Project 3: Applying Your Skills

Group Members - Sruti Munukutla and Mahvash Maghrabi

Github - https://github.com/skeerti2/ML_Project3

Working Pattern -

Task 1, 4 - Mahvash Maghrabi
Task 2, 3 - Sruti Munukutla

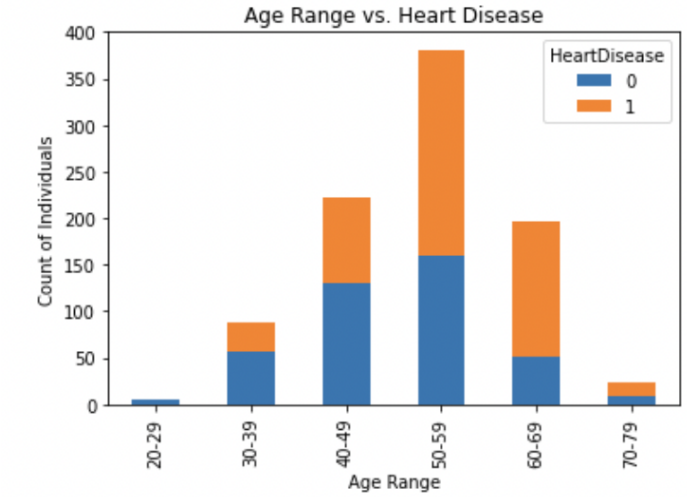
About the Dataset
The data set provided was the heart disease detection dataset. In this dataset the patients were classified as having or not having heart disease based on the independent variables Age, Sex, ChestPainType, FastingBS, RestingECG, MaxHR, Exercise Angina, Oldpeak, ST_Slope. The dependent variable Heart Disease specifies whether the person has the disease or not. The training and testing dataset was also provided along with the original dataset.

Task 1:
Data Cleaning
The first step towards the project was to clean the dataset to make a more accurate Machine Learning model. The dataset had categorical values which were converted to numerical values using Label Encoding. The columns Sex, ChestPainType, RestingECG, ExcerciseAngina, ST_Slope all has categorical values which were converted to numerical values accordingly.
The next step was to remove outliers from the dataset. The columns RestingBP, Cholesterol, MaxHR, Oldpeak had numerous outliers. They were removed using the IQR method but after removing the outliers and plotting the box plot the plot had no boxes and skewers which clearly depicted that data in the column may not be suitable for analysis as the dataset is small to remove any outliers. Hence, removing the outliers functionality was neglected for the project.
The dataset was checked for any null values and if any duplicates are presnet in the dataset. The same data cleaning methods were applied on the test and train dataset as well.

Data Visualization
The next step was visualizing the dataset and looking at the contribution of each independent variable towards the dependent variable. Bar chart plotting was used for this. Stacked and Grouped bar charts were used for visualization and analysis.

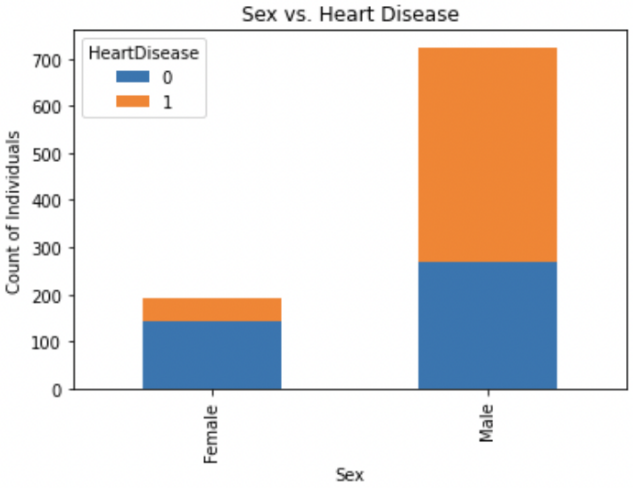
Plotting Age vs. Heart Disease

The bar plot below shows the count of individuals within a particular age range having a heart disease or not. It is a stacked bar plot where blue shows No heart disease and Orange shows Heart disease. It shows that the people in the age range of 50-59 have the higher number of people having heart disease as well as people in the age range of 60-69 have higher number of people having a heart disease.



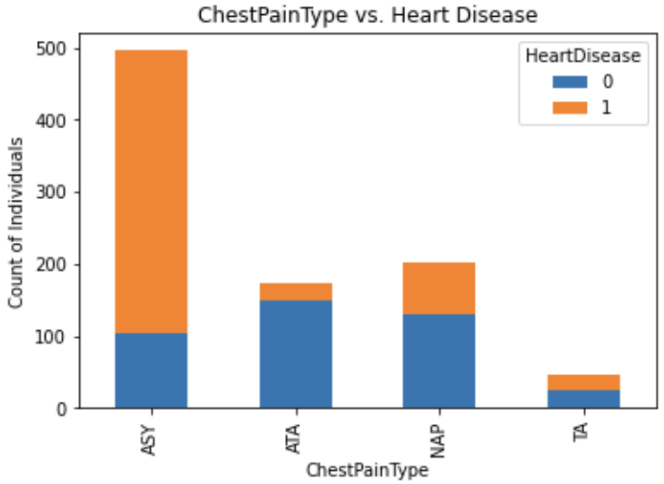
Plotting Sex vs. Heart Disease

This barplot depicts the relationship between the independent variable Sex and the dependent variable Heart Disease. According to the graph the Male population has a higher count of individuals having a heart disease.



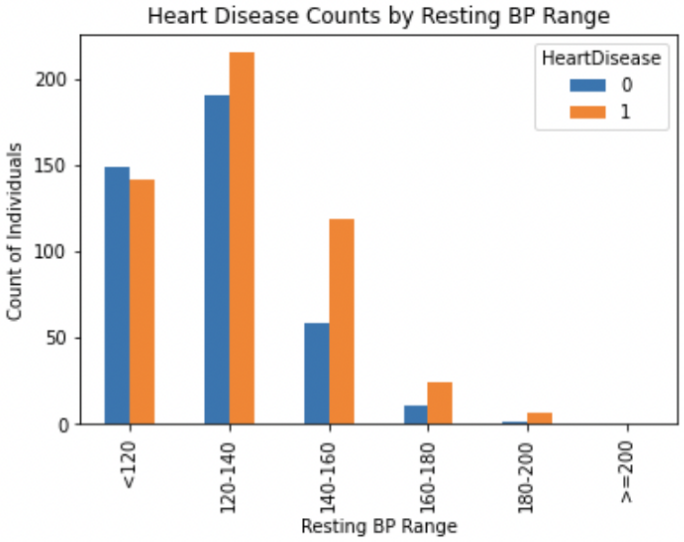
Plotting the ChestPainType vs. HeartDisease

This graph shows the four types of Chest pain type ASY, ATA, NAP,TA and the count of individuals having the heart disease with the particular type of pain. The graph shows that the maximum number of people having a heart disease had a ASY chest pain type.



