**Final Report on**

***House Price Prediction***

***Submitted to***



**Great Learning Olympus**

***Prepared by***

**Mrinmoy Majumdar**

**June-D Batch**

***Under the guidance of***

**Ms Keya Choudhury Ganguli**

**Post Graduate Program in Data Science & Business Analytics**

**From**

****

UT Austin

**July, 2022**

**TABLE OF CONTENT**

**1. Introduction**………………………………………………………………...4

**2. EDA and Business Implication………………………………….**………....5

**3. Data Cleaning and Pre-Processing………………………………...............**17

###### **4. Model Building………………………….**………………………………...27

**5. Model Validation …………...**.....................................................................42

6. Final Interpretation / recommendation.………...........................................46

**LIST OF FIGURES**

|  |  |
| --- | --- |
| **Images** | **Page No** |
| **Count plot** | **5** |
| **Histogram** | **8** |
| **Bar plot** | **12** |
| **Regression plot** | **13** |
| **Scatter plot** | **13** |
| **Box plot** | **14** |
| **Correlation plot** | **15** |
| **Pair plot** | **16** |
| **Point plot** | **17** |
| **Missing values** | **18** |
| **Skewness** | **21** |
| **Log transformation** | **24** |
| **Coefficient** | **28** |
| **Intercept** | **28** |
| **Training and testing score** | **28** |
| **VIF** | **29** |
| **Parameter values** | **30** |
| **Residual plot** | **32** |
| **Density plot** | **32** |
| **Q-Q plot** | **33** |
| **Regression equation** | **33** |
| **MCE score and plot** | **38** |
| **MSE, RMSE and MAE** | **42** |

**LIST OF TABLES**

|  |  |
| --- | --- |
| Table 1 | 11 |
| Table 2 | 24 |
| Table 3 | 26 |
| Table 4 | 29 |
| Table 5 | 30 |
| Table 6 | 47 |
| Table 7 | 48 |
| Table 8 | 49 |

1. **Introduction**

**Brief Introduction about the problem statement**

House price increase every year. Buyers and sellers are unaware of the factors influencing the price. Real estate industry has knowledge of the factors that affect property price which are shown as below –

* House area
* House age
* Transportation availability
* Proximity to park, supermarket stores and schools
* House prices in the neighbourhood
* Population density
* Construction quality
* Basic amenities and recreation

Real estate company has given the responsibility to us as a data analyst to analyse the various features provided that may have an impact on property rates. It is also our duty to address common concerns of customers based on the analysis like the appropriate price to buy / sell the house, determining prospective location and investing on type of property. Variables in hand to predict price –

* Room bed
* Room bath
* Condition
* Quality
* Coast
* Sight
* Basement
* Latitude
* Longitude
* Zip code
* Living measure
* Lot measure
* Ceil
* Year built
* Year renovated
* Total area
* Furnished

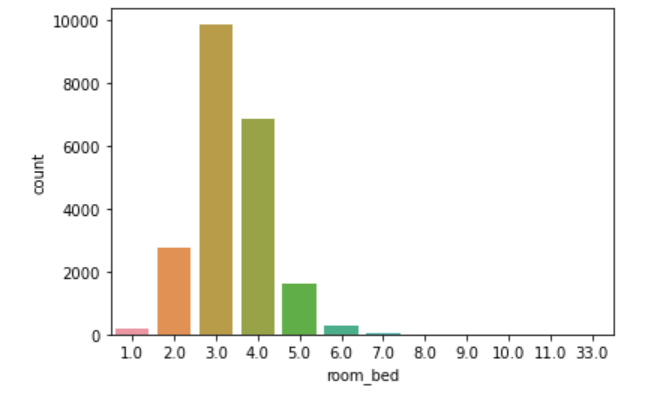
**Need of problem solving**

Real estate is a very complex business. It contributes to the country’s economy and employment. Solving the house price problem would help to ensure profits for all parties involved including buyers, sellers and the company. There is a need for a stable system to predict price of residential homes. As a data scientist, we would be delivering a great service to the nation by resolving a real-life problem that would challenge our data analysis skills.

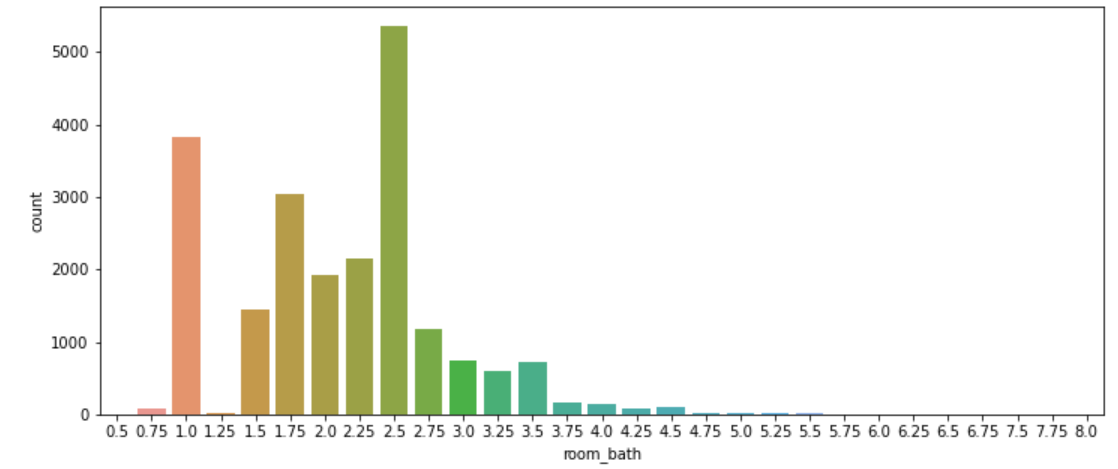
We wish to achieve a high accuracy model leading to suitable insights and recommendations for the business.

1. **EDA and Business Implication**

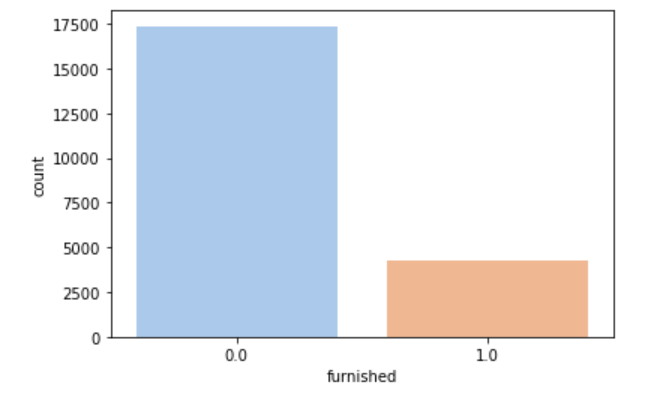
**Uni-variate / Bi-variate / Multi-variate analysis to understand relationship b/w variables**



