MACHINE LEARNING PROJECT

NAME: BISHMAY RANJAN SAHOO

STUDENT ID: 19SDATDEL018

AND

NAME: A.C.HARSITA PANDA

STUDENT ID: 19SDATDEL019

TOPIC: CLASSIFICATION OF HEART DISEASE

COURSE: DATA ANALYTICS AND MACHINE LEARNING

SUBMITTED TO: Technex IIT BHU Varanasi,

EISYSTEM

ABSTRACT

In this project, we were asked to experiment with a real world data set and to explore how machine learning algorithms can be used to find the patterns in data. We were expected to gain experience using a common data mining and machine learning library and were expected to submit a report about the data set and the algorithms used. After performing the required tasks on a data set of our choice, herein lies my final project report.

Keywords: Machine learning, Classification, Supervised learning, Neural Network, Decision Tree.

INTRODUCTION

Of all the applications of machine-learning, diagnosing any serious disease using a black box is always going to be a hard sell. If the output from a model is the particular course of treatment (potentially with side-effects), or surgery, or the *absence* of treatment, people are going to want to know **why.**

This data set gives a number of variables along with a target condition of having or not having heart disease. Below, the data is first used in a simple Neural Network and Decision Tree model, and then the model is compared between the two techniques.

About the Heart Disease:

Diagnosis:

The diagnosis of heart disease is done on a combination of clinical signs and test results. The types of tests run will be chosen on the basis of what the physician thinks is going on ranging from electrocardiograms and cardiac computerized tomography (CT) scans, to blood tests and exercise stress tests.

More info:
https://www.mayoclinic.org/diseases-conditions/heart-disease/diagnosis-treatment/drc-20353124

https://www.heartfoundation.org.au/your-heart/living-with-heart-disease/medical-tests

https://www.bhf.org.uk/informationsupport/risk-factors

https://www.heart.org/en/health-topics/heart-attack/understand-your-risks-to-prevent-a-heart-attack

RISK FACTORS:

Looking at information of heart disease risk factors led me to the following:

high cholesterol, high blood pressure, diabetes, weight, family history and smoking . According to another source 4, the major factors that can't be changed are: **increasing age, male gender and heredity**. Note that **Tallahassee**, one of the variables in this data set, is heredity. Major factors that can be modified are: **Smoking, high cholesterol, high blood pressure, physical inactivity, and being overweight and having diabetes**. Other factors include **stress, alcohol and poor diet/nutrition**.

There is no reference to the 'number of major vessels', but given that the definition of heart disease is **"...what happens when your heart's blood supply is blocked or interrupted by a build-up of fatty substances in the coronary arteries"**, it seems

logical the *more* major vessels is a good thing, and therefore will reduce the probability of heart disease.

Given the above, We would hypothesis that, if the model has some predictive ability, we'll see these factors standing out as the most important.

ABOUT DATA SET:

This database contains 303 samples and 14 features. We have a data which classified if patients have heart disease or not according to features in it. We will try to use this data to create a model which tries predict if a patient has this disease or not.

DATA-SETS:

It's a clean, easy to understand set of data. However, the meaning of some of the column headers are not obvious. Here's what they mean,

- **age**: The person's age in years
- **sex**: The person's sex (1 = male, 0 = female)
- **cp:** The chest pain experienced (Value 1: typical angina, Value 2: atypical angina, Value 3: Non-Angeline pain, Value 4: asymptomatic)
- **trestbps:** The person's resting blood pressure (mm Hg on admission to the hospital)
- **chol:** The person's cholesterol measurement in mg/dl

- **fbs:** The person's fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false)
- **restecg:** Resting electrocardiogram measurement (0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria)
- **thalach:** The person's maximum heart rate achieved
- **exang:** Exercise induced angina (1 = yes; 0 = no)
- **oldpeak:** ST depression induced by exercise relative to rest ('ST' relates to positions on the ECG plot.)
- **slope:** the slope of the peak exercise ST segment (Value 1: upsloping, Value 2: flat, Value 3: Down sloping)
- **ca:** The number of major vessels (0-3)
- **thal:** A blood disorder called Tallahassee (3 = normal; 6 = fixed defect; 7 = reversable defect)
- **target:** Heart disease (0 = no, 1 = yes)

OUR APPROACH:

We are doing this project through knn, decision tree, svm, logistic regression and neural network.

