PROJECT: Supervised Modelling For Heart Disease Prediction

Name: Debi Majumdar

JUNE 2022

SUBMITTED TO: IEMA RESEARCH & DEVELOPMENT PVT LTD

ABSTRACT

As of 2018, 30.3 million U.S. adults were diagnosed with heart disease. Every year, about 647,000 Americans die from heart disease, making it the leading cause of death in the United States.Heart disease causes 1 out of every 4 deaths. Cardiovascular diseases (CVDs) are the leading cause of death globally.(Figure 1) An estimated 17.9 million people died from CVDs in 2019, representing 32% of all global deaths. Of these deaths, 85% were due to heart attack and stroke. With the rate of heart disease having an exponential growth every year it is becoming more and more important for us to identify who can have a heart disease early on so it can be treated in time. In this project the aim is to predict heart disease based on a dataset taken from kaggle. I have used machine learning algorithms in order to predict whether or not a person is suffering from heart disease.

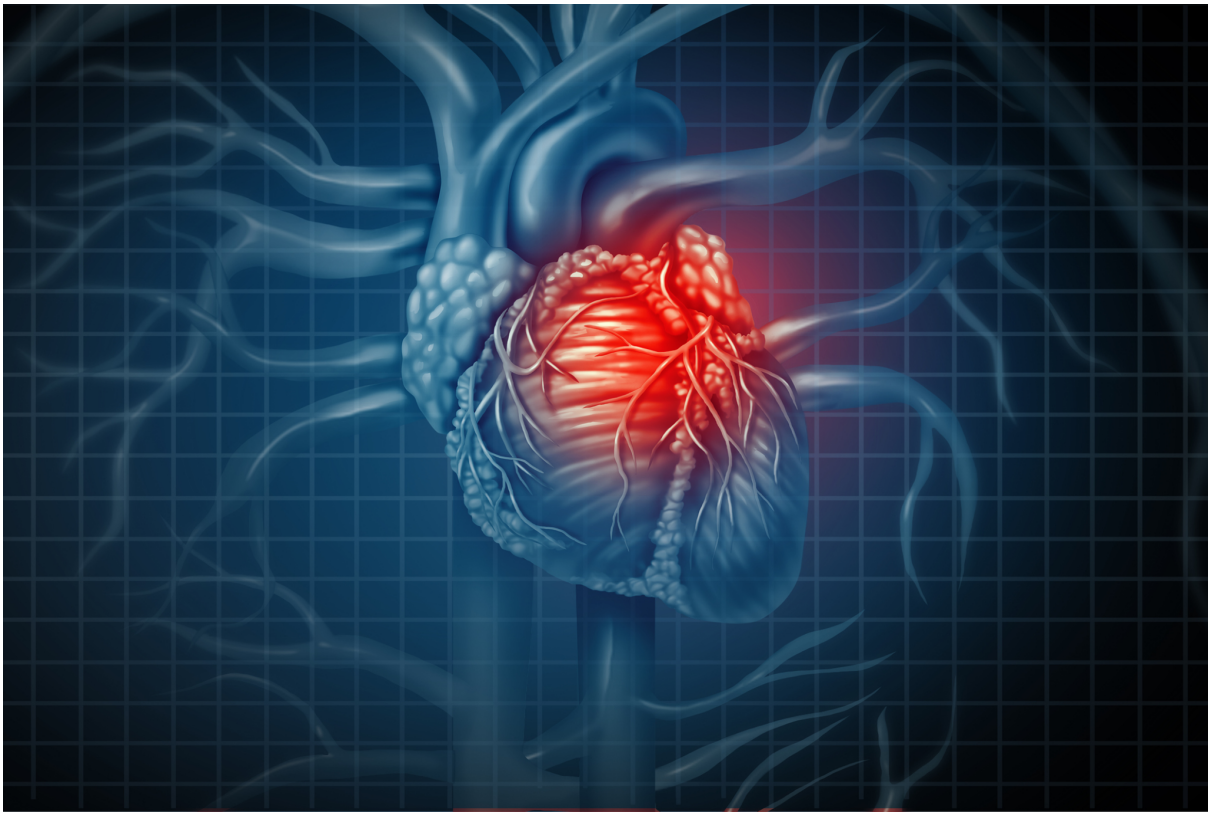

Figure 1: Heart Disease overview

TABLE OF CONTENTS

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make predictions with minimal human intervention. Machine learning Incorporates various classifiers of Supervised, Unsupervised and Ensemble Learning which are used to predict and Find the Accuracy of the given dataset. These are some of the commonly used supervised machine learning algorithms. :-

