MA515 Project Report
On
Exploratory Data Analysis

Gattadi Vivek (2019MCB1217)
Pre-Final Year , B.Tech. Mathematics and Computing,
IIT Ropar

A project under MA515 (Foundations of Data Science) course

Submitted to Dr. Arun Kumar, Department of Mathematics,
IIT Ropar

1. Problem Statement:

To do exploratory data analysis on the data. Use logistic regression and LDA to predict whether a given suburb has a crime rate above or below the median. Compare the findings from different methods.

2. Data Description:

- ➔ The Boston dataset was allocated to me. The task was to classify the suburb as whether the crime rate is higher or lower than the median.
- ➔ The dataset has a shape of (506,14) to which we have added an additional column for classifying whether the crime rate is higher ('True') or lower ('False') than the median. Hence the shape of the data becomes (506,15).
- ➔ The shape indicates that we have 14 features, 1 target column and 506 data points.

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv	target
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0	False
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6	False
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7	False
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4	False
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2	False

In the above 'target' is the new column added to the dataset.

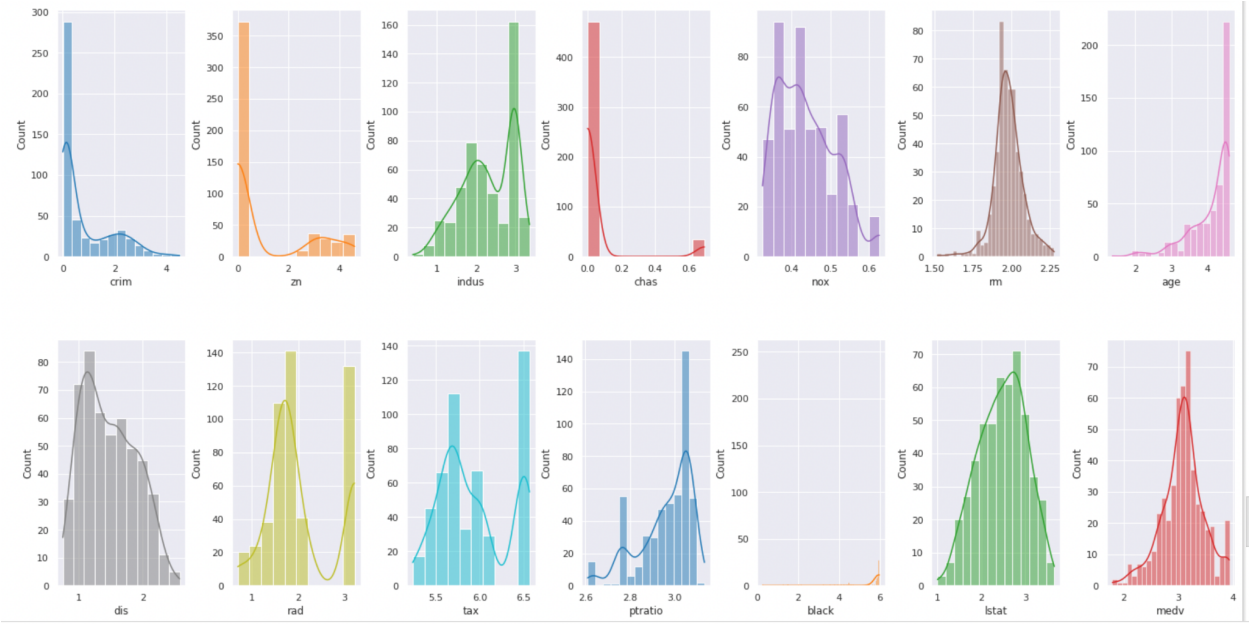
3. Exploratory Data Analysis:

1. I have described the data using data.describe().

```
[ ] data.describe()
```

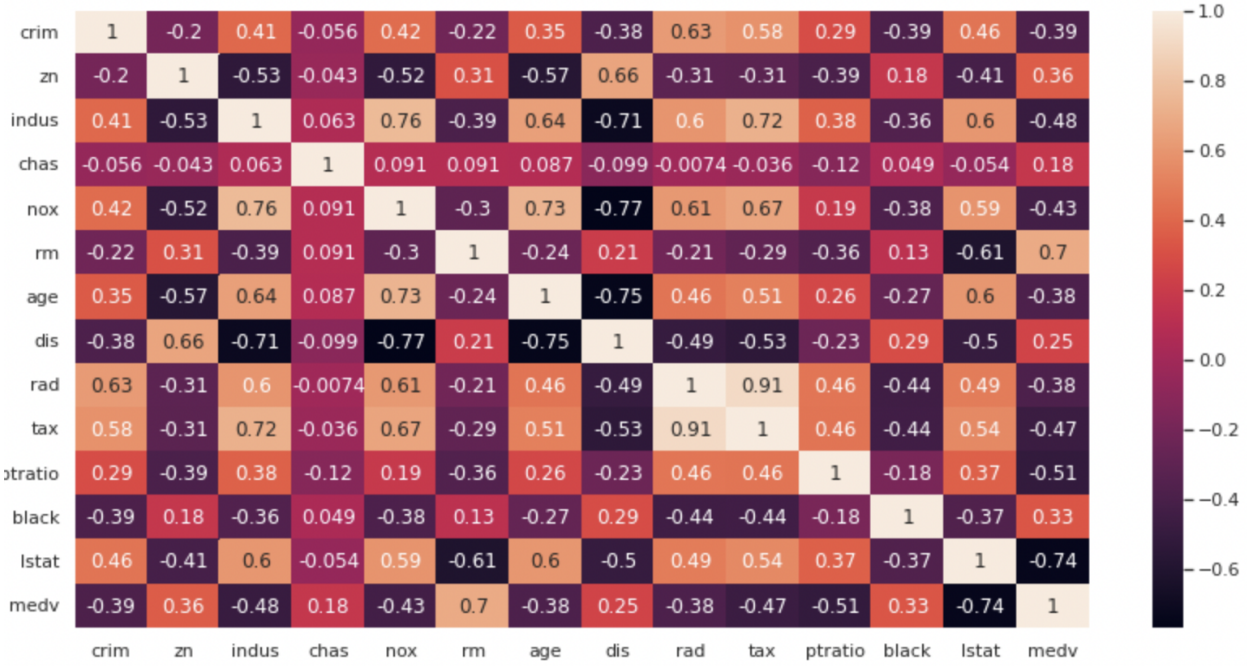
	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	3.795043	9.549407	408.237154	18.455534	356.674032	12.653063	22.532806
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2.105710	8.707259	168.537116	2.164946	91.294864	7.141062	9.197104
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187.000000	12.600000	0.320000	1.730000	5.000000
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2.100175	4.000000	279.000000	17.400000	375.377500	6.950000	17.025000
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3.207450	5.000000	330.000000	19.050000	391.440000	11.360000	21.200000
75%	3.677082	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5.188425	24.000000	666.000000	20.200000	396.225000	16.955000	25.000000
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711.000000	22.000000	396.900000	37.970000	50.000000