1. SKLEARN MODEL AND PACKAGES USED:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score,jaccard_similarity_score,confusion_matrix,classification_report,auc,roc_curve
from matplotlib import pyplot as plt
from sklearn import tree
import numpy as np
import pydotplus
import seaborn as sb
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.neural_network import MLPClassifier
```

2. READING DATA:

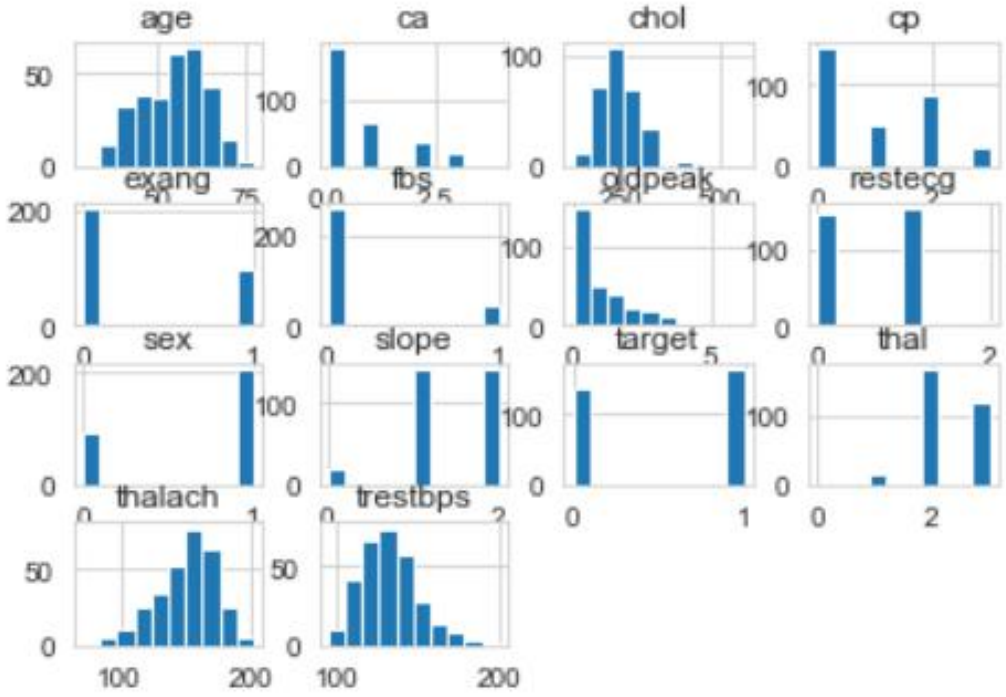
```
my_data = pd.read_csv(r"C:/Users/KIIT/Desktop/DataSets-master/DataSets-master/heart.csv")
```

