Final Report Capstone Project House Price Prediction

Data Analytics Using Sql (Jain (Deemed-to-be University))

JAIN DEEMED-TO-BE UNIVERSITY | ONLINE

MBA
Semester – IV
Capstone
Project – Interim Report

Name	Rakesh K
Project	House Price Prediction
Group	25
Date of Submission	15/08/2023

A study on "House Price Prediction "

Capstone Project submitted to Jain Online (Deemed-to-be University)

In partial fulfillment of the requirements for the award of:

Master of Business Administration

Submitted by:

Rakesh K

USN:

211VMBR03495

Under the guidance of:

Dr. C. S. Jyothirmayee

(Faculty-JAIN Online)

Jain Online (Deemed-to-be University)

Bangalore

2022-23

DECLARATION

I, Rakesh K hereby declare that the Capstone Project Report titled “House Price Prediction” has been prepared by me under the guidance of the Dr. C. S. Jyothirmayee . I declare that this Project work is towards the partial fulfillment of the University Regulations for the award of the degree of Master of Computer Application by Jain University, Bengaluru. I have undergone a project for a period of Eight Weeks. I further declare that this Project is based on the original study undertaken by me and has not been submitted for the award of any degree/diploma from any other University / Institution.

Place: Bengaluru

Date: 15/08/2023

_____

Rakesh K

211VMBR03495

CERTIFICATE

This is to certify that the Capstone Project report submitted by Mr. Rakesh. K bearing 211VMBR03495 on the title “House Price Prediction" is a record of project work done by him during the academic year 2022-23 under my guidance and supervision in partial fulfillment of Master of Business Administration.

Place: Bengaluru

Date: 15/08/2023

_____

Dr. C. S. Jyothirmayee

TABLE OF CONTENTS

List of Graphs

List of Tables		
Table No.	**Table Title**	**Page No.**
1	Model Evaluation Comparison between all models	27

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 EXECUTIVE SUMMARY

EDA is an important step in any Data Analysis or Data Science project. EDA involves generating summary statistics for numerical data in the dataset and creating various graphical representations to understand the data better. The goal of EDA is to identify patterns, anomalies, and relationships in the data that can be used to inform subsequent steps in the data science process, such as building models or identifying insights. EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables. It also helps answer the questions about standard deviations, categorical variables, and confidence intervals. Finally, once EDA is complete and insights are drawn, its features can then be used for more sophisticated data analysis or modelling, including machine learning.

Data scientists can use exploratory analysis to ensure the results they produce are valid and applicable to any desired business outcomes and goals. EDA also helps stakeholders by confirming they are asking the right questions. EDA can help answer questions about standard deviations, categorical variables, and confidence intervals. Once EDA is complete and insights are drawn, its features can then be used for more sophisticated data analysis or modelling, including machine learning.

In this article, we will understand EDA with the help of an example dataset. We will use python language for this purpose. In this dataset, we used Pandas, Numpy, matplotlib, seaborn, and open datasets libraries. Then loading the dataset into a data frame and reading the dataset using pandas, view the columns and rows of the data, perform descriptive statistics to know better about the features inside the dataset, write the observations, finding the missing values and duplicate rows. Discovering the anomalies in the given set and remove those

anomalies. Univariate visualization of each field in the raw dataset, with summary statistics. Bi-variate visualizations and summary statistics that allow you to assess the relationship between each variable in the dataset and the target variable you're looking at. Predictive models, such as linear regression, use statistics and data to predict outcomes.

Plotting the graphs with different attributes of the dataset and analyzing the given dataset. Then Use the algorithms of regression to understand which is better fit for the data set in house price prediction using model matrix i.e., Mean Squared error, Mean absolute error , Root Mean squared error, R-Squared. Analyze these model matrix for all algorithms in the form of table then identify the best fit.

Some of the most common data science tools used to create an EDA include python, Jupyter. The common packages used are pandas, numpy, matplotlib, seaborn, etc.

One important benefit of conducting exploratory data analysis is that it can help you organize a dataset before you model it. This can help you start to make assumptions and predictions about your dataset. Another benefit of EDA is that it can help you understand the variables in your dataset. This can help you organize your dataset and begin to pinpoint relationships between variables, which is an integral part of data analysis.

Conducting EDA can also help you identify the relationships between the variables in your dataset. Identifying the relationships between variables is a critical part of drawing conclusions from a dataset.

Another important benefit of EDA is helping you choose the right model for your dataset. You can use all of the information that you gain from conducting an EDA to help you choose a data model. It's important to choose the right data model because it can make it easier for everyone in your organization to understand your data. Some commonly used data models that you can choose from include:

You can also use EDA to help you find patterns in a dataset. Finding patterns in a dataset is important because it can help you make predictions and estimations. This can help your organization plan for the future and anticipate problems and solutions.

1.2 Introduction and Background

If you come across any random home buyer questioning them about their dream house, then there are high chances that their descriptions would not start off describing the various aspects of house like the height of basement ceiling or the nearness to a commercial building. Thousands of people seek to place their home on market with the motto of coming up with a reasonable price. Generally, assessors apply their experience and common knowledge to gauge a home based on its various characteristics like its location, commodities and its dimensions. But, regression analysis comes up with another approach which provides much better home prices with reliable predictions. Better still, assessor experience can help guide the modeling process to fine tune a final predictive model. So, this model will help for both the home buyers and home sellers. There is ongoing competition hosted by Kaggle.com from where I am gathering the required data set [1]. The dataset of the competition furnishes good amount of info which helps in price negotiations than the other features of home. This dataset also supports advanced machine learning techniques like random forests and gradient boosting.

The real estate sector is an important industry with many stakeholders ranging from regulatory bodies to private companies and investors. Among these stakeholders, there is a high demand for a better understanding of the industry operational mechanism and driving factors. Today there is a large amount of data available on relevant statistics as well as on additional contextual factors, and it is natural to try to make use of these in order to improve our understanding of the industry.

Let’s suppose we want to make a data science project on the house price prediction of a company. But before we make a model on this data we have to analyze all the information which is present across the dataset like as what is the price of the house, what is the price they are getting, what is the area of the house, and the living measures. These all steps of analyzing and modifying the data come under EDA.

