Predictive Analysis of Boston House Prices

Using Multiple Linear Regression

The Boston House-Price Data

Khusanov Javokhir

Table of Content

Introduction

The dataset is publicly available in kaggle.com and Carnegie Mellon University Dataset Archive.

Task

The main task is to build a model that can predict house prices based on different parameters (independent variables/factors/features/attributes).

Descriptive Statistics

Data Description

Training Data consists of 506 rows and 8 columns (attributes):

1. *CRIM*: crime rate per capita by town (Numerical)

2. *ZN*: proportion of residential land zoned for lots over 25000 sq. ft. (Numerical)

3. *RM*: average number of rooms per dwelling (Numerical)

4. *AGE*: proportion of owner-occupied units built prior to 1940 (Numerical)

5. *RAD*: index of accessibility to radial highways (Numerical)

6. *TAX*: full-value property-tax rate per 10000$ (Numerical)

7. *LSTAT*: % lower status of the population (Numerical)

8. *MEDV*: Median value of owner-occupied homes in $1000's

The response variable (y): *MEDV*

Regressors: *CRIM*, *ZN*, *RM*, *AGE*, *RAD*, *TAX*, *LSTAT*, *MEDV*

First 5 rows of the dataset:

	CRIM	ZN	RM	AGE	RAD	TAX	LSTAT	MEDV
0	0.00632	18.0	6.575	65.2	1	296	4.98	24.0
1	0.02731	0.0	6.421	78.9	2	242	9.14	21.6
2	0.02729	0.0	7.185	61.1	2	242	4.03	34.7
3	0.03237	0.0	6.998	45.8	3	222	2.94	33.4
4	0.06905	0.0	7.147	54.2	3	222	5.33	36.2

Figure 1

Descriptive Statistics for Independent and Dependent variables

Summary statistics of the dataset:

	CRIM	ZN	RM	AGE	RAD	TAX	LSTAT	MEDV
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	6.284634	68.574901	9.549407	408.237154	12.653063	22.532806
std	8.601545	23.322453	0.702617	28.148861	8.707259	168.537116	7.141062	9.197104
min	0.006320	0.000000	3.561000	2.900000	1.000000	187.000000	1.730000	5.000000
25%	0.082045	0.000000	5.885500	45.025000	4.000000	279.000000	6.950000	17.025000
50%	0.256510	0.000000	6.208500	77.500000	5.000000	330.000000	11.360000	21.200000
75%	3.677083	12.500000	6.623500	94.075000	24.000000	666.000000	16.955000	25.000000
max	88.976200	100.000000	8.780000	100.000000	24.000000	711.000000	37.970000	50.000000

