Machine Learning Capstone Project on House Price Prediction using Advanced Regression Techniques

**Authors:** Indra Chatterjee, Manju Kiruthiga, Ranganathan Sriraman, Sunil Kumar, Shyamla

**Project Mentor:** Mr. Sanjoy Krishna Ghosh

**Project Coordination:** Ms. Rashika

Contents

[Figures 3](#_Toc93169859)

[Acknowledgement 4](#_Toc93169860)

[Chapter 1: INTRODUCTION 5](#_Toc93169861)

[1.1 Problem statement 5](#_Toc93169862)

[1.2 Background 5](#_Toc93169863)

[1.3 Dataset (Feature descriptions) 5](#_Toc93169864)

[Chapter 2: Planning & Execution 7](#_Toc93169865)

[2.1 Project Plan 7](#_Toc93169866)

[2.2 Flow Chart 8](#_Toc93169867)

[Chapter 3: Data Insights - Exploratory Data Analysis 9](#_Toc93169868)

[3.1 Five Point Summary 9](#_Toc93169869)

[3.2. EDA and Visualizations 10](#_Toc93169870)

[3.2.1: Univariate and Bivariate Analysis (Selective features) 10](#_Toc93169871)

[3.2.2: Geographical data & Observations 13](#_Toc93169872)

[Chapter 4: Model Selection and Building 20](#_Toc93169873)

[4.1 Selection of Model 20](#_Toc93169874)

[4.2 Model Baseline 20](#_Toc93169875)

[4.2.1 Data Segregation 20](#_Toc93169876)

[4.2.2 Feature Scaling 20](#_Toc93169877)

[4.2.3 Model Training and Testing 21](#_Toc93169878)

[4.2.4 Baseline 21](#_Toc93169879)

[Chapter 5: Model Tuning and Cross Validation 25](#_Toc93169880)

[5.1 Grid Search CV 25](#_Toc93169881)

[5.2 Random Search CV 26](#_Toc93169882)

[5.3 Principal component analysis (PCA) 26](#_Toc93169883)

[5.4 Pipelining 27](#_Toc93169884)

[Chapter 6: Pickling and Model deployment 28](#_Toc93169885)

[Chapter 7: Conclusion 29](#_Toc93169886)

[Bibliograph 30](#_Toc93169887)

Figures

[Figure 1 Snapshot of the Gantt chart of the project plan for illustration purposes 8](#_Toc91339093)

[Figure 2 Project Flow Chart 9](#_Toc91339094)

[Figure 3 Five-point statistical summary of the given dataset 10](#_Toc91339095)

[Figure 4 Distribution of Price 11](#_Toc91339096)

[Figure 5 Bivariate Plots regression plots (Price) 13](#_Toc91339097)

[Figure 6 Box Plot (Showing outliers) 13](#_Toc91339098)

[Figure 7 Plain Map of Seattle city. 15](#_Toc91339099)

[Figure 8 Dataset plotted on a map 15](#_Toc91339100)

[Figure 9 Heat Map (Correlation) 17](#_Toc91339101)

[Figure 10 A snapshot of the Variable inflation factor of the Features. 18](#_Toc91339102)

[Figure 11 A Box plot illustrating the outlier in lot\_measure feature.. 19](#_Toc91339103)

[Figure 12 Histogram of lot\_measure vs. frequency before & after the anomaly handling 19](#_Toc91339104)

# Acknowledgement

First and foremost, we would like to express our special thanks of gratitude to **Mr. Sanjoy Krishna Ghosh (Project Mentor)** who guided us with this project and has given innumerable, invaluable advice throughout. His guidance helped tremendously to the successful completion of this project.

Besides, we would like to thank Ms. Rashika (Project coordinator), Great Learning institution to have given us this golden opportunity to do a wonderful project on the topic of House Price Prediction using Machine Learning.

Thank you.

# Chapter 1: INTRODUCTION

## 1.1 Problem statement

Creation of Intelligent Regression based data model to predict house/home prices on basis of sales data in Seattle region from 2014 to 2015. Data models take into account various features like area, location, amenities and condition

## 1.2 Background

The dataset given to us is a House price sale data based from Seattle, WA during 2014-15. We will be using **supervised machine learning** algorithms since the given dataset has input variables with known output price. The outcome of our project is to make predictions on the sales prices of the houses of Seattle with the dataset provided.

In machine learning (ML), statistical methods are used to empower machines to learn without being programmed explicitly. The field focuses on letting algorithms learn from the provided data, collect insights, and make predictions on unanalyzed data based on the gathered information.

In general, ML is based on three key models of learning algorithms:

● Supervised learning - a dataset is present with inputs and known outputs

● Unsupervised learning - the machine learns from a dataset that comes with input variables only

● Reinforcement learning - algorithms are used to select an action

## 1.3 Dataset (Feature descriptions)

The dataset contains House Sale data of Seattle, WA area from 2014-May through 2015-May. The shape of the dataset is 21613\*23 (23 attributes and 21613 observations). Below information would describe the features given.

Price is the target attribute and the remaining 22 Features acts as predictors.

*Table 1- illustrates the Feature Name and a high-level description of the Features.*

|  |  |  |
| --- | --- | --- |
| **S.No** | **Feature Name** | **Feature Description** |
| 1 | cid | A unique 7 to 10 digit ID representing the house property. This column is a transaction key (possibly auto-generated) & do not add much to the price prediction. |
| 2 | dayhours | Represents when the date/time when the sale happened. This column do not add much to the price prediction. |
| 3 | price | Selling price of the house. This will be our target variable. |
| 4 | room\_bed | Represents number of bedrooms in the house. |
| 5 | room\_bath | Represents number of bathrooms in the house.   * 0.25 bathroom is a bathroom that has either a sink, a shower, toilet or a bathtub * 0.5 bathroom is a half bath/powder room typically a Toilet and sink * 0.75 is a bathroom that contains one sink, one toilet and a shower or a bath |
| 6 | living\_measure | Represents square footage of living area in the home (carpet area) |
| 7 | lot\_measure | Represents square footage of the lot |
| 8 | ceil | Represents number of floors/levels in the house |
| 9 | coast | Is a boolean variable representing whether the house has a water front or not |
| 10 | sight | Is a boolean variable representing whether the house has been viewed by potential clients or not |
| 11 | condition | Represents overall condition of the house on a scale of 5 |
| 12 | quality | Grade given to housing unit based on grading system ranges from 1 to 13 |
| 13 | ceil\_measure | Represents square footage of the house except basement |
| 14 | basement | Represents square footage of the basement |
| 15 | yr\_built | Year when this house was built originally, it basically represents the age of the house |
| 16 | yr\_renovated | Year when the house was renovated. We assume that the last renovation year is captured in the data set |
| 17 | zipcode | Represents zipcode of the property |
| 18 | lat | Represents latitude of the property |
| 19 | long | Represents longitude of the property |
| 20 | living\_measure15 | Represents square footage of living area in the home (carpet area) after renovation 2015 |
| 21 | lot\_measure15 | Represents square footage of lot area in the home after renovation 2015 |
| 22 | furnished | Is a boolean variable representing whether the property is furnished (personal property) |
| 23 | total\_area | Represents total sum of both living and lot measure |

# Chapter 2: Planning & Execution

## 2.1 Project Plan

To provide an overall understanding of the progress that needs to be made in order to complete the project achieving all milestones we have made use a Gantt chart. The project plan will be reviewed and updated periodically, to be maintained in a centralized cloud GitHub repository.

A snapshot of the project plan is given below for illustration purposes.

This Project spans for 7 weeks, starting 28-Nov-2021 and ends on 07-Jan-2022. The progress of the project is tracked weekly. The 2 yellow highlights denote the Milestone delivery.

