A SUMMER TRAINING PROJECT REPORT

On

Heart issues among various age groups

Submitted in partial fulfillment of requirements for the award of the

Degree of

Bachelor of Technology

In

Electronics and Communication Engineering

Submitted by

ABHINAV

00211502820

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BHARTI VIDYAPEETH COLLEGE OF ENGINEERNG

A-4 PASCHIM VIHAR, ROHTAK ROAD, NEW DELHI

NOVEMBER, 2022*

CONTENTS

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this report entitles " Heart issues among various age groups", in fulfilment of the requirement for the award of the degree Bachelor of Technology in Electronics and Communication Engineering, submitted in ECE Department, BVCOE affiliated to Guru Gobind Singh Indraprastha University, New Delhi, is an authentic record of my own work carried out during my degree.

The work reported in this has not been submitted by me for award of any other degree or B.Tech Graduation.

Date: ABHINAV
Place: (00211502820)

CERTIFICATE

This is to certify that the Project work entitled "Heart ages among various age groups" submitted by in fulfilment for the requirements of the in House Summer Training of Bachelor of Technology Degree in Electronics and Communication Engineering at BVCOE, New Delhi is an authentic work carried out by his/her under my supervision and guidance. To the best of my knowledge, the matter embodied in the project has not been submitted to any other University/Institute for Project .

Mrs Shipra Singh

(Training Coordinator)

ACKNOWLEDGEMENT

I express my deep gratitude to **Ms. Shipra Singh**, Assistant Manager, **Sofcon India Pvt.Ltd** , for her valuable guidance and suggestions throughout my training.

Sign (ABHINAV)

Enrolment No:**00211502820**

Sofcon India Pvt. Ltd.

Sofcon India Pvt. Ltd. (An ISO 9001:2015 Certified) is knowledge based multi-disciplinary training company professionally run by technocrats having three decades of rich experience in providing turnkey solutions for applications like Power Plants ,Cement Plants, Oil & Gas Plants, Petrochemical Plants, pharmaceutical Plants, Refineries, Food Processing Plants, Water Treatment Plants, Process Plants, Fertilizer,DG Automation, Energy Monitoring, Load Management, Material Handling, Automation System, Automobile, Ash Handling, Coal Plants and many more.

Sofcon India Pvt Ltd is NSDC (National Skill Development Corporation), MSDE (Ministry of Skill Development & Entrepreneurship-Govt. of India) Affiliated & Funded Company. We are also affiliated with ESSCI (Electronic Sector Skill Council), IASC (Instrumentation Automation Surveillance & Communication Sector Skill Council) & HSSC (Healthcare Sector Skill Council).

Their vision is to be the Brand bridging gap between Industry & Academia, To be a workplace where everyone is inspired to be the best they can be, To inspire innovation, learning, creativity, To be a responsible, effective, dynamic and fast-moving organization.

WEEKLY REPORT

1.1 Course Content

The entire six-week duration of the internship was divided into three phases: Phase I & II. Phase I consisted of python while Phase II consistedof machine learning and phase III consisted of AI.

Week	Content
Week 1	Python basics
Week 2	Python libraries- Numpy, Pandas, Matplotlib
Week 3	Got introduced to Machine Learning
Week 4	Types of Learning and Algorithms in ML
Week 5& 6	Introduction to Artificial Intelligence and made project on Heart Issues among various age groups.

TABLE 1.1 WEEKLY CONTENT

Abstract

This data set contains information of heart problems seen among people of different ages and gender. All personally identifying information has been removed from the data. The dataset describes heath issues and contains 14 columns and 304 rows and contains data of-

1. Age: age of person
2. Sex: gender of person (1 = male; 0 = female)
3. Cp: chest pain type (1 = typical angina; 2 = atypical angina; 3 = non-anginal pain; 4 = asymptomatic).
4. trestbps - resting blood pressure (in mm Hg on admission to the hospital)
5. chol - serum cholestoral in mg/dl
6. fbs - fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
7. restecg - resting electrocardiographic results (0 = normal; 1 = having ST-T; 2 = hypertrophy)
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 (1 = upsloping; 2 = flat; 3 = downsloping)
12. ca - number of major vessels (0-3) colored by flourosopy
13. thal - 3 = normal; 6 = fixed defect; 7 = reversable defect
14. target - the predicted attribute - diagnosis of heart disease (angiographic disease status) (Value 0 = < 50% diameter narrowing; Value 1 = > 50% diameter narrowing)

INTRODUCTION

Heart problems are a major cause of death across the world. Here in this project we would be seeing the age groups who are more likely to be under the risk of these issues and we would also be predicting the possibility of heart issues in a population by making use of ML algorithms and modules.