Figure 2

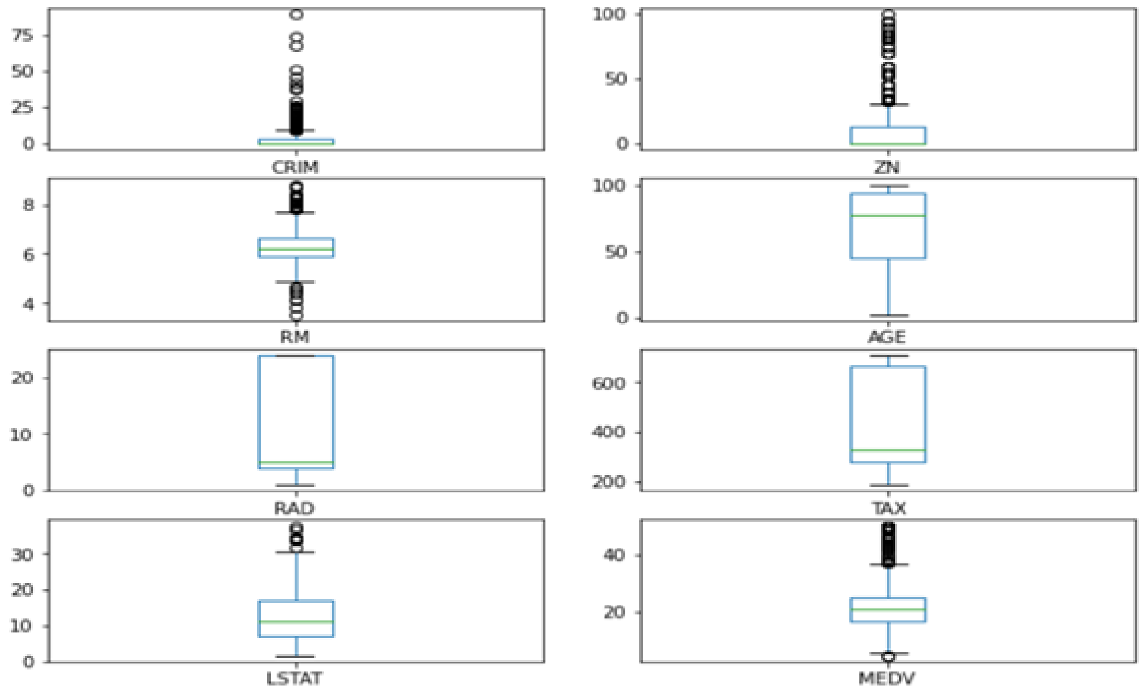

Figure 3

Figure 3 illustrates the boxplot for all 8 columns. From the first glance what we can observe is above and below of boxplots, these black circles which are outliers in those columns. All columns have mild outliers and extreme outliers, except columns AGE, RAD, and TAX that are outlier-free columns.

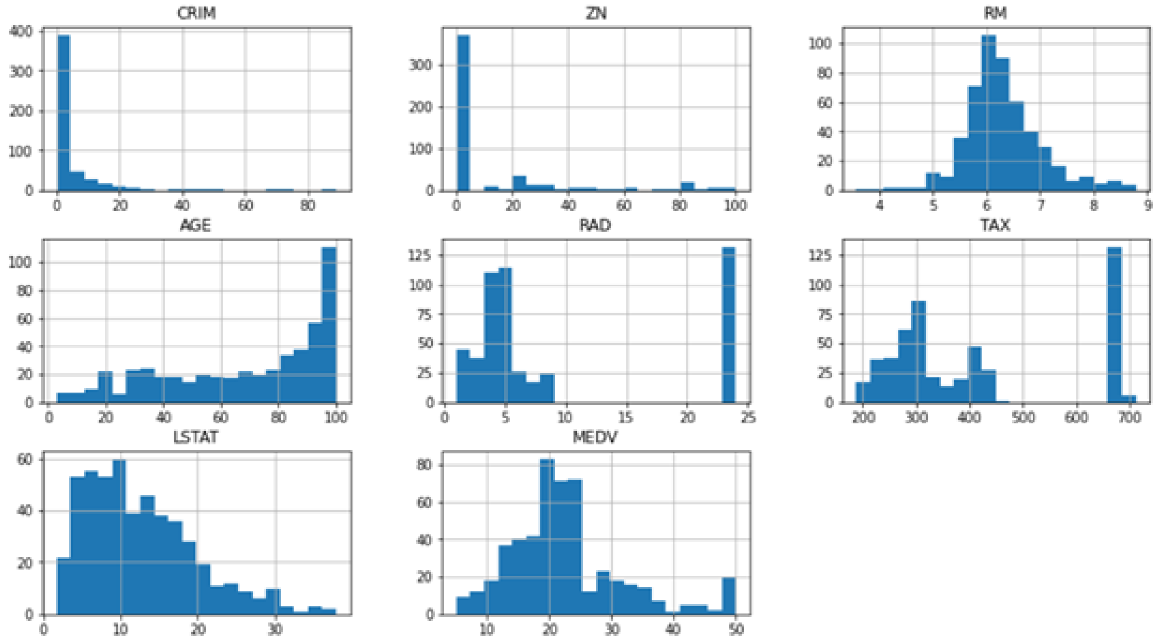

Figure 4

Figure 3 shows the distribution of the attributes using the histogram. At first glance, we can observe that CRIM, AGE, LSTAT columns are skewed where is RM columns normally distributed.

