Intro to Artificial Intelligence
Section C

Group Members:

Haris Amin – 231494566
Muhammad Ammar Ahmad – 231492719
Fahad Siddiqui – 231471220

Topic For the Project:

Heart Disease Prediction

FORMAN CHRISTIAN COLLEGE
(A CHARTERED UNIVERSITY)

Table of Contents

Abstract

We have focused upon the heart attack prediction before it happens by measuring the attributes such as main the heartbeat blood pressure and the blood sugar. The main purpose of this project and choosing this dataset is that thousands of people die each day due to sudden heart attack and majority of the death cases reasons are the only that they could not get the first Aid as in there really 5-10 minutes. Now, by using this reject you required only to enter up your all required things into the software, and it simply acts as a doctor and predicts that what percentage of you are at the risk f the heart attack.

The benefit of using this software is clear that this predicts your heat attack before it happens, and it is lifesaving software being known if understand the importance and usability of this idea.

In this project we have used Three algorithms for getting the best results from the dataset which included K-NN algorithm which produces 64% percent accurate predictions but as we also use Logistic Regression and expectedly we have getting the better results from the K-NN which is 85% after giving epochs of 1000 iterations, and lastly we used Random forest which gave us 88% accurate predictions but as we know as this is medical related work we should achieve minimum the 90% of accuracy cause this is mater of someone health when people start relay upon this program, an we concluded this article here with a note that this project would get evolved as more research and finest data is collected. It is hence proven that both Random forest and Logistic Regression are better machine learning models to be used on this data set of patients to predict if they are suffering from a heart disease or not.

The most stable out of two more accurate classifiers is Logistic Regression as it gives one value of accuracy after epochs of 1000 whereas Random Forest would give different accuracy value for different iterations but not too different from one another but it reaches to a value at last. As for the future work the most approachable thing which is that the program could only use some of the main attributes and prediction should be goes up to the extent of the 90% then after which this software is able to relay upon it in real world.

Literature Review

Heart disease is becoming one of most common diseases in all over the world and many lives are just wasted because of late first aid or any medical facility. These deaths can be minimized if the heart disease can be diagnosed on an early stage or before a heart attack. Some algorithms are used to show the data graphically for better understanding, algorithms which have been used include K-NN, naïve Bayes, Decision tree J48, SVM, Adaboost, Stochastic Gradient Decent (SGD) and Decision Table (DT). The accuracy with the KNN(k=1) is of 99.7073. Different classifiers have been used and compared to classify the HD dataset. A feature extraction method was performed using Classifier Subset Evaluator on the HD dataset, and results show enhanced performance in term of the classification accuracy for K-NN (N = 1) and Decision Table classifiers to 100 and 93.8537% respectively after using the selected features by only applying a combination of up to 4 attributes instead of 13 attributes for the predication of the HD cases. The benefit of the having a reliable feature selection method for disease prediction by using minimal number of attributes instead of having to consider all available attributes.

The medical care ventures gather tremendous measures of information that contain some secret data, which is helpful for settling on successful choices. For giving proper outcomes and settling on successful choices on information, some high-level information mining procedures are utilized. In this investigation, a viable heart disease expectation framework (EHDPS) is created utilizing neural organization for anticipating the danger level of heart disease. The framework utilizes 15 clinical boundaries like age, sex, pulse, cholesterol, and stoutness for forecast. The EHDPS predicts the probability of patients getting heart disease. It empowers critical information, e.g., connections between clinical components identified with heart disease and examples, to be set up. We have utilized the multi-facet perceptron neural organization with backpropagation as the preparation calculation. The results have delineated that the planned symptomatic framework can successfully foresee the danger level of heart sicknesses.

An EHDPS has been introduced utilizing information mining procedures. From ANN, a MLPNN along with BP calculation is utilized to foster the framework. The MLPNN model demonstrates the better outcomes and helps the area specialists and surprisingly the individual identified with the clinical field to get ready for a superior and early analysis for the patient. This framework performs sensibly well even without retraining. Moreover, the test results show that the framework predicts heart disease with ∼100% exactness by utilizing neural organizations.

Heart diseases are getting more common in people and it’s a serious disease. So, it’s better to predict it so that people can take extra precautions to avoid serious illness. These articles suggest using KNN, SVM, NB, DT, and RF algorithms for prediction of the heart disease using the Cleveland data set.

Many researchers are using machine learning models to predict cardiovascular diseases among the people. Since the cardiovascular diseases are getting more common. This disease is fatal in most of the cases so an early diagnosis of disease may control the death rate due to these diseases. An early detection can save lives of million, so it is important to use machine learning models to predict this disease. The author has used logistic regression model for the prediction in the data set. There are some drawbacks in the research paper. The biggest problem is that the author has predicted the risk of heart disease, COVID-19, and diabetes in an individual based on answering a few questions related to various factors like travel history, age, gender, and blood pressure. The author should have used machine learning model for the prediction of only cardiovascular diseases. Our research paper focuses mainly on the prediction of cardiovascular diseases and also uses logistic regression model.

