Prediction of Heart Disease

Introduction:

In this project, we have downloaded the dataset from Kaggle and we will be using Machine Learning to make predictions on whether a person is suffering from Heart Disease or not.

Information about dataset attributes:

There are a lot of factors that can predict whether a person is suffering from Heart Disease or not. But we have taken mainly 13 attributes for prediction of heard decease.

```
dataset.head()
```

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

age	The person's age in years
sex	The person's sex (1 = male, 0 = female)
cp	The chest pain experienced (Value 1: typical angina, Value 2: atypical angina, Value 3: non-anginal pain, Value 4: asymptomatic)
treatbps	The person's resting blood pressure (mm Hg on admission to the hospital)
chol	The person's cholesterol measurement in mg/dl
fbs	The person's fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false)
restecg	Resting electrocardiographic measurement (0 = normal, 1 = having

	ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria)
thalach	The person's maximum heart rate achieved
exang	Exercise induced angina (1 = yes; 0 = no)
oldpeak	ST depression induced by exercise relative to rest
slop	the slope of the peak exercise ST segment (Value 1: upsloping, Value 2: flat, Value 3: down sloping)
ca	number of major vessels (0-3) coloured by fluoroscopy
thal	A blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect)
target	Heart disease (0 = no, 1 = yes)

Libraries Imported:

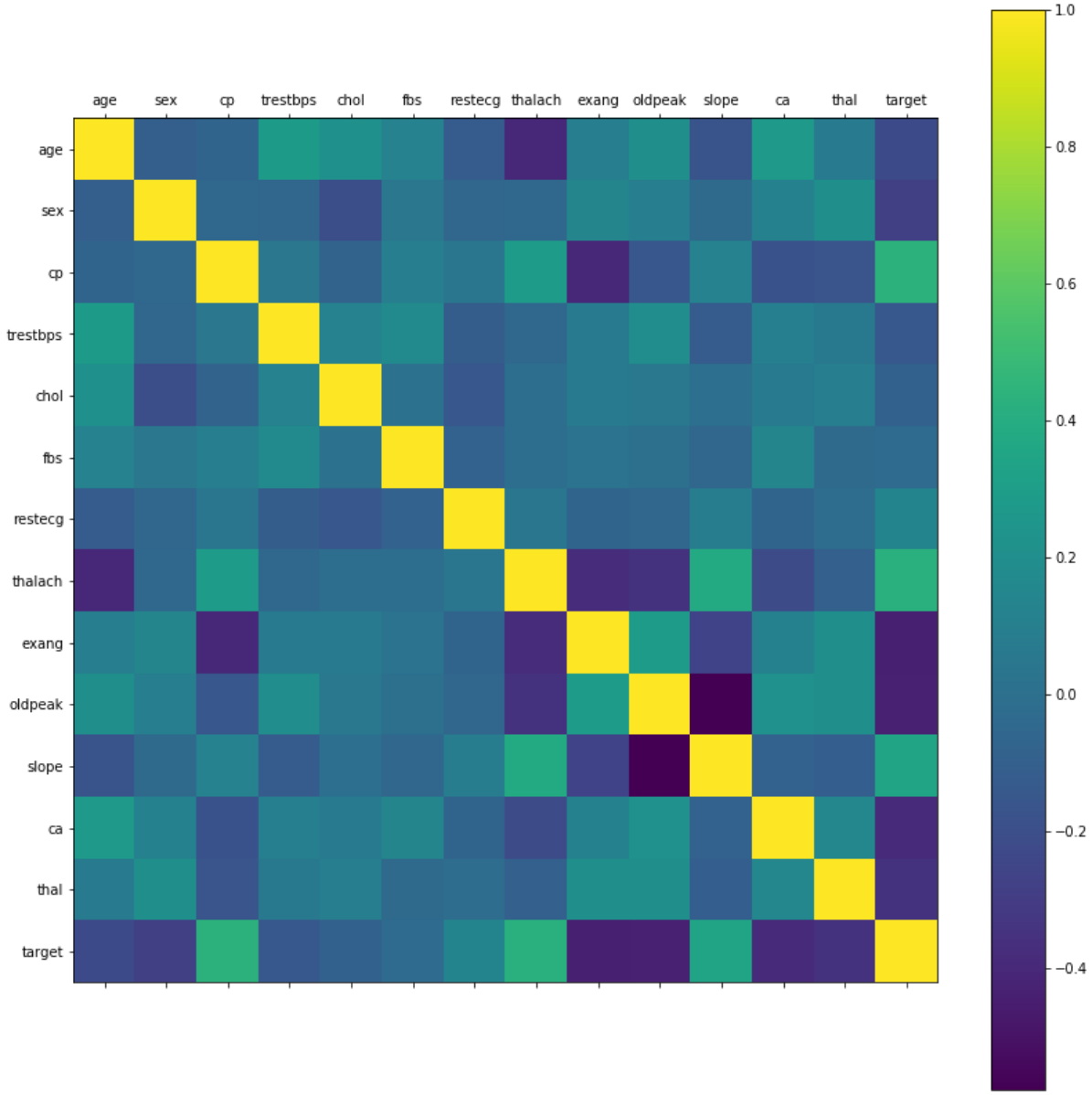
- **numpy:** To work with arrays
- **pandas:** To work with csv files and dataframes
- **matplotlib:** To create charts using pyplot
- Define parameters using **rcParams** and color them with **cm.rainbow**
- **warnings:** To ignore all warnings which might be showing up in the notebook due to past/future depreciation of a feature
- **train_test_split:** To split the dataset into training and testing data
- **StandardScaler:** To scale all the features, so that the Machine Learning model better adapts to the dataset
- **sklearn:** For implementing Machine Learning models and processing of data.