Graphs and fitted lines

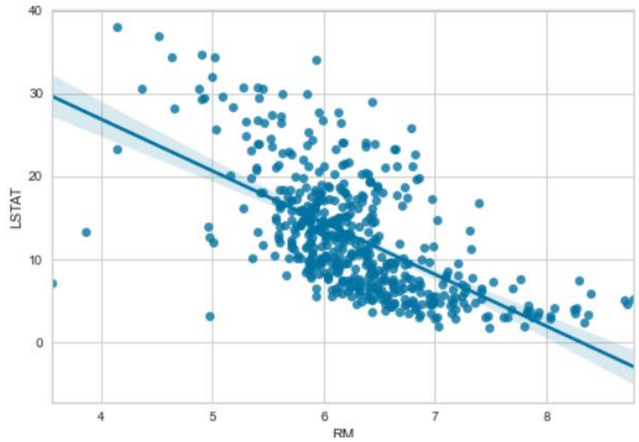

Figure 5

RM vs LSTAT
RM and LSTAT be moderately negatively correlated as the points seem to fall on a line. There is a possibility of a linear relationship.

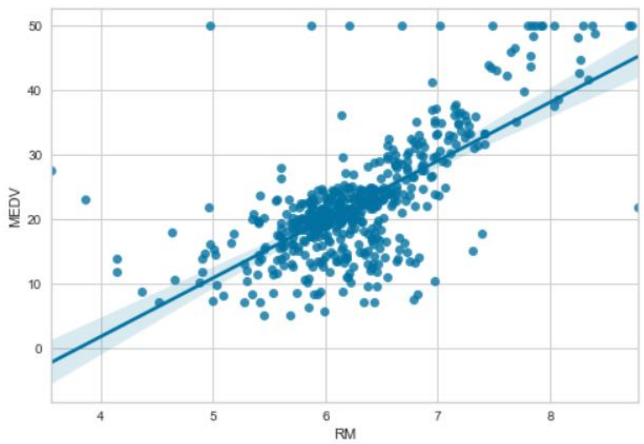

Figure 6

RM vs MEDV
RM and MEDV be moderately positively correlated as the points seem to fall on a line. There is a possibility of a linear relationship.

	CRIM	ZN	RM	AGE	RAD	TAX	LSTAT	MEDV
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	6.284634	68.574901	9.549407	408.237154	12.653063	22.532806
std	8.601545	23.322453	0.702617	28.148861	8.707259	168.537116	7.141062	9.197104
min	0.006320	0.000000	3.561000	2.900000	1.000000	187.000000	1.730000	5.000000
25%	0.082045	0.000000	5.885500	45.025000	4.000000	279.000000	6.950000	17.025000
50%	0.256510	0.000000	6.208500	77.500000	5.000000	330.000000	11.360000	21.200000
75%	3.677083	12.500000	6.623500	94.075000	24.000000	666.000000	16.955000	25.000000
max	88.976200	100.000000	8.780000	100.000000	24.000000	711.000000	37.970000	50.000000

Figure 7

Figure 7 illustrates the dataset's basic summary statistics including mean standard deviation.

	Number of missing values
CRIM	0
ZN	0
RM	0
AGE	0
RAD	0
TAX	0
LSTAT	0
MEDV	0

Figure 8

Figure 8 shows that we don't have any missing values.