2. Also I have plotted a histogram for each feature to understand the distribution of each feature.



3. From the histogram plot we observe that 3 are left skewed (crim, zn, chas), 2 are right skewed (age, black) and 2 are normally distributed (rm, lstat).

4. Later we have also plotted the correlation matrix between all the features using data.corr().



5. After this I have taken our input matrix as X and target variable as y.

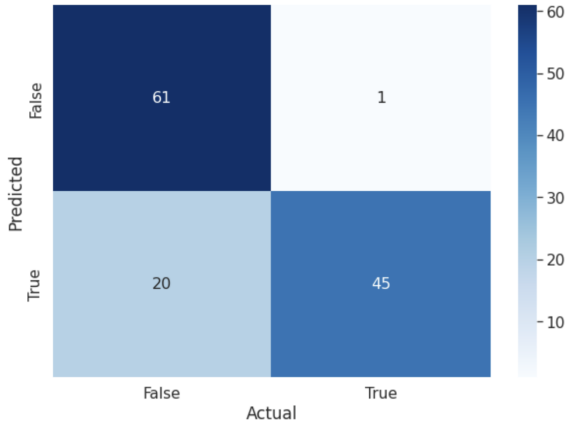
4. Split:

1. Here I have splitted the dataset into a training set and test set.
2. 75% of the data points are in the training set and remaining 25% in the test. This will be a reasonable split since the dataset consists of a very less number of data points.

5. Train-Test:

1. Now, I have trained our model using the LDA() classifier and tested our data on the test set. I got an accuracy of around 83.46%.

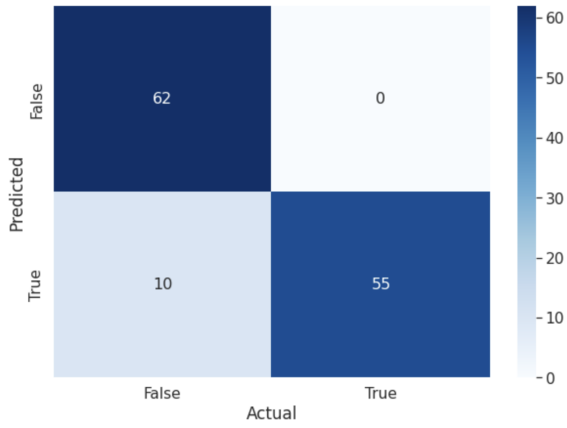
   The confusion matrix which have obtained is:



- The accuracy for predicting yes is 45/46 = 97.82%
- The accuracy for predicting no is 61/81 = 75.30%

2. Later I trained the model on the LogisticRegression() classifier and tested our data on the test set and got an accuracy around 92.12%.

   The confusion matrix which have obtained is:



- The accuracy for predicting no is 55/55 = 100%
- The accuracy for predicting no is 62/72 = 86.11%

Note:

➔ While applying Logistic Regression, it was observed that the convergence was not taking place unless the number of iterations crossed a threshold value.

➔ The problem got solved in two ways:

1. Increase the number of iterations. In this case it was 2500.
2. Scale/Normalize the data.

6. K-Fold Cross Validation:

Since there will be variability in the accuracy of the dataset, I have used the K-Fold Cross Validation resampling method and calculated the accuracy.

1. I got 80.33% accuracy for the LDA and
2. 84.27% accuracy for Logistic Regression.

7. Comparison of LDA Vs Logistic Regression:

1. From the above K-Fold accuracy, we can see that Logistic Regression is a better classifier than LDA here. The main reason behind it is LDA assumes that the observations are drawn from a Gaussian distribution with a common covariance matrix in each class, and so can provide some improvements over logistic regression when this assumption approximately holds. Conversely, **logistic regression** can outperform LDA if these Gaussian assumptions are not met. From the above histograms plots, we can observe that only 2 features approximately follow the normal distribution and remaining don't.

2. Hence the logistic regression performs better in overall accuracy. Also, we see that both of them show very good results in predicting YES but logistic regression performs far better when it comes to predicting NO due to which it's overall accuracy increases.