Exploratory Data Analysis (EDA) is an approach that is used to analyze the data and discover trends, patterns, or check assumptions in data with the help of statistical summaries and graphical representations.

The main goal of the project is to find out the accurate predictions of the houses/ properties for the next upcoming years. Here are the step by step process involved

1. Requirement Gathering – We have to gather the information extract the main information from it.
2. Normalizing the data
3. Detecting Outliners in the data
4. Analysis and visualisation using the data

Types of EDA

Depending on the number of columns we are analyzing we can divide EDA into two types.

1. **Univariate Analysis** – In univariate analysis, we analyze or deal with only one variable at a time. The analysis of univariate data is thus the simplest form of analysis since the information deals with only one quantity that changes. It does not deal with causes or relationships and the main purpose of the analysis is to describe the data and find patterns that exist within it.
2. **Bi-Variate analysis** – This type of data involves two different variables. The analysis of this type of data deals with causes and relationships and the analysis is done to find out the relationship between the two variables.
3. Multivariate Analysis – When the data involves three or more variables, it is categorized under multivariate.

Depending on the type of analysis we can also subcategorize EDA into two parts.

1. **Non-graphical Analysis** – In non-graphical analysis, we analyze data using statistical tools like mean median or mode or skewness
2. **Graphical Analysis** – In graphical analysis, we use visualizations charts to visualize trends and patterns in the data

Data Encoding

There are some models like Linear Regression which does not work with categorical dataset in that case we should try to encode categorical dataset into the numerical column. we can use different methods for encoding like Label encoding or One-hot encoding. pandas and sklearn

provide different functions for encoding in our case we will use the Label Encoding function from sklearn to encode.

In this article, we will understand EDA with the help of an example dataset. We will use python language for this purpose. In this dataset, we used Pandas, Numpy, matplotlib, seaborn, and open datasets libraries. Then loading the dataset into a data frame and reading the dataset using pandas, view the columns and rows of the data, perform descriptive statistics to know better about the features inside the dataset, write the observations, finding the missing values and duplicate rows. Discovering the anomalies in the given set and remove those anomalies. Univariate visualization of each field in the raw dataset, with summary statistics. Bivariate visualizations and summary statistics that allow you to assess the relationship between each variable in the dataset and the target variable you're looking at. Predictive models, such as linear regression, use statistics and data to predict outcomes.

Plotting the graphs with different attributes of the dataset and analyzing the given dataset. Then Use the algorithms of regression to understand which is better fit for the data set in house price prediction using model matrix i.e., Mean Squared error, Mean absolute error , Root Mean squared error, R-Squared. Analyze these model matrix for all algorithms in the form of table then identify the best fit.

1.3 Problem Statement

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.

1.4 Objective of the study:

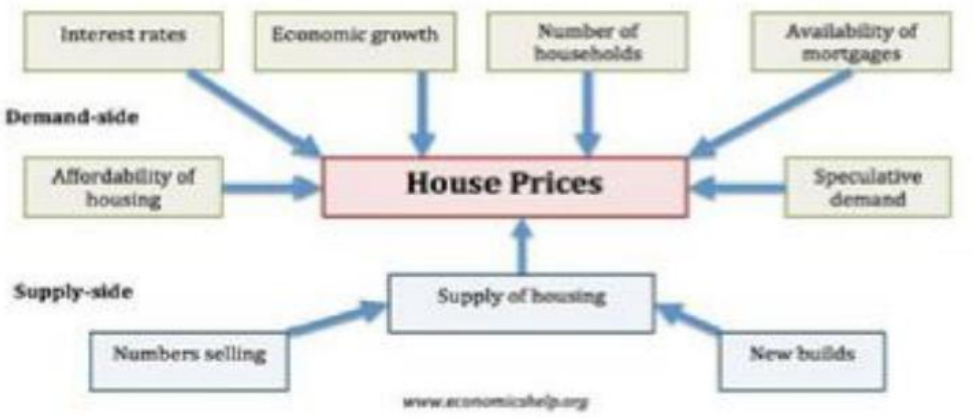
- Create an effective price prediction model
- Validate the model's prediction accuracy
- Identify the important home price attributes which feed the model's predictive power

Take advantage of all of the feature variables available below, use it to analyse and predict house prices.

1. cid: a notation for a house
2. day hours: Date house was sold
3. price: Price is prediction target
4. room_bed: Number of Bedrooms/House
5. room_bath: Number of bathrooms/bedrooms
6. living_measure: square footage of the home
7. lot_measure: quare footage of the lot
8. ceil: Total floors (levels) in house
9. coast: House which has a view to a waterfront
10. sight: Has been viewed
11. condition: How good the condition is (Overall)
12. quality: grade given to the housing unit, based on grading system
13. ceil_measure: square footage of house apart from basement
14. basement_measure: square footage of the basement
15. yr_built: Built Year
16. yr_renovated: Year when house was renovated
17. zip code: zip
18. lat: Latitude coordinate
19. long: Longitude coordinate
20. living_measure15: Living room area in 2015(implies-- some renovations) This might or might not have affected the lot size area
21. lot_measure15: lot Size area in 2015(implies-- some renovations)
22. furnished: Based on the quality of room
23. total_area: Measure of both living and lot

1.5 Literature Survey

The real estate market is one of the most competitive in terms of pricing and same tends to be vary significantly based on lots of factor, forecasting property price is an important modules in decision making for both the buyers and investors in supporting budget allocation, finding property finding stratagems and determining suitable policies hence it becomes one of the prime fields to apply the concepts of machine learning to optimize and predict the prices with high accuracy. The literature review give the clear idea and it will serve as the support for the future projects. most of the authors have concluded that artificial neural network have more influence in predicting the but in real world there are other algorithms which should have taken into the consideration. Investors decisions are based on the market trends to reap maximum returns. Developers are interested to know the future trends for their decision making, this helps to know about the pros and cons and also help to build the project. To accurately estimate property prices and future trends, large amount of data that influences land price is required for analysis, modeling and forecasting. The factors that affect the land price have to be studied and their impact on price has also to be modeled. It is inferred that establishing a simple Regression linear mathematical relationship for these time-series data is found not viable for prediction. Hence it became imperative to establish a non-linear model which can well fit the data characteristic to analyze and predict future trends. As the real estate is fast developing sector, the analysis and prediction of land prices using mathematical modeling and other techniques is an immediate urgent need for decision making by all those concerned.