Cardiovascular diseases or more commonly known as diseases of the heart are getting more common day and people are getting contracted to it. These cardiovascular diseases are linked to many other diseases which can be life threating. So, machine learning models can be used to determine the risk of cardiovascular diseases. Researchers apply many techniques of data analysis to predict diseases. The author has used Naïve Bayes, decision tree, K-nearest neighbor, and random forest algorithms. The dataset comprises of 303 instances and 76 attributes out of which the author has used 14 attributes. The results were the highest using the KNN model. The author has used 14 attributes for this prediction which are more, and the author has used KNN model for prediction. With more attributes, the KNN doesn't give accurate results. The lesser the attributes are the more precise the predictions are. Our research paper is much better in the way that we have used lesser attributes and we have used logistic regression model for our prediction.

Methodology and Experiment

The dataset that we have chosen for our research is the "heart.csv" dataset which contained patients' information leading to the factor that if they suffer from heart disease or not. Our Data has 303 instances and 14 attributes. The attributes are as follows:

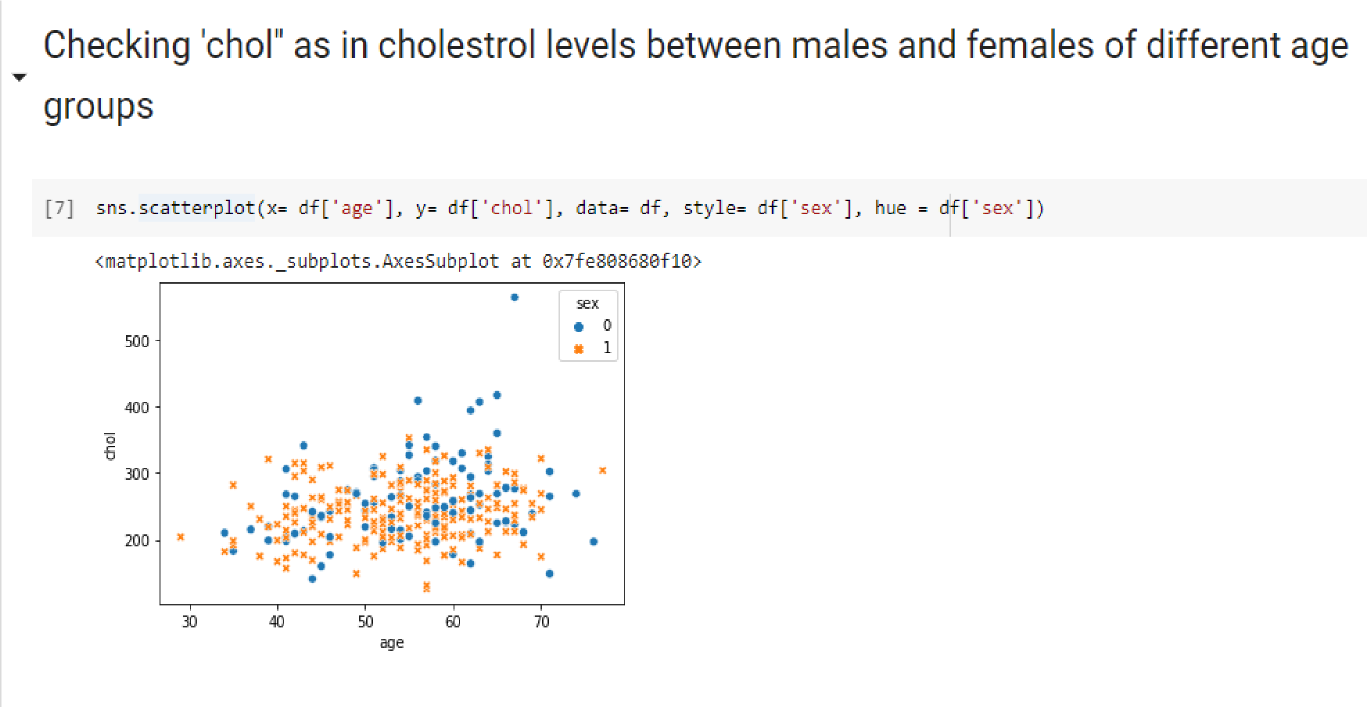
1. Age
2. Sex
3. Chest pain type (4 values)
4. Resting blood pressure
5. Serum cholestoral in mg/dl
6. Fasting blood sugar > 120 mg/dl
7. Resting electrocardiographic results (values 0,1,2)
8. Maximum heart rate achieved
9. Exercise induced angina
10. Oldpeak = ST depression induced by exercise relative to rest
11. The slope of the peak exercise ST segment
12. Number of major vessels (0-3) colored by flourosopy
13. Thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

The 14th one being the output variable/label. We have to make predictions based on the above listed factors that if the patient has a heart disease or not. We have used three machine learning models on our dataset; K-Nearest neighbour, Logistic Regression and Random Forest. It is really useful program that is to be used in hospitals and clinics to predict heart disease in patients beforehand so treatment and cure could be provided to the patients on time and the ones who are not suffering from heart disease can be made cleared by doctors.

We started off by extracted the dataset by using Pandas library. We used some of the library's methods to better analyze our data.

Moving forward, we used two libraries Matplotlib and Seaborne to visualize our data and to understand it. We constructed a scatterplot to see relationship between two attributes and checked their variation between both the genders.