A. K neighbors classifier: A non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.
B. Decision tree classifier: They have the capability of capturing descriptive decision making knowledge from the supplied data
C. Random tree classifier: is a supervised machine-learning classifier based on constructing a multitude of decision trees, choosing random subsets of variables for each tree, and using the most frequent tree output as the overall classification.
D. Logistic regression: It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable.
E. Support Vector Machine(SVM): The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

I have been using three supervised machine learning algorithms to predict whether a person has heart disease or not. The algorithms I have used are Logistic Regression, Random forest and Decision tree classifier.

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. Machine learning is becoming more and more important now because it gives enterprises a view of trends in customer behaviour and business operational patterns, as well as supports the development of new products. Many of today's leading companies, such as Facebook, Google and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies. Classical machine learning is often categorised by how an algorithm learns to become more accurate in its predictions. There are four basic approaches:supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning (Figure 2). The type of algorithm data scientists choose to use depends on what type of data they want to predict.

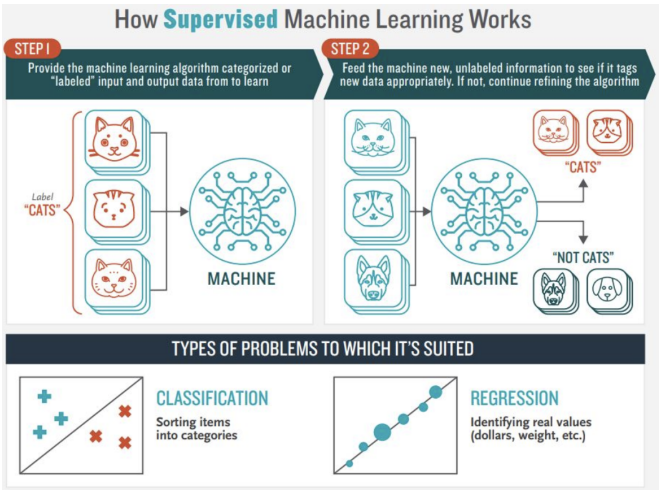

Figure 2: Supervised learning overview

Supervised learning is the type of machine learning in which machines are trained using well "labelled" training data, and on the basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to **find a mapping function to map the input variable(x) with the output variable(y)**.

These are some of the commonly used supervised machine learning algorithms. :-

A. K neighbors classifier- A non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.
B. Decision tree classifier- They have the capability of capturing descriptive decision making knowledge from the supplied data
C. Random tree classifier- is a supervised machine-learning classifier based on constructing a multitude of decision trees, choosing random subsets of variables for each tree, and using the most frequent tree output as the overall classification.
D. Logistic regression: It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable.
E. Support Vector Machine(SVM): The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

DATASET OVERVIEW

The dataset worked upon is taken from Kaggle 'Heart Disease Cleveland UCI'
There are 13 attributes and 303 rows. (Figure 3&4)

1. age: age in years
2. sex: sex (1 = male; 0 = female)
3. cp: chest pain type
   -- Value 0: typical angina
   -- Value 1: atypical angina
   -- Value 2: non-anginal pain
   -- Value 3: asymptomatic
4. trestbps: resting blood pressure (in mm Hg on admission to the hospital)
5. chol: serum cholesterol in mg/dl
6. fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
7. restecg: resting electrocardiographic results
   -- Value 0: normal
   -- Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
   -- Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
8. thalach: maximum heart rate achieved
9. exang: exercise induced angina (1 = yes; 0 = no)
10. oldpeak = ST depression induced by exercise relative to rest
11. slope: the slope of the peak exercise ST segment
    -- Value 0: upsloping
    -- Value 1: flat
    -- Value 2: downsloping
12. ca: number of major vessels (0-3) colored by fluoroscopy
13. thal: 0 = normal; 1 = fixed defect; 2 = reversible defect
    and the label
14. condition: 0 = no disease, 1 = disease

```
# print first 5 rows of the dataset
heart_data.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Figure 3: Dataset description of first 5 rows

```
# print last 5 rows of the dataset
heart_data.tail()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

Figure 4: Dataset description of last 5 rows

PRE-QUISTIC LIBRARY:

The following libraries (Figure 5) have been implemented:

1. A library called NumPy, or Numerical Python,provides multidimensional array objects and routines for processing them. With NumPy, arrays can be mathematically and logically manipulated.