CHAPTER 2

Research Methodology

2.1 Scope of the Study

This study has been organized through theoretical research and practical implementation of regression algorithms. The theoretical part relies on peer-reviewed articles to answer the research questions, which is going to be detailed. The practical part will be performed according to the design described below and detailed furthermore.

2.2 Methodology

2.2.1 Experimental Methods and Algorithms

2.2.1.1 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list, especially in case of operating systems. The minimal hardware requirements are as follows,

1. PROCESSOR : Intel/AMD, etc.

2. RAM : 8 GB

3. PROCESSOR : 2.4 GHZ

4. MAIN MEMORY : 8GB RAM

5. PROCESSING SPEED : 600 MHZ

6. HARD DISK DRIVE : 1TB

7. KEYBOARD :104 KEYS

2.2.1.2 Software Requirements

Software requirements deals with defining resource requirements and prerequisites that needs to be installed on a computer to provide functioning of an application. These requirements are need to be installed separately before the software is installed. The minimal software requirements are as follows,

1. FRONT END : PYTHON

2. IDE : JUPYTER

3. OPERATING SYSTEM : WINDOWS 10

2.2.1.3 Importing the libraries

In this project, I used python's powerful libraries to make the machine learning models efficient. Majorly three essential libraries NumPy, Pandas, Sci-kit learn had been used in all the machine learning models. NumPy is a powerful library for implementing scientific computing with Python. The most important object of NumPy's is the homogeneous multidimensional array[16]. NumPy saves us from writing inefficient and tiresome huge calculations. NumPy provides a way more elegant solution for mathematical calculations in python. It provides an alternative to the regular python lists. Numpy array is similar to a regular python list with one additional feature. You can perform calculations over all entire arrays easily, super-fast as well. Pandas is a flexible open source python library with high performance, flexible and expressive data structures. Pandas works better with relational and labeled data. Though python is great for data mining and preparation, python lags great in practical, real world data analysis and modeling [17]. Pandas helps great in filling these gaps. It is called the most powerful tool for data analysis and data manipulation. Scikit-learn is a great open source package providing a good chain of supervised and unsupervised algorithms [18]. Scikit-learn is built up on scientific python(SciPy). This library is primarily focused on modeling data. Few popular models of Scikit-learn are clustering, cross validation, ensemble methods, feature extraction and feature selection [18].

2.2.1.4 Getting the dataset :

In this section I will discuss how to load a dataset. In this project, pandas library was used to load all the dataset files. Pandas is powerful and very efficient in analyzing the data and also enables us to read the data of different formats. I choose CSV format because it is very easy to transfer huge databases between the programs. Read_csv pandas function is used in reading the data. This function assumes that the fields are comma separated by default. When a CSV is loaded, we get a kind of object called a Data Frame, which is made up of rows and columns. Part of a data frame is shown in Figure below

The data extracted as:

```
listings=pd.read_excel("innercity.xlsx")
listings.head()
```

	cid	dayhours	price	room_bed	room_bath	living_measure	lot_measure	ceil	coast	sight	condition	quality	ceil_measure	basement
0	3.876101e+09	20150427T000000	600000.0	4.0	1.75	3050.0	9440.0	1.0	0.0	0.0	3.0	8.0	1800.0	1250.0
1	3.145600e+09	20150317T000000	190000.0	2.0	1.00	670.0	3101.0	1.0	0.0	0.0	4.0	6.0	670.0	0.0
2	7.129303e+09	20140820T000000	735000.0	4.0	2.75	3040.0	2415.0	2.0	1.0	4.0	3.0	8.0	3040.0	0.0
3	7.338220e+09	20141010T000000	257000.0	3.0	2.50	1740.0	3721.0	2.0	0.0	0.0	3.0	8.0	1740.0	0.0
4	7.950301e+09	20150218T000000	450000.0	2.0	1.00	1120.0	4590.0	1.0	0.0	0.0	3.0	7.0	1120.0	0.0

2.2.2 Implementation

The mean of the dataset:

```
#we can also find the mean , column wise by axis =0 and we can also find median row wise by axis=1
listings.mean(axis=0)
```

```
C:\Users\Anushri.k\AppData\Local\Temp\ipykernel_6420\919911279.py:2: FutureWarning: Dropping of nuisance columns in DataFrame r
eductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns b
efore calling the reduction.
  listings.mean(axis=0)
```

```
cid                4.580302e+09
price              5.401822e+05
room_bed           3.371355e+00
room_bath          2.115171e+00
living_measure     2.079861e+03
lot_measure        1.510458e+04
sight              2.343663e-01
quality            7.656857e+00
ceil_measure       1.788367e+03
basement           2.915225e+02
yr_renovated       8.440226e+01
zipcode            9.807794e+04
lat                4.756005e+01
living_measure15   1.987066e+03
lot_measure15      1.276654e+04
furnished          1.967198e-01
dtype: float64
```

The median of the dataset:

```
#median of the data set,column wise  and we can also find median row wise by axis=1
listings.median(axis=0)
```

```
C:\Users\Anushri.k\AppData\Local\Temp\ipykernel_6420\3639605630.py:2: FutureWarning: Dropping of nuisance columns in DataFrame
reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns
before calling the reduction.
  listings.median(axis=0)
```

```
cid                3.904930e+09
price              4.500000e+05
room_bed           3.000000e+00
room_bath          2.250000e+00
living_measure     1.910000e+03
lot_measure        7.618000e+03
sight              0.000000e+00
quality            7.000000e+00
ceil_measure       1.560000e+03
basement           0.000000e+00
yr_renovated       0.000000e+00
zipcode            9.806500e+04
lat                4.757180e+01
living_measure15   1.840000e+03
lot_measure15      7.620000e+03
furnished          0.000000e+00
dtype: float64
```

