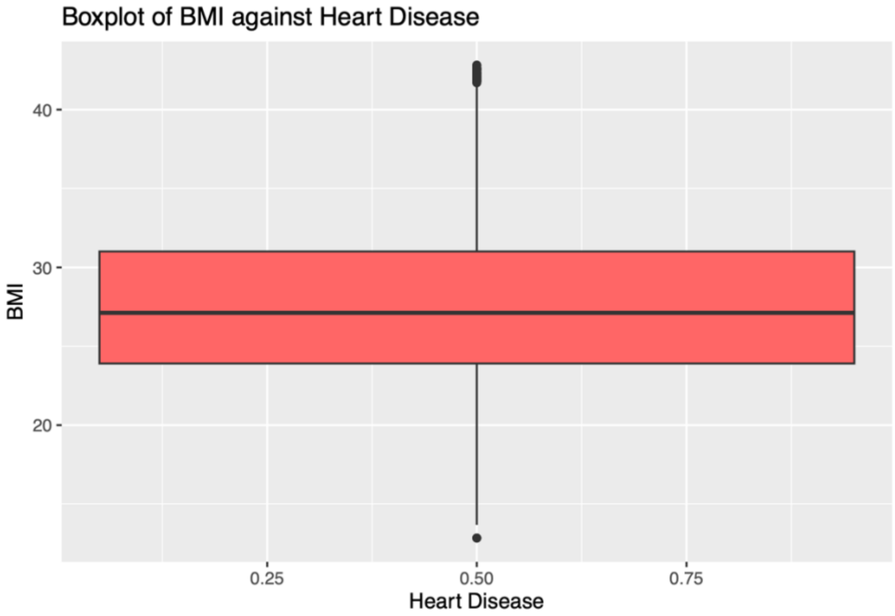


Figure 2. Boxplot of BMI against Heart Disease (After Removal of Outliers)

4. *Scaling of continuous variable:* The continuous variable “BMI” was scaled using “scale()” function. The function find the difference of each observation with the mean of the variable and find the product by standard deviation [3].

Data Visualisation

1. *Histogram of “HeartDiseae”:* Figure 3 shows the histogram of “HeartDisease” against frequency. This histogram helps us to identify whether the response variable, “HeartDisease” is balanced or imbalanced. Checking of balance is crucial since this step allows us to understand which metric is best for the classification problem.

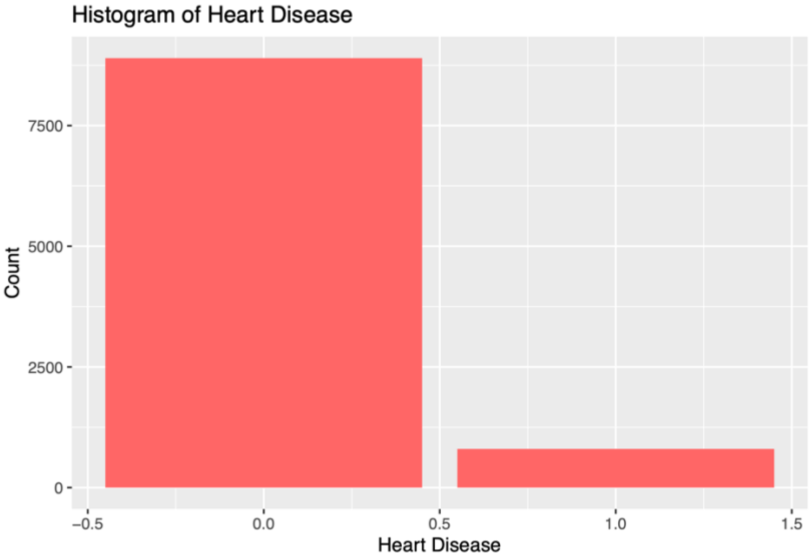


Figure 3. Histogram of Heart Disease

From Figure 3, it is crystal that the response variable is imbalanced due to which I'm aligned to choose F1-score as my calculation metric.

2. *Histogram of BMI:* The histogram of BMI aids to understand how the values in BMI looks like.

