Machine Learning Project Report

House Price Prediction Model

INDEX

Contents

Figures

I. Executive Summary

A house value is simply more than location and square footage. Like the features that make up a person, an educated party would want to know all aspects that give a house its value. For example, you want to sell a house and you don't know the price which you may expect — it can't be too low or too high. To find house price you usually try to find similar properties in your neighbourhood and based on gathered data you will try to assess your house price.

II. Introduction

The problem statement discussed above is classified as a *Regression* problem in the domain of machine learning. The various input features (x_1, x_2, …) can be used to determine a best fitting model $h_\theta(x)$ such that the output price is a real number. The equation is described as:

$$Y = h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

Hence, in this project report, we discuss about applying this model to the given dataset. We will explore the data and do some analysis to get insights on the provided data, detect important features — scale and encode them — and at last fit a Linear Regression model to predict the value of price.

III. Reading and Sampling Data

We read the given Excel to create a pandas data frame.
Sample of Dataset:

	cid	dayhours	price	room_bed	room_bath	living_measure	lot_measure	ceil	coast	sight	...	basement	yr_built	yr_renovated	zipcode	
0	3876100940	20150427T000000	600000	4.0	1.75	3050.0	9440.0	1	0	0.0	...	1250.0	1966	0	98034	47.72
1	3145600250	20150317T000000	190000	2.0	1.00	670.0	3101.0	1	0	0.0	...	0.0	1948	0	98118	47.55
2	7129303070	20140820T000000	735000	4.0	2.75	3040.0	2415.0	2	1	4.0	...	0.0	1966	0	98118	47.51
3	7338220280	20141010T000000	257000	3.0	2.50	1740.0	3721.0	2	0	0.0	...	0.0	2009	0	98002	47.33
4	7950300670	20150218T000000	450000	2.0	1.00	1120.0	4590.0	1	0	0.0	...	0.0	1924	0	98118	47.56

Figure 1: Data Sampling

The dataset has 21,613 data points and 23 features for each data point.

IV. Data Analysis

A. Data Information

We can observe that there is no blank(null) value present in the dataset and no duplicate rows in the dataset. Columns cid, price, yr_renovated, zipcode are int64. Columns dayhours, ceil, coast, condition, yr_built, long, total_area are object type. Columns room_bed, room_bath, living_masure, lot_measure, sight, quality, ceil_measure, basement, lat, living_measure15, lot_measure15, furnished are object type.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21472 entries, 0 to 21612
Data columns (total 23 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   cid                21472 non-null  int64
 1   dayhours           21472 non-null  object
 2   price              21472 non-null  int64
 3   room_bed           21406 non-null  float64
 4   room_bath          21406 non-null  float64
 5   living_measure     21455 non-null  float64
 6   lot_measure        21430 non-null  float64
 7   ceil               21430 non-null  object
 8   coast              21471 non-null  object
 9   sight              21415 non-null  float64
 10  condition          21415 non-null  object
 11  quality            21471 non-null  float64
 12  ceil_measure       21471 non-null  float64
 13  basement           21471 non-null  float64
 14  yr_built           21471 non-null  object
 15  yr_renovated       21472 non-null  int64
 16  zipcode            21472 non-null  int64
 17  lat                21472 non-null  float64
 18  long               21472 non-null  object
 19  living_measure15   21348 non-null  float64
 20  lot_measure15      21443 non-null  float64
 21  furnished          21443 non-null  float64
 22  total_area         21443 non-null  object
dtypes: float64(12), int64(4), object(7)
memory usage: 3.9+ MB
```

Figure 2: Data Information

Data Summary:

	cid	price	room_bed	room_bath	living_measure	lot_measure	sight	quality	ceil_measure	basement	yr_renovated	zipcode	lat	living_measure15	lot_measure15	furnished
count	2.161300e+04	2.161300e+04	21505.000000	21505.000000	21596.000000	2.157100e+04	21556.000000	21612.000000	21612.000000	21612.000000	21613.000000	21613.000000	21613.000000	21447.000000	21584.000000	21584.000000
mean	4.580302e+09	5.401822e+05	3.371355	2.115171	2079.860761	1.510458e+04	0.234366	7.656857	1788.366556	291.522534	84.402258	98077.939805	47.560053	1987.065557	12766.543180	0.196720
std	2.876566e+09	3.673622e+05	0.930289	0.770248	918.496121	4.142362e+04	0.766438	1.175484	828.102535	442.580840	401.679240	53.505026	0.138564	685.519629	27286.987107	0.397528
min	1.000102e+06	7.500000e+04	0.000000	0.000000	290.000000	5.200000e+02	0.000000	1.000000	290.000000	0.000000	0.000000	98001.000000	47.155900	399.000000	651.000000	0.000000
25%	2.123049e+09	3.219500e+05	3.000000	1.750000	1429.250000	5.040000e+03	0.000000	7.000000	1190.000000	0.000000	0.000000	98033.000000	47.471000	1490.000000	5100.000000	0.000000
50%	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e+03	0.000000	7.000000	1560.000000	0.000000	0.000000	98065.000000	47.571800	1840.000000	7620.000000	0.000000
75%	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000	1.068450e+04	0.000000	8.000000	2210.000000	560.000000	0.000000	98118.000000	47.678000	2360.000000	10087.000000	0.000000
max	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	4.000000	13.000000	9410.000000	4820.000000	2015.000000	98199.000000	47.777600	6210.000000	871200.000000	1.000000

Figure 3: Data Summary

We can observe that mean and median vary for all features. Hence for model to work affectively, we need to scale the features.

B. Data Skewness