* 19-Dec is the milestone for the Interim Report submission &
* 07-Jan is the milestone for the Final project code and Final Report submission.

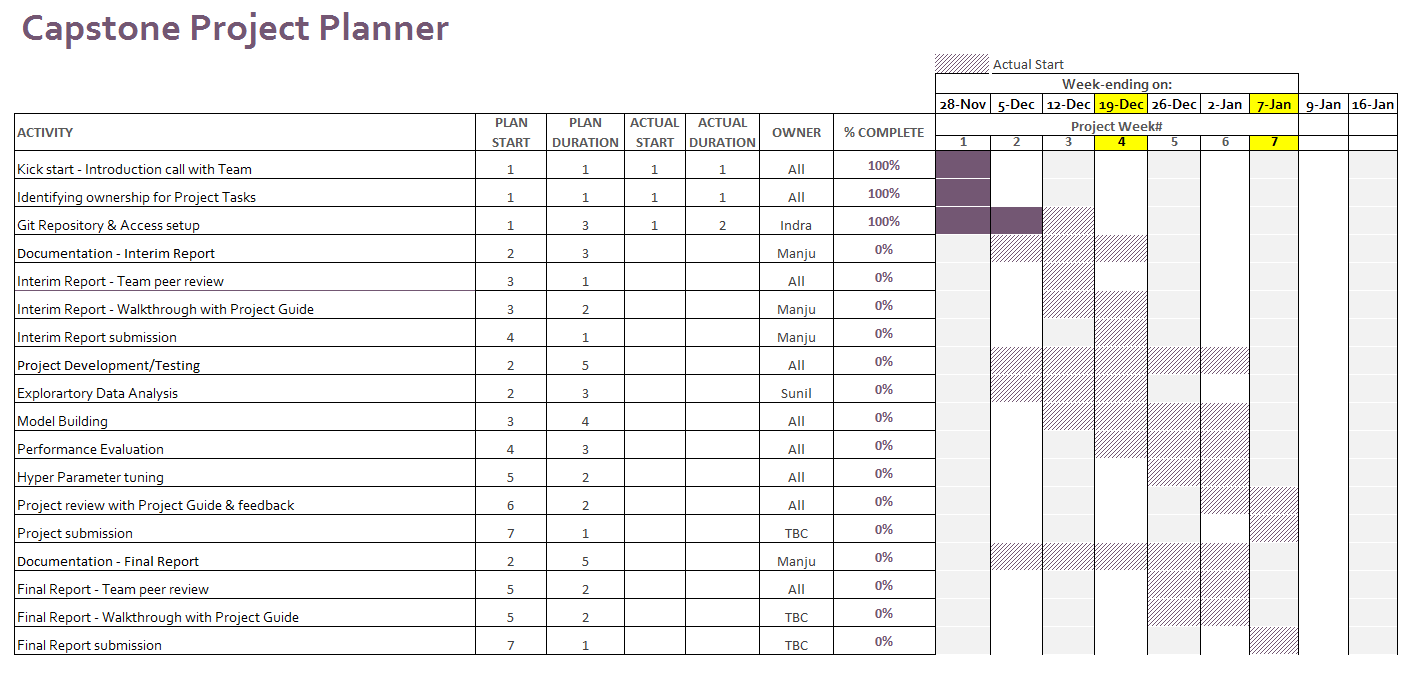


Figure 1 Snapshot of the Gantt chart of the project plan for illustration purposes

****

## 2.2 Flow Chart

Flow chart below depicts the project life cycle.

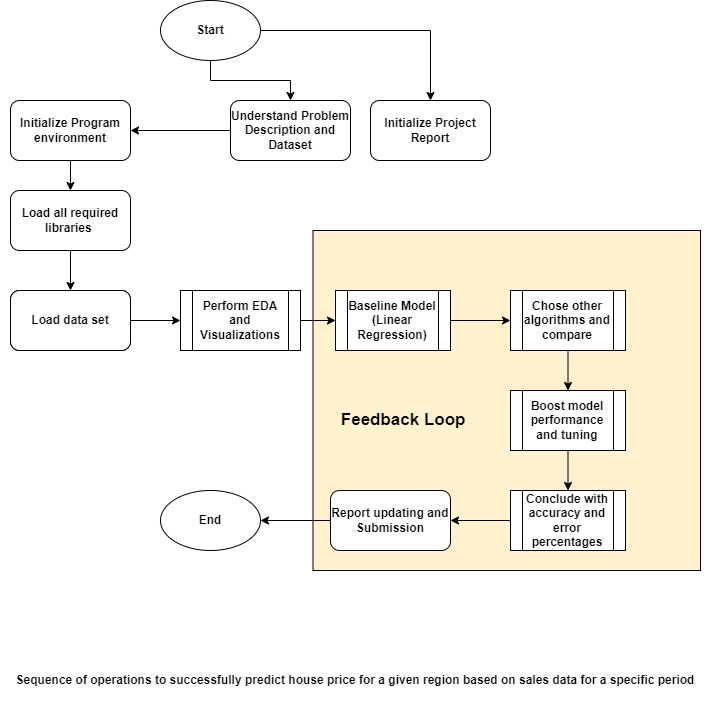
**

Figure 2 Project Flow Chart

# Chapter 3: Data Insights - Exploratory Data Analysis

## 3.1 Five Point Summary

The five-point summary involves the calculation of 5 summary statistical quantities, namely:

* **Median**: The middle value in the sample, called as the 50th percentile or the 2nd quartile.
* **1st Quartile**: The 25th percentile.
* **3rd Quartile**: The 75th percentile.
* **Minimum**: The smallest observation in the sample.
* **Maximum**: The largest observation in the sample.

Below code and output depicts the 5-point summary statistical quantities of the given House price prediction dataset.