PROBLEM STATEMENT

This dataset contains information regarding the health issues seen across various age groups and the various heart problems in the population, such as chest pain, resting blood pressure, serum cholesterol levels, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved etc.

We will tackle the problem statement by the following steps

Step1: Data Overview

Step2: Data visualization through color maps and histogram

Step3: Data Processing

Step4: Implementation using ML modules

Step5: Concluding Analysis

Prerequisites for this project

Before beginning with this project we shall see some of the prerequisites that are required to better understand it.

Python packages

This ML project is based on a python program and requires the use of python packages that can be imported in our program. These packages are precoded libraries which has various functionalities, which can be imported as required by our program.

The python packages we would be dealing with include-

1. Numpy- It is a python library used for working with arrays. It also has functions for working with linear algebra, fourier transform and matrices.

2. Pandas- Pandas is an open source library in Python. It provides ready to use high-performance data structures and data analysis tools. Pandas module runs on top of NumPy and it is popularly used for data science and data analytics.

3. Matplotlib -Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter.

4. Sklearn- It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

WHAT is ML and why do we require it

Machine learning (ML) is a field of inquiry devoted to understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks.

The term *machine learning* was coined in 1959 by ARTHUR SAMUEL, an IBM employee and pioneer in the field of COMPUTER GAMING AND A.I.

TYPES

Supervised and Unsupervised learning are the two techniques of machine learning

SUPERVISED LEARNING-

Supervised learning is a machine learning method in which models are trained using labeled data. In supervised learning, models need to find the mapping function to map the input variable (X) with the output variable (Y).

UNSUPERVISED LEARNING-

Unsupervised learning is another machine learning method in which patterns inferred from the unlabeled input data. The goal of unsupervised learning is to find the structure and patterns from the input data. Unsupervised learning does not need any supervision. Instead, it finds patterns from the data by its own.

REGRESSION

It is a subtype of supervised learning, which is based on the process of predicting continuous values. It has two variables dependent variable (y) and independent variable(x). Eg. Historical data for co2 emission, consumption of fuel etc.

It's algorithms are

1. Ordinal regression
2. Poisson regression
3. Fast forest quantile regression
4. Linear, polynomial, lasso, stepwise, ridge regression
5. Bayesian linear regression
6. Neural network regression
7. Decision forest regression
8. Boosted decision tree regression
9. KNN(K-NEAREST NEIGHBOURs)

Linear regression

IN this a line can be fitted through data

Fit line->

$\hat{Y}= \theta 0+ \theta 1x1$, on the LHs is the predicted value, $\theta 0$ is the slope/gradient, $\theta 1$ is the parameter and x1 is the independent value. Linear regression target is to minimize the error using the best parameters. Some of it's advantages include-

1.Very fast

2. No parameter tuning

3.Easy to understand, and highly interpretable

Non-linear regression

It is used to model non-linear relationship between the dependent variable and a set of independent variables.

$\hat{Y}$ must be a non-linear function of the parameters θ, not necessarily the features of X, eg. $\hat{Y}= \theta_0+ \theta_2{}^2X$.

For modelling the data in non-linear regression

- Polynomial regression
- Non-linear regression model
- Transform your data

CLASSIFICATION

It is a type of supervised learning, used for categorizing some unknown items into a discrete set of categories or classes. The target attribute is a categorical variable with discrete values. It determines the class label for an unlabelled test case.

Classification algorithms

- Decision Trees
- Naïve bayes
- Linear Discriminant Analysis
- K-nearest Neighbour
- Logistic Regression
- Neural Networks
- Support Vector Machines(SVM)

KNN classifier

It is a method for classifying cases based on their similarity to other cases. Cases that are near to each other are said to be neighbours. Based on similar cases with same class labels are near each other. It's working is as follows

1. Pick a value for k
2. Calculate the distance of unknown case from all cases
3. Select the k-observations in the training data that are nearest to the unknown data point
4. Predict the response of the unknown data point using the most popular response value from the k-nearest neighbours. The k number of observations shouldn't be too small and can't be too large.

Support vector machine(SVM)

It is a supervised algorithm that classifies cases by finding a separator.