```
my_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

3. HITOGRAM REPRESENTATION OF DATA:

```
my_data.hist()
plt.show()
```



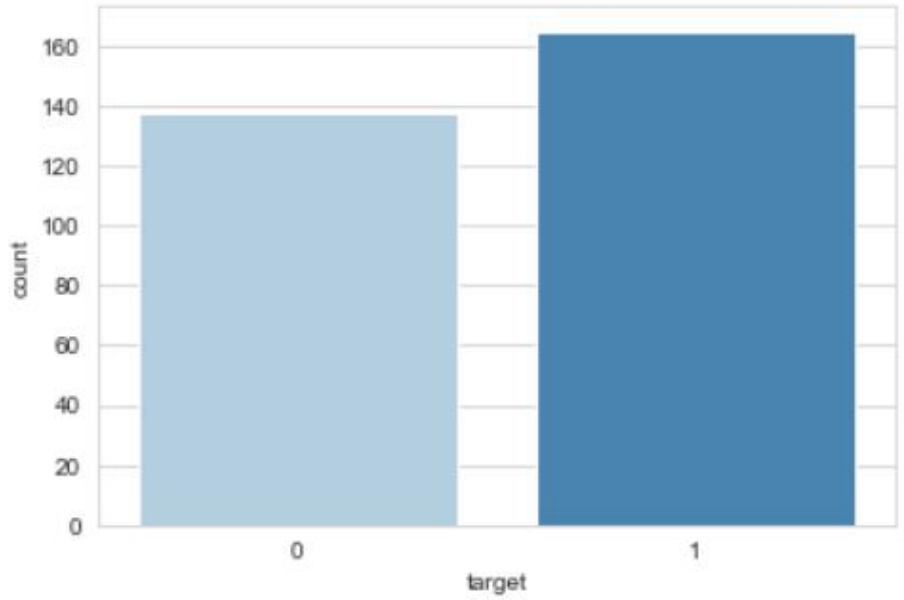
4. CORRELATION BETWEEN THE DATA:

```
my_data.corr()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.398522	0.096801	0.210013	-0.168814	0.276326	0.068001	-0.225439
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.044020	0.141664	0.096093	-0.030711	0.118261	0.210041	-0.280937
cp	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.295762	-0.394280	-0.149230	0.119717	-0.181053	-0.161736	0.433798
stbps	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.046698	0.067616	0.193216	-0.121475	0.101389	0.062210	-0.144931
chol	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.009940	0.067023	0.053952	-0.004038	0.070511	0.098803	-0.085239
fbs	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.008567	0.025665	0.005747	-0.059894	0.137979	-0.032019	-0.028046
stecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.044123	-0.070733	-0.058770	0.093045	-0.072042	-0.011981	0.137230
alach	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.000000	-0.378812	-0.344187	0.386784	-0.213177	-0.096439	0.421741
xang	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.378812	1.000000	0.288223	-0.257748	0.115739	0.206754	-0.436757
dpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.344187	0.288223	1.000000	-0.577537	0.222682	0.210244	-0.430696
slope	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.386784	-0.257748	-0.577537	1.000000	-0.080155	-0.104764	0.345877
ca	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.213177	0.115739	0.222682	-0.080155	1.000000	0.151832	-0.391724
thal	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.096439	0.206754	0.210244	-0.104764	0.151832	1.000000	-0.344029
arget	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.421741	-0.436757	-0.430696	0.345877	-0.391724	-0.344029	1.000000

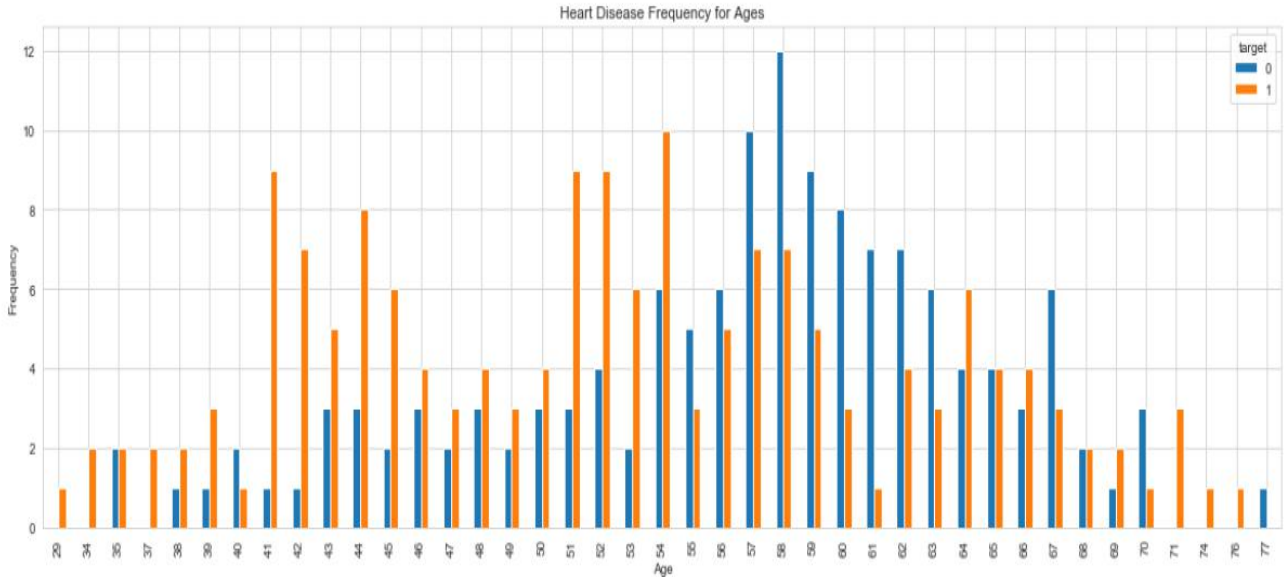
5. TOTAL NO OF TARGETS:

```
sb.countplot(x="target", data=my_data, palette="Blues")
plt.show()
```



6. HEART DISEASE FREQUENCY FOR AGES:

```
pd.crosstab(my_data.age,my_data.target).plot(kind="bar",figsize=(20,6))
plt.title('Heart Disease Frequency for Ages')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.savefig('heartDiseaseAndAges.png')
plt.show()
```



7. CHECKING FOR MISSING VALUES:

```
my_data.isnull().sum()
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

8. DECISION TREE:

8.1: Accuracy Score & Jaccard Similarity Score:

```
dt_acc = accuracy_score(ya,dt_yp)
dt_jss = jaccard_similarity_score(ya,dt_yp)
print("Accuraccy Score is {} and Jaccard Similarity Score is {}".format(dt_acc,dt_jss))
```

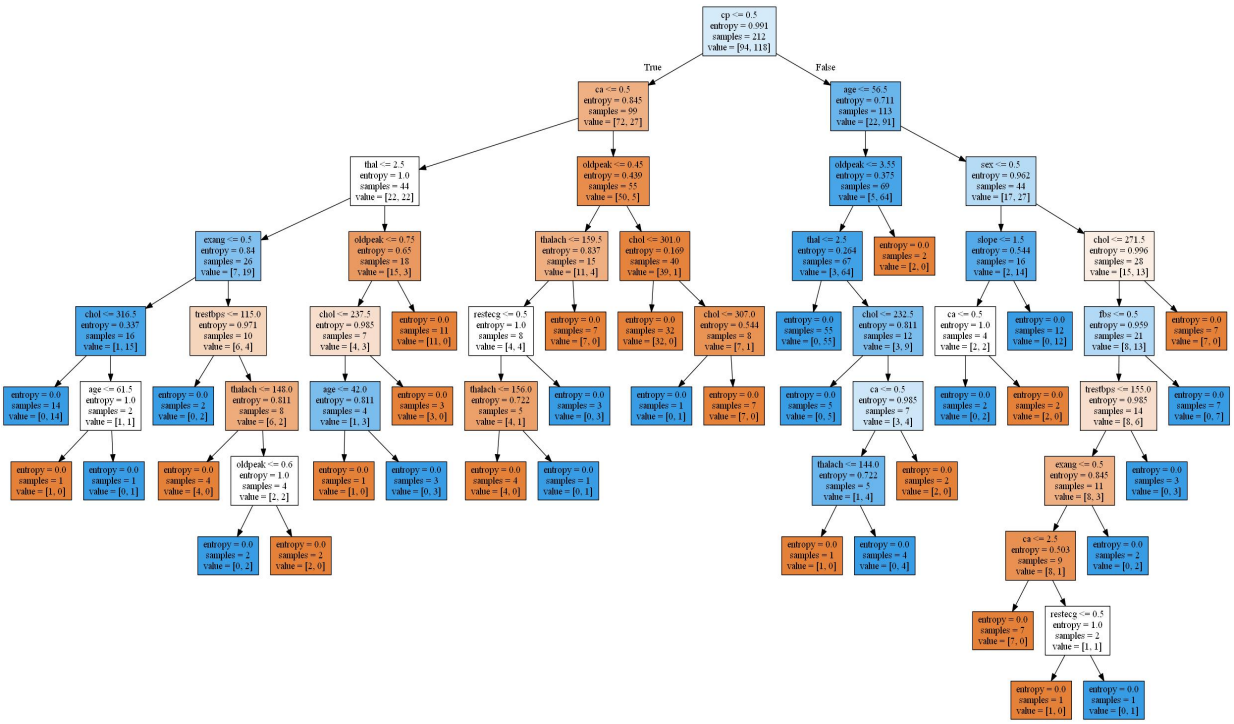
```
Accuraccy Score is 0.8351648351648352 and Jaccard Similarity Score is 0.8351648351648352
```

8.2: Classification Report:

```
dt_cm = confusion_matrix(ya,dt_yp)
dt_report = classification_report(ya,dt_yp)
print(dt_report)
```

