***HOUSE PRICE PREDICTION USING ML***

**Project Description:**

House is very important part of everyone’s life. Buying house is dream for most of the people. Price of House depends on many parameters like Area, Bedrooms, Parking, Furnishing Status, Main road, etc. We need to predict price of house based on these parameters. We can do that using Machine Learning. We need to train Algorithm based on available Datasets after which we can predict price of house by giving based on parameters values, we give. **Linear Regression** **Algorithm** is used to train data and build a model from it after which we can predict price of house.

Since the machine learning supports only Numerical Data to get trained, it’s necessary to convert categorical data to numerical data before moving to any steps further. We can use **Map method** to convert from categorical data to numerical data which is simple and easy to understand and execute. Initially Dataset we have may have outliers so it’s necessary to handle outliers to get accurate predictions. We can find outliers using Boxplot and handle it using **Capping using IQR technique**. Another important part of this project visualization or EDA (Exploratory Data Analysis) which is used to analyse all columns individually and can get relationships between 2 or more columns to get clear picture of Data graphically. We can analyse one column using **Boxplot, distplot, Countplot, etc.,** and we can analyse two or more columns using **Scatter plot, Barchart, Boxplot, Pairplot, etc**. We can find correlation of 2 column or with all column, we can use **Heatmap** and check how much it’s corelated and check for positive or negative correlation.

**Pre requisites:**

**To complete this project, you must require following software’s, concepts and packages**

Anaconda navigator and PyCharm

**Python packages:**

* Open anaconda prompt as administrator – Go to drive (i.e., J:) – type “Jupyter Notebook”
* Type “pip install NumPy” and click enter.
* Type “pip install Pandas” and click enter.
* Type “pip install scikit-learn” (i.e., sklearn) and click enter.
* Type “pip install matplotlib” and click enter.
* Type “pip install seaborn” and click enter.

**Project Objectives:**

By the end of this project, you will:

* Know fundamental concepts and techniques used for machine learning.
* Gain a broad understanding about data.
* Have knowledge on pre-processing the data/transformation techniques on outlier and
* some visualization concepts.

**Project Flow:**

* User interacts with the UI to enter the input.
* Entered input is analysed by the model which is integrated.
* Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

1. **Data collection**

* Collect the dataset or create the dataset

1. **Visualizing and analysing the data**

* Importing the libraries
* Read the Dataset

1. **Data pre-processing**

* Converting Datatype (categorical data to numerical data)
* Descriptive analysis
* Checking Correlation
* Splitting the dataset as x and y
* Splitting dataset into training and test set

1. **Model building**

* Import the model building libraries
* Initializing the model
* Training and testing the model
* Finding Accuracy and Loss
* Predicting with own Data

***1) Data Collection***

ML depends heavily on data; it is most crucial aspect that makes algorithm training possible. So, this section allows you to download the required dataset or use dataset available in PC.

**Download the dataset**

There are many popular open sources for collecting the data.

Eg: kaggle.com, UCI repository, etc.

In this project I have used Housing.csv data which was available with me.

***2) Visualizing and analysing the data***

As the dataset is downloaded. Let us read and understand the data properly with the help of some visualization techniques and some analysing techniques.

*(Note: There is n number of techniques for understanding the data. But here we have used some of it. In an additional way, you can use multiple techniques.)*

**Step 1: Importing the libraries**

Import the necessary libraries as shown in the image.



**Step 2: Read the Dataset**

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

In pandas we have a function called read\_csv() to read the dataset. As a parameter we have to

give the directory of csv file.



***3) Data Pre-processing***

As we have understood how the data is, let’s pre-process the collected data.

The download data set is not suitable for training the machine learning model as it might have so much of randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps:

* Descriptive analysis
* Converting Datatype (categorical data to numerical data)
* Splitting the dataset as x and y
* Checking Correlation
* Splitting dataset into training and test set

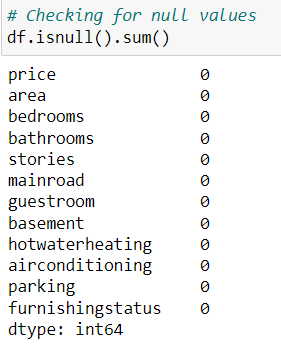
(Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.)

**Step 1: Checking for null values**

For checking the null values, data.isnull() function is used. To sum those null values, we

Use .sum() function to it. From the below image we found that there are no null values

present in our dataset. So, we can skip handling of missing values step.



**Step 2: Checking for Datatypes**

For checking datatypes we can use .info() to get information of all columns belonging to which datatype individually.

We can use “table\_name.select\_dtypes(exclude = 'object')” to get all numerical data (int).

We can use “table\_name.select\_dtypes(include = 'object')” to get all categorical data (object).