Machine learning algorithms:

For this project we will implement the following 5 Machine Learning algorithms.

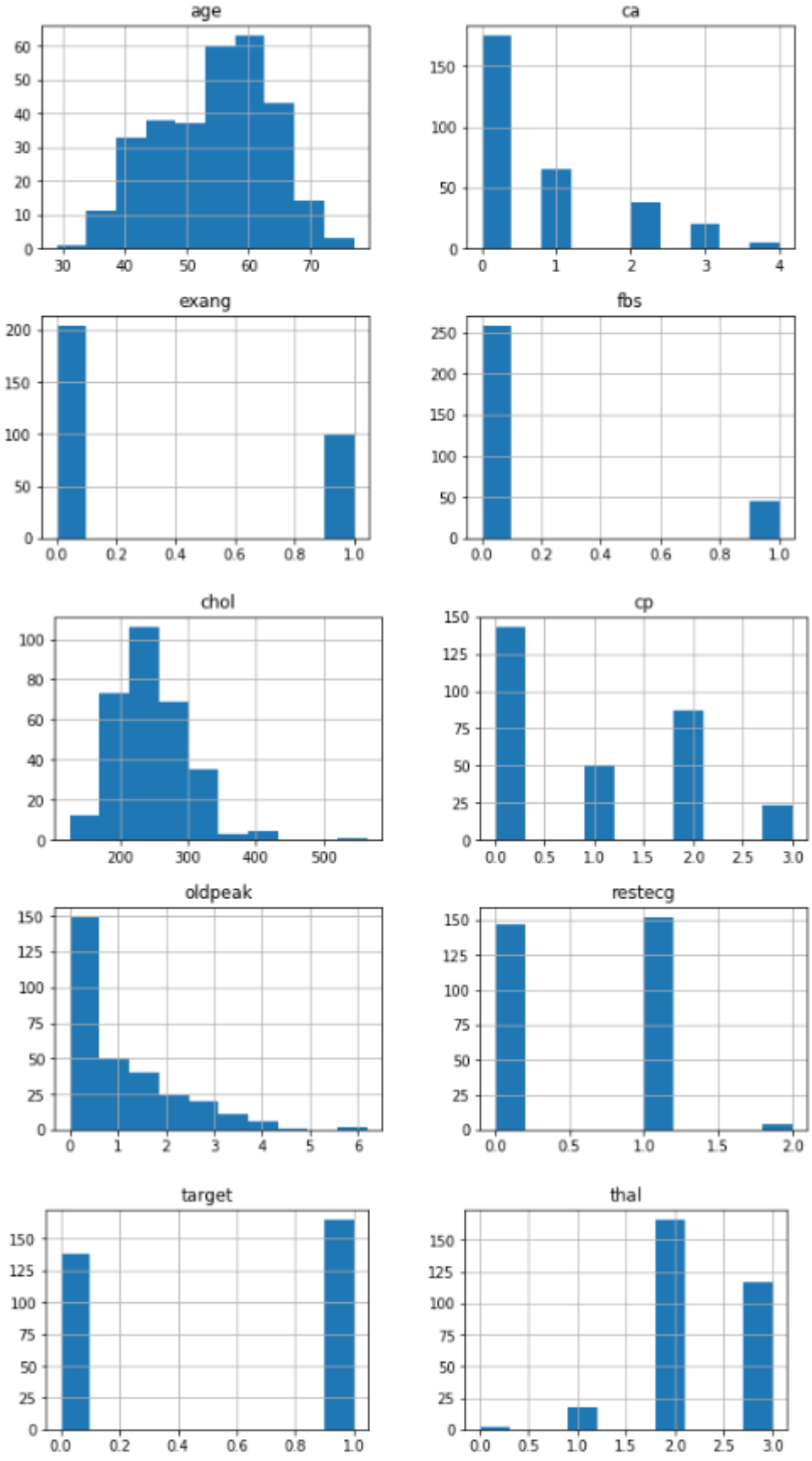
1) K Neighbors Classifier
2) Support Vector Classifier
3) Decision Tree Classifier
4) Random Forest Classifier
5) Logistic Regression