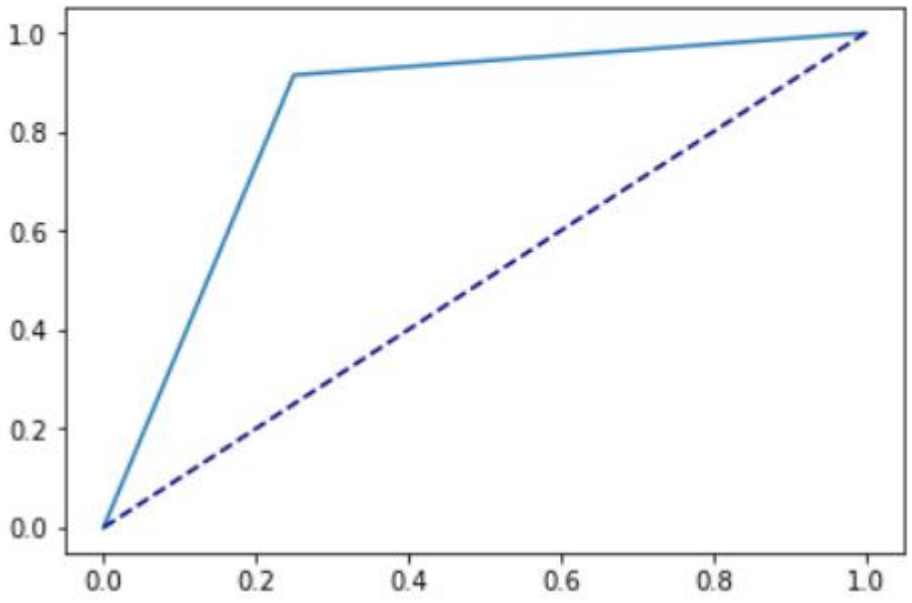
```
              precision    recall  f1-score   support

           0       0.91      0.73      0.81        44
           1       0.79      0.94      0.85        47

   micro avg       0.84      0.84      0.84        91
   macro avg       0.85      0.83      0.83        91
weighted avg       0.85      0.84      0.83        91
```

8.3: Tree:



8.4: Roc curve:

```
plt.plot(fpr,tpr,label = "Roc Curve")
plt.plot([0, 1], [0, 1], color='navy', linestyle='--')
plt.show()
```



8.5: AUC:

```
area  = auc(fpr,tpr)
print("Area under the curve is: {}".format(area))
```

```
Area under the curve is: 0.8324468085106383
```

9. K NEAREST NEIGHBOURS:

9.1: Accuracy Score & Jaccard Similarity Score:

```
knn_acc = accuracy_score(ya,knn_yp)
knn_jss = jaccard_similarity_score(ya,knn_yp)
print("Accuraccy Score is {} and Jaccard Similarity Score is {}".format(knn_acc,knn_jss))
```

```
Accuraccy Score is 0.7142857142857143 and Jaccard Similarity Score is 0.7142857142857143
```

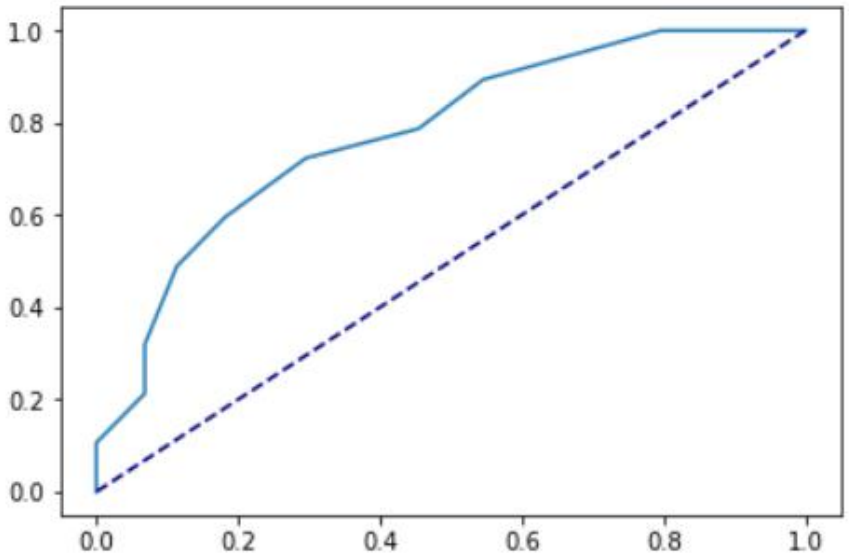
9.2: Classification Report:

```
knn_cm = confusion_matrix(ya,knn_yp)
knn_report = classification_report(ya,knn_yp)
print(knn_report)
```

```
              precision    recall  f1-score   support

           0       0.70      0.70      0.70        44
           1       0.72      0.72      0.72        47

   micro avg       0.71      0.71      0.71        91
   macro avg       0.71      0.71      0.71        91
weighted avg       0.71      0.71      0.71        91
```