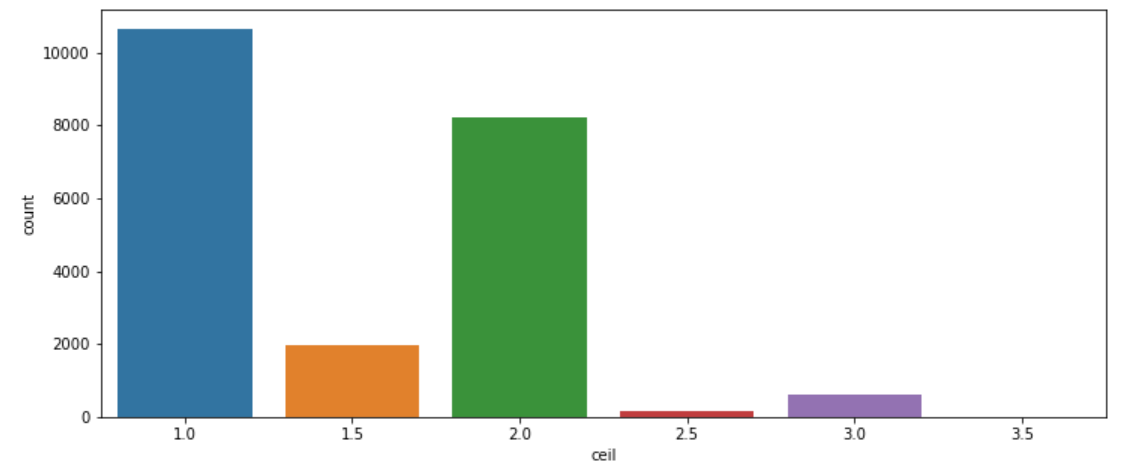
**Figure 1: Room bed count plot**



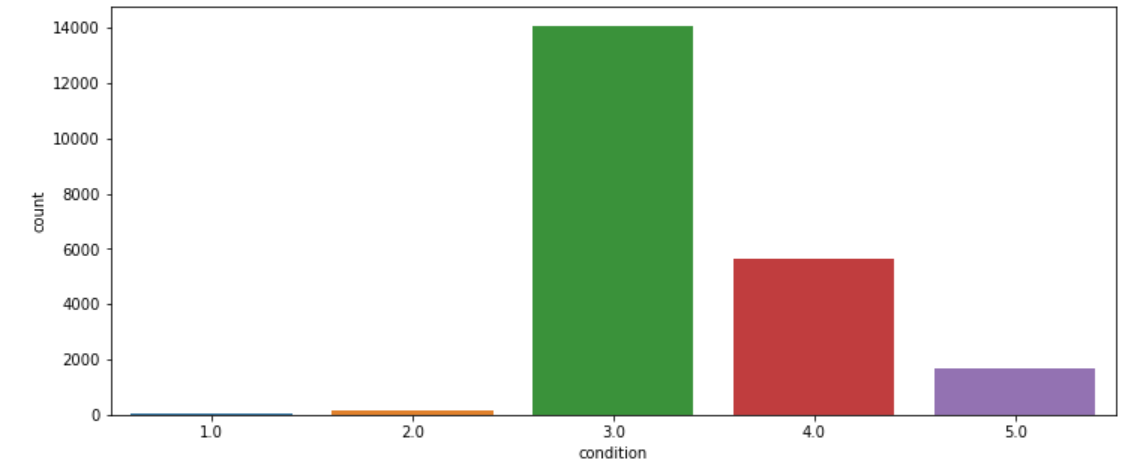
**Figure 2: Room bath count plot**



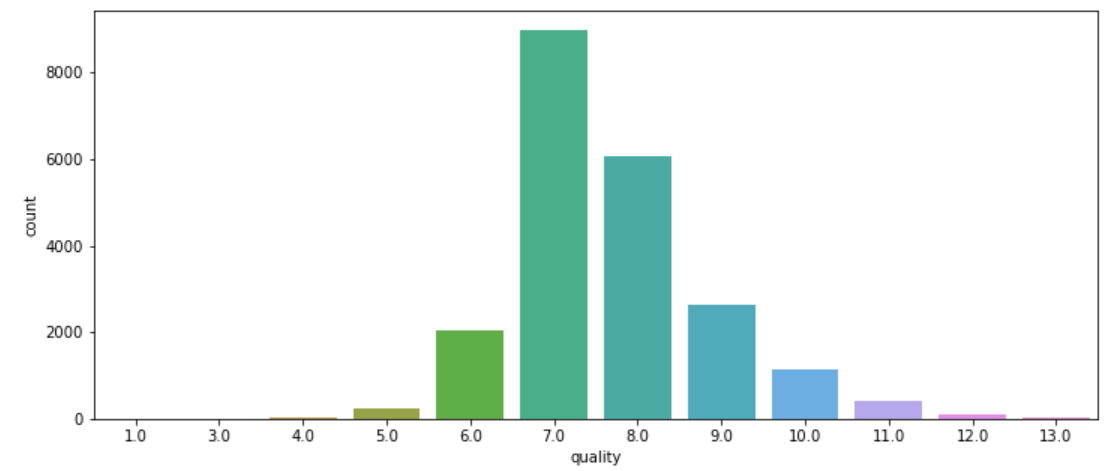
**Figure 3: Furnished count plot**



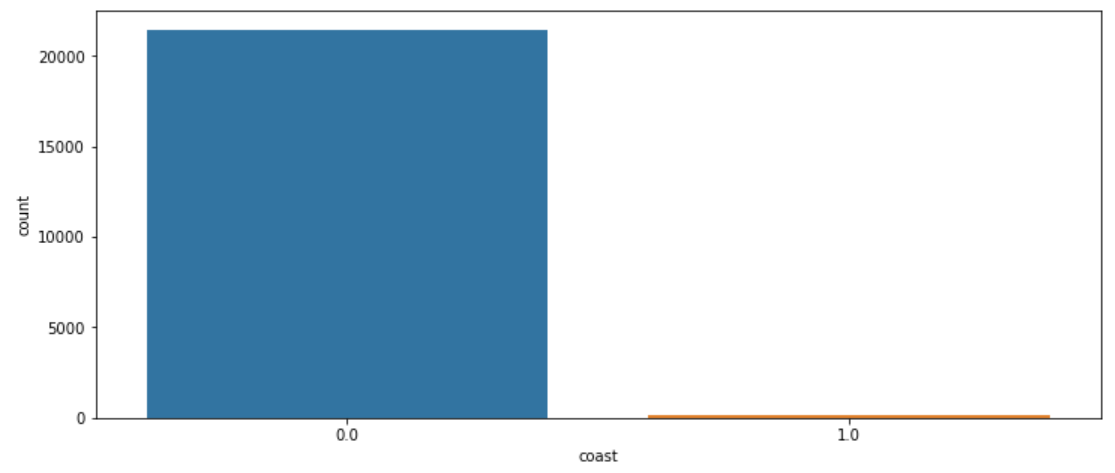
**Figure 4: Ceil count plot**



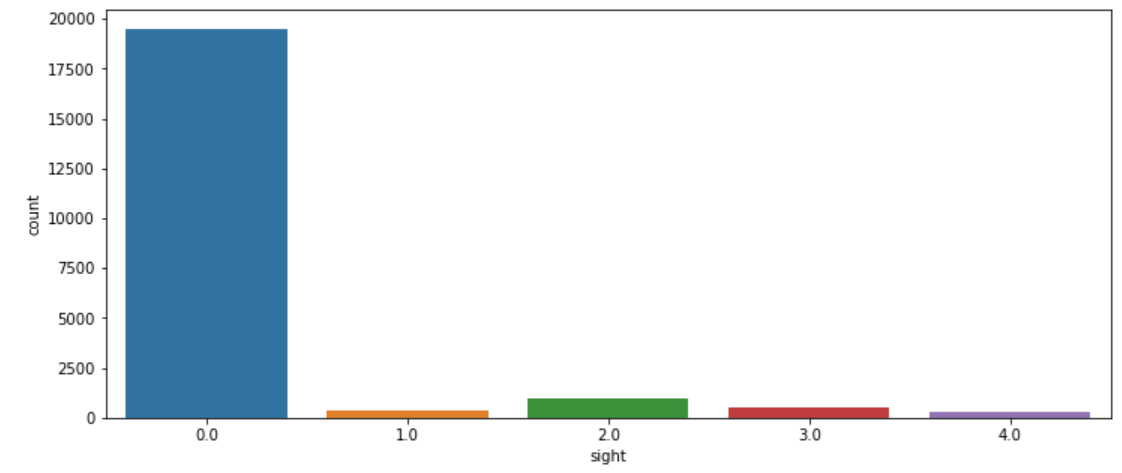
**Figure 5: Condition count plot**



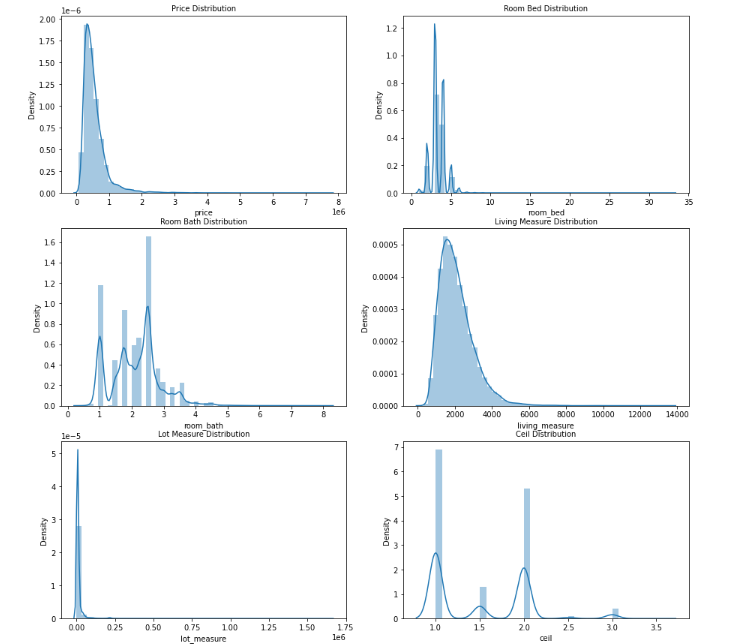
**Figure 6: Quality count plot**

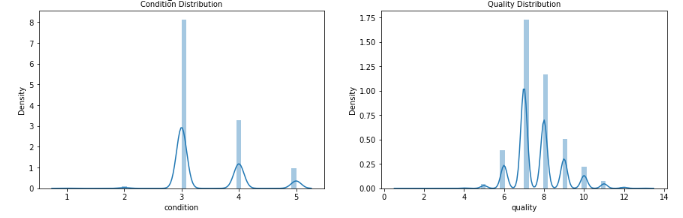


**Figure 7: Coast count plot**

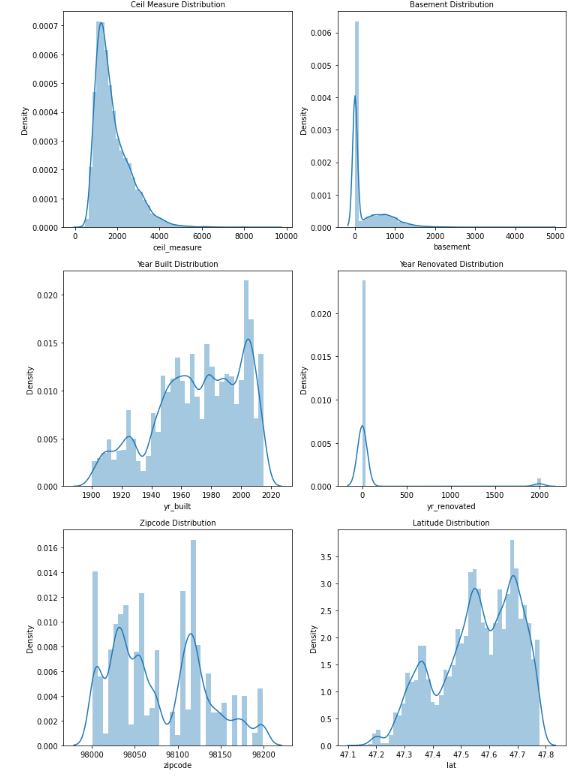


**Figure 8: Sight count plot**

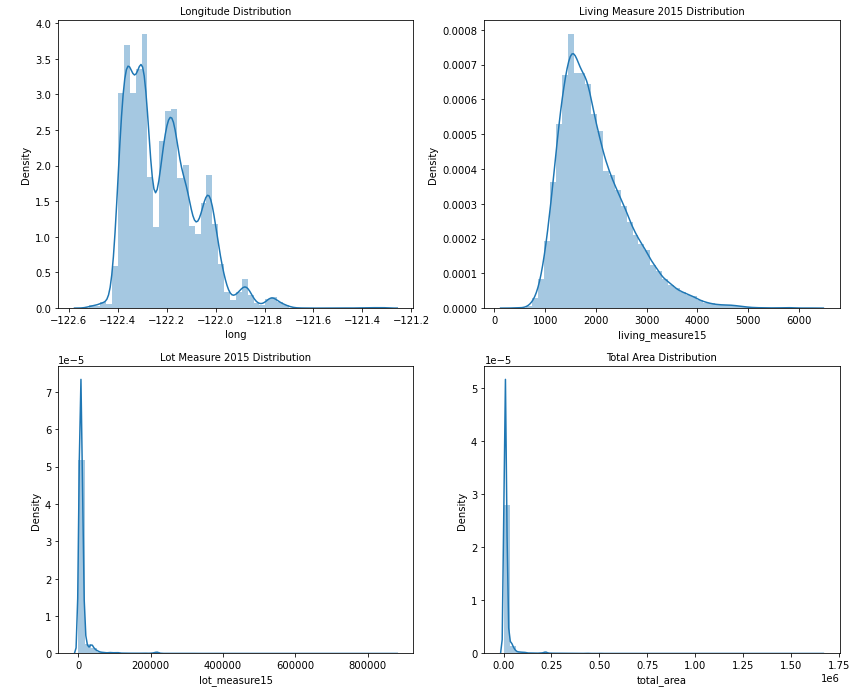




**Figure 9: Histogram set 1**



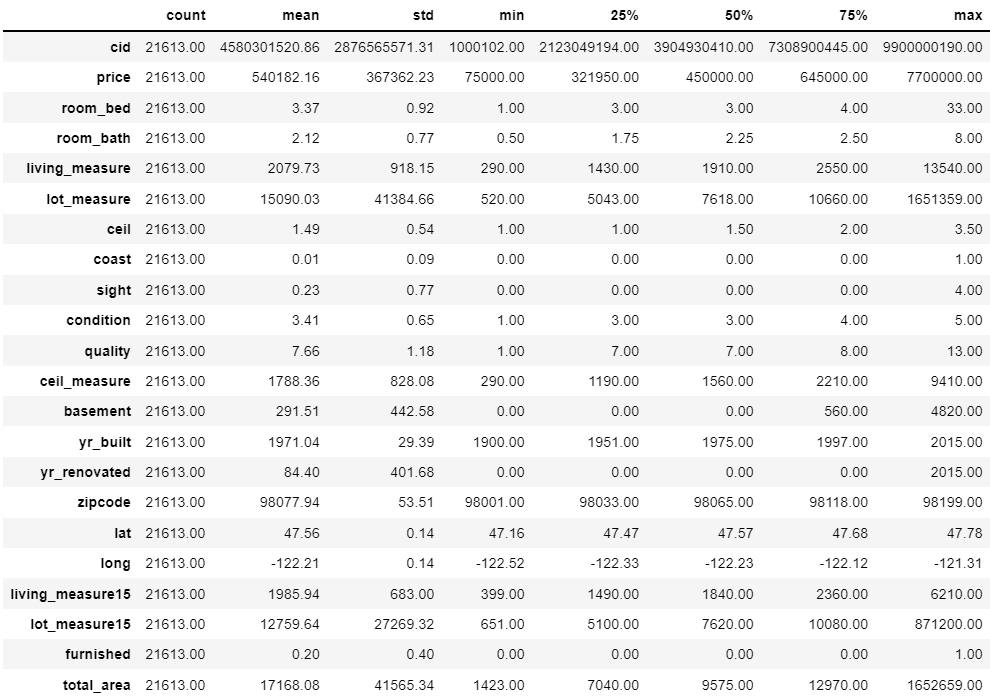
**Figure 10: Histogram set 2**



**Figure 11: Histogram set 3**

**Visual understanding from univariate analysis** **–**

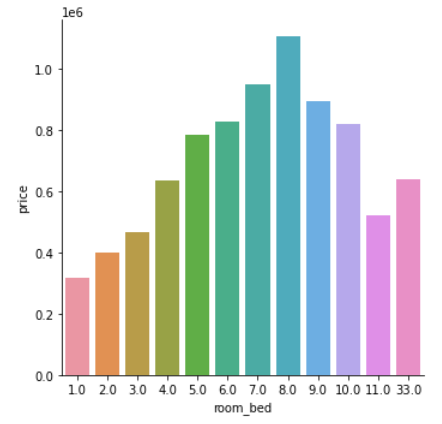
* It is not possible to have a house with no bedroom and is treated similar to missing values. House with 33 beds suggests presence of outliers. Majority of the houses consist of 3 bedrooms.
* Room bath also has 10 values with 0 values which is practically not feasible in real world and treated in the same way as for room bed. Customers prefer maximum 2 bathrooms or more based on their family size and requirements.
* Most of the houses put up for sale are not furnished which may impact the prices going ahead considering the factor that buyers prioritize furnished residents for living.
* Majority of the houses have maximum 1 floor which may represent a nuclear family falling under the category of medium income group.
* Most of the buyers or real estate agents after looking at the houses have decided to give an average rating of 3 that could certainly influence potential buyers if shared on an online portal.
* Only 13 properties have been able to attain the best grades.