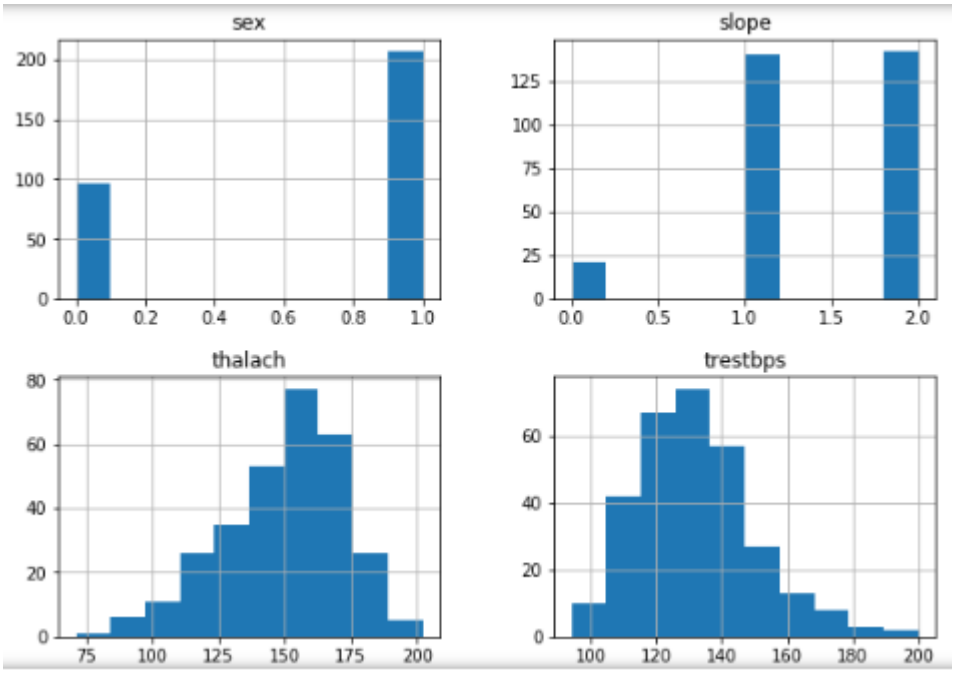
Data visualization for understanding the dataset



This is the correlation plot of the dataset attributes with each other. The 1 correlation is coloured yellow which has the highest effect on another feature while the negative correlations are purple accordingly.