2. Pandas is an open-source Python package that is primarily used for data science, data analysis, and machine learning tasks. In addition to supporting multidimensional arrays, it is built on top of Numpy.

3. Matplotlib is a cross-platform, data visualisation and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

4. Seaborn is a Python data visualisation library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphs. Used for generating a heatmap.

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib import rcParams
from matplotlib.cm import rainbow
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

Figure 5: Pre-quistic libraries implementation

CORRELATION BETWEEN DATA:

The statistical relationship between two variables is referred to as their correlation. (Figure 6)It is plotted in the heatmap below by seaborn. (Figure 7)

```
#get correlations of each features in dataset
corrmat = heart_data.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(20,20))
#plot heat map
g=sns.heatmap(heart_data[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```

Figure 6: Correlation for dataset of Heart Disease prediction

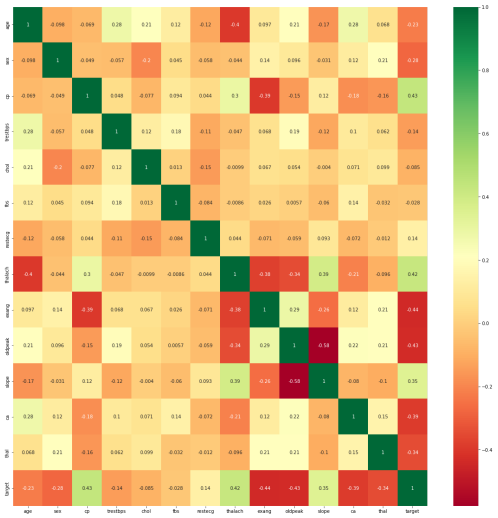

Figure 7: Heat Map plotting

MODEL TRAINING AND TESTING

I have used the train test split library in this project. Train test split is a model validation procedure that allows you to simulate how a model would perform on new/unseen data. Here is how the procedure works. (Figure 8)

1. First arranging the data into a format acceptable for train test split. In scikit-learn, this consists of separating the full dataset into Features and Target.
2. Splitting the dataset into two pieces: a training set and a testing set. This consists of random sampling without replacement about 80% of the rows and putting them into the training set and putting the remaining 20% to the test set. Note that the colours in “Features” and “Target” indicate where their data will go (“X_train”, “X_test”, “y_train”, “y_test”) for a particular train test split.
3. Training the model on the training set. This is “X_train” and “y_train” in the image.
4. Testing the model on the testing set (“X_test” and “y_test” in the image) and evaluating the performance.

```
from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)

print("Original X data is:    " , X.shape)
print("X_train is:            " , X_train.shape)
print("X_test is:             " , X_test.shape)
print("\n========================================\n")
print("The original Y data is:  ", Y.shape)
print("Y_train is:            " , Y_train.shape)
print("Y_test is:             " , Y_test.shape)
```

```
Original X data is:     (303, 13)
X_train is:             (242, 13)
X_test is:              (61, 13)

========================================

The original Y data is:   (303,)
Y_train is:             (242,)
Y_test is:              (61,)
```

Figure 8: Training and testing the Heart Disease Dataset

MODEL FITTING

MODEL1 : LOGISTIC REGRESSION

Logistic Regression is much similar to Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

Training the logistic regression model with the data used for training as shown before. (Figure 9)

```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()

# training the LogisticRegression model with Training data
model.fit(X_train, Y_train)

LogisticRegression()
```

Figure 9: Implementing Logistic Regression

MODEL2 : RANDOM FOREST

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. (Figure 10)

```
#apply RANDOM FOREST CLASSIFIER in heart dataset
from sklearn.ensemble import RandomForestClassifier
rforest = RandomForestClassifier()
rforest.fit(X_train_SC , Y_train)
y_pred3 = rforest.predict(X_test_SC)
```

Figure 10: Implementation of Random Forest Classifier

MODEL3 : DECISION TREE

The decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. (Figure 11)

