Heart Disease Prediction

Machine Learning (CSE4020) - J component

Group Members	
Rajat Rathi	17BCE0900
Anuraj Srivastava	17BCE2006
Shashwat Sahai	17BCE2275

Video Link: https://drive.google.com/file/d/1jZ0RnZ5TEEwFq8U1d01QoY1JyzjQ8c15/view?usp=sharing

- **ABSTRACT:**

Machine Learning is used across many spheres around the world. The healthcare industry is no exception. Machine Learning can play an essential role in predicting presence/absence of locomotor disorders, Heart diseases and more. Such information, if predicted well in advance, can provide important insights to doctors who can then adapt their diagnosis and treatment per patient basis. The healthcare industry collects large amounts of Healthcare data, but unfortunately not all the data are mined is required for discovering hidden patterns and effective decision making. In this project, we will be predicting potential Heart Diseases in people using Machine Learning algorithms. The algorithms included are K Neighbours Classifier, Support Vector Classifier, Decision Tree Classifier and Random Forest Classifier. We will analyse prediction systems for Heart disease using a greater number of input attributes. The system uses medical terms such as Sex, Age, blood pressure, cholesterol like 13 attributes to predict the likelihood of patient getting a Heart disease.

- **MOTIVATION:**

Heart diseases are fatal and if not taken care of at the right time, they can be fatal. In India, heart diseases and strokes contribute to 12% to 15% of our annual death rate. A large majority of the fatal strokes are unforeseen and can strike to seemingly healthy individuals. Doctors have proved that even though the strokes and other heart diseases seem unprecedented to an individual, they can be prevented by following certain healthy dietary regimes which implies that there is a pattern or a correlation between the person's habits and the risk of stroke or other heart diseases. This has motivated us to study the healthcare data of heart patients and compare it with other healthy people. We developed a machine learning model which will take a variety of inputs and predict whether a person is susceptible to hear diseases or not. This way they can start taking precautions early on and avert the risk of having a stroke.

- **OBJECTIVE:**

The main objective of this project is to develop a machine learning model using various algorithms like K Neighbours Classifier, Support Vector Classifier, Decision Tree Classifier and Random Forest Classifiers and use a medical data set to predict the susceptibility of a person to heart diseases and strokes with as much accuracy as possible. Our aim is to analyse prediction systems for Heart disease using a greater number of input attributes. The system uses medical terms such as Sex, Age, blood pressure, cholesterol like 13 attributes to predict the likelihood of patient getting a Heart disease.

- **INTRODUCTION**

Machine learning is used across many spheres around the world. The healthcare industry is no exception. Machine Learning and AI can play an essential role in predicting presence/absence of locomotor disorders, Heart diseases and more.

Our project work is to create a system for predicting potential Heart Diseases in people using Machine Learning algorithms. The algorithms included are K Neighbours Classifier, Support Vector Classifier, Decision Tree Classifier, Random Forest Classifier and Neural Networks. The dataset has been taken

from Kaggle. Our objective is to analyse prediction systems for Heart disease using a greater number of input attributes. The system uses medical terms such as Sex, Age, blood pressure, cholesterol like 13 attributes to predict the likelihood of patient getting a Heart disease.

We will also compare the accuracy by which these algorithms can predict the heart disease

- **LITERATURE SURVEY**

[1] **Improved Study of Heart Disease Prediction System using Data Mining Classification Techniques.**

Published in International Journal of Computer Applications 2017.

Author: Chaitrali S. Dangare

This paper has analysed prediction systems for Heart disease using more number of input attributes. The system uses medical terms such as sex, blood pressure, cholesterol like 13 attributes to predict the likelihood of patient getting a Heart disease. Until now, 13 attributes are used for prediction. This research paper added two more attributes i.e. obesity and smoking. The data mining classification techniques, namely Decision Trees, Naive Bayes, and Neural Networks are analysed on Heart disease database. The performance of these techniques is compared, based on accuracy. As per our results accuracy of Neural Networks, Decision Trees, and Naive Bayes are 100%, 99.62%, and 90.74% respectively. Our analysis shows that out of these three classification models Neural Networks predicts Heart disease with highest accuracy.

[2] **Heart Disease Prediction using Data Mining with Mapreduce Algorithm**

Published in International Journal of Innovative Technology and Exploring Engineering (IJITEE) 2019

Authors: T.Nagamani, S.Logeswari, B.Gomathy

In the proposed system, large set of medical instances are taken as input. From this medical dataset, it is aimed to extract the needed information from the record of heart patients using MapReduce technique. The performance of the proposed MapReduce Algorithm's implementation in parallel and distributed systems was evaluated by using Cleveland dataset and compared with that of the predictable ANN method. The trial results verify that the projected method could achieve an average prediction accuracy of 98%, which is greater than the conventional recurrent fuzzy neural network. In addition, this MapReduce technique also had better performance than previous methods that reported prediction accuracies in the range of 95– 98%. These findings suggest that the MapReduce technique could be used to accurately predict HD risks in the clinic.

[3] HEART DISEASE PREDICTION USING DATA MINING TECHNIQUES

Published in ICTACT Journal on Soft Computing 2018 Authors: H. Benjamin Fredrick David and S. Antony Belcy

In this work, three data mining classification algorithms like Random Forest, Decision Tree and Naïve Bayes were addressed and used to develop a prediction system in order to analyse and predict the possibility of heart disease. The main objective of this significant research work was to identify the best classification algorithm suitable for providing maximum accuracy when classification of normal and abnormal person is carried out. Thus prevention of the loss of lives at an earlier stage is possible. It was found that Random Forest algorithm performs best with 81% precision when compared to other algorithms for heart disease prediction

[4] Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques

Publisher: IEEE

Authors: Senthilkumar Mohan ; Chandrasegar Thirumalai ; Gautam Srivastava

In this work, the authors introduce a technique called the Hybrid Random Forest with Linear Model (HRFLM). The main objective of this research is to improve the performance accuracy of heart disease prediction. Many studies have been conducted that results in restrictions of feature selection for algorithmic use. In contrast, the HRFLM method uses all features without any restrictions of feature selection. Here they conduct experiments used to identify the features of a machine learning algorithm with a hybrid method. The experiment results show that their proposed hybrid method has stronger capability to predict heart disease compared to existing methods.

