**Machine Learning Project Report**

**House Price Prediction Model**

**INDEX**

**Contents**

1. **Executive Summary** **3**
2. **Introduction 3**
3. **Reading and Sampling Data 3**
4. **Data Analysis 3**
   1. *Data Information 3*
   2. *Data Skewness 5*
   3. *Data Normal Distribution 5*
   4. *Pair Plot 6*
   5. *Heatmap 7*
   6. *Outliers Detection 8*
   7. *Outliers Treatment 9*
   8. *Feature Drop 1*0
5. **Machine Learning Model 10**
   1. *Linear Regression 10*
   2. *Feature Scaling 10*
   3. *Feature Encoding 10*
   4. *Applying Linear Regression Model 10*
   5. *Model Evaluation 11*
   6. *Final Results 11*
6. **Appendix: 11**
   1. *Libraries Used 11*

**Figures**

Figure1: Data Sampling 3

Figure2: Data Information 4

Figure3: Data Summary 4

Figure4: Data Skewness 5

Figure5: Data Distribution 6

Figure6: Pair Plot - Relationship between different variables 7

Figure7: Heatmap - Correlation between different variables 8

Figure8: Box Plot of all features 9

Figure9: Prediction v/s Actual Value & Best Fit Line 11

1. **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.

1. **Introduction**

The problem statement discussed above is classified as a *Regression* problem in the domain of machine learning. The various input features (x1, x2, …) can be used to determine a best fitting model *hθ(x)* such that the output price is a real number. The equation is described as:

*Y = hθ(x) = θ0 + θ1x1 + θ2x2 + … + θnxn*

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.

1. **Reading and Sampling Data**

We read the given Excel to create a pandas data frame.

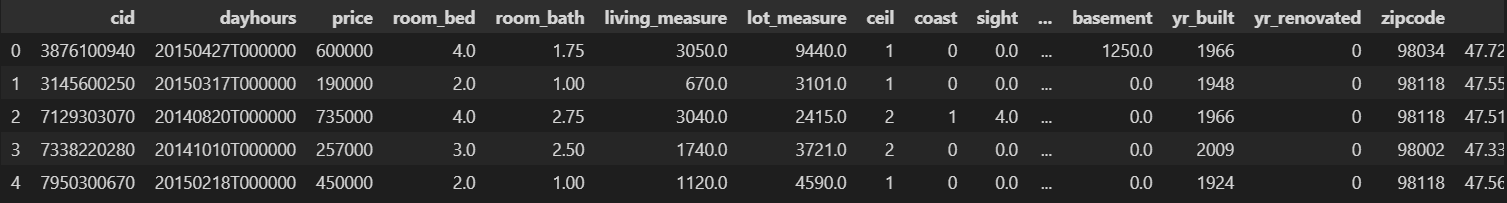
Sample of Dataset:

Figure 1: Data Sampling

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

1. **Data Analysis**
2. **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.

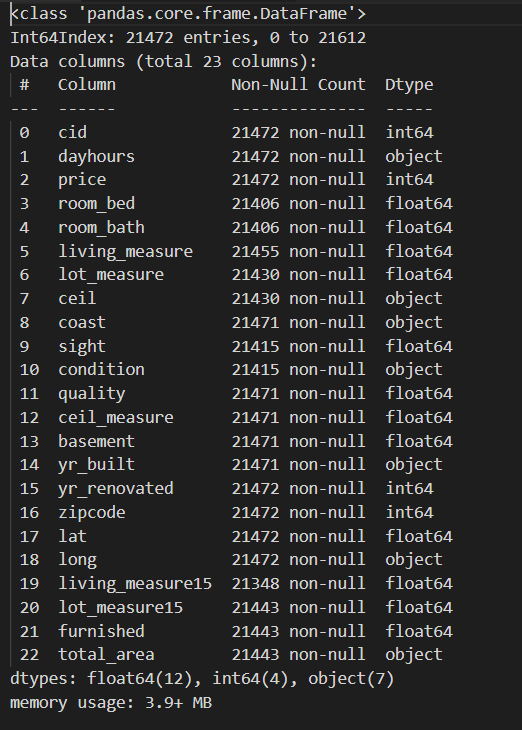


Figure 2: Data Information

Data Summary:

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.

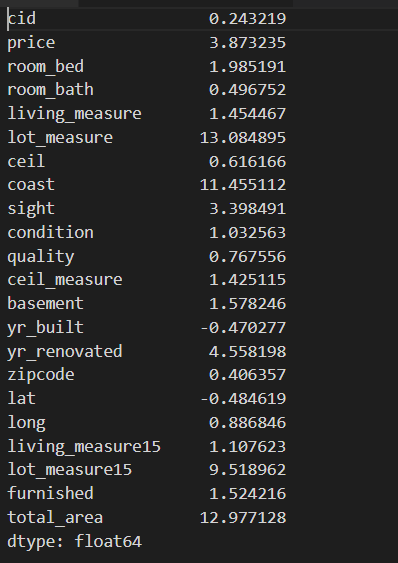
1. **Data Skewness**

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.