```
#apply DECISSION TREE CLASSIFIER in heart dataset
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier()
dtree.fit(X_train_SC , Y_train)
y_pred2 = dtree.predict(X_test_SC)
```

Figure 11: Implementation of Decision Tree classifier

MODEL EVALUATION

A confusion matrix is a tabular summary of the number of correct and incorrect predictions made by a classifier. It can be used to evaluate the performance of a classification model through the calculation of performance metrics like accuracy, precision, recall, and F1-score.

The Confusion Matrix created has four different quadrants:

a. True Negative (Top-Left Quadrant)
b. False Negative (Top-Right Quadrant)
c. False Positive (Bottom-Left Quadrant)
d. True Positive (Bottom-Right Quadrant

Logistic Regression:

```
# Constructing the confusion matrix.
from sklearn.metrics import confusion_matrix

lr_cn_mat = confusion_matrix(y_pred,Y_test)
print("Confusion matrix is :\n",lr_cn_mat)
```

```
Confusion matrix is :
 [[23  6]
 [ 5 27]]
```

```
import matplotlib.pyplot as plt
import numpy
from sklearn import metrics



confusion_matrix = metrics.confusion_matrix(Y_test, y_pred)

cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix, display_labels = [False, True])

cm_display.plot()
plt.show()
```

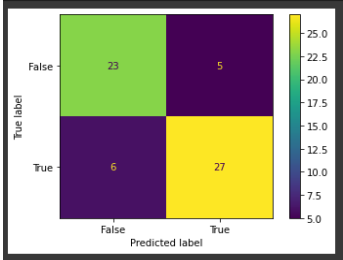

Figure 12: Confusion Matrix for Logistic Regression

Random Forest:

```
import matplotlib.pyplot as plt
import numpy
from sklearn import metrics
confusion_matrix = metrics.confusion_matrix(Y_test, y_pred3)
cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix, display_labels = [False, True])
cm_display.plot()
plt.show()
```

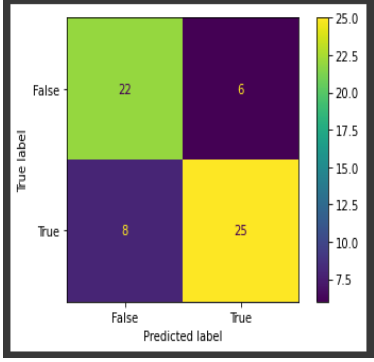

Figure 13: Confusion Matrix for Random Forest

Decision tree:

```
import matplotlib.pyplot as plt
import numpy
from sklearn import metrics
confusion_matrix = metrics.confusion_matrix(Y_test, y_pred2)
cm_display = metrics.ConfusionMatrixDisplay(confusion_matrix = confusion_matrix, display_labels = [False, True])
cm_display.plot()
plt.show()
```

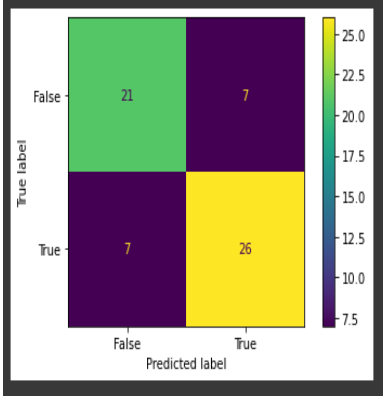

Figure 14: Confusion Matrix for Decision Tree

True means that the values were accurately predicted, False means that there was an error or wrong prediction. Now that we have made a Confusion Matrix, we can calculate different measures to quantify the quality of the model. First, let's look at Accuracy.

ACCURACY SCORE

The accuracy score method is used to calculate the accuracy of either the fraction or count of correct prediction in Python Scikit learn. Mathematically it represents the ratio of the sum of true positives and true negatives out of all the predictions.
The accuracy score for my program using logistic regression is 0.819672131147541% (Figure 15), using random forest is 0.8032786885245902% (Figure 17), and using decision tree is also 0.774918032786885% (Figure 19).

Logistic Regression:

```
from sklearn.metrics import accuracy_score

# accuracy on test data
test_data_accuracy = accuracy_score(X_pred, Y_test)

print("Accuracy score for Logistic Regression is :", accuracy_score(y_pred,Y_test))
```

```
Accuracy score for Logistic Regression is : 0.819672131147541
```

Figure 15: Accuracy Score for Logistic Regression

ROC curve, which stands for “receiver operating characteristic” curve. This is a plot that displays the sensitivity and specificity of a logistic regression model. (Figures 16,18&20)

```
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(Y_test, model.predict(X_test))
fpr, tpr, thresholds = roc_curve(Y_test, model.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('Log_ROC')
plt.show()
```

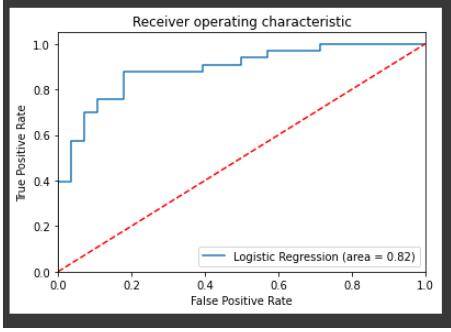

Figure 16: ROC Curve for Logistic Regression

Random Forest:

```
print("Accuracy score for Random Forest classifier is :\n", accuracy_score(y_pred3,Y_test))
```

```
Accuracy score for Random Forest classifier is :
 0.8032786885245902
```

Figure 17: Accuracy Score for Random Forest

ROC Curve:

```
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(Y_test, rforest.predict(X_test))
fpr, tpr, thresholds = roc_curve(Y_test, rforest.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='Random Forest (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('Log_ROC')
plt.show()
```

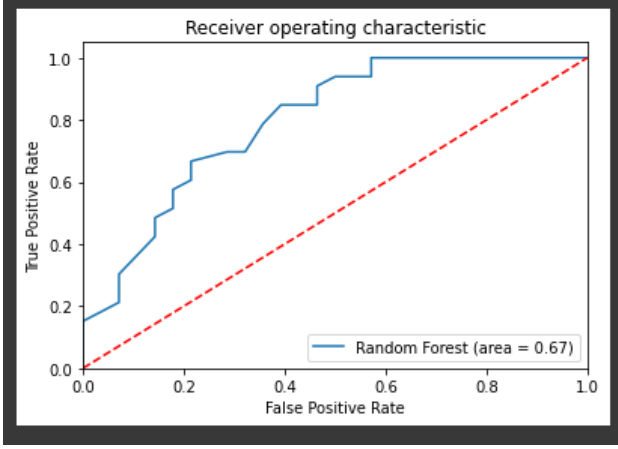

Figure 18: ROC Curve for Random Forest

Decision Tree:

```
from sklearn.metrics import accuracy_score
print("Accuracy score for Decision Tree is :\n", accuracy_score(y_pred2,Y_test))
```

```
Accuracy score for Decision Tree is :
 0.7704918032786885
```

Figure 19: Accuracy Score for Decision Tree

ROC Curve:

```
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(Y_test, dtree.predict(X_test))
fpr, tpr, thresholds = roc_curve(Y_test, dtree.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fpr, tpr, label='Decision Tree (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('Log_ROC')
plt.show()
```

Figure 20: ROC Curve for Decision Tree

DATA VALIDATION

Here I have used a numpy array which is a random data set imputed by the user, it is not related to the data set given before which was used for training and testing. By using reshape I have changed the shape of the array without affecting its data and implemented it in the prediction model.
The model predicts whether a person has a heart disease or not based on the data inputted right now.
Where 0= The Person does not have a Heart Disease
And, 1= The Person has a Heart Disease

```
def predict(ip_data):
  # change the input data to a numpy array
  input_data_as_numpy_array= np.asarray(ip_data)

  # reshape the numpy array as we are predicting for only on instance
  input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

  prediction = model.predict(input_data_reshaped)
  print("The predicted outcome is: ",prediction)
  if (prediction[0]== 0):
    print('Which means The Person does not have a Heart Disease')
  else:
    print('Which means The Person has Heart Disease')
  #print(prediction)
  return prediction

# driver code for prediction
input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)
op_data = predict(input_data)
```