```
cid                   0.243219
price                 3.873235
room_bed              1.985191
room_bath             0.496752
living_measure        1.454467
lot_measure          13.084895
ceil                  0.616166
coast                11.455112
sight                 3.398491
condition             1.032563
quality               0.767556
ceil_measure          1.425115
basement              1.578246
yr_built             -0.470277
yr_renovated          4.558198
zipcode               0.406357
lat                  -0.484619
long                  0.886846
living_measure15      1.107623
lot_measure15         9.518962
furnished             1.524216
total_area           12.977128
dtype: float64
```

Figure 4: Data Skewness

We observe that the majority of skewness is greater than 0 that means more weight on the right tailed that is data is right/positive skewed. The features yr_built, and lat are slightly left skewed.

C. Data Normal Distribution

The histogram is used to check the distribution of the data. If the data is normally distributed then the histogram will be a bell curve. If the data is not normally distributed then the histogram will not be a bell curve.

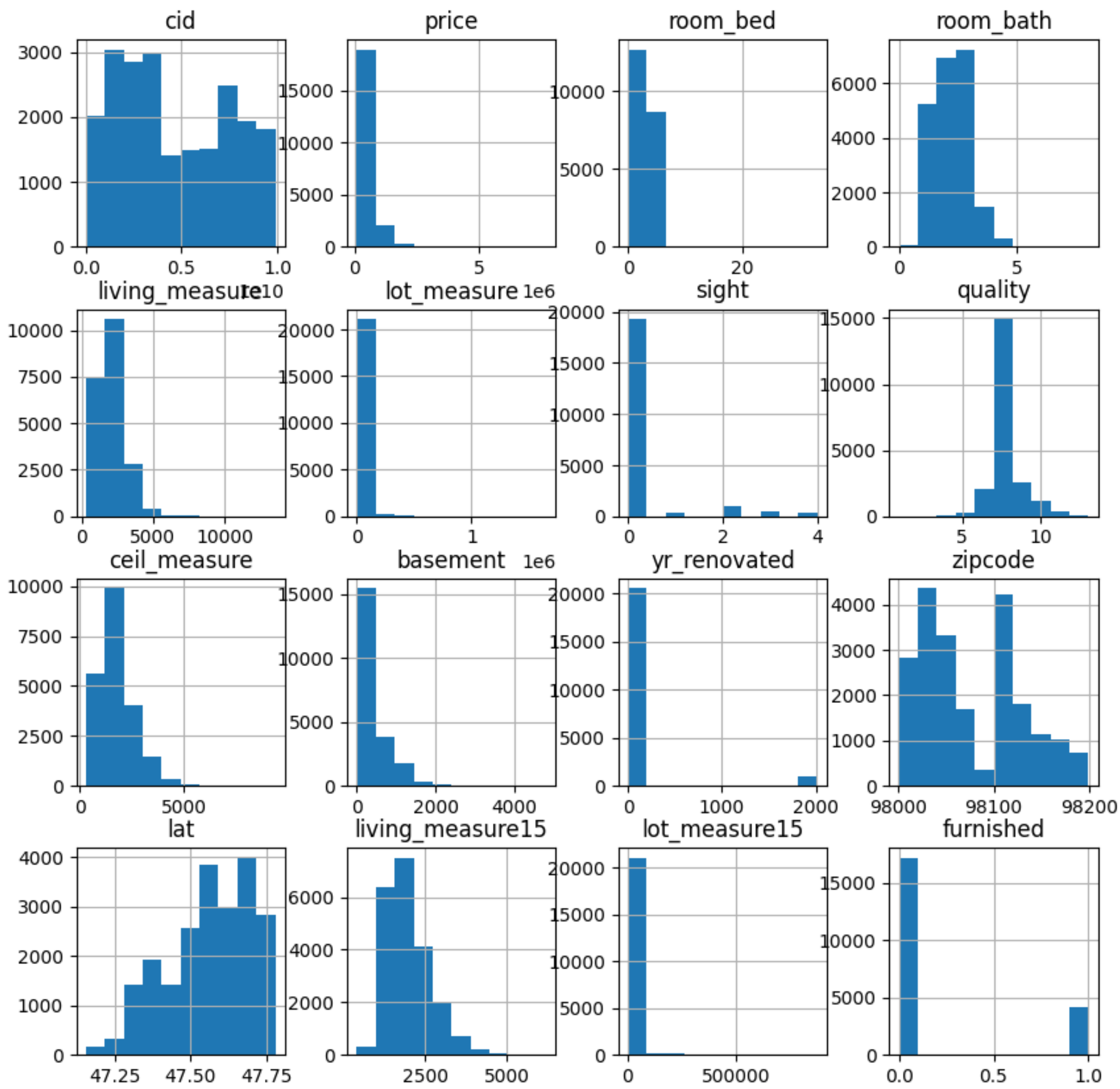

Figure 5: Data Distribution

Here the data is not normally distributed as the histogram is not symmetric.

D. Pair Plot

Pair plots shows relationship between the variables and the diagonal shows the distribution of the variables. It is done by taking the variables one by one and plotting them against each other.

Here we can see that there is a linear relationship between the variables as the data is not normally distributed.

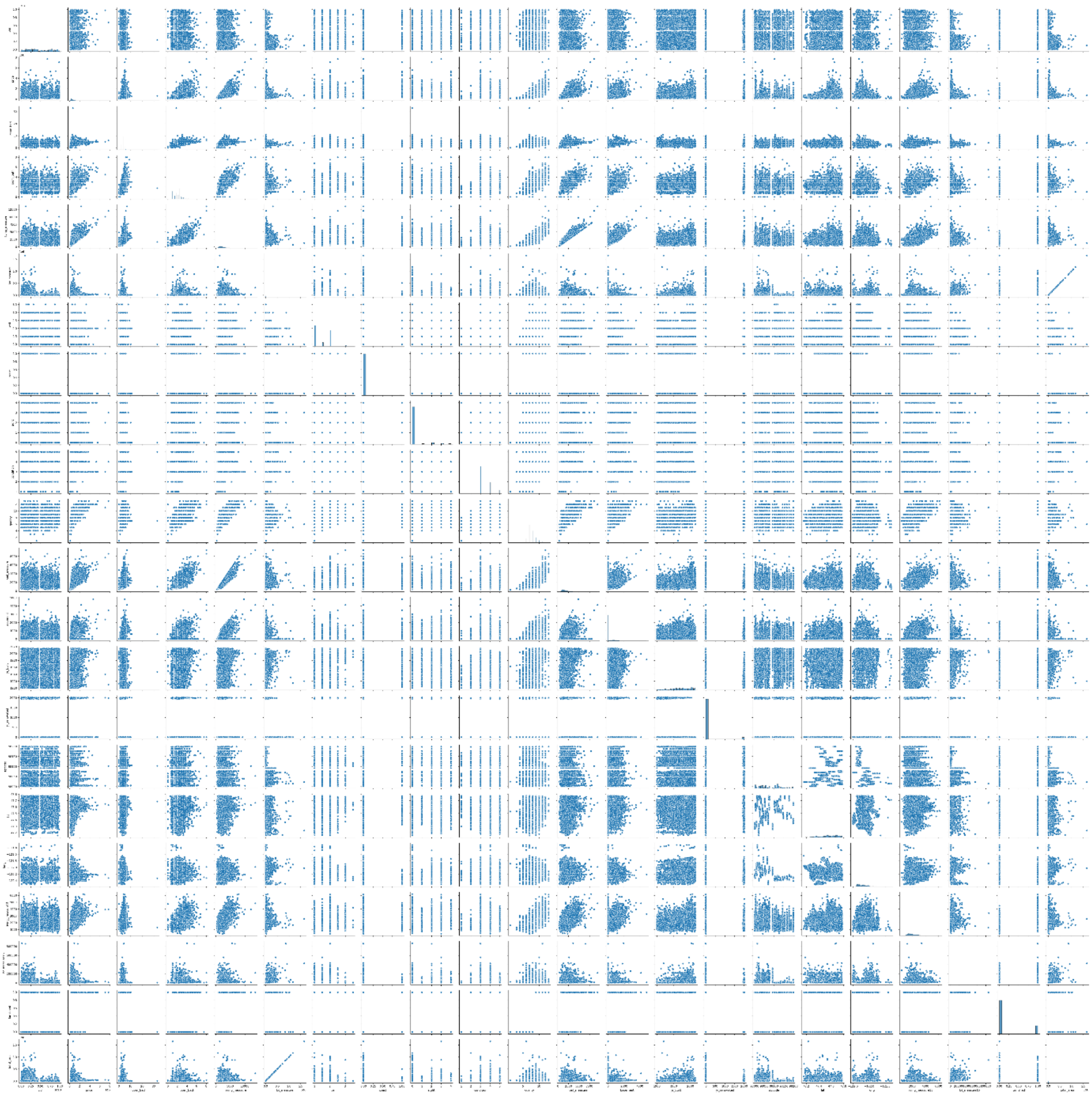

Figure 6: Pair Plot - Relationship between different variables

E. Heatmap

Heat map shows the correlation between the variables. The darker the color the more the correlation between the variables.
