DELHI TECHNOLOGICAL UNIVERSITY

Shahbad Daulatpur, Bawana Road,Delhi 110042

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Project Report : Housing Price Prediction Using Linear Regression

Subject code : AI505

Submitted To:

Dr Anil Singh Parihar

Submitted By:

Gautam Aggarwal

M.Tech AI 1ST Year

Roll No-24/AFI/02

TABLE OF CONTENTS

ABSTRACT

This report investigates the prediction of housing values in Boston suburbs using linear regression, with a focus on the median value of owner-occupied homes as the target variable. The analysis utilizes the Boston Housing Dataset, which comprises 506 observations and 14 features, including crime rate, zoning proportion, non-retail business proportion, and various other socio-economic and environmental indicators. The primary objective is to develop a linear regression model that accurately estimates housing values based on these predictors. The study involves data preprocessing, exploratory data analysis, and the application of linear regression techniques. The model's performance is evaluated using the Root Mean Squared Error (RMSE), which measures the accuracy of predictions.

INTRODUCTION

Predicting housing prices is a complex and essential task for various stakeholders, including homeowners, investors, and policymakers. Housing prices are driven by a combination of factors, ranging from the local crime rate and proximity to employment centers to environmental conditions and neighborhood characteristics. Understanding the influence of these factors is key to making informed decisions in the real estate market.

In this project, we aim to build a predictive model using the Boston Housing dataset, which provides a rich set of features related to housing in Boston's suburbs. By applying linear regression, a widely-used method for modeling relationships between variables, we will estimate the median value of homes (`medv`) based on predictors such as the number of rooms, crime rate, tax rates, and distance to important urban hubs. Linear regression is well-suited for this analysis because it allows for easy interpretation of coefficients, providing a clear understanding of how each feature contributes to the prediction of housing prices.

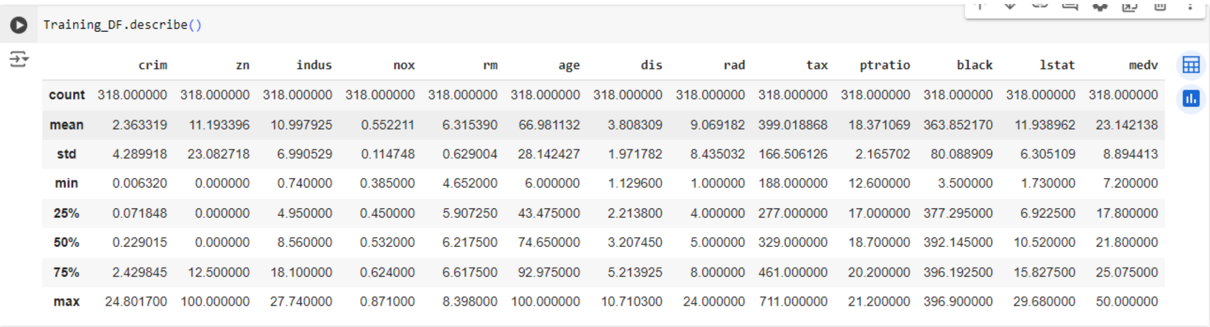
Our objective is twofold: to develop a model capable of accurately predicting home values and to uncover insights into the key drivers of housing prices in Boston. This analysis will highlight which variables have the greatest influence on property values, providing valuable guidance for market participants and future research in the field of real estate economics.

DATA OVERVIEW

The Boston Housing dataset contains information about housing in the suburbs of Boston, with 506 samples and 13 features that describe characteristics of house and various socioeconomic and geographical characteristics of the area. The target variable (medv) represents the median value of owner-occupied homes, which we aim to predict. The dataset includes predictors such as crime rate, property tax rates, and the average number of rooms per dwelling, which serve as inputs for the model.

Here is a summary of the features in the dataset:

1) **crim**: Per capita crime rate by town.
2) **zn**: Proportion of residential land zoned for lots over 25,000 sq. ft.
3) **indus**: Proportion of non-retail business acres per town.
4) **chas**: Charles River dummy variable (1 if tract bounds river, 0 otherwise).
5) **nox**: Concentration of nitrogen oxides (parts per 10 million).
6) **rm**: Average number of rooms per dwelling.
7) **age**: Proportion of owner-occupied units built before 1940.
8) **dis**: Weighted distances to five Boston employment centers.
9) **rad**: Index of accessibility to radial highways.
10) **tax**: Full-value property tax rate per $10,000.
11) **ptratio**: Pupil-teacher ratio by town.
12) **black**: 1000(Bk - 0.63)^2 where Bk is the proportion of the Black population by town.
13) **lstat**: Percentage of lower status of the population.
14) **medv**: Median value of owner-occupied homes (target variable in $1000s).

