Report on Project-2 (Heart_dataset)

- Importing Libraries

Step1: First of all I had import all the libraries which are required for our model like pandas, numpy, matplotlib, seaborn , scipy.stats.

- Reading And Understanding Data

Step2: I had created a variable as 'df' in which I stored the dataset which is in csv format. Then after that I had made copy of my dataset so that if my original data gets lots then also I have a backup dataset.

Step3: By applying data.head I checked the data. By shape function I got to know that data have 303 rows and 14 columns.

Step4: By using countplot function available in seaborn library I observed that the dependent column is nearly of same size.

Step5: In this step by using duplicated function I had checked if there is any duplicate row in our dataset or not. Only one row is duplicate so by using drop_duplicates function I droped that particular row. Then I again checked for duplicates and now I found none.

Step6: Indexing , datatype of columns and are their any nan values in dataset, memory usage were found by applying info function.

Step7: To check mean, median, minimum, maximum , etc I had used describe function , after that by corr function I had observed the correlation of each columns.

- Checking Outliers

Step8: In this step by using subplot function I had plot boxplots of four columns i.e age, cp, trtbps, chol , figsize is used so that plots don't get overlapped. From this step I had observed that trtbps and chol has outliers in it. So I decided to replace outliers by mean of that column.

Step9: In this this step I take ls variable in which I stored that values which are greater that 0.97 percentile. From boxplot I observed that

approximately above 170 are outliers in column ‘trtbps’. Now to replace outliers with mean I had used lambda function. For that I had created a list in which I stored all rows of trtbps column and then I simply passed it to original dataset. And then when I saw boxplot there were no outliers.

Step10: Then I did the same procedure for ‘chol’ column.

Step11: By using subplot I had plot boxplots of columns ‘thall’, ‘restecg’, ‘thalachh’, ‘oldpeak’ from that I observed there are very less outliers in ‘thall’ and ‘thalachh’ so I removed that outliers. In both columns outliers are below the lower limit.

Step11: From quantile I had find the values which are below 0.01 percentage. And then from boxplot I found the exact value that is outlier and at the end I removed that value from dataset.

Step12: After that I checked outliers in column ‘oldpeak’. By quantile function I found the outliers which are very less. And that’s why I deleted that rows from dataset. Values which are outliers in ‘oldpeak’ column gets permanently deleted from dataset.

Step13: In this step I checked for a value counts of ‘caa’ column from that I observed very less records belongs to class 3 and 4 and that’s why I deleted that rows. I simply wrote a code that says where ever 3 and 4 class are just remove that rows. And now all my dataset is free from outliers.

- Visualization

Step14: In visualization first I plot a barplot on features output and age and take hue as sex and I ovserved that age of males is high for being suspect for heart deasease and for not.After that a bargraph for cholesterol and trtbps is almost same for male aswell as women.

Step15: I did some visualization on column in which I used countplot from that I observed the count of males and females belongs to 0 and

1. From that I observed large number of men belongs to 0 and nearly equal for 1.

Step16: Now I had go for the distribution of the data. For distribution I used histplot with kde so that I could get easily understood the data is normally distributed or not along with that I used Q-Q plot from which we could clearly see the distribution of data.

Step17: I plot the distribution for all the columns and they all comes nearly normally distributed.

Step18: But when I saw distribution of 'thalachh' I observed that the data is left skewed . To make it normally distributed I had used various transforms. Firstly I take square root transform but it not gets normally distributed then I go for log1p transform (log1p is used for 0 values in our columns) but then also I didn't get proper result. Then finally by using box-cox transform my column gets normally distributed.

Step19: Then I add that transformed column into my dataset and deleted the original column from dataset.

Step20: In column 'oldpeak' when I saw distribution I observed that column is highly right hand skewed. And it also had lots of zero values in it. As it is right skewed I used log1p transform but then also data not got properly distributed. So I goes for box-cox transformation but this transformation does not accept zero values and therefore I used yeojohnson transformation but it also not worked for column.

Step21: Then I checked the correlation of that column with output column and it shows negative correlation with 43 percent. So I didn't drop that column.

Step22: Again I checked for info and I observed that I need to reset the index. So I used reset_index function to reset index.

That's all with EDA process.

- Spliting a Data And Stadardizing a Data

Step23: Firstly I import standard scaler , train_test_split , accuracy score, classification report.

Step24: Then I split the data into x and y in which in x except output all columns and in y only output column.

Step25: After that I split the data into xtrain, xtest, ytrain, ytest, in that train size was given about 77 percent. And then I scaled the data by using Standard scaler in which mean is 0 and standard deviation is 1.

- Applying Algorithms
- Logistic Regression

Step26: After scaling down the data I suppose to algorithms for training and testing. First I used Logistic Regression, for that from sklearn import Logistic Regression then created a object. After that I fit the data in which I took scaled data in it. After fitting the model score comes 88 percent. By using heatmap I observed by confusion matrix that where my model get confused. And after that print the classification report.

Step27: After prediction the accuracy of model comes to 87.5 percent.

	precision	recall	f1-score	support
1	0.90	0.90	0.90	42
2	0.00	0.00	0.00	0
3	0.00	0.00	0.00	0
micro avg	0.90	0.90	0.90	42
macro avg	0.30	0.30	0.30	42
weighted avg	0.90	0.90	0.90	42

Step28: Then I did hyperparameter tunning where I train and test my model on l2 regularization where I observed no effect on accuracy.

- Random Forest

Step29: Then after that I used Random Forest Classifier whose training score comes 100 percent and testing score comes 85 percent. From this we conclude that model get overfit.

	precision	recall	f1-score	support
0	0.86	0.79	0.83	24
1	0.88	0.93	0.90	40
accuracy			0.88	64
macro avg	0.87	0.86	0.86	64
weighted avg	0.87	0.88	0.87	64

- Support Vector Machine

Step30: Support vector machine had given me better result. Training score was 92 percent and testing score was 90 percent.

	precision	recall	f1-score	support
0	0.77	0.89	0.83	19
1	0.95	0.89	0.92	45
accuracy			0.89	64
macro avg	0.86	0.89	0.87	64
weighted avg	0.90	0.89	0.89	64

- KNN

KNN gives me accuracy of 88 percent.

	precision	recall	f1-score	support
0	0.77	0.85	0.81	20
1	0.93	0.89	0.91	44
accuracy			0.88	64
macro avg	0.85	0.87	0.86	64
weighted avg	0.88	0.88	0.88	64

Step31: By using Principle Component Analysis I got a accuracy of 80 percent. At the end I tried Stacking Algorithm in which I took random forest, knn, logistic regression as a estimators and SVM as a final estimator and by using this accuracy reached to 89 percent.

Best accuracy for this dataset comes from stacking and SVM aglorithms which is 89 percent.