The standard deviation of the dataset:

```
#we an also find standard deviation in column wise and we can also find median row wise by axis=1
listings.std(axis=0)
```

```
C:\Users\Anushri.k\AppData\Local\Temp\ipykernel_6420\1046079274.py:2: FutureWarning: Dropping of nuisance columns in DataFrame
reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select only valid columns
before calling the reduction.
  listings.std(axis=0)
```

```
cid                2.876566e+09
price              3.673622e+05
room_bed           9.302886e-01
room_bath          7.702481e-01
living_measure     9.184961e+02
lot_measure        4.142362e+04
sight              7.664376e-01
quality            1.175484e+00
ceil_measure       8.281025e+02
basement           4.425808e+02
yr_renovated       4.016792e+02
zipcode            5.350503e+01
lat                1.385637e-01
living_measure15   6.855196e+02
lot_measure15      2.728699e+04
furnished          3.975279e-01
dtype: float64
```

2.2.3 Handling Missing data :

The important part and problem of data preprocessing is handling missing values in the dataset. Data scientists must manage missing values because it can adversely affect the operation of machine learning models. Data can be imputed in such a procedure, missing values can be filled based on the other observations.

Techniques involved in imputing unknown or missing observations include:

1. Deleting the whole rows or columns with unknown or missing observations.

2. Missing values can be inferred by averaging techniques like mean, median, mode.

3. Imputing missing observations with the most frequent values.

4. Imputing missing observations by exploring correlations.

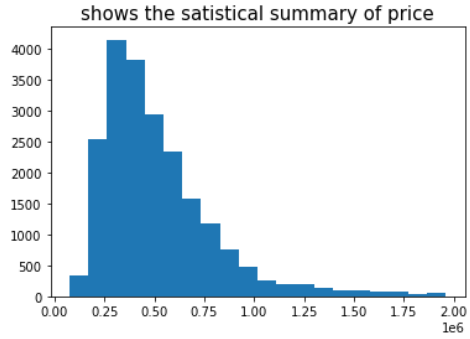
5. Imputing missing observations by exploring similarities between cases.

Missing values are usually represented with ‘nan’, ’NA’ or ‘null’(Refer image 5). Below is the list of variables with missing variables in the train dataset

2.2.4 Uni-Variate, Bi-Variate, Multi-Variate:

Uni-Variate: Uni-Variate in House Price Prediction , chosen attribute like price because by price is independent each other.

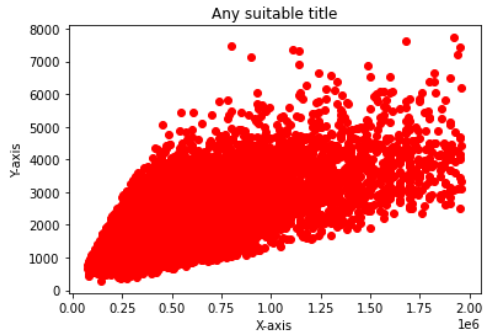
```
plt.hist(listings['price'],bins=20,)
plt.title('shows the satistical summary of price',fontsize=15)
plt.show()
```



Bi-Variate: Bi-variate in House Price Prediction, chosen attributes like price, living_measure because by living_measure price is calculated so these two variables are dependent to each other.

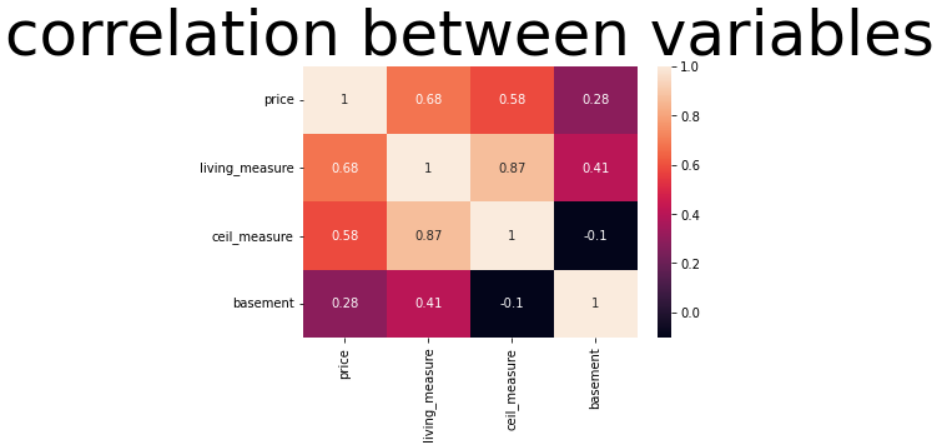
Bi-Variate: The below code explains Bi-variate in House Price Prediction,Choosed attributes like price,living_measure because by living_measure price is calculated so these two variables are dependent to each other.

```
plt.scatter(x=listings['price'],y=listings['living_measure'],color = 'red')
plt.xlabel("X-axis")  # add X-axis label
plt.ylabel("Y-axis")  # add Y-axis label
plt.title("Any suitable title")  # add title
plt.show()
```