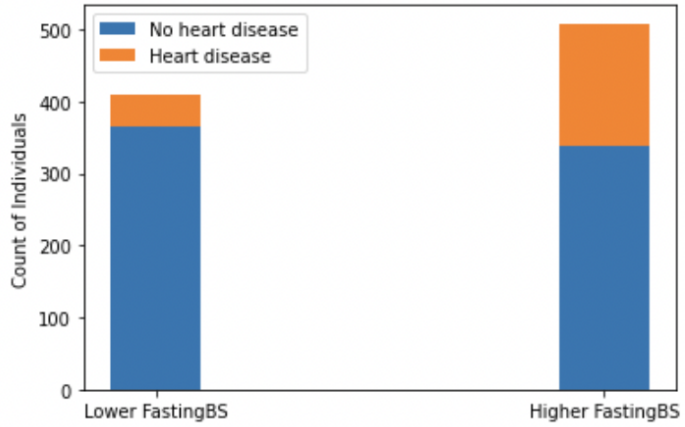
Plotting the RestingBP vs. Heart Disease

This is a grouped bar chart that depicts the RestingBP range and shows the count of individuals within the particular RestingBP range having a heart disease or not. The graph shows maximum people having the RestingBP in the in the range of 120-140 have individuals with higher rate of heart disease.



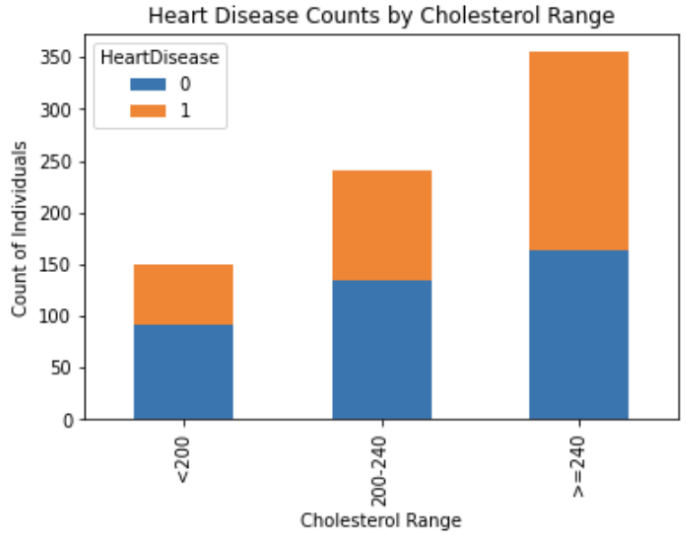
Plotting the FastingBS vs. Heart Disease

The graph depicts the Fasting blood sugar in two types higher or lower and according to visualisation it is clear that people having Higher fasting blood sugar have higher chances of Heart disease.



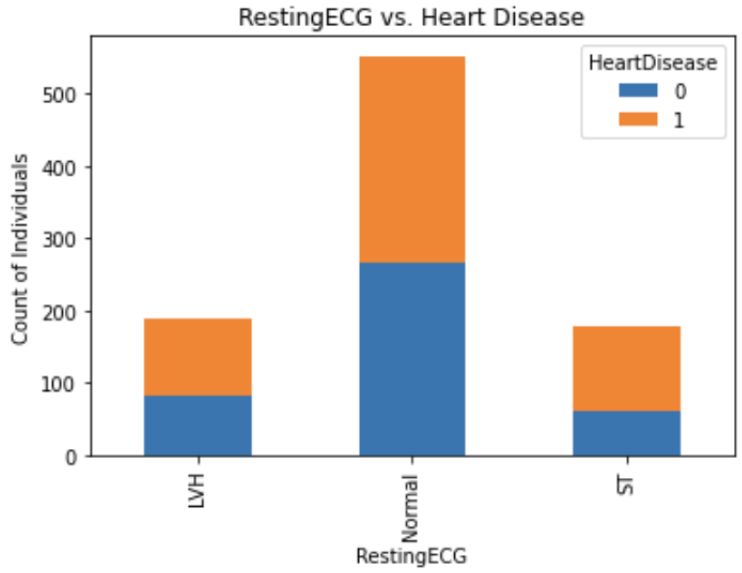
Plotting the Cholesterol and the Heart Disease

The graph shows the range of cholesterol levels and the count of people in the particular range having a heart disease or not. The graph clearly shows that people having a cholesterol level greater than or equal to 240 have higher chances of heart disease.



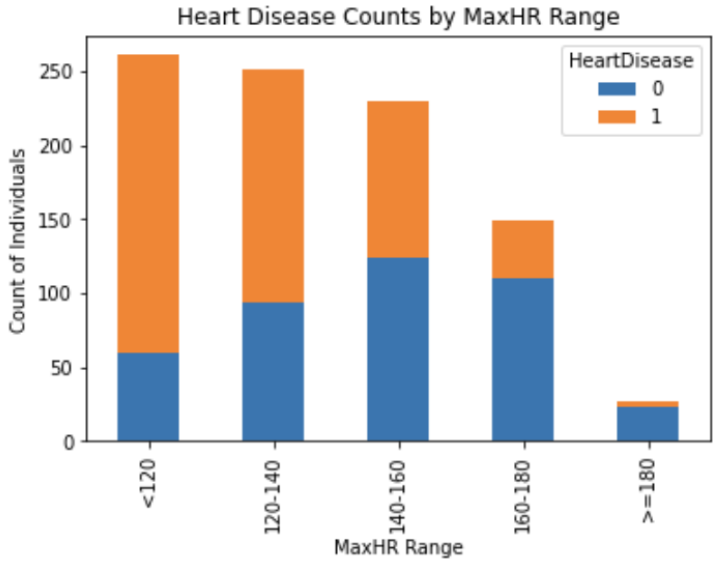
Plotting Resting ECG vs. Heart Disease

The stacked bar plot below shows the three types of ECG and the count of people having the particular RestingECG are prone to heart disease or not. The graph below depicts the people having ST as the Resting ECG have higher number of people having the heart disease.



