Machine Learning – An investigation into the early detection of heart disease using Machine Learning Technologies

14 January 2022

Table of Contents

Table of Tables

Table of Figures

List of Definitions and Abbreviations

1. WHO – World Health Organisation
2. ML – Machine Learning
3. DT – Decision Tree
4. RF – Random Forest
5. KNN – K-Nearest Neighbour
6. SMOTE - Synthetic Minority Oversampling Technique
7. DNN – Deep Neural Network

1 Introduction

According to the World Health Organization (2019), heart disease is one of the leading causes of premature death. Further to this, every year an estimated 17.9 million people die from a form of heart disease. One third of these deaths occur in people under 70 years of age. Common symptoms of the disease include fluttering in the chest, a racing heartbeat, chest pain or discomfort, shortness of breath and dizziness. (Mayo Clinic 2018)

The early detection of heart disease, therefore, is vital to preventing millions of premature deaths every year. This investigation aims to evaluate whether the detection of the disease can be improved by using Machine Learning algorithms.

The remainder of this report is structured as follows: In section 2, the dataset will be described and analysed for its quality. In section 3 will set out the method and aims of this investigation, section 4 will discuss the results. Finally, section 5 will conclude this investigation.

2 Data Analysis

2.1 Data Description

The dataset provided by Dr Heart contains 920 instances. There are 13 features included in the dataset. These are as follows:

1. Age
2. Gender
3. Chest Pain Type
4. Resting Blood Pressure
5. Serum Cholesterol
6. Fasting bloody Sugar
7. Resting Electrocardiographic
8. Maximum Heart Rate
9. Exercise Induced Angina
10. ST Depression
11. ST Segment
12. Number of Major Vessels
13. Thalassemia

Each feature is numeric. Each patient is assigned a class according to the absence or presence of coronary heart disease. This is in the form of an integer value (0 to 4) based on the severity of the disease.

2.1.1 Class Distribution

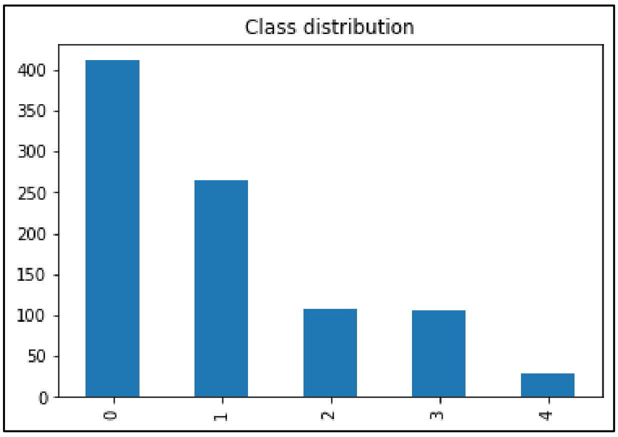


Figure 1 - Class Balance Bar Chart

The above figure shows the class balance for the severity of the patients' heart disease. Most patients do not have heart disease and only 28 patients are classified as having severe heart disease. 411 patients are classified as 0, 265 classified as 1, 109 classified as 3, 107 classified as 4 and 28 classified as 4. 44.67% of patients are not classified as having heart disease compared to 55.33% that are.

A full table of descriptive statistics for every feature in the dataset can be found in **A: Descriptive Statistics**. A key insight discovered from the mean of the classification is that on average a patient is classified as having mild heart disease (0.99). Another statistic of value is the variance of the Thalassemia feature. At 12272.39 this shows that the data for this feature is heavily dispersed.

2.2 Data Analysis

2.2.1 Missing Values:

In the dataset there is a total of 1752 missing values. These are spread across 10 of the features. The below table breaks down the number of missing values per feature. To apply ML algorithms to the dataset, these missing values will need to either be dropped or imputed. Due to the high number of missing values, dropping them is unviable as this would remove a large amount of the dataset rows. Therefore, the missing values will be imputed.

Table 1 - The number of missing values per feature

Feature	No. Missing Values
Age	0
Gender	0
Chest Pain Type	0
Resting Blood Pressure	59
Serum Cholesterol	29
Fasting Blood Sugar	90
Resting Electrocardiographic	2
Maximum Heart Rate	55
Exercise Induced Angina	55
ST Depression	62
ST Segment	307
Number of Major Vessels	609
Thalassemia	482

2.2.2 Duplicates

In the dataset there is two duplicated rows. These are rows 187 and 605.

Table 2 - Duplicated rows in dataset

	Age	Gender	Chest Pain Type	Resting Blood Pressure	Serum Cholesterol	Fasting blood sugar	Resting electrocardiographic	Maximum heart rate	Exercise induced angina	ST depression	ST segment	Number of major vessels	Thal	Class
187	58.0	1.0	3.0	150.0	219.0	0.0	1.0	118.0	1.0	0.0	NaN	NaN	NaN	2
605	49.0	0.0	2.0	110.0	NaN	0.0	0.0	160.0	0.0	0.0	NaN	NaN	NaN	0