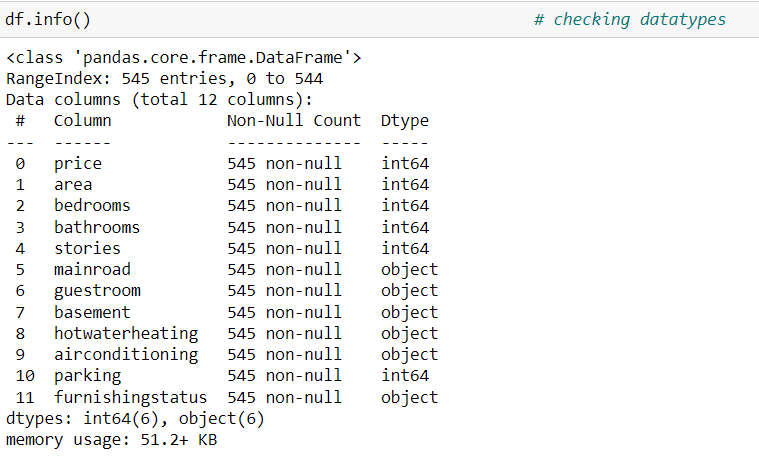


Fig 2.2.1: Information about datatypes

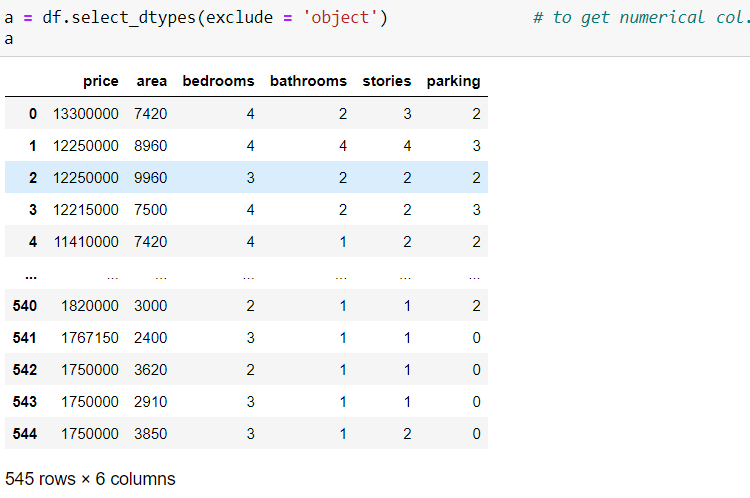


Fig 2.2.1: Numerical Datatypes

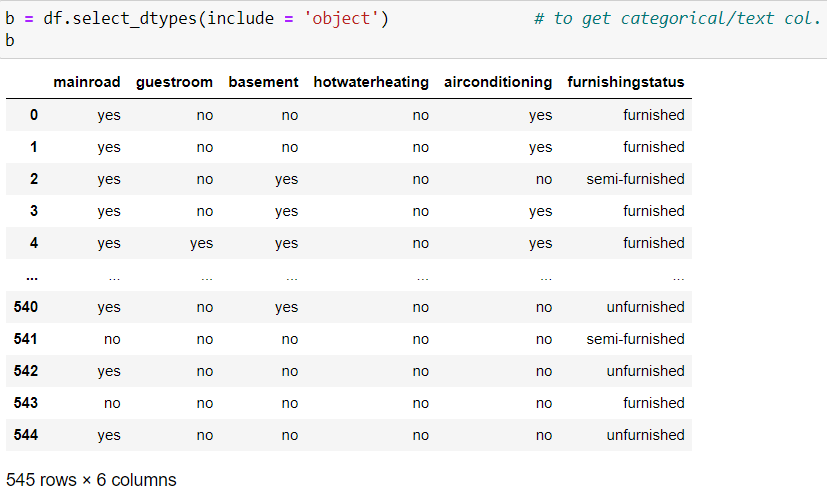
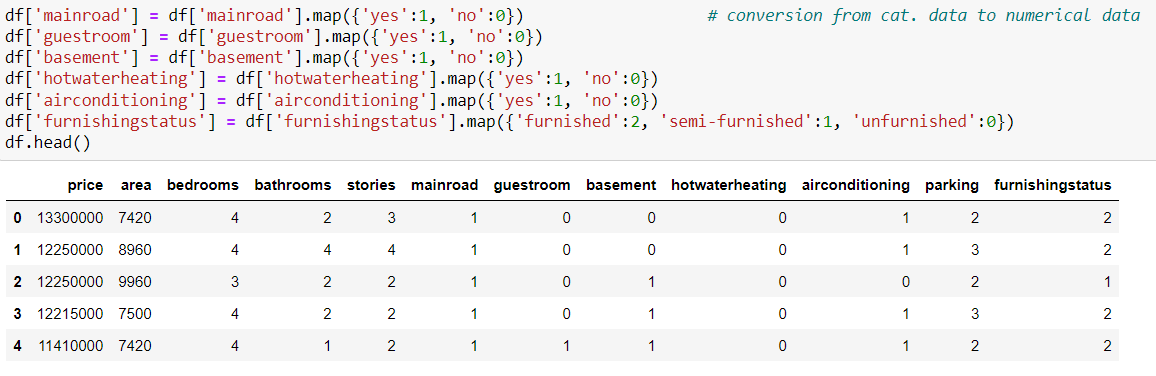


Fig 2.2.1: Categorical/Text Datatypes

**Step 3: Converting the Datatype**

Converting the datatype from object to int. So that we will get output properly.

We need to convert all categorical data to numerical data as we can’t train categorical data in ML. We have many methods to convert but I have chosen map() method as it’s simpler to understand and use.

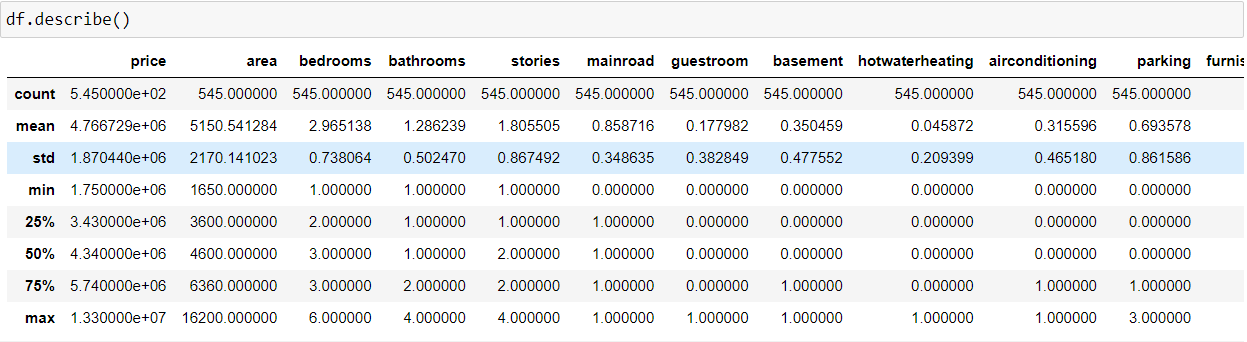


**Step 4: Descriptive analysis**

Descriptive analysis is to study the basic features of data with the statistical process. Here

pandas have a worthy function called describe. With this describe function we can find

mean, std, min, max and percentile values of continuous features.