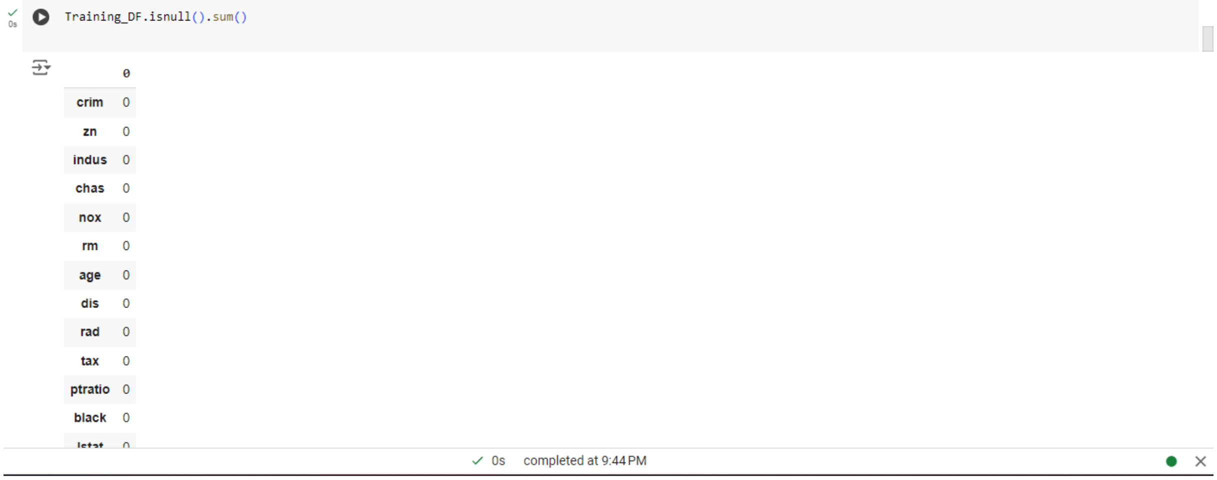


```
Training_DF.describe()
```

	crim	zn	indus	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
count	318.000000	318.000000	318.000000	318.000000	318.000000	318.000000	318.000000	318.000000	318.000000	318.000000	318.000000	318.000000	318.000000
mean	2.363319	11.193396	10.997925	0.552211	6.315390	66.981132	3.808309	9.069182	399.018868	18.371069	363.852170	11.938962	23.142138
std	4.289918	23.082718	6.990529	0.114748	0.629004	28.142427	1.971782	8.435032	166.506126	2.165702	80.088909	6.305109	8.894413
min	0.006320	0.000000	0.740000	0.385000	4.652000	6.000000	1.129600	1.000000	188.000000	12.600000	3.500000	1.730000	7.200000
25%	0.071848	0.000000	4.950000	0.450000	5.907250	43.475000	2.213800	4.000000	277.000000	17.000000	377.295000	6.922500	17.800000
50%	0.229015	0.000000	8.560000	0.532000	6.217500	74.650000	3.207450	5.000000	329.000000	18.700000	392.145000	10.520000	21.800000
75%	2.429845	12.500000	18.100000	0.624000	6.617500	92.975000	5.213925	8.000000	461.000000	20.200000	396.192500	15.827500	25.075000
max	24.801700	100.000000	27.740000	0.871000	8.398000	100.000000	10.710300	24.000000	711.000000	21.200000	396.900000	29.680000	50.000000

Some key observations are :

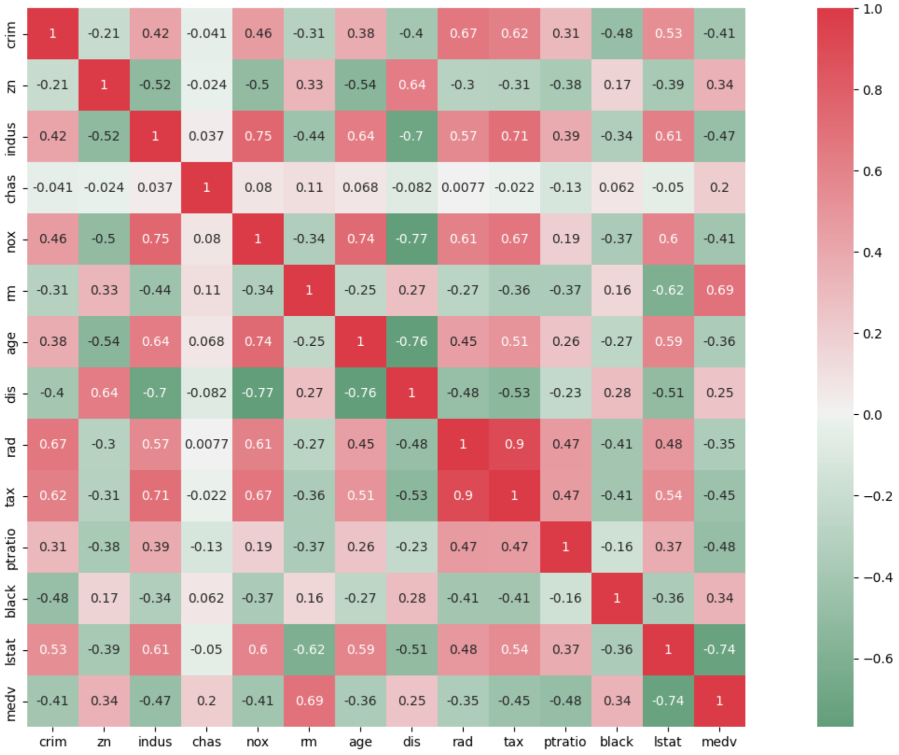
1) **Missing Values**: There are no missing values in the dataset.



```
Training_DF.isnull().sum()
```