Results

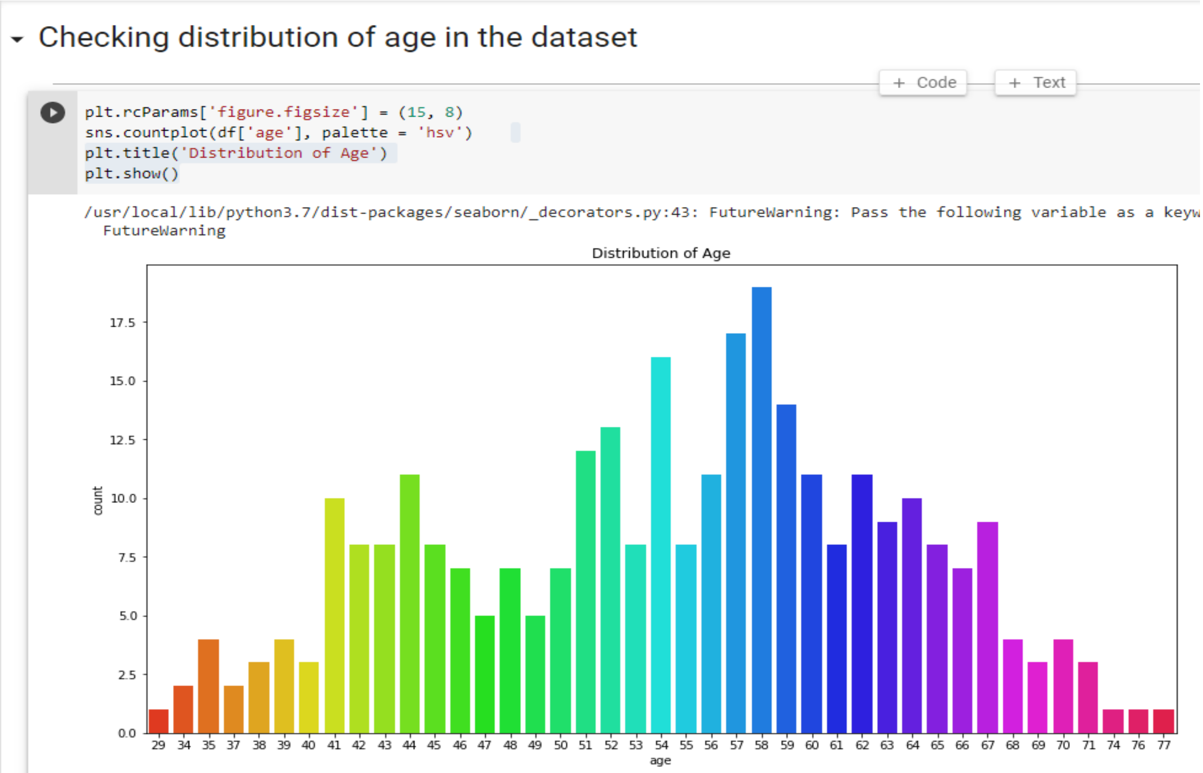


Checking 'chol" as in cholestrol levels between males and females of different age groups

```
[7] sns.scatterplot(x= df['age'], y= df['chol'], data= df, style= df['sex'], hue = df['sex'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe808680f10>
```