Correlation chart

	Attribute Correlation with Target
CRIM	-0.39
ZN	0.36
RM	0.70
AGE	-0.38
RAD	-0.38
TAX	-0.47
LSTAT	-0.74
MEDV	1.00

Figure 9 represents correlation of all features with dependent variable

Figure 9

$$r = \frac{\Sigma(X-\overline{X})(Y-\overline{Y})}{\sqrt{\Sigma(X-\overline{X})^2}\sqrt{(Y-\overline{Y})^2}}$$

Where, $\overline{X}$ - mean of X variable
$\overline{Y}$ - mean of Y variable

Correlation coefficient (r) matrix of 8 numeric variables:

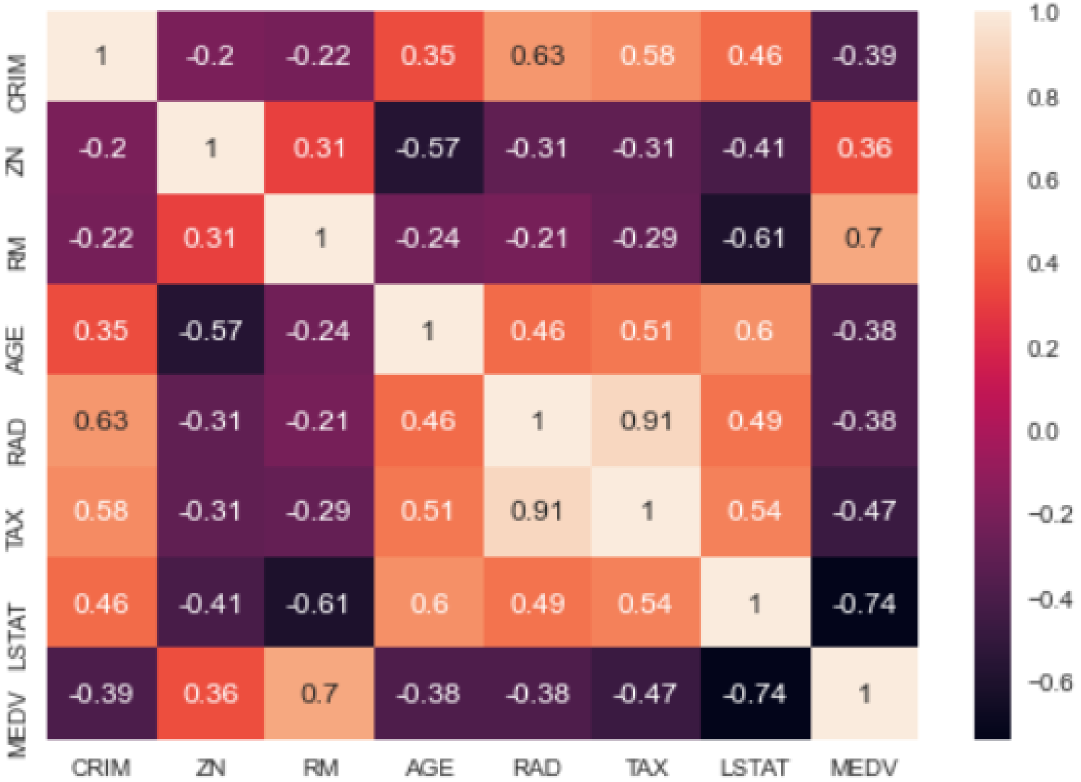

Figure 10

Note: Figure 10 depicts correalation between different feature. The darker purple the stronger negative relation, while the lighter the orange stronger the positive relation. Almost all columns correlated with response variable roughly same r value which is 0.4 while the RM and LSTAT columns correlated with target columns the highest r value around 0.

Multiple Linear Regression Model

Multiple linear regression prediction model for Y(MEDV) and X_i (CRIM, ZN, RM, AGE, RAD, TAX, LSTAT)

$$\boxed{\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \beta_6 X_6 + \beta_7 X_7}$$

Where,

$\hat{Y}$ is the predicted value of dependent variable Y.

X_i is the actual value of independent/explanatory variable:

$X_{1:}$ *CRIM*

X_2 : *ZN*

X_3 : *RM*

X_4 : *AGE*

X_5 : *RAD*

X_6 : *TAX*

X_8 : *LSTAT*

β_i is the regression coefficient of respective $X_{i\,.}$

This section is divided into 3 parts dedicated to –

1. Analyse the regression statistic table.

2. Hypothesis test to check model utility.

3. Regression coefficient table and final model.

3. Regression coefficient table and final model.

Resgression Statistics Table

The following table is the regression statistics table. R^2(coefficient of determination) is the most important factor in it.

