**INTRODUCTION**

Heart disease is the one of the most common disease. This disease is quite common now days we used different attributes which can relate to this heart diseases well to find the better method to predict. Heart Disease (including Coronary Heart Disease, Hypertension, and Stroke) remains the No. 1 cause of death in the US. The Heart Disease and Stroke Statistics—2019 Update from the **American Heart Association** indicates that:

* 116.4 million, or 46% of US adults are estimated to have hypertension. These are findings related to the new 2017 Hypertension Clinical Practice Guidelines.
* On average, someone dies of CVD every 38 seconds. About 2,303 deaths from CVD each day, based on 2016 data.
* On average, someone dies of a stroke every 3.70 minutes. About 389.4 deaths from stroke each day, based on 2016 data.

This case study aims to model the probability of heart disease of the adults , available on UCI ML Repository. This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The "goal" field refers to the presence of heart disease in the patient.

Below is a table showing names of all columns and their description.

**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)

**trestbps:** The person's resting blood pressure (mm Hg on admission to the hospital)

**chol:** The person's cholesterol measurement in mg/dl

**fbs:**  The person's fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false)

**restecg:** Resting 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 ('ST' relates to positions on the ECG plot. See more here)

**slope:** the slope of the peak exercise ST segment (Value 1: upsloping, Value 2: flat, Value 3: downsloping)

**ca:** The number of major vessels (0-3)

**thal:** A blood disorder called thalassemia (3 = normal; 6 = fixed defect; 7 = reversable defect)

**target:** Heart disease (0 = no, 1 = yes)

**Task:** Classification

**DATA PREPROCESSING**

**Importing Libraries**

First we will import some important libraries which are numpy as np, pandas as pd, matplotlib.pyplot as plt, seaborn as sns.

**NumPy:** NumPy stands for Numerical Python. NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

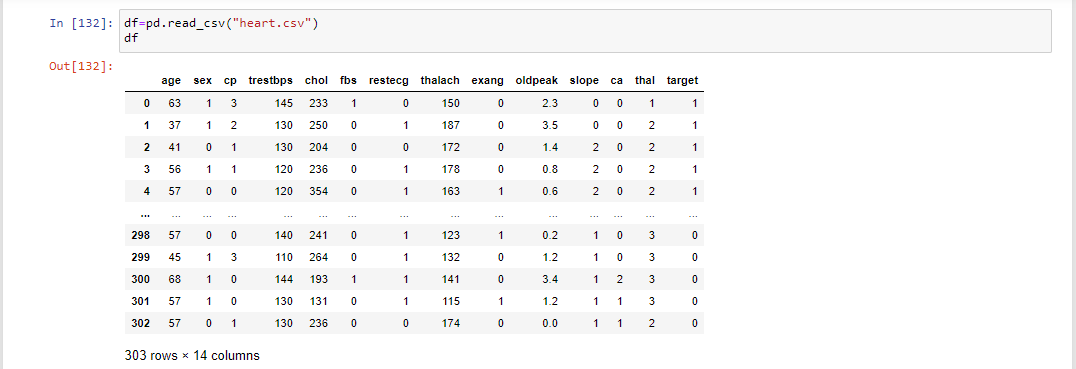
**Pandas:** The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" used for working with data sets.It has functions for analysing, cleaning, exploring, and manipulating data.

**Matplotlib:** Matplotlib is a low level graph plotting library in python that serves as a visualization utility. Most of the Matplotlib utilities lies under pyplot sub module, and are usually imported under the plt alias.