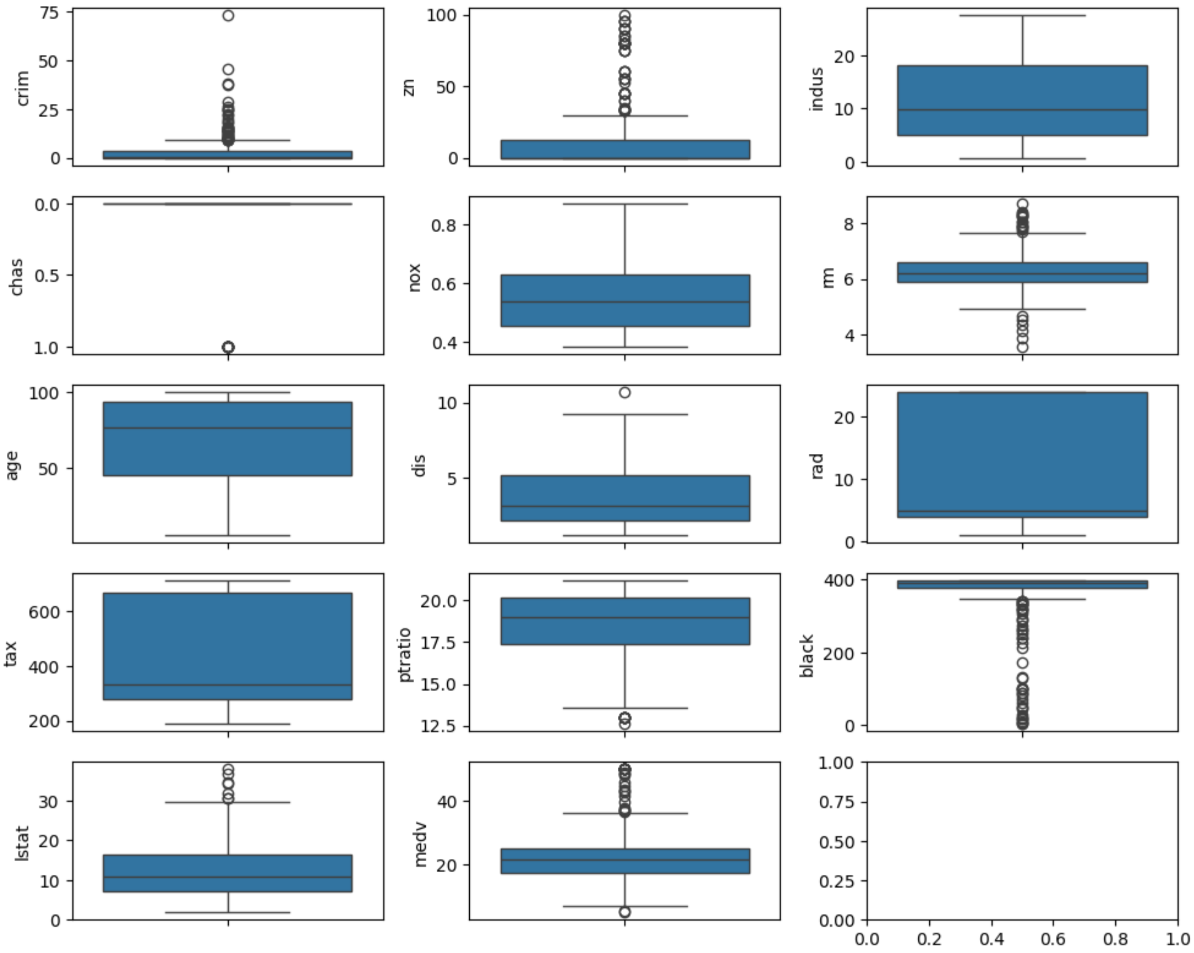
```
         0
crim     0
zn       0
indus    0
chas     0
nox      0
rm       0
age      0
dis      0
rad      0
tax      0
ptratio  0
black    0
lstat    0
```

2) **Correlations**: Initial correlation analysis reveals strong relationships between certain variables and housing prices. Notably, ***rm*** (number of rooms) has a strong positive correlation with ***medv***, indicating that homes with more rooms tend to be more expensive. Conversely, ***lstat*** (percentage of lower-status population) shows a strong negative correlation with ***medv***, suggesting that areas with a higher proportion of lower-status residents tend to have lower home values.