[5] Intelligent heart disease prediction system using data mining techniques

Publisher: IEEE

Authors: Sellappan Palaniappan Rafiah Awang

The healthcare industry collects huge amounts of healthcare data which, unfortunately, are not "; mined"; to discover hidden information for effective decision making. Discovery of hidden patterns and relationships often goes unexploited. Advanced data mining techniques can help remedy this situation. This research has developed a prototype Intelligent Heart Disease Prediction System (IHDPS) using data mining techniques, namely, Decision Trees, Naive Bayes and Neural Network. Results show that each technique has its unique strength in realizing the objectives of the defined mining goals. IHDPS can answer complex "; what if"; queries which traditional decision support systems cannot. Using medical profiles such as age, sex, blood pressure and blood sugar it can predict the likelihood of patients getting a heart disease. It enables significant knowledge, e.g. patterns, relationships between medical factors related to heart disease, to be established. IHDPS is Web-based, user-friendly, scalable, reliable and expandable. It is implemented on the .NET platform.

[6] Cartery Calcium Score and Risk Classification for Coronary Heart Disease Prediction

Authors: Tamar S. Polonsky, MD; Robyn L. McClelland, PhD; Neal W. Jorgensen, BS;

The coronary artery calcium score (CACS) has been shown to predict future coronary heart disease (CHD) events. However, the extent to which adding CACS to traditional CHD risk factors improves classification of risk is unclear. The objective was to determine whether adding CACS to a prediction model based on traditional risk factors improves classification of risk. We evaluated the extent to which adding CACS to a model based on traditional risk factors correctly reclassifies participants in the MESA cohort in terms of risk of future CHD events. We determined how the addition of CACS to a prediction model changes the overall distribution of estimated risk. In contrast to previous studies that reported statistical associations only, we sought to clarify the potential utility of CACS as a tool for risk stratification. We evaluated the extent to which adding CACS to a model based on traditional risk factors correctly reclassifies participants in the MESA cohort in terms of risk of future CHD events. We determined how the addition of CACS to a prediction model changes the overall distribution of estimated risk. In contrast to previous studies that reported statistical associations only, we sought to clarify the potential utility of CACS as a tool for risk stratification.

[7] Intelligent Heart Disease Prediction System using CANFIS and Genetic Algorithm

Authors: Latha Parthiban and R.Subramanian

Medical diagnosis is an important but complicated task that should be performed accurately and efficiently and its automation would be very useful. All doctors are unfortunately not equally skilled in every sub specialty and they are in many places a scarce resource. A system for automated medical diagnosis would enhance medical care and reduce costs. In this paper, a new approach based on coactive neuro-fuzzy inference system (CANFIS) was presented for prediction of heart disease. The proposed CANFIS model combined the neural network adaptive capabilities and the fuzzy logic qualitative approach which is then integrated with genetic algorithm to diagnose the presence of the disease. The performances of the CANFIS model were evaluated in terms of training performances and classification accuracies and the results showed that the proposed CANFIS model has great potential in predicting the heart disease.

[8] Prediction system for heart disease using naive bayes

Authors: Shadab Adam Pattekari and Asma Parveen

The main objective of this research is to develop an Intelligent System using data mining modeling technique, namely, Naive Bayes. It is implemented as web based application in this user answers the predefined questions. It retrieves hidden data from stored database and compares the user values with trained data set. It can answer complex queries for diagnosing heart disease and thus assist healthcare practitioners to make intelligent clinical decisions which traditional decision support systems cannot. By providing effective treatments, it also helps to reduce treatment costs.

The healthcare industry collects huge amounts of healthcare data which, unfortunately, are not "mined" to discover hidden information for effective decision making. Discovery of hidden patterns and relationships often goes unexploited. Advanced data mining techniques can help remedy this situation. This research has developed a prototype Heart Disease Prediction System (HDPS) using data mining techniques, namely, Decision Trees, Naïve Bayes and Neural Network. Results show that each technique has its unique strength in realizing the objectives of the defined mining goals. HDPS can answer complex what if queries which traditional decision support systems cannot. Using medical profiles such as age, sex, blood pressure and blood sugar it can predict the likelihood of patients getting a heart disease. It enables significant knowledge, e.g. patterns, relationships between medical factors related to heart disease, to be established. HDPS is Web-based, user-friendly, scalable, reliable and expandable.

[9] Prediction of Heart Disease using Classification Algorithms

Author: Hlaudi Daniel Masethe, Mosima Anna Masethe

Heart attack diseases remains the main cause of death worldwide, including South Africa and possible detection at an earlier stage will prevent the attacks. Medical practitioners generate data with a wealth of hidden information present, and it's not properly being used effectively for predictions. For this purpose, the research converts the unused data into a dataset for modelling using different data mining techniques. People die having experienced symptoms that were not taken into considerations. The features that increase the possibility of heart attacks are smoking, lack of physical exercises, high blood pressure, high cholesterol, unhealthy diet, harmful use of alcohol, and high sugar levels. Cardio Vascular Disease (CVD) incorporates coronary heart, cerebrovascular (Stroke), hypertensive heart, congenital heart, peripheral artery, rheumatic heart, inflammatory heart disease. Data mining is a knowledge discovery technique to analyze data and encapsulate it into useful information. The current research intends to predict the probability of getting heart disease given patient data set.

Due to change in life styles in developing countries, like South Africa, Cardio Vascular Disease (CVD) has become a leading cause of deaths. CVD is projected to be a single largest killer worldwide accounting for all deaths. An endeavor to exploit knowledge, experience and clinical screening of patients to diagnose or recognize heart attacks is regarded as a treasured opportunity. In the health sectors data mining play an important role to predict diseases. The predictive end of the research is a data mining model.