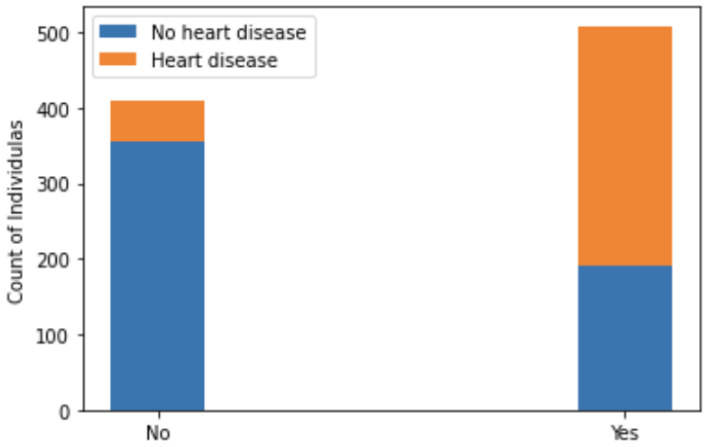
Plotting the Maximum Heart Range and the Heart Disease

The graph below depicts the MaxHR range and the count of individuals in that particular range having a heart disease or not. The graph shows that the people having MaxHR range less than 120 have higher chances of having a heart disease.



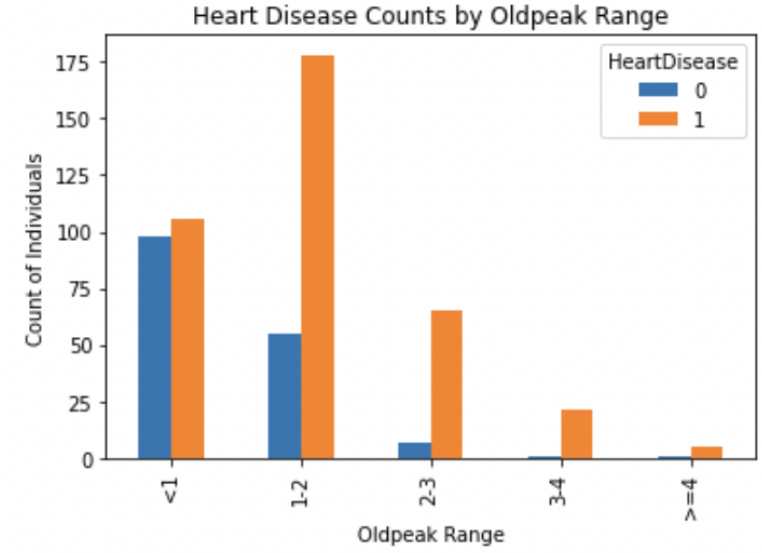
Plotting Exercise Angina and Heart Disease

The bar plot below shows the number of individuals having Excerise Angina(Yes) and those not having Exercise Angina (No) and shows which of them have a heart disease. The bar plot clearly depicts that the people having Exercise Angina have higher risk of having a heart disease.



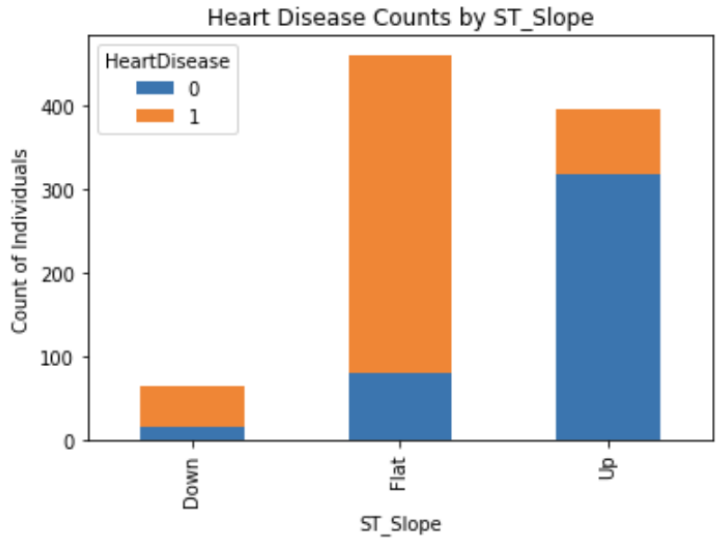
Plotting Oldpeak and Heart Disease

The grouped bar chart below shows the oldpeak range and the number of individuals in that range having chances of a heart disease or not. It depicts that the people in the range of oldpeak 1-2 have higher chances of having a heart disease.



Plotting the St_Slope and the Heart Disease

The stacked bar plot below shows the three types of ST_Slope and the count of individuals in that range having a heart disease or not. The bar plot with the Down ST_Slope has higher chances of having a heart disease.



Linear Regression

Linear regression is applied with the dependent variable on the Y-axis and the independent variable on the X-axis. Linear regression is applied separately for each independent variable along with the dependent variable. The relationship between each independent variable and the dependent variable is as follows in regard to the slope and R coefficient :

The slope of a linear regression model represents the change in the dependent variable for one-unit change in the independent variable and the R-coefficient depicts how well the regression model fits the data and it ranges from 0 to 1.

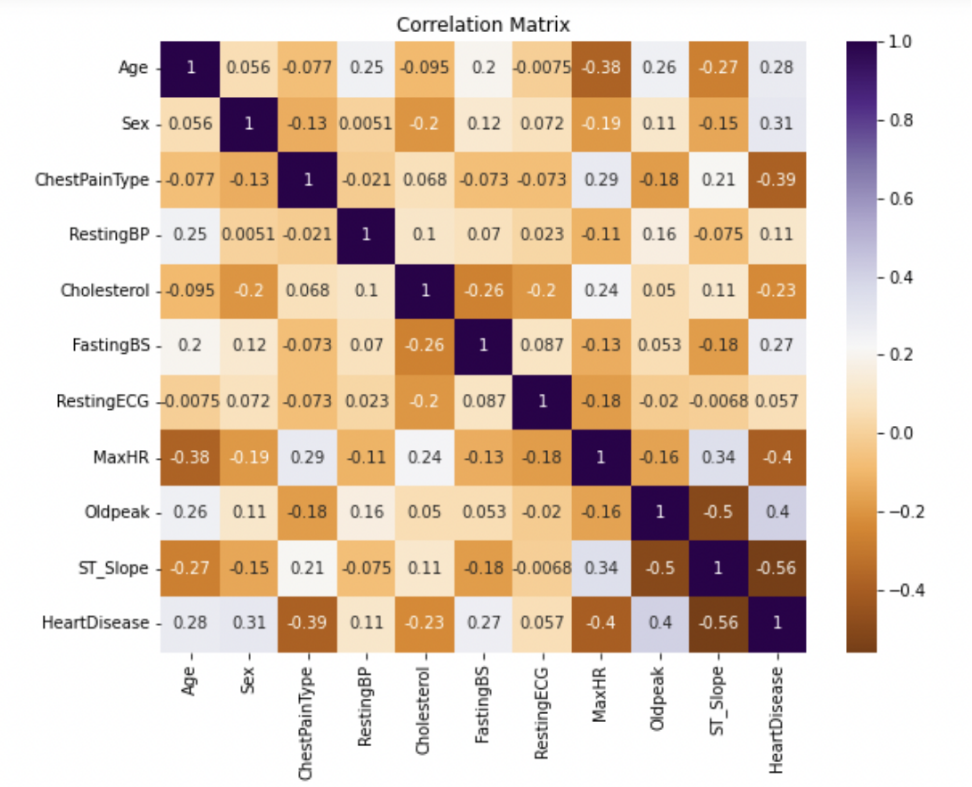
After applying Linear Regression on the HeartDisease dataset the results obtained are as follows :