2.2.3 Noise

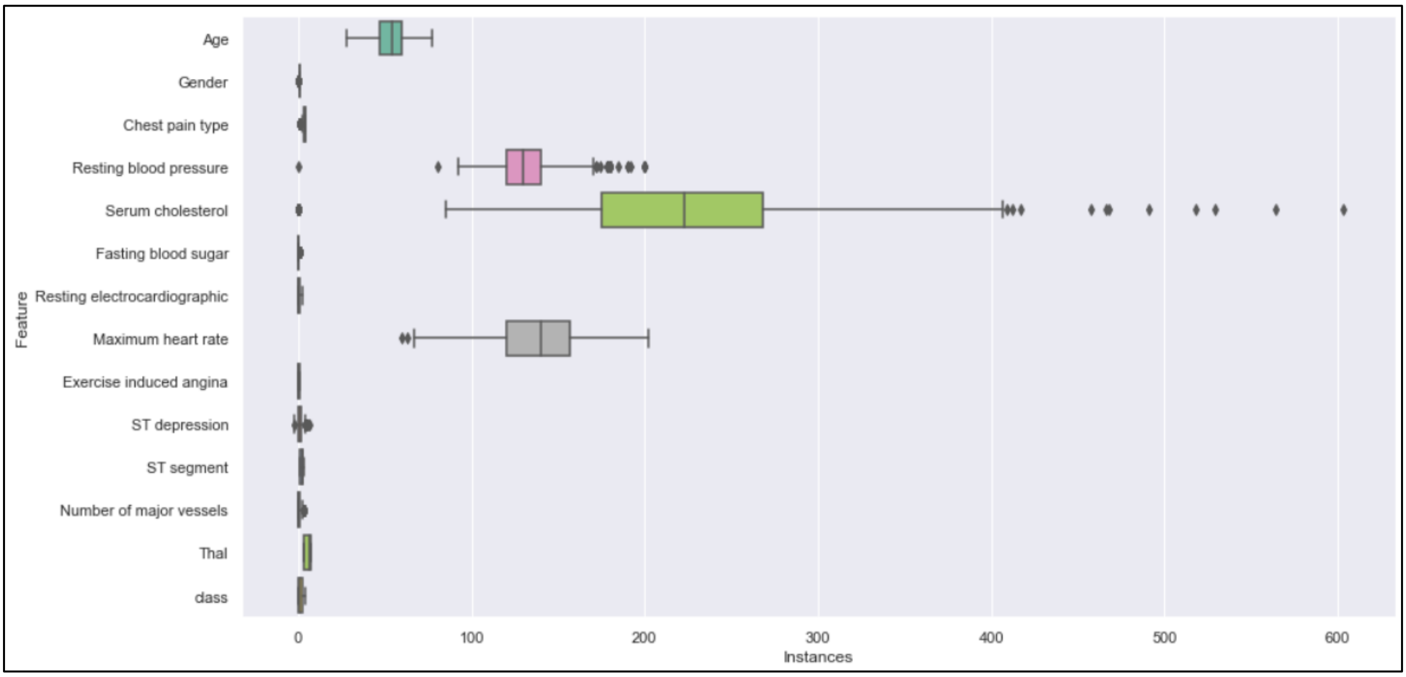


Figure 2 - Box Plots showing noise for each feature

The above box plots show noise for each feature in the dataset. **Figure 2** shows that the dataset is not very noisy. The features affected by noise are Serum Cholesterol, ST depression, Maximum heart rate and Resting blood pressure. The number of noise points for each of these features is low. Serum cholesterol has the greatest number of outliers.

2.2.4 Correlation Analysis

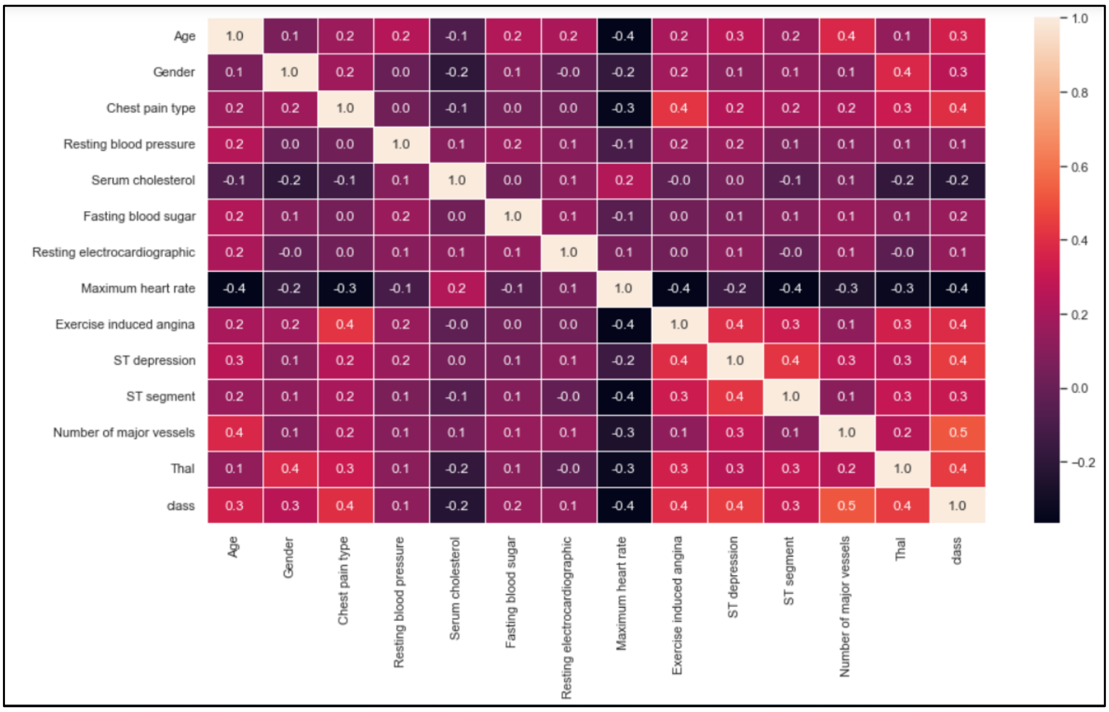


Figure 3 - Correlation Analysis of the Dataset

Performing a correlation analysis on the dataset shows that there are mainly weak correlations between the features. The Fasting blood sugar feature specifically has very low correlations – thus making it a feature likely to be dropped during feature selection. There is a 0.5 correlation between the number of major vessels and the class of heart disease. This suggests that there is a relationship between of significance between these two features. Some ML classifiers assume that features are independent such as Naive Bayes. If this was to be used, its accuracy could be affected by the features that are dependent on one another.

2.2.5 Clustering Analysis

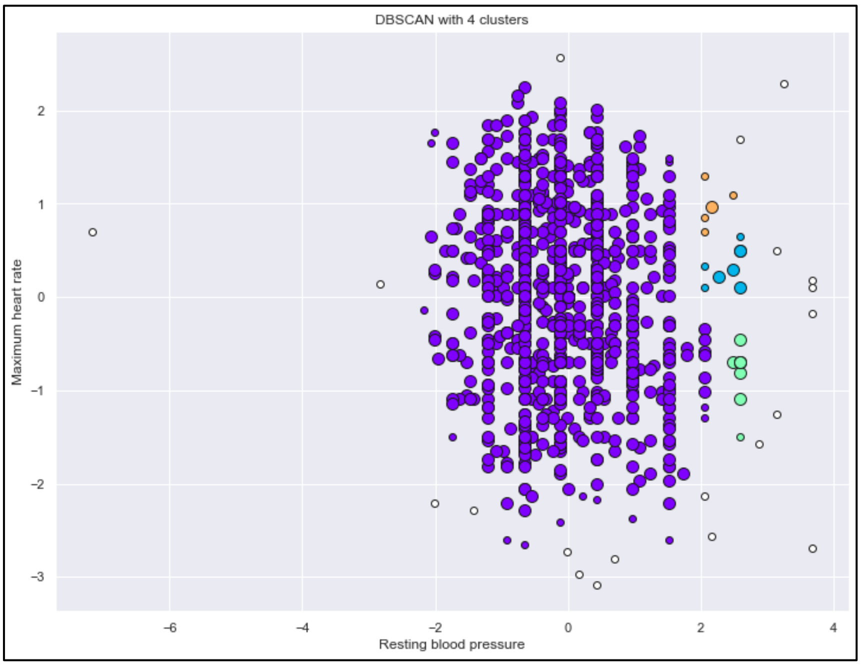


Figure 4 – DBSCAN Clustering Analysis of Resting blood pressure and Maximum heart rate features

For clustering analysis, DBSCAN was used. This was because this method visualises noise points as well as patterns in the data. For these features, 20 noise points were detected. This analysis shows that these two features are not correlated as there is one large cluster. Therefore, meaning these features are independent of one another.