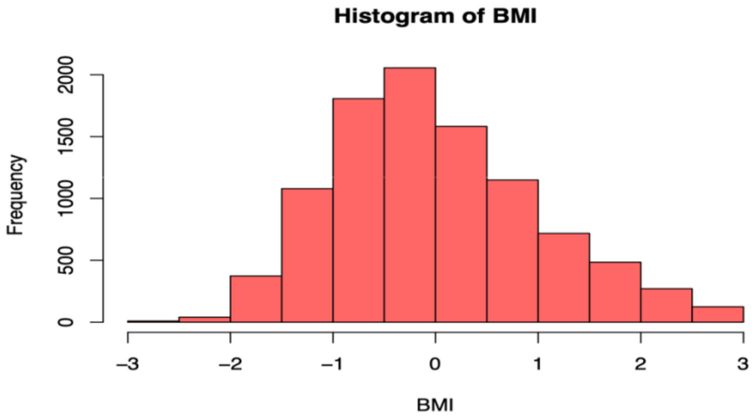


Figure 4. Histogram of BMI

With the help of Figure 4, we can see that the values in BMI are skewed towards the right.

3. *Plot Grid of Boxplots of Categorical Variables:* Figure 5 helps to visualise the remaining categorical variables with the help of boxplots.

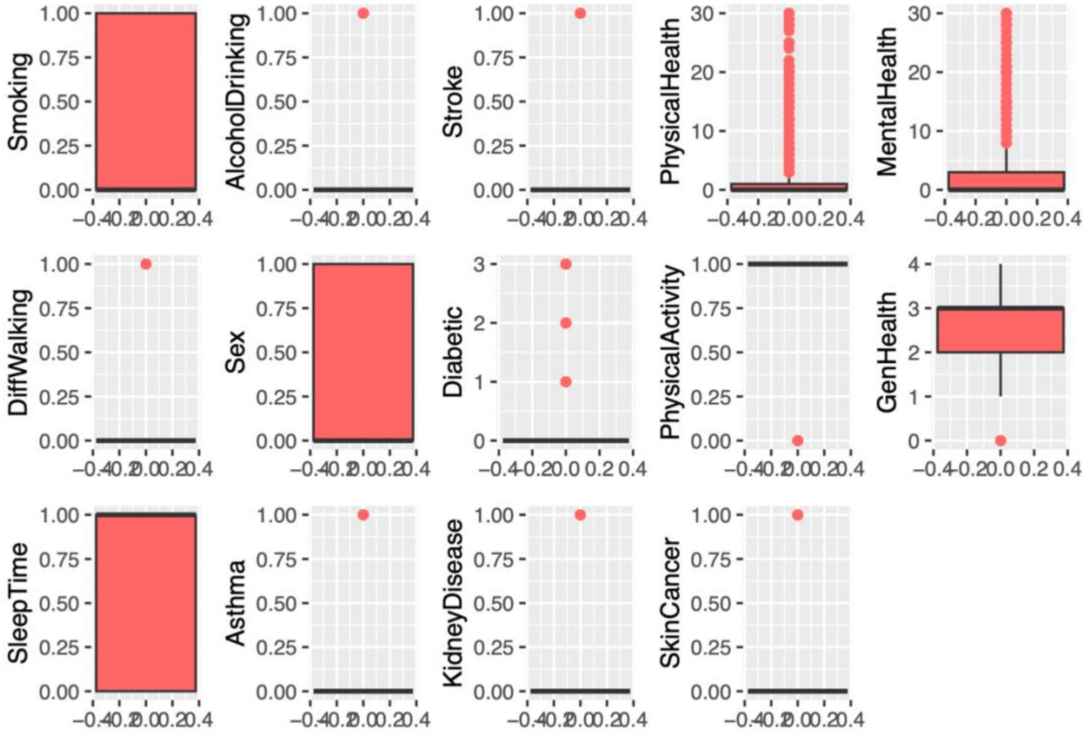


Figure 5. Plot Grid of Boxplots of Categorical Variables

Methodology

This study emphasis on comparing different classification models to predict Heart Disease Risk. Before performing any of the classification model training, the dataset sampled was divided into 70% training and 30% testing using the “createPartition()” function in the “caret” package. The five classification models for this project and they are:

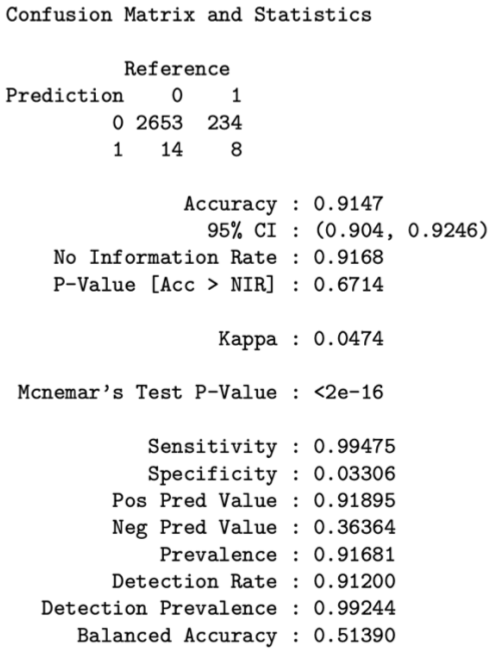
1. *K-Nearest Neighbours (KNN) Classifier:* The algorithm focuses on locating k nearest neighbours of a data point and assigning it to the class with the majority [4]. It is a simple, easy-to-use and widely used machine learning classification Algorithm.
   For this dataset, the KNN method applies the in-built package "caret", which includes the "train()" function [5]. In addition, the train() function contains the parameter "method", which can be assigned to "knn" to train the model using the training data. Finally, the results were predicted using the "predict()" function. To attain the F1 score, the built-in package "ML metric" was used, and the Confusion Matrix was printed using the "confusionMatrix()" function.
2. *Support Vector Machine (SVM) Classifier:* The algorithm works by separating the data into two classes by finding the hyperplane that maximizes the margin between these classes. It is a binary classifier [6].
   The SVM classifier was used on the dataset using the “svm()” function from the in-built package “e1071”. This is the easiest method to train the model using the training data. The prediction is performed using the test data. The F1 Score is calculated along with the Confusion Matrix.
3. *Random Forest Classifier:* Many decision trees are combined in the Random Forest Classifier ensemble approach to create a more precise and reliable prediction [7]. A random subset of the data and a random subset of the characteristics are used to train each tree in the forest.
   I used the in-built package “randomForest” to use the “randomForest()” function to train the model. Whereas the predict() function from caret package was used to make predictions using the test data. The F1 Score was printed along with the confusion matrix.
4. *Decision Tree Classifier:* A decision tree is a model that resembles a tree and represents a set of classification rules. The tree's nodes stand for decisions based on features, while the branches reflect the decisions' potential outcomes[8].
   For this classification method we use the built-in “rpart” package and the function “rparr()” to train the model. While predicting the result we “type = class” to predict the results using the test data. The confusion Matrix and the F1-score is printed.
5. *Naïve Bayes Classifier:* The probabilistic model, Naive Bayes, makes the assumption that each attribute of the data is independent of the others. Naive Bayes is based on Bayes' theorem, which asserts that the likelihood of the evidence given the hypothesis and the prior probability of the hypothesis are multiplied together to determine the probability of the hypothesis given the observed evidence [9].
   The “naivebayes” package is used to perform Naïve Bayes Classification. Here the “naiveBayes()” function is used train the model using the training data. Like in other models the predictions are made using “predict()” function. The F1- Score and the confusion matrix is printed.

The F1-Scores of all five classification models along with their names is loaded into a table which is used to create a histogram to clearly understand the difference between the F1-Scores of each model.

Results

The results of each classification models is printed using its confusion Matrix results and the F1-Score. The F1-score, which combines precision and recall into a single statistic, is the harmonic mean of these two metrics [10]. When you have an imbalanced dataset, where the proportion of instances in one class is significantly higher than the other, it is an excellent statistic to use to balance the weights assigned to precision and recall.
From Figure 3, we came to a conclusion that the response variable is imbalanced due to which the F1- Score is the best decision metric in this scenario.