**Seaborn:** Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions.

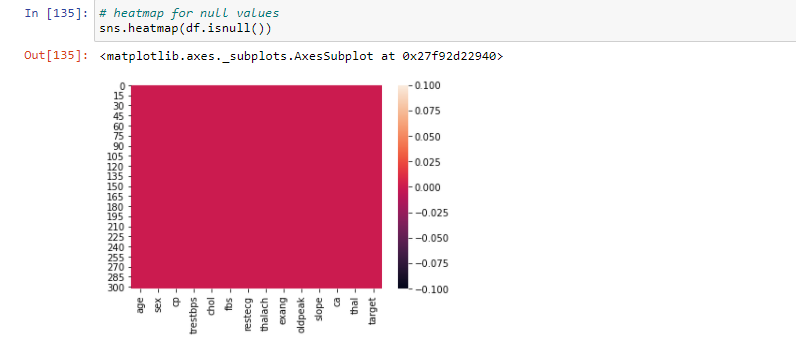
**Loading Dataset**

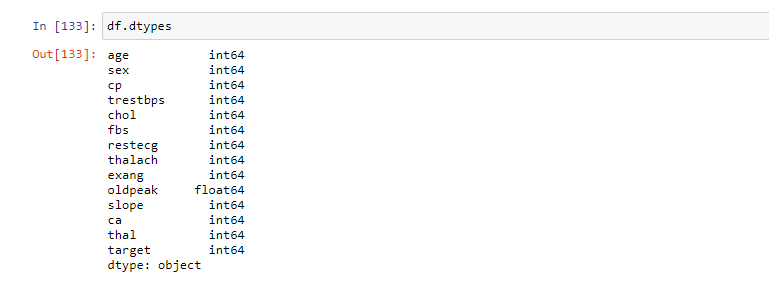
Now we are loading the datasets using pandas. Dataset are loaded as df. Take a look at the dataset...



**Checking null values and datatypes**

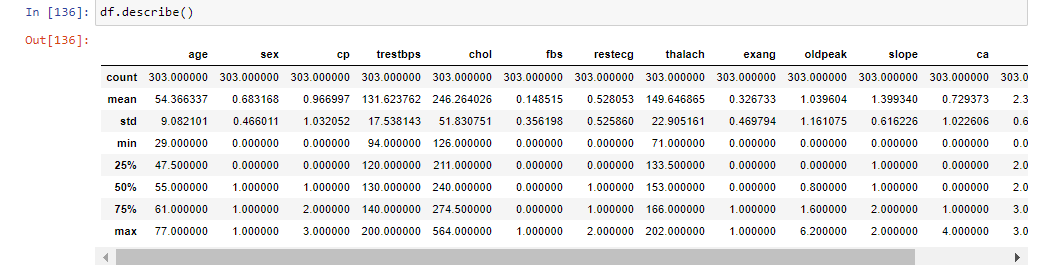
We are checking that if our datasets contains any null values also check the datatypes of every column of datasets.





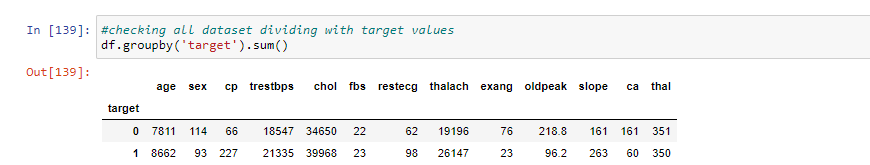
We observe that our dataset doesnot contains any null value and each column in the dataset is in numeric datatype.

We also checked for unique value present in our data by using unique() function with column name. After that we also checked data describe() and conclude the min , max , mean value of every column.



We clearly observed that minimum age is 29, maximum age is 77 and mean age is 54.3, similarly minimum trestbps is 94, maximum trestbps is 200 and mean is 131.6 same way minimum chol is 126 maximum is 564 and mean is 246.26, again for thalach minimum is 71 maximum is 202 and mean is 149.6.

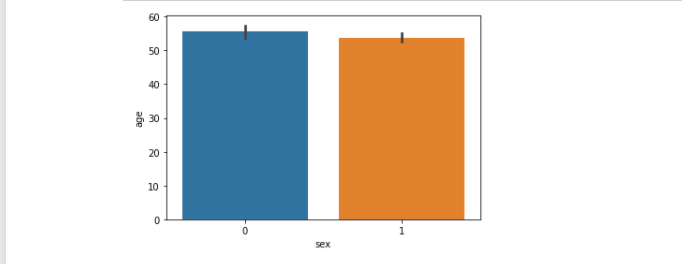
Now we have grouped our dataset according to target variable:



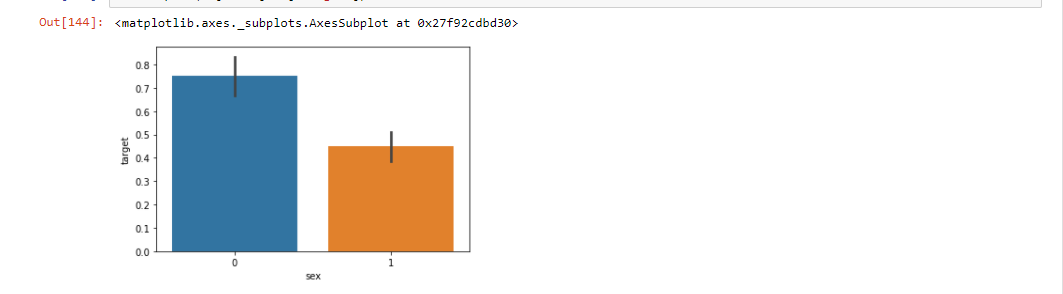
**DATA VIZUALISATION**

We will do some visualizations on training dataset only by using sns and plt which we already imported at the beginning.