Bar Graph

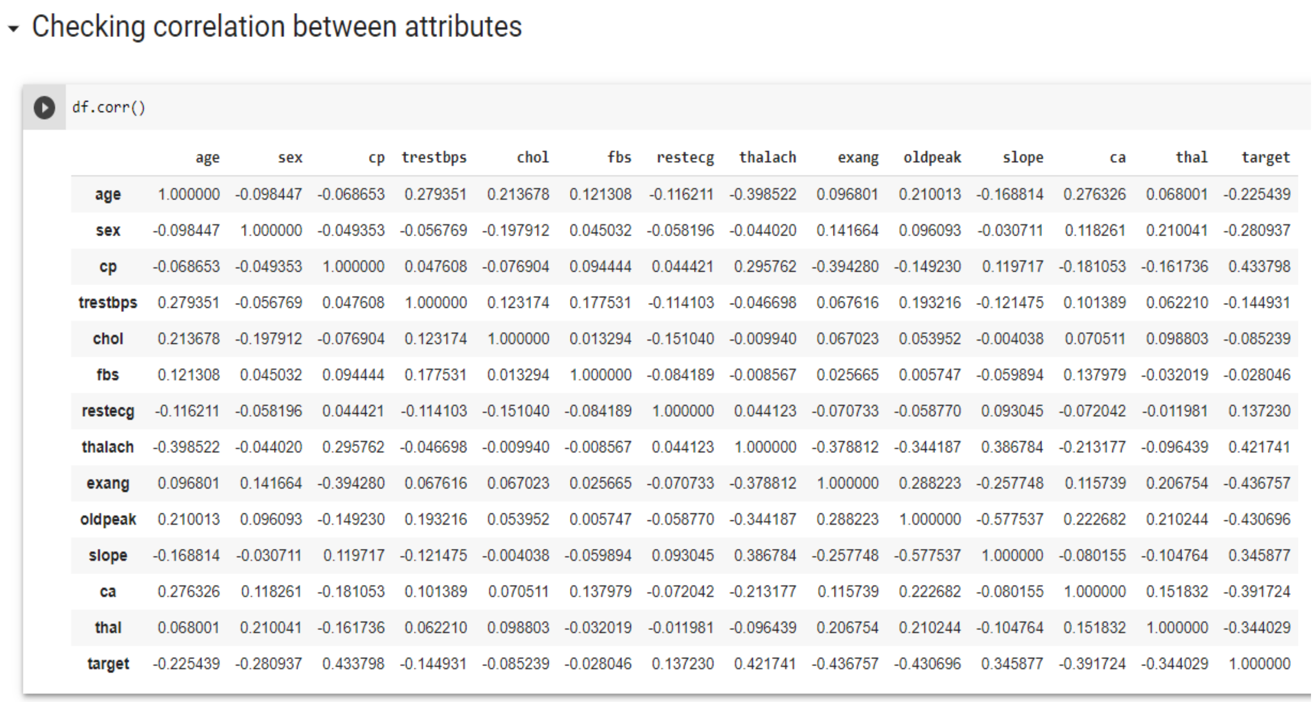


Checking distribution of age in the dataset

```
plt.rcParams['figure.figsize'] = (15, 8)
sns.countplot(df['age'], palette = 'hsv')
plt.title('Distribution of Age')
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyw
  FutureWarning
```

We then started **Pre-processing** of our data. We used Skit learn's StandardScalar() method to normalize the data so its noise can be removed. We also used some of the pandas methods to understand our label class which contains two unique classes; 1 (binary for 'yes') and 0 (binary for 'no'). As part of data pre-processing the irrelevant attributes are to be dropped in order to get accurate prediction results. Correlation analysis was conducted to study relationship between all attributes.



Checking correlation between attributes

```
df.corr()
```

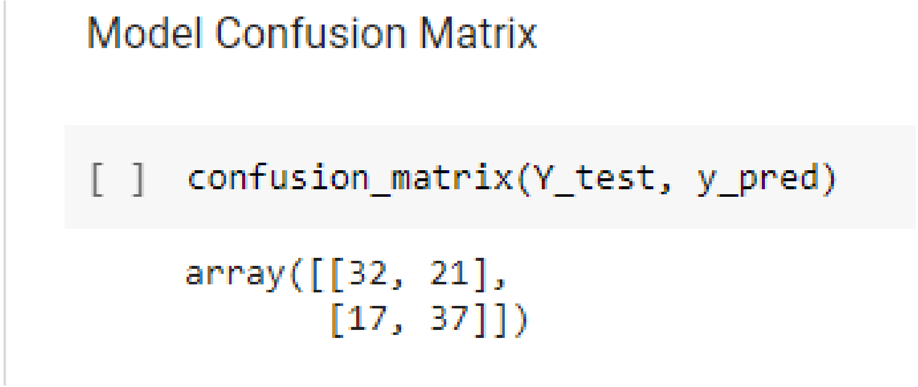
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.398522	0.096801	0.210013	-0.168814	0.276326	0.068001	-0.225439
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.044020	0.141664	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
trestbps	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
restecg	-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
thalach	-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
exang	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
oldpeak	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
target	-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

After the study the negatively correlated attributes were dropped from the dataset. Now instead of all 14 attributes, 11 are being used and out of those 11 one is the output attributes so in real we are using 10 attributes to train and test our models. We drop the 'target' label/output class/attribute from the dataset. And store it to a variable Y and all the remaining 10 variables are stored in X since they are the input variables. The same pre-processing techniques have been used on all three of our machine learning models which we used.

We now have to split the data into Test-Train mechanism. We have now done test-train split as follows:

- For K-Nearest Neighbor Machine learning model:
  We split the data by using the test size of 0.35 meaning we used 65% of the data for training the model and 35% of the data for testing the model. We used Skit learn library to import and implement KNN model on the data. We used **3 as the value of k (nearest neighbor**). It is preferred to use odd numbered k to get better results. We fit the training data into the model and we then made the predictions by using the test data.
  An accuracy of **64.5%** was achieved.

We used several evaluation techniques to measure accuracy of the system such as Mean Squared Error which is **0.355**. We then used confusion matrix which is attached as follows:

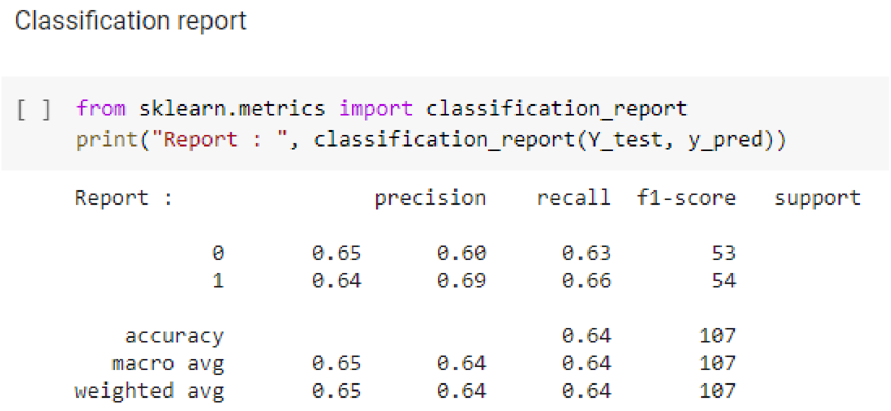


Model Confusion Matrix

```
[ ]  confusion_matrix(Y_test, y_pred)

     array([[32, 21],
            [17, 37]])
```