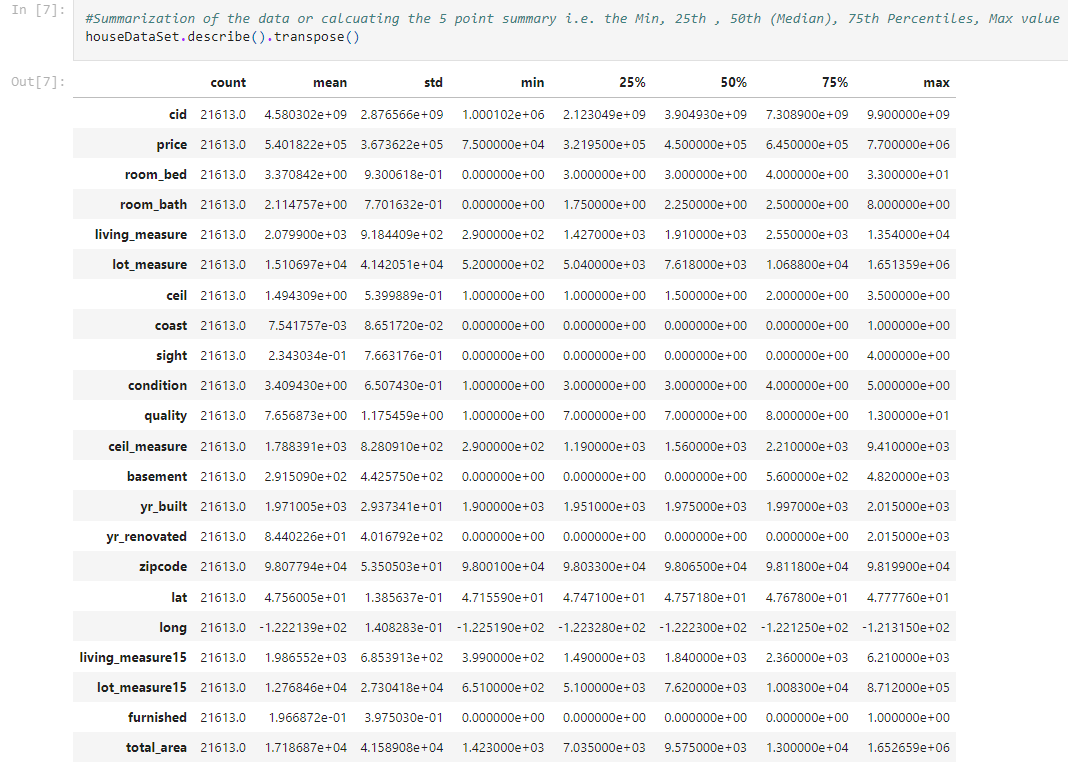


Figure 3 Five-point statistical summary of the given dataset

**Observation from the above figure:**

* Looking through the distribution of features coast, sight, basement, yr\_renovated and furnished it is **left skewed**. It is clearly evident from the above summary that the values for minimum, 25 percentile, 50 percentile and 75 percentile is 0. This is also referred as negative skewness in data.
* Some of the attributes seems like **normally distributed since the mean values are most probably equal to median values**: room\_bed, room\_bath, ceil, condition, yr\_built, zipcode, lat and lang.

### 3.2. EDA and Visualizations

The preliminary analysis of data to discover relationships between measures in the data and to gain an insight on the trends, patterns, and relationships among various entities present in the data set with the help of statistics and visualization tools is called Exploratory Data Analysis (EDA). Exploratory data analysis is cross-classified in two different ways where each method is either graphical or non-graphical. And then, each method is either univariate, bivariate or multivariate.

### 3.2.1: Univariate and Bivariate Analysis (Selective features)

**Univariate Analysis: ‘**Uni’ means one and variate means variable, so in univariate analysis, there is only one dependable variable. The objective of univariate analysis is to derive the data, define and summarize it, and analyze the pattern present in it. In a dataset, it explores each variable separately. Some patterns that can be easily identified with univariate analysis are Central Tendency (mean, mode and median), Dispersion (range, variance), Quartiles (interquartile range), and Standard deviation.

**Distribution of Price**

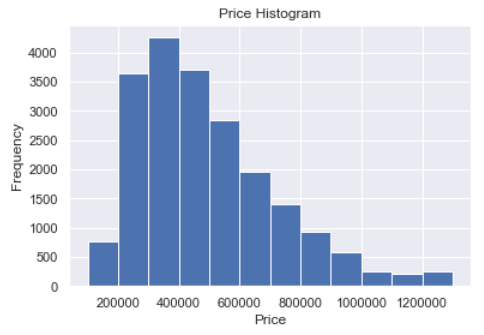


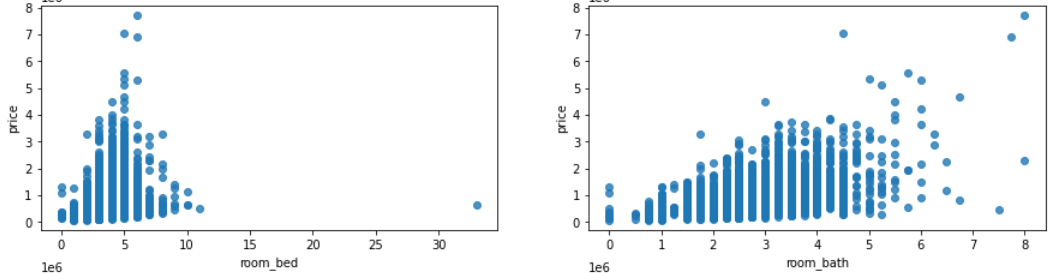
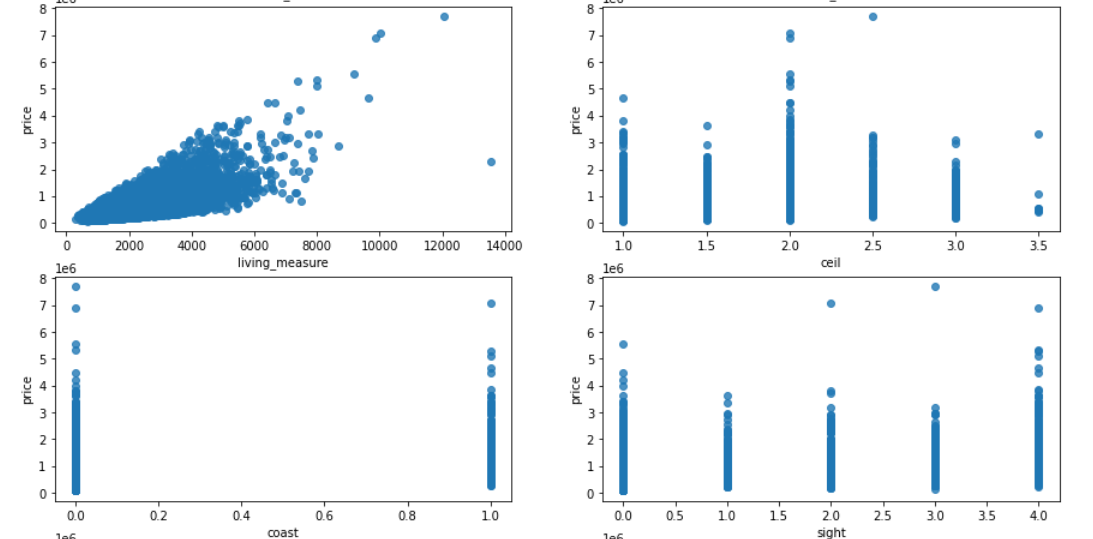
Figure 4 Distribution of Price

**Observation:** Target variable Price distribution is shown in the picture. It is clearly understandable that the target variable is right skewed which is a positive skewness. Most of the house property prices ranges in 40000

**Bivariate Analysis**: Bi means two and variate means variable, so here there are two variables. The analysis is related to cause and the relationship between the two variables. There are three types of bivariate analysis. There are three types of bivariate analysis.

* Bivariate Analysis of two Numerical Variables (Numerical-Numerical)
* Bivariate Analysis of two categorical Variables (Categorical-Categorical)
* Bivariate Analysis of one numerical and one categorical variable (Numerical-Categorical)

**Plot of Key Features vs. Price**



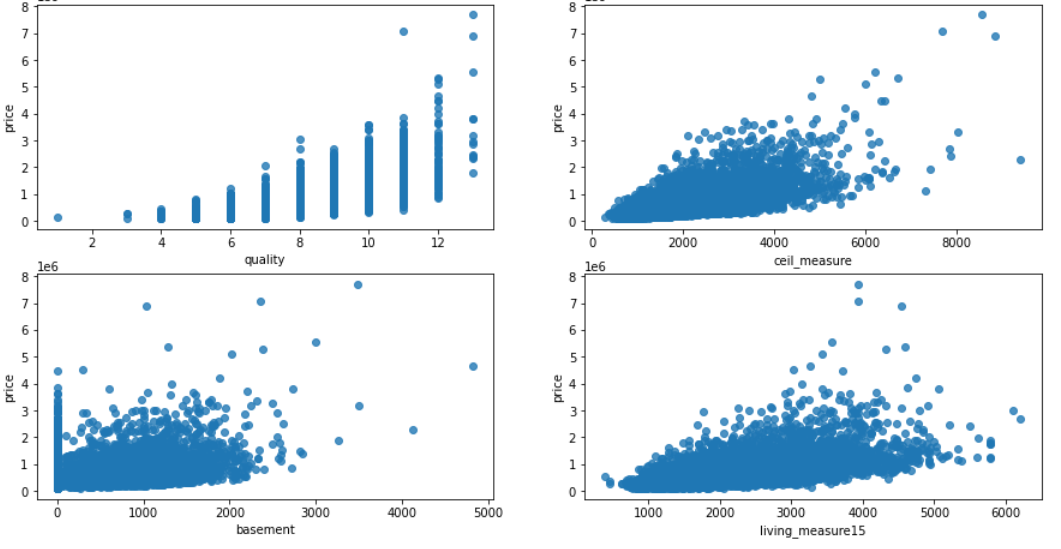


Figure 5 Bivariate Plots regression plots (Price)

Above depicted plots explain:

* Some of the continuous variables like living\_measure, ceil\_measure and living\_measure15 are certainly affecting the target variable whereas basement is not probably affecting the price.
* Referring the categorical variables, quality plays a major role in prediction of prices but condition, sight and coast are going to affect much.