**Step 5: EDA (Exploratory Data Analysis)**

## **Univariate Analysis**

* Univariate analysis is the simplest form of analysing data. “Uni” means “one”, so in other words our data has only one variable. It doesn’t deal with causes or relationships and its major purpose is to describe. It takes data, summarizes that data and finds patterns in the data.
* There are 2 types of Data, Categorical Data and Numerical Data. For Categorical Data, Countplot and Piechart are best suit. For Numerical Data Boxplot and Distplot are best suit which gives proper distribution of Data

### Categorical Data

### Taking 2 plots for this analysis namely Countplot and Piechart.

### Instead of creating plots for each column, we can use for loop which will create all 2 plots for all categorical columns.

### We can use for loop for variable ‘b’ where all categorical columns are stored, instead of writing it many times for each column we can use for loop which gives the same result, quicker result and less coding. In below fig. I have pasted just one countplot and piechart as an example but it’s there for all categorical column.

### 

Fig: Countplot and Piechart

**Column 1: Countplot**

Col1 shows the Countplot which is used to plot for categorical data in general. We can see differences in categorical data in dataset. We can know how many data has mainroad, how many doesn't like 400+ have mainroad and around 80 doesn't have mainroad.

Similarly, can be checked for other categorical columns as well as shown in above plots.

**Column 2: Piechart**

Col2 shows the piechart which gives precise values in percentage. Like 85.87% data have mainroads and 14.13% data doesn't have mainroad. Similarily, can be checked for all the columns as shown in above plots.

We can observe that form above Countplot and Piechart data, most of the ppl prefer mainroad and doesn't prefer guestroom, basement, hotwaterheating, airconditioning and most of the ppl prefer semi-furnished followed by unfurnished and furnished.

### Numerical Data

* Taking 2 plots for this analysis namely Boxplot and Distplot
* Instead of creating plots for each column, we can use for loop which will create all 2 plots for all numerical columns.

### We can use for loop for variable ‘a’ where all categorical columns are stored, instead of writing it many times for each column we can use for loop which gives the same result, quicker result and less coding. In below fig. I have pasted just one boxplot and distplot as an example but it’s there for all categorical column.

### 

### Fig: Boxplot and distplot

### Column 1: Boxplot

### We can see from above that outliers of all col. are there are hardly 1 outlier except price and area so we can ignore it. Since, there are many outliers present in Price and area we need to handle handle it to increase its accuracy and effeciency.

### Column 2: Distplot

### Other side of col, we can see distplot which is advanced version of Histogram which gives KDE (Kernell Density Estimator) along with graph which estimate the probability density function of a random variable.

## **(b) Bivariate Analysis**

* Bivariate analysis is a statistical method examining how two different things are related.
* The bivariate analysis aims to determine if there is a statistical link between the two variables. and, if so, how strong and in which direction that link is.