Here we can see that the variables are not correlated with each other. There is randomness in relationships between different variables. We see that our target variable price is somewhat equally related to all variables.

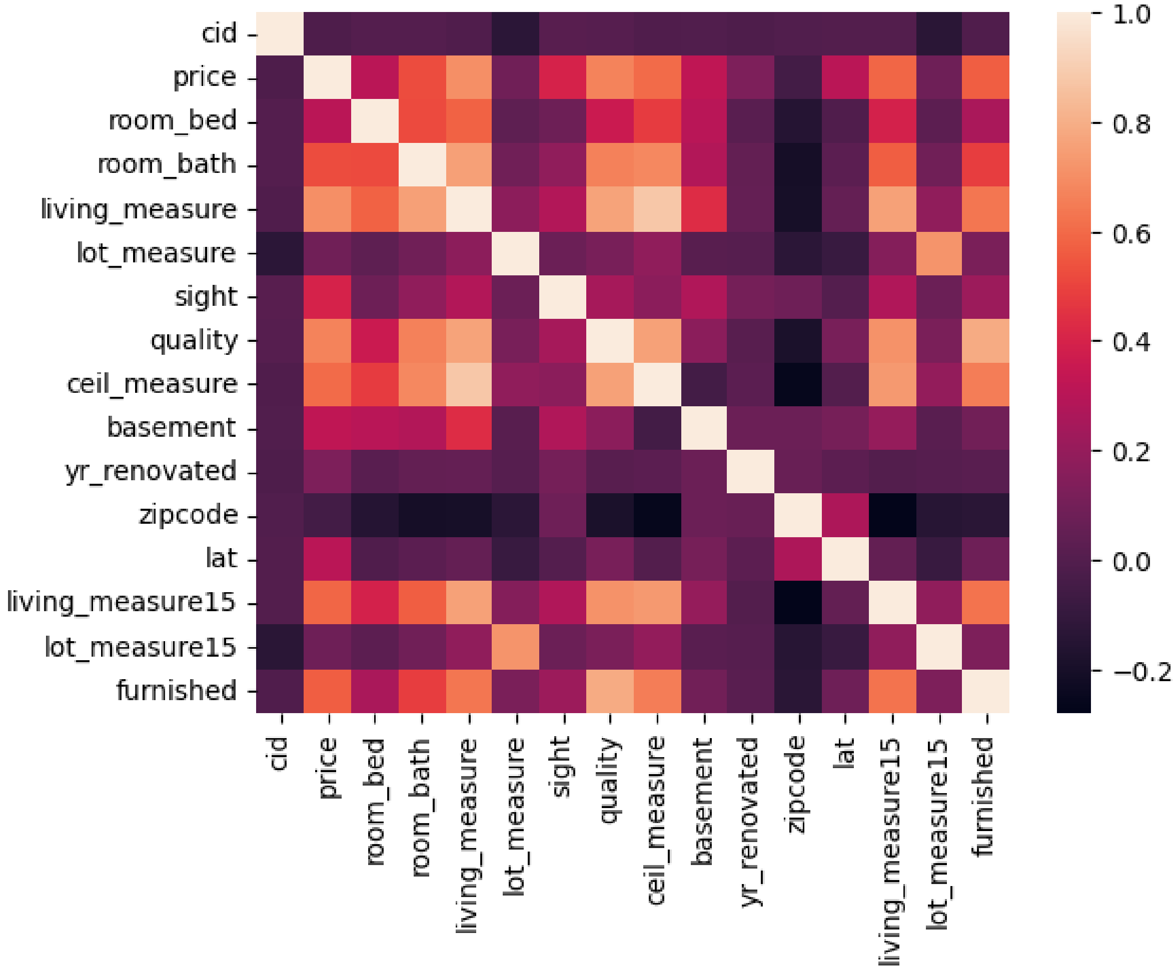

Figure 7: Heatmap - Correlation between different variables

F. Outlier Detection

We will plot bar graphs of all the features, except CID and Time Stamp to check if we have any outliers in data so we can adjust accordingly.

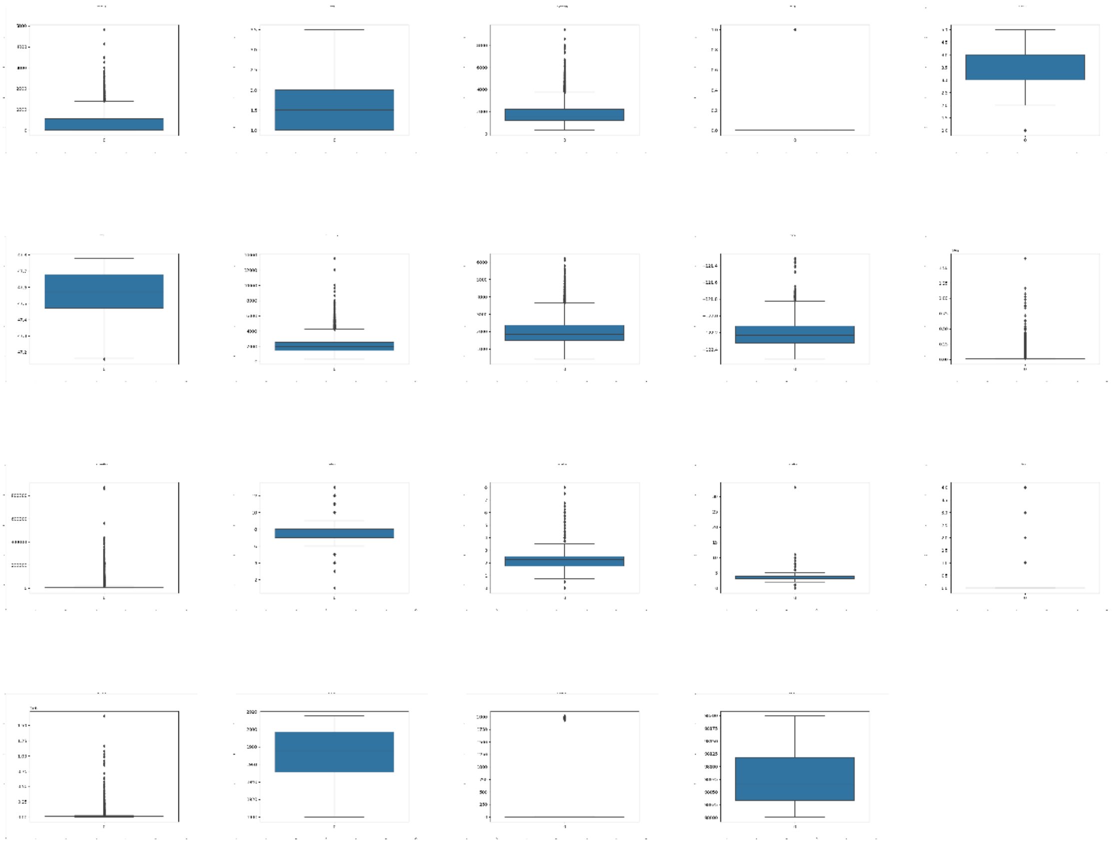

Figure 8: Box Plot of all features

G. Outlier Treatment

Once we have identified the outliers, we can set boundaries so to avoid the outliners.

Set:

- room_bed < 8
- room_bath < 5
- living_measure < 6000
- lot_measure < 100000
- ceil < 4
- coast < 2
- sight < 5
- condition < 5
- quality < 12
- ceil_measure < 6000
- basement < 4000
- yr_built > 1900
- yr_renovated < 2015
- zipcode < 98080
- lat > 47
- long < -120

- living_measure15 < 6000
- lot_measure15 < 100000
- total_area < 100000