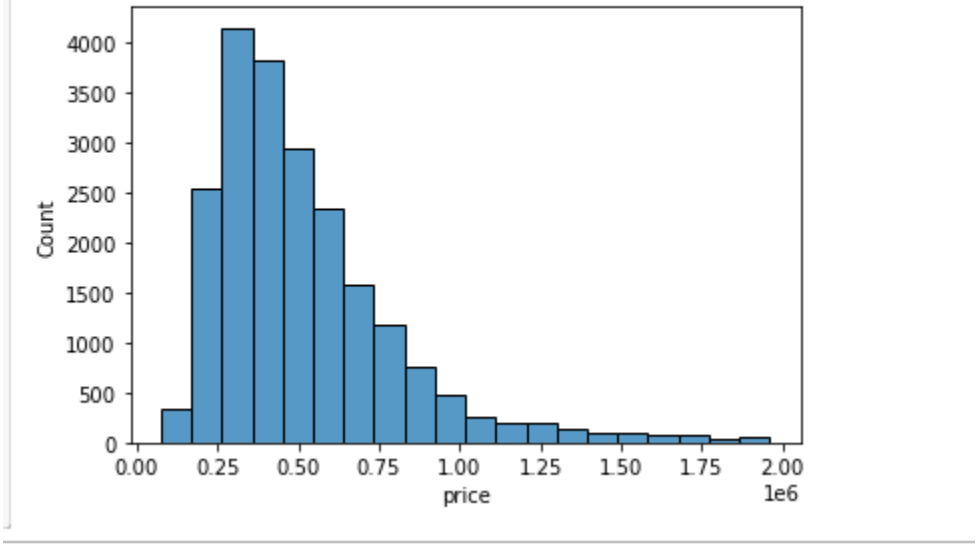
Multi-Variate: Multi-variate in House Price Prediction, chosen attributes like price, living_measure, ceil_measure, basement because ceil_measure ,basement will calculates living_measure and by living_measure price is calculated so these four variables are dependent to each other.

```
Correlation_matrix=listings[['price','living_measure','ceil_measure','basement']].corr()
sns.heatmap(Correlation_matrix,annot = True)
plt.title('correlation between variables',fontsize=50)
plt.show()
```



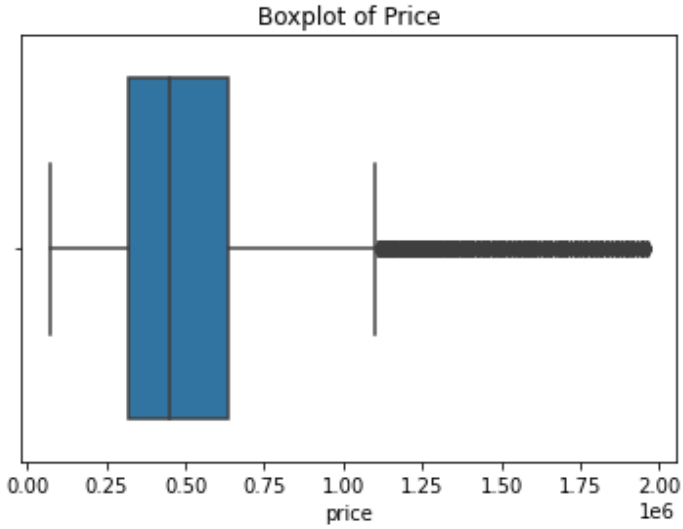
2.2.2.1 Plots: Histogram plot

```
sns.histplot(listings['price'],bins=20);
```



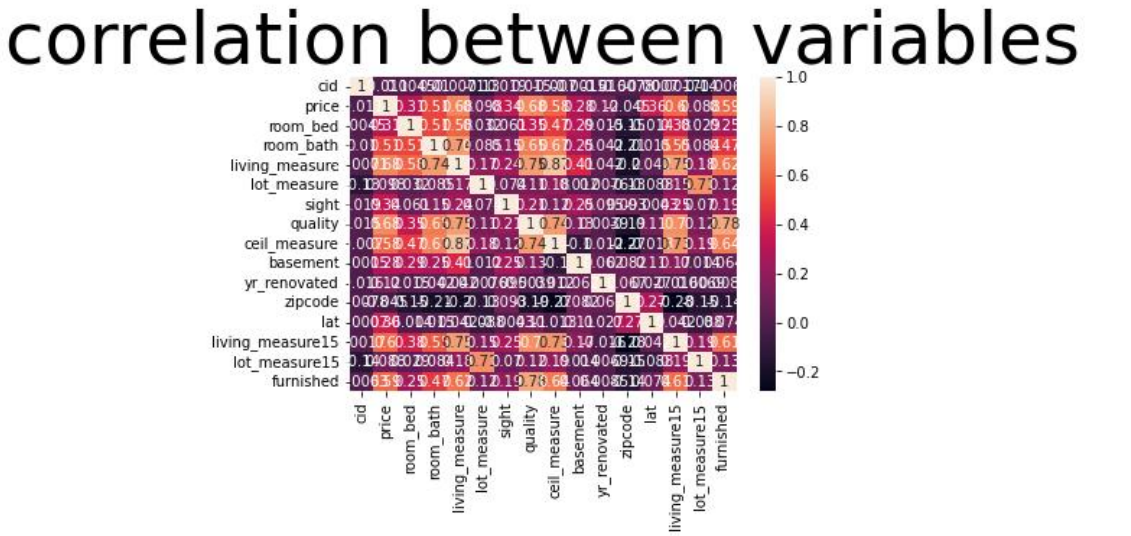
2.2.2.2 Plots: Box plot

```
sns.boxplot(x=listings['price'])
plt.title('Boxplot of Price')
plt.show()
```



2.2.2.3 The correlation between variables:

```
Correlation_matrix=listings.corr()
sns.heatmap(Correlation_matrix,annot = True)
plt.title('correlation between variables',fontsize=50)
plt.show()
```



CHAPTER 3

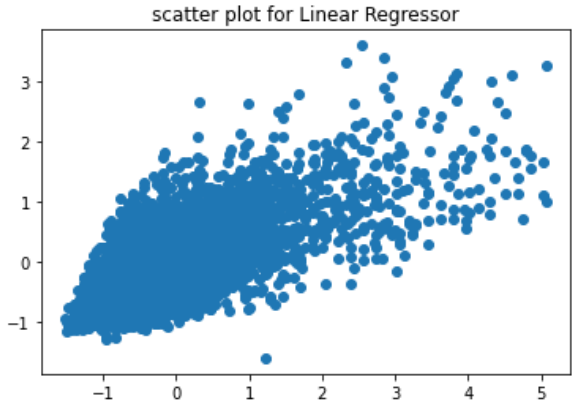
DATA ANALYSIS AND INTERPRETATION

3.1 Linear regression model

Linear regression model shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

Scatter Plot for Linear Regression Model

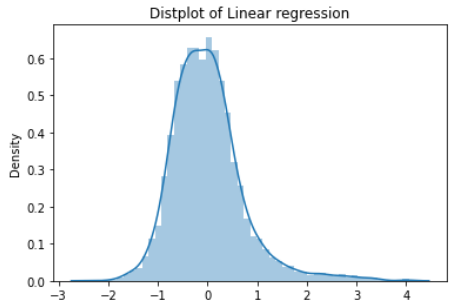
```
from sklearn.linear_model import LinearRegression # to implement Linear Regression
from sklearn import metrics #to check the performance of data i.e MAE,MSE,RSME,R_squared
lm = LinearRegression()
lm.fit(X_train,Y_train)#training the data using linear regression algorithm
y_pred = lm.predict(X_test) #we are prediting y value by using X_test data
plt.scatter(Y_test,y_pred)
plt.title('scatter plot for Linear Regressor')
plt.show()
```