1. **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.

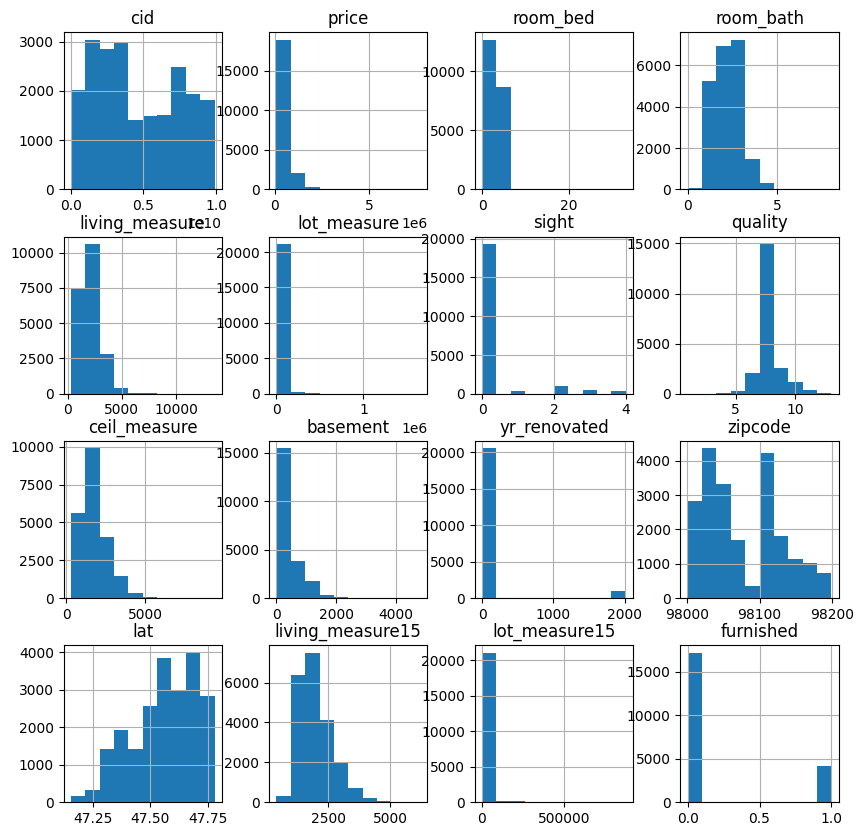
Here the data is not normally distributed as the histogram is not symmetric.

Figure 5: Data Distribution

1. **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.