We used f1-score technique and got these as results:



Classification report

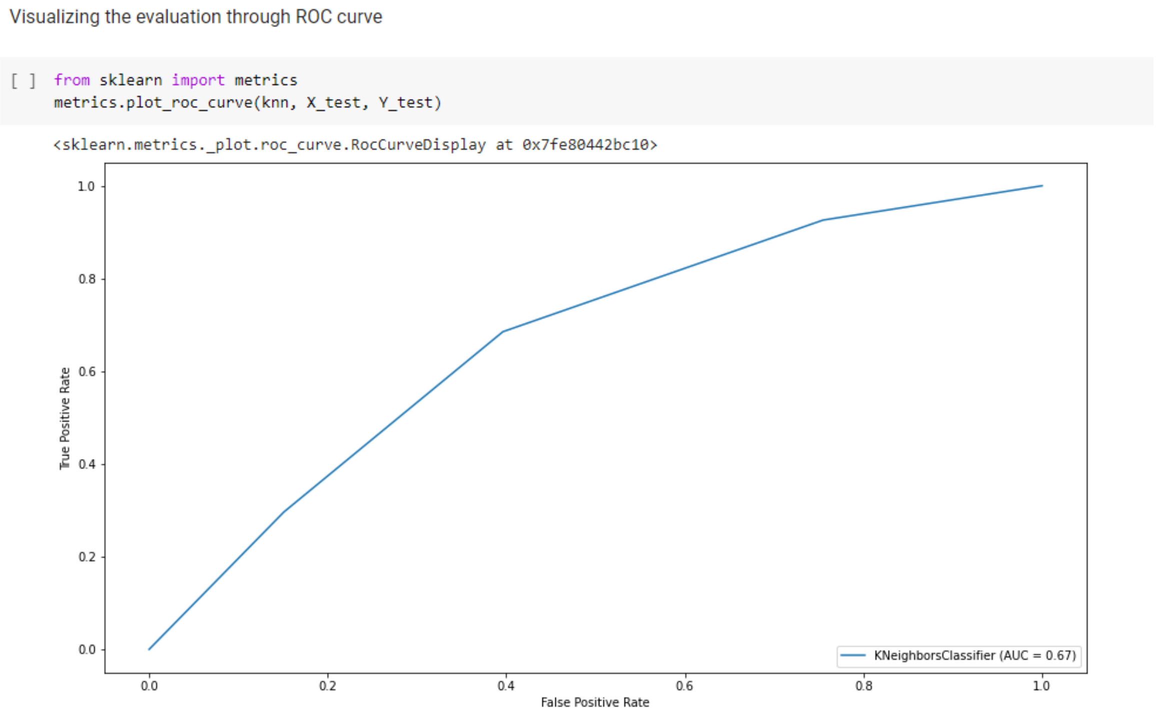
```
[ ]  from sklearn.metrics import classification_report
     print("Report : ", classification_report(Y_test, y_pred))

     Report :               precision    recall  f1-score   support

                0       0.65      0.60      0.63        53
                1       0.64      0.69      0.66        54

         accuracy                           0.64       107
        macro avg       0.65      0.64      0.64       107
     weighted avg       0.65      0.64      0.64       107
```

We then visualized our evaluation through ROC curve:



Visualizing the evaluation through ROC curve

```
[ ]  from sklearn import metrics
     metrics.plot_roc_curve(knn, X_test, Y_test)
```

```
<sklearn.metrics._plot.roc_curve.RocCurveDisplay at 0x7fe80442bc10>
```

We used Numpy to create an array of new unseen user-given data points and tested the model on it too.

We **now used 5 as value of k** and we discovered that this value lowered our accuracy to 59% from 64% with a mean squared error of 40%.

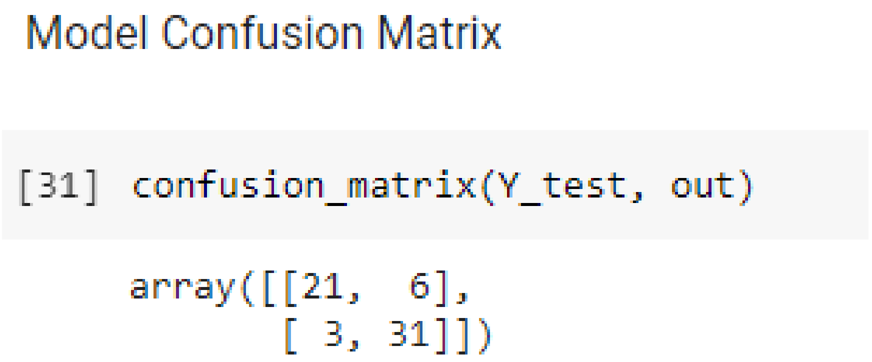
- For Logistic Regression:

  We have now used Logistic Regression which is proven to be fairly suitable to classify when we have binary attributes and binary target variable and in our case our target variable is binary.