2.2.6 Probability Distributions

A plot of each features' probability distribution can be found in **B: Probability distributions for all features**. One distribution of note is **Figure 8 - Probability Distribution for Age Feature**. This feature is normally distributed. In **Figure 10** - **Probability Distribution for Chest pain type feature**, the most probable type is number 4 (asymptomatic). Therefore, most patients have no symptoms of chest pain before it is identified. For the distributions, kernel density estimation was used as this method displays the distribution more efficiently than histograms. (riyaaggarwal 2020)

492 rows of the dataset include patients under the age of 70 classified as having a form of heart disease. This connects to the **WHO's statement** that most deaths from the disease occur in those under 70 years old. As this number of rows is over half of the dataset this suggests that the dataset is representable and accurate.

2.2.7 Feature Significance

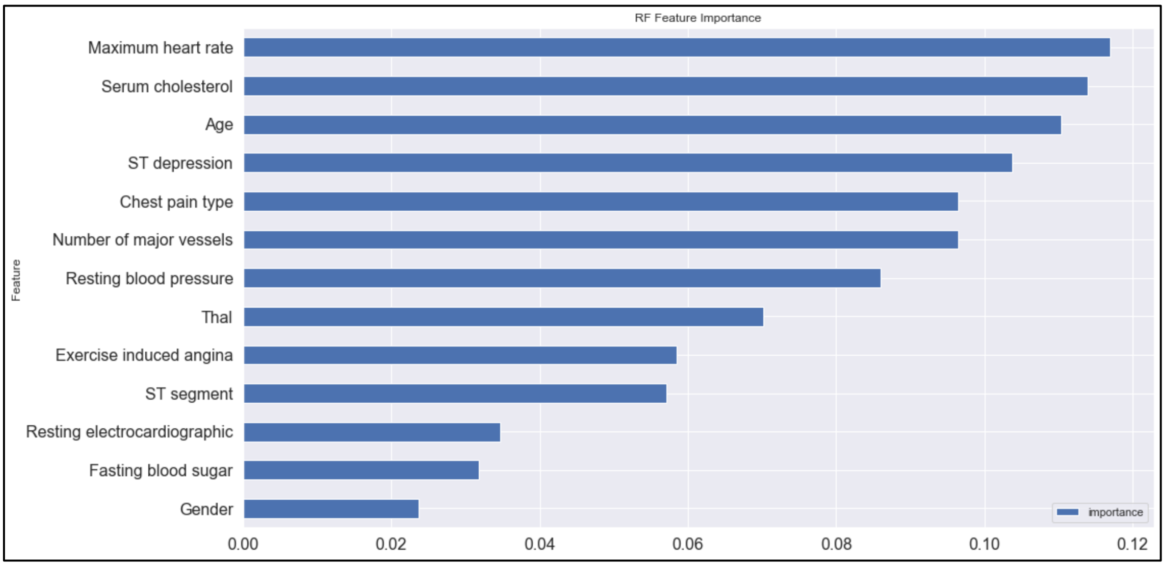


Figure 5 - Importance of each feature using Random Forest

Figure 5 shows that the most important feature in the dataset is maximum heart rate, closely followed by Serum cholesterol and age. It shows that gender is the least important feature. Also, of little importance are resting electrocardiographic and fasting blood sugar. This feature importance analysis will impact which features are selected in Error! Reference source not found.**.** This is because including every feature can r educe accuracy and increase computational time for ML classifiers.

3 Method

3.1 Aims and Objectives

Primary Objective: Successfully apply machine learning classifiers to classify whether a patient has heart disease. For this investigation to be successful an accuracy of greater than 70% must be achieved with each classifier also achieving at least 80% precision.

Secondary Objective: Identify the optimum method of feature selection.

3.2 Data Processing

3.2.1 Missing Values

All features in the dataset are numeric, however, the data is both numeric and categorical. Therefore, values will be imputed using the mean for numeric data and the mode for categorical data. Imputing using the mean for categorical data is unviable as it would create decimal numbers that are not part of the categories for each feature thus making imputing using the mode is more suitable. Numeric data will be imputed using the mean as this is computationally efficient.

3.2.2 Noise

The dataset is not free from noise - however there is little noise in the data. Therefore, the noise will not be processed.

3.2.3 Duplicates

There are only two duplicated rows in the dataset. Therefore, both rows will be dropped from the dataset as this will increase accuracy when running ML classifiers on the data.

3.2.4 Dimensionality / Feature Selection

Based on the feature importance in **2.2.7**, the dimensionality of the dataset will be reduced when running ML classifiers. This will reduce computational time when running the classifiers.

<u>Experiment 1 – Random Forest Feature Selection:</u>

Random Forest will be used for this experiment. This method will be used because the features were ranked using RF and this feature selection is highly interpretable, has low overfitting and has good predictive performance. (Dubey 2018)

<u>Experiment 2 – Recursive feature elimination:</u>

This experiment will use Recursive feature elimination. This will give a direct comparison to RF feature selection and allow the secondary objective to be evaluated.

3.2.5 Scaling / Transforming Data

Prior to running classifiers, the dataset will be scaled to improve classifier performance. The Standard Scaler from Scikit Learn (2019a) will be used. This is because running classifiers that use distance metrics are sensitive to data with wide ranges. The Standard Scaler ensures accuracy is not affected by these ranges.

3.2.6 Class Imbalance

The class imbalance identified in **2.1** would affect classifier performance if not handled correctly. Therefore, Synthetic Minority Oversampling Technique or SMOTE will be used. This technique will be used as it oversamples minority classes randomly by replicating instances. SMOTE will increase performance when running classifiers on the data.