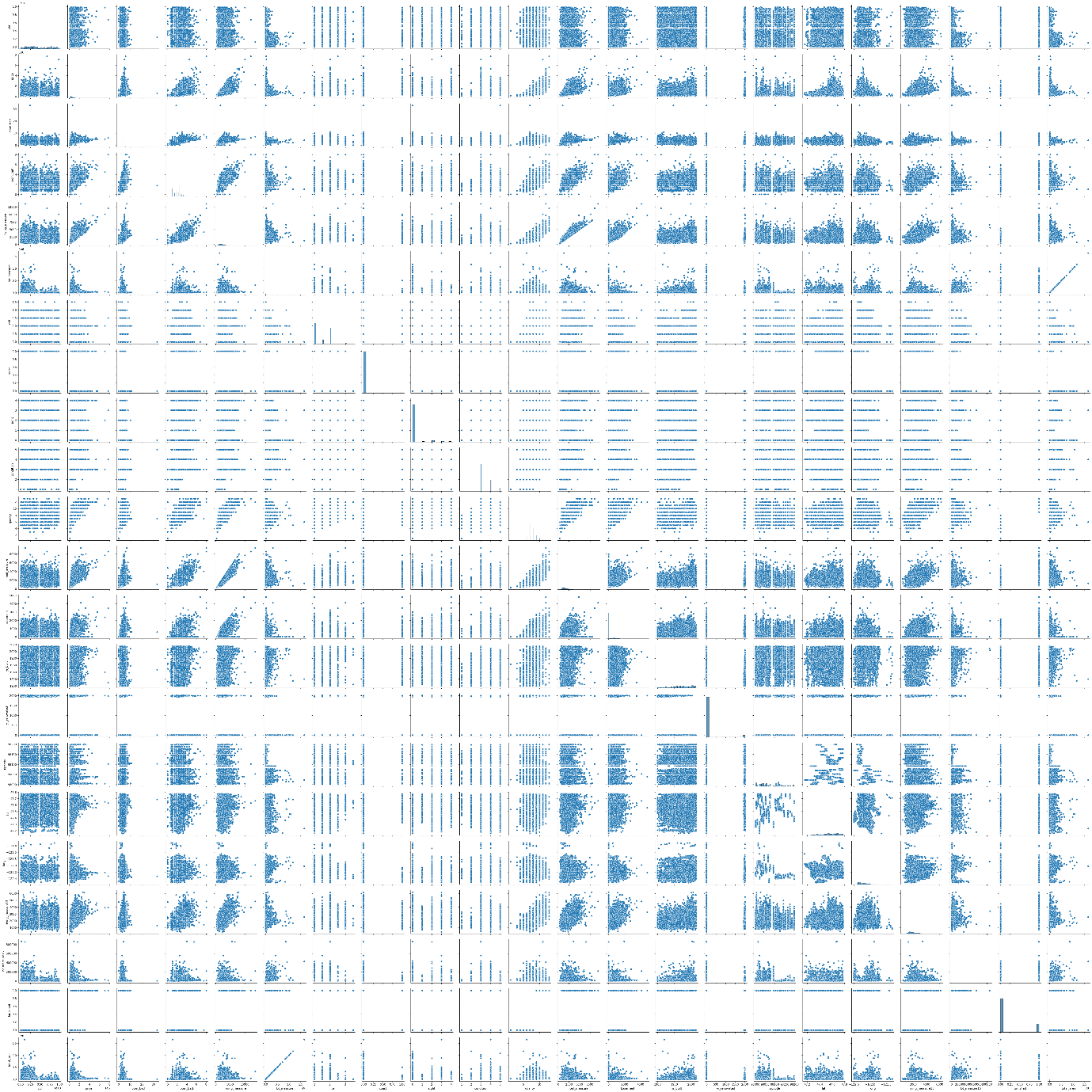


Figure 6: Pair Plot - Relationship between different variables

1. **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.

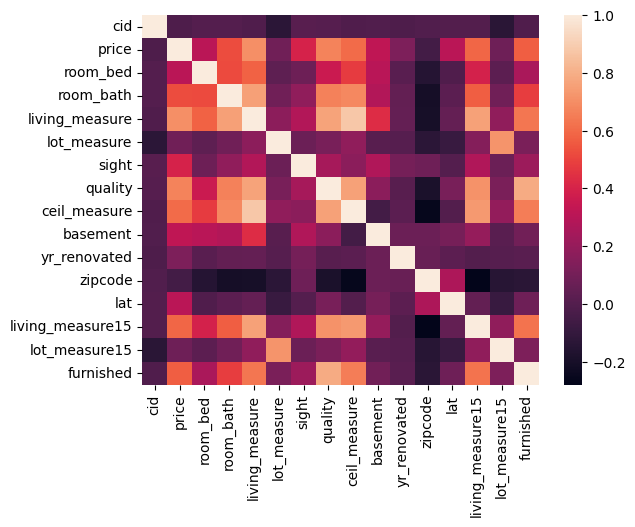


Figure 7: Heatmap - Correlation between different variables

1. **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.