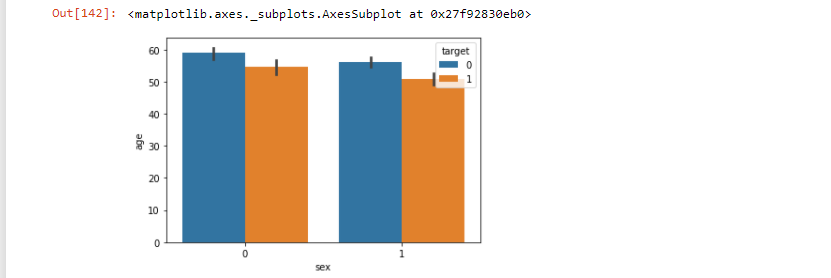
Plot between sex and target column:



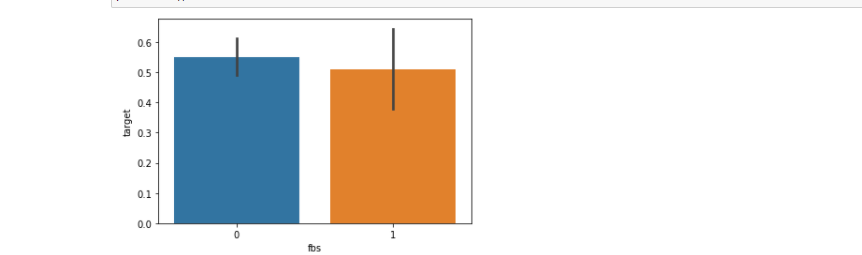
Plot between sex and age column:



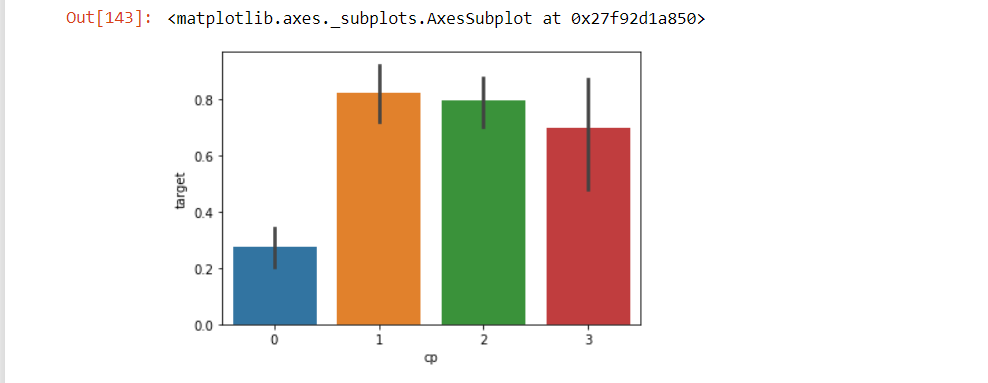
Plot between sex and age with respect to target column:



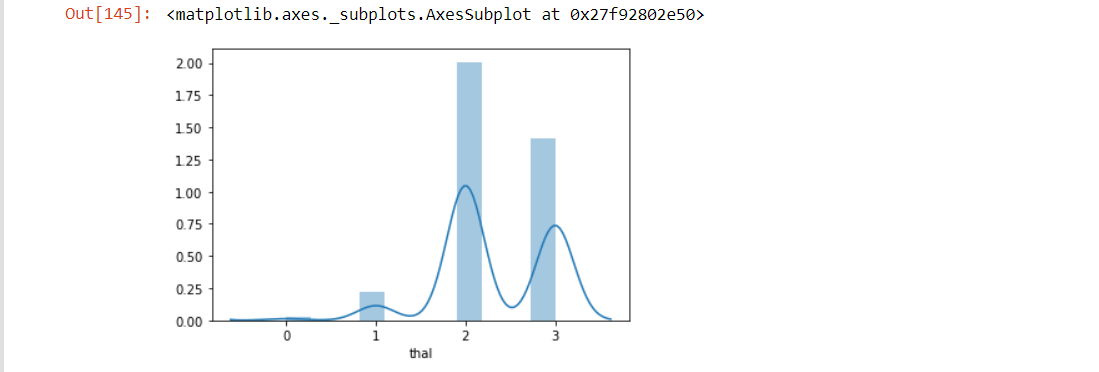
Plot between fbs and target column:



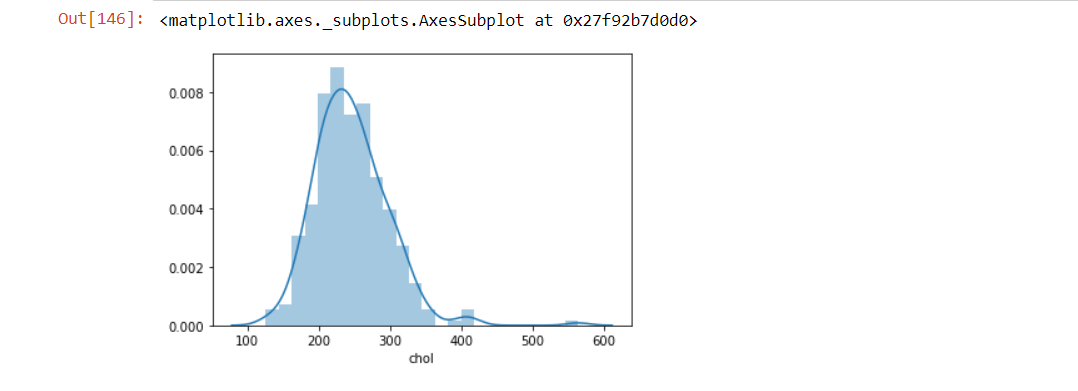
Plot between cp and target column:



Distribution plot of thal column:



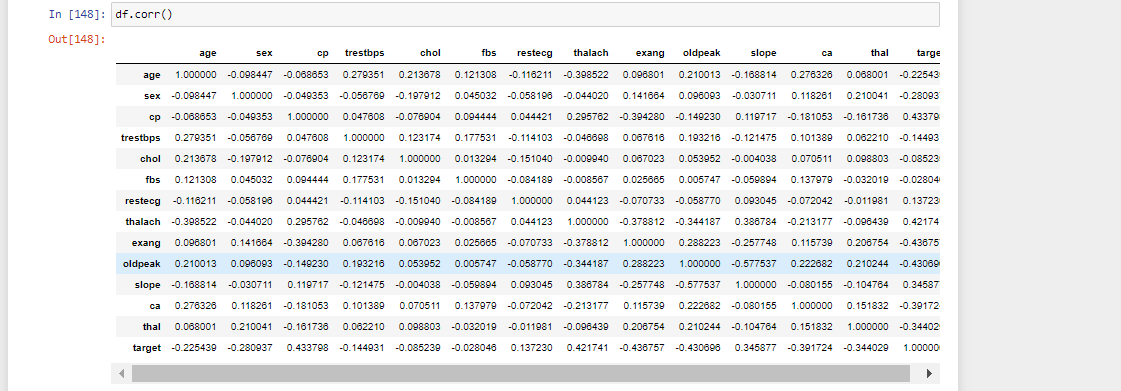
Distribution plot for chol column:

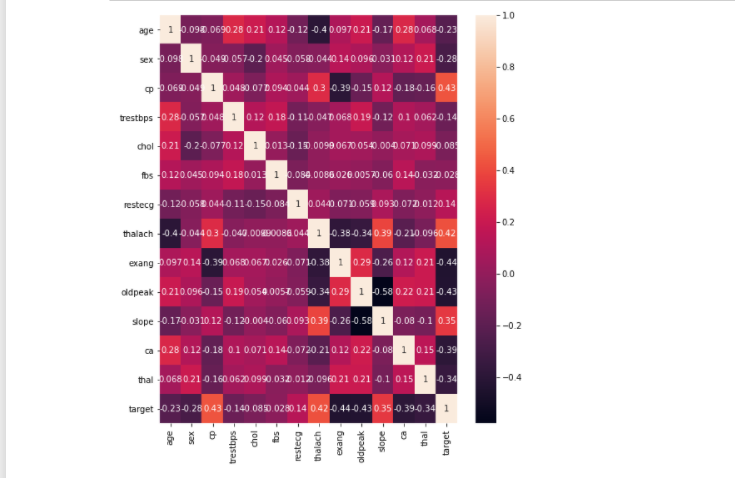


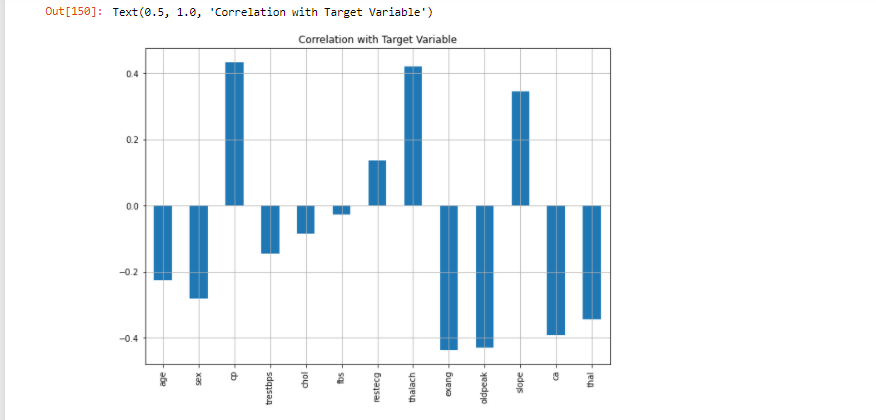
EDA

EDA is known as the exploratory data analysis. In this analysis it is shown that which variable has the most impact on the target variable and how a column is performing in predicting the target variable.

In this section first we check correlation using df.corr() than we used heatmap correlation. For implementing this we imported Seaborn library. After applying heatmap correlation matrix. We can see the correlation between variables. At last we also check corelation using visualization method by plotting graph with target variable.





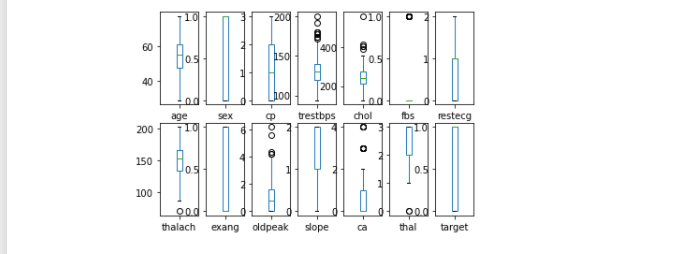


We observe that most of the columns are negatively corelated, so the highly positively correlated column is cp which is 0.43 and negatively correlated column is t1 which is -0.44.

**Checking Outliers**

An outlier is a data point in a data set which is distant or far from all other observations available. It is a data point which lies outside the overall distribution which is available in the dataset.

Now we will check outliers by using boxplot .



We observed that trestbps, chol, fbs, oldpeak, ca contains outliers.

**Removing the Outliers**

We remove the outliers using z-score.

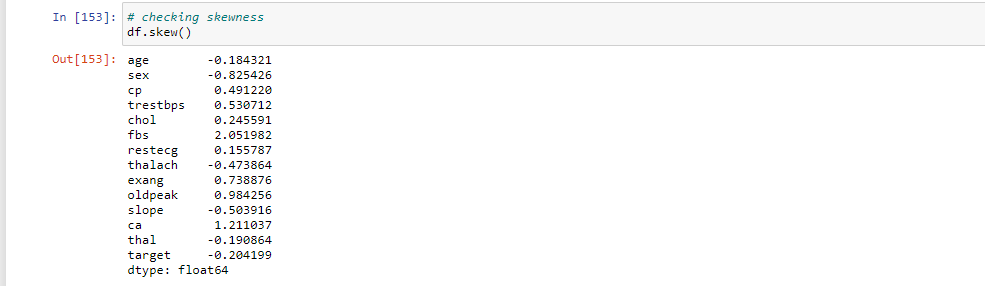
A Z-score is a numerical measurement that describes a value’s relationship to the mean of a group of values in the dataset. Z-score is measured in terms of standard-deviation from the mean. For removing outliers we have detected Z-score. We have removed data which has Z-score more than 3. We import it from scipy.stats library.