1. **Scatterplot (Num-Num)**

### A scatter plot uses dots to represent values for two different numeric variables.

### This plot are majorly suitable for numerical data.

### 

### Fig: Scatterplot

### Barchart

### Purpose of a bar graph is to convey relational information quickly in a visual manner.

### We can see relationship between price and other variables from below fig (in graphs)

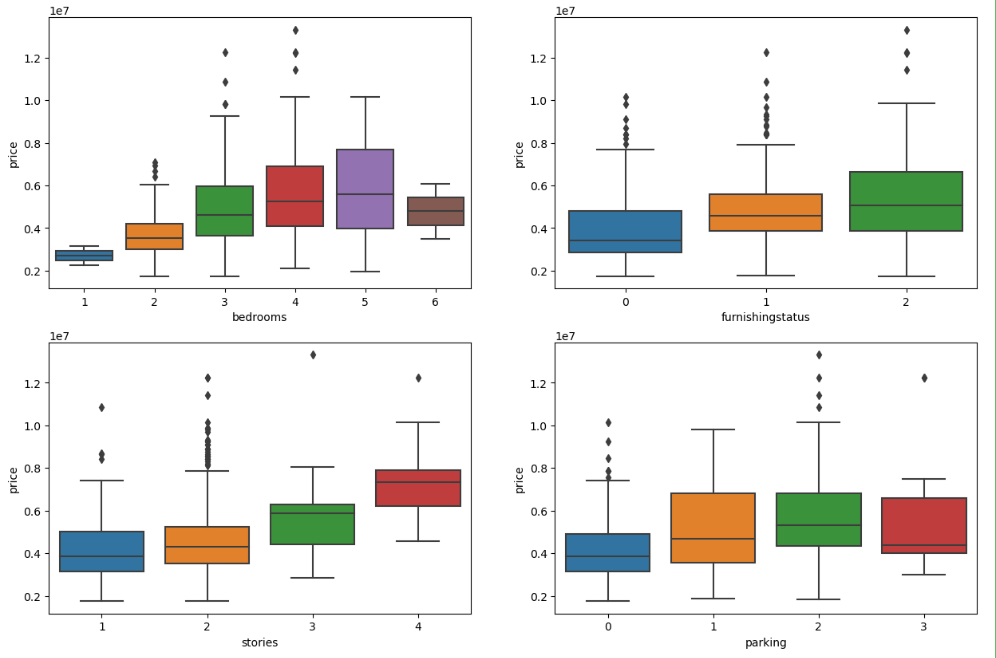
### 

### 

### Fig: Barchart

1. **Boxplot**



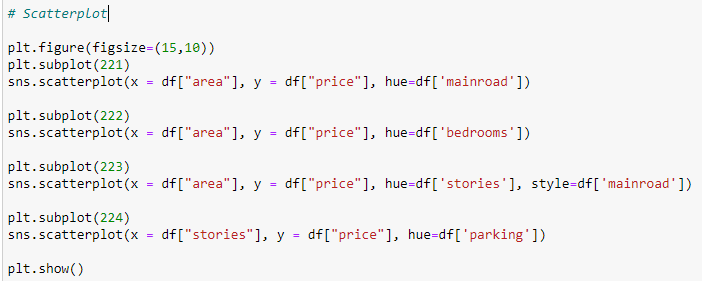


## **Fig: Boxplot**

## **(c) Multivariate Analysis**

* Multivariate analysis involves evaluating multiple variables (more than two) to identify any possible association among them.

1. **Scatterplot**



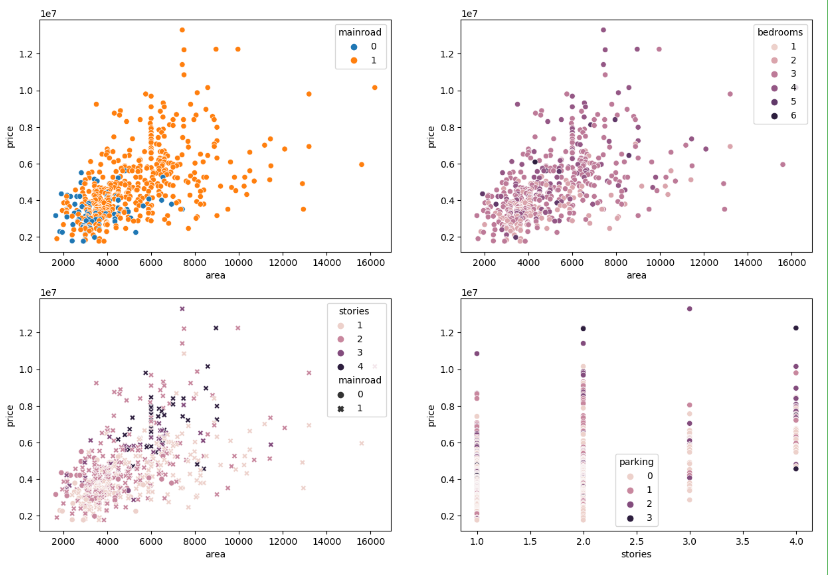
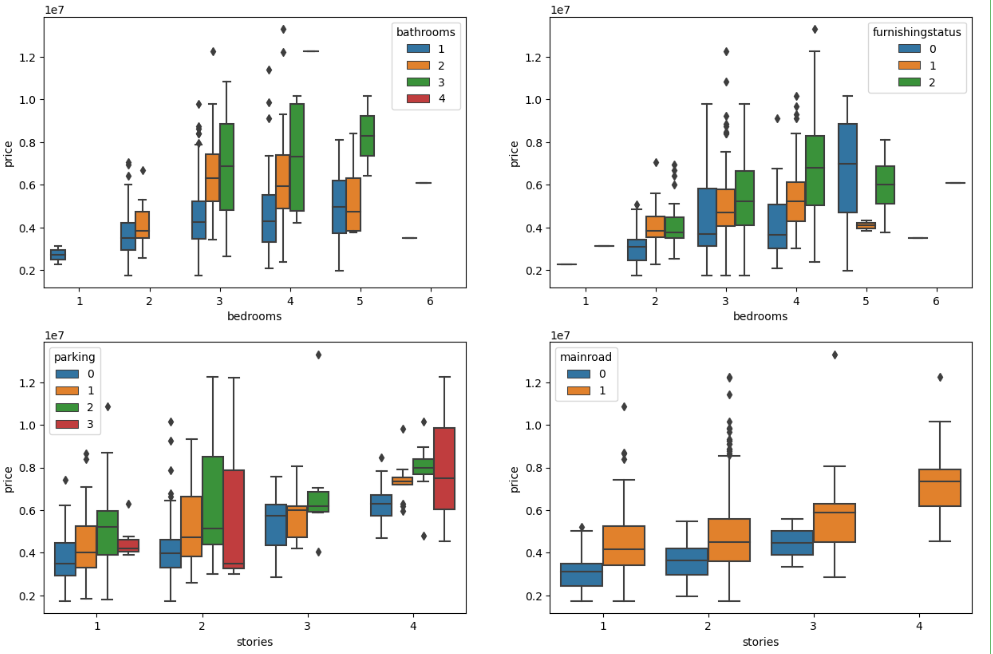


Fig: Scatterplot

* Using scatter plot we can see relationship between more than 3 variables,
* While comparing price and area, we can see how many have mainroad which is in orange color and how many doesn't in blue.
* Similarly, we can see for bedrooms, area and stories which us indicated in different colors.