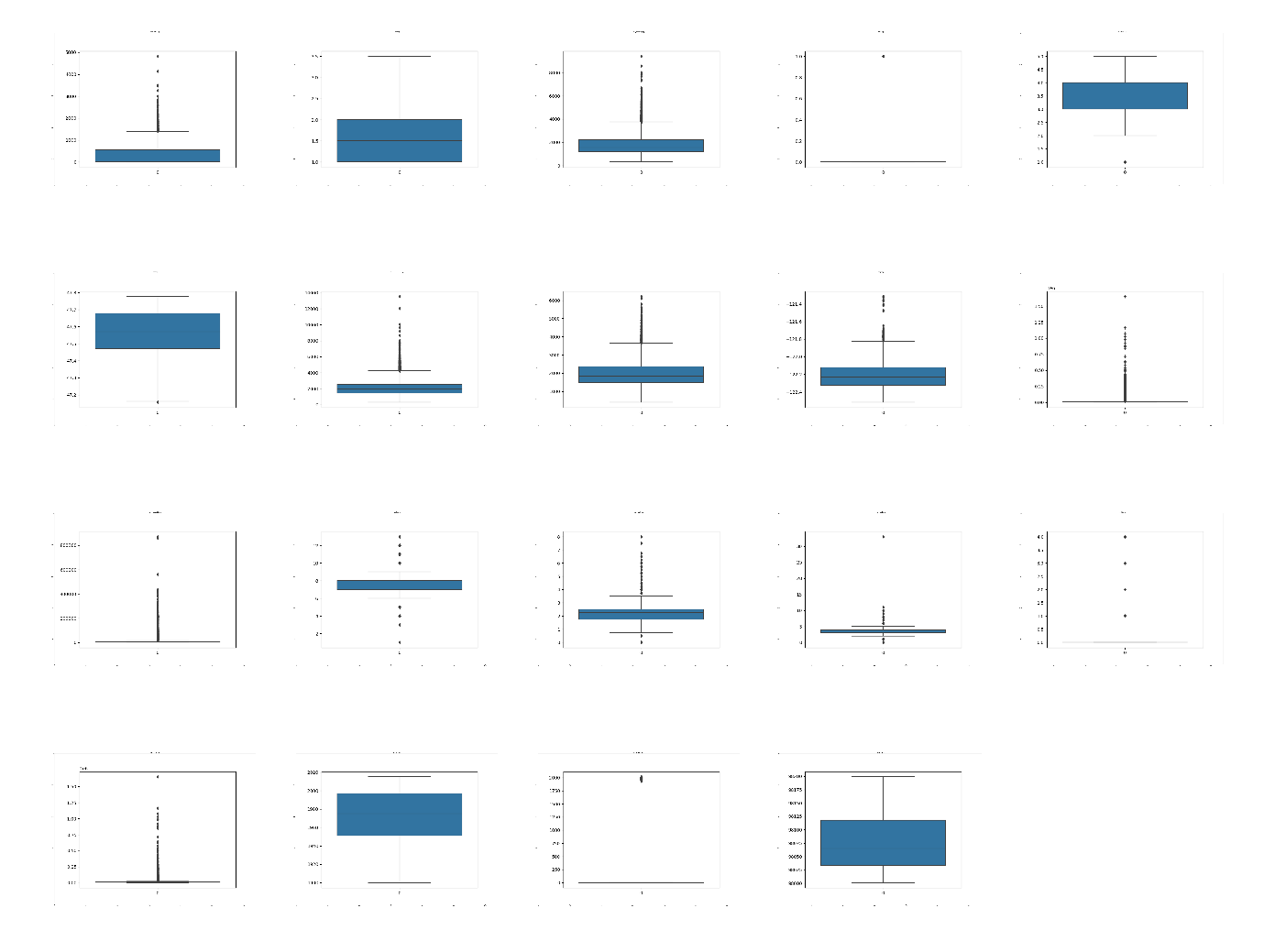
1. **Outlier Treatment**

Figure 8: Box Plot of all features

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.

1. **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.

1. **Machine Learning Model**
2. **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.

1. **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.

1. **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.

1. **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.

1. **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.

1. **Final Results**

The model is **85% accurate** in predicting the price correctly.

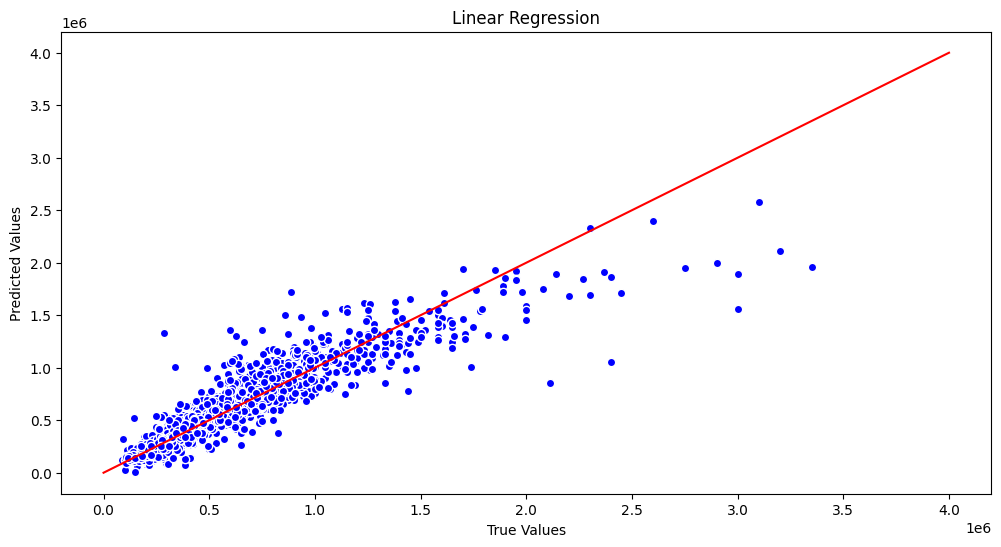
The plot of predicted values against the actual values and the line of best fit is:

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

1. **Appendix**
2. **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