Distplot for Linear Regression Model

```
sns.distplot((Y_test-y_pred),bins=50)
plt.title('Distplot of Linear regression')
plt.show()
```

```
C:\Users\Anushri.k\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



3.1.1 Model evaluation against Linear Regression

Mean Absolute Error (MAE) : It is the simplest error metric used in regression problems. It is basically the sum of average of the absolute difference between the predicted and actual values.

Mean Square Error (MSE) : MSE is like the MAE, but the only difference is that the it squares the difference of actual and predicted output values before summing them all instead of using the absolute value.

Root Mean Squared Error (RSME) : RSME provides information about the short-term performance of a model by allowing a term-by-term comparison of the actual difference between the estimated and the measured value.

R Squared (R2): R Squared metric is generally used for explanatory purpose and provides an indication of the goodness or fit of a set of predicted output values to the actual output values.

```
Model Evaluations
```

```
Perfomance Metrics classification:

Mean Absolute Error (MAE) It is the simplest error metric used in regression problems. It is basically the sum of average of
the absolute difference between the predicted and actual values.

Mean Square Error (MSE) MSE is like the MAE, but the only difference is that the it squares the difference of actual and
predicted output values before summing them all instead of using the absolute value.

R Squared (R2) R Squared metric is generally used for explanatory purpose and provides an indication of the goodness or fit of
a set of predicted output values to the actual output values.
```

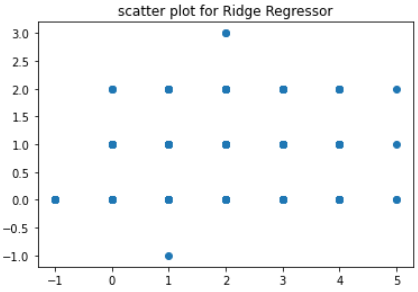
```
import math
print('Mean absloute error(MAE):',metrics.mean_absolute_error(Y_test,y_pred))
print('Mean Squared error(MSE):',metrics.mean_squared_error(Y_test,y_pred))
print('Root Mean Squared error(RMSE):',math.sqrt(metrics.mean_squared_error(Y_test,y_pred)))
print('R Squared:',metrics.r2_score(Y_test,y_pred))
```

```
Mean absloute error(MAE): 0.5303725366294774
Mean Squared error(MSE): 0.5202615514088408
Root Mean Squared error(RMSE): 0.721291585566365
R Squared: 0.4734464966148727
```

3.2 Ridge regression model

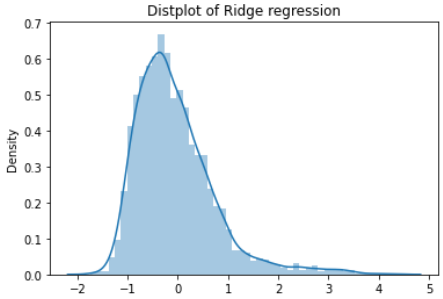
Ridge regression is a technique used to analyze multi-linear regression (multi-collinear), also known as L2 regularization.

```
6]: from sklearn.linear_model import Ridge# to implement Ridge Regression
    from sklearn import metrics #to check the performance of data i.e MAE,MSE,RSME,R_squared
    lm = Ridge()
    lm.fit(X_train,Y_train)#training the data using Ridge regression algorithm
    y_pred = lm.predict(X_test.astype(int)) #we are prediting y value by using X_test data
    plt.scatter(Y_test.astype(int),y_pred.astype(int))
    plt.title('scatter plot for Ridge Regressor')
    plt.show()
```




```
]: sns.distplot((Y_test-y_pred),bins=50)
   plt.title('Distplot of Ridge regression')
   plt.show()
```

```
C:\Users\Anushri.k\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



3.2.1 Model evaluation against Ridge Regression

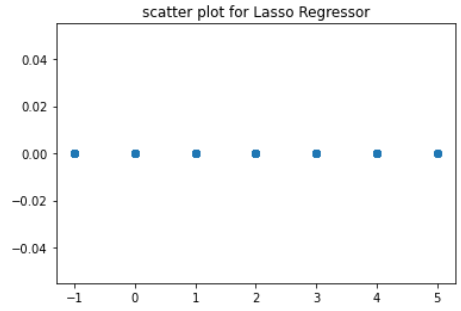
```
import math
print('Mean absloute error(MAE):',metrics.mean_absolute_error(Y_test,y_pred))
print('Mean Squared error(MSE):',metrics.mean_squared_error(Y_test,y_pred))
print('Root Mean Squared error(RMSE):',math.sqrt(metrics.mean_squared_error(Y_test,y_pred)))
print('R Squared:',metrics.r2_score(Y_test,y_pred))
```

```
Mean absloute error(MAE): 0.6000971373576406
Mean Squared error(MSE): 0.6383777773987123
Root Mean Squared error(RMSE): 0.7989854675766714
R Squared: 0.3539017936992397
```

3.3 Lasso Regression

Lasso. It stands for – Least Absolute Shrinkage and Selection Operator is a technique where data points are shrunk towards a central point, like the mean. Lasso is also known as L1 regularization.

```
from sklearn.linear_model import Lasso# to implement Ridge Regression
from sklearn import metrics #to check the performance of data i.e MAE,MSE,RSME,R_squared
lm = Lasso(alpha=1)
lm.fit(X_train,Y_train)#training the data using Ridge regression algorithm
y_pred = lm.predict(X_test.astype(int)) #we are prediting y value by using X_test data
plt.scatter(Y_test.astype(int),y_pred.astype(int))
plt.title('scatter plot for Lasso Regressor')
plt.show()
```