```
The predicted outcome is:  [0]
Which means The Person does not have a Heart Disease
```

Figure 21: Data Validation for Heart Disease prediction

CLASSIFICATION REPORT

It is one of the performance evaluation metrics of a classification-based machine learning model. It displays your model's precision, recall, F1 score and support. It provides a better understanding of the overall performance of our trained model.

Metrics	Definition
Precision	Precision is defined as the ratio of true positives to the sum of true and false positives.
Recall	Recall is defined as the ratio of true positives and false negatives.
F1 Score	The F1 is the weighted harmonic mean of precision and recall. The closer the value of the F1 score is 1.0, the better the expected performance of the model is.
Support	Support is the number of actual occurrences of the class in the dataset. It doesn't vary between models, it just diagnoses the performance evaluation process.

Logistic regression:

```
from sklearn.metrics import classification_report
print(classification_report(Y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.79      0.82      0.81        28
           1       0.84      0.82      0.83        33

    accuracy                           0.82        61
   macro avg       0.82      0.82      0.82        61
weighted avg       0.82      0.82      0.82        61
```

Figure 22: Classification Report of Logistic Regression

Random Forest:

```
from sklearn.metrics import classification_report
print(classification_report(Y_test, y_pred3))
```

```
              precision    recall  f1-score   support

           0       0.73      0.79      0.76        28
           1       0.81      0.76      0.78        33

    accuracy                           0.77        61
   macro avg       0.77      0.77      0.77        61
weighted avg       0.77      0.77      0.77        61
```

Figure 23: Classification Report of Random Forest

Decision Tree:

```
from sklearn.metrics import classification_report
print(classification_report(Y_test, y_pred2))
```

```
              precision    recall  f1-score   support

           0       0.75      0.75      0.75        28
           1       0.79      0.79      0.79        33

    accuracy                           0.77        61
   macro avg       0.77      0.77      0.77        61
weighted avg       0.77      0.77      0.77        61
```

Figure 24: Classification Report of Decision Tree

Final Comparison between the algorithms

Evaluation Metrics	Logistic Regression		Random Forest		Decision Tree	
Accuracy	82%		80%		77%	
	0	1	0	1	0	1
Precision	0.79	0.84	0.73	0.81	0.75	0.79
Recall	0.82	0.82	0.79	0.76	0.75	0.79
F1-score	0.81	0.83	0.76	0.78	0.75	0.79
Support	28	33	28	33	28	33

From doing this project it can be concluded that there are a lot of other machine learning algorithms left to experiment with which can give an even higher accuracy rate. Machine learning can vastly increase our understanding of curing and predicting human and other diseases if experimented on enough to find the perfect algorithm, there is a huge scope for machine learning algorithms in predicting cardiovascular diseases or heart related diseases as well. As the future is AI and machine learning combining them with the medical world is very necessary.

1. https://www.kaggle.com/datasets/cherngs/heart-disease-cleveland-uci
2. https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML#:~:text=Machine%20learning%20(ML)%20is%20a,to%20predict%20new%20output%20values.
3. https://www.javatpoint.com/supervised-machine-learning#:~:text=Supervised%20learning%20is%20the%20types,tagged%20with%20the%20correct%20output.
4. https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/#:~:text=Matplotlib%20is%20a%20cross%2Dplatform,embed%20plots%20in%20GUI%20applications.
5. https://www.w3schools.com/python/numpy/numpy_random_seaborn.asp
6. https://machinelearningmastery.com/how-to-use-correlation-to-understand-the-relationship-between-variables/
7. https://towardsdatascience.com/understanding-train-test-split-scikit-learn-python-ea676d5e3d1
8. https://towardsdatascience.com/understanding-the-confusion-matrix-and-how-to-implement-it-in-python-319202e0fe4d
9. https://thecleverprogrammer.com/2021/07/07/classification-report-in-machine-learning/
10. https://numpy.org/doc/stable/reference/generated/numpy.reshape.html
11. https://github.com/krishnaik06/Predicting-Heart-Disease/blob/master/Heart%20Disease%20Predictions.ipynb
12. https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)#:~:text=Cardiovascular%20diseases%20(CVDs)%20are%20the,to%20heart%20attack%20and%20stroke.
13. https://www.healthline.com/health/heart-disease/statistics#Who-is-at-risk?