# Blog/Article on Heart Disease Prediction Dataset

Contents:-

1. Problem Definition
2. Data Analysis
3. EDA
4. Pre-Processing Pipeline
5. Building Machine Learning Models
6. Conclusion

## Problem Definition:

In this project we are given heart disease dataset to predict if the patient is suffering from the heart disease or not. This dataset is containing 14 attributes. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The "goal" field or target variable of the dataset refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 1, 2, 3, 4(presence of the heart disease). Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1, 2, 3, 4) from absence (value 0). As values 1, 2, 3, 4 denotes the presence of heart disease we can map them as value 1 and value 0 denotes absence of heart disease so we can map it as 0. Now there are only two types of values in the target variable as 0 and 1 where 0 – heart disease absent and 1 – heart disease present.

So, from the information above we can say that it is classification problem as we have detect the presence of heart disease or not. In this project we will be different types of classifier algorithms like Logistic, KNN , SVC, Decision tree algorithms.

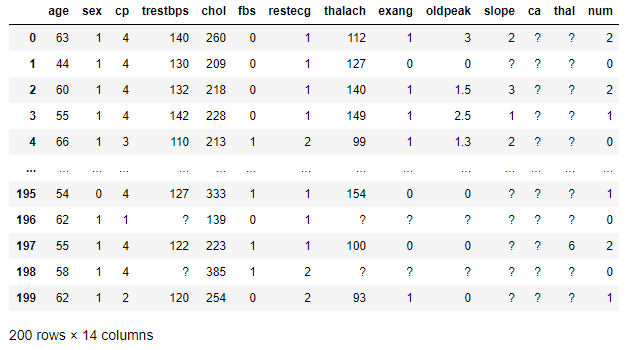
## *Data* *Analysis*:

Here in this section, we will analyse the dataset by loading it from Jupyter notebook where the file is stored in the CSV(comma separated values) format, checking for data types of attributes of the dataset, checking for information of the dataset to see how many numerical and categorical columns are present in the dataset, how much the memory is occupied by the data, checking the dimension of the dataset(rows and columns), checking for the null or Not a number (NaN) values in the dataset to see whether data cleaning is required or not.

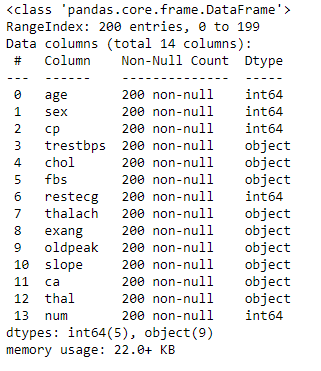
First of all, we will import the required libraries which we will be used for coding the dataset. We will import pandas library as “pd” to import the dataset as csv (comma separated value) file from the Jupyter Notebook.

First, we have loaded the dataset using panda’s library and saved the dataset in the instance “df” variable. We have printed the dataset by calling instance of the dataset “df”. The dataset contains 14 attributes which are as follows: -

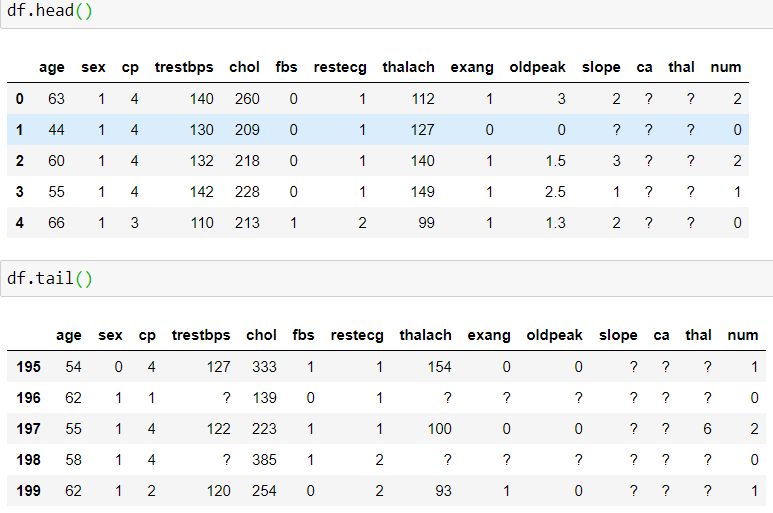
* age - age in years
* sex - (1 = male; 0 = female)
* cp - chest pain type 🡪
  + Value 1: typical angina
  + Value 2: atypical angina
  + Value 3: non-anginal pain
  + Value 4: asymptomatic
* trestbps - resting blood pressure (in mm Hg on admission to the hospital)
* chol - serum cholesterol in mg/dl
* fbs - (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
* restecg - resting electrocardiographic results
* thalach - maximum heart rate achieved
* exang - exercise induced angina (1 = yes; 0 = no)
* oldpeak - ST depression induced by exercise relative to rest
* slope - the slope of the peak exercise ST segment
* ca - number of major vessels (0-3) coloured by fluoroscopy
* thal - 3 = normal; 6 = fixed defect; 7 = reversable defect
* target - have disease or not (1=yes, 0=no)