* Houses mostly are not near to any waterbody which might be due to soaring prices of those in proximity to coast.
* Maximum customers are yet to view the property which might be causing delay in the process of finalizing the sale price.
* Most of the residents have no basement with 8487 properties having basement the reason for which may be huge cost of construction or lack of requirement or interest in the same.



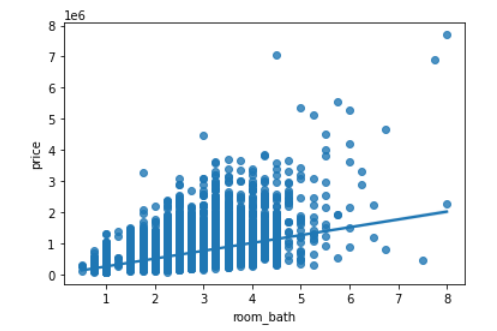
**Table 1: Dataset description**

**Non-Visual understanding from univariate analysis** **–**

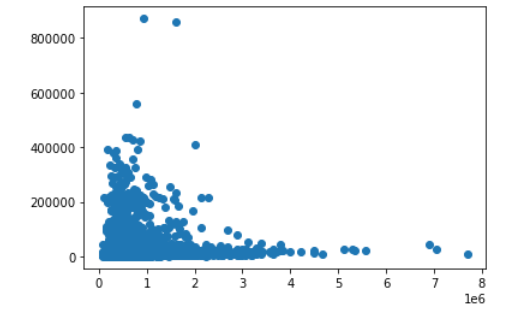
* price distribution seems to be right skewed as mean > median
* room bed distribution is also right-skewed since mean > median
* room bath follows left skewed distribution due to mean < median
* Since mean > median, living measure is right skewed
* lot measure is highly skewed to the right
* ceil is normally distributed with mean and median having identical values
* coast is a categorical variable with binary values
* sight is positively skewed
* condition has values ranging from 1 to 5 with 1 indicating poor and 5 signifying the best condition of the house
* ceil measure, basement, zip code, long, living measure15, lot measure15 and total area follow positively skewed distribution
* yr. built and lat follow negatively skewed distribution
* furnished and yr. renovated are categorical variables with 2015 being the year when house was renovated while majority of the houses have no renovation done till date
* Presence of outliers can also be suspected based on the wide range of values



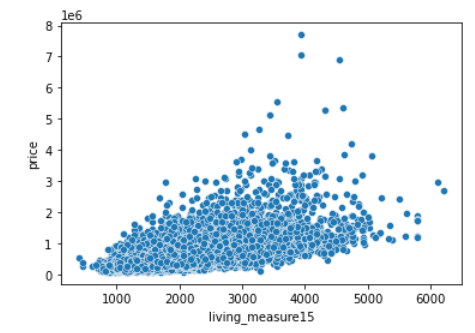
**Figure 12: Room bed vs price**



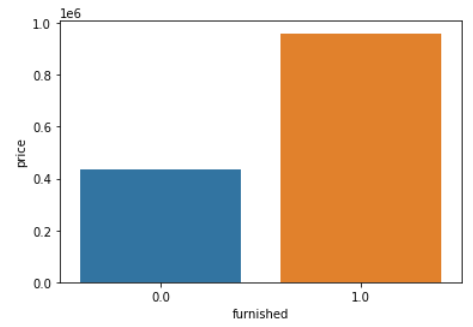
**Figure 13: Room bath vs price**



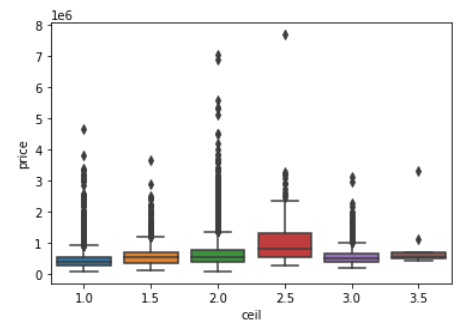
**Figure 14: Lot measure15 vs price**



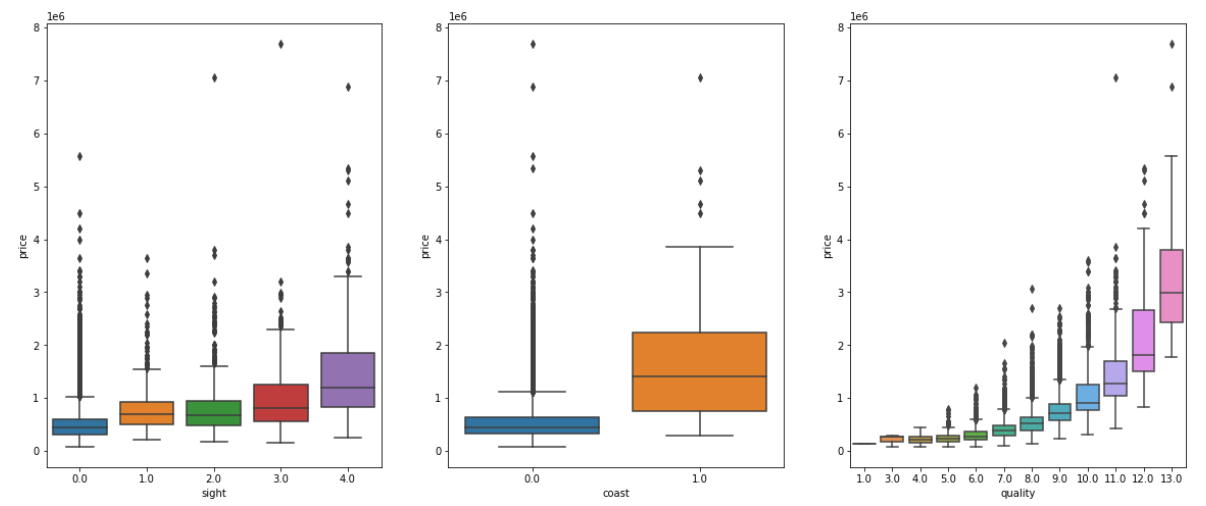
**Figure 15: Living measure15 vs price**