[10] HDPS: Heart disease prediction system

A H Chen ; S Y Huang ; P S Hong ; C H Cheng ; E J Lin

The diagnosis of heart disease in most cases depends on a complex combination of clinical and pathological data. Because of this complexity, there exists a significant amount of interest among clinical professionals and researchers regarding the efficient and accurate prediction of heart disease. In this paper, we develop a heart disease predict system that can assist medical professionals in predicting heart disease status based on the clinical data of patients. Our approaches include three steps.

Firstly, we select 13 important clinical features, i.e., age, sex, chest pain type, trestbps, cholesterol, fasting blood sugar, resting ecg, max heart rate, exercise induced angina, old peak, slope, number of vessels colored, and thal. Secondly, we develop an artificial neural network algorithm for classifying heart disease based on these clinical features. The accuracy of prediction is near 80%. Finally, we develop a user-friendly heart disease predict system

(HDPS). The HDPS system will be consisted of multiple features, including input clinical data section, ROC curve display section, and prediction performance display section (execute time, accuracy, sensitivity, specificity, and predict result). Our approaches are effective in predicting the heart disease of a patient. The HDPS system developed in this study is a novel approach that can be used in the classification of heart disease.

- **METHODOLOGY**

1. DATASET and EXTERN LIBRARIES:

We have used the 'sklearn' machine learning library along with 'Keras' with 'TensorFlow' as backend for all our computations.

Data source

The source of the data can have a variety of different sources. It can be local data and can also be based on the Internet data. There are a variety of databases like it can be a traditional relational database, in the form of text data, multimedia database, oriented object database and web database etc.

2. IMPORTING THE DATASET AND UNDERSTANDING IT:

After downloading the dataset from Kaggle, we saved it to our working directory with the name dataset.csv. Next, we used read_csv() to read the dataset and save it to the dataset variable.

Before any analysis, we just wanted to take a look at the data. So, we used the info () method.

Observation: There are a total of 13 features and 1 target variable. Also, there are no missing values so we don't need to take care of any null values.

3. DATA PRE-PROCESSING:

Data pre-processing for data mining algorithms provide complete and clean, more accurate data which reduces the computation and improves the data mining efficiency and accuracy. Data cleaning through the data cleaning process to fill in the missing value to identify outliers removes the original data noise and irrelevant data.

Database. Almost the entire database supports the concept of the collection which equivalent to the table in the relational database.

4. USE OF A CORRELATION MATRIX TO UNDERSTAND THE DATA:

It's easy to see if there is any single feature that has a very high correlation with our target value. Also, some of the features might have a negative correlation with the target value and some have positive.

Bar plot for target class

It's really essential that the dataset we are working on should be approximately balanced. An extremely imbalanced dataset can render the whole model training useless and thus, will be of no use. Let's understand it with an example.

Let's say we have a dataset of 100 people with 99 non-patients and 1 patient. Without even training and learning anything, the model can always say that any new person would be a non-patient and have an accuracy of 99%. However, as we are more interested in identifying the 1 person who is a patient, we need balanced datasets so that our model actually learns.

- **ALGORITHMS:**

The algorithms included are K Neighbours Classifier, Support Vector Classifier, Decision Tree Classifier, Random Forest Classifier and Neural Networks.

1) K-Nearest Neighbours

K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection. In this the model structure determined from the dataset. This will be very helpful in practice where most of the real-world datasets do not follow mathematical theoretical assumptions

This classifier looks for the classes of K nearest neighbours of a given data point and based on the majority class, it assigns a class to this data point. However, the number of neighbours can be varied. We varied them from 1 to 25 neighbours and calculated the test score in each case.

2) Support Vector Machine (SVM)

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well

This classifier aims at forming a hyperplane that can separate the classes as much as possible by adjusting the distance between the data points and the hyperplane. There are several kernels based on which the hyperplane is decided.

We tried 4 kernels based on which the hyperplane is decided.

- **Linear**
- **Rbf**
- **Sigmoid**
- **Poly**

3) Decision Tree Classifier

Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

This classifier creates a decision tree based on which, it assigns the class values to each data point. Here, we can vary the maximum number of features to be considered while creating the model.

Decision Tree Algorithm Pseudocode

1. Place the best attribute of the data set at the root of the tree.

2. Split the training set into subsets. Subsets should be made in such a way that each subset contains data with the same value for an attribute.

3. Repeat step 1 and step 2 on each subset until you find leaf nodes in all the branches of the tree.

Information Gain

By using information gain as a criterion, we try to estimate the information contained by each attribute. We are going to use some points deducted from information theory.

To measure the randomness or uncertainty of a random variable X is defined by Entropy.

For a binary classification problem with only two classes, positive and negative class.

• If all examples are positive or all are negative then entropy will be zero i.e, low.

• If half of the records are of positive class and half are of negative class then entropy is one i.e, high.

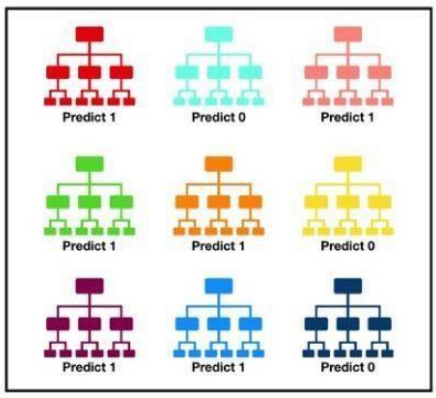
We vary the maximum number of features to be considered while creating the model. We range features from 1 to 13.

Plot a line graph and see the effect of the number of features on the model scores.

4) Random Forest Algorithm

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting.