1. Mapping data to high dimensional feature space.
2. Finding a separator

Some of it's advantage include, higher accuracy in high dimensional spaces and it is memory efficient. The disadvantage being it is prone to overfitting, no probability estimation, and small datasets.

SVM applications

1. Image recognition
2. Text category assignment
3. Detecting spam
4. Sentiment analysis
5. Gene Expression Classification
6. Regression, outlier detection and clustering

Decision-Tree Classifier

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset.

It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

Random forest classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model*.

As the name suggests, "*Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.*" Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Now moving on with the project,

HEART ISSUES AMONG VARIOUS AGE GROUPS

First beginning with importing all the above mentioned libraries in our programmes

I/P

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

For splitting the dataset and testing and training and simultaneously including the required ML algorithms

I/P

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

Our dataset will be based on a CSV file. CSV stand for Comma Separated Value. A comma-separated values file is a delimited text file that uses a comma to separate values. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. The following data will be used for training and testing

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
58	0	3	150	283	1	0	162	0	1	2	0	2	1
50	0	2	120	219	0	1	158	0	1.6	1	0	2	1
58	0	2	120	340	0	1	172	0	0	2	0	2	1
66	0	3	150	226	0	1	114	0	2.6	0	0	2	1

This csv is stored as dataset.csv in the system, so we will make the program read by using,

I/P

```
dataset = pd.read_csv('dataset.csv')
```

and the data can be viewed by->

I/P

dataset.info()

O/P

```
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

dataset.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

The scale of each feature column is different and quite varied as well. While the maximum for age reaches 77, the maximum of chol (serum cholestoral) is 564.

VISUALIZATION OF DATA

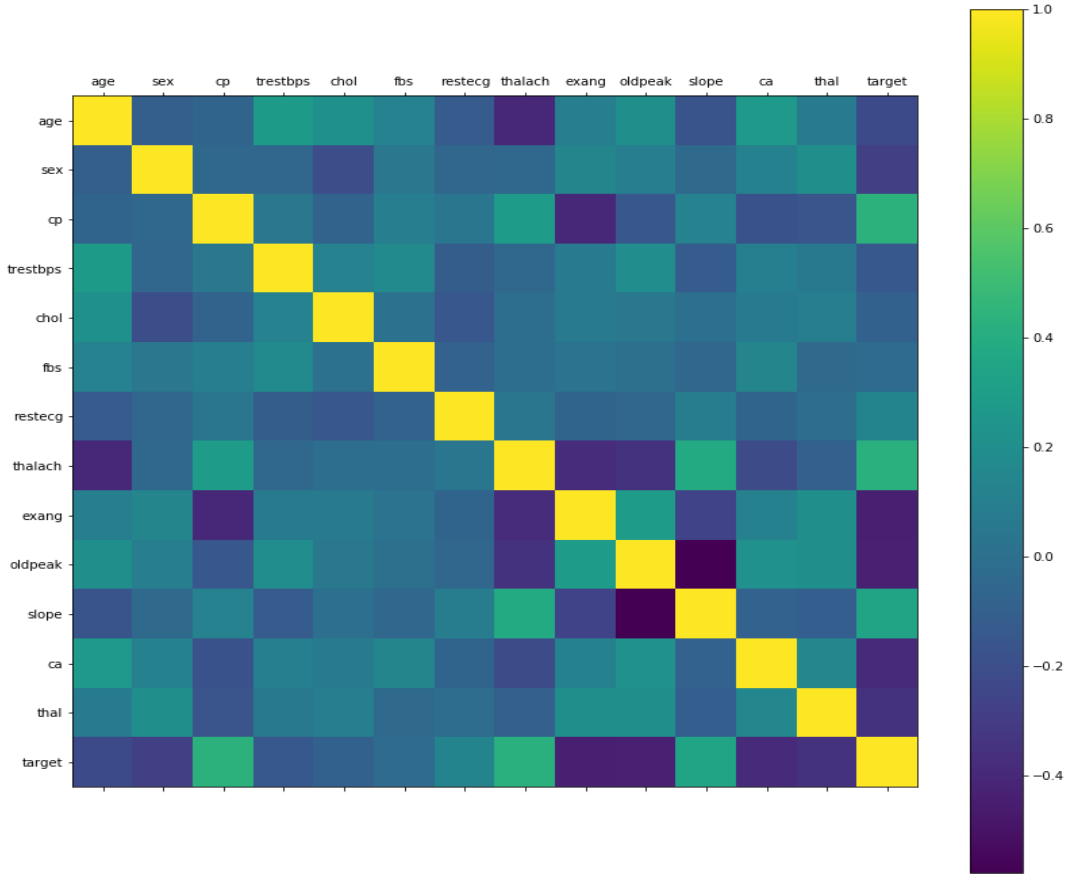
Heat mapping

Now the for visualization we will make use of the rcParams that we imported from the matplotlib module.

I/P