**Figure 16: Furnished vs price**



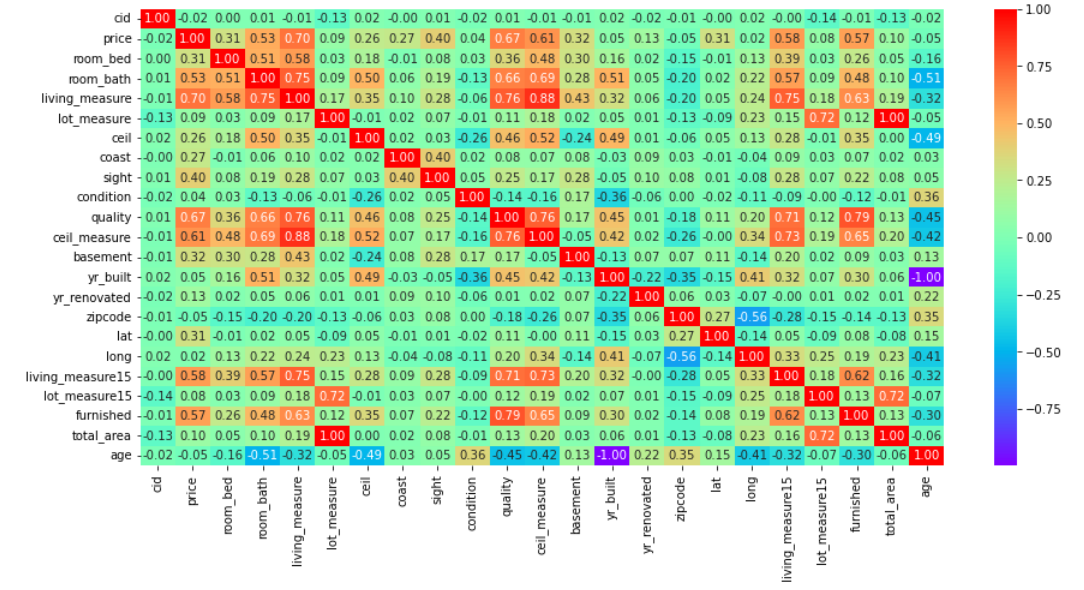
**Figure 17: Ceil vs price**



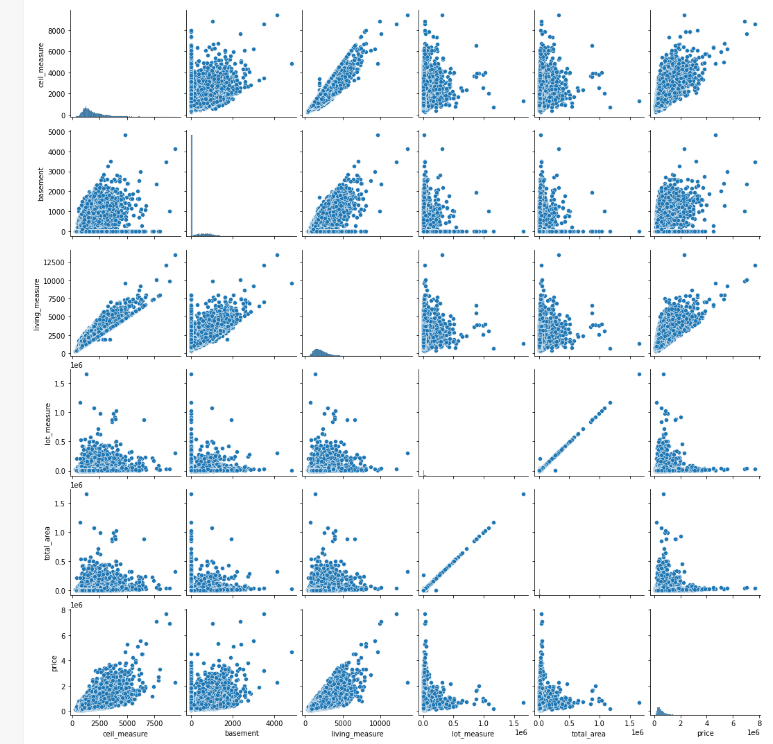
**Figure 18: Three features vs price**

**Visual understanding from bivariate analysis** **–**

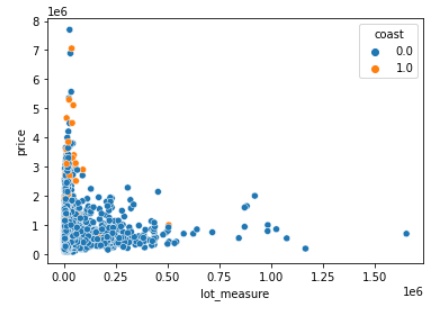
* Houses with 8 bedrooms have the highest price while 1-bedroom residents are yielding low prices.
* The observation points seem to be universally spread for 50% of room bath data while it fizzles out for the rest. Prices are increasing with rise in bathrooms and the regression line is a proof of the statement.
* Average prices are quite similar for floors in the range of 1-2. The price slightly rises for floors greater than 2. The outliers are acceptable in the data due to its categorical nature.
* Median prices are highest for houses visited maximum number of times by buyers. Sea facing houses are more likely to get sold and yield handsome returns for the sellers. Customers are willing to select highly rated houses and thereby giving maximum prices for the same.
* Scatterplot shows cloud like relationship between price and living measure15 and lot measure15 and no conclusion can be made out of these plots.



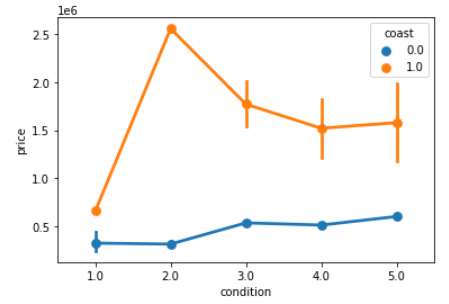
**Figure 19: Correlation plot**



**Figure 20: Pair plot**



**Figure 21: Multivariate scatterplot**



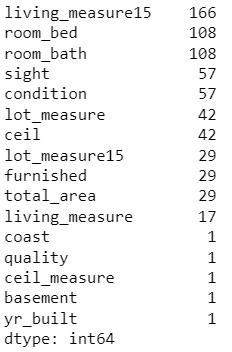
**Figure 22: Multivariate point plot**

**Visual understanding from multivariate analysis** **–**

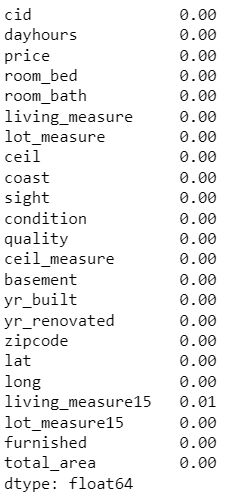
* House condition with no waterside view is not influencing prices whereas house with condition rating of 2 and coast view are experiencing a price shoot up is a matter of research.
* Price has strong correlation with room bath, living measure, quality, ceil measure, living measure15 and furnished as it ranges from 0.5 to 0.75.
* Some variables are showing negative correlations which may be due to the fact that it was performed prior to outlier removal.
* The numerical variables in pair plot are having linear relationship with target variable and hence could be important factors in model design going forward.
* Initial lot measure is densely populated with very minute coast view observations.
* Multicollinearity behaviour is also observed for few variables like total area and lot measure, ceil measure and living measure, quality and furnished and quality and ceil measure which display significant correlation and either of these related variables need to be dropped post checking their VIF values.
* Cid has poor relation with price as well as other variables and it is better to get rid of it before model building.

1. **Data Cleaning and Pre-processing**

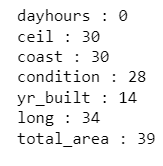
**Approach used for identifying and treating missing values and outlier treatment**



**Figure 23: Missing values per column**

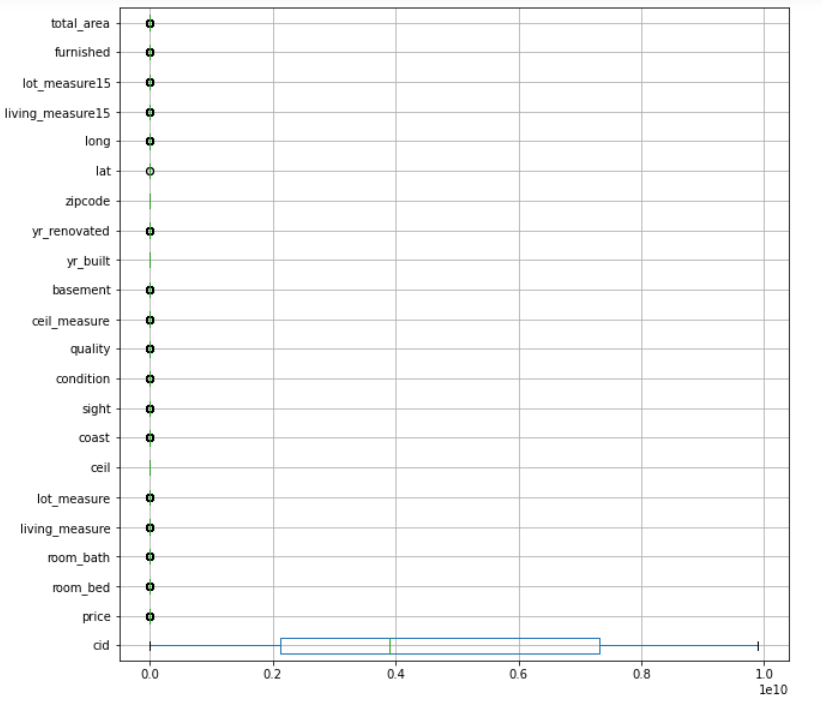


**Figure 24: Percentage of null values**



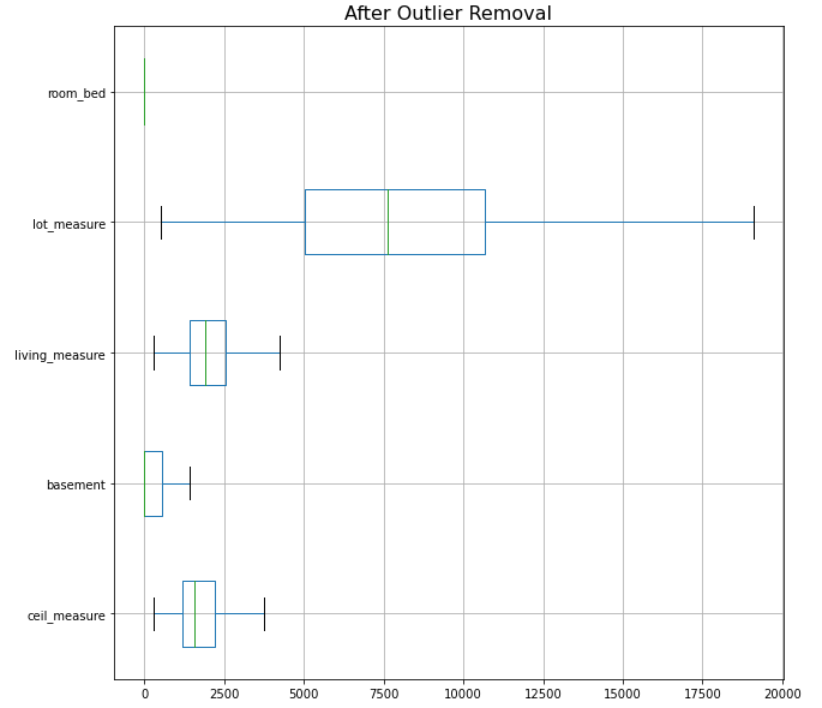
**Figure 25: Special character count**

* There is total 689 missing values in the dataset.
* For categorical data, if missing value percentage is less than 2%, we impute with mode and for numeric data we impute with median values.
* Percentage wise null value distribution makes it clear that we need to impute with median and mode values respectively for numerical and categorical fields.
* Few columns are inappropriately labelled as object due to presence of special character “$”.
* Ceil, long and total area special characters are replaced with median while the remaining are replaced with mode values.
* Living measure15, room bed, room bath, living measure, lot measure, lot measure15, total area, basement, ceil measure, quality, condition, coast, sight and furnished null values are imputed with median values.
* Ceil is imputed with median and yr. built with mode. Here we conclude our missing value treatment.