3.3.1 Model evaluation against Lasso Regression

```
import math
print('Mean absloute error(MAE):',metrics.mean_absolute_error(Y_test,y_pred))
print('Mean Squared error(MSE):',metrics.mean_squared_error(Y_test,y_pred))
print('Root Mean Squared error(RMSE):',math.sqrt(metrics.mean_squared_error(Y_test,y_pred)))
print('R Squared:',metrics.r2_score(Y_test,y_pred))
```

```
Mean absloute error(MAE): 0.7345057184025944
Mean Squared error(MSE): 0.9881106470572734
Root Mean Squared error(RMSE): 0.9940375481123805
R Squared: -6.068396026193135e-05
```

3.4 Support Victor Regression (SVR)

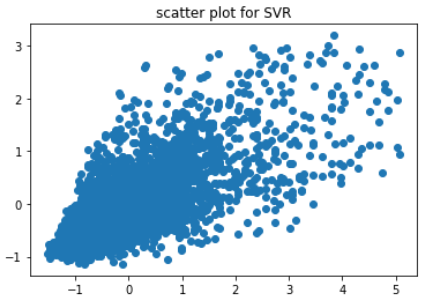
Support Vector Regression (SVR) is a type of machine learning algorithm used for regression analysis. The goal of SVR is to find a function that approximates the relationship between the input variables and a continuous target variable, while minimizing the prediction error.

```
from sklearn.svm import SVR
from sklearn import metrics
sr=SVR(kernel='rbf')
sr.fit(X_train,Y_train)
```

```
C:\Users\Anushri.k\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was pa
ssed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

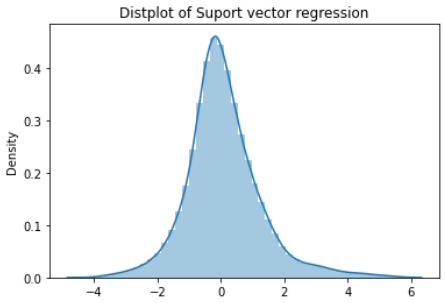
```
SVR()
```

```
y_pred=sr.predict(X_test)
plt.scatter(Y_test,y_pred)
plt.title('scatter plot for SVR')
plt.show()
```




```
sns.distplot((Y_test-y_pred),bins=50)
plt.title('Distplot of Suport vector regression')
plt.show()
```

```
C:\Users\Anushri.k\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



3.4.1 Model evaluation against SVR

```
print('Mean absloute error(MAE):',metrics.mean_absolute_error(Y_test,y_pred))
print('Mean Squared error(MSE):',metrics.mean_squared_error(Y_test,y_pred))
print('Root Mean Squared error(RMSE):',math.sqrt(metrics.mean_squared_error(Y_test,y_pred)))
print('R Squared:',metrics.r2_score(Y_test,y_pred))
```

```
Mean absloute error(MAE): 0.5367886192145526
Mean Squared error(MSE): 0.556198100338712
Root Mean Squared error(RMSE): 0.7457869000852133
R Squared: 0.43707533736362003
```

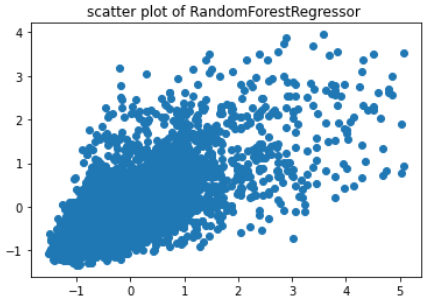
3.5 Random forest regression

Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. Random Forest has multiple decision trees as base learning models. Randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model.

```
from sklearn.ensemble import RandomForestRegressor
from sklearn import metrics
classifier=RandomForestRegressor(n_estimators=150)
classifier.fit(X_train,Y_train)
y_pred=classifier.predict(X_test)
```

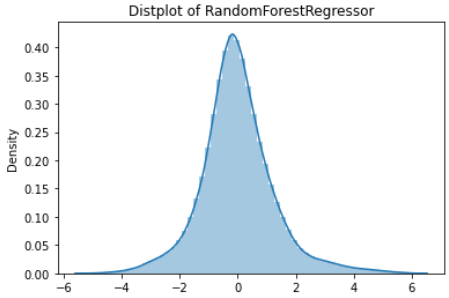
```
C:\Users\Anushri.k\AppData\Local\Temp\ipykernel_6420\2990940304.py:4: DataConversionWarning: A column-vector y was passed when
a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  classifier.fit(X_train,Y_train)
```

```
plt.scatter(Y_test,y_pred)
plt.title('scatter plot of RandomForestRegressor ')
plt.show()
```



```
sns.distplot((Y_test-y_pred),bins=60)
plt.title('Distplot of RandomForestRegressor')
plt.show()
```

```
C:\Users\Anushri.k\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated functio
n and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



3.5.1 Model evaluation against random forest regression

```
: print('Mean absloute error(MAE):',metrics.mean_absolute_error(Y_test,y_pred))
  print('Mean Squared error(MSE):',metrics.mean_squared_error(Y_test,y_pred))
  print('Root Mean Squared error(RMSE):',math.sqrt(metrics.mean_squared_error(Y_test,y_pred)))
  print('R Squared:',metrics.r2_score(Y_test,y_pred))
```