1. **Boxplot**





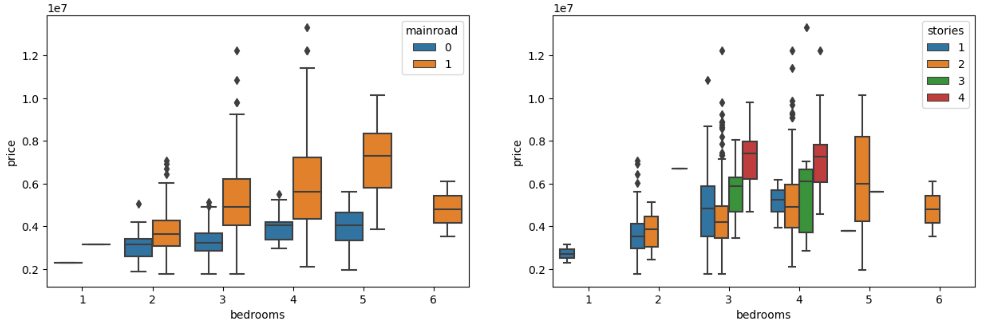
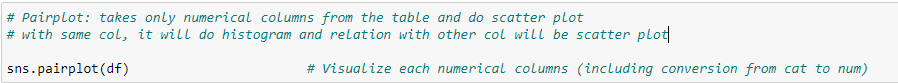


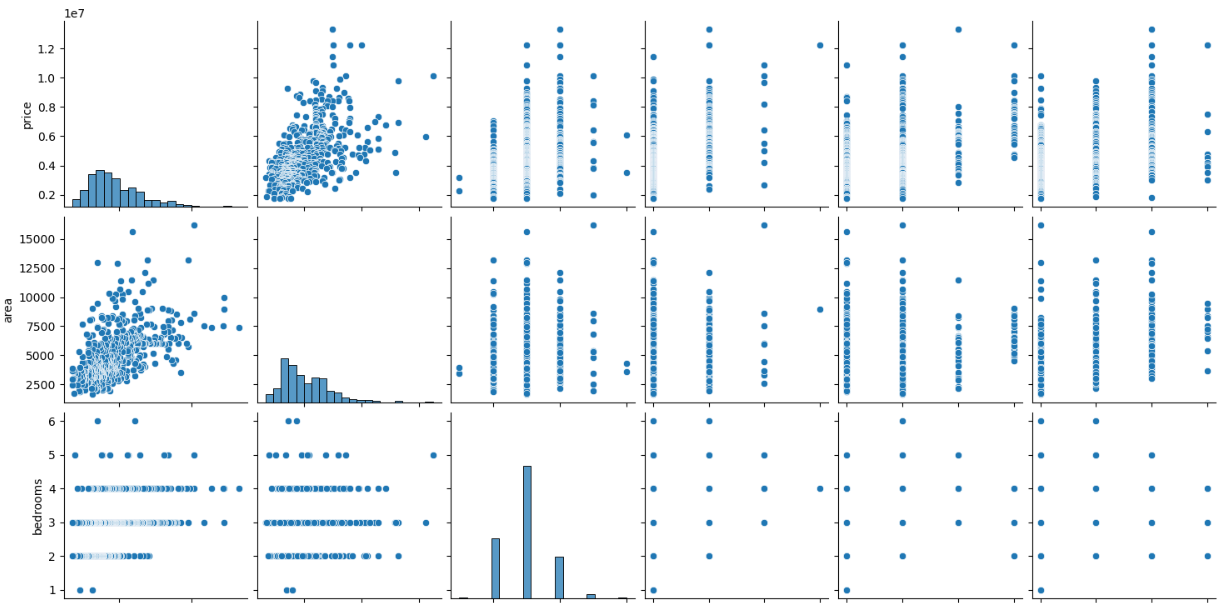
Fig: Boxplot

* We can observe following observations from Boxplot for more than 2 variables,
* For 3 bathrooms 4BHK price is more, for 2 bathrooms 3 & 4BHK prices are more and for 1-bathroom 5BHK price is more.
* If house is not furnished then 5BHK price is high, if semi-furnished then 4BHK and if furnished then 4BHK price is high.
* Price is high for 4stories 3parking followed by 4stories 2parking, 4stories 1parking and 4stories no parking.
* Price depends on stories and mainroad, more the stories high the price and mainroad is included in it.
* Price depends on bedrooms and mainroad, if there is mainroad then price is high depending on stories.
* Price is high if stories and bedrooms are more.

1. **Pairplot**

Pairplot considers only numerical columns from the table and do scatter plot with same col, it will do histogram and relation with another col will be scatter plot.





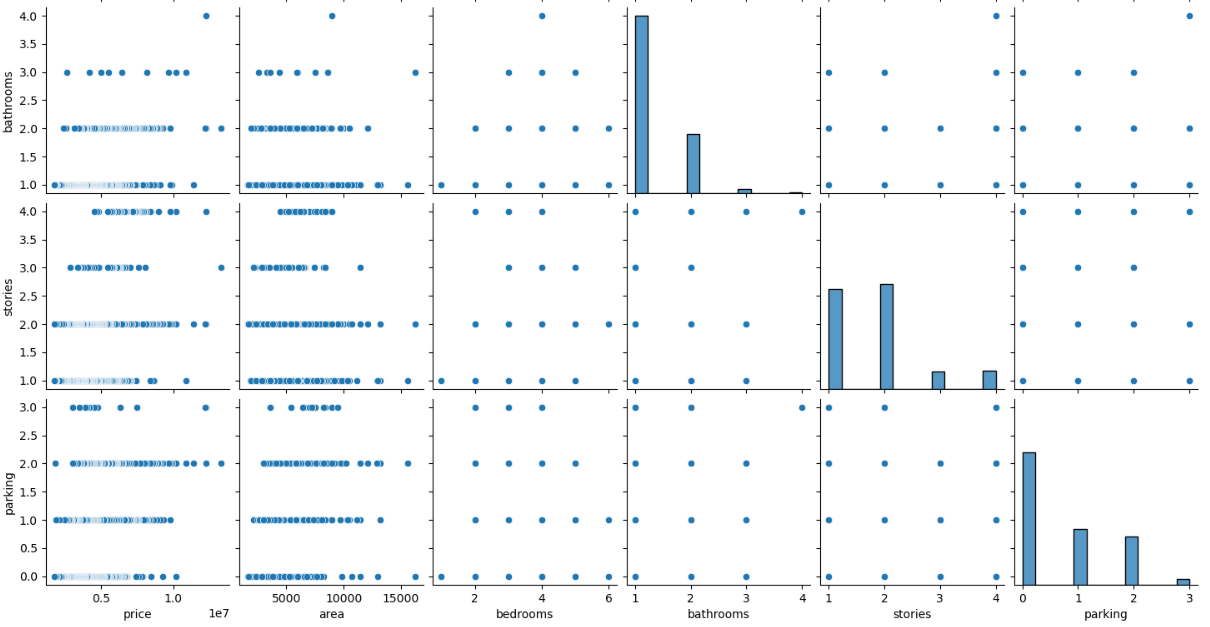


Fig: Pairplot

1. **Checking Correlation (Heatmap)**

Heatmap is used to find correlation between the variables of all columns which makes easier to check corelation between columns like how much strong or week, positive or negative correlated.

