Chennai House Price Prediction

Guided By:
Shweta Bedarkar

Submitted By:
Nirmal Kumar Ganesan

OUTLINE

- Abstract
- Problem Statement
- Project Specification
- Dataset Description
- Pipeline
- Data Mining
- Data Cleaning
- EXPLORATORY DATA ANALYSIS
- DATA VISUALIZATION
- MODEL EVALUATION
- K-fold cross validation
- Result/Output
- Drawback

ABSTRACT

- We developed a machine learning model to predict house prices in an Indian city for this project (Chennai).
- This project will be extremely beneficial to the real estate market.
- Our model is applicable to both home sellers and home buyers.
- Chennai house sale price data is shared here and the participants are expected to build a sale price prediction model.

PROBLEM STATEMENT

- Real estate transactions can be quite opaque at times, making it challenging for first-time buyers to determine the fair market value of any given home.
- Several real estate websites can predict house prices based on a variety of factors.
- Chennai house sale price data is shared here, and participants are expected to create a sale price prediction model that will help customers find a fair price for their homes while also assisting sellers in understanding what factors are fetching more money for their houses.
- help sellers in understanding what factors are bringing in more money for their homes?

PROJECT SPECIFICATION

- The goal of this project is to predict Chennai house prices based on factors like location, size (area), number of bedrooms, and number of bathrooms.
- The model is created using the Chennai house price dataset.
- To create a predictive model, we are applying a machine-learning algorithm.
- In our project, the multiple regression algorithm is used to train and test the model.

DATASET DESCRIPTION

- The dataset was obtained from kaggle.com. The dataset can be found at https://www.kaggle.com/code/kunwarakash/chennai-house-price-prediction/data.
- There are 7,109 observations in our dataset.
- Our dataset contains a total of 22 columns and attributes.
- The 22 columns are: PRT ID, AREA, INT SQFT, DATE SALE, DIST MAINROAD, N BEDROOM, N BATHROOM, N ROOM, SALE COND, PARK FACIL, DATE BUILD, BUILDTYPE, UTILITY_AVAIL, STREET, MZZONE, QS_ROOMS, QS_BATHROOM, QS_BEDROOM, QS_OVERALL, REG_FEE, COMMIS, SALES_PRICE.
- To train our machine learning model, we used a total of 13 features.

PIPELINE

- DATA MINING
- DATA CLEANING
- EXPLORATORY DATA ANALYSIS
- OUTLIER TREATMENT
- DATA VISUALIZATION
- MODEL EVALUATION
- CROSS VALIDATION

DATA MINING

- The process of extracting information to identify patterns, trends, and useful data that would allow the business to take the data-driven decision from huge sets of data is called Data Mining.
- MS Zoning: Identifies the general zoning classification of the sale.
  - **A**: Agriculture, **C**: Commercial, **FV**: Floating Village Residential, **I**: Industrial, **RH**: Residential High Density, **RL**: Residential Low Density, **RP**: Residential Low Density Park, **RM**: Residential Medium Density
- Street: Type of road access to property
  - Grvl Gravel, Pave Paved, No Access
- Utilities: Type of utilities available
  - **AllPub**: All public Utilities (E,G,W,& S), **NoSewr**: Electricity, Gas, and Water (Septic Tank) **NoSeWa**: Electricity and Gas Only, **ELO**: Electricity only

DATA CLEANING

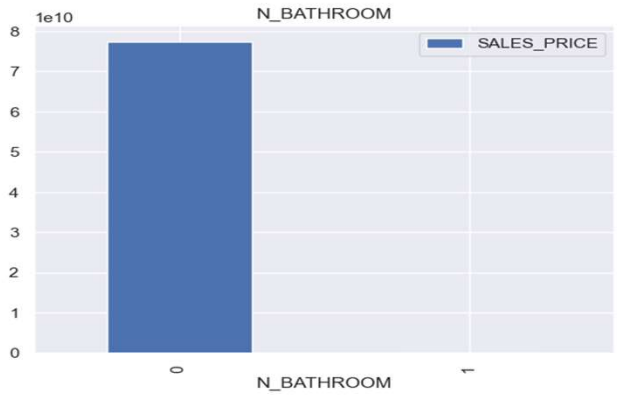
- The main goal of data cleaning is to identify and remove errors and duplicate data in order to create a trustworthy dataset.
- Pandas, a well-known library programme, is used in the data cleaning process.
- Those columns and features are initially removed from our dataset because they are unimportant in determining the final price.
- Rows with null values in any column are removed from our dataset.

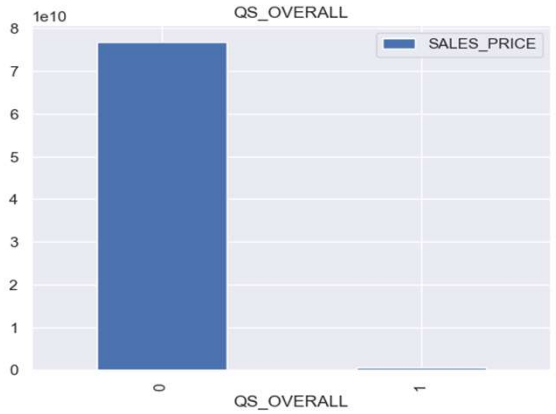
DATA CLEANING

```
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   PRT_ID          7109 non-null   object
 1   AREA            7109 non-null   object
 2   INT_SQFT        7109 non-null   int64
 3   YEAR_SALE       7109 non-null   int64
 4   DATE_SALE       7109 non-null   object
 5   DIST_MAINROAD   7109 non-null   int64
 6   N_BEDROOM       7108 non-null   float64
 7   N_BATHROOM      7104 non-null   float64
 8   N_ROOM          7109 non-null   int64
 9   SALE_COND       7109 non-null   object
 10  PARK_FACIL      7109 non-null   object
 11  YEAR_BUILD      7109 non-null   int64
 12  DATE_BUILD      7109 non-null   object
 13  BUILDTYPE       7109 non-null   object
 14  UTILITY_AVAIL   7109 non-null   object
 15  STREET          7109 non-null   object
 16  MZZONE          7109 non-null   object
 17  QS_ROOMS        7109 non-null   float64
 18  QS_BATHROOM     7109 non-null   float64
 19  QS_BEDROOM      7109 non-null   float64
 20  QS_OVERALL      7061 non-null   float64
 21  REG_FEE         7109 non-null   int64
 22  COMMIS          7109 non-null   int64
 23  SALES_PRICE     7109 non-null   int64
dtypes: float64(6), int64(8), object(10)
memory usage: 1.3+ MB
```

```
1  ##we will check the percentage of nan values present in each feature
2
3  features_with_na=[features for features in data.columns if data[features].isnull().sum()>1]
4
5  for feature in features_with_na:
6      print(feature, data[feature].isnull().sum(),  '  missing values')
```

```
N_BATHROOM 5   missing values
QS_OVERALL 48   missing values
```

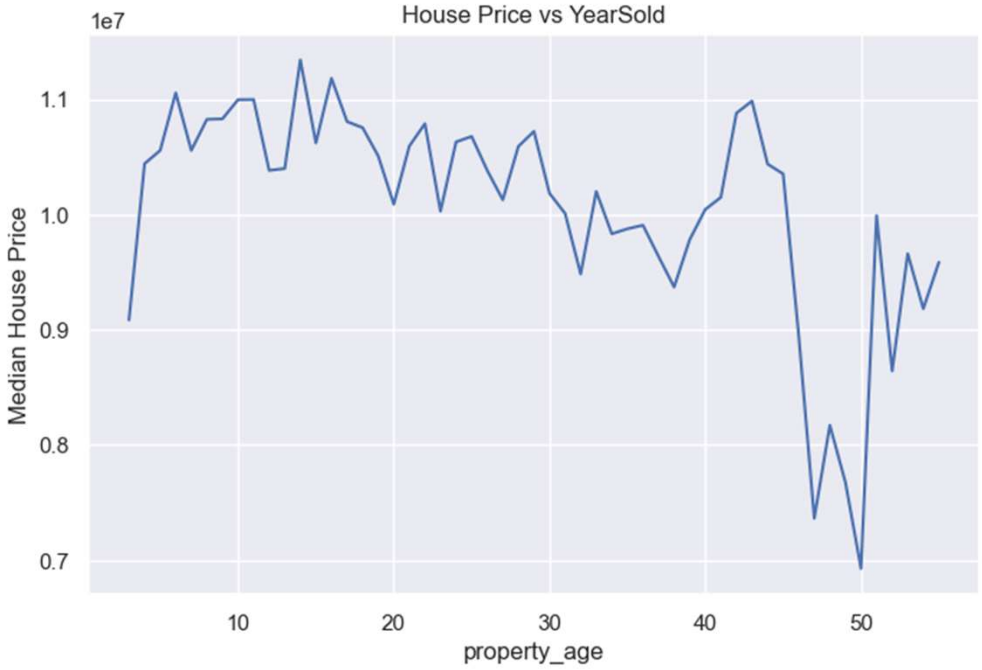


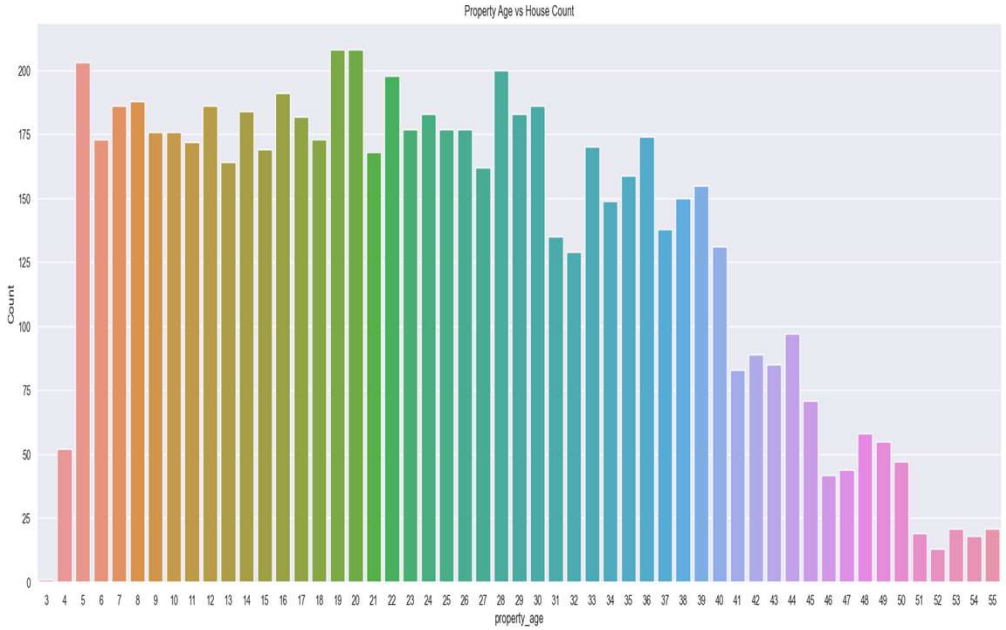