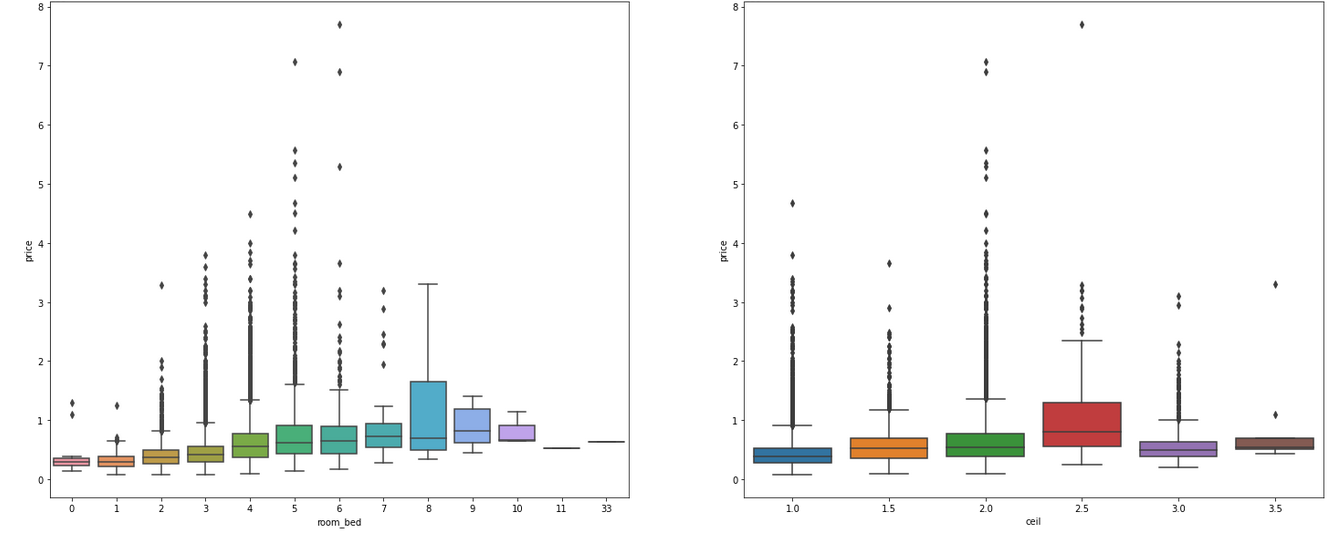
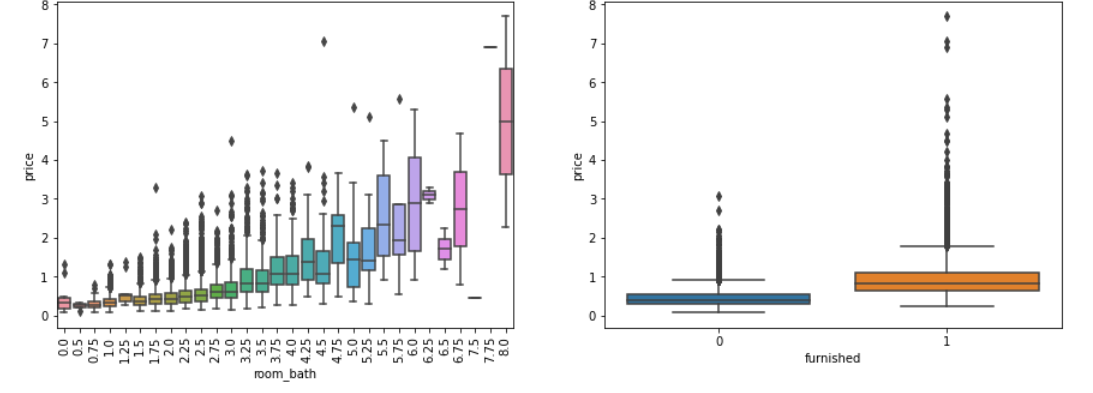


Figure 6 Box Plot (Showing outliers)



From the above charts, it can be seen that there are very few houses which have some features or price appears far from others like 33 bedrooms or price around 7000000. However, determining their possible negative effect will be time consuming and in the real data sets there will always be some outliers like some luxury house prices in this dataset. That's why it is better to not planning for remove outliers

### 3.2.2: Geographical data & Observations

In the Dataset given, we have Geographical information like Latitude, Longitude & Zipcode for each property. We examined further to see which Geographical area this data represents. The Geo information might not be directly relevant to the model building, but it was interesting to find out that the data is from properties in Seattle, WA, USA area. (zip codes 98001 to 98199). We plotted the data using the library, Folium over the Seattle map to see the distribution of the properties. The density of the properties of the given dataset can be clearly seen in Picture 5.

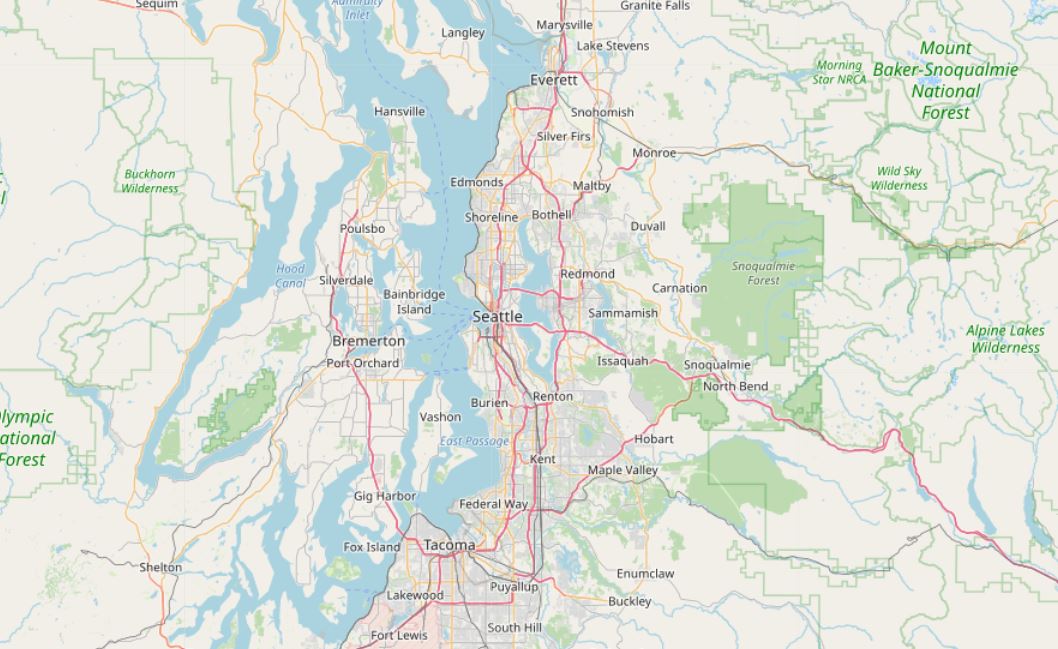


Figure 7 Plain Map of Seattle city.