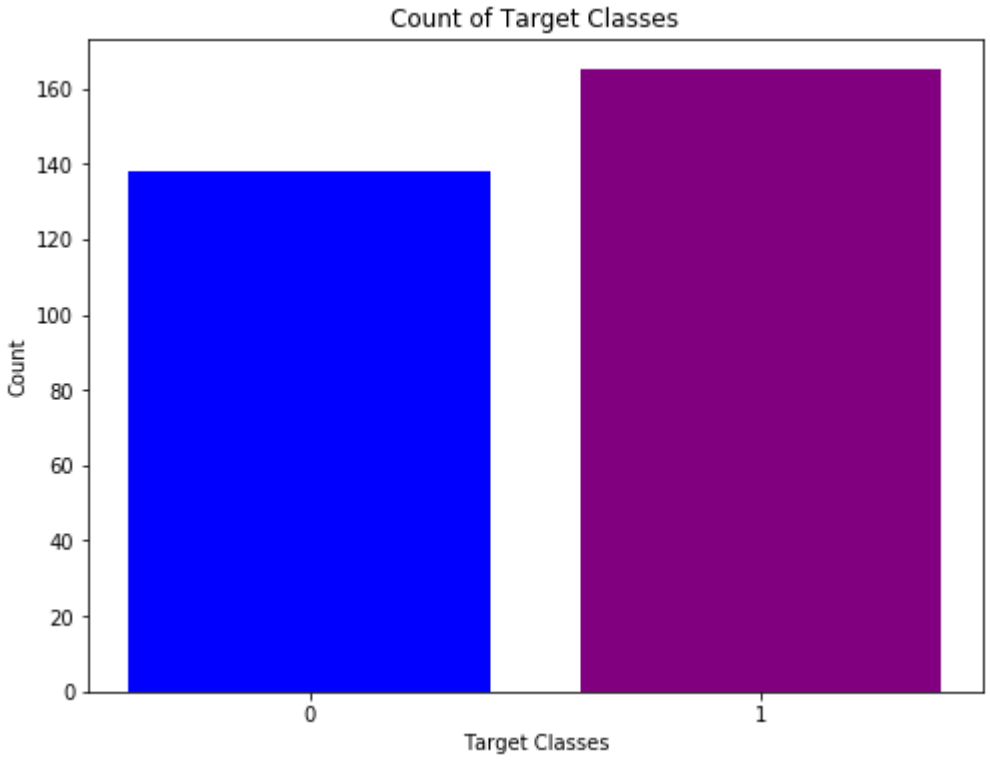
Histogram of dataset attributes:





Taking a look at the histograms above, I can see 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.

Now let's check how many people have Hear Disease in this dataset that contain 303 rows.



We can see that around 160+ have Heart Disease and 130+ don't have the Heart Disease.

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.

First, we'll use the `get_dummies` method to create dummy columns for categorical variables.

And then we will use the `StandardScaler` from `sklearn` to scale the dataset.

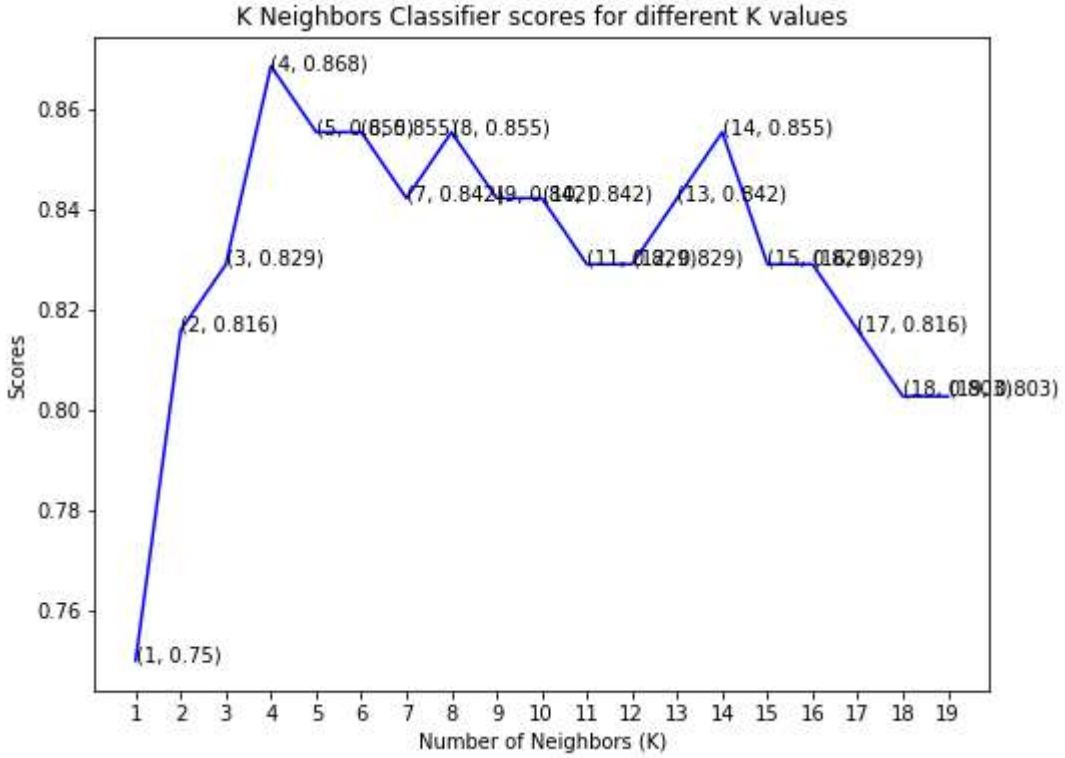
Machine learning:

We have imported the train_test_split to split our dataset into training and testing datasets. We have taken 75% for training and the rest 25% for testing.

Now we will build machine learning models through different Machine learning algorithms to check which algorithm is performing better in our case.

1. **K Neighbors Classifier:**

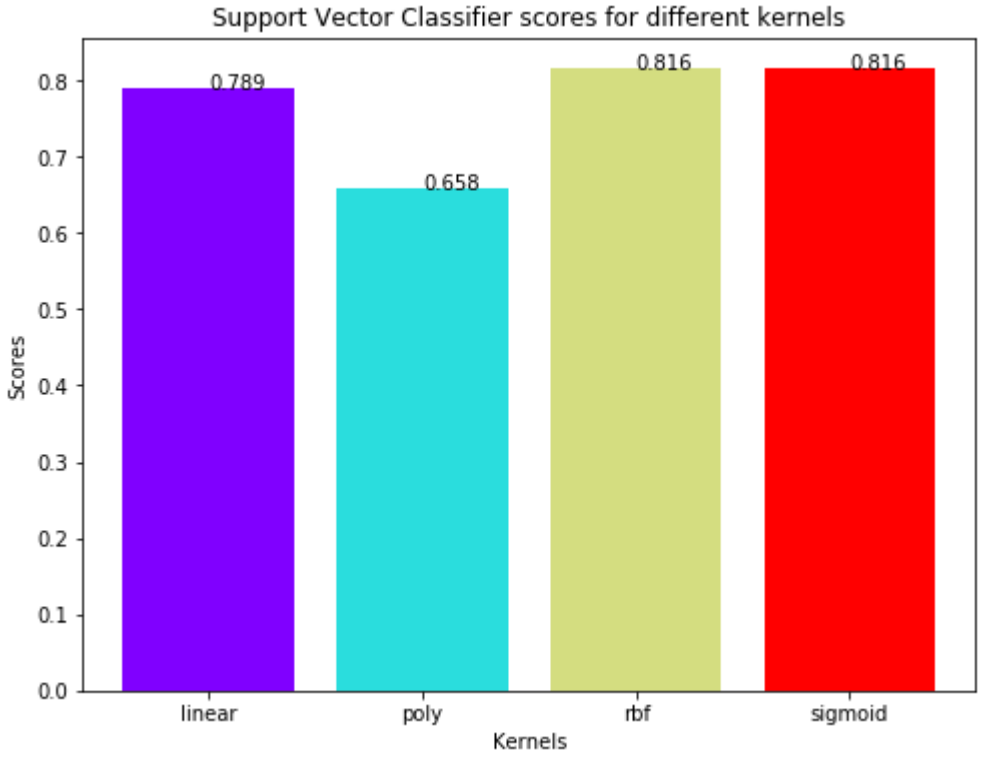
   The classification score varies based on different values of neighbours that we choose. Thus, we'll plot a score graph for different values of K (neighbours) and check when do we achieve the best score.



The score for K Neighbors Classifier is 86.8421052631579% with 4 nieghbors.

2. **Support Vector Classifier:**

There are several kernels for Support Vector Classifier. we'll test some of them and check which has the best score.