  We split the data by using the test size of 0.20 meaning we used 80% of the data for training the model and 20% of the data for testing. We used Skit Learn library to import and implement Logistic regression model and we have set max_iter to 1000 meaning that we will do epochs of 1000 iterations. We fit the training data into the model and we then made the predictions by using the test data.

  An accuracy of **85.24%** is achieved.

We used several evaluation techniques to measure accuracy of the system such as Mean Squared Error which is 14.75%. We then used confusion matrix which is attached as follows:

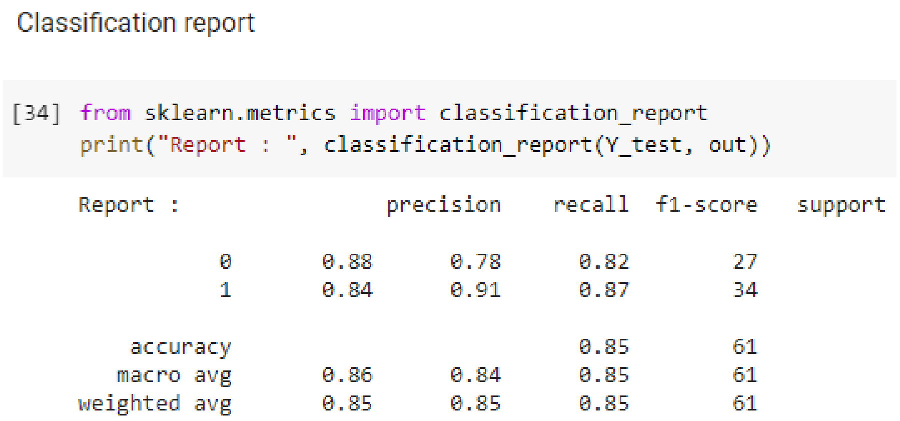


Model Confusion Matrix

```
[31]  confusion_matrix(Y_test, out)

      array([[21,  6],
             [ 3, 31]])
```

We used f1-score technique and got these as results:



Classification report

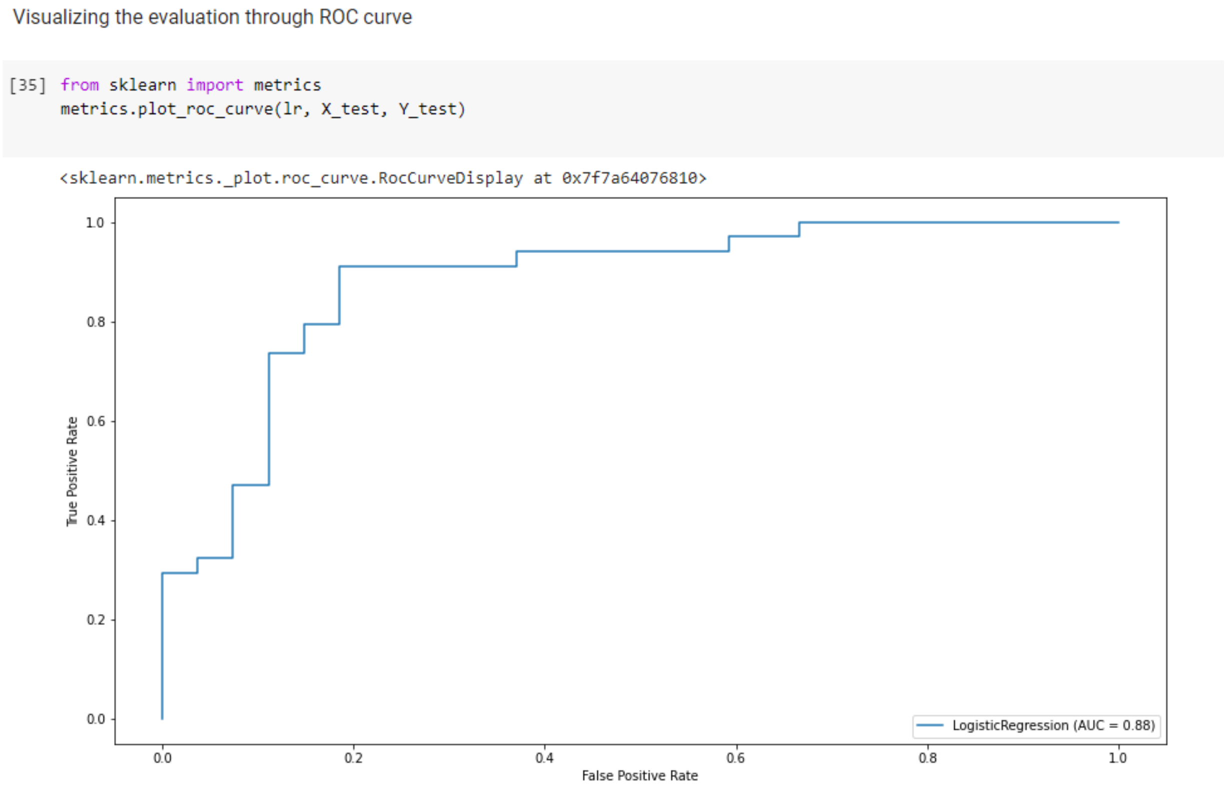
```
[34] from sklearn.metrics import classification_report
     print("Report : ", classification_report(Y_test, out))

     Report :               precision    recall  f1-score   support

                0       0.88      0.78      0.82        27
                1       0.84      0.91      0.87        34

         accuracy                           0.85        61
        macro avg       0.86      0.84      0.85        61
     weighted avg       0.85      0.85      0.85        61
```