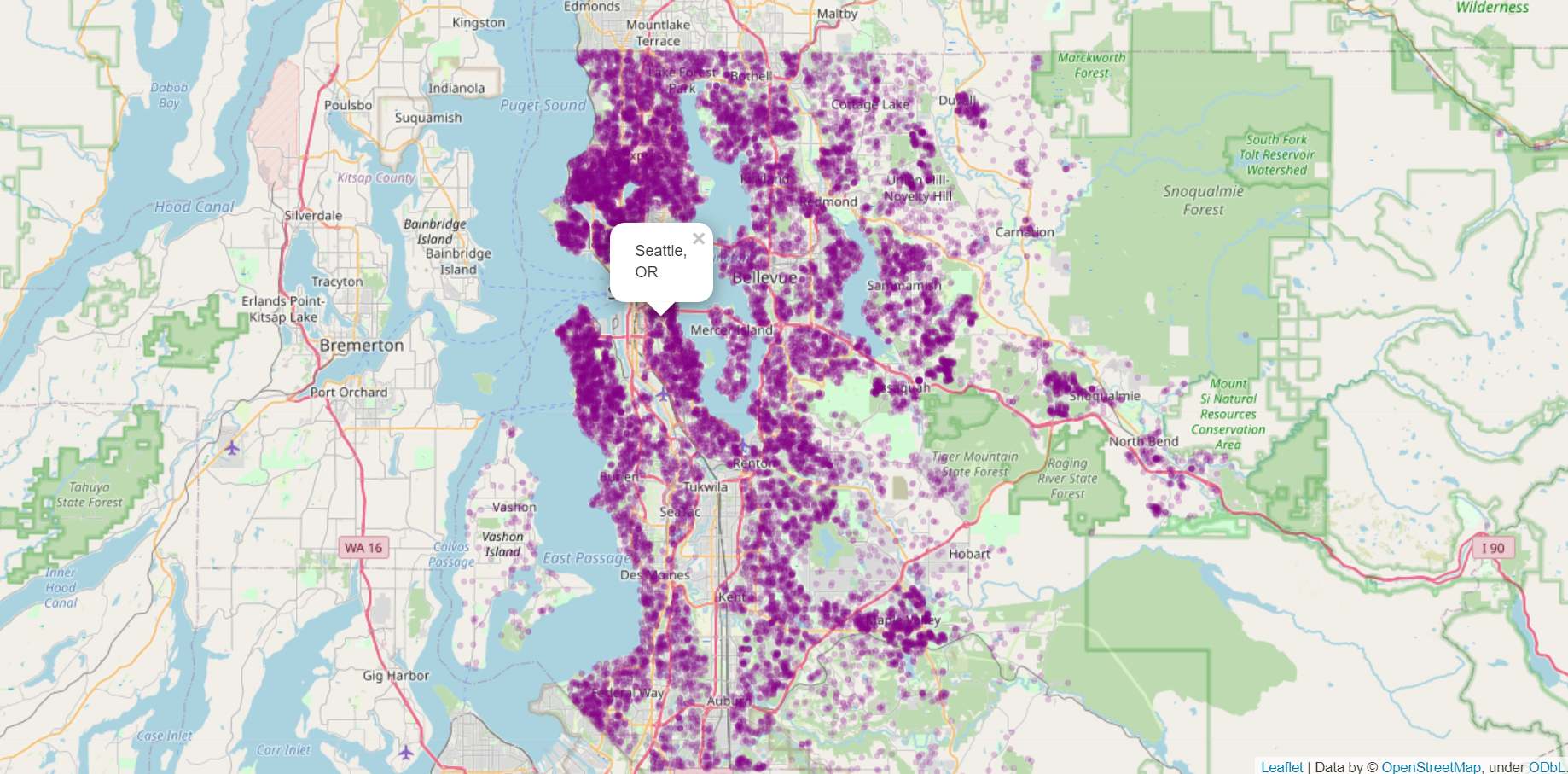


Figure 8 Dataset plotted on a map

3.2.3: Correlation plots (Heat Map)

Correlation is a statistical term describing the degree to which two variables move in coordination with one another. If the two variables move in the same direction, then those variables are said to have a positive correlation. If they move in opposite directions, then they have a negative correlation.

Here we are going with Pearson Correlation. There are other types of correlation also available like

* Positive, Negative or Zero Correlation
* Linear or Curvilinear Correlation
* Scatter Diagram Method
* Pearson's Product Moment Co-efficient of Correlation
* Spearman's Rank Correlation Coefficient

The correlation coefficient is a numerical value ranging from -1 to 1. Below table indicates what correlation coefficient value means to a given data.

*Table 2- illustrates correlation coefficient*

|  |  |
| --- | --- |
| **correlation coefficient** | **Means** |
| 1 | The Features are highly Positively correlated with each other |
| 0 | The Features are not correlated with each other |
| -1 | The Features are highly Negatively correlated with each other |