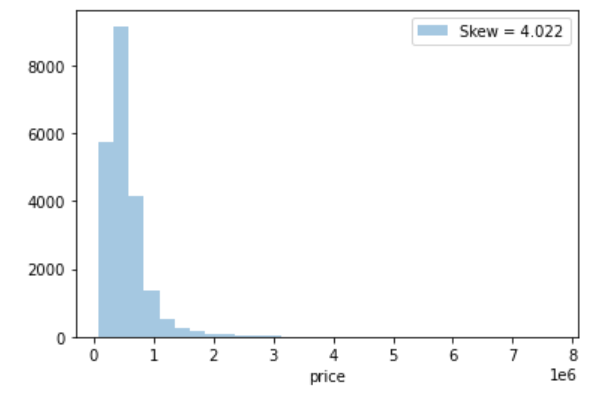
**Figure 26: Outlier detection boxplot**

* Boxplot is used to identify outliers. We have substituted the outlier present on the higher side with the 95th Percentile and the outliers which are present on the lower side with the 5th percentile.
* Target variable relationship has already been checked with outlier containing columns and those having positive correlation are treated.
* As seen previously room bed 33 was an outlier and hence it was treated.
* Lot measure, living measure, basement and ceil measure have good relation with target variable and hence they were treated.
* Rest of the variable though contain outliers were not treated as they may lead to loss of valuable data.

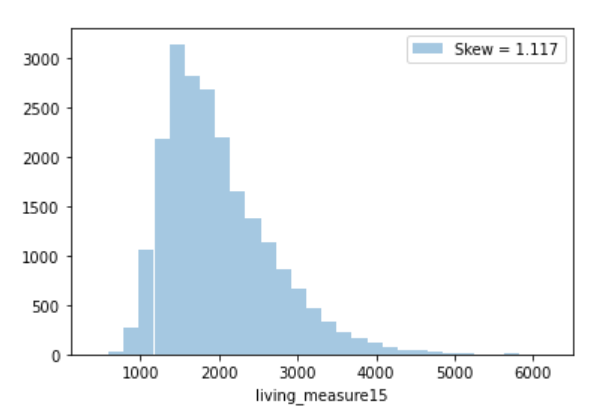


**Figure 27: Boxplot post outlier treatment**

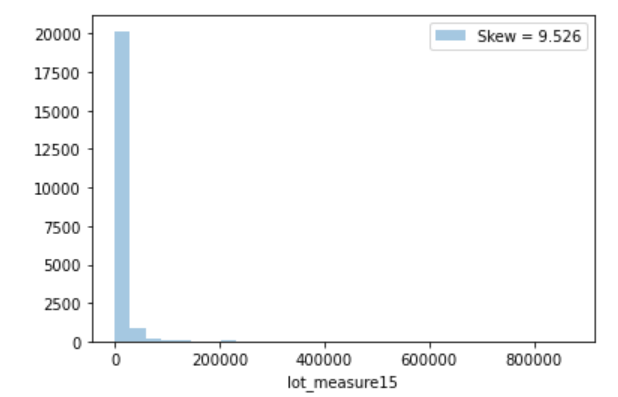
**Need for variable transformation**



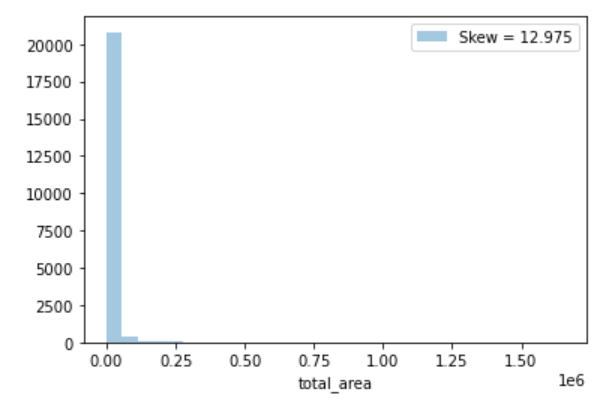
**Figure 28: Price skewness**



**Figure 29: Living measure15 skewness**

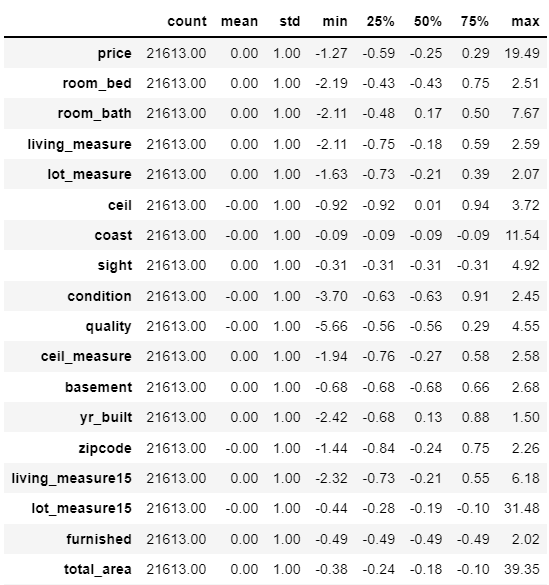


**Figure 30: Lot measure15 skewness**

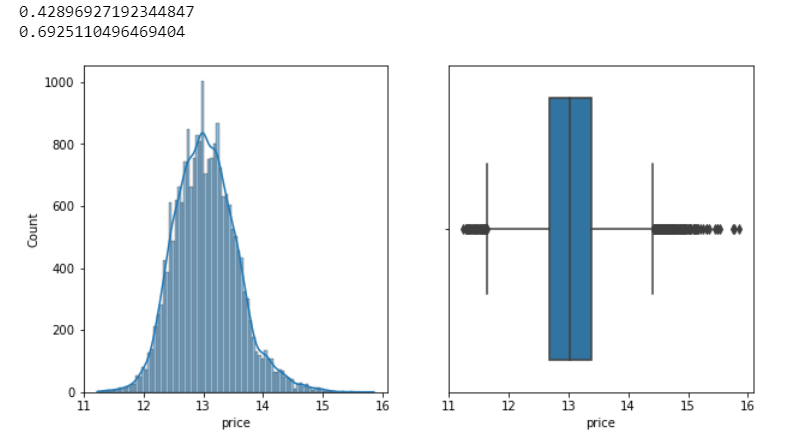


**Figure 31: Total area skewness**

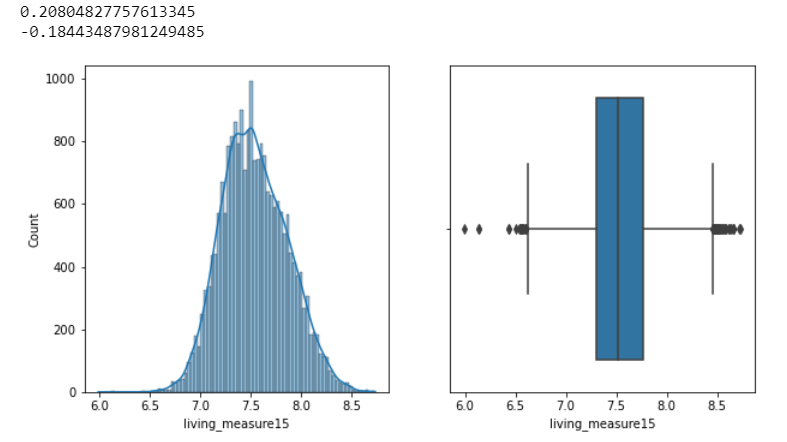
* Numerical columns like price, living measure15, lot measure15, basement and total area have skewness and kurtosis more than 1.
* Skewness and kurtosis should be in the range of -1 to +1. These variables need to be transformed to improve skewness and kurtosis.
* Before applying variable transformation, scaling is performed since variables are of different scales. Standard scaler technique scales the data in such a way that the mean value of the features tends to 0 and the standard deviation tends to 1.
* Log transformation was performed to convert the features to follow normal distribution.
* Dummy encoding is used to transform categorical columns like coast and furnished.