Age - Intercept: -0.234, Coefficient: 0.015
Sex - Intercept: 0.238, Coefficient: 0.404
ChestPainType - Intercept: 0.708, Coefficient: -0.188
RestingBP - Intercept: 0.293, Coefficient: 0.002
Cholesterol - Intercept: 0.779, Coefficient: -0.001
FastingBS - Intercept: 0.484, Coefficient: 0.316
RestingECG - Intercept: 0.509, Coefficient: 0.053
MaxHR - Intercept: 1.592, Coefficient: -0.008
ExerciseAngina - Intercept: 0.364, Coefficient: 0.491
Oldpeak - Intercept: 0.401, Coefficient: 0.181
ST_Slope - Intercept: 1.187, Coefficient: -0.464

After carefully analyzing the above coefficient values, there seems to be a strong positive relationship between Sex and Heart Disease. There is a strong positive relation between the FastingBS and HeartDisease. Also the relationship between ExerciseAngina and Heart disease is strong and positive and that between the ST_Slope and Heart Disease is strong and negative compared to all other values.

Correlation Matrix

A correlation matrix shows the correlation coefficients between all the pairs of variables in the dataset. Correlation coefficients measure the strength and direction of the linear relationship between two variables. A positive correlation coefficient indicates a positive linear relationship whereas a negative correlation coefficient indicates a negative linear relationship. A correlation coefficient of 0 indicates no linear relationship between two variables. The heatmap below shows the correlation coefficient between two variables.



The heatmap here shows the the correlation strength from Purple to Orange with purple being the strongest. All the diagonals are purple as it has a correlation of 1 as the the relationship of a variable with itself will be the strongest. We are here looking at the correlation matrix of all independent variables against the one dependent variable that is the HeartDisease.

- The strongest positive relation here is between the ST_Slope and HeartDisease with a negative correlation of -0.56.

- The relationship between MaxHR and HeartDisease is a negative correlation of -0.4.
- The relationship between Oldpeak and HeartDisease has a positive correlation of 0.4
- The next strong relationship is that between ChestPainType and HeartDisease which is a negative correlation of -0.39
- There is a positive relationship between Sex and HeartDisease which is a positive 0.31.
- The relationship between FastingBS and HeartDisease is a positive correlation of 0.27
- Cholesterol and HeartDisease have a negative correlation of -0.23
- RestingBP and HeartDisease has a positive correlation of 0.11
- The weakest relationship of all is the relationship between the RestingECG and the HeartDisease.

Principal Component Analysis

PCA is used to convert a large set of variables into a smaller set of principal components that capture as much of the variation in the original data as possible. Proportion of variance which is also known as explained variance, is a measure that indicates how much of the total variance in a dataset is explained by a particular principal component in Principal Component Analysis (PCA). Proportion of variance is a useful metric for understanding the contribution of each principal component to the overall variation in the dataset. Loadings are the coefficients that represent the correlation between each variable and each principal component. The magnitude of the loading represents the strength of the correlation, and the sign represents the direction of the relationship. After implementing PCA while keeping the number of principal components as 2 and interpreting loadings in conjunction with proportion of variance gave the below results.

```
Explained variance ratio: [0.25139665 0.1330889 ]
                     PC1       PC2
Age             0.550796 -0.097884
Sex             0.357974  0.324108
ChestPainType  -0.482329 -0.022524
RestingBP       0.276850 -0.327280
Cholesterol    -0.272560 -0.724928
FastingBS       0.328908  0.408744
RestingECG      0.177159  0.485206
MaxHR          -0.673393 -0.207029
ExerciseAngina  0.704270 -0.207496
Oldpeak         0.602190 -0.444820
ST_Slope       -0.708256  0.177564
```

The snapshot above shows the proportion of variance/ explained variance for both the principal components. It also shows the contribution of each independent variable in the principal components which helps determine the strength of the correlation. Here the maximum contributions to PC1 are by the independent variables MaxHR,

ExerciseAngina, Oldpeak and ST_Slope. The maximum contributions to PC2 are by the independent variables Cholesterol, FastingBS, Oldpeak.

K-Means Clustering

To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs repetitive calculations to optimize the positions of the centroids It halts creating and optimizing clusters when these conditions are met :

- The centroids have stabilized — there is no change in their values because the clustering has been successful.
- The defined number of iterations has been achieved

The dataset used for the project depicts heart disease detection. The dataset has 12 features. The dataset is standardized to ensure that all variables are on the same scale and to improve the algorithm performance. Tried plotting the clusters in k-means clustering using scatter plot keeping the value of k=4 but the clusters were too compact. Checked the silhouette score to see the highest value of silhouette score and the corresponding clusters which was 2. Then implemented k means clustering keeping the value of k = 2. As the dataset contains 12 features it is difficult to plot it in a two-dimensional array. So PCA is used as it identifies the underlying structure in the data and reduces the number of features to a smaller number of components that capture most of the information in the original data. The results are then plotted on the graph and it is noticed that it has natural clusters. After visualizing it shows that there are natural clusters in the dataset.