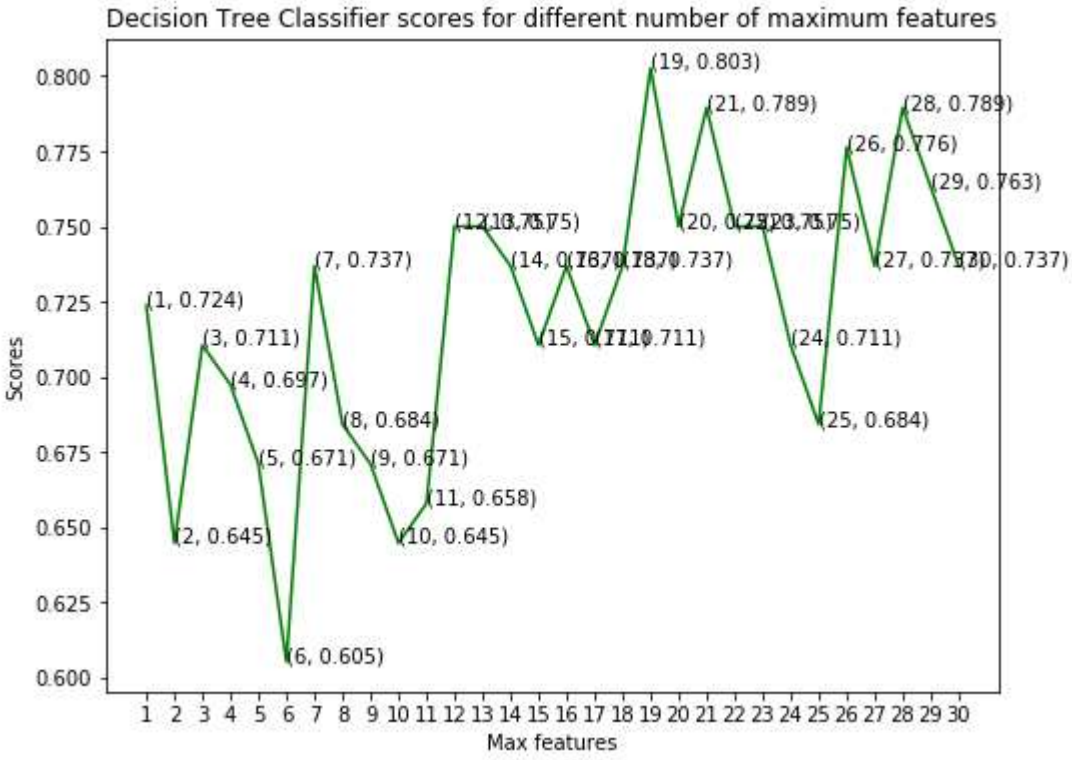


We can clearly see that we have achieved our best score through two kernels that are ‘rbf’ and ‘sigmoid’ with the accuracy score of 81.6

3. **Decision Tree Classifier:**

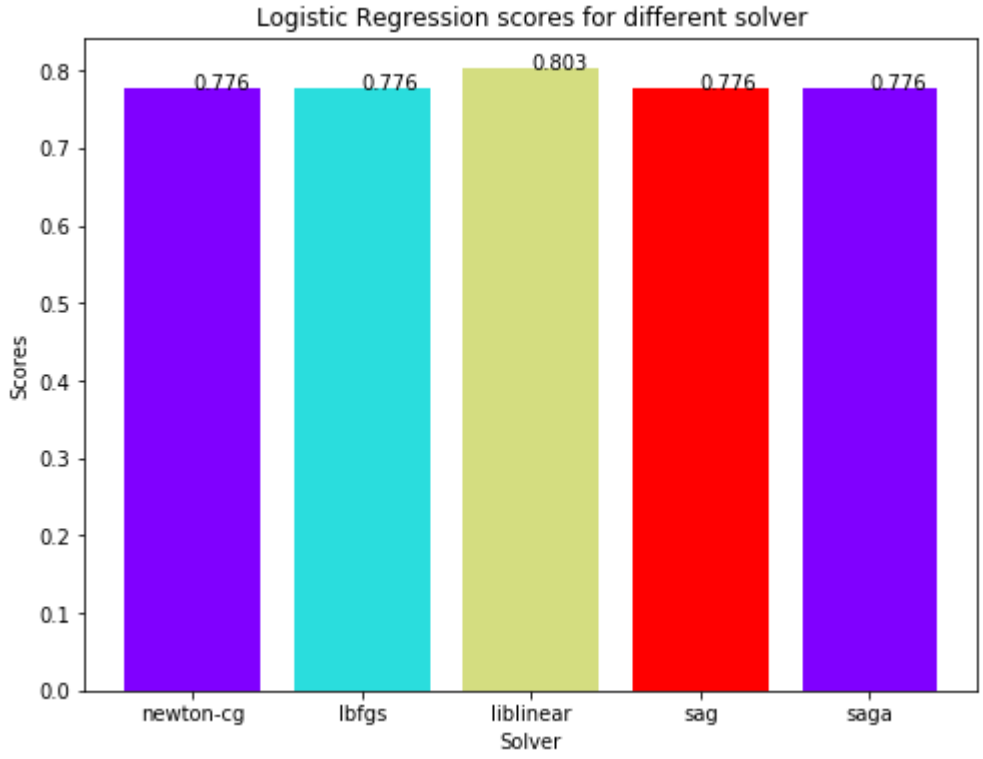
We'll use the Decision Tree Classifier to model the problem at hand. We'll vary between a set of `max_features` and see which one returns the best accuracy.