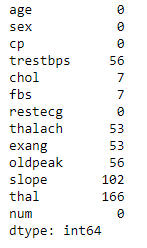
After loading the dataset, we check the information of the dataset by using df.info() command, then we check for the dimension of the dataset by calling the df.shape, which tells us that 200 rows and 14 columns. Then we check for the data types of the dataset using the df.dtypes which tell us that the dataset contains 5 columns of int(numerical) data type and 9 columns of object(categorical) datatype.



We have used the command df.head() to display the first five rows of the dataset, df.tail() to display the last five rows of the dataset, df.sample() to display a random sample of the dataset which is displayed in the figures below.



Then we proceed to check for the cleaning of the data if required i.e, by checking if the dataset contains the null values by using df.isnull().sum() command and using graphical method with the help of heatmap also we check for the null values, which tells us that there are some null values present in the dataset. So, data cleaning is required. There are “?” values also present in the dataset which we have to first convert them to “np.NaN” values then we have to fill the null values by mean, median or mode values of particular column.



The null values present in the dataset are: -

1.trestbps 56

2. chol 7

3. fbs 7

4. thalach 53

5. exang 53

6. oldpeak 56

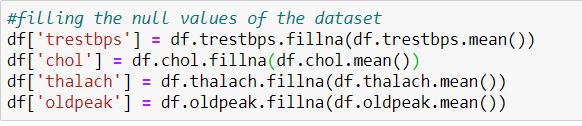
7. slope 102

8. thal 166

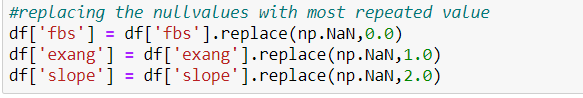
In the “ca” column there are “?” – 198 values and 0 – 2 values out of 200 total values. So, these columns do not provide any information. So, we can drop it from the dataset.

Similarly, in the “thal” column there are 166 null values out of 200 total values which tell us that more than 80% of data is missing in the column. So, we can drop this column too.

In the columns “trestbps”, “chol”, “thalach”, “oldpeak” there are some null values which is filled with mean of the particular column as the columns are of numeric type.



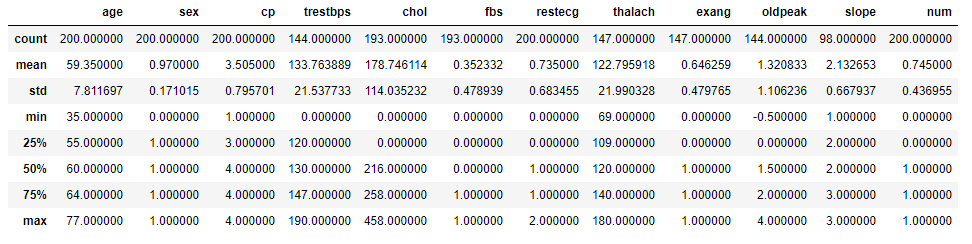
In the columns “exang”, “fbs”, “slope” there are some null values which can be handled with mode of that column as it is categorical column.



Now we convert the target column “num” into two values 0 or 1 using mapping the value 1, 2, 3, 4 as value 1 and value 0 as 0.



Then we check the statistical summary using the describe method i.e., df.describe(). The figure is attached below.



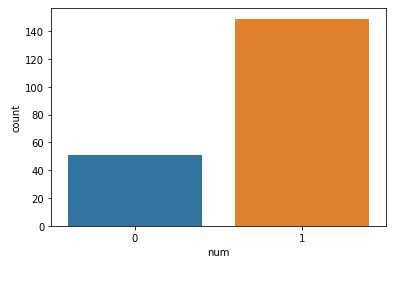
The observations of the statistical summary are as follows: -

1. first of all, the count of all the columns is different. so, there are some missing values in the data.
2. In the columns “age”, “cp”, “chol”, “restecg”, “exang”, “oldpeak”, “num” the mean values are smaller than the median values. So, data is skewed.
3. In the columns “trestfbs”, “fbs”, “thalach”, “slope” the mean values are greater than median values.so data is skewed.
4. From above two points we need to normalize the data by removing the skewness.