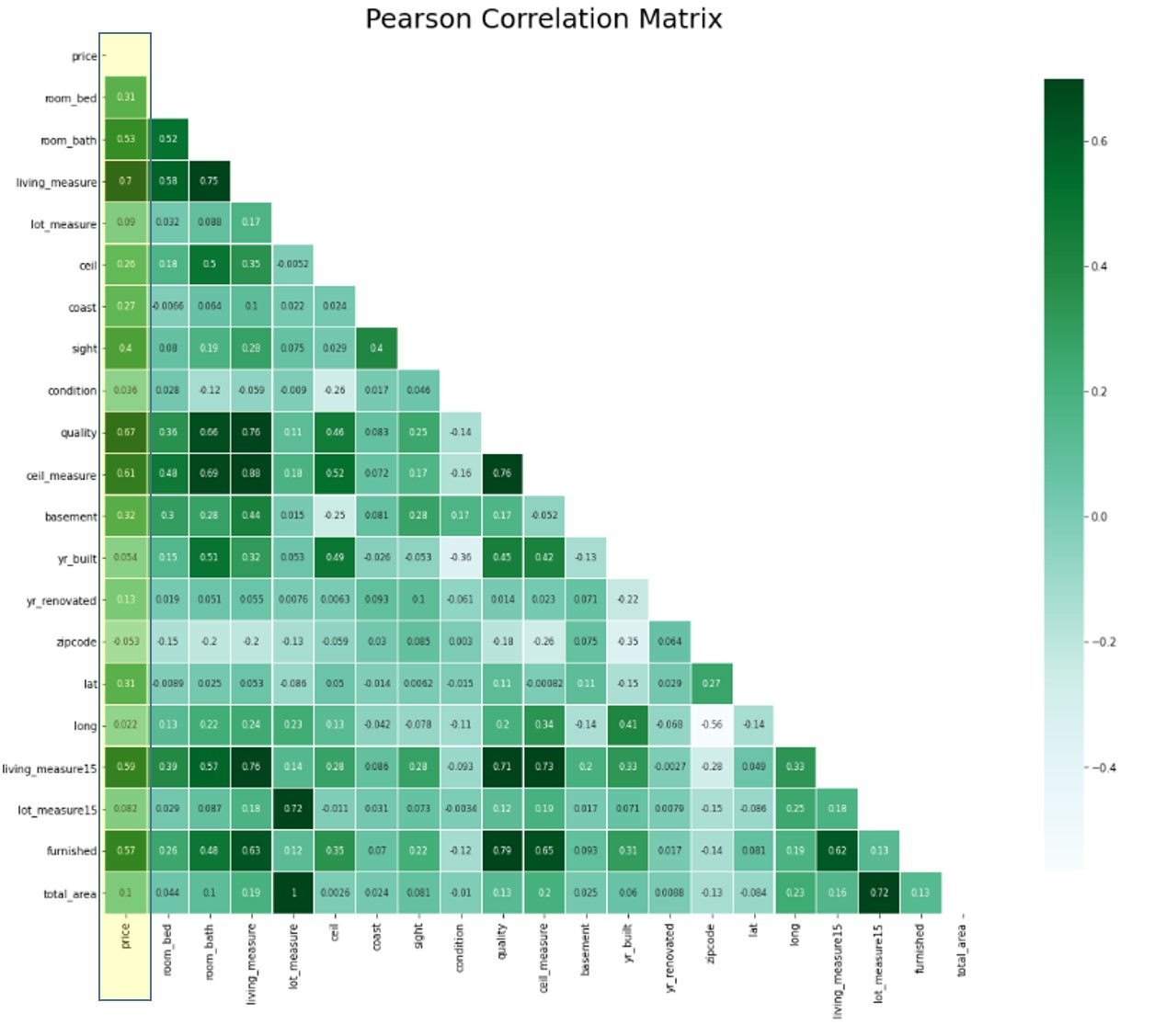


Figure 9 Heat Map (Correlation)

Observation from the Correlation heat map chart (Target Variable Vs. Features):

* **Highly positively correlated Features** – Features like living\_measure (0.7), quality (0.67), ceil\_measure (0.61), living\_measure15 (0.59), furnished (0.57), rooms\_bath (0.53) are more than 0 and closer to 1, hence highly positively correlated with the Target feature Price.
* **Not correlated Features** – Features like lot\_measure (0.09), condition (0.03), zipcode (0.05), longitude (0.02), yr\_built (0.05), lot\_measure15 (0.08), total\_area (0.1) are very close to 0, hence Not correlated with Price
* **Highly Negatively correlated Features** – From the correlation chart, we do not find any Negative correlation for features with price.
* Interesting observation here is on the yr\_built feature. We usually assume that age of a property would be a key factor in determining the price. However, the yr\_built has a correlation closer to 0. During the Development, we will examine further during the model building to see if any new column ("building\_age") can be derived based on yr\_built and yr\_renovation to improve the model efficiency.
* **Multicollinearity** occurs when two or more independent variables are highly correlated with one another in a regression model. This means that an independent variable can be predicted from another independent variable in a regression model. From the above heat map, it is clearly visible that there is a possibility for Multicollinearity hence we checked for VIF (Variable Inflation Factor)

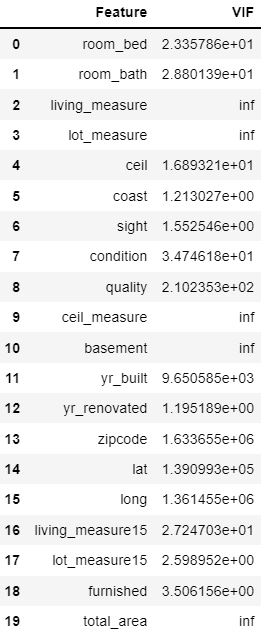


Figure 10 A snapshot of the Variable inflation factor of the Features.

The dataset we have has only 22 Features and 1 Target column. Since the number of Features are only 22, not high we should be good to use all the Features given in the Dataset. Out of the 22 Features, we can clearly see that cid and dayhours are just audit columns and do not add any value to the model in terms of prediction. Based on above correlation heat map/VIF and summary listed, we identify the below features with a correlation less than 0.25 with target variable 'price' as potential columns to be excluded from the data. However, we will be experimenting the different Features to be used/eliminated in the models and finalize the Feature list.

|  |
| --- |
| cid |
| dayhours |
| zipcode |
| lot\_measure |
| lot\_measure15 |
| long |
| condition |
| yr\_built |
| yr\_renovated |
| total\_area |

3.3 Outlier Summary (Identification and handling)

Based on the EDA, we can see some glaring observations.

**Lot\_measure**: In general, 43560 sqft. makes 1 acre. From the below image & the data given, we can notice that there are many multi-acre properties in the dataset. Such big lot house are not very common family houses in a neighborhood within a city. We will be filtering the data to exclude homes that have >10-acre lot space i.e. lot\_measure > 43560 sqft.

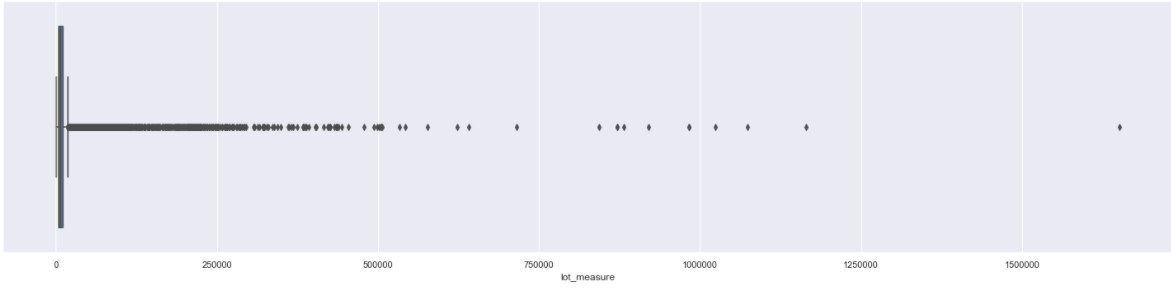


Figure 11 A Box plot illustrating the outlier in lot\_measure feature..

*Before handling lot\_measure anomaly After handling lot\_measure anomaly*

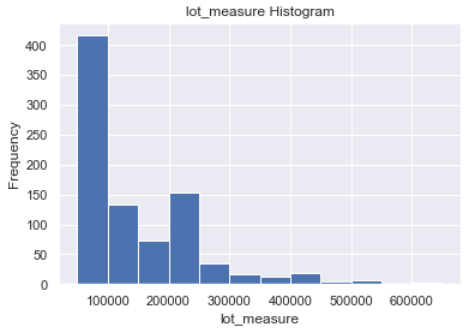
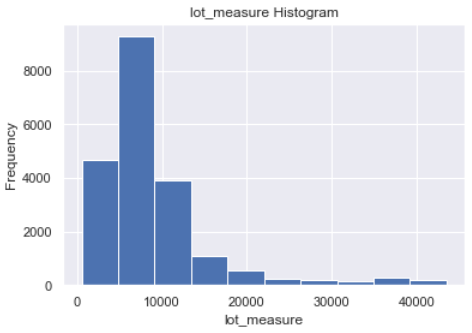
 

Figure 12 Histogram of lot\_measure vs. frequency before & after the anomaly handling

**room\_bed**: In the dataset given, for the record # 750, the no.of Bedroom is given as 33 while the living\_measure and lot\_measure are 1620 & 6000 sqft respectively and the no.of Bath is 1.75. For this lot size, 33 bedrooms seem abnormal and unrealistic. This could very well be a typo error during data collection. We can either impute this record (with average) or filter out this record. Since this is just 1 record with such anomaly, we are filtering out this record to avoid any skewness

# Chapter 4: Model Selection and Building

## 4.1 Selection of Model

The problem at hand is prediction of house property prices. Here the Target variable is given & the Price feature is a continuous variable. Hence this would be a **Supervised - Regression problem**.

For the baseline model to bench mark the performance we used **LazyRegressor** algorithm, an offering from the *Lazy Predict* library.