Questions

1. What variables do you plan to use as the input features?

   After visualizing the results of linear regression, correlation matrix and PCA the crucial variables for the input features should be ST_Slope, MaxHR, Oldpeak, ChestPainType as these are the ones having the highest values of correlation. Also the features Sex, FastingBS, Cholesterol, RestingBP, Resting ECG can be considered in the given priority order.

2. What pre-processing (if any) did you execute on the variables?
   The first step towards data pre processing was converting the categorical values to numerical values. The next step was to remove outliers but that did not give accurate results as the dataset was very small hence that part was later decided to be removed. The null values and the duplicates were checked for and removed accordingly.

3. Which independent variables are strongly correlated (positively or negatively)?
   The variables that are strongly correlated according to the correlation matrix are as follows :
   1) Age and MaxHR : - 0.38 (negative correlation)
   2) Sex and HeartDisease : 0.31 (positive correlation)
   3) ChestPainType and HeartDisease : -0.39 (negative correlation)
   4) RestingBP and Age : 0.25 (positive correlation)
   5) Cholesterol and FastingBS : -0.26 (negative correlation)
   6) FastingBS and HeartDisease : 0.27 (positive correlation)
   7) RestingECG and Cholesterol : -0.2 (negative correlation)
   8) MaxHR and Age : -0.38 (negative correlation)
   9) Oldpeak and ST_Slope : -0.5 (negative correlation)
   10) ST_Slope and HeartDisease : -0.56 (negative correlation)

4. How many significant signals exist in the independent variables?
   Lasso Regression was performed to determine the significant signals in the independent variables. In Lasso Regression a penalty term L1 is added to the model which shrinks the coefficients of less important independent variables to zero and the number of significant signals is the number of independent variables with non-zero coefficients .There are 7 significant variables that are Age, ChestPainType, RestingBP, Cholesterol, MaxHR, Oldpeak, ST_Slope.

5. What derived or alternative features might be useful for analysis (e.g. polynomial features)?
   There are numerous alternative features that can be useful for analyzing. One of them is polynomial features. We can create new features by raising an existing feature to a power, such as a square or a cubic value and this will help us get the non-linear relationships between the features and the target variable. We can also standardize the features to have zero mean and unit variance which will eventually help us to reduce the impact of outliers and make the model more robust. One more approach could be to add more features to make our results more good and accurate . Features like stress level, family history of heart diseases, blood pressure, Body Mass Index and Exercise habits can be helpful towards analysis.

Task 2, 3

Both results for Part 2 and 3 are combined and compiled model-wise for better coherence in reading.

The ML methods we used for part 2 are:

- Logistic Regression
- Support Vector Machines
- Decision Trees

Extensions:
- Random Forest
- Naive Bayes

Since the dataset is a binary classification, we wanted to analyse the performance of Logistic Regression vs other models. As extensions, we used Naive Bayes and Random Forest Classifier as well.

Pre-processing:

For the training and test dataset used in the models, we used LabelEncoder from sklearn library to change the categorical variables to integers.

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope
0	66	0	2	146	278	0	0	152	0	0.0	1
1	65	1	0	150	235	0	1	120	1	1.5	1
2	63	1	0	150	223	0	1	115	0	0.0	1
3	58	1	1	136	164	0	2	99	1	2.0	1
4	54	1	1	192	283	0	0	195	0	0.0	2

Shown here are all the independent variables and the respective numerical values which were converted using LabelEncoder for the training dataset. This was done for both test and training sets.

Logistic Regression

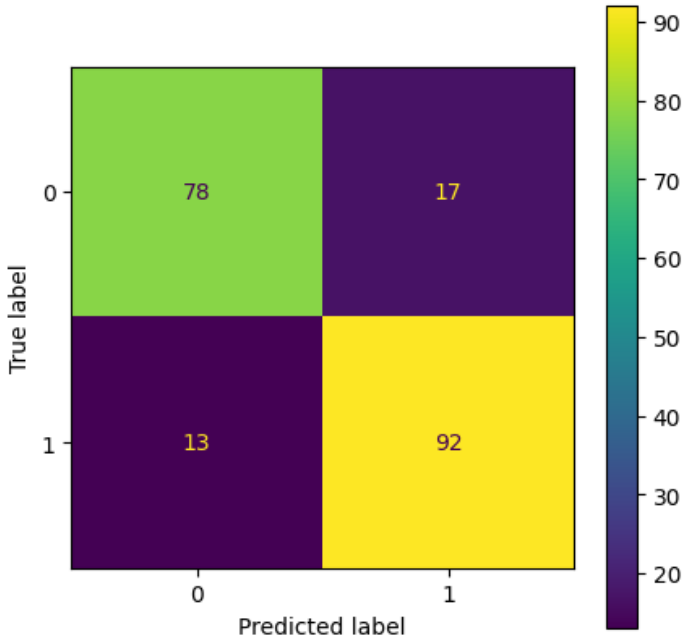
Logistic Regression is used to predict the outcome of an event happening (Heart Disease or not Heart Disease) based on the prior observations of a dataset.

The Logistic Regression model in our analysis did not converge for the default number of iterations and gave us a convergence error. On further research about this error, we learned that sklearn's Logistic Regression error "ConvergenceWarning: lbfgs failed to converge (status=1): STOP: TOTAL NO. of ITERATIONS REACHED LIMIT" happens when the error varies noticeably for multiple iterations above a certain threshold. Stack overflow posts suggested that increasing the 'max-iter' argument can give better results and hence, on setting max_iter = 1000, we were able to get an accuracy of 85% on the test set.