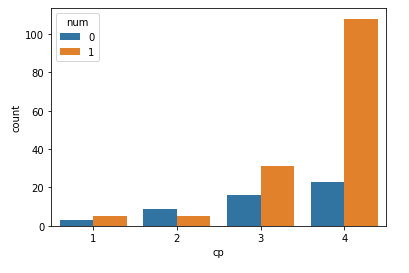
## Exploratory Data Analysis:

Here in this section, we will visualize the dataset by using some libraries like Matplotlib which is imported as instance “plt” and Seaborn which is imported as instance “sns”. Both these libraries help us to plot the different attributes of the dataset by using bar plot, histogram, count plot, scatter plot, swarm plot, etc.

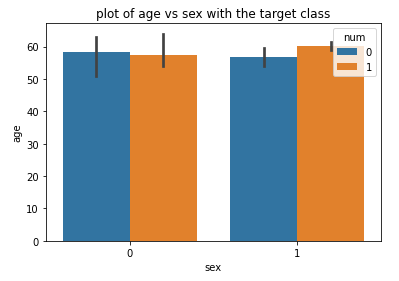
First of all we have made a count plot of the target to see how much patient are suffering from the heart disease. The plot tells us that there are more patients who are suffering from the heart disease.



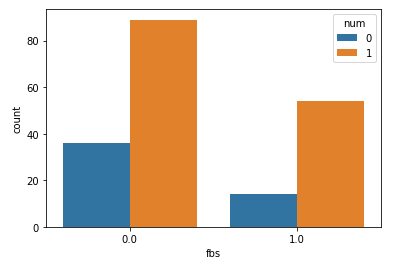
Then we have a count plot for the “cp” column with respect to target column. The plot tells us that there four types of chest pain and they are converted into numeric types 1, 2, 3, 4. The plot tell us that patient with chest pain type 4 is more prone to heart disease.



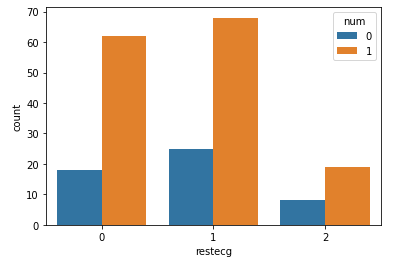
Then we have bar plot of age vs sex with respect to target column. There are two type sex: 1- male, 2- female. The plot tells us that male with mean age of 60 are more prone to heart disease as compared to female.



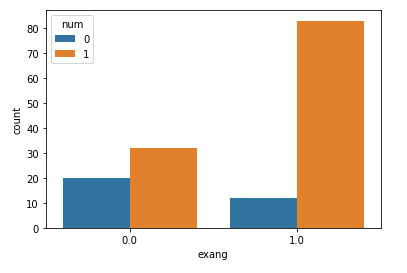
Here we have count plot of “fbs” with respect to target column. The plot tells us that “fbs” with type 0 are more prone to heart disease as compared to the type 1 patients.



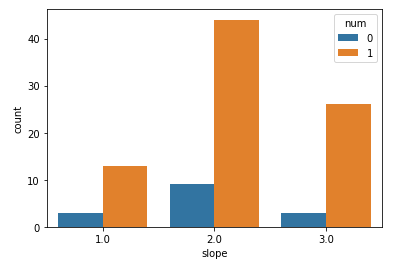
We have made count plot for “restecg” with respect to target. The plot tells us that patients with restecg type 0 and 1 are more prone to heart disease as compared to the type 2 patients.



Here we have count plot for “exang” with respect to target. There are two types in “exang” columns 0 and 1. The plot tells us that patients with type 1 “exang” are more prone to heart diseases.