*Lazy predict* library helps build a lot of basic models without much code and helps understand which models works better without any parameter tuning. Lazy predict library is a quick reference to understand the different algorithm performance to identify a range of the performance score and to bench mark the performance scores. Lazy predict offers algorithm for both Classification and Regression problems.

## 4.2 Model Baseline

## 4.2.1 Data Segregation

The model building process starts with separating features for our model from the target variable. Notice that in our case all columns except *‘Price’* are features that we want to use for the model.

* Our target variable is ‘*Price’* - y\_data
* After dropping cid and dayhours columns the remaining features are candidates for Predictors – x\_data

### 4.2.2 Feature Scaling

Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step. As part of the Baseline model, Models were built with and without scaling for observation purposes to see if any differences were noticed. In this case, the scaling did not alter the performance measure of the models while testing with LazyRegressor.

We used **MinMaxScaler function from sklearn** python library for feature scaling.

### 4.2.3 Model Training and Testing

We are using **train\_test\_split** from the function **scikit-learn** to divide features data (x\_data) and target data (y\_data) even further into train and test. We can control the train\_test split fraction by using the ***test\_size*** parameter. Note that we had it set to 0.3 in our example. It can be any number between 0.0 and 1.0. As part of the model tuning and performance improvement, we will be experimenting with different test size like 0.25, 0.2 to see how the overall model behavior is.

### 4.2.4 Baseline

For the baseline, we experimented with 4 models.

Models 1 & 2 were built and tested with 19 Features (except cid and dayhours). The models were tried with and without scaling and the Max R Squared score, Best model from the LazyRegressor was observed to be **XGBRegressor**.

Models 3 & 4 were built and tested with 11 Features. As per the EDA analysis the below Features were excluded and model was built using LazyRegressor.

lot\_measure (0.09)

condition (0.03)

zipcode (0.05)

longitude (0.02)

yr\_built (0.05)

lot\_measure15 (0.08)

total\_area (0.1)

The co-relation coefficients are given in the bracket for reference.

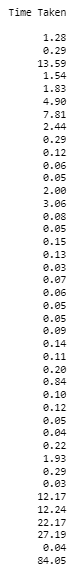
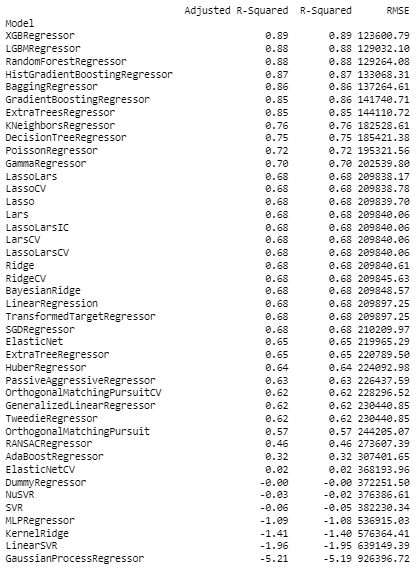
The models were tried with and without scaling and the Max R Squared score, best model from the LazyRegressor was observed to be **XGBRegressor**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Model 1 & 2 (with X=20 Features)\*\*** | | **Model 3 & 4 (with X=12 Features)\*\*** | |
|  | without scaling | with scaling | without scaling | with scaling |
| **Max R-Squared** | 0.89 | 0.89 | 0.83 | 0.83 |
| **Best Model** (based on R-Square, RMSE & Execution time) | XGBRegressor | XGBRegressor | XGBRegressor | XGBRegressor |

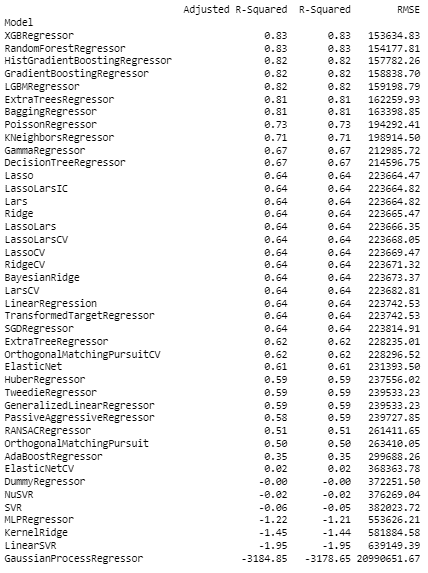
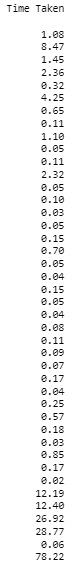
|  |  |  |
| --- | --- | --- |
|  | **Features used\*\*** | |
| **Feature Names** | **Models 1 & 2** | **Models 3 & 4** |
| cid | N | N |
| dayhours | N | N |
| room\_bed | Y | Y |
| room\_bath | Y | Y |
| living\_measure | Y | Y |
| lot\_measure | Y | N |
| ceil | Y | Y |
| coast | Y | Y |
| sight | Y | Y |
| condition | Y | N |
| quality | Y | Y |
| ceil\_measure | Y | Y |
| basement | Y | Y |
| yr\_built | Y | N |
| yr\_renovated | Y | N |
| zipcode | Y | N |
| lat | Y | N |
| long | Y | N |
| living\_measure15 | Y | Y |
| lot\_measure15 | Y | N |
| furnished | Y | Y |
| total\_area | Y | N |

The overall score of these models with and without Scaling is not too different, however the Execution time was less while using scaling.

The LazyRegressor Report for Models 1 & 2 (with scaling) is as below:



The LazyRegressor Report for Models 3 & 4 (with scaling & with reduced Features) is as below:



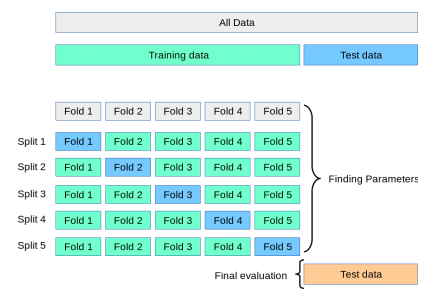
Based on the different experiments done with the LazyRegressor, we can determine that XGBRegressor is the “Best” performing model for this given dataset. **XGBRegressor** is the chosen model.

***We are getting R-Squared score range of 83 to 89% using XGBRegressor model***.

# Chapter 5: Model Tuning and Cross Validation

**Hyperparameter tuning:** All machine learning models have hyperparameters which you must tune or set such that the given model is customized according to the dataset. In definition hyperparameters are model configuration argument specified by the developer to guide the learning process for a specific dataset. [Cross-validation](https://en.wikipedia.org/wiki/Cross-validation_(statistics)) is often used to estimate this generalization performance.

**Cross-Validation** is a resampling technique with the fundamental idea of splitting the dataset into 2 parts- training data and test data. Train data is used to train the model and the unseen test data is used for prediction. If the model performs well over the test data and gives good accuracy, it means the model hasn’t over fitted the training data and can be used for prediction.



In this chapter we would be dealing with hyper parameter tuning in order to increase our regression model accuracy. To select the best set of hyperparameters for a model for a given dataset is often challenging.

There are different types of CV techniques available and here we will look at grid and Randomized search in this section.

## 5.1 Grid Search CV

Defines a search space as a grid of hyperparameter values and evaluate every position in the grid. It helps to loop through predefined hyperparameters and fit the estimator (model) on training set. So, in the end, we can select the best parameters from the listed hyperparameters.

\*\*\*\*Screenshot from master copy\*\*\*

Here we used Grid Search CV on the given dataset which is giving the optimal hyperparameters to achieve the best accuracy score for the model.