PART 3 Results - Logistic Regression

Bias, Variance, Mean Squared Error - Logistic Regression

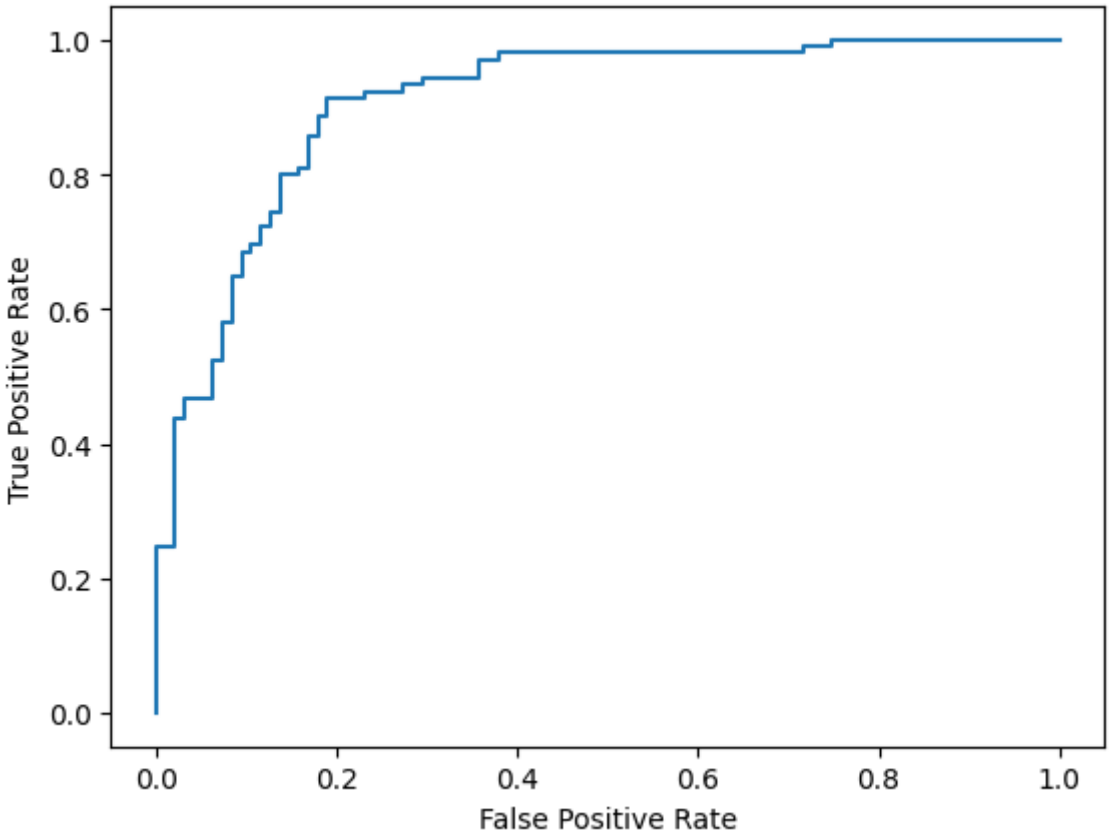
Mean Squared Error: 0.024
Bias: 0.007
Variance: 0.018



F1 score, Classification Report - Logistic Regression

	precision	recall	f1-score	support
0	0.86	0.82	0.84	95
1	0.84	0.88	0.86	105
accuracy			0.85	200
macro avg	0.85	0.85	0.85	200
weighted avg	0.85	0.85	0.85	200

Receiver-Operator Curve - Logistic Regression



Support Vector Machine

Support Vector Machine is a supervised Machine Learning algorithm which can be used for both classification and regression problems. In SVM, each data is a point in an n-dimensional space and the aim is to find a hyperplane that divides the classes involved in an optimal way. An increased margin along the hyperplane helps with better classification.

The parameters considered for SVM are: kernel, gamma and C

Gamma: Higher values of gamma can result in the model overfitting to the training dataset and reduced generalisation on the actual/test data.

C: This is the regularisation parameter.

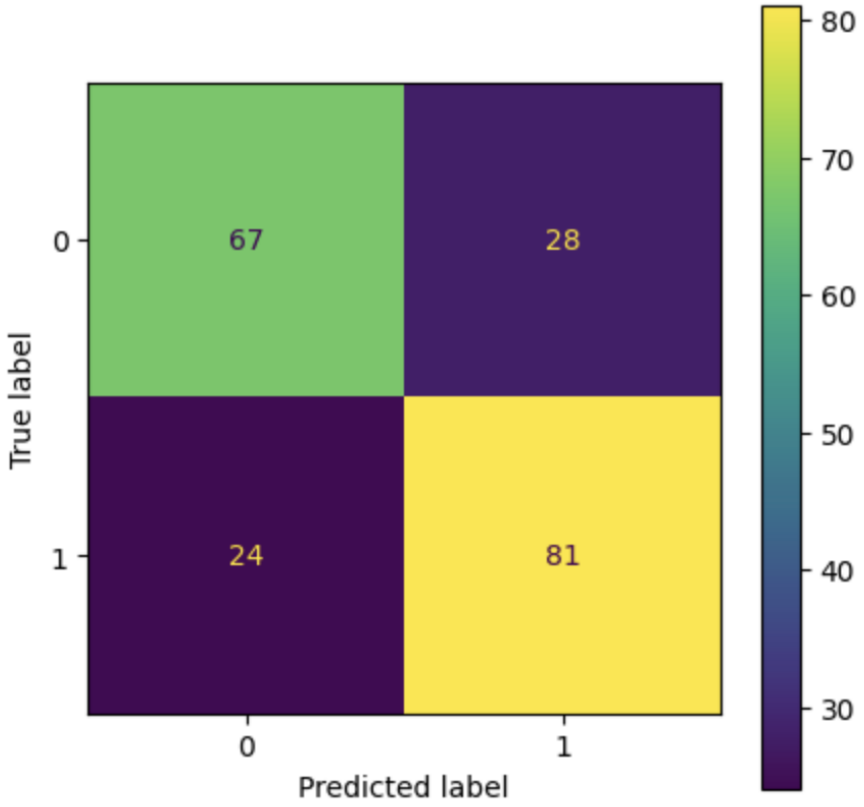
Kernel: The function which is used to separate the data between the classes in a high-dimensional input space.

Part 3 results:

With the default parameters of sklearn svm model, the results obtained are as below:

Model Accuracy Score: 74%

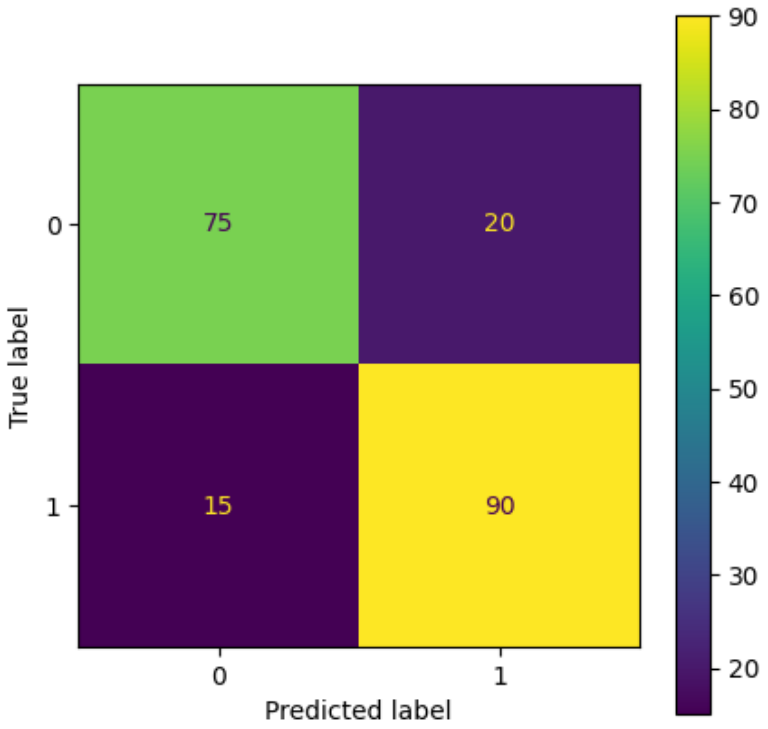
Mean Squared Error: 0.054
Bias: 0.015
Variance: 0.039