9.3: Roc curve:

```
plt.plot(knn_fpr,knn_tpr,label = "Roc Curve")
plt.plot([0, 1], [0, 1], color='navy', linestyle='--')
plt.show()
```



9.4: AUC:

```
print("Area under the curve is: {}".format(auc(knn_fpr,knn_tpr)))
```

```
Area under the curve is: 0.7790135396518375
```

10: Support Vector Machine (SVM):

10.1 Accuracy Score and Jaccard Similarity Score:

```
svm_acc = accuracy_score(ya,svm_yp)
svm_jss = jaccard_similarity_score(ya,svm_yp)
print("Accuraccy Score is {} and Jaccard Similarity Score is {}".format(svm_acc,svm_jss))
```

```
Accuraccy Score is 0.7032967032967034 and Jaccard Similarity Score is 0.7032967032967034
```

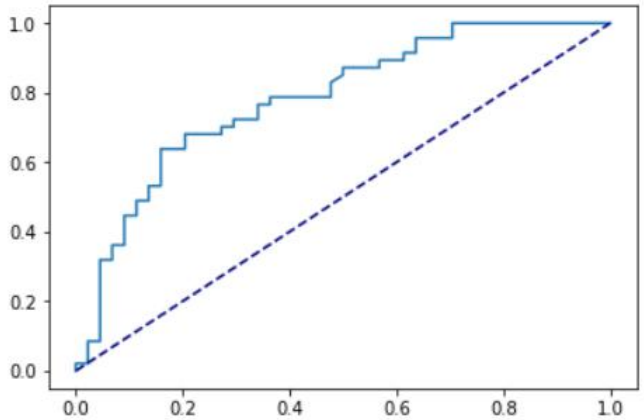
10.2: Classification Report:

```
svm_cm= confusion_matrix(ya,svm_yp)
svm_report = classification_report(ya,svm_yp)
print(svm_report)
```

```
              precision    recall  f1-score   support

           0       0.72      0.64      0.67        44
           1       0.69      0.77      0.73        47

   micro avg       0.70      0.70      0.70        91
   macro avg       0.71      0.70      0.70        91
weighted avg       0.70      0.70      0.70        91
```

10.3: ROC Curve:

```
plt.plot(svm_fpr,svm_tpr,label = "Roc Curve")
plt.plot([0, 1], [0, 1], color='navy', linestyle='--')
plt.show()
```



10.4: AUC:

```
print("Area under the curve is: {}".format(auc(svm_fpr,svm_tpr)))
```

```
Area under the curve is: 0.7831237911025144
```

11: LOGISTIC REGRESSION:

11.1: Accuracy Score & Jaccard Similarity Score:

```
lr_acc = accuracy_score(ya,lr_yp)
lr_jss = jaccard_similarity_score(ya,lr_yp)
print("Accuraccy Score is {} and Jaccard Similarity Score is {}".format(lr_acc,lr_jss))
```

```
Accuraccy Score is 0.8791208791208791 and Jaccard Similarity Score is 0.8791208791208791
```

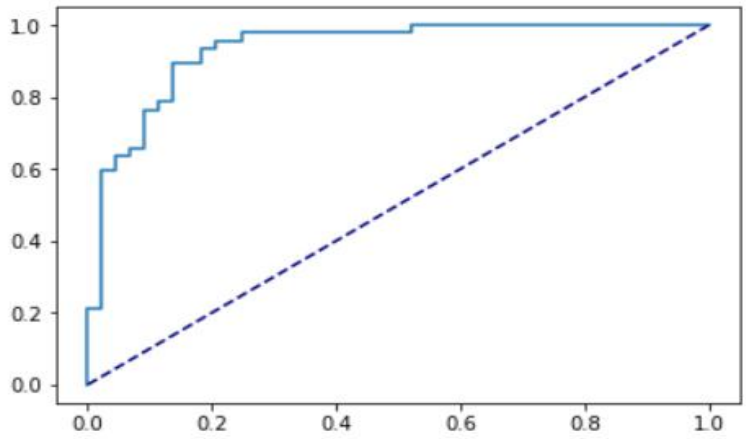
11.2 : Classification Report:

```
lr_cm= confusion_matrix(ya,lr_yp)
lr_report = classification_report(ya,lr_yp)
print(lr_report)
```

```
              precision    recall  f1-score   support

           0       0.95      0.80      0.86        44
           1       0.83      0.96      0.89        47

   micro avg       0.88      0.88      0.88        91
   macro avg       0.89      0.88      0.88        91
weighted avg       0.89      0.88      0.88        91
```