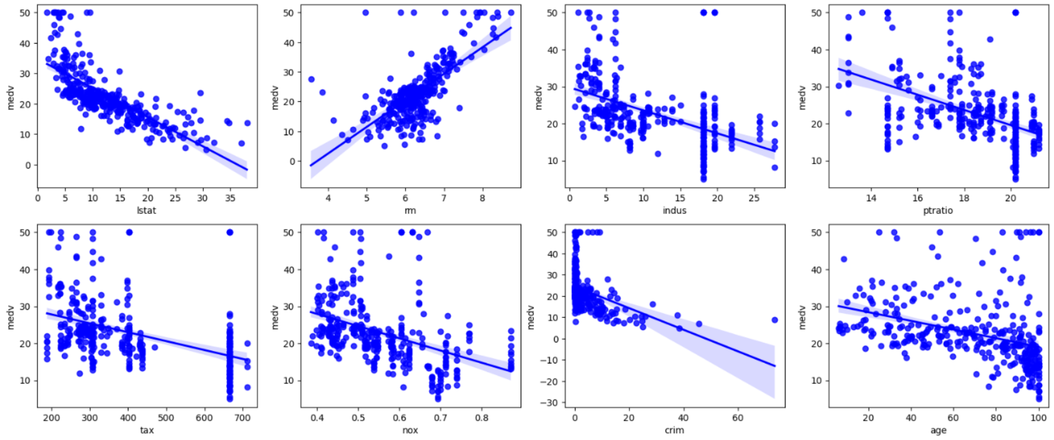


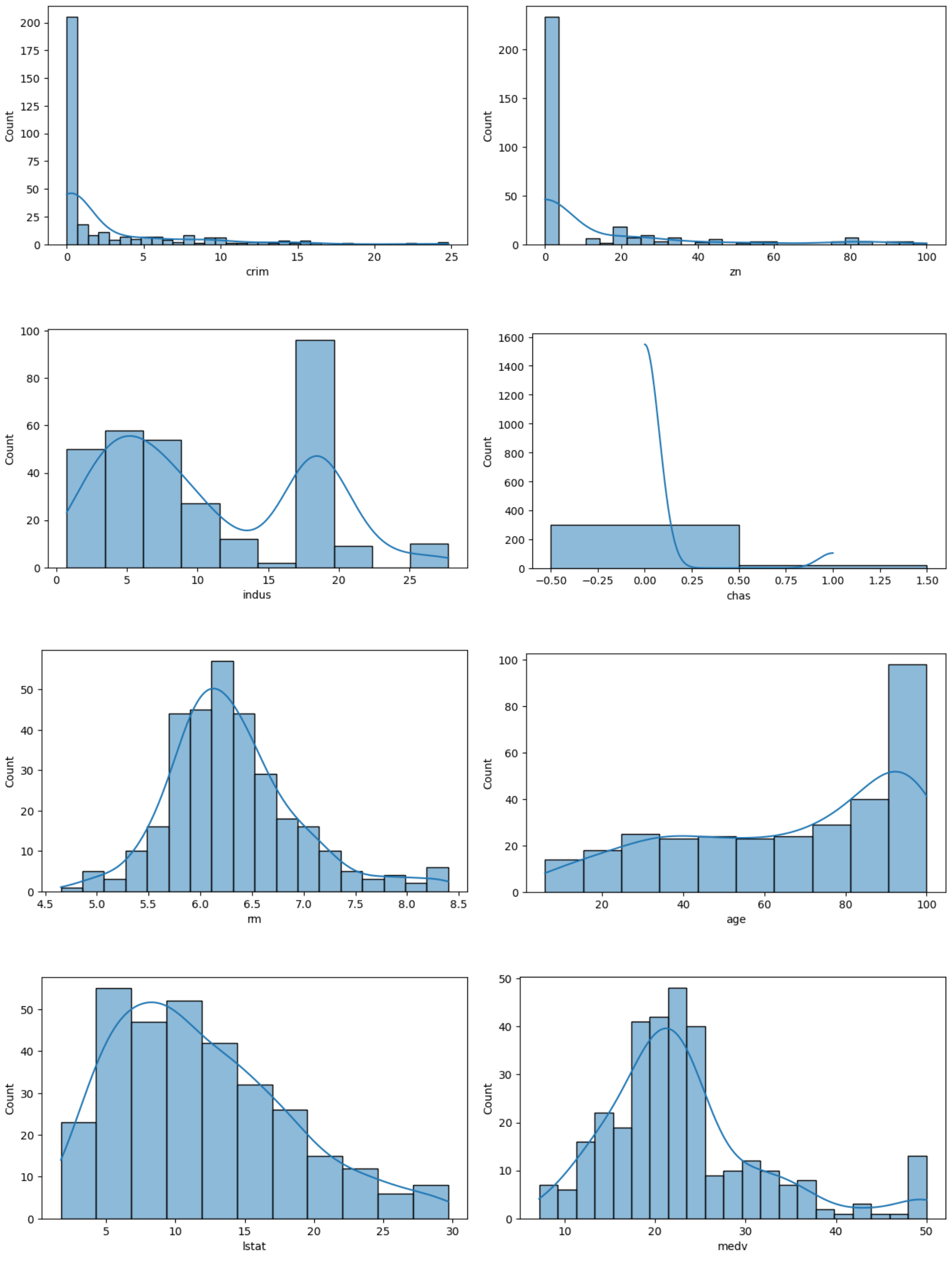
	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
crim	1	-0.21	0.42	-0.041	0.46	-0.31	0.38	-0.4	0.67	0.62	0.31	-0.48	0.53	-0.41
zn	-0.21	1	-0.52	-0.024	-0.5	0.33	-0.54	0.64	-0.3	-0.31	-0.38	0.17	-0.39	0.34
indus	0.42	-0.52	1	0.037	0.75	-0.44	0.64	-0.7	0.57	0.71	0.39	-0.34	0.61	-0.47
chas	-0.041	-0.024	0.037	1	0.08	0.11	0.068	-0.082	0.0077	-0.022	-0.13	0.062	-0.05	0.2
nox	0.46	-0.5	0.75	0.08	1	-0.34	0.74	-0.77	0.61	0.67	0.19	-0.37	0.6	-0.41
rm	-0.31	0.33	-0.44	0.11	-0.34	1	-0.25	0.27	-0.27	-0.36	-0.37	0.16	-0.62	0.69
age	0.38	-0.54	0.64	0.068	0.74	-0.25	1	-0.76	0.45	0.51	0.26	-0.27	0.59	-0.36