```
rcParams['figure.figsize'] = 20, 14
plt.matshow(dataset.corr())
plt.yticks(np.arange(dataset.shape[1]), dataset.columns)
plt.xticks(np.arange(dataset.shape[1]), dataset.columns)
plt.colorbar()
```

O/P



Taking a look at the correlation matrix above, it's easy to see that a few features have negative correlation with the target value while some have positive.

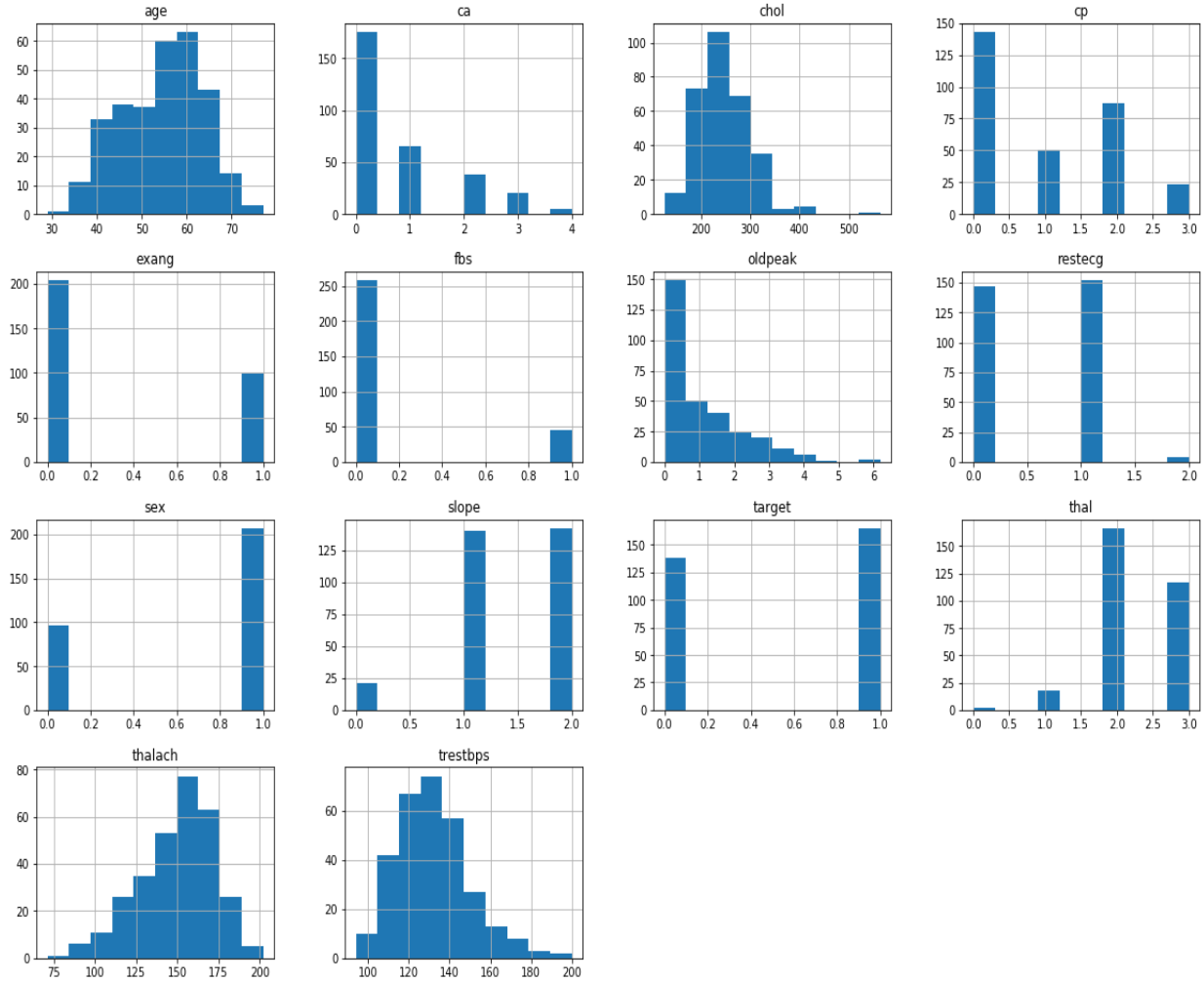
Now we take a look at the histograms for each variable.

I/p