3.3 Classifiers and Configuration

3.3.1 Classifiers

The classifiers chosen for this investigation are:

Decision Tree: A tree is constructed of a root node, decision nodes and leaf nodes. The root node is selected from Attribute selection measure. This measure is repeated until a leaf node cannot be split into sub-nodes. (Hemanth Chowdary 2020) This classifier will be used as it is computationally efficient and easy to interpret. (Python Programming Language 2021)

K-Nearest Neighbour: This classifier is a supervised algorithm that is instance based. It works by checking each data point and looking for ‘K’ number of nearest neighbours. Test data is assigned the class most frequently occurring in the ‘K’ nearest neighbours. This classifier will be used as it makes highly accurate predictions. (IBM 2020)

Random Forest: Another supervised algorithm, Random Forest is constructed using multiple decision trees. It uses ensemble learning, which is a technique that combines many classifiers. The outcome of the classifier is based on the prediction of the decision trees. Random Forest provides greater accuracy than the decision tree algorithm and produces good predictions (Mbaabu 2020) – hence its inclusion in this investigation.

Deep Neural Network: This type of neural network is an artificial neural network with multiple hidden layers between the input and output layers. This is implemented using the Scikit Learn MLP Classifier. (Scikit Learn 2010) Using a DNN increases accuracy as it can handle large amounts of data well. (Mahapatra 2019)

3.3.2 Configuration

The hyperparameters for these classifiers will be tuned using the Grid Search from Scikit Learn (2019b). A Grid Search will be used because it completes an exhaustive search of the grid of hyperparameter values. This means that every single combination of hyperparameter values is checked – resulting in accurate predictions for hyperparameter values to use for each classifier. For each classifier, a grid search will be completed using a pre-defined parameter dictionary.

To ensure fairness between ML classifiers, the number of folds will stay the same for each classifier. This will be the same for the random state – which will be kept at none for each classifier that includes the hyperparameter.

3.4 Validation Method

The chosen validation method for this investigation is Cross-Validation. The strategy used will be K-Folds. The number of folds used will be 10. Cross-Validation will be used because using K-Folds increases the amount of testing based on the same amount of data, thus improving the accuracy of performance metrics and model accuracy. (Shulga 2018) Furthermore, when using classifiers on multiple test sets, confidence in algorithm performance increases as more results are created.

3.5 Evaluation Metrics

The evaluation metrics used in this investigation are Accuracy, Precision, Recall and Confusion Matrix. Also evaluated will be training and testing time. These metrics will be used because they will allow for a critical evaluation of ML classifier performance and efficiency. The use of a confusion matrix for each classifier will allow for direct comparisons of true positives and false positives and true and false negatives.

4 Results and Discussion

Experiment 1 Results

During experiment 1, RF feature selection was performed before running classifiers on the dataset. The results of running the classifiers are shown in the below table. The features removed were Gender, Fasting Blood sugar and Resting electrocardiographic.

Table 3 - Classifier Performance (RF Feature Selection)

Classifier	Accuracy Score	Precision Score	Recall Score	Train Time	Test Time
Decision Tree	71.0%	83.1%	82.9%	0.004s	0.002s
Random Forest	81.2%	90.8%	88.6%	1.27s	0.03s
K-Nearest Neighbour	80.2%	93.3%	84.8%	0.003s	0.007s
Deep Neural Network	79.2%	90.8%	86.3%	7.63s	0.004s

From **Table 3** it is clear to see that all four classifiers performed well on the dataset – all achieving an accuracy score of over 70% and precision of over 80% - satisfying the main objective of this investigation. For precision and recall, this was over 80% for all classifiers. The most accurate classifier in this experiment was Random Forest. The great performance of this classifier is likely down to the numerical nature of the data and repeated decision trees performing better than the one used in the DT classifier – explaining the 10.2% increase in accuracy over DT. The KNN classifier was the most precise and Random Forest also had the highest recall score.

In terms of computational performance, the DT classifier was trained and tested in the least amount of time – making it the most efficient classifier. The DNN classifier took the longest to train and test but scored better than the DT classifier – but it was still the least efficient classifier. However, the DNN classifier is the most complex and included 30 hidden layers and 5000 maximum iterations, explaining the high training time. RF performed well again, taking just 1.3s to test and train the data, however this is considerably longer than the 0.01s needed to test and train the KNN classifier.

A full appendix of T-tests can be found in **C: T-Tests for Statistical Significance**. A key insight from these tests is that only the DT classifier performance is significantly different (and lower) compared to the other classifiers. RF, KNN and DNN had performance that was similar between the three. From this it can be deduced that the DT classifier performed the worst.

Experiment 2 Results

The following table shows the results from experiment 2 - where recursive feature elimination was used to select features. The features removed were ST Segment, Exercise induced angina and Number of major blood vessels.

Table 4 - Classifier Performance (Recursive Feature Elimination)

Classifier	Accuracy Score	Precision Score	Recall Score	Test Time	Train Time
Decision Tree	70.8%	84.3%	85.7%	0.003s	0.002s
Random Forest	80.8%	90.2%	88.2%	1.28s	0.03s
K-Nearest Neighbour	79.2%	94.1%	83.5%	0.003s	0.008s
Deep Neural Network	78.8%	90.8%	86.0%	7.15s	0.003s

All four of the classifiers again scored an accuracy of over 70% and precision over 80%, thus meaning the main objective was also satisfied in this experiment. RF was again the most accurate classifier, whilst KNN was again the most precise. RF also has the highest recall score of 88.2%. The classifiers rank the same in accuracy as in experiment 1.

In terms of computational performance, the classifiers were trained and tested within milliseconds of the corresponding times in experiment 1. DT again being the most efficient and DNN the least efficient.

The same can be deduced for the T-tests for experiment 2 in that only the DT classifier had performance significantly different and lower when compared to the other classifiers. Again, it can be deduced that the DT classifier performed the worst in this experiment.

Feature Selection Method Comparison

Table 5 - Comparison of Feature Selection Methods on Accuracy Score

Classifier	**Accuracy Score Experiment 1 (RF Feature Selection)**	**Accuracy Score Experiment 2 (Recursive Feature Elimination)**	**Difference**
Decision Tree	71.0%	70.8%	0.2%
Random Forest	81.2%	80.8%	0.4%
K-Nearest Neighbour	80.2%	79.2%	1.0%
Deep Neural Network	79.2%	78.8%	0.4%

The secondary objective of this investigation was to determine which method of feature selection was the optimum. Every classifier achieved a higher accuracy score when RF feature selection was used. The greatest difference in accuracy was 1% for the KNN classifier. Whilst the difference in accuracy was not large for any of the classifiers, Random Forest feature selection still performed better than recursive feature elimination across all classifiers. From these results, it can be determined that RF feature selection was better than Recursive Feature Elimination.