	precision	recall	f1-score	support
0	0.74	0.71	0.72	95
1	0.74	0.77	0.76	105
accuracy			0.74	200
macro avg	0.74	0.74	0.74	200
weighted avg	0.74	0.74	0.74	200

It can be observed that the model accuracy is low, and also, the False Negative score in the confusion matrix is quite high, which can lead to wrongly identifying a patient with heart disease as healthy. On increasing the regularisation parameter C, the results obtained are as follows:

Accuracy Score: 82.5%
Mean Squared Error: 0.055
Bias: 0.017
Variance: 0.038



	precision	recall	f1-score	support
0	0.83	0.79	0.81	95
1	0.82	0.86	0.84	105
accuracy			0.82	200
macro avg	0.83	0.82	0.82	200
weighted avg	0.83	0.82	0.82	200

By increasing C to 100, the model accuracy improved and also the number of False Negatives reduced by 37.5%. On increasing the value of gamma to 10 along with the regularisation, the accuracy dropped to 50% and hence further analysis was not done.

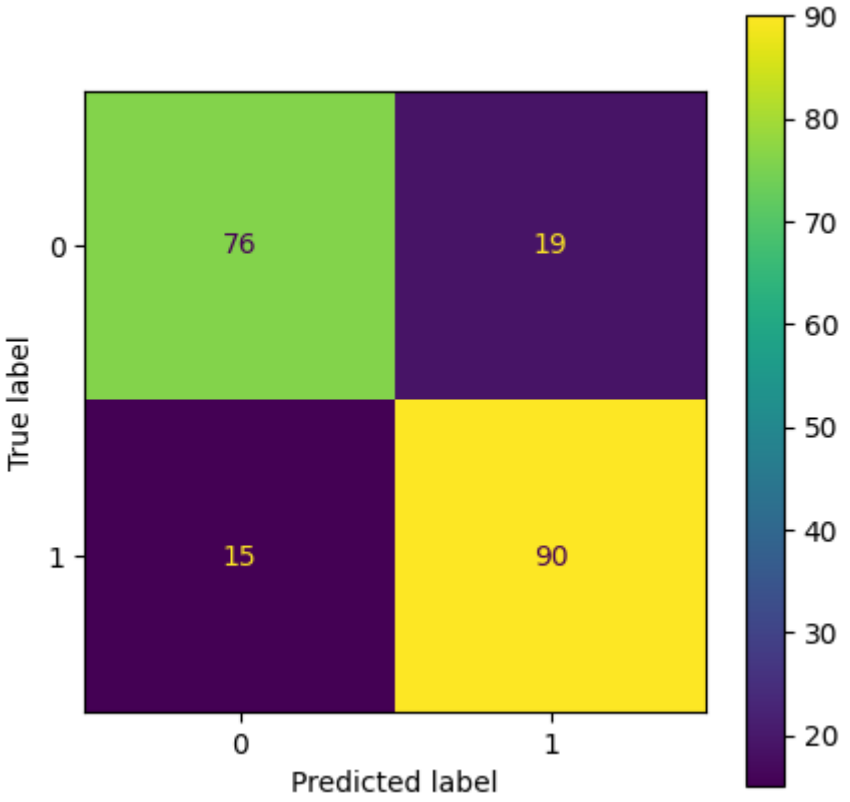
The bias and variance however, do not show much change.

Decision Trees

A decision tree is a supervised learning algorithm which is used for both classification and regression. WIth a hierarchical and tree structure, at each node, the best attribute to consider to make a decision is based on : Entropy and Information Gain.

Accuracy: 83.5%

Mean Squared Error: 0.187
Bias: 0.082
Variance: 0.105



	precision	recall	f1-score	support
0	0.84	0.80	0.82	95
1	0.83	0.86	0.84	105
accuracy			0.83	200
macro avg	0.83	0.83	0.83	200
weighted avg	0.83	0.83	0.83	200

It can be observed that with default parameters, Decision tree gives better accuracy and f1-score.

1. Which classifier did the best?

   As per our analysis, Decision Tree performed the best.

2. What statistic are you using to decide which classifier is the best performer?

Though Logistic Regression has high accuracy and low bias, variance compared to other models, the number of iterations required for the algorithm to converge was very high (= 1000) w.r.t default iterations of 100. This indicates that the model was overfit for the training data and justifies the lower bias. Hence, it was ruled out from consideration. Between SVM and Decision Tree, the higher accuracy was achieved by Decision Tree for default parameters. Hence, based on this performance statistic, and the fact that decision tree is a slightly more complex model which can help in identifying non-linear relationships between the data better, we decided to optimise it further for Part 4.

3. What does the bias and variance indicate as to what the best next steps to take would be to improve performance?

For the model that we selected, we observed higher variance and lower bias, indicating that the model is not overfitting and is generalising well. Hence, Decision Tree model was selected to observe if accuracy can be improved further by hyperparameter tuning.

4. What would be a good operating point for the classifier for which you generated the ROC curve?

Setting the threshold between 0.3 - 0.35 would be a good operating point. This would give True Positives rate closer to 1 and also a slightly higher false Positive rate as well. But from our data/problem since the risk involved for a False Positive is lower and since it is better to have a higher True Positive rate, we chose a lower threshold.