We then visualized our evaluation through ROC curve:



Visualizing the evaluation through ROC curve

```
[35] from sklearn import metrics
     metrics.plot_roc_curve(lr, X_test, Y_test)
```

```
<sklearn.metrics._plot.roc_curve.RocCurveDisplay at 0x7f7a64076810>
```

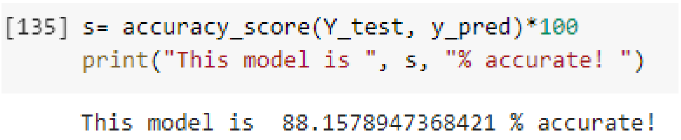
- For Random Forest:

We have used the Random Forest classifier as our third classifier to classify our dataset. It increases accuracy as it is an ensemble of several decision trees.

We split the data by using the test size of 0.25 meaning we used 75% of the data for training the model and 25% of the data for testing. We used Skit Learn library to import and implement Random Forest model. We fit the training data into the model and we then made the predictions by using the test data.

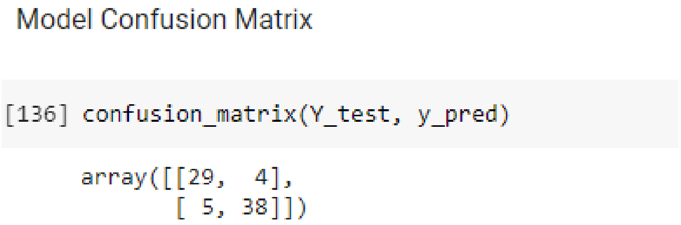
An Accuracy of **88.16%** is achieved.

We used several evaluation techniques to measure accuracy of the system such as Mean Squared Error which is 11.84%. We then used confusion matrix which is attached as follows:



```
[135] s= accuracy_score(Y_test, y_pred)*100
      print("This model is ", s, "% accurate! ")

      This model is  88.1578947368421 % accurate!
```

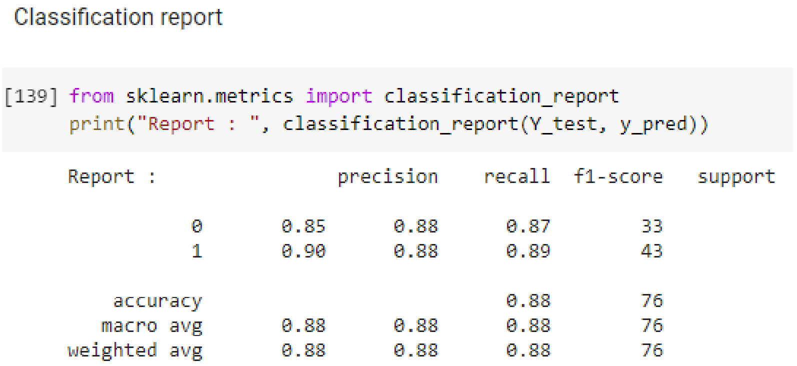


Model Confusion Matrix

```
[136] confusion_matrix(Y_test, y_pred)

      array([[29,  4],
             [ 5, 38]])
```

We used f1-score technique and got these as results:



Classification report

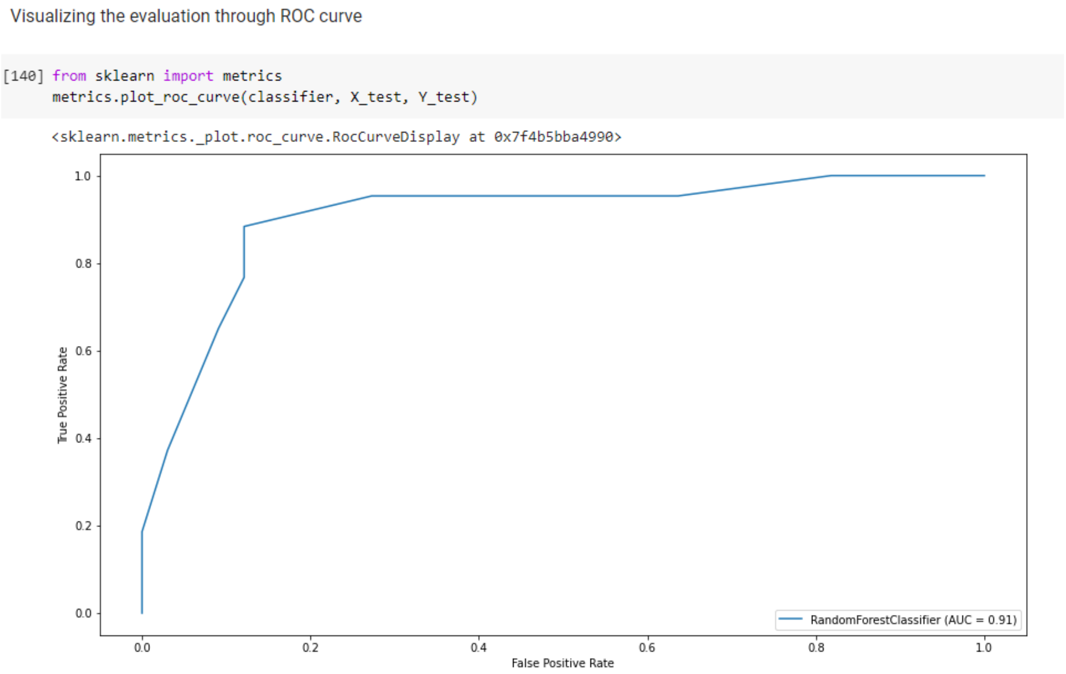
```
[139] from sklearn.metrics import classification_report
      print("Report : ", classification_report(Y_test, y_pred))

      Report :               precision    recall  f1-score   support

                 0       0.85      0.88      0.87        33
                 1       0.90      0.88      0.89        43

          accuracy                           0.88        76
         macro avg       0.88      0.88      0.88        76
      weighted avg       0.88      0.88      0.88        76
```