It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction

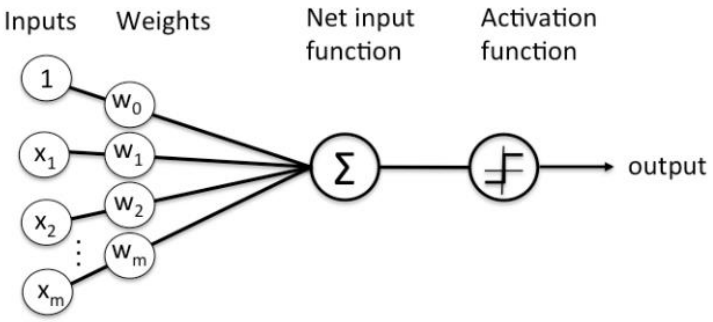


Visualization of a Random Forest Model Making a Prediction

5) Neural Net Classifier:

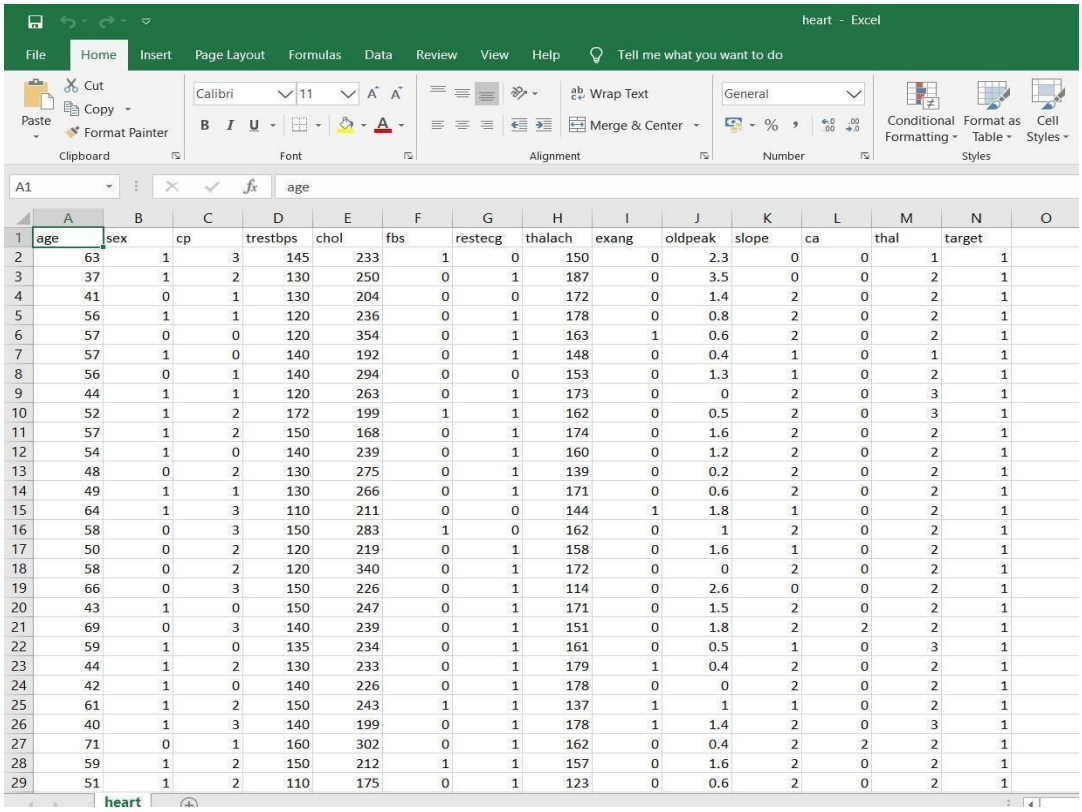
Neural nets take inspiration from the learning process occurring in human brains. They consists of an artificial network of functions, called parameters, which allows the computer to learn, and to fine tune itself, by analysing new data. Each parameter, sometimes also referred to as neurons, is a function which produces an output, after receiving one or multiple inputs. Those outputs are then passed to the next layer of neurons, which use them as inputs of their own function, and produce further outputs.

Those outputs are then passed on to the next layer of neurons, and so it continues until every layer of neurons have been considered, and the terminal neurons have received their input. Those terminal neurons then output the final result for the model.



- **Experimentations and Results**

1. **DATASET:** Dataset.csv. this is the csv file:



heart - Excel

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

Cut Copy Format Painter Paste Clipboard | Calibri 11 Font | Wrap Text Merge & Center Alignment | General Number | Conditional Formatting Format as Table Cell Styles Styles

A1 age

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	
2	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1	
3	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1	
4	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1	
5	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1	
6	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1	
7	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1	
8	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1	
9	44	1	1	120	263	0	1	173	0	0	2	0	3	1	
10	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1	
11	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1	
12	54	1	0	140	239	0	1	160	0	1.2	2	0	2	1	
13	48	0	2	130	275	0	1	139	0	0.2	2	0	2	1	
14	49	1	1	130	266	0	1	171	0	0.6	2	0	2	1	
15	64	1	3	110	211	0	0	144	1	1.8	1	0	2	1	
16	58	0	3	150	283	1	0	162	0	1	2	0	2	1	
17	50	0	2	120	219	0	1	158	0	1.6	1	0	2	1	
18	58	0	2	120	340	0	1	172	0	0	2	0	2	1	
19	66	0	3	150	226	0	1	114	0	2.6	0	0	2	1	
20	43	1	0	150	247	0	1	171	0	1.5	2	0	2	1	
21	69	0	3	140	239	0	1	151	0	1.8	2	2	2	1	
22	59	1	0	135	234	0	1	161	0	0.5	1	0	3	1	
23	44	1	2	130	233	0	1	179	1	0.4	2	0	2	1	
24	42	1	0	140	226	0	1	178	0	0	2	0	2	1	
25	61	1	2	150	243	1	1	137	1	1	1	0	2	1	
26	40	1	3	140	199	0	1	178	1	1.4	2	0	3	1	
27	71	0	1	160	302	0	1	162	0	0.4	2	2	2	1	
28	59	1	2	150	212	1	1	157	0	1.6	2	0	2	1	
29	51	1	2	110	175	0	1	123	0	0.6	2	0	2	1	