Dep. Variable:	y	R-squared:	0.649
Model:	OLS	Adj. R-squared:	0.643
Method:	Least Squares	F-statistic:	104.6
Date:	Thu, 02 Sep 2021	Prob (F-statistic):	5.41e-86
Time:	00:03:54	Log-Likelihood:	-1248.9
No. Observations:	404	AIC:	2514.
Df Residuals:	396	BIC:	2546.
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	5.5801	3.748	1.489	0.137	-1.789	12.950
x1	-0.0872	0.038	-2.293	0.022	-0.162	-0.012
x2	0.0381	0.015	2.603	0.010	0.009	0.067
x3	4.1231	0.522	7.891	0.000	3.096	5.150
x4	0.0380	0.014	2.641	0.009	0.010	0.066
x5	0.1871	0.083	2.255	0.025	0.024	0.350
x6	-0.0137	0.004	-3.211	0.001	-0.022	-0.005
x7	-0.6152	0.063	-9.705	0.000	-0.740	-0.491

Omnibus:	155.268	Durbin-Watson:	2.002
Prob(Omnibus):	0.000	Jarque-Bera (JB):	650.038
Skew:	1.661	Prob(JB):	7.01e-142
Kurtosis:	8.251	Cond. No.	6.26e+03

Figure 11

Explanation of the terms in the table –

1. Multiple R - square root of R^2

2. R square – Coefficient of determination given be the formula:

Where,

$$SSResid = \Sigma(Y - \hat{Y})^2$$

$$SST_o = \Sigma(Y - \bar{Y})^2$$

R-squared is a statistical measure of how close the data are to the fitted regression line. An R^2 value of 0.64eans that our model predicts with an accuracy of 64 percent.

3. Adjusted R square - Adjusted R-squared adjusts the statistic based on the number of independent variables in the model.

4. Standard error – Standard deviation of the error/residual

5. Observations – Total number of observation

The table gives the overall goodness of fit measures.

Hypothesis Testing

To check model utility.

Hypothesis -

H_0 : $\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = \beta_7 = 0$

There is no linear relationship between the dependent and independent variables.

H_A : $\beta_j \neq 0$ where $j = 1,2,3,4,5,6,7$

There is at least one independent variable which has a linear relationship with dependent variable.

ANOVA table –

Source	Sum of squares	Degree of Freedom	Mean squares	F
Treatment	SS_T	k-1	$MS_T = \frac{SS_T}{k-1}$	$F = \frac{MS_T}{MS_E}$
Error	SS_E	N-k	$MS_E = \frac{SS_E}{N-k}$	
Total	TotalSS	N-1		

Figure 12

Here k is the number of coefficients (7) and N is the total number

of observations (506).

With significance α = 0.05 (from figure 12)

	Degrees of freedom	*Sum of squares*	*Mean squares*	*F*	*Significance F (p – value)*
Regression	6	32635.68	5,439.28	237.005	2.11673764
Residual	499	11455.023	22.95		
Total	505	32162.91			

Figure 13

F – critical (df1 = 6, df2 = 499) = 2.11673764; F – statistic = 237.005

Since F – statistic = 237.005 > F – critical = 2.11673764 ,

We reject the null hypothesis. Hence, there is at least one independent variable which has a linear relationship with the dependent variable.