```
dataset.hist()
```

o/p

```
array([[,
        ,
        ,
        ],
       [,
        ,
        ,
        ],
       [,
        ,
        ,
        ],
       [,
        ,
        ,
        ]],
      dtype=object)
```



Some of the interesting trends can be seen from here. We can clearly see how the majority of people who are having heart issues are in the age group of 50-70 which include majority of middle aged and

old aged people. And with ‘1’ denoting the male and ‘0’ denoting the female population, judging by the above visualization we can also see how males are under a higher risk of having heart issues than females.

We can notice that each feature has a different range of distribution. Thus, using scaling before our predictions should be of great use. Also, the categorical features do stand out.

It's always a good practice to work with a dataset where the target classes are of approximately equal size. Thus, let's check for the same.

I/p

```
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')
```

O/p

```
Text(0.5, 1.0, 'Count of each Target Class')
```

Data Processing

After exploring the dataset, we observed that we need to convert some categorical variables into dummy variables and scale all the values before training the Machine Learning models.

I/p

```
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])
```

Now that we have scaled our data, we can split our dataset into training and testing datasets, so that we can import various ML modules to train and test the data.

I/P

```
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)
```

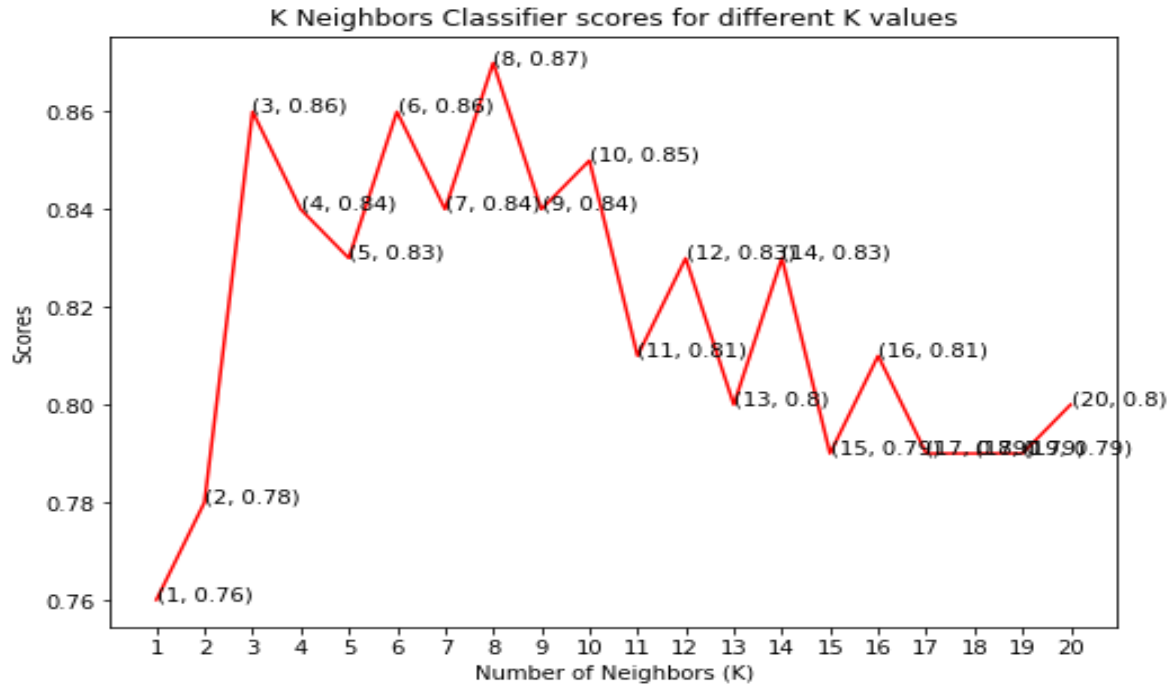
Now Using the K neighbour classifier module and plotting it simultaneously

I/P