In the below figure we can see that the model achieved the best accuracy at the value of 19 maximum features, with an accuracy score of 80.3



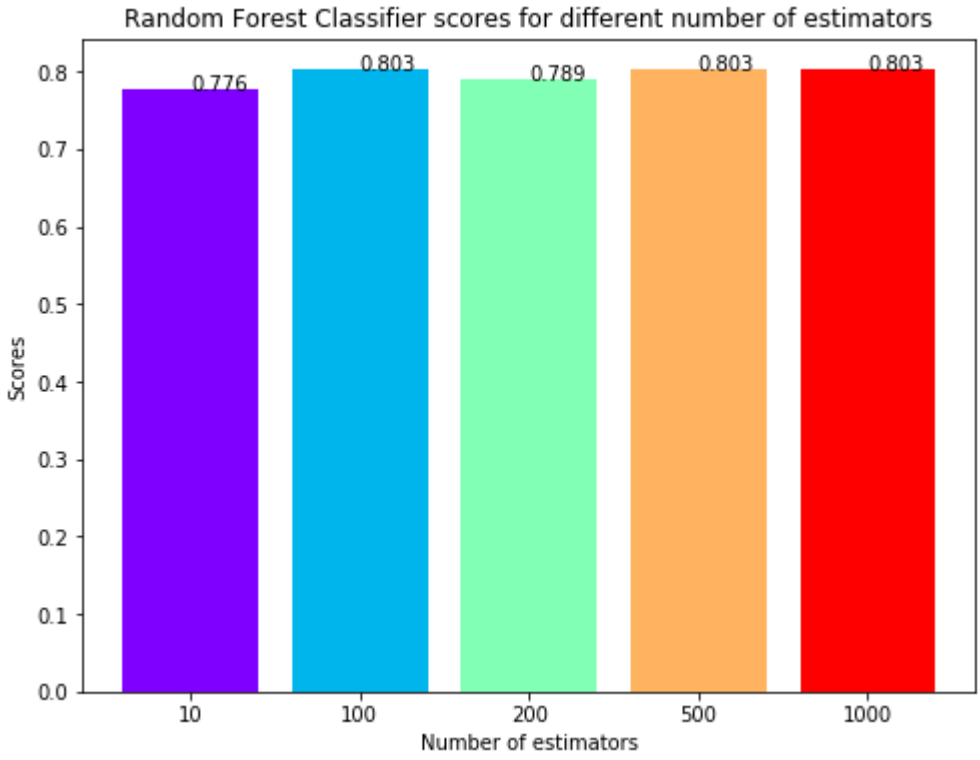
4. Logistic Regression:

In logistic regression, we have checked through different solver for achieving our best score. In the bar plot below we can see that liblinear solver performed the best at the accuracy score of 80.3



5. Random Forest Classifier:

We'll use the ensemble method, Random Forest Classifier, to create the model and vary the number of estimators to see their effect and check which estimator is achieving the best score.



The maximum score is achieved when the total estimators are 100 or 500 or 1000.

Report:

Algorithms	Accuracy Score
KNN	86.8%
Logistic Regression	80.3%
Decision Tree	80.3%
SVM	81.6%
Random Forest classifier	80.3%

Conclusion:

In this project, we used Machine Learning to predict whether a person is suffering from a heart disease. After importing the data, we analysed it using plots. Then, we did generated dummy variables for categorical features and scaled other features.

we then applied five Machine Learning algorithms, `K Neighbors Classifier`, `Support Vector Classifier`, `Decision Tree Classifier`, `Logistic Regression` and `Random Forest Classifier`. We varied parameters across each model to improve their scores.

In the end, `K Neighbors Classifier` achieved the highest score of `87%` with `4 nearest neighbors`.