Issues with high accuracy and misclassification of minor classes

Whilst it is positive that all classifiers in both experiments reached over 70% accuracy, it is important to note that in a real-world application of classifying if a patient has heart disease, it would be ideal to have accuracy greater than 90% as it could be the difference in someone being diagnosed or not – potentially leading to an unnecessary death. However, having high accuracy does not always mean that the classifier performance is perfect, the percentage of false positives/negatives would also need to be considered to ensure the classification is precise too.

Another issue in a real-world application is misclassification of minority classes. In this dataset class 0 was the majority class which would mean the classifiers may have been biased towards that class. Using SMOTE to balance the classes improved classifier performance but would not have eliminated all potential bias. In the real-world, this could lead to classes identifying the presence of heart disease being misclassified and patients' heart-disease class being incorrect.

Confusion Matrices

Figure 6 shows each confusion matrix for the classifiers used in experiment 1. In each matrix, it is noteworthy that the classification of class 1 has the lowest number of true positives. Class 1 is a minority class and was oversampled during data processing, however, was not the class with the smallest number of instances. This is the same in **Figure 7** – suggesting that neither feature selection method was preferable for this class. Class 4 was always the class with the highest number of true positives across both experiments.

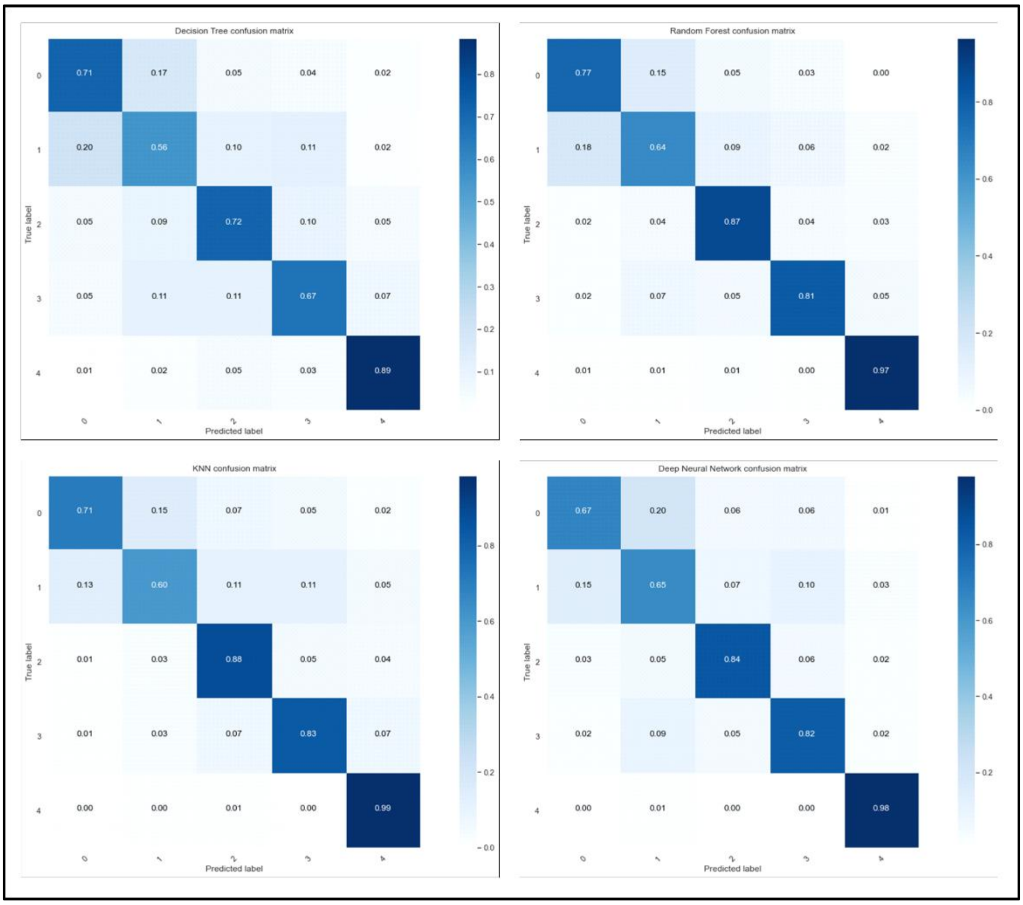


Figure 6 - Confusion Matrices for Experiment 1 Classifiers

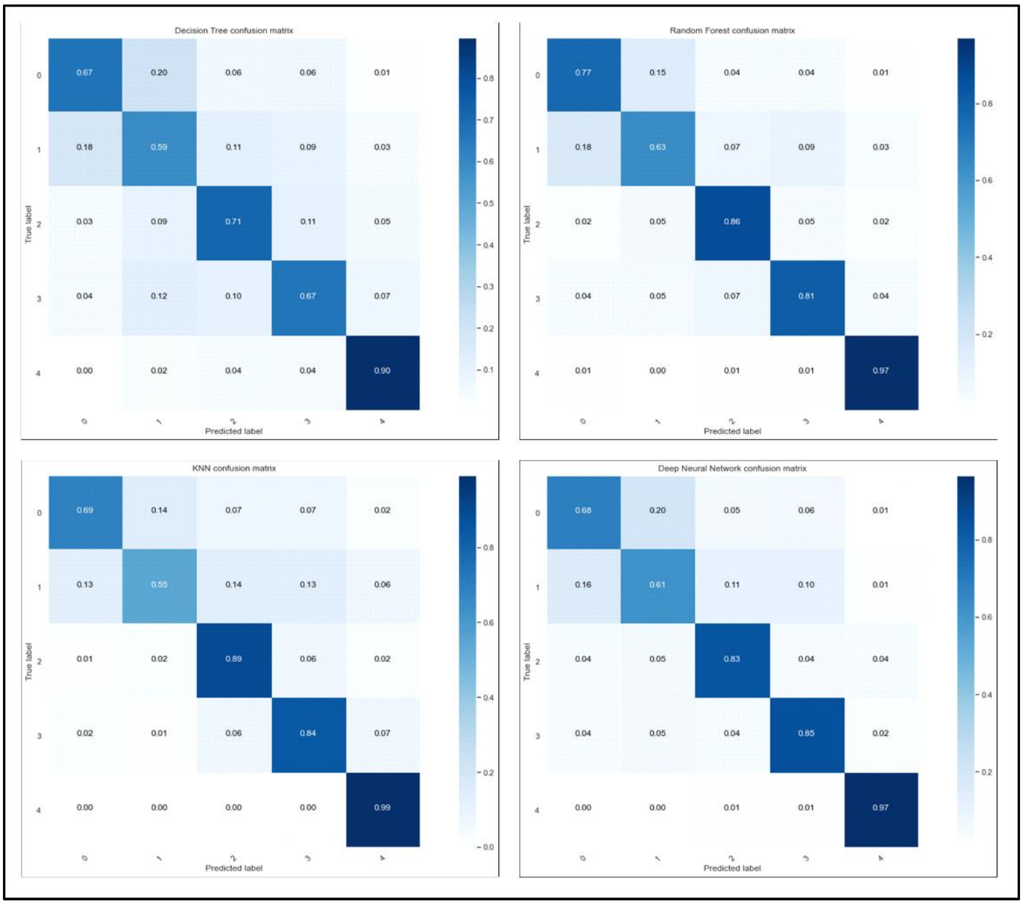


Figure 7 - Confusion Matrices for Experiment 2 Classifiers

5 Conclusions and Further Work

In conclusion this investigation was successful as the primary objective of achieving classifications with 70% accuracy and 80% precision was achieved. The detection of heart disease is critical in saving lives and application of these classifiers could identify an early diagnosis.

It is clear from the results that the decision tree classifier performed the worst. It scored the lowest accuracy, precision and recall in both experiments. RF, KNN and DNN all performed similar and were statistically similar in terms of performance. However, KNN and RF were much more efficient in terms of computational efficiency. When classifying whether a patient has heart disease, it is vital that each classification is as accurate as possible. Therefore, as the Random Forest classifier achieved the highest accuracy, it can be concluded that this was the best performing classifier.

It can also be concluded that Random Forest feature selection was the optimum feature selection method as classifiers run after RF feature selection all performed better than after Recursive feature elimination.

To improve this investigation, more classifiers could be used on the dataset to get a wider range of results that could be analysed. Furthermore, more methods of feature selection could be experimented with to judge differences in performance. Another improvement would be to use more performance metrics to improve the comparison of ML classifiers.

Word Count: 2966 (Headers)

Word Count: 2885 (Excluding Headers)

References

World Health Organization, 2019. Cardiovascular Diseases [online]. Who.int. Available from: https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1 [Accessed 14 Nov 2021].