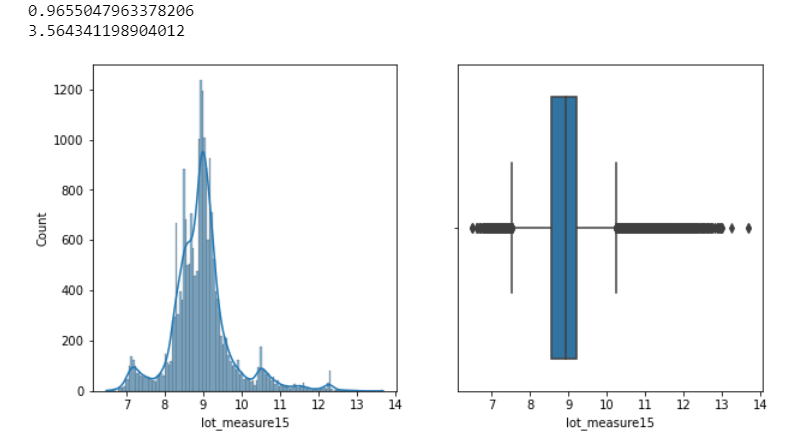
**Table 2: Description of scaled data**



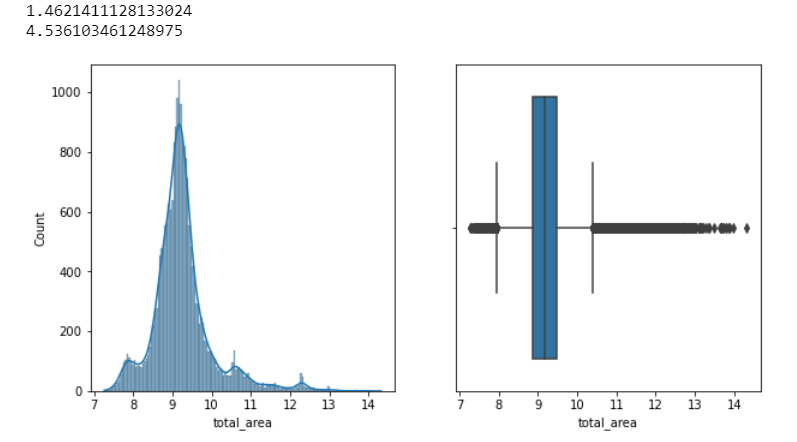
**Figure 32: Price log transformation**



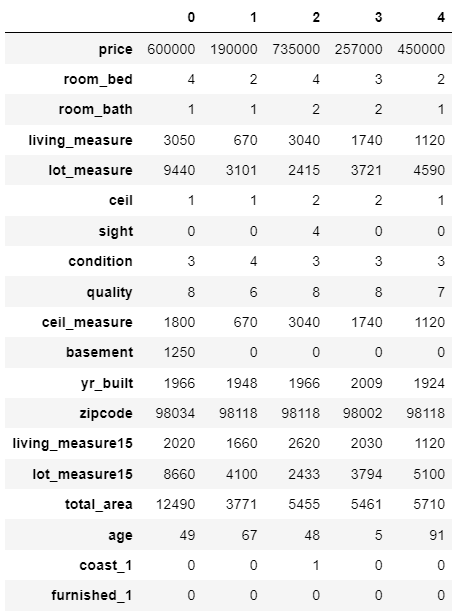
**Figure 33: Living measure15 log transformation**



**Figure 34: Lot measure15 log transformation**



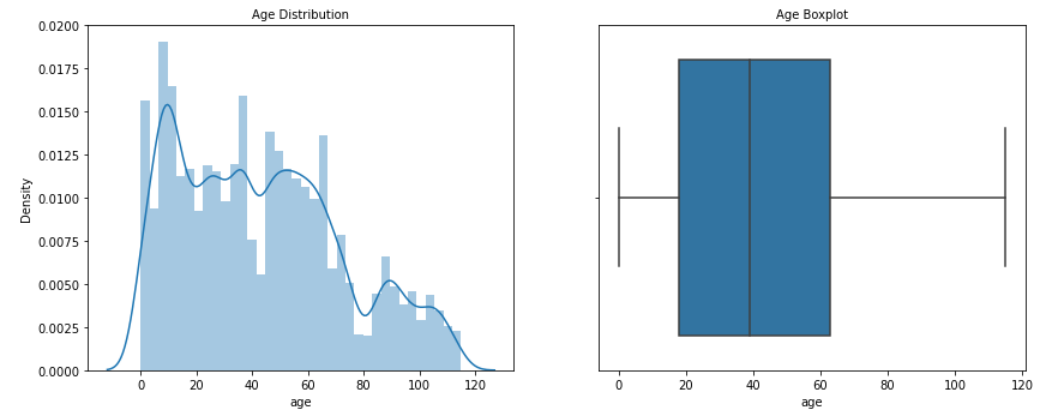
**Figure 35: Total area log transformation**



**Table 3: Dataset head post encoding**

**Variables removed or added and why**

Age variable is added to the dataset which is the difference between extracted year from day hours and year built.



**Figure 36: Age distribution**

Cid, day hours, lat, long and yr. renovated were dropped from the analysis as they have no relation with price.

1. **Model building**

* Price is the target variable used to extract training and testing sets.
* The train and test samples are divided into 70:30 ratio.
* The various models that were built for the problem –
* Multiple linear regression
* Ridge and lasso regression
* Random Forest
* KNN
* The models are run and we look at R^2 values.
* The best model is selected based on the performance metrics like root mean square error, mean square error and mean absolute error.

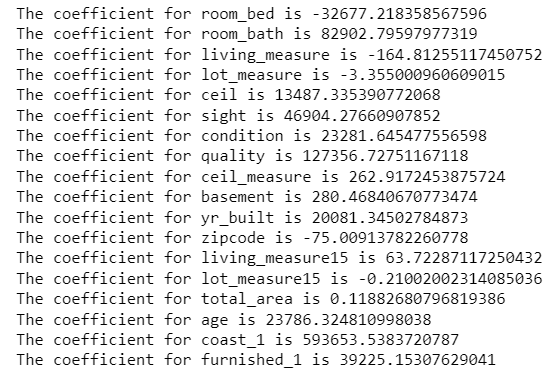
**Clear on why was a particular model(s) chosen**

**Linear Regression**

Since it is a regression-based problem, we will start off with linear regression model which can be implemented in 2 ways – either using stats models or basic model package. Assumptions of linear regression also needs to be tested violating of which can lead to severe performance issues.

Copy all the predictor variables into X data frame. Since 'price' is dependent variable drop it. Copy the 'price' column alone into the y data frame. This is the dependent variable. Let us break the X and y data frames into training set and test set. For this we will use scikit learn package's data splitting function which is based on random function. Split X and y into training and test set in 70:30 ratio.

The linear regression function is invoked and the best fit model is found on the training data. Let us explore the coefficients of each of the independent attributes.



**Figure 37: Coefficients for the linear regression model**

Let us check the intercept for the model-



**Figure 38: Intercept for the linear regression model**

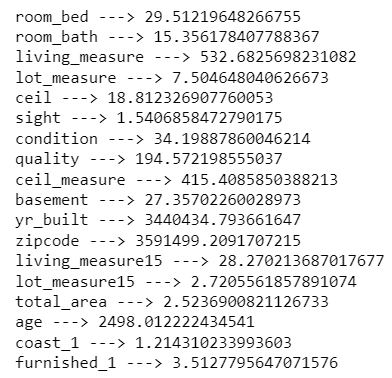


**Figure 39: Training score for the linear regression model**



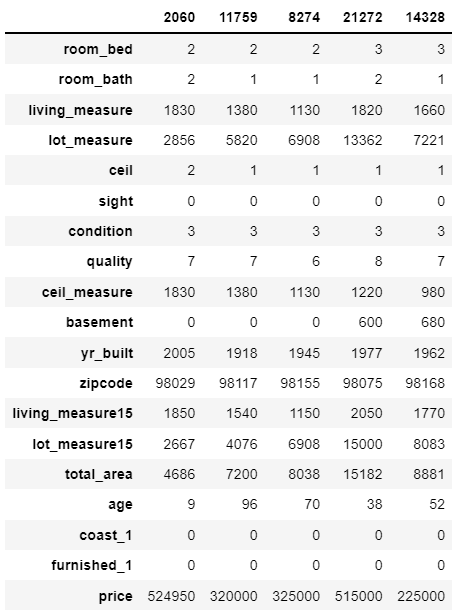
**Figure 40: Testing score for the linear regression model**

So, the base linear regression model explains 65% of the variability in Y using X.



**Figure 41: VIF for the dataset variables**

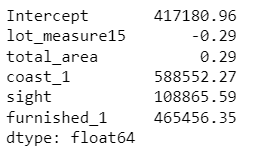
Columns with VIF > 5 is said to have high multicollinearity which violates one of the assumptions of the linear regression model and hence needs to be dropped before going to the next iteration.



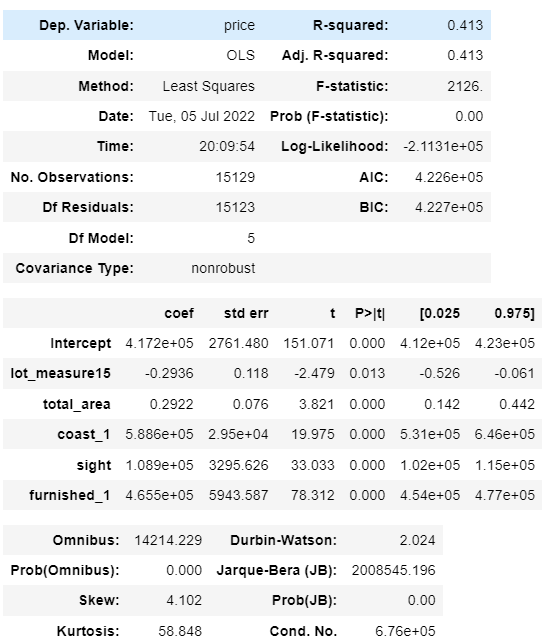
**Table 4: OLS training model head**

Formula for OLS model -

price ~ lot\_measure15+total\_area+coast\_1+sight+furnished\_1



**Figure 42: Parameter values for OLS model**



**Table 5: OLS summary**

The overall P value is less than alpha, so rejecting H0 and accepting Ha that at least 1 regression co-efficient is not 0. Here all regression co-efficient are not 0.

Inferences –

* Both the R-squared and Adjusted R squared of our model are very low. This is a clear indication that we have been able to create a very good model that is able to explain variance in price of used cars for up to 41%.
* The model is not an underfitting model.
* To be able to make statistical inferences from our model, we will have to test that the linear regression assumptions are followed.

Before we move on to assumption testing, we'll do a quick performance check on the test data.

Observations -

* Root Mean Squared Error of train and test data is identical, indicating that our model is not overfitting the train data.
* Mean Absolute Error indicates that our current model is able to predict house prices within mean error on test data.
* The units of both RMSE and MAE are same. But RMSE is greater than MAE because it penalises the outliers more.

Linear Regression Assumptions -

* No Multicollinearity
* Mean of residuals should be 0
* No Heteroscedasticity
* Linearity of variables
* Normality of error terms

1st assumption has already been checked and treated accordingly.

Mean of residuals is very close to 0. The second assumption is also satisfied.

Homoscedastic - If the residuals are symmetrically distributed across the regression line, then the data is said to homoscedastic.

Heteroscedasticity- - If the residuals are not symmetrically distributed across the regression line, then the data is said to be heteroscedastic. In this case the residuals can form a funnel shape or any other non-symmetrical shape.