```
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')
```

O/P

```
Text(0.5, 1.0, 'K Neighbors Classifier scores for different K values')
```



From the plot above, it is clear that the maximum score achieved was 0.87 for the 8 neighbours.

I/p

```
print("The score for K Neighbors Classifier is {}% with {}
nieghbors.".format(knn_scores[7]*100, 8))
```

So, now the score for K neighbour classifier is 87.0% with 8 neighbours

Support Vector Classifier

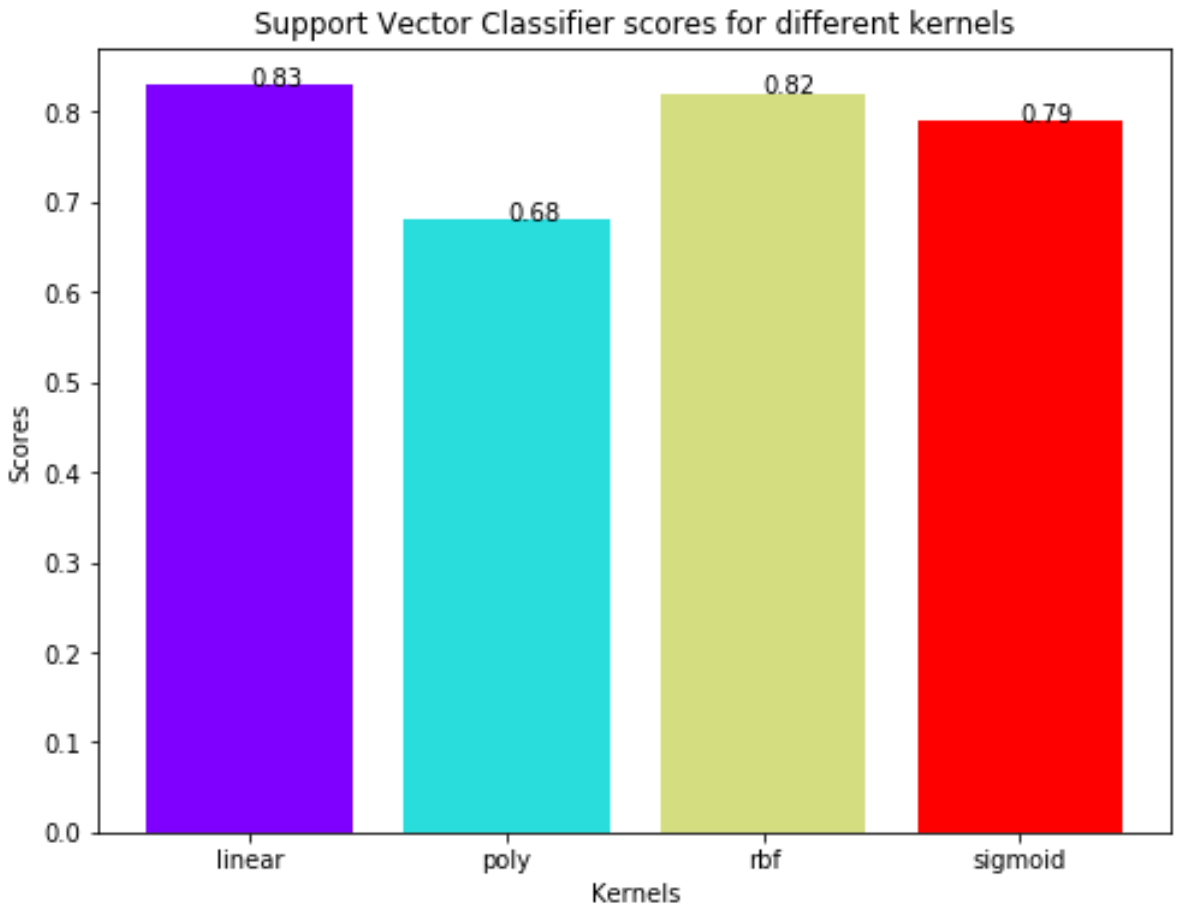
Using the support vector classifier and subsequent plotting

I/P

```
svc_scores = []
kernels = ['linear', 'poly', 'rbf', 'sigmoid']
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')
```

O/p

```
Text(0.5, 1.0, 'Support Vector Classifier scores for different kernels')
```



The linear kernel performed the best, being slightly better than rbf kernel.

I/p

```
print("The score for Support Vector Classifier is {}% with {}
kernel.".format(svc_scores[0]*100, 'linear'))
```

O/p