Regression coefficient table and final model

he following table gives the value, standard error (SE), t statistic, p-value

and confidence interval of regression coefficients –

	coef	std err	t	P>|t|	[0.025	0.975]
const	5.5801	3.748	1.489	0.137	-1.789	12.950
x1	-0.0872	0.038	-2.293	0.022	-0.162	-0.012
x2	0.0381	0.015	2.603	0.010	0.009	0.067
x3	4.1231	0.522	7.891	0.000	3.096	5.150
x4	0.0380	0.014	2.641	0.009	0.010	0.066
x5	0.1871	0.083	2.255	0.025	0.024	0.350
x6	-0.0137	0.004	-3.211	0.001	-0.022	-0.005
x7	-0.6152	0.063	-9.705	0.000	-0.740	-0.491

Omnibus:	155.268	Durbin-Watson:	2.002
Prob(Omnibus):	0.000	Jarque-Bera (JB):	650.038
Skew:	1.661	Prob(JB):	7.01e-142
Kurtosis:	8.251	Cond. No.	6.26e+03

Figure 13

Analysis of the coefficient table:

- As expected values of β_2, β_3, β_4 , β_5 are positive and β_1, β_6 , $\beta_7$is negative.
- The large intercept takes may take care of the negative relationship as 5.58 (β_0) is much greater than - 0.087(β_1)

A simple summary of the above output is that the fitted line is (By substituting coefficients in equation *Figure 13*):

$$\boxed{\hat{Y} = 5.58 - 0.087\ X_1 + 0.038X_2 + 4.12\ X_3 + 0.03\ X_4 - 0.18\ X_5 - 0.0.13X_6 - 0.61\ X_5}$$

Model Evaluation

Model performance was tested on 20% of the whole dataset and 3 different accuracy metrics were utilized namely: MAE, MSE, RMSE.

Mean Absolute Error (MAE): Absolute Error is the amount of error in your measurements. It is the difference between the measured value and "true" value

$$\frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

MAE: 4:34

Mean Squared Error (MSE): measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

MSE: 30.17

Root Mean Squared Error (RMSE): is a measure of how spread out these residuals are.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

RMSE: 5.49

Following table allows compare the accuracy of the model with different metrics

	Evaluation Metric	Accuracy
0	MAE	4.34
1	MSE	30.17
2	RMSE	5.49

Figure 14

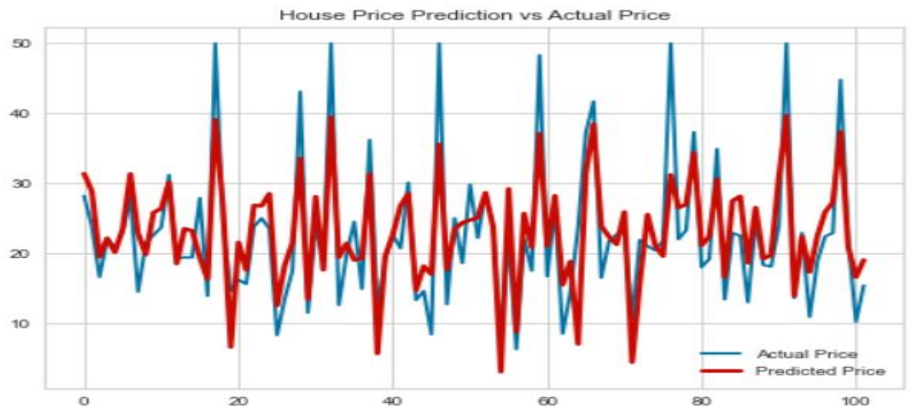

Figure 15

Figure 15 illustrates fit of the model in red line and actual price of the house in blue line.