We'll use Goldfeldquandt Test to test the following hypothesis

Null hypothesis: Residuals are homoscedastic Alternate hypothesis: Residuals have heteroscedasticity

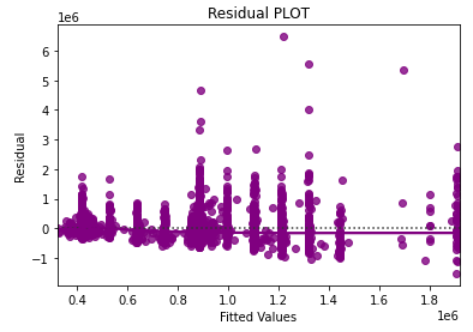
alpha = 0.05

Since p-value > 0.05 we cannot reject the Null Hypothesis that the residuals are homoscedastic.

Assumptions 3 is also satisfied by our model.

Predictor variables must have a linear relation with the dependent variable.

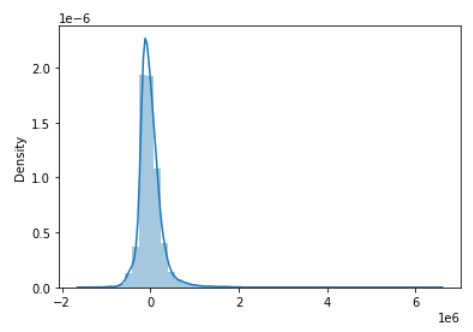
To test the assumption, we'll plot residuals and fitted values on a plot and ensure that residuals do not form a strong pattern. They should be randomly and uniformly scattered on the x axis.



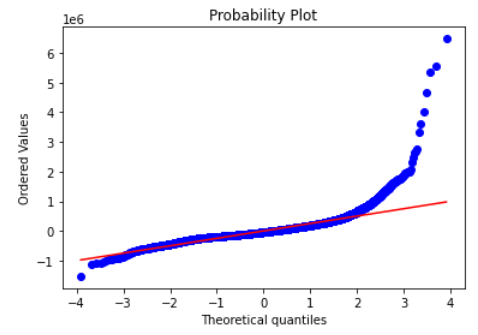
**Figure 43: Plot to check linearity**

Assumptions 4 is satisfied by our model. There is no pattern in the residual vs fitted values plot.

The residuals should be normally distributed.

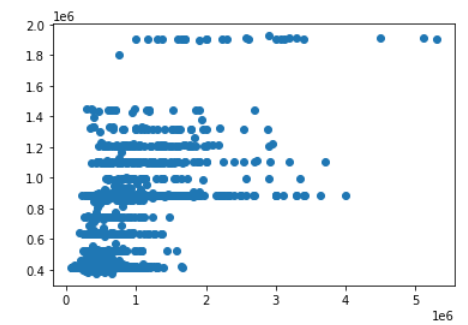


**Figure 44: Residual density plot**



**Figure 45: Residual Q-Q plot**

The residuals have a close to normal distribution. Assumption 5 is also satisfied. We should further investigate these values in the tails where we have made huge residual errors.



**Figure 46: Scatter plot for OLS model**

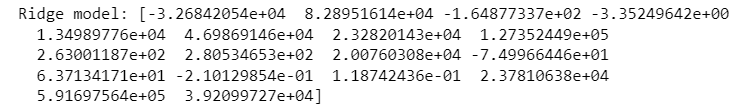


**Figure 47: Regression equation for OLS model**

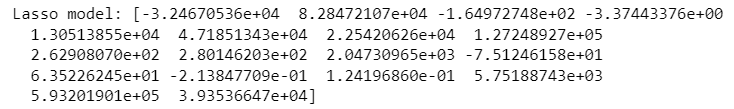
When total area increases by 1 unit, price increases by 0.29 units, keeping all other predictors constant.  
Similarly, when lot measure15 increases by 1 unit, price decreases by 0.29 units, keeping all other predictors constant.

**Ridge and Lasso regression**

We will perform ridge and lasso regression to regularize the linear regression model.



**Figure 48: Ridge model coefficient**



**Figure 49: Lasso model coefficient**



**Figure 50: Ridge model performance**



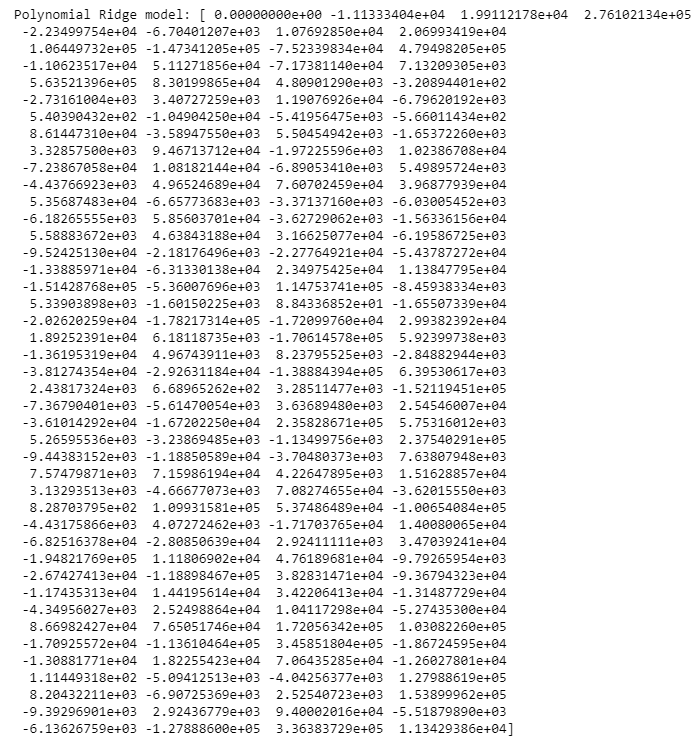
**Figure 51: Lasso model performance**

More or less similar results but with less complex models. Complexity is a function of variables and coefficients. With Lasso, we get equally good result in test and in training. Further, the number of dimensions is much less in LASSO model than ridge or un-regularized model.

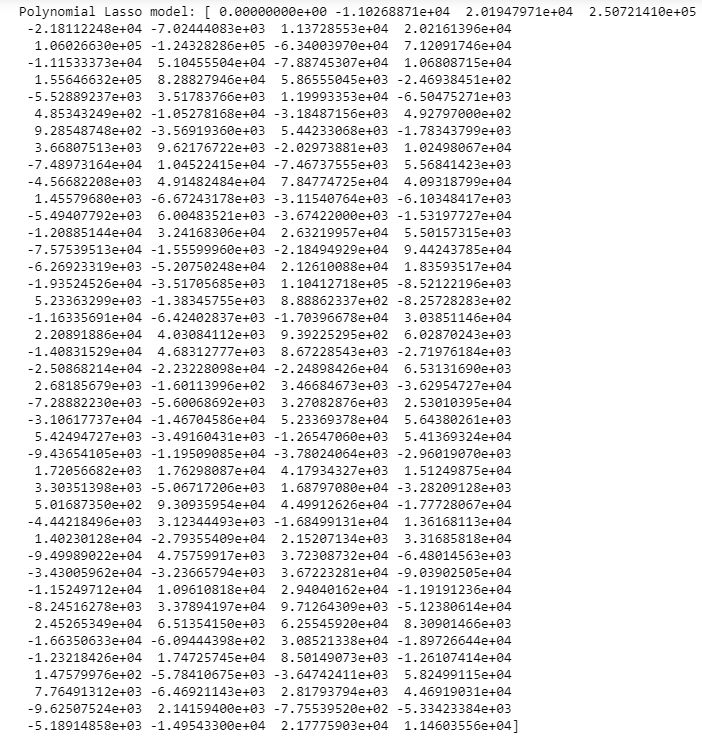
Since on many dimensions, the relationship is not really linear, let us try polynomial models (quadratic). Let us generate polynomial models reflecting the non-linear interaction between some dimensions. Fit a simple non regularized linear model on poly features after scaling the dataset.



**Figure 52: Coefficient for polynomial regression**



**Figure 53: Coefficient of polynomial ridge model**



**Figure 54: Coefficient of polynomial lasso model**



**Figure 55: Performance of polynomial ridge regression**



**Figure 56: Performance of polynomial lasso regression**

Even with polynomial function, we are getting better results.

**Random Forest**

The third model that we look at is random forest which is instantiated using random forest regressor function with maximum depth as 4 since earlier we found 4 clusters were optimum for this problem.

In random forests, each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set.

Furthermore, when splitting each node during the construction of a tree, the best split is found either from all input features or a random subset of size max features

The purpose of these two sources of randomness is to decrease the variance of the forest estimator. Indeed, individual decision trees typically exhibit high variance and tend to overfit. The injected randomness in forests yield decision trees with somewhat decoupled prediction errors. By taking an average of those predictions, some errors can cancel out. Random forests achieve a reduced variance by combining diverse trees, sometimes at the cost of a slight increase in bias. In practice the variance reduction is often significant hence yielding an overall better model.

The scikit-learn implementation combines classifiers by averaging their probabilistic prediction.



**Figure 57: Random Forest train model score**



**Figure 58: Random Forest test model score**

**K-Nearest Neighbours (KNN)**

The next model in line is KNN. For naive bayes algorithm while calculating likelihoods of numerical features it assumes the feature to be normally distributed and then we calculate probability using mean and variance of that feature only and also it assumes that all the predictors are independent to each other. Scale doesn’t matter. Performing a feature scaling in these algorithms may not have much effect.

**Neighbours-based classification is a type of instance-based learning or non-generalizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbours of each point: a query point is assigned the data class which has the most representatives within the nearest neighbours of the point.**

Generally, good KNN performance usually requires pre-processing of data to make all variables similarly scaled and centred.

Now let’s apply standard scaler on continuous columns and see the performance for KNN.

The model is built with different values of k. Instantiate learning model with default value of k=5.



**Figure 59: KNN train score for k=5**



**Figure 60: KNN test score for k=5**



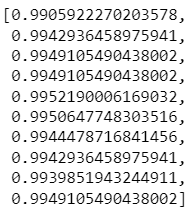
**Figure 61: KNN test score for k=7**



**Figure 62: KNN test score for k=7**

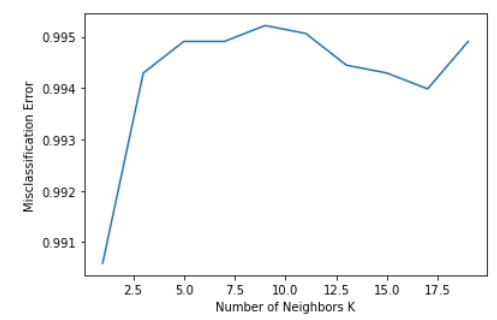
Run the KNN with no of neighbours to be 1, 3, 5 ...19 and find the optimal number of neighbours from the above list using the Misclassification error.