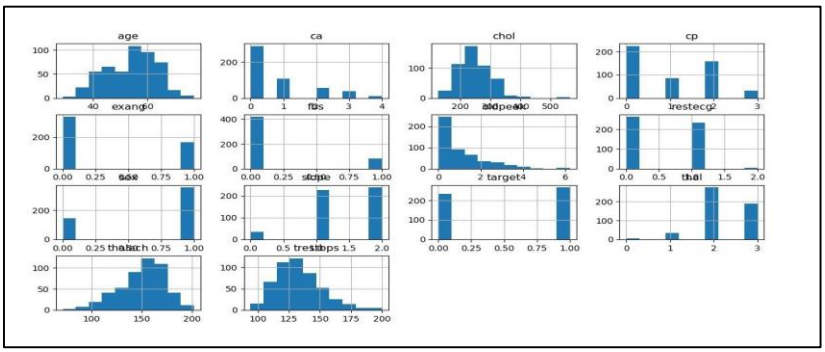
heart

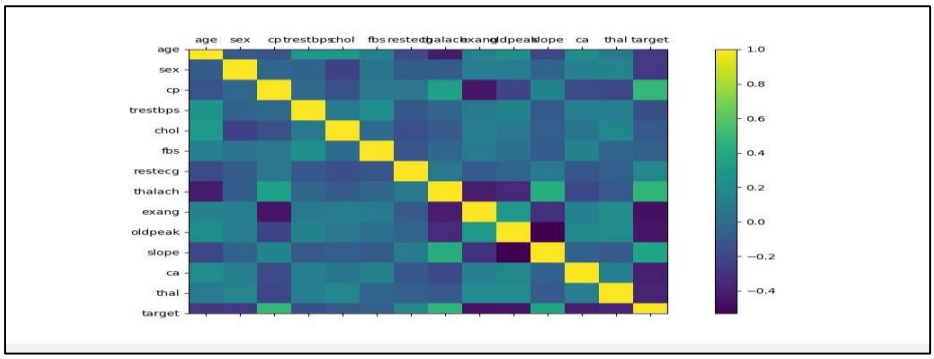
```
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
age         303 non-null int64
sex         303 non-null int64
cp          303 non-null int64
trestbps    303 non-null int64
chol        303 non-null int64
fbs         303 non-null int64
restecg     303 non-null int64
thalach     303 non-null int64
exang       303 non-null int64
oldpeak     303 non-null float64
slope       303 non-null int64
ca          303 non-null int64
thal        303 non-null int64
target      303 non-null int64
dtypes: float64(1), int64(13)
memory usage: 33.2 KB

>>>
```

2. UNDERSTANDING THE DATA

To begin with, let's see the correlation matrix of features and try to analyse it. The figure size is defined to 12 x 8 by using rcParams. Then, we used pyplot to show the correlation matrix. Using xticks and yticks,





3. **Conclusions from plot:**

It shows how each feature and label is distributed along different ranges, which further confirms the need for scaling. Next, wherever you see discrete bars, it basically means that each of these is actually a categorical variable. We will need to handle these categorical variables before applying Machine Learning. Our target labels have two classes, 0 for no disease and 1 for disease.

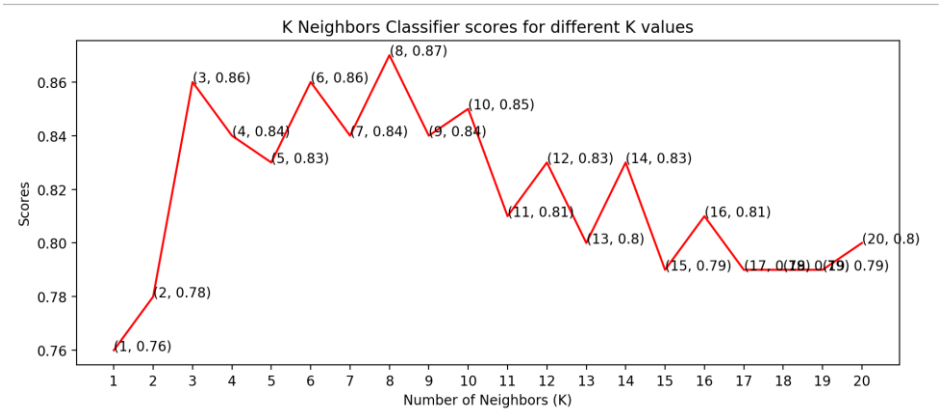
There is no single feature that has a very high correlation with our target value.

Some of the features have a negative correlation with the target value and some have positive

- **RESULTS:**

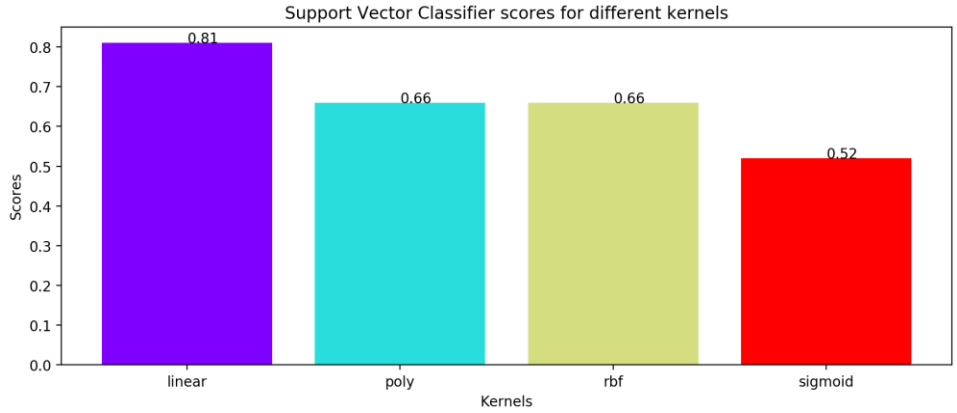
1) K-Nearest Neighbors:

Highest score: 87% at k =8



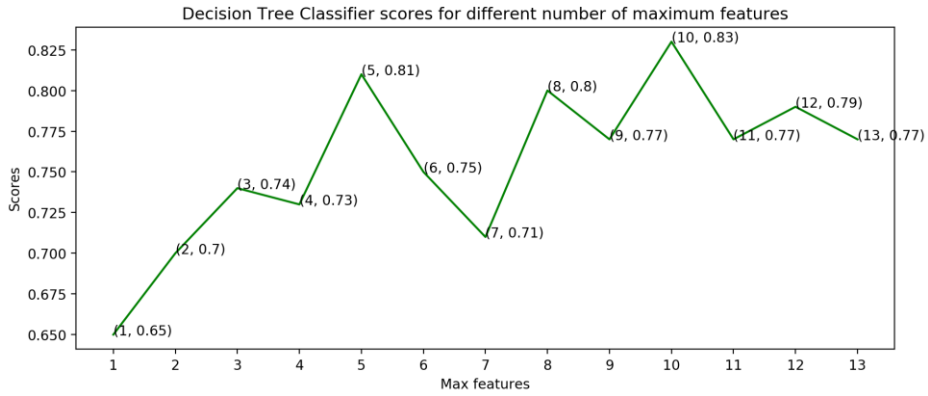
2) Support Vector Machine (SVM):

Highest Score: 81%
Kernel: Linear



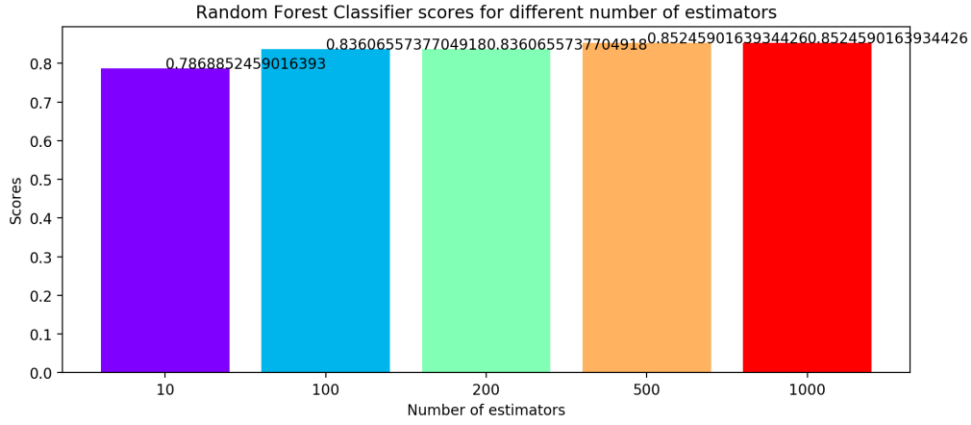
3) Decision Tree Classifier:

Highest Score: 83%
Features: 10



4) Random Forest Algorithm:

Highest Score: 85%
Minimum Estimators: 500



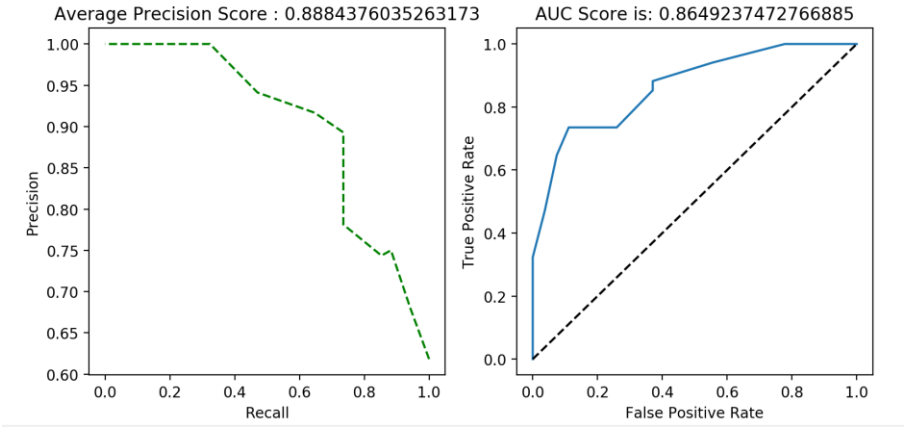
5) Neural Network Classifier:

Accuracy: 88.5%
Epochs: 50

```
Epoch 40/50
242/242 [==============================] - 0s 218us/step - loss: 0.2779 - accuracy: 0.8884
Epoch 41/50
242/242 [==============================] - 0s 217us/step - loss: 0.2768 - accuracy: 0.8884
Epoch 42/50
242/242 [==============================] - 0s 223us/step - loss: 0.2727 - accuracy: 0.8926
Epoch 43/50
242/242 [==============================] - 0s 222us/step - loss: 0.2706 - accuracy: 0.8967
Epoch 44/50
242/242 [==============================] - 0s 217us/step - loss: 0.2692 - accuracy: 0.8926
Epoch 45/50
242/242 [==============================] - 0s 216us/step - loss: 0.2682 - accuracy: 0.8967
Epoch 46/50
242/242 [==============================] - 0s 222us/step - loss: 0.2647 - accuracy: 0.9008
Epoch 47/50
242/242 [==============================] - 0s 218us/step - loss: 0.2623 - accuracy: 0.8926
Epoch 48/50
242/242 [==============================] - 0s 230us/step - loss: 0.2609 - accuracy: 0.8967
Epoch 49/50
242/242 [==============================] - 0s 222us/step - loss: 0.2594 - accuracy: 0.9050
Epoch 50/50
242/242 [==============================] - 0s 221us/step - loss: 0.2565 - accuracy: 0.9008

Neural Network Accuracy:  0.8852459016393442
```

6. Precision and AUC Curve:



- **FINAL OBSERVATION:**

Heart disease patient dataset with proper data processing.

Then, 4 models were trained and tested with maximum scores as follows:

S no.	Model used	Description	Performance
1	**K Neighbors Classifier**	**Best performance at K=8**	87%
2	**Support Vector Classifier**	**Best performance with linear kernel**	81%
3	**Decision Tree Classifier**	**At max features = 10 we get best performance**	83%
4	**Random Forest Tree Classifier**	**Best performance of 5% at estimator ount=80**	85%

5	**Neural net classifier**	**Best performance at 50 eepochs and 2 hidden layers**	88.5

- **CONCLUSIONS**

Based on the accuracy levels of all five algorithms used, namely, K-neighbours classifier, support vector classifier, decision tree classifier and random forest classifier and finally the neural net classifier, it is evident that the neural net classifier scored the best score and therefore has the highest accuracy.