We observed that before removing outliers we have 303 rows in our dataset now after removing outliers we have 287 rows in our dataset.

**Checking the skewness:**

Skewness refers to distortion or asymmetry in a symmetrical bell curve, or normal distribution in a set of data. Besides positive and negative skew, distributions can also be said to have zero or undefined skew. The skewness value can be positive, zero, negative, or undefined. For normal distribution it should be between -0.5 to 0.5.

We are now checking the skewness of every column. let’s see it below



**Treating the skewness:**

For normal distribution skewness should be varying between -0.5 to 0.5 so we have to treat the skewness hare to fit it inti these parameters so that it would not be left skewed or right skewed. There are many methods to skewed it like log transform, square root transform, cube root transform, power transform. Here we are using log transform method to treat the skewness.

**Splitting the Data**

Now we will split our data into two parts x and y where x consists of all the data except target column to train the model while other y contains target column which is target column in this dataset.

**Using Standard Scaler**

Now we use standard scaler to transform our data such that its distribution will have mean value “0” and standard deviation if “1”. In case of multi variate data like in our dataset this is done feature wise which means independently for each column of the data. According to distribution of the data, each value in the dataset will have the mean value subtracted, and then divided by the standard deviation of the whole dataset. We have to import the standard scaler from sklearn and then just pass the data by calling it.

**Splitting the data into train and test:**

Now we will split our data into x\_train, x\_test, y\_train, y\_test in the ratio 80%(for training) and 20%(for testing), for this we have to import train\_test split from sklearn model selection.

**Sending Data to Models:**

Now we are sending our data to various models for training and testing to check the accuracy score. Models used here are:

**1. Logistic Regression –** Logistic Regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).

**2.** **Gaussian Naive Bayes** - algorithm is a special type of NB algorithm. It's specifically used when the features have continuous values. It's also assumed that all the features are following a gaussian distribution i.e, normal distribution. A Gaussian classifier is a generative approach in the sense that it attempts to model class posterior as well as input class-conditional distribution. Therefore, we can generate new samples in input space with a Gaussian classifier.

**3.** **SVC** - “Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification and regression challenges. It is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving a SVM model sets of labeled training data for each category, they're able to categorize new text.

**4.** **Decision Tree Classifier** - Decision tree learning is one of the predictive modelling approaches used in statistics, data mining and machine learning. It uses a decision tree to go from observations about an item to conclusions about the item's target value.

**5. KNeighbors** **Classifier** - It is a method based on k-nearest neighbors. In the KNeighbors model target is predicted by local interpolation of the targets which associated to the nearest neighbors in the training set. KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification).

**Results:**

|  |  |
| --- | --- |
| **Models** | **Accuracy Score** |
| Logistic Regression | 0.849375 |
| Gaussian Naive Bayes | 0.8125 |
| SVC | 0.828125 |
| Decision Tree Classifier | 0.703125 |
| KNeighbors Classifiers | 0.84375 |

Here Logistic Regression and KNeighbors are giving good results but we will also see accuracy by using some ensemble tecniques too.

**Using Ensemble Techniques**

**Ensemble methods** is a machine learning technique that combines several base models in order to produce one optimal predictive model. Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model on order to decrease variance (bagging), bias (boosting), or improve predictions (stacking).

**Models Used:**

**1. Random Forest Classifier -** Random Forest uses multiple decision trees as base learning models in the dataset. Random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting in the dataset. The main concept of Random Forest is to combine multiple decision trees in determining the final result rather than relying on individual decision trees.

**2.** **AdaBoost Classifier** - AdaBoost is best used to boost the performance of decision trees on binary classification problems. AdaBoost was originally called AdaBoost. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers.

**3. Gradient boosting classifiers** - are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting.

**Results**

|  |  |
| --- | --- |
| **Model** | **Accuracy Score** |
| Random Forest Classifier | 0.78125 |
| Adaboost Classifier | 0.8125 |
| Gradient Boosting Classifier | 0.75 |