Misclassification error (MCE) = 1 - Test accuracy score. Calculated MCE for each model with neighbours = 1,3,5...19 and find the model with lowest MCE



**Figure 63: MCE scores**

Plot misclassification error vs k (with k value on X-axis) using matplotlib.



**Figure 64: MCE plot**

For K = 1 it is giving the best test accuracy let’s check train and test for K=3 with other evaluation metrics.



**Figure 65: Optimum KNN model training score**



**Figure 66: Optimum KNN model testing score**



**Figure 67: KNN model training score with K=3**



**Figure 68: KNN model testing score with K=3**

**As the difference between train and test accuracies is less than 10%, it is a valid model.**

**Support Vector Machine**

**Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outlier detection.**



**Figure 69: Parameters for SVM model**



**Figure 70: Training score for SVM model 1**



**Figure 71: Testing score for SVM model**

Negative coefficient of determination for both train and set samples indicates poor performance of SVM which can be improved by tuning the model.

**Bagging**

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction.



**Figure 72: Bagging parameters**

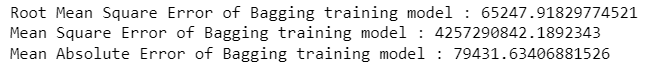


**Figure 73: Bagging training accuracy**

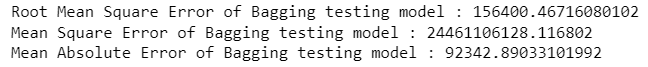


**Figure 74: Bagging testing accuracy**

The model suffers from overfitting as it performs excellently on trained data but not as much on test data.



**Figure 75: Bagging train performance metrics**



**Figure 76: Bagging test performance metrics**

The performance metrics of the model suggests a mixed bag of highs and lows for both types of samples.

**Ada Boosting**

The module sklearn. ensemble includes the popular boosting algorithm AdaBoost, introduced in 1995 by Freund and Schapire

The core principle of AdaBoost is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction.

The number of weak learners is controlled by the parameter estimators. The learning rate parameter controls the contribution of the weak learners in the final combination. By default, weak learners are decision stumps. Different weak learners can be specified through the base estimator parameter. The main parameters to tune to obtain good results are estimators and the complexity of the base estimators (e.g., its depth maximum depth or minimum required number of samples to consider a split minimum samples split).



**Figure 77: AdaBoost test score**



**Figure 78: AdaBoost train score**



**Figure 79: AdaBoost RMSE scores**



**Figure 80: AdaBoost MSE scores**



**Figure 81: AdaBoost MAE scores**

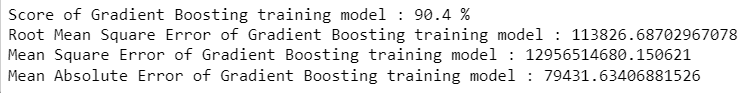
Both the training and testing coefficient of determination shows that the model suffers from underfitting.

**Gradient Boosting**

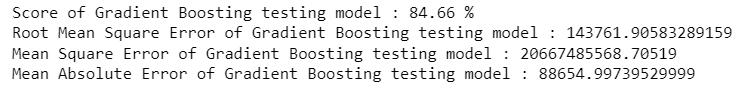
**Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.**



**Figure 82: Gradient Boosting parameters**



**Figure 83: Gradient Boosting training performance**



**Figure 84: Gradient Boosting testing performance**

The model performance is the best amongst all the models built till now and can be considered as the final and optimum model for the problem in hand.

The most optimum model selected for house price prediction problem is gradient boosting algorithm. Good performance measures on train and test sets lead to an easy decision to choose the algorithm.

**Effort to improve model performance**

Model tuning measures like hyperparameter tuning or regularizations were performed on optimum model to improve performance metrics but it led to underwhelming results.

1. **Model validation**

**How was the model validated?**

Let us check the sum of squared errors by predicting value of y for test cases and subtracting from the actual y for the test cases.



**Figure 85: Training MSE for the linear regression model**



**Figure 86: Testing MSE for the linear regression model**

Under root of mean squared error is standard deviation i.e., average variance between predicted and actual.



**Figure 87: Training RMSE for the linear regression model**



**Figure 88: Testing RMSE for the linear regression model**



**Figure 89: Training MAE for the linear regression model**



**Figure 90: Testing MAE for the linear regression model**

So, there is average of 216884 (roundoff) price difference from real price on an average.

R^2 is not a reliable metric as it always increases with addition of more attributes even if the attributes have no influence on the predicted variable. Instead, we use adjusted R^2 which removes the statistical chance that improves R^2.Scikit does not provide a facility for adjusted R^2, so we use statistical model, a library that gives results similar to what you obtain in R language. This library expects the X and Y to be given in one single data frame.



**Figure 91: MSE of OLS training data**



**Figure 92: MSE of OLS testing data**



**Figure 93: RMSE of OLS training data**



**Figure 94: RMSE of OLS testing data**



**Figure 95: MAE of OLS training data**



**Figure 96: MAE of OLS testing data**



**Figure 97: MSE of ridge training model**



**Figure 98: MSE of ridge testing model**



**Figure 99: MSE of lasso training model**



**Figure 100: MSE of lasso testing model**



**Figure 101: RMSE of ridge training model**



**Figure 102: RMSE of ridge testing model**



**Figure 103: RMSE of lasso training model**



**Figure 104: RMSE of lasso testing model**



**Figure 105: MAE of ridge training model**



**Figure 106: MAE of ridge testing model**



**Figure 107: MAE of lasso training model**



**Figure 108: MAE of lasso testing model**



**Figure 109: MSE train for polynomial ridge regression**



**Figure 110: MSE test for polynomial ridge regression**



**Figure 111: RMSE train for polynomial ridge regression**



**Figure 112: RMSE test for polynomial ridge regression**



**Figure 113: MAE train for polynomial ridge regression**



**Figure 114: MAE test for polynomial ridge regression**



**Figure 115: MSE train for polynomial lasso regression**



**Figure 116: MSE test for polynomial lasso regression**



**Figure 117: RMSE train for polynomial lasso regression**



**Figure 118: RMSE test for polynomial lasso regression**



**Figure 119: MAE train for polynomial lasso regression**



**Figure 120: MAE test for polynomial lasso regression**



**Figure 121: Random Forest train RMSE**



**Figure 122: Random Forest test RMSE**



**Figure 123: Random Forest train MSE**



**Figure 124: Random Forest test MSE**



**Figure 125: Random Forest train MAE**



**Figure 126: Random Forest test MAE**



**Figure 127: KNN model RMSE train**



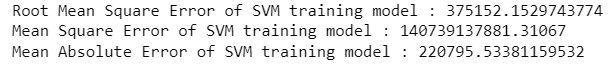
**Figure 128: KNN model RMSE test**



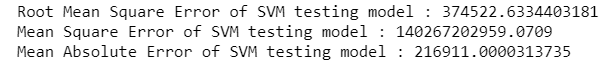
**Figure 129: KNN model MAE train and test**



**Figure 130: KNN model MSE train and test**



**Figure 131: SVM training performance**



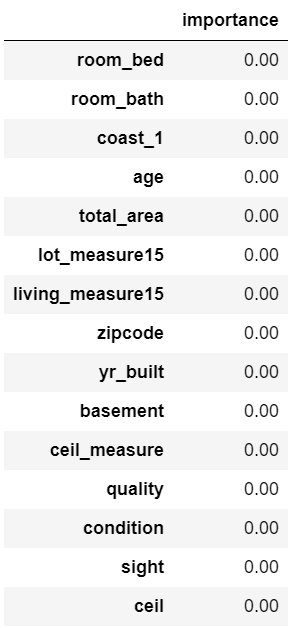
**Figure 132: SVM testing performance**

**6) Final interpretation / recommendation**

**Interpretation of the model(s)**

* Linear regression base model performs better on testing data than training data but the scores are pretty decent.
* KNN model has high accuracy on training and poor scores in testing a clear indication of an overfit model.
* Ridge and lasso regression perform the best among all the models having identical training and testing scores which were improved by leaning towards quadratic models.
* Random Forest scores on training and testing samples prove it to be an average model at best.
* Support Vector Machine is the worst performer of the lot.
* Performance measures are enhanced or decreased according to the model scores.

We will look at feature importance for the given dataset and accordingly help potential sellers/buyers to quote suitable prices mutually agreed upon by both parties.



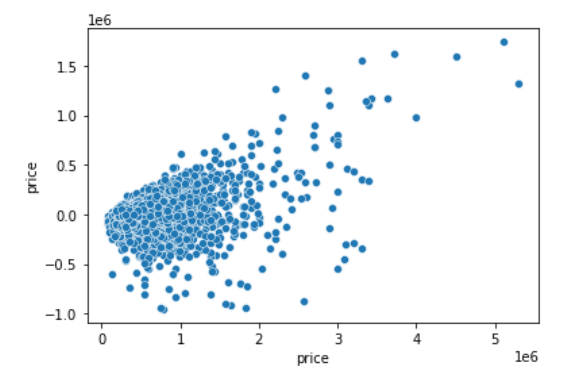
**Table 6: Feature importance of the model**



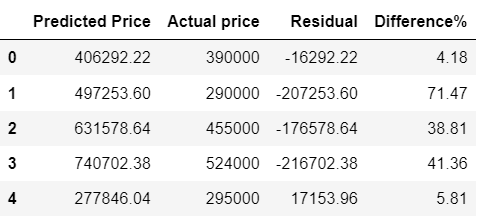
**Table 7: Model comparison**

**Detailed Recommendations for client based on analysis** -

* Our final Gradient Boosting model has a MAPE of 15% on the test data, which means that we are able to predict within 15% of the price value. This is a very good model and we can use this model in production.
* With our linear regression model, we have been able to capture ~85 variation in our data.
* The model indicates that the most significant predictors of price of houses are number of bedrooms, number of bathrooms, age, total area and zip code.
* We will have to analyse the cost side of things before we can talk about profitability in the business. We should gather data regarding that.
* The next step post that would be to cluster different sets of data and see if we should make multiple models for different zip code/house age.
* Bedroom and bathroom presence is the basic requirement for any customer looking for a house hence justifiably it tops the in-demand features.
* Houses present near the coast is a priority surprisingly hence real estate businesses should target potential sellers present in this location.
* Condition and quality of the house languishes at the bottom therefore customers are not making decisions based on other’s feedback as expected.
* Business should be keeping in mind the significant predictors before taking major decisions.
* Business should focus on the locations in USA as most of the zip codes available are centred around that location.



**Figure 133: Actual vs predicted price for optimum model**



**Table 8: Actual and predicted price table for optimum model**