Residual Plot

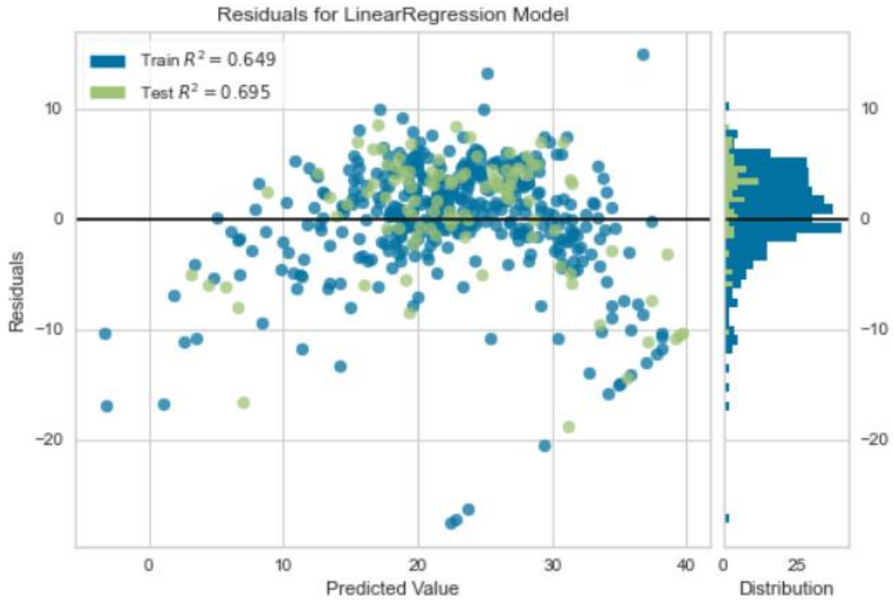

Figure 16

Figure 16 displays the residual plot and distribution of the model for both train and test set.

References

1. *Introduction to statistics and Data Analysis* by Roxy Peck, Chris Olsen and Jay L. Devore
2. *Statistics for Business and Economics* by Anderson, Sweeney, Williams, Camm, Cochran
3. https://www.dataschool.io/applying-and-interpreting-linear-regression/
4. https://people.duke.edu/~rnau/regintro.htm
5. https://www.graduatetutor.com/statistics-tutor/interpreting-regression-output/
6. https://blog.minitab.com/en/adventures-in-statistics-2/understanding-analysis-of-variance-anova-and-the-f-test

7. https://www.wallstreetmojo.com/f-test-formula/
8.
9. https://www.kaggle.com/fedesoriano/the-boston-houseprice-data
10. https://www.statsmodels.org/stable/index.html
11. https://www.scikit-yb.org/en/latest/
12. ht tps://scikit-learn.org/stable/
13. https://matplotlib.org/
14. https://pandas.pydata.org/
15. https://numpy.org/
16. https://www.python.org/

Link to Github repo of dataset and notebook: https://github.com/Javokheer/Multiple-Linear-Regression-MLR

Code Snippets

Importing necessary libraries

In [ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
from scipy import stats
from sklearn.metrics import mean_squared_error, mean_absolute_error
from statsmodels.formula.api import ols
from yellowbrick.regressor import ResidualsPlot
import pickle
```

In [ ]:
```
dataset_df = pd.read_csv('boston-home.csv')
dataset_df.head()
```

In [ ]:
```
dataset_df.shape
```

In [ ]:
```
dataset_df.columns
```

Exploratory Data Analysis

In [ ]:
```
sns.pairplot(dataset_df)
plt.show()
```

In [ ]:
```
# Histrogram represents distribtuion of numerical columns in the train set
dataset_df.hist(alpha=1, figsize=(15, 8), bins=20)
plt.show()
```

In [ ]:
```
dataset_df.plot(kind='box', subplots=True, layout=(4,2), sharex=False, sharey=False, figsize = (10,8))
plt.show()
```

In [ ]:
```
sns.regplot(x='RM', y='LSTAT', data=dataset_df)
plt.show()
```

In [ ]:
```
sns.regplot(x='RM', y='MEDV', data=dataset_df)
plt.show()
```

In [ ]:
```
sns.regplot(x='AGE', y='MEDV', data=dataset_df)
plt.show()
```

In [ ]:
```
dataset_df.describe()
```

In [ ]:
```
dataset_df.isnull().sum().to_frame(name='Number of missing values')
```

In [ ]:
```
df_cor = dataset_df.corr().round(2)
df_cor.MEDV.to_frame('Attribute Correlation with Target')
```

In [ ]:
```
sns.heatmap(df_cor, annot=True)
plt.show()
```

Splitting into dependent and independent variables

In [ ]:
```
x = dataset_df.iloc[:, 0:7].values
y = dataset_df.iloc[:, -1].values
```

In [ ]:
```
x.shape, x
```

In [ ]:
```
y.shape, y
```

Splitting X and y into training and testing sets

In [ ]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=1)
```