Mayo Clinic, 2018. Heart Disease - Symptoms and Causes [online]. Mayo Clinic. Available from: https://www.mayoclinic.org/diseases-conditions/heart-disease/symptoms-causes/syc-20353118 [Accessed 14 Nov 2021].

riyaaggarwal, 2020. Density Plots with Pandas in Python. GeeksforGeeks [online]. Available from: https://www.geeksforgeeks.org/density-plots-with-pandas-in-python/ [Accessed 19 Dec 2021].

Dubey, A., 2018. Feature Selection Using Random forest [online]. Towards Data Science. Available from: https://towardsdatascience.com/feature-selection-using-random-forest-26d7b747597f [Accessed 21 Dec 2021].

Scikit Learn, 2019a. sklearn.preprocessing.StandardScaler [online]. Scikit-learn.org. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html [Accessed 23 Dec 2021].

Hemanth Chowdary, D., 2020. Decision Trees Explained with a Practical Example [online]. Towards AI. Available from: https://towardsai.net/p/programming/decision-trees-explained-with-a-practical-example-fe47872d3b53 [Accessed 23 Dec 2021].

Python Programming Language, 2021. Advantages of a Decision Tree for Classification - Python [online]. pythonprogramminglanguage.com. Available from: https://pythonprogramminglanguage.com/what-are-the-advantages-of-using-a-decision-tree-for-classification/ [Accessed 23 Dec 2021].

IBM, 2020. Usage of k-Nearest Neighbors (KNN) [online]. www.ibm.com. Available from: https://www.ibm.com/docs/en/ias?topic=knn-usage [Accessed 23 Dec 2021].

Mbaabu, O., 2020. Introduction to Random Forest in Machine Learning [online]. Section. Available from: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/ [Accessed 23 Dec 2021].

Scikit Learn, 2010. sklearn.neural_network.MLPClassifier [online]. Scikit-learn.org. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html [Accessed 23 Dec 2021].

Mahapatra, S., 2019. Why Deep Learning over Traditional Machine Learning? [online]. Towards Data Science. Available from: https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063 [Accessed 23 Dec 2021].

Scikit Learn, 2019b. sklearn.model_selection.GridSearchCV [online]. Scikit-learn.org. Available from: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html [Accessed 23 Dec 2021].

Shulga, D., 2018. 5 Reasons why you should use Cross-Validation in your Data Science Projects [online]. Towards Data Science. Available from: https://towardsdatascience.com/5-reasons-why-you-should-use-cross-validation-in-your-data-science-project-8163311a1e79 [Accessed 23 Dec 2021].

A: Descriptive Statistics

The following table contains descriptive statistics for each feature. Note that the mode has been calculated including missing values, as these have not yet been cleaned from the dataset.

Table 6 - Descriptive Statistics for all Features

		Feature												
		Age	Gender	Chest Pain Type	Resting Blood Pressure	Serum Cholesterol	Fasting Blood Sugar	Resting Electrocardiographic	Maximum Heart Rate	Exercise Induced Angina	ST Depression	ST Segment	Number of Major Vessels	Thalassemia
Statistic	Mean	53.51	0.79	3.25	132.13	199.13	0.17	0.60	137.55	0.39	0.88	1.77	0.68	5.09
	Median	54.0	1.0	4.0	130.0	223.0	0.0	0.0	140.0	0.0	0.5	2.0	0.0	6.0
	Mode	54.0	1.0	4.0	120.0	0.0	0.0	0.0	NaN	0.0	0.0	2.0	NaN	NaN
	Max	77.0	1.0	4.0	200.0	603.0	1.0	2.0	202.0	1.0	6.2	3.0	3.0	7.0
	Min	28	0.0	1.0	0.0	0.0	1.0	0.0	60.0	0.0	-2.6	1.0	0.0	3.0
	Standard Deviation	9.42	0.41	0.93	19.07	110.78	0.37	0.81	25.93	0.49	1.09	0.62	0.94	1.92
	Variance	88.82	0.17	0.87	363.52	12272.39	0.14	0.65	672.17	0.24	1.19	0.38	0.88	3.68

The following table contains descriptive statistics for the classification of each patient.

Table 7 - Descriptive Statistics for Patient Classification

Statistic	Value
Mean	0.99
Median	1.0
Mode	0
Max	4
Min	0
Standard Deviation	1.14
Variance	1.30

B: Probability distributions for all features

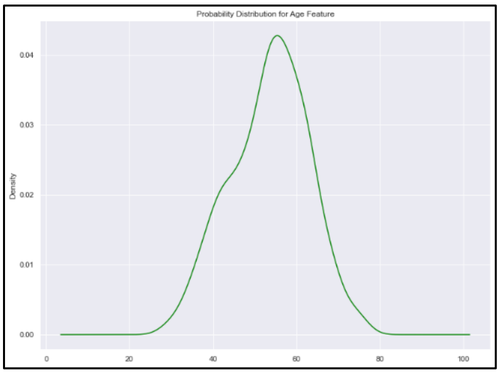


Figure 8 - Probability Distribution for Age Feature

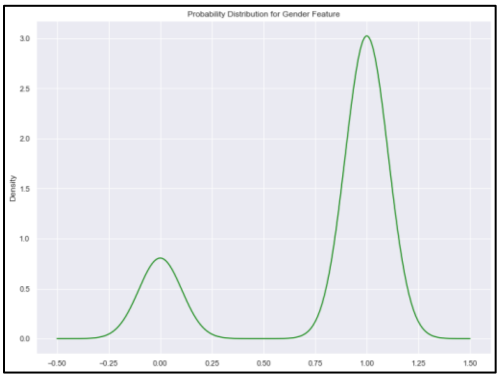


Figure 9 - Probability Distribution for Gender Feature

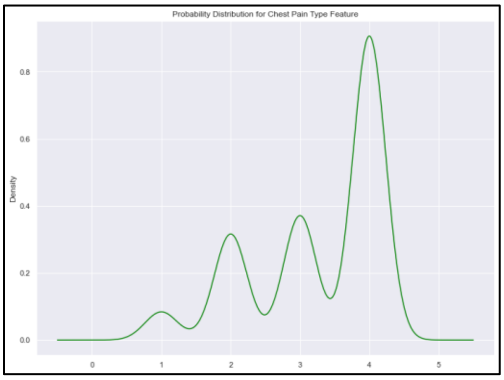


Figure 10 - Probability Distribution for Chest pain type feature

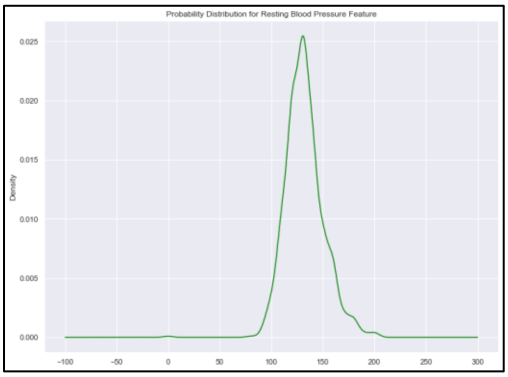


Figure 11 - Probability Distribution for Resting blood pressure feature

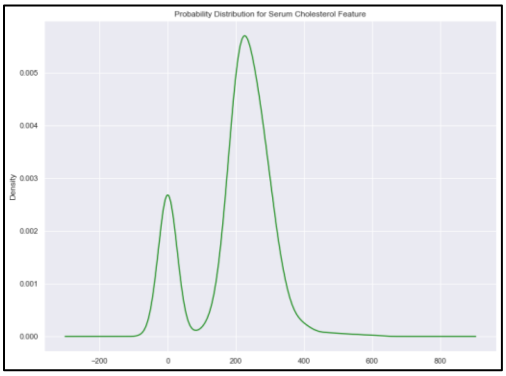


Figure 12 - Probability Distribution for Serum cholesterol feature

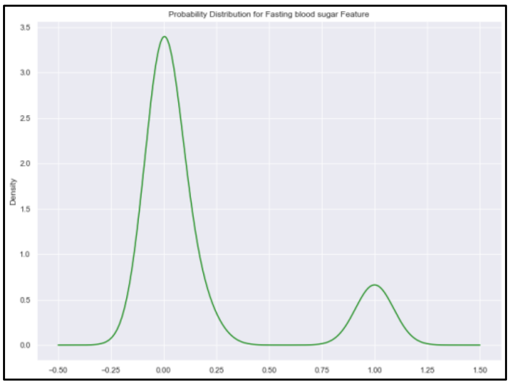


Figure 13 - Probability Distribution for Fasting blood sugar feature

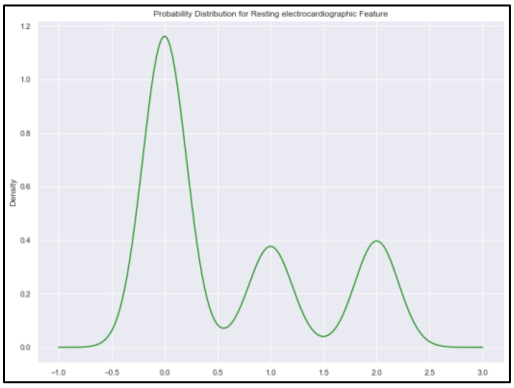


Figure 14 - Probability Distribution for Resting electrocardiographic feature

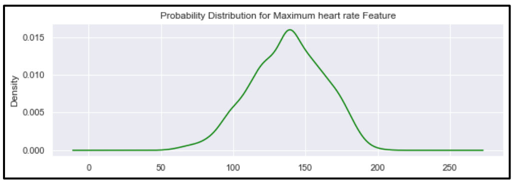


Figure 15 - Probability Distribution for Maximum heart rate feature

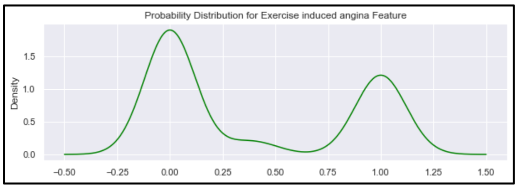


Figure 16 - Probability Distribution for Exercise induced angina feature

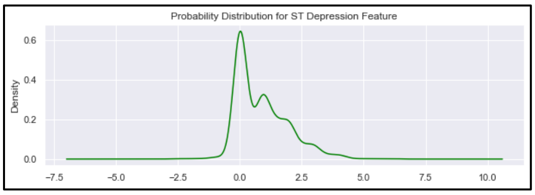


Figure 17 - Probability Distribution for ST depression feature

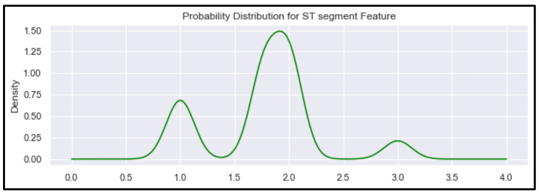


Figure 18 - Probability distribution for ST segment feature

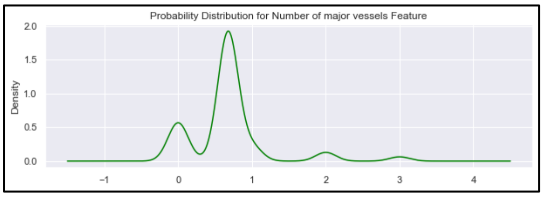


Figure 19 - Probability Distribution for Number of major vessels feature

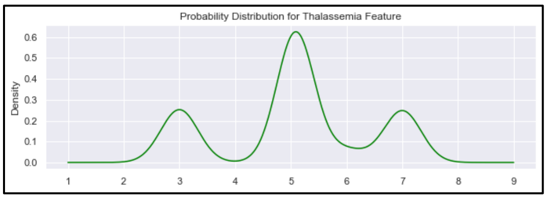
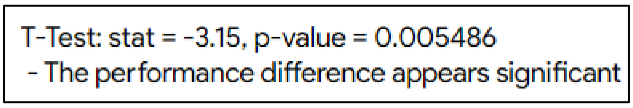


Figure 20 - Probability Distribution for Thalassemia feature

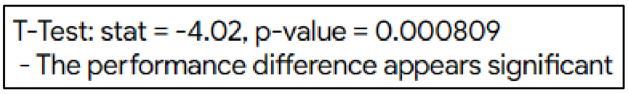
C: T-Tests for Statistical Significance

Experiment 1:



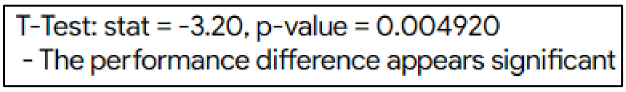
```
T-Test: stat = -3.15, p-value = 0.005486
 - The performance difference appears significant
```

Figure 21 - T-test for DT vs RF (Experiment 1)



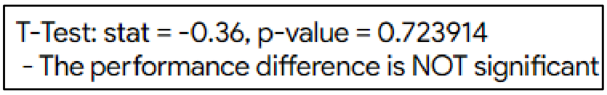
```
T-Test: stat = -4.02, p-value = 0.000809
 - The performance difference appears significant
```

Figure 22 - T-test for DT vs KNN (Experiment 1)



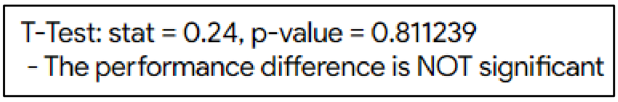
```
T-Test: stat = -3.20, p-value = 0.004920
 - The performance difference appears significant
```

Figure 23 - T-test for DT vs DNN (Experiment 1)



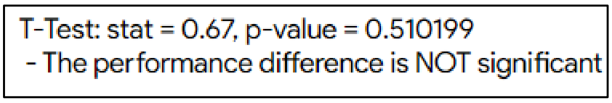
```
T-Test: stat = -0.36, p-value = 0.723914
 - The performance difference is NOT significant
```

Figure 24 - T-test for RF vs KNN (Experiment 1)



```
T-Test: stat = 0.24, p-value = 0.811239
 - The performance difference is NOT significant
```

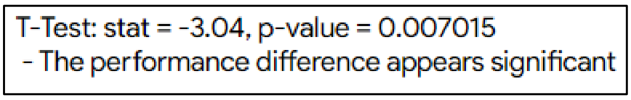
Figure 25 - T-test for RF vs DNN (Experiment 1)



```
T-Test: stat = 0.67, p-value = 0.510199
 - The performance difference is NOT significant
```

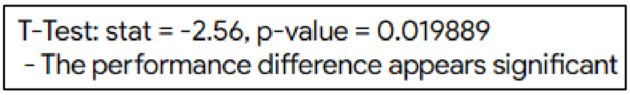
Figure 26 - T-test for KNN vs DNN (Experiment 1)

Experiment 2:



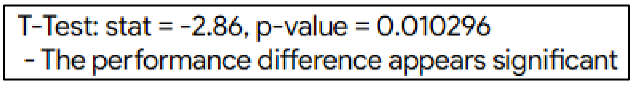
```
T-Test: stat = -3.04, p-value = 0.007015
 - The performance difference appears significant
```

Figure 27 - T-test for DT vs RF (Experiment 2)



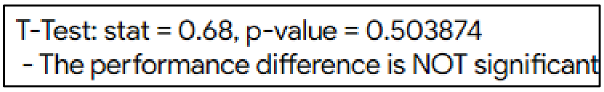
```
T-Test: stat = -2.56, p-value = 0.019889
 - The performance difference appears significant
```

Figure 28 - T-test for DT vs KNN (Experiment 2)



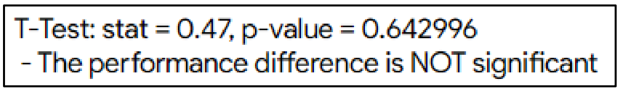
```
T-Test: stat = -2.86, p-value = 0.010296
 - The performance difference appears significant
```

Figure 29 - T-test for DT vs DNN (Experiment 2)



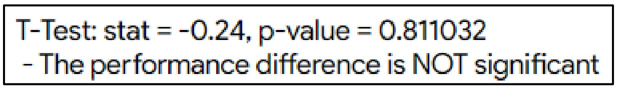
```
T-Test: stat = 0.68, p-value = 0.503874
 - The performance difference is NOT significant
```

Figure 30 - T-test for RF vs KNN (Experiment 2)



```
T-Test: stat = 0.47, p-value = 0.642996
 - The performance difference is NOT significant
```

Figure 31 - T-test for RF vs DNN (Experiment 1)



```
T-Test: stat = -0.24, p-value = 0.811032
 - The performance difference is NOT significant
```

Figure 32 - T-test for KNN vs DNN (Experiment 2)