dis	-0.4	0.64	-0.7	-0.082	-0.77	0.27	-0.76	1	-0.48	-0.53	-0.23	0.28	-0.51	0.25
rad	0.67	-0.3	0.57	0.0077	0.61	-0.27	0.45	-0.48	1	0.9	0.47	-0.41	0.48	-0.35
tax	0.62	-0.31	0.71	-0.022	0.67	-0.36	0.51	-0.53	0.9	1	0.47	-0.41	0.54	-0.45
ptratio	0.31	-0.38	0.39	-0.13	0.19	-0.37	0.26	-0.23	0.47	0.47	1	-0.16	0.37	-0.48
black	-0.48	0.17	-0.34	0.062	-0.37	0.16	-0.27	0.28	-0.41	-0.41	-0.16	1	-0.36	0.34
lstat	0.53	-0.39	0.61	-0.05	0.6	-0.62	0.59	-0.51	0.48	0.54	0.37	-0.36	1	-0.74
medv	-0.41	0.34	-0.47	0.2	-0.41	0.69	-0.36	0.25	-0.35	-0.45	-0.48	0.34	-0.74	1

3) **Outliers**: We identified potential outliers in variables such as ***crim*** (crime rate).

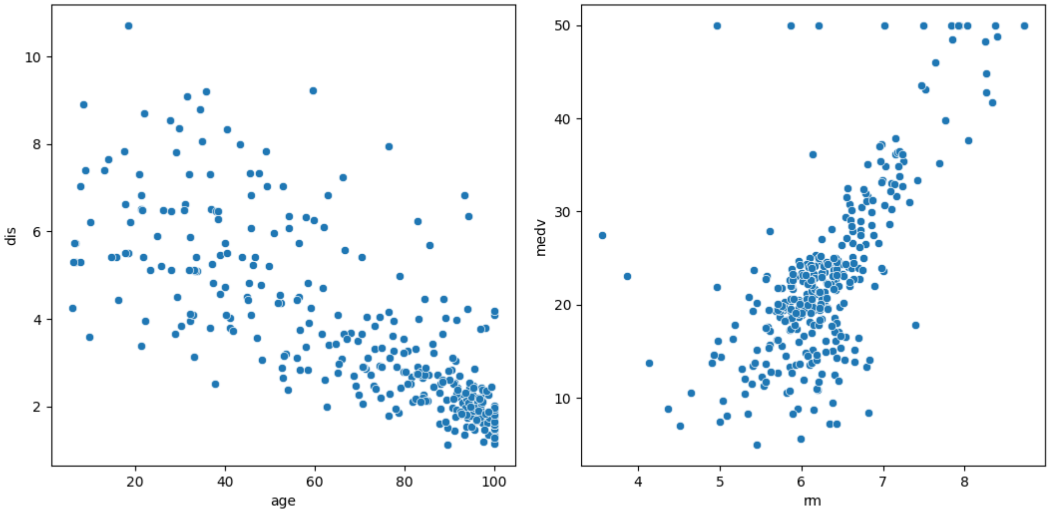


Some important visualisations are as follows:



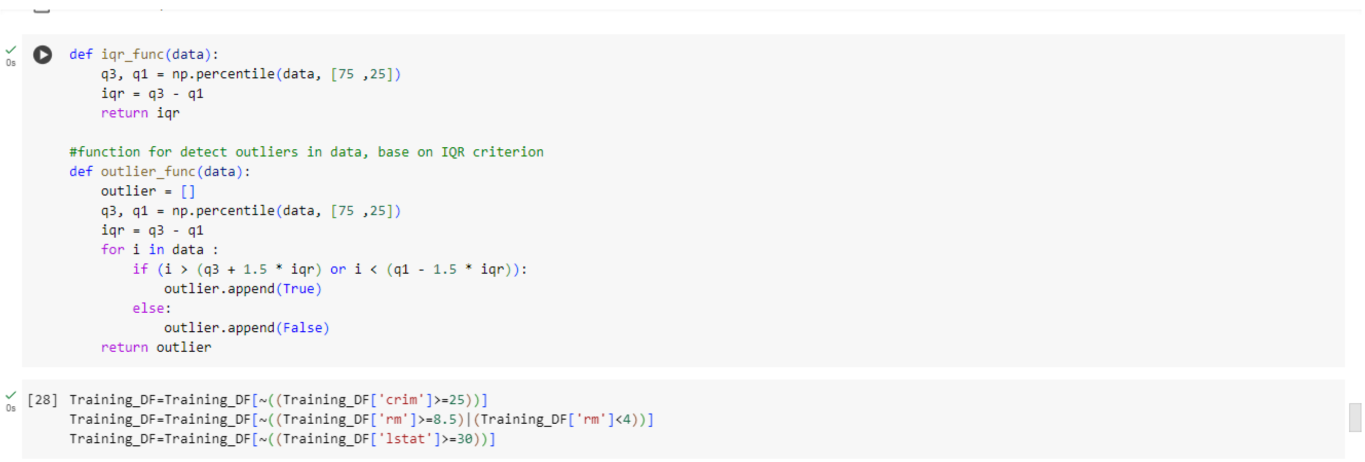


Count
0
25
50
75
100
125
150
175
200
0
5
10
15
20
25
crim
Count
0
50
100
150
200
0
20
40
60
80
100
zn
Count
0
20
40
60
80
100
0
5
10
15
20
25
indus
Count
0
200
400
600
800
1000
1200
1400
1600
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
chas
Count
0
10
20
30
40
50
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
rm
Count
0
20
40
60
80
100
20
40
60
80
100
age
Count
0
10
20
30
40
50
5
10
15
20
25
30
lstat
Count
0
10
20
30
40
50
10
20
30
40
50
medv


dis
10
8
6
4
2
20
40
60
80
100
age
medv
50
40
30
20
10
4
5
6
7
8
rm

METHODOLOGY

1) **Data Preprocessing :** There are no missing values in the dataset. We remove all outliers using function for calculating the IQR criterion using the code below :

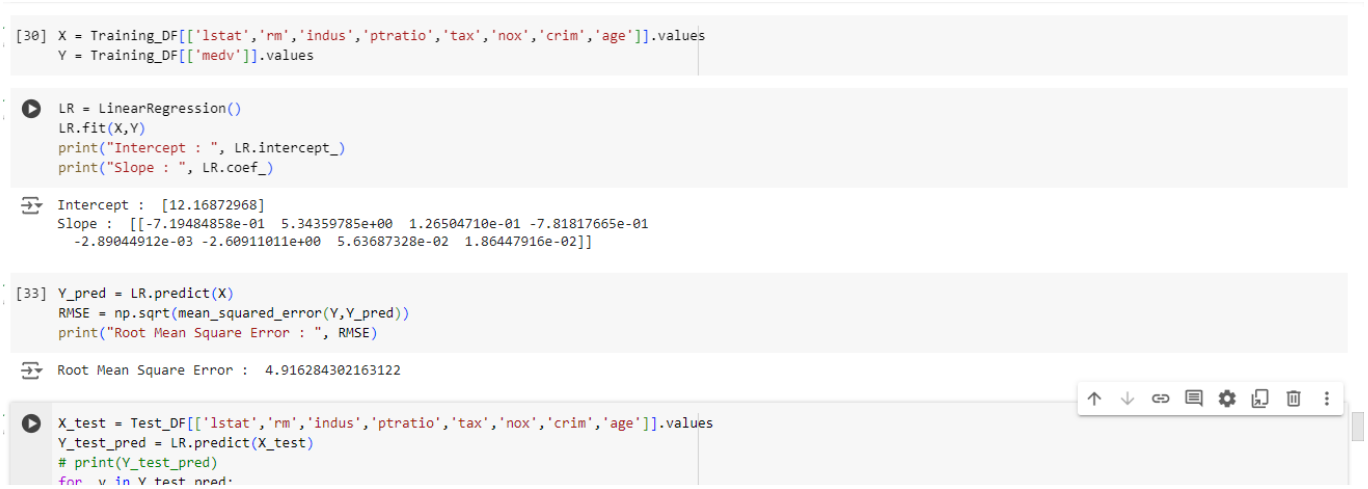


```
def iqr_func(data):
    q3, q1 = np.percentile(data, [75 ,25])
    iqr = q3 - q1
    return iqr

#function for detect outliers in data, base on IQR criterion
def outlier_func(data):
    outlier = []
    q3, q1 = np.percentile(data, [75 ,25])
    iqr = q3 - q1
    for i in data :
        if (i > (q3 + 1.5 * iqr) or i < (q1 - 1.5 * iqr)):
            outlier.append(True)
        else:
            outlier.append(False)
    return outlier
```

```
[28] Training_DF=Training_DF[~((Training_DF['crim']>=25))]
     Training_DF=Training_DF[~((Training_DF['rm']>=8.5)|(Training_DF['rm']<4))]
     Training_DF=Training_DF[~((Training_DF['lstat']>=30))]
```