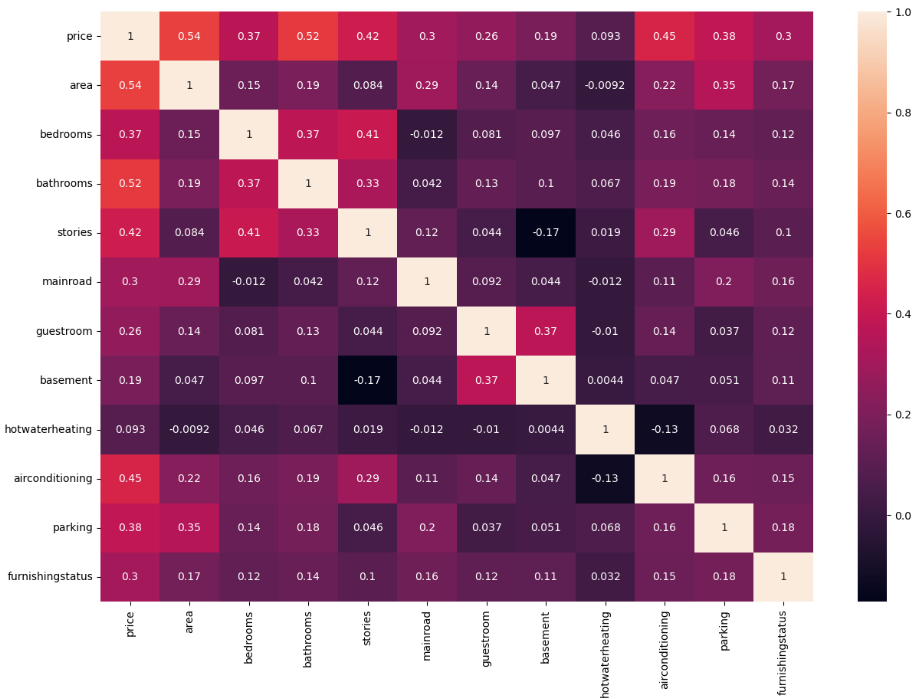
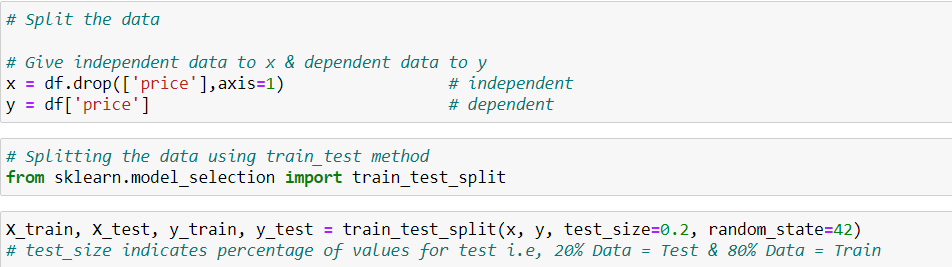


Fig: Heatmap

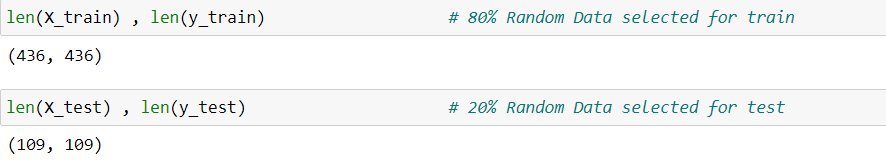
We can see from above heatmap, no feature/col has highly positive or negative correlated so all col. is equally important.

## **Step 6: Splitting the data (Without Handling Outliers)**

* Now let’s split the Dataset into train and test sets.
* Changes: first split the dataset into x and y and then split the data set into train and test.
* Here x and y variables are created. On x variable, data is passed with dropping the target variable. And on y target variable is passed. For splitting training and testing data we are using train\_test\_split() function from sklearn. As parameters, we are passing x, y, test\_size and random\_state. We can save independent columns in x variable and dependent column in y variable.



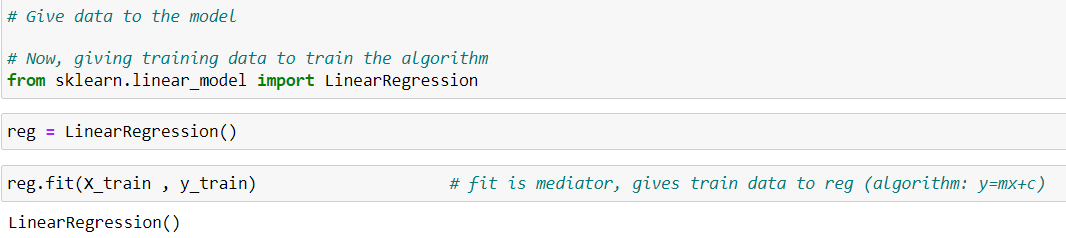
* **We can check no. of rows selected for test and train using len()**