This project, heart disease prediction has a great scope since neural net classifier can predict if a person will have heart disease or not with an accuracy of 88%.

Machine Learning is used across many spheres around the world. The healthcare industry is no exception. Machine Learning can play an essential role in predicting presence/absence of locomotor disorders, Heart diseases and more. In this project, we will be predicting potential Heart Diseases in people using Machine Learning algorithms. The algorithms included are K Neighbours Classifier, Support Vector Classifier, Decision Tree Classifier and Random Forest Classifier.

We can use this result, to build future models where we use Neural Networks for prediction of Heart Diseases for new patients on the basis of patient's health record and vitals. Further research can be made in this model to increase the accuracy and/or precision so that this model becomes more efficient.

- **REFERENCES**

[1] ***Improved Study Of Heart Disease Prediction System Using Data Mining Classification Techniques.***

Published In International Journal Of Computer Applications 2017.

Author: Chaitrali S. Dangare

[2] ***Heart Disease Prediction Using Data Mining With Mapreduce Algorithm***

Published In *International Journal Of Innovative Technology And Exploring Engineering (Ijitee) 2019*

Authors: T.Nagamani, S.Logeswari, B.Gomathy

[3] ***Heart Disease Prediction Using Data Mining Techniques***

Published In *Ictact Journal On Soft Computing 2018*

Authors: H. Benjamin Fredrick David And S. Antony Belcy

[4] ***Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques***

Publisher: Ieee

Authors: Senthilkumar Mohan ; Chandrasegar Thirumalai ; Gautam Srivastava

[5] Intelligent Heart Disease Prediction System Using Data Mining Techniques

Https://Ieeexplore.Ieee.Org/Abstract/Document/4493524

[6] Coronary Artery Calcium Score And Risk Classification For Coronary Heart Disease Prediction

Https://Jamanetwork.Com/Journals/Jama/Article-Abstract/185757

[7] Intelligent Heart Disease Prediction System Using Canfis And Genetic Algorithm

Http://Citeseerx.Ist.Psu.Edu/Viewdoc/Download?Doi=10.1.1.451.9421&Rep=Rep1&Type=Pdf

[*8] Prediction System For Heart Disease Using Naive Bayes*

Https://Pdfs.Semanticscholar.Org/D32e/E90a5de89093a4fc95f43e0409cb91414726.Pdf

[9] Prediction Of Heart Disease Using Classification Algorithms

Http://Www.Iaeng.Org/Publication/Wcecs2014/Wcecs2014_Pp809-812.Pdf

[10] HDPS: Heart disease prediction system

Publisher: IEEE

https://ieeexplore.ieee.org/abstract/document/6164626

- **APPENDIX – I**

1. **CODE:**

Pre-Processing (Correlation Matrix)
CODE:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
from matplotlib.cm import rainbow

import warnings
warnings.filterwarnings('ignore')

# Other libraries
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Machine Learning
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

```
dataset = pd.read_csv(r"D:/Heart Disease/heart.csv")
rcParams['figure.figsize'] = 8, 6
plt.matshow(dataset.corr())
plt.yticks(np.arange(dataset.shape[1]), dataset.columns)
plt.xticks(np.arange(dataset.shape[1]), dataset.columns)
plt.colorbar()
dataset.hist()

rcParams['figure.figsize'] = 8,6
plt.bar(dataset['target'].unique(), dataset['target'].value_counts(), color = ['red', 'green'])
plt.xticks([0, 1])
plt.xlabel('Target Classes')
plt.ylabel('Count')
plt.title('Count of each Target Class')
plt.show()
```

A. K-Nearest Neighbour Algorithm

Code:

```
import warnings

from matplotlib import pyplot as plt
import pandas as pd

warnings.filterwarnings('ignore')

# Other libraries
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Machine Learning
from sklearn.neighbors import KNeighborsClassifier

dataset = pd.read_csv(r"D:/Heart Disease/heart.csv")
dataset.info()
dataset.describe()
dataset = pd.get_dummies(dataset, columns = ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca',
'thal'])
standardScaler = StandardScaler()
columns_to_scale = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
dataset[columns_to_scale] = standardScaler.fit_transform(dataset[columns_to_scale])
y = dataset['target']
X = dataset.drop(['target'], axis = 1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 0)
knn_scores = []
for k in range(1,21):
        knn_classifier = KNeighborsClassifier(n_neighbors = k)
        knn_classifier.fit(X_train, y_train)
```

```
        knn_scores.append(knn_classifier.score(X_test, y_test))
plt.plot([k for k in range(1, 21)], knn_scores, color = 'red')
for i in range(1,21):
        plt.text(i, knn_scores[i-1], (i, knn_scores[i-1]))
plt.xticks([i for i in range(1, 21)])
plt.xlabel('Number of Neighbors (K)')
plt.ylabel('Scores')
plt.title('K Neighbors Classifier scores for different K values')
plt.show()
```

B. Support Vector Classifier (SVC):
CODE:

```
import warnings

import numpy as np

from matplotlib import pyplot as plt

import pandas as pd

warnings.filterwarnings('ignore')

from matplotlib.cm import rainbow


# Other libraries

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler


# Machine Learning

from sklearn.svm import SVC


dataset = pd.read_csv(r"D:/Heart Disease/heart.csv")

svc_scores = []

kernels = ['linear', 'poly', 'rbf', 'sigmoid']

y = dataset['target']

X = dataset.drop(['target'], axis = 1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 0)
```

```
for i in range(len(kernels)):
    svc_classifier = SVC(kernel = kernels[i])
    svc_classifier.fit(X_train, y_train)
    svc_scores.append(svc_classifier.score(X_test, y_test))
colors = rainbow(np.linspace(0, 1, len(kernels)))
plt.bar(kernels, svc_scores, color = colors)
for i in range(len(kernels)):
        plt.text(i, svc_scores[i], svc_scores[i])
plt.xlabel('Kernels')
plt.ylabel('Scores')
plt.title('Support Vector Classifier scores for different kernels')
plt.show()
```