## 5.2 Random Search CV

Random search is a technique where random combinations of the hyperparameters are used to find the best solution for the built model. This works best under the assumption that not all hyperparameters are equally important

\*\*\*Attach screenshot from master copy of code\*\*\*

We have explored both Grid search CV & Random search CV in the previous step. However, Grid search CV gives us more flexibility by allowing us to set the grid for the different parameters to be explored. Hence we would be considering **Grid search CV for our model**.

## 5.3 Principal component analysis (PCA)

Principal Component Analysis, or PCA, is a dimensionality-reduction method that is often used **to reduce the dimensionality of large data sets**, by transforming a large set of variables into a smaller one that still contains most of the information in the large set. PCA helps in identifying relationships among different variables & then coupling them.

The dataset given to us has only 21 features (cid, dayhours columns were removed because they were audit columns and not offering much insights for the prediction). The 21 features in comparison is not a high number of dimensions. However, we tried PCA technique on the dataset given to identify feature importance.

\*\*\*Attach screenshot from master copy of code\*\*\*

The observation while testing out the PCA is that, with all 21 features without PCA, our model score was 90% while with PCA (18 Features), the model score was 87%. Here the number of features reduced/eliminated is only 2 or 3. Considering that the number of features reduced and the model score difference is negligible, **we decided to consider the model with all 21 Features without PCA**.

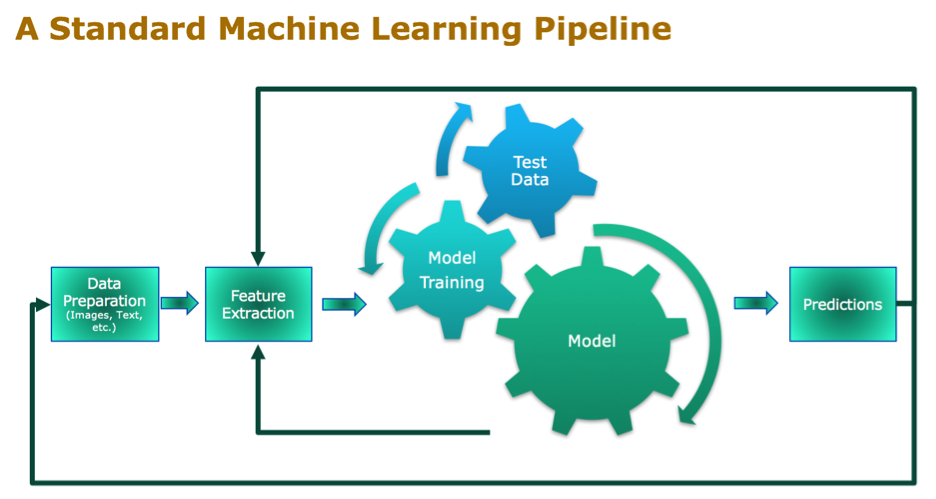
|  |  |  |
| --- | --- | --- |
|  | **With PCA** | **Without PCA** |
| **Number of Features** | 18 | 21 |
| **Accuracy score** | 87% | 90% |

## 

## 5.4 Pipelining

A machine learning pipeline is the end-to-end construct that orchestrates the flow of data into, and output from, a machine learning model (or set of multiple models). It includes raw data input, features, outputs, the machine learning model and model parameters, and prediction outputs.

Machine learning (ML) pipelines consist of several steps to train a model. Machine learning pipelines are iterative as every step is repeated to continuously improve the accuracy of the model and achieve a successful algorithm.



As highlighted in the above section, we tried implementing 2 Pipelines.

* Pipeline 1: Standard scalar --> PCA --> Model (XGB)
* Pipeline 2: Standard scalar --> Model (XGB)

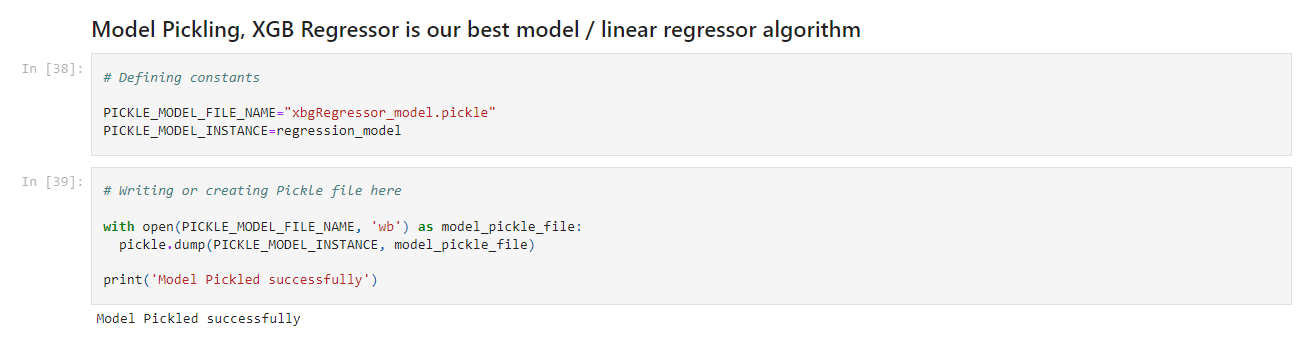
For our final model, we are opting with Pipeline2 for the reasons mentioned above.

\*\*Attach contents from master copy\*\*

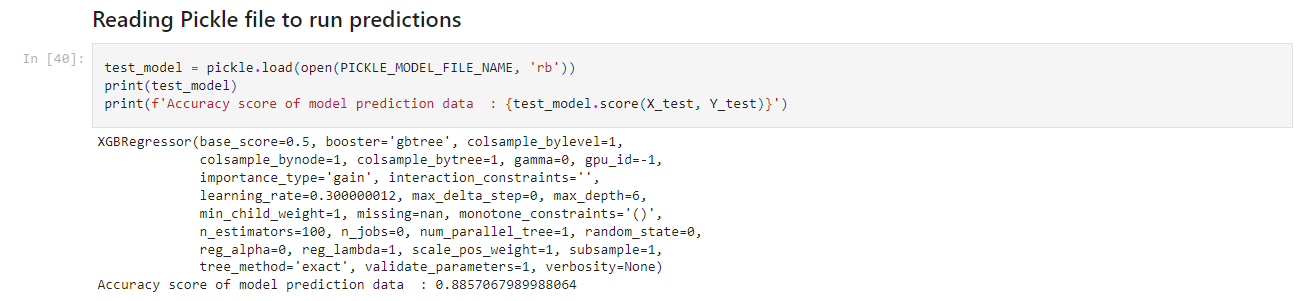
# Chapter 6: Pickling and Model deployment

Pickle is a module in Python used for serializing and de-serializing Python objects. This converts Python objects like lists, dictionaries, etc. into byte streams. You can convert the byte streams back into Python objects through a process called **unpickling**. Pickling is also known as serialization, flattening, or marshalling.

We are pickling the XGB regressor model which is the best model after CV analysis in our code.



Unpickling the pickled model and checking the test scores.



# Chapter 7: Conclusion

\*\*No Snapshots required \*\*

\*\* What was intended, how it was done \*\*

\*\* Deficiencies that were observed during model prediction. \*\*

\*\* Scope of Improvement on the complete approach and Future Enhancements \*\*

\*\* Performance (Model Execution Times) \*\*

# Bibliograph

#### References

1. [Pickling Machine Learning Models. Train your ML models once and use them… | by Aryan Deore | Better Programming](https://betterprogramming.pub/pickling-machine-learning-models-aeb474bc2d78)
2. [Mastering Exploratory Data Analysis(EDA) For Data Science Enthusiasts (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2021/04/mastering-exploratory-data-analysiseda-for-data-science-enthusiasts/)
3. [Linear Regression Algorithm To Make Predictions Easily (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2021/06/linear-regression-in-machine-learning/)