11.3: ROC Curve:

```
plt.plot(lr_fpr,lr_tpr,label = "Roc Curve")
plt.plot([0, 1], [0, 1], color='navy', linestyle='--')
plt.show()
```



11.4: AUC:

```
print("Area under the curve is: {}".format(auc(lr_fpr,lr_tpr)))
```

```
Area under the curve is: 0.9327852998065764
```

12: Neural Network:

12.1: Accuracy score And Jaccard Similarity Score:

```
nn_acc = accuracy_score(ya,nn_yp)
nn_jss = jaccard_similarity_score(ya,nn_yp)
print("Accuraccy Score is {} and Jaccard Similarity Score is {}".format(nn_acc,nn_jss))
```

```
Accuraccy Score is 0.8351648351648352 and Jaccard Similarity Score is 0.8351648351648352
```

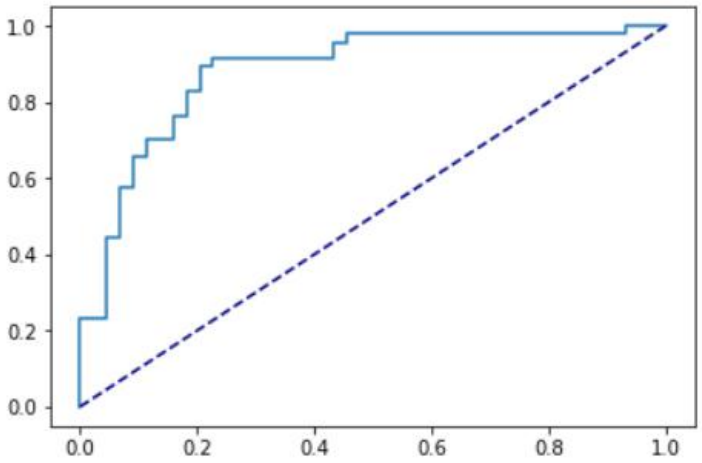
12.2: Classification Report:

```
nn_cm= confusion_matrix(ya,nn_yp)
nn_report = classification_report(ya,nn_yp)
print(nn_report)
```

```
              precision    recall  f1-score   support

           0       0.89      0.75      0.81        44
           1       0.80      0.91      0.85        47

   micro avg       0.84      0.84      0.84        91
   macro avg       0.84      0.83      0.83        91
weighted avg       0.84      0.84      0.83        91
```

12.3: ROC Curve:

```
plt.plot(nn_fpr,nn_tpr,label = "Roc Curve")
plt.plot([0, 1], [0, 1], color='navy', linestyle='--')
plt.show()
```