Task 4

Iterations

After carefully analyzing all the statistics in the classification report the **Decision Tree Classifier** was found to be the best performing classifier and the iterations and modifications were performed on the same to improve it's performance. I performed four iterations focusing on improving the F1 score of the decision tree classifier.
Hyperparameters of the Decison Tree classifier are :

1. Maximum depth: This hyperparameter specifies the maximum depth of the decision tree.
2. Minimum samples split: This hyperparameter specifies the minimum number of samples required to split a node.
3. Minimum samples leaf: This hyperparameter specifies the minimum number of samples required to be at a leaf node.
4. Maximum features: This hyperparameter specifies the maximum number of features to consider when splitting a node.
5. Split criterion: This hyperparameter specifies the criterion used to evaluate the quality of a split. The two common split criteria are Gini impurity and Entropy.

Four iterations were performed manually on the decision tree classifier and later GridSearchCV was used to get more accuracy for the classifier model.

Iteration 1

In the first iteration the value of maximum depth was considered to be 5 and the criterion entropy and the sample split was 10. This was done manually using trial and error methods and checking the changes in the F1 score of the model. The first iteration increased the model accuracy from 81.50% to 82.50% and the F1 scores also increased from the original Decision tree model.

```
Accuracy: 82.50%
Classification report:
               precision    recall  f1-score   support

           0       0.79      0.85      0.82        95
           1       0.86      0.80      0.83       105

    accuracy                           0.82       200
   macro avg       0.83      0.83      0.82       200
weighted avg       0.83      0.82      0.83       200

Confusion matrix:
 [[81 14]
 [21 84]]
```

Iteration 2

In the second iteration the maximum depth was taken as 7 and the criterion as gini and the minimum sample split was considered to be 5. This iteration gave good results and the model accuracy increased to 84.00% and the F1 scores also increased compared to the original results. The confusion matrix showed good results too as it had higher number of True Positives and True Negatives and a lower number of False Positives and False Negatives.

```
Accuracy: 84.00%
Classification report:
              precision    recall  f1-score   support

           0       0.84      0.82      0.83        95
           1       0.84      0.86      0.85       105

    accuracy                           0.84       200
   macro avg       0.84      0.84      0.84       200
weighted avg       0.84      0.84      0.84       200

Confusion matrix:
 [[78 17]
 [15 90]]
```

Iteration 3

In the third iteration the value of maximum depth was taken as 5 and the entropy criterion was considered for this iteration keeping the minimum sample split as 10. This iteration did not lead to increase in the model's accuracy or the F1 score compared to the original model. The accuracy went down from 81.50% to 80.00 % and also the F1 scores decreased.

```
Accuracy: 80.50%
Classification report:
              precision    recall  f1-score   support

           0       0.77      0.83      0.80        95
           1       0.84      0.78      0.81       105

    accuracy                           0.81       200
   macro avg       0.81      0.81      0.80       200
weighted avg       0.81      0.81      0.81       200

Confusion matrix:
 [[79 16]
 [23 82]]
```

Iteration 4

In the fourth iteration only two hyperparameters were changed the maximum depth was taken as 10 with the gini entropy. The model's accuracy increased a bit compared to the original model but not as good as the second iteration. The accuracy increased from 81.50% to 80%.

```
Accuracy: 82.00%
Classification report:
              precision    recall  f1-score   support

           0       0.84      0.77      0.80        95
           1       0.81      0.87      0.83       105

    accuracy                           0.82       200
   macro avg       0.82      0.82      0.82       200
weighted avg       0.82      0.82      0.82       200

Confusion matrix:
 [[73 22]
 [14 91]]
```

After performing the four iterations the best results were obtained from the second iteration where the value of the maximum depth was taken as 7 and the criterion as gini and the minimum sample split was considered to be 5. It increased the overall model accuracy to 84%

Grid Search CV

GridSearchCV is a hyperparameter tuning technique used in machine learning to find the optimal set of hyperparameters for a given model.It works by taking a set of hyperparameters, usually defined as a dictionary, and creating all possible combinations of hyperparameters from that set. Then the model is then trained and evaluated for each combination of hyperparameters and the best combination is selected. The hyperparameters considered for the dictionary were maximum depth, minimum sample split and minimum leaf nodes. After performing GridSearchCV the results obtained for these hyperparameters were 5, 2, 1 respectively. The model accuracy increased to 85% and the F1 scores as well as the confusion matrix gave very good results.

```
Best hyperparameters: {'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 2}
Classification report:
              precision    recall  f1-score   support

           0       0.84      0.85      0.84        95
           1       0.86      0.85      0.86       105

    accuracy                           0.85       200
   macro avg       0.85      0.85      0.85       200
weighted avg       0.85      0.85      0.85       200

Confusion matrix:
 [[81 14]
 [16 89]]
```

Extensions

Random Forest Classifier

Random Forest Classifier is a type of ensemble learning algorithm used for classification tasks in machine learning. It is a variant of decision tree algorithms and is based on the idea of combining multiple decision trees that improves the accuracy of the model. Performing Random Forest Classifier gave good results in the original model without any iterations.

```
Accuracy: 85.00%
Classification report:
              precision    recall  f1-score   support

           0       0.85      0.83      0.84        95
           1       0.85      0.87      0.86       105

    accuracy                           0.85       200
   macro avg       0.85      0.85      0.85       200
weighted avg       0.85      0.85      0.85       200
```

Naive Bayes Classification

Naive Bayes Classification is a probabilistic algorithm used for classification tasks in machine learning. The algorithm works by calculating the probability of each class given a set of features and then choosing the class with the highest probability as the predicted class. It uses Bayes theorem to calculate the probability of each class given the features:
P(class|features) = P(features|class) * P(class) / P(features)
The results obtained for Naive Bayes were slightly better than Random Forest. The overall model accuracy was 85.50%.

```
Accuracy: 85.50%
Classification report:
              precision    recall  f1-score   support

           0       0.84      0.86      0.85        95
           1       0.87      0.85      0.86       105

    accuracy                           0.85       200
   macro avg       0.85      0.86      0.85       200
weighted avg       0.86      0.85      0.86       200
```

References

1. https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset
2. https://www.kaggle.com/datasets/redwankarimsony/heart-disease-data
3. http://www.sthda.com/english/wiki/correlation-matrix-a-quick-start-guide-to-analyze-format-and-visualize-a-correlation-matrix-using-r-software
4. https://www.w3schools.com/python/python_ml_decision_tree.asp
5. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
6. https://www.ibm.com/topics/decision-trees#:~:text=A%20decision%20tree%20is%20a,internal%20nodes%20and%20leaf%20nodes
7. https://stackoverflow.com/questions/62658215/convergencewarning-lbfgs-failed-to-converge-status-1-stop-total-no-of-iter
8. https://medium.com/analytics-vidhya/calculation-of-bias-variance-in-python-8f96463c8942