We treat outliers to be the values which are more than 3 standard deviations away from the mean.

H. Feature Drop

We drop the first two columns – CID, Time stamp of house sale, basement, yr_renovated.
We drop features to avoid multicollinearity and these are not related much to predicting final price.

V. Machine Learning Model

A. Linear Regression

As discussed we use a simple linear regression model to solve the predicting model. To do this, we split the data set into 70:30 training to testing ratio. We will apply feature scaling and encoding.

B. Feature Scaling

As we saw that mean and median vary for all features, we will need to scale them. The higher value numerical features — living_measure, lot_measure, ceil_measure, living_measure15, lot_measure15, total_area will be scaled using StandardScaler API call of sklearn library.

C. Feature Encoding

The features which have a fixed number of outcome — room_bed, room_bath, ceil, coast, sight, condition, quality, zipcode will be encoded.
These will convert into dummy variables to train the model.

D. Applying Linear Regression Model

After setting the data pipeline, scaling, and encoding, we can final envoke the linear regression model from sklearn.linear_model python library and fit the model.

E. Model Evaluation

After training, the model has a Mean Squared Error:
15625649747.690655 = 1.5625*10^6
It has an R2 score of 0.8516218103729474.
It has a Coefficient of determination: 0.85.

F. Final Results

The model is **85% accurate** in predicting the price correctly.
The plot of predicted values against the actual values and the line of best fit is:

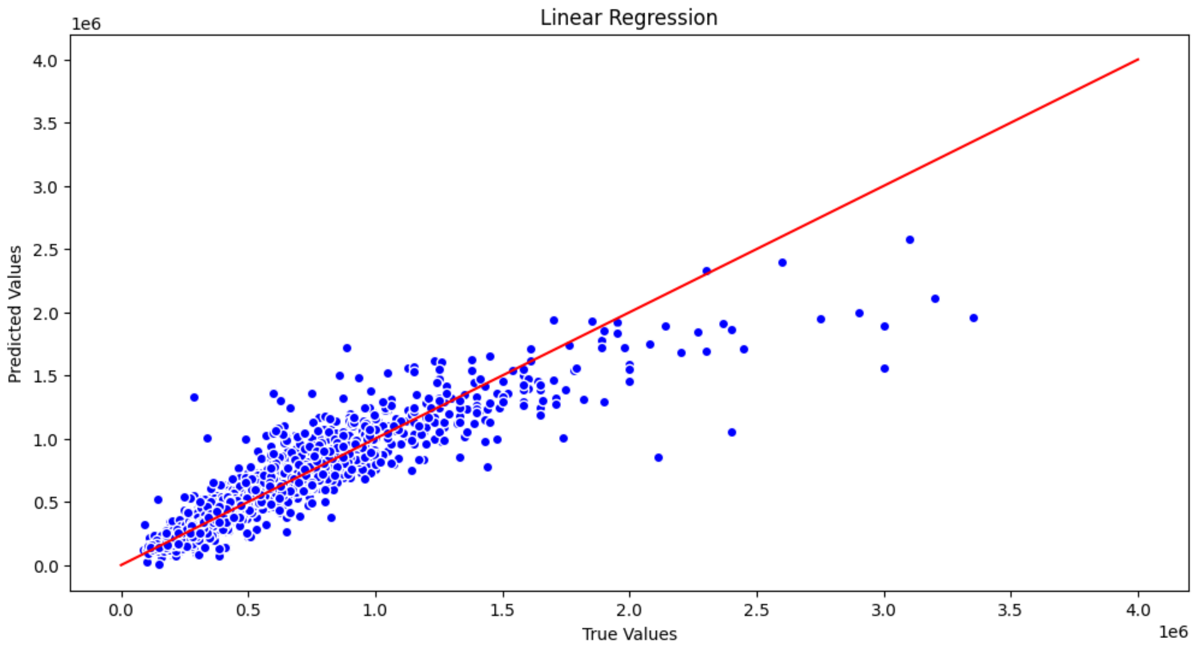

Figure 9: Prediction v/s Actual Value & Best Fit Line

VI. Appendix

A. Libraries Used

The list of python libraries used for solving above problem statement.

- Pandas
- MatPlotLib
- Seaborn
- SkLearn – PreProcessing
- SkLearn – Model_Selection
- SkLearn – Linear_Model
- SkLearn - Metrics