```
The score for Support Vector Classifier is 83.0% with linear kernel.
```

Decision Tree Module

We'll vary between a set of max_features and see which returns the best accuracy.

I/p

```
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))
```

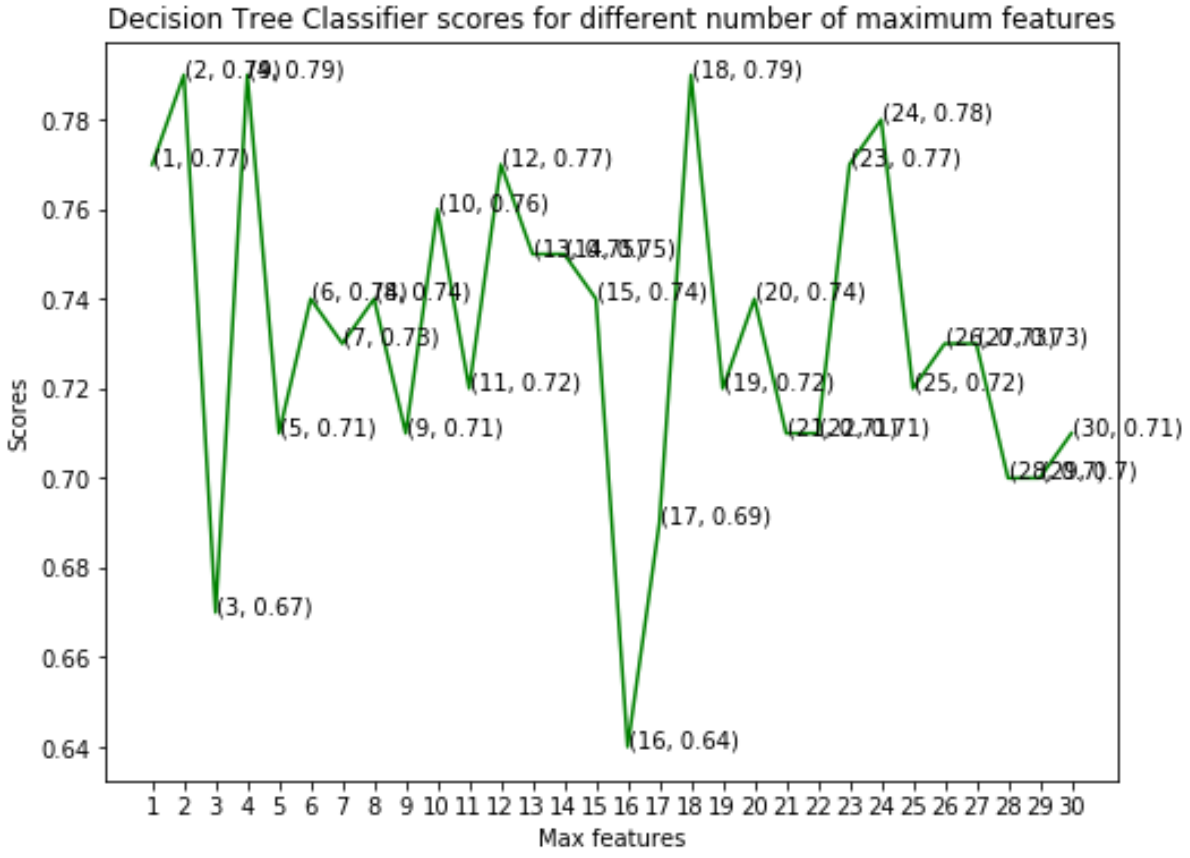
we have selected the maximum number of features from 1 to 30 for split, we'll see what will be the score for each of those cases

I/p

```
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')
```

O/p

```
Text(0.5, 1.0, 'Decision Tree Classifier scores for different number of max
imum features')
```



It achieved the best accuracy at 3 values of max features i.e. 2, 4 and 18

I/p

```
print("The score for Decision Tree Classifier is {}% with {} maximum
features.".format(dt_scores[17]*100, [2,4,18]))
```