**4) Model Building**

**Step 1: Train the model**

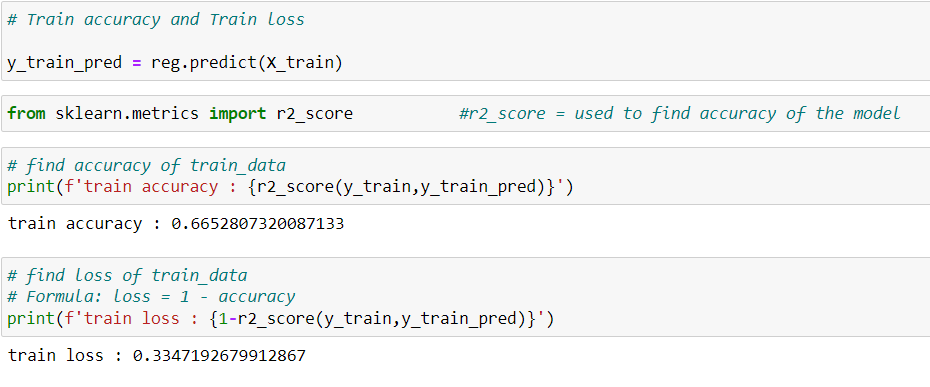
To train the model we need to give test data and train data to the algorithm. Since I am using Linear Regression algorithm, I am importing it which is in sklearn then fitting it to get trained.



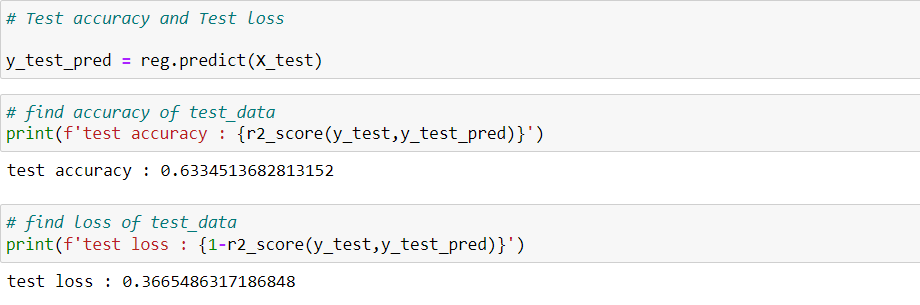
**Step 2: Check Accuracy and Loss of the Model**

Since, model is trained we need to test the model so as to find accuracy and loss of model trained.

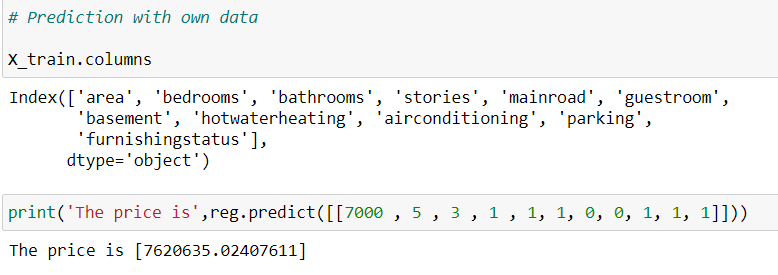
**(i) Train Accuracy and Train Loss**



**(ii) Test Accuracy and Test Loss**



**Step 3: Prediction with own data**



## **Caping using IQR method - Handling Outliers**

* We can see from Boxplot that there are outliers in many columns but only in Price and area col there are more outliers. So, we need to handle outliers so as to increase accuracy of the model. Caping using IQR technique is used to handle the outliers.
* Since, there are many outliers in Price and area column, we can handle only these columns. Others columns have 1 or 2 outliers which can be ignored.

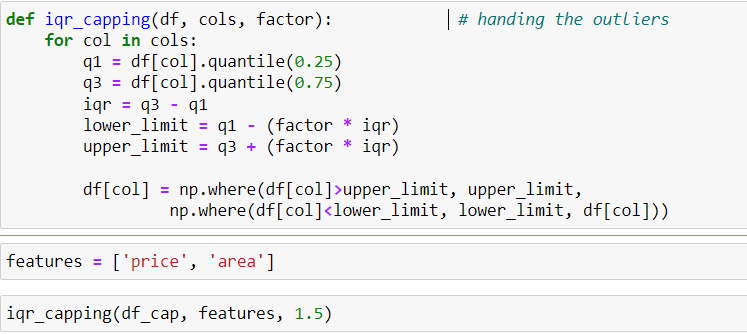
**Step1: Copy Table**

Firstly, we need to copy df (table) instead of disturbing the original table.



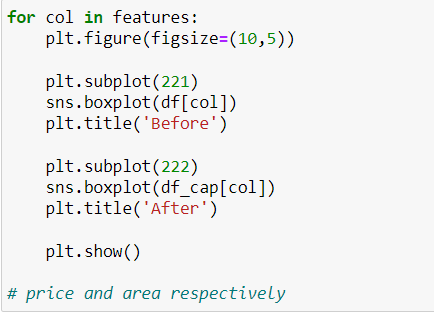
**Step 2: Handling Outliers**

Now, I have used function and for loop so it can be applied for all columns that is selected to handle outliers at once instead of doing it separately. By doing so, it reduces time, increases speed and avoids confusions further.



**Step3: Check for outliers and compare with outliers using Boxplot**

I am plotting Boxplot now to see whether outliers are handled and also compare it without and with handling outliers so as to get clear picture of it.