**Using GridSearchCV**

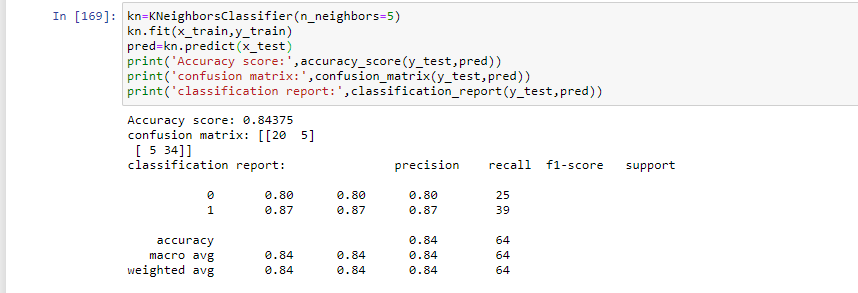
GridSearchCV is the process of performing hyperparameter tuning in order to determine the optimal values for a given model. The performance of a model significantly depends on the value of hyperparameters and there is no way to know in advance the best values for hyperparameters so ideally, we need to try all possible values to know the optimal values. Doing this manually could take a considerable amount of time and resources and thus we use GridSearchCV to automate the tuning of hyperparameters.

In short GridSearchCV is a library function that is a member of sklearn's model\_selection package. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters.

Here we find the best parameters for Logistic Regression and KNeighbors. So best parameters for Logistic Regression is: ‘C’=0.1 and for KNeighbors is: n\_neighbors=5.

**Best Model**

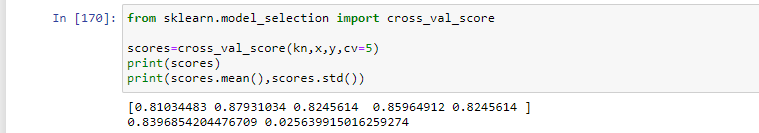
From above we have calculated the accuracy score with best parameters for logistic regression and kneighbors. From these two kneighbor is the best. Below is the result with confusion matrix and classification report.



**Cross validation**

Cross validation helps to find out the over fitting and under fitting of the model. In the cross validation the model is made to run on different subsets of the dataset which will get multiple measures of the model. If we take 5 folds, the data will be divided into 5 pieces where each part being 20% of full dataset. While running the Cross validation the 1 st part (20%) of the 5 parts will be kept out as a hold out set for validation and everything else is used for training data. This way we will get the first estimate of the model quality of the dataset. In the similar way further iterations are made for the second 20% of the dataset is held as a hold out set and remaining 4 parts are used for training data during process. This way we will get the second estimate of the model quality of the dataset. These steps are repeated during the cross validation process to get the remaining estimate of the model quality.

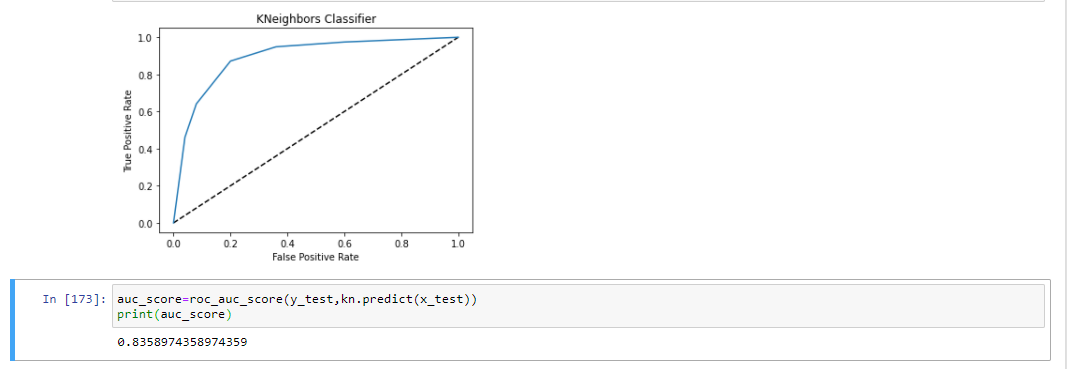
Below is our cross validation result:



**AUC ROC CURVE**

AUC – ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve, and AUC represents the degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between patients with the disease and no disease.

The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.



**Saving The Model**

We will save this model which is kneighbors classifier for future use using joblib by importing joblin library.

**Conclusion**

We conclude that dataset contains null values. While doing EDA we have remove outliers and also treat the skewness than sent our data to models also used ensemble techniques to get the best accuracy score and find that logistic and kneighbors both giving high accuracy so we find best parameters for both and again checked the accuracy score using best parameters and get to know that the best model for prediction is KNeighbors Classifier which we saved using joblib for future use.