2) **Feature Selection** : ***lstat, rm, indus, ptratio, tax, nox, crim, age*** are the features selected for model training as based on exploratory data analysis, they have strong correlation with ***medv***.



```
[30] X = Training_DF[['lstat','rm','indus','ptratio','tax','nox','crim','age']].values
     Y = Training_DF[['medv']].values
```

```
LR = LinearRegression()
LR.fit(X,Y)
print("Intercept : ", LR.intercept_)
print("Slope : ", LR.coef_)
```

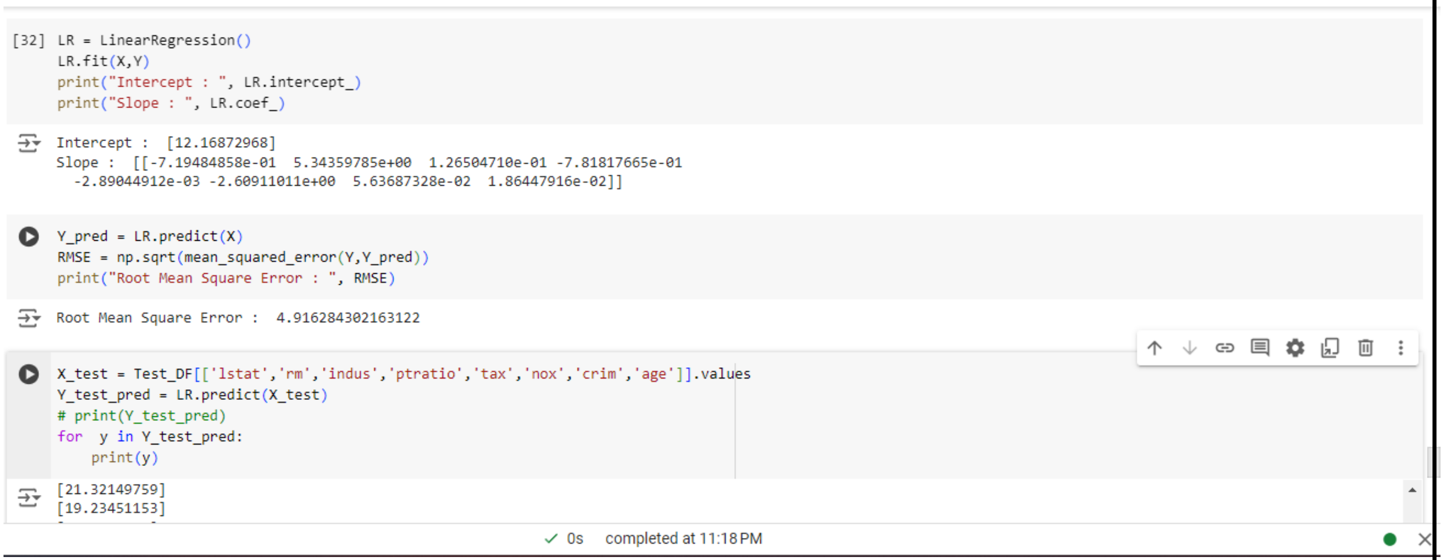
```
Intercept :  [12.16872968]
Slope :  [[-7.19484858e-01  5.34359785e+00  1.26504710e-01 -7.81817665e-01
  -2.89044912e-03 -2.60911011e+00  5.63687328e-02  1.86447916e-02]]
```

```
[33] Y_pred = LR.predict(X)
     RMSE = np.sqrt(mean_squared_error(Y,Y_pred))
     print("Root Mean Square Error : ", RMSE)
```

```
Root Mean Square Error :  4.916284302163122
```

```
X_test = Test_DF[['lstat','rm','indus','ptratio','tax','nox','crim','age']].values
Y_test_pred = LR.predict(X_test)
# print(Y_test_pred)
for  y in Y_test_pred:
```