```
Mean absloute error(MAE): 0.5367886192145526
Mean Squared error(MSE): 0.556198100338712
Root Mean Squared error(RMSE): 0.7457869000852133
R Squared: 0.43707533736362003
```

Model Evaluation Comparison between all Models

SL. No	Algorithms	Mean Absolute Error(MAE)	Mean Squared Error(MSE)	Root Mean Squared Error(RMSE)	R Squared
1	Linear Regression	0.53	0.52	0.72	0.47
2	Ridge Regression	0.60	0.63	0.79	0.35
3	Lasso Regression	0.73	0.98	0.99	-6.06
4	Epsilon-Support Vector Regression	0.50	0.50	0.70	0.49
5	Random Forest Regression	0.53	0.55	0.74	0.43

CHAPTER 4

FINDINGS, RECOMMENDATIONS AND CONCLUSION

4.1 Findings Based on Observations

- The experiment is done to pre-process the data and evaluate the prediction accuracy of the models. The experiment has multiple stages that are required to get the prediction results. These stages can be defined as:
- Pre-processing: Datasets will be checked and pre-processed using the methods. These methods have various ways of handling data. Thus, the preprocessing is done on multiple iterations where each time the accuracy will be evaluated with the used combination.
- Data splitting: dividing the dataset into two parts is essential to train the model with one and use the other in the evaluation. The dataset will be split 80% for training and 20% for testing.
- Evaluation: the accuracy of dataset will be evaluated by measuring the R2 and RMSE rate when training the model alongside an evaluation of the actual prices on the test dataset with the prices that are being predicted by the model.
- Performance: alongside the evaluation metrics, the required time to train the model will be measured to show the algorithm vary in terms of time.
- Correlation: correlation between the available features and house price will be evaluated using the Pearson Coefficient Correlation to identify whether the features have a negative, positive or zero correlation with the house price.

4.2 Findings Based on analysis of Data

- Pre-processing methods played a significant role to provide the final prediction accuracy, as shown in the experiment sequence in both public and local data.
- outlier, as suggested by gave a worse outcome than Isolation Forest where it has improved the prediction accuracy.

- The performance of trained models has been measured by evaluating the RMSE, R2 metrics, MAE, MSE .
- The accuracy has been evaluated by plotting the actual prices on the predicted values, as shown below

4.3 Recommendation based on findings

4.3 Experiment Results

- Many machine learning algorithms are used to predict. However, previous researches have shown a comparison between all algorithms.
- Therefore, using these algorithms is beneficial so that the result can be as near to the claimed results.
- However, the prediction accuracy of these algorithms depends heavily on the given data when training the model.
- If the data is in bad shape, the model will be over fitted and inefficient, which means that data pre-processing is an important part of this experiment and will affect the final results.
- Thus, multiple combinations of pre-processing methods need to be tested before getting the data ready to be used in train

4.4 Scope for future research

Future work on this study could be divided into seven main areas to improve the result even further. Which can be done by:

- The used pre-processing methods do help in the prediction accuracy. However, experimenting with different combinations of pre-processing methods to achieve better prediction accuracy.
- Make use of the available features and if they could be combined as binning features has shown that the data got improved.
- Training the datasets with different regression methods such as Elastic net regression that combines both L1 and L2 norms. In order to expand the comparison and check the performance.

- The correlation has shown the association in the local data. Thus, attempting to enhance the local data is required to make rich with features that vary and can provide a strong correlation relationship.
- The factors that have been studied in this study has a weak correlation with the sale price. Hence, by adding more factors to the local dataset that affect the house price, such as GDP, average income, and the population. In order to increase the number of factors that have an impact on house prices. This could also lead to a better finding for question 1 and 2.

The results answer the research questions as follows:

Question 1 – Which machine learning algorithm performs better and has the most accurate result in house price prediction? And why?

Lasso made the best performance overall when both R2 and RMSE scores are taking into consideration. It has achieved the best performance due to its L1 norm regularization for assigning zero weights to the insignificant features.

Question 2 – What are the factors that have affected house prices in Malmö over the years?

The number of crimes, repo, lending, and deposit rates has a weak correlation with the house prices. Which means there are lower likelihood relationships between these factors and sale price. However, when these factors increase the house price decrease. Besides, inflation and year have changed the house prices positively, which means when these factors increase, the house price increase

Conclusion :

Machine Learning technologies brought a scientific revolution in Business Industries. Many of the top notch real estate websites are using machine learning technologies to predict the value of every piece of real estate property accurately to delight their customers. Adopting and integrating machine learning technologies improved customer home buying experience and helped them

prepare and optimize their home for sale.

In this paper, I presented machine learning regression models to predict home prices, which helps people to buy or sell their properties without the help of assessors. By using various regression techniques, I am able to predict the prices of homes using 270 home features. By the use of backward elimination and the Pearson coefficient test, I optimized all the feature selection process

to build accurate models. From my analysis, I have created acceptable Multiple linear regression, random forest regression and polynomial regression. Using K fold cross validation technique, I measured the performance of all models. After comparing all my models with other competitors' in kaggle competition, Random forest regression and Multiple Linear regression performed better whereas polynomial regression gave poor results. Applying regression analysis, backward elimination, Pearson correlation test and k-fold cross validation technique, I obtained the optimal linear regression prediction functions. I would like to work on more machine learning business problems in various industries which helps me to setup a great platform to showcase my skills.

Reference:

- Real Estate Value Prediction Using Linear Regression, Nehal N Ghosalkar ; Sudhir N Dhage.

  https://www.diva-portal.org/smash/get/diva2:1456610/FULLTEXT01.pdf

- Predicting Sales Prices of the Houses Using Regression Methods of Machine Learning, Parasich Andrey Viktorovich ; Parasich Viktor Aleksandrovich ; Kaftannikov Igor Leopoldovich ; Parasich Irina Vasilevna.

  https://sist.sathyabama.ac.in/sist_naac/documents/1.3.4/b.e-cse-batchno-106.pdf

- Uyanık GK GN. A study on multiple linear regression analysis. Procedia-Social and Behavioral Sciences. 2013 Dec ; 106(1): 234-240.

  https://m2pi.ca/project/2020/bc-financial-services-authority/BCFSA-final.pdf

- David HW, William GM. No Free Lunch Theorems for Optimization. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. 1997 April; I(1): 67-82.

  http://103.47.12.35/bitstream/handle/1/9651/BT3083_RPT%20-%20Amit%20Kumar.pdf?sequence=1&isAllowed=y

- Kumar S, Chong I. Correlation analysis to identify the effective data in machine learning: Prediction of depressive disorder and emotion states. International journal of environmental research and public health. 2018 Dec; 15(12): 3907.

  https://www.jetir.org/papers/JETIR2204579.pdf