In [ ]:
```
# default split is 80% for training and 20% for testing
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

In [ ]:
```
lr = LinearRegression()
lr.fit(x_train, y_train)
y_pred = lr.predict(x_test)
```

In [ ]:
```
print(lr.intercept_.round(3))
print(lr.coef_.round(3))
```

In [ ]:
```
features = ['CRIM', 'ZN', 'RM', 'AGE', 'RAD', 'TAX', 'LSTAT']
list(zip(features, lr.coef_.round(3)))
```

$$y = 5.58 - 0.087 \times CRIM + 0.038 \times ZN + 4.123 \times RM + 0.038 \times AGE + 0.187 \times RAD - 0.014 \times TAX - 0.615 \times LSTAT$$

Model evaluation metrics for regression

Evaluation metrics for classification problems, such as **accuracy**, are not useful for regression problems. Instead, we need evaluation metrics designed for comparing continuous values.

Let's create some example numeric predictions, and calculate **three common evaluation metrics** for regression problems:

Mean Absolute Error (MAE) is the mean of the absolute value of the errors:

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

In [ ]:
```
print('Mean absolute error: ', mean_absolute_error(y_test, y_pred).round(2))
```

Mean Squared Error (MSE) is the mean of the squared errors:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

In [ ]:
```
print('Mean squared error: ', mean_squared_error(y_test, y_pred).round(2))
```

Root Mean Squared Error (RMSE) is the square root of the mean of the squared errors:

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

In [ ]:
```
print('Root mean squared error: ', np.sqrt(mean_squared_error(y_test, y_pred)).round(2))
```

```
performance = pd.DataFrame({'Evaluation Metric': ['MAE', 'MSE', 'RMSE'], 'Accuracy':[4.34, 30.17, 5.49]})
performance
```

```
x2 = sm.add_constant(x_train)
model = sm.OLS(y_train, x2)
est2 = model.fit()
print(est2.summary())
```

```
plt.plot(y_test, color='b', label='Actual Price', linewidth=2, markersize=8)
plt.plot(y_pred, color='r', label='Predicted Price', linewidth=3, markersize=10)
plt.legend()
plt.title('House Price Prediction vs Actual Price')
plt.show()
```

Hypothesis testing

```
class Stats:
    def __init__(self, X, y, model):
        self.data = X
        self.target = y
        self.model = model
        ## degrees of freedom population dep. variable variance
        self._dft = X.shape[0] - 1
        ## degrees of freedom population error variance
        self._dfe = X.shape[0] - X.shape[1] - 1

    def sse(self):
        '''returns sum of squared errors (model vs actual)'''
        squared_errors = (self.target - self.model.predict(self.data)) ** 2
        return np.sum(squared_errors)

    def sst(self):
        '''returns total sum of squared errors (actual vs avg(actual))'''
        avg_y = np.mean(self.target)
        squared_errors = (self.target - avg_y) ** 2
        return np.sum(squared_errors)

    def r_squared(self):
        '''returns calculated value of r^2'''
        return 1 - self.sse()/self.sst()

    def adj_r_squared(self):
        '''returns calculated value of adjusted r^2'''
        return 1 - (self.sse()/self._dfe) / (self.sst()/self._dft)
```

```
stats = Stats(x_train, y_train, lr)
```

```
stats.sse().round(3)
```

```
stats.sst().round(3)
```

```
stats.r_squared().round(3)
```

```
stats.adj_r_squared().round(3)
```

Residual Plot

```
# Instantiate the linear model and visualizer
model = LinearRegression()
visualizer = ResidualsPlot(model)

visualizer.fit(x_train, y_train)  # Fit the training data to the visualizer
visualizer.score(x_test, y_test)  # Evaluate the model on the test data
visualizer.show()                 # Finalize and render the figure
plt.show()
```

```
with open ('model_saved', 'wb') as f:
    pickle.dump(model, f)
```