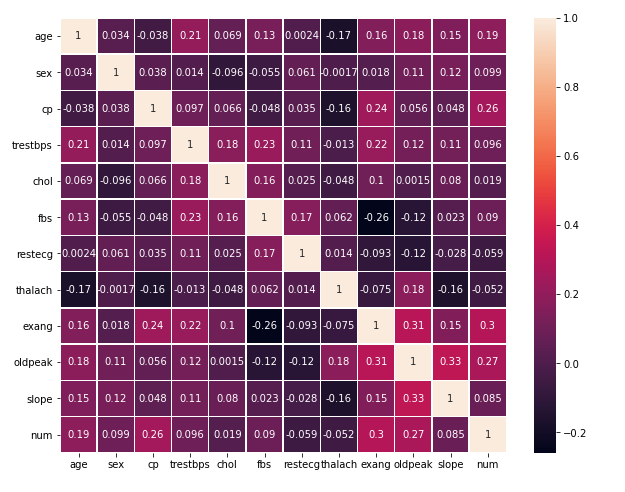
We have plotted count plot for “slope” column. The “slope” is of 3 types. The plot tells us that patients with type “2” slope are suffering more from heart disease as compared to other types.



Now we have done the multi variate analysis using the correlation. We fid the correlation us df.corr() method and using heatmap we plot the correlation table to see how attributes are correlated to each other.

The observations of the correlation are as follows: -

1. There is good positive correlation between “exang” and “num” column i.e 0.3.
2. There is positive correlation between “oldpeak” and “num” column i.e 0.27.
3. There is positive correlation between “cp” and “num” i.e 0.26.
4. There is good positive correlation between “slope” and “oldpeak” column 0.33.



1. There is positive correlation between “exang” and “oldpeak” column i.e 0.31.
2. In the columns there is weak positive correlation with “num” columns – “sex”, “testfbs”, “chol”, “fbs”, “slope”.
3. In the columns there is weak negative correlation with “num” columns – “restecg”, “thalach”.

## Pre-processing Pipeline:

In this section, we undertake data pre-processing steps to prepare the datasets for Machine Learning algorithm implementation. We can do encoding, feature scaling, split the train test data, handle the skewness and then go for the machine learning algorithms. We check for the skewness of the dataset so we can normalize it. We obtain z score of dataset o remove the outliers which can affect our models.

Encoding:

We have mapped the target variable in 0 and 1 type.

We have saved input features in “x” and target in the “y” variables.

Feature Scaling:

We can do feature scaling with standard Scaler or min max scaler which will standardise the input features. With help of Standard scaler, we will convert the attributes from the range 0 to 1 and in min max scaler we can convert the attributes from the range 0 to n.

Here we have used the standard Scaler which will convert the data from 0 to 1.

Train Test split:

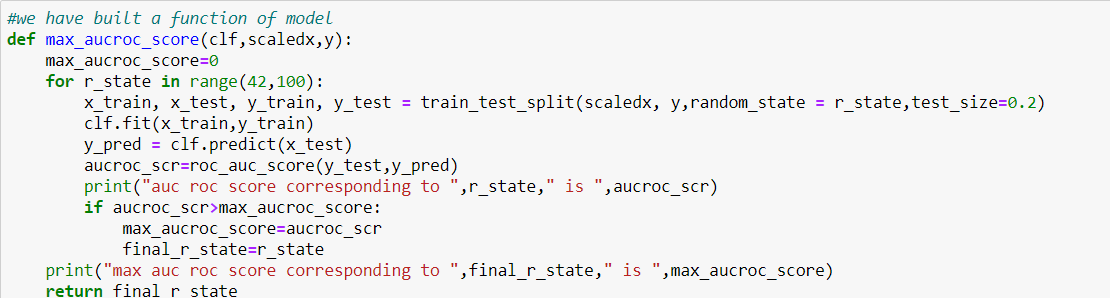
Prior to implementation or applying machine learning algorithms, me must break the data into training and testing data frame from our original dataset. Here we have split the data into 80:20 ratio i.e 80% for training purpose and 20% for testing purpose.

## *Machine Learning Algorithms*:

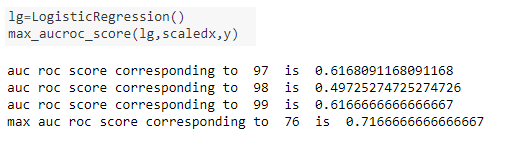
There are many machine learning algorithms available in the scikit leran library. We have used four types of algorithms i.e Logistic Regression, Decision tree Algorithm, KNN, SVC. As our target column is imbalanced so we have used roc auc score as our metrics.

We have also used metrics like accuracy score, roc auc score, confusion matrix , classification report from the sklearn.metrics .

First, we have a defined a function so it is simplified for calling the different algorithms by just calling the name of the function and passing the “x” and “y” in it.