C. Decision Tree Classifier

CODE:

```
import numpy as np
from matplotlib import pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
dataset = pd.read_csv("D:/Heart Disease/heart.csv")
y = dataset['target']
X = dataset.drop(['target'], axis = 1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 0)

dt_scores = []
for i in range(1, len(X.columns) + 1):
        dt_classifier = DecisionTreeClassifier(max_features = i, random_state = 0)
        dt_classifier.fit(X_train, y_train)
        dt_scores.append(dt_classifier.score(X_test, y_test))
plt.plot([i for i in range(1, len(X.columns) + 1)], dt_scores, color = 'green')
```

```
for i in range(1, len(X.columns) + 1):
    plt.text(i, dt_scores[i-1], (i, dt_scores[i-1]))
plt.xticks([i for i in range(1, len(X.columns) + 1)])
plt.xlabel('Max features')
plt.ylabel('Scores')
plt.title('Decision Tree Classifier scores for different number of maximum features')
plt.show()
```

D. Random Forest and Neural Networks
CODE:

```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
import keras
from keras.models import Sequential
from keras.layers import Dense

import warnings
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import average_precision_score
from sklearn.metrics import roc_curve
```

```
from sklearn.metrics import auc
from sklearn.model_selection import cross_val_score
from sklearn.metrics import f1_score

import matplotlib.cm as lm

df = pd.read_csv("D:/Heart Disease/heart.csv")
df.head(5)
chest_pain=pd.get_dummies(df['cp'],prefix='cp',drop_first=True)
df=pd.concat([df,chest_pain],axis=1)
df.drop(['cp'],axis=1,inplace=True)
sp=pd.get_dummies(df['slope'],prefix='slope')
th=pd.get_dummies(df['thal'],prefix='thal')
rest_ecg=pd.get_dummies(df['restecg'],prefix='restecg')
frames=[df,sp,th,rest_ecg]
df=pd.concat(frames,axis=1)
df.drop(['slope','thal','restecg'],axis=1,inplace=True)
X = df.drop(['target'], axis = 1)
y = df.target.values

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, random_state = 0)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
classifier = Sequential()

# input layer
classifier.add(Dense(output_dim = 11, init = 'uniform', activation = 'relu', input_dim = 22))
```

```
#hidden layer
classifier.add(Dense(output_dim = 11, init = 'uniform', activation = 'relu'))
classifier.add(Dense(output_dim = 11, init = 'uniform', activation = 'relu'))

# output layer
classifier.add(Dense(output_dim = 1, init = 'uniform', activation = 'sigmoid'))
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
classifier.fit(X_train, y_train, batch_size = 10, nb_epoch = 50)

# Predicting the Test set results
y_pred = classifier.predict(X_test)
# import seaborn as sns
# from sklearn.metrics import confusion_matrix
# cm = confusion_matrix(y_test, y_pred.round())

# sns.heatmap(cm,annot=True,cmap="Blues",fmt="d",cbar=False)

#accuracy score
from sklearn.metrics import accuracy_score
ac=accuracy_score(y_test, y_pred.round())
print()
print('Neural Network Accuracy: ',ac)
print()


rdf_c=RandomForestClassifier(n_estimators=10,criterion='entropy',random_state=0)
rdf_c.fit(X_train,y_train)
rdf_pred=rdf_c.predict(X_test)
rdf_cm=confusion_matrix(y_test,rdf_pred)
rdf_ac=accuracy_score(rdf_pred,y_test)
```

```
# plt.title("rdf_cm")
# sns.heatmap(rdf_cm,annot=True,fmt="d",cbar=False)
print('Random Forest Accuracy:',rdf_ac)
rf_scores = []
estimators = [10, 100, 200, 500, 1000]
for i in estimators:
        rf_classifier = RandomForestClassifier(n_estimators = i, random_state = 0)
        rf_classifier.fit(X_train, y_train)
        rf_scores.append(rf_classifier.score(X_test, y_test))
estimators = [10, 100, 200, 500, 1000]
colors = lm.rainbow(np.linspace(0, 1, len(estimators)))
plt.bar([i for i in range(len(estimators))], rf_scores, color = colors, width = 0.8)

for i in range(len(estimators)):
        plt.text(i, rf_scores[i], rf_scores[i])
plt.xticks(ticks = [i for i in range(len(estimators))], labels = [str(estimator) for estimator in
estimators])
plt.xlabel('Number of estimators')
plt.ylabel('Scores')
plt.title('Random Forest Classifier scores for different number of estimators')
plt.show()

def plotting(true,pred):
        fig,ax=plt.subplots(1,2,figsize=(10,5))
        precision,recall,threshold = precision_recall_curve(true,pred[:,1])
        ax[0].plot(recall,precision,'g--')
        ax[0].set_xlabel('Recall')
        ax[0].set_ylabel('Precision')
        ax[0].set_title("Average Precision Score :
{}".format(average_precision_score(true,pred[:,1])))
        fpr,tpr,threshold = roc_curve(true,pred[:,1])
        ax[1].plot(fpr,tpr)
```

```
        ax[1].set_title("AUC Score is: {}".format(auc(fpr,tpr)))
        ax[1].plot([0,1],[0,1],'k--')
        ax[1].set_xlabel('False Positive Rate')
        ax[1].set_ylabel('True Positive Rate')

plotting(y_test,rdf_c.predict_proba(X_test))
#plt.figure()
plt.show()

model_accuracy = pd.Series(data=[rdf_ac,ac], index=['RandomForest','Neural Network'])
fig= plt.figure(figsize=(8,8))
model_accuracy.sort_values().plot.barh()
plt.title('Model Accracy')
plt.show()
```