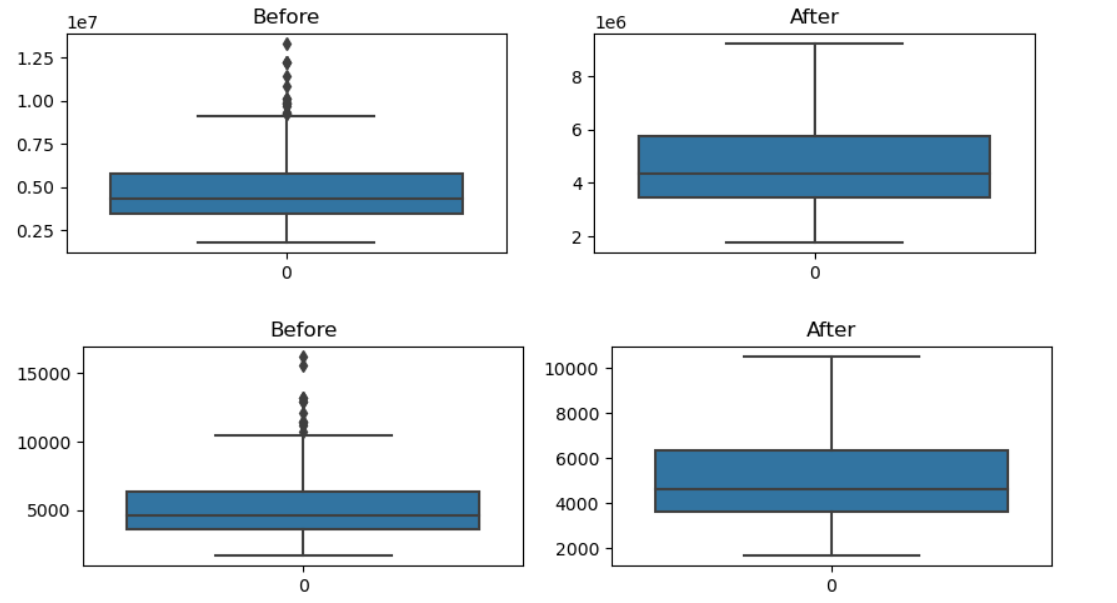
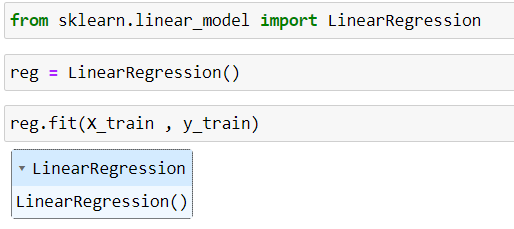


Fig: Boxplot

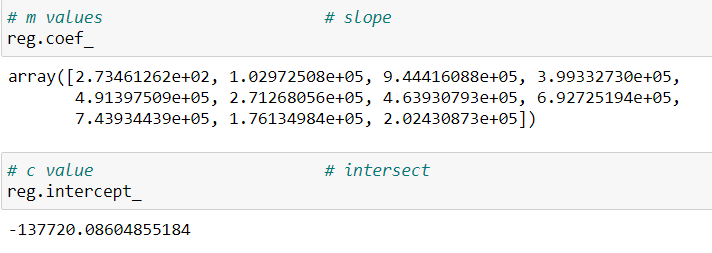
### Step 4: Split the data (After Handling Outliers)

### 

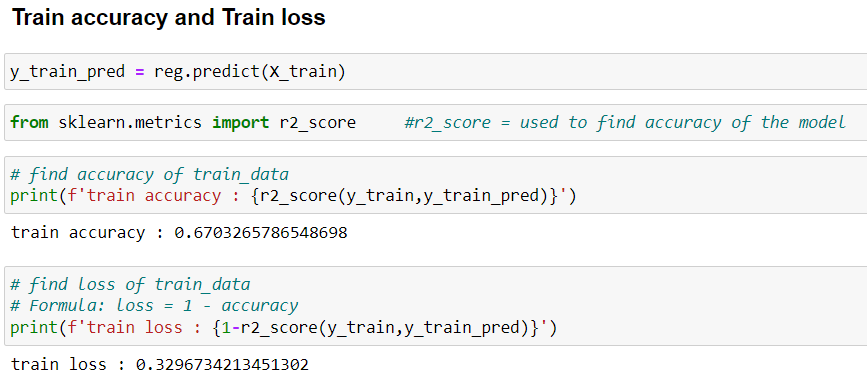
**Step 4: Train the data**

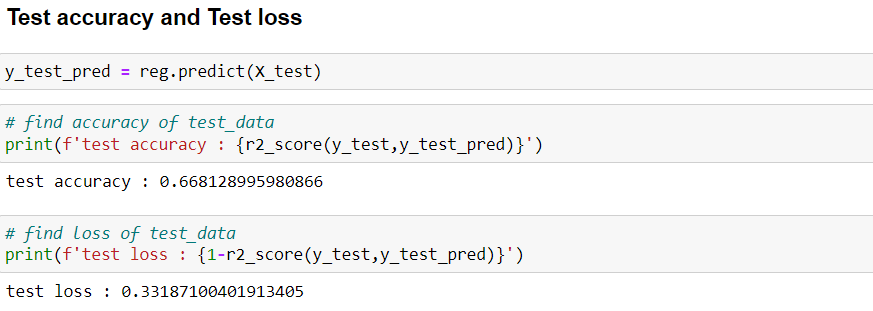


**Step5: Check Slope and Intersect values**



**Step6: Find Accuracy and Loss of the model**





### Step 7: Prediction with own data

### 

### We can observe after and before handling outliers, before handling test accuracy was 63% and after handling it's 66%.

### So, handling outliers has some impact in accuracy which is important for a model to be more precise.

**References**

1. https://www.analyticsvidhya.com/blog/2022/07/step-by-step-exploratory-data-analysis-eda-using-python/.
2. https://medium.com/code-heroku/introduction-to-exploratory-data-analysis-eda-c0257f888676.
3. https://www.youtube.com/watch?v=4HyTlbHUKSw&t=532s.
4. https://www.youtube.com/watch?v=6D3VtEfCw7w&t=1113s.