```
The score for Decision Tree Classifier is 79.0% with [2, 4, 18] maximum fea
tures.
```

Random forest classifier

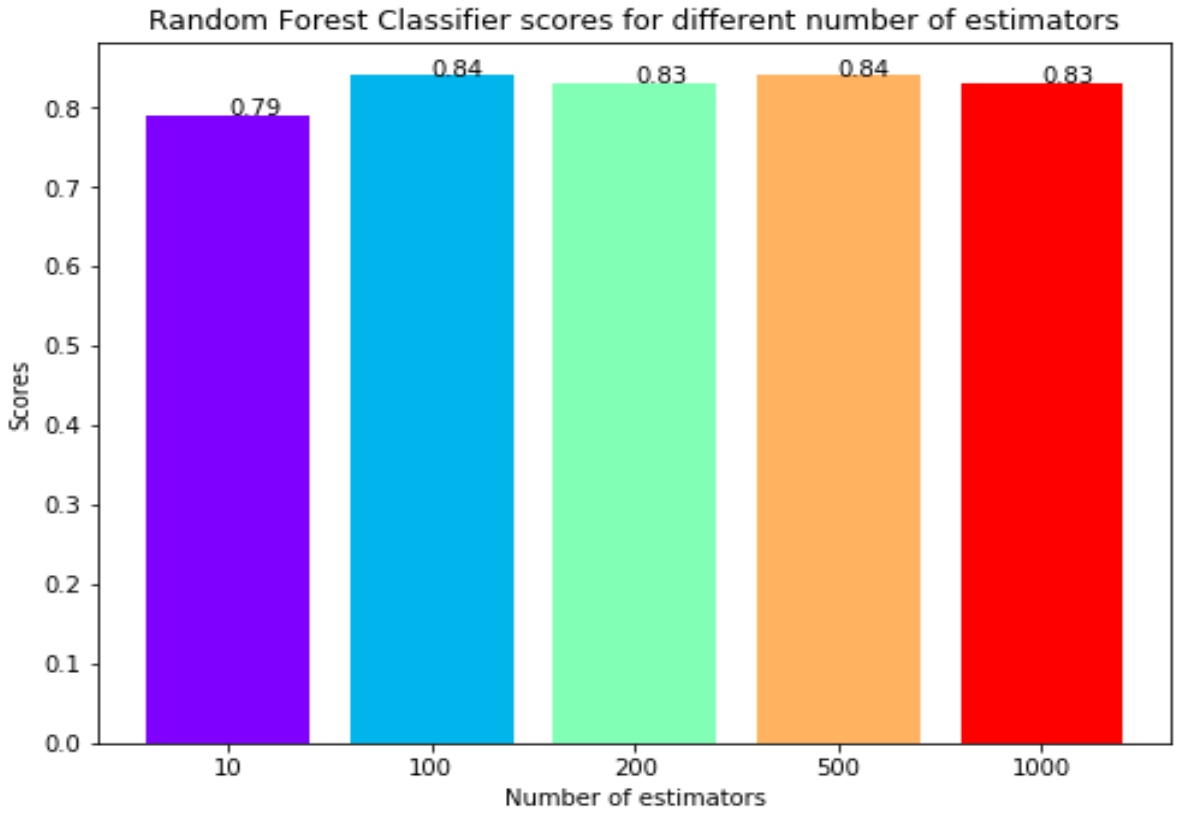
We'll be using the ensemble method to create the model and vary The number of estimators to see their effect

I/p

```
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))
colors = 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')
```

O/p

```
Text(0.5, 1.0, 'Random Forest Classifier scores for different number of est
imators')
```



The maximum score is achieved when the total estimators are 100 or 500.

```
print("The score for Random Forest Classifier is {}% with {}
estimators.".format(rf_scores[1]*100, [100, 500]))
The score for Random Forest Classifier is 84.0% with [100, 500] estimators.
```

CONCLUSION

In this project, I used Machine Learning to see the age groups most affected by heart issues and to predict whether a person is suffering from a heart disease. After importing the data, I analysed it using plots. Then, I generated dummy variables for categorical features and scaled other features. I then applied four Machine Learning algorithms, K Neighbours Classifier, Support Vector Classifier, Decision Tree Classifier and Random Forest Classifier. I varied parameters across each model to improve their scores. In the end, K Neighbours Classifier achieved the highest score of 87% with 8 nearest neighbours and the age group of 50-70 were found to be under the highest risk of heart diseases.

REFRENCES

[1] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In CVPR, 2016. 2, 6, 18, 19

[2] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In ICLR, 2021. 2, 6, 7, 8, 9, 10, 17, 18, 19

[3] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In CVPR, 2019.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. NeurIPS, 2014.

[5] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning Transferable Architectures for Scalable Image Recognition. In CVPR, 2018