3) **Linear Regression Model :** We used the linear regression model from Scikit-learn's ***LinearRegression()*** function. The model was trained on the training set using the ***fit()*** method, and predictions were made on the test set using the ***predict()*** method. The model's primary objective was to minimize the residual sum of squares between the actual and predicted values of ***medv***.



```
[32] LR = LinearRegression()
     LR.fit(X,Y)
     print("Intercept : ", LR.intercept_)
     print("Slope : ", LR.coef_)
```

```
Intercept :  [12.16872968]
Slope :  [[-7.19484858e-01  5.34359785e+00  1.26504710e-01 -7.81817665e-01
  -2.89044912e-03 -2.60911011e+00  5.63687328e-02  1.86447916e-02]]
```

```
Y_pred = LR.predict(X)
RMSE = np.sqrt(mean_squared_error(Y,Y_pred))
print("Root Mean Square Error : ", RMSE)
```

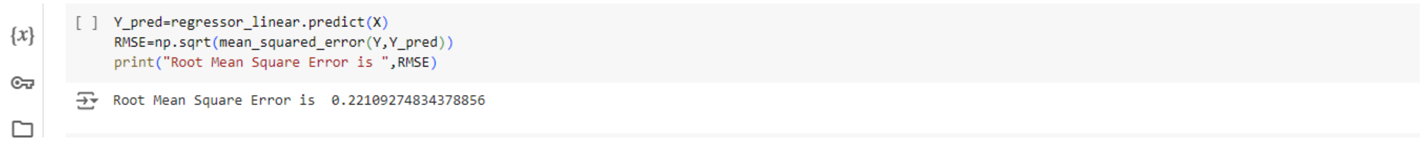
```
Root Mean Square Error :  4.916284302163122
```

```
X_test = Test_DF[['lstat','rm','indus','ptratio','tax','nox','crim','age']].values
Y_test_pred = LR.predict(X_test)
# print(Y_test_pred)
for  y in Y_test_pred:
    print(y)
```

```
[21.32149759]
[19.23451153]
```

✓ 0s completed at 11:18 PM

4) **Model Evaluation :** Evaluating the performance of a linear regression model is crucial to understanding its accuracy and reliability in predicting the target variable. In this project, the model was assessed using Root Mean Squared Error (RMSE). Our trained data had RMSE value of 0.2210.



```
[ ] Y_pred=regressor_linear.predict(X)
    RMSE=np.sqrt(mean_squared_error(Y,Y_pred))
    print("Root Mean Square Error is ",RMSE)
```

```
Root Mean Square Error is  0.22109274834378856
```

5) **Addressing Model Limitations :** While the linear regression model performed well, it showed limitations in capturing non-linear relationships. Future work may involve trying polynomial regression or adding interaction terms to better model the complex relationships in the data.

RESULTS AND DISCUSSION

The RMSE on the training set was 0.21, meaning the model's predictions were off by an average of $210 in the training data. These results suggest that the model performs well but is not perfect. The relatively small difference between the training and test performance indicates minimal overfitting, and the model is likely capturing the main relationships in the data.

The linear regression model highlighted certain features that play a significant role in predicting housing prices:

- **Number of Rooms (rm)**: As expected, the number of rooms per dwelling showed a strong positive correlation with the median home value. Homes with more rooms tend to be priced higher, reflecting their larger size and likely higher market appeal.
- **Lower-Status Population (lstat)**: The percentage of lower-status residents had a strong negative correlation with housing prices, indicating that areas with higher concentrations of lower-income households tend to have lower home values.
- **Pupil-Teacher Ratio (ptratio)**: The model found a slight negative relationship between the pupil-teacher ratio and home prices, suggesting that areas with lower pupil-teacher ratios, which typically reflect better school quality, have higher home values.

These results align with real-world intuition: homes in areas with more space, fewer lower-income residents, and better school quality are typically valued higher in the housing market.

CONCLUSION AND FUTURE SCOPE

The linear regression model successfully predicted Boston housing prices with a fair degree of accuracy. It provided valuable insights into the key factors driving housing prices, such as the number of rooms and the socio-economic status of the population. While the model performed well, there is potential for improvement by capturing non-linear relationships and using regularization techniques. These refinements could lead to more precise predictions and a deeper understanding of the complex factors influencing the housing market.

REFERENCES

1) **Harrison, D., & Rubinfeld, D. L. (1978).** *Hedonic housing prices and the demand for clean air*. Journal of Environmental Economics and Management, 5(1), 81–102.

2) **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., et al. (2011).** *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.