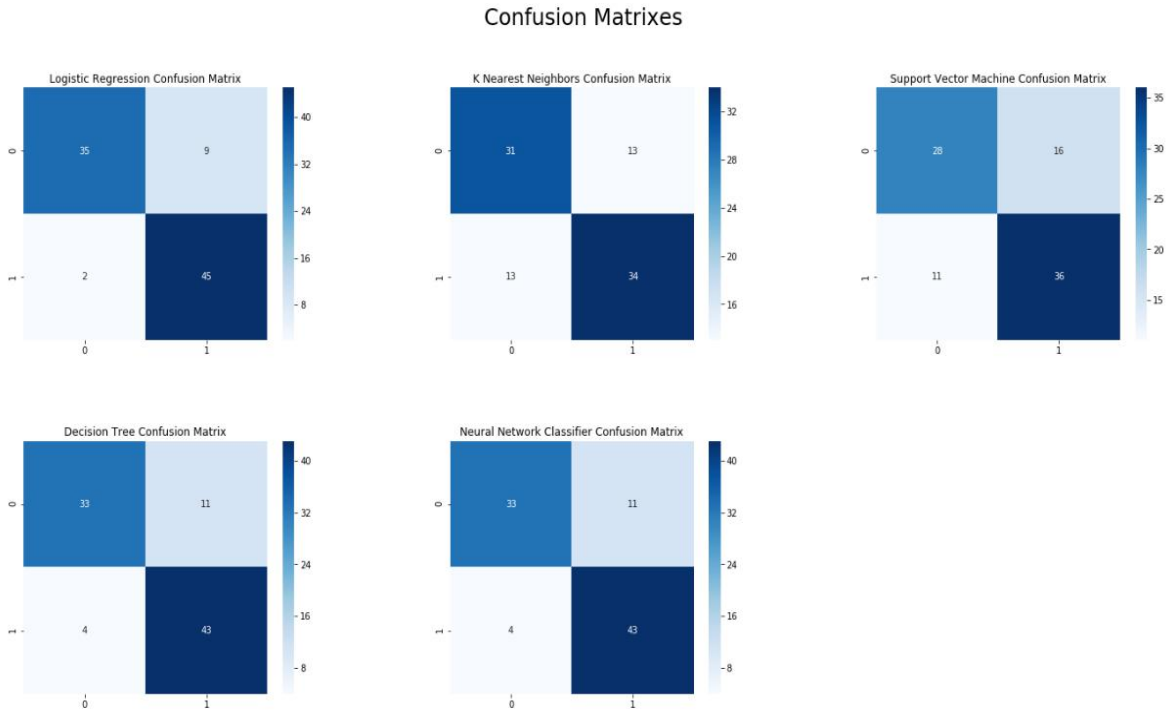


12.4: AUC:

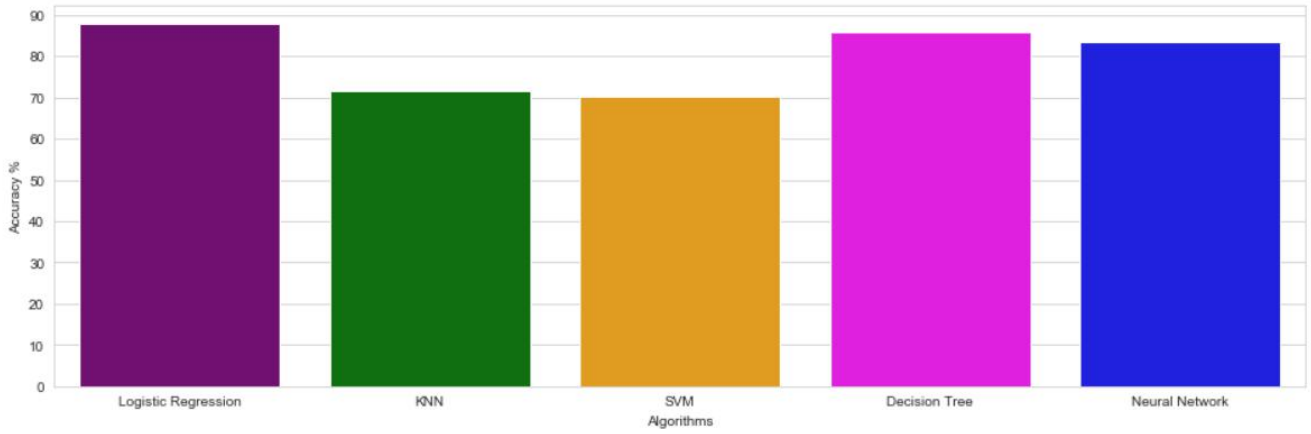
```
print("Area under the curve is: {}".format(auc(nn_fpr,nn_tpr)))
```

```
Area under the curve is: 0.8815280464216634
```

13: CONFUSION MATRIXES:



14: COMPARISION BETWEEN ALL THE MODLES:



15: CONCLUSION:

From this project we conclude that there is no data missing and it's a classification problem. From all the models used, logistic regression gives 87.91%, the best performance and the svm gives 70.32%, the least performance. In future, if similar studies are conducted to generate the data set used in this report, more feature and samples need to be calculated so that the models can do better calculations and more accurate model can be constructed.

16: References:

- ✓ https://www.google.com/
- ✓ Hands on machine learning by Aurelien Geron
- ✓ Kaggle
- ✓ https://scikit-learn.org/stable/

https://www.bhf.org.uk/informationsupport/risk-factors

- ✓ https://archive.ics.uci.edu
- ✓ https://github.com/chandanverma07/MachineLearningPdf/blob/master/sktlearn.pdf
- ✓ Code can be viewed from: https://github.com/bishmayRanjanSahoo/ML_PROJECT_ON_CLASSIFICATION_OF_HEART_DISEASE