We then visualized our evaluation through ROC curve:

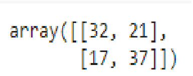
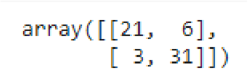
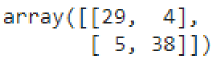
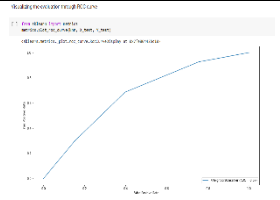
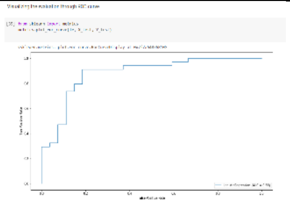
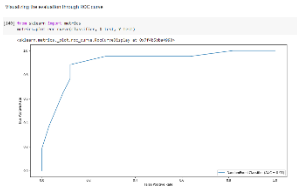


Visualizing the evaluation through ROC curve

```
[140] from sklearn import metrics
      metrics.plot_roc_curve(classifier, X_test, Y_test)

      <sklearn.metrics._plot.roc_curve.RocCurveDisplay at 0x7f4b5bba4990>
```

Comparison of the 3 Machine Learning Classifiers We Used:

Evaluation Techniques	K-NN	Logistic Regression	Random Forest
Accuracy	64.49%	85.25%	88.16%
Confusion matrix	 array([[32, 21], [17, 37]])	 array([[21, 6], [3, 31]])	 array([[29, 4], [5, 38]])
Mean-Squared error	35.5%	14.75%	11.84
F1 Score	Class: Score 1: 53 0: 54	Class: Score 1: 27 0: 34	Class: Score 1: 33 0: 43
ROC	 Visualizing the evaluation through ROC curve from sklearn import metrics metrics.plot_roc_curve(knn, X_test, Y_test)	 Visualizing the evaluation through ROC curve from sklearn import metrics metrics.plot_roc_curve(lr, X_test, Y_test)	 Visualizing the evaluation through ROC curve from sklearn import metrics metrics.plot_roc_curve(classifier, X_test, Y_test)

Evaluation and Conclusion

As we evaluated this article that we read, a lot of articles and all the articles have their own flaws and benefits of using dataset techniques. But here we concluded by reading all the articles that cardiovascular attacks being increased day by day and the deaths caused by this is major. we came up with the best program by keeping all the plus points of these studies and experiments, we se the data set of which they used and develop a more accurate and precise result telling of heart attack prediction model.

We discovered that the most stable machine learning model for this data is Logistic Regression as it gave a higher accuracy. Random Forest although gave the most accurate predictions but it is not stable as in every iteration it moves along the accuracy rates of 80-89% but it reaches to a value at last.

KNN is not suitable classifier for this data as the number of attributes is more for it to compute and KNN works well with a very small number of attributes. Critical analysis can be applied to another bioinformatics dataset and see the performance of these classifiers to classify or predict the diseases.

References

1. Almustafa, K. M. (2020, July 2). Prediction of heart disease and classifiers' sensitivity analysis. BMC Bioinformatics. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-020-03626-y

2. Effective heart disease prediction system using data mining techniques. (2018). PubMed Central (PMC) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5863635/

3. Gao, X. (2021, February 10). Improving the Accuracy for Analyzing Heart Diseases Prediction Based on the Ensemble Method. Www.Hindawi.Com. https://www.hindawi.com/journals/complexity/2021/6663455/#conclusion

4. Kumar, N. (2021, May 6). Efficient Automated Disease Diagnosis Using Machine Learning Models. Www.Hindaw.Com. https://www.hindawi.com/journals/jhe/2021/9983652/#abstract

5. Shah, D., Patel, S., & Bharti, S. K. (2020). Heart Disease Prediction using Machine Learning Techniques. SN Computer Science, 1(6). https://doi.org/10.1007/s42979-020-00365-y