1. Logistic Regression:

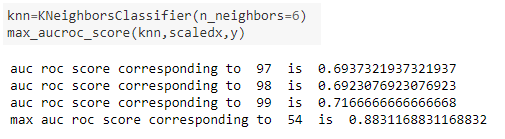


We have got max roc auc score for logistic regression: 71.66667

Mean cross val score for logistic classifier: 0.7206374

standard deviation in cross val score for logistic classifier: 0.0664

1. KNN

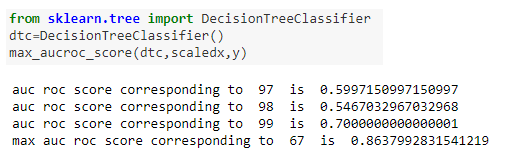


We have got max roc auc score for KNN: 88.311688

Mean cross val score for KNN classifier: 0.76365308

Standard deviation in cross val score for KNN : 0.08226424

1. DTC

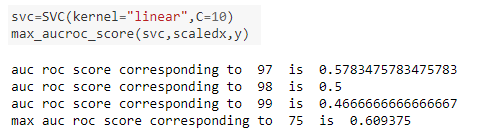


We have got max roc auc score for DTC is: 86.379928

Mean cross val score of DTC is: 0.6221076

Standard deviation in cross val score for DTC is: 0.06000385

1. SVC



We have got max roc auc score for SVC is: 60.9375

Mean cross val score of SVC is: 66.03

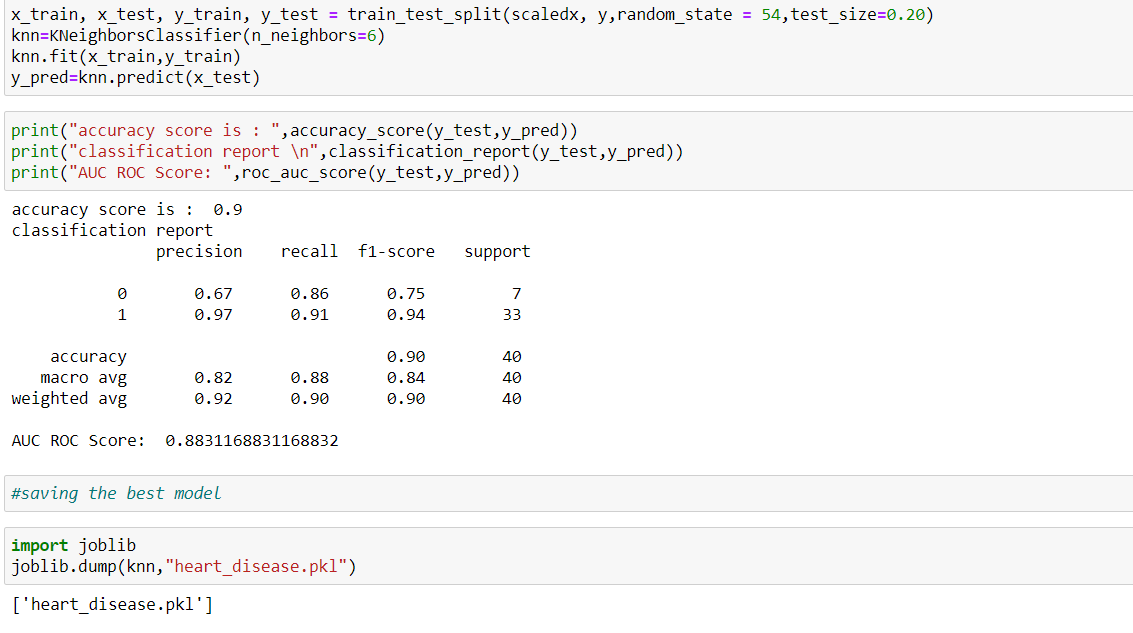
Standard deviation in cross val score for SVC is: 0.05615555

## Conclusion:

From the above algorithms we have chosen the K nearest neighbors (KNN) classifier as our best model. Before saving it, we have done the hyper parameter tuning of KNN to check the overfitting and underfitting of the data.

Then we got these as the KNN classifier best parameters as best random state as 54 and n\_neighbors as 6 which has helped in improving the accuracy score of the model. The

accuracy score, classification report and auc roc score is shown as below.



The max roc auc score for KNN classifier we got is 88.311 and **Accuracy score** of the KNN model is **90.00%**. then we have saved the model using joblib as “heart\_disease.pkl” as pickle file. We can load the pickle file and predict the future data with the help of KNN algorithm whenever we require it in the future.