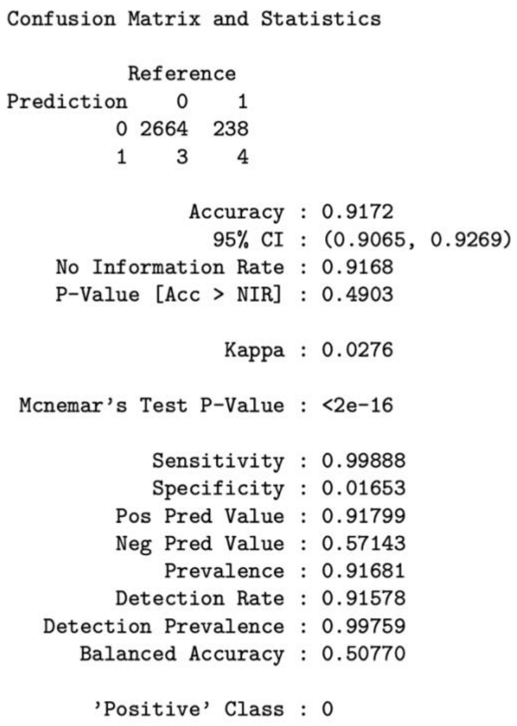
1. *KNN Classifier Results:* The KNN Classifier gave a F1-Score result of 0.9539521. The Figure 6 shows the results of the "confusionMatrix()" function.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2653  234
         1   14    8

               Accuracy : 0.9147
                 95% CI : (0.904, 0.9246)
    No Information Rate : 0.9168
    P-Value [Acc > NIR] : 0.6714

                  Kappa : 0.0474

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.99475
            Specificity : 0.03306
         Pos Pred Value : 0.91895
         Neg Pred Value : 0.36364
             Prevalence : 0.91681
         Detection Rate : 0.91200
   Detection Prevalence : 0.99244
      Balanced Accuracy : 0.51390
```

Figure 6. Results of KNN Classifier Confusion Matrix

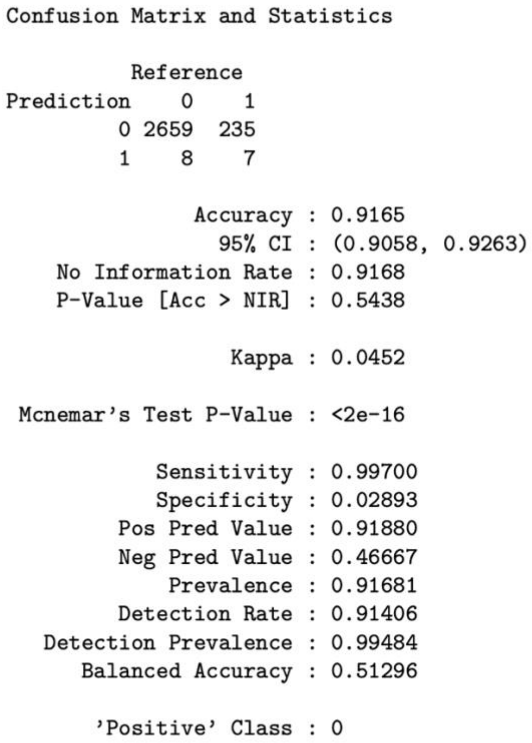
2. *SVM Classifier Results:* The SVM Classifier gave an F1-Score of 0.9560360. The results of the confusion matrix is shown in Figure 7.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2664  238
         1    3    4

               Accuracy : 0.9172
                 95% CI : (0.9065, 0.9269)
    No Information Rate : 0.9168
    P-Value [Acc > NIR] : 0.4903

                  Kappa : 0.0276

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.99888
            Specificity : 0.01653
         Pos Pred Value : 0.91799
         Neg Pred Value : 0.57143
             Prevalence : 0.91681
         Detection Rate : 0.91578
   Detection Prevalence : 0.99759
      Balanced Accuracy : 0.50770

       'Positive' Class : 0
```

Figure 7. Results of SVM Classifier Confusion Matrix

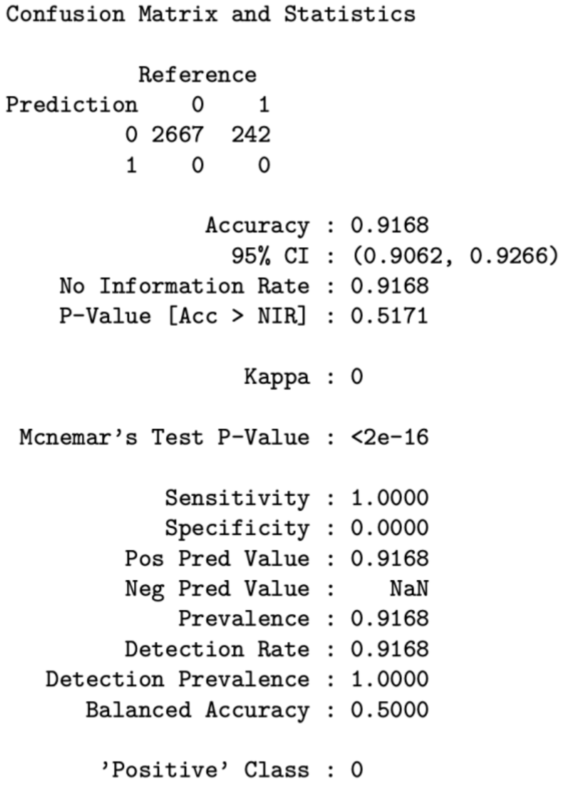
3. *Random Forest Classifier Results:* The Random Forest Classifier resulted in an F1- Score of 0.9558131. Figure 8 shows the results of confusion matrix.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2659  235
         1    8    7

               Accuracy : 0.9165
                 95% CI : (0.9058, 0.9263)
    No Information Rate : 0.9168
    P-Value [Acc > NIR] : 0.5438

                  Kappa : 0.0452

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.99700
            Specificity : 0.02893
         Pos Pred Value : 0.91880
         Neg Pred Value : 0.46667
             Prevalence : 0.91681
         Detection Rate : 0.91406
   Detection Prevalence : 0.99484
      Balanced Accuracy : 0.51296

       'Positive' Class : 0
```

Figure 8. Results of Random Forest Classifier Confusion Matrix

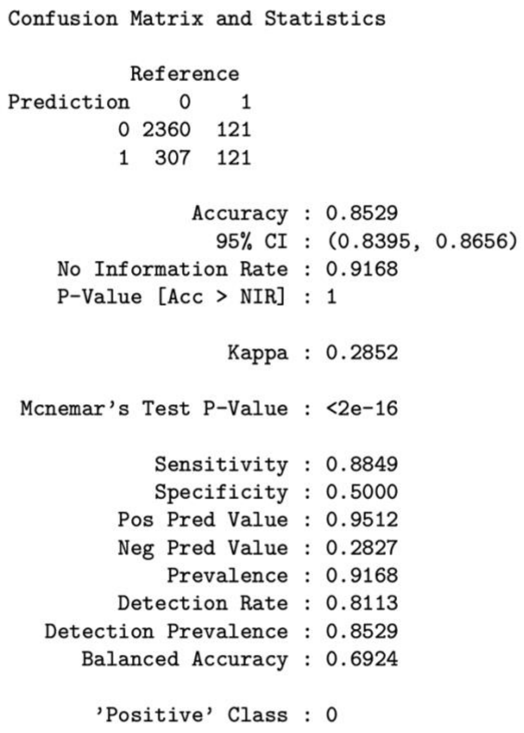
4. *Decision Tree Classifier Results:* The F1- Score of Decision Tree Classifier was 0.9557235. The results of confusion matrix is shown in Figure 9.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2667  242
         1    0    0

               Accuracy : 0.9168
                 95% CI : (0.9062, 0.9266)
    No Information Rate : 0.9168
    P-Value [Acc > NIR] : 0.5171

                  Kappa : 0

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.0000
            Specificity : 0.0000
         Pos Pred Value : 0.9168
         Neg Pred Value :    NaN
             Prevalence : 0.9168
         Detection Rate : 0.9168
   Detection Prevalence : 1.0000
      Balanced Accuracy : 0.5000

       'Positive' Class : 0
```

Figure 9. Results of Decision Tree Classifier Confusion Matrix

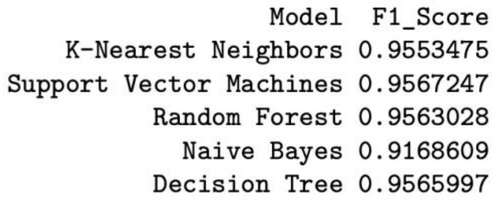
5. *Naïve Bayes Classifier Results:* The Naïve Bayes Classifier has a F1- Score of 0.9120470. The Figure 10 shows the confusion matrix results.



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 2360  121
         1  307  121

               Accuracy : 0.8529
                 95% CI : (0.8395, 0.8656)
    No Information Rate : 0.9168
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2852

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8849
            Specificity : 0.5000
         Pos Pred Value : 0.9512
         Neg Pred Value : 0.2827
             Prevalence : 0.9168
         Detection Rate : 0.8113
   Detection Prevalence : 0.8529
      Balanced Accuracy : 0.6924

       'Positive' Class : 0
```

Figure 10. Results of Naïve Bayes Classifier Confusion Matrix

The F1-Scores were used to create a data frame using the model names (Figure 11). The data frame was then used to create a histogram in Figure 11. This aids in understanding which classification models works best for this data set.



```
                 Model  F1_Score
   K-Nearest Neighbors 0.9553475
Support Vector Machines 0.9567247
         Random Forest 0.9563028
           Naive Bayes 0.9168609
         Decision Tree 0.9565997
```

Figure 11. F1-Scores of Classification Models

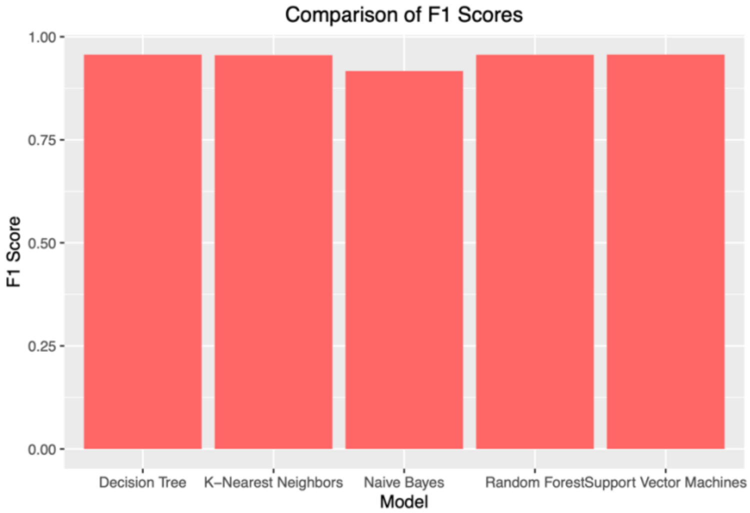


Figure 12. Histogram for the Comparison of F1-Scores

Conclusion

The initial goal of this study was to determine which classification model is the most effective at predicting the risk of heart disease in the dataset. Following the development of models and forecasts, it was determined that the Support Vector Machine performs the best at forecasting the risk of heart disease. This is because the F1-Score provided by the Support Vector Machine Classifier, which was the highest of all F1-Scores, was 0.9560360.
Also, it was noted that all of the F1-Scores were higher than 0.90. This can result from the dataset being initially sampled to only 10,000 for simpler model creation due to poor processing capacity on personal PCs.

References

[1] Kamil Pytlak. (2020). Personal Key Indicators of Heart Disease. Kaggle. Retrieved from https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease

[2] Jason Brownlee. "How to Calculate Outliers for Machine Learning in Python." Machine Learning Mastery, 2019, https://machinelearningmastery.com/how-to-calculate-outliers-for-machine-learning-in-python/.

[3] R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[4] Keller, J. M., Gray, M. R., & Givens, J. A. (1985). A fuzzy K-nearest neighbor algorithm. IEEE Transactions on systems, man, and cybernetics, (4), 580-585.

[5] Li, J., & Wang, W. (2017). Fast K-nearest neighbor search via locality-sensitive hashing. IEEE Transactions on Knowledge and Data Engineering, 29(7), 1374-1388.

[6] Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification. National Taiwan University.

[7] Liu, H., Li, Y., & Wong, L. (2008). A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. Genome informatics, 20(1), 57-68.

[8] Singh, R., Rani, R., & Singh, K. (2018). Decision tree algorithm for classification: A review. International Journal of Computer Science and Information Technologies, 9(5), 6455-6459.

[9] Rish, I. (2001). An empirical study of the Naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, pp. 41-46).

[10] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427-437. doi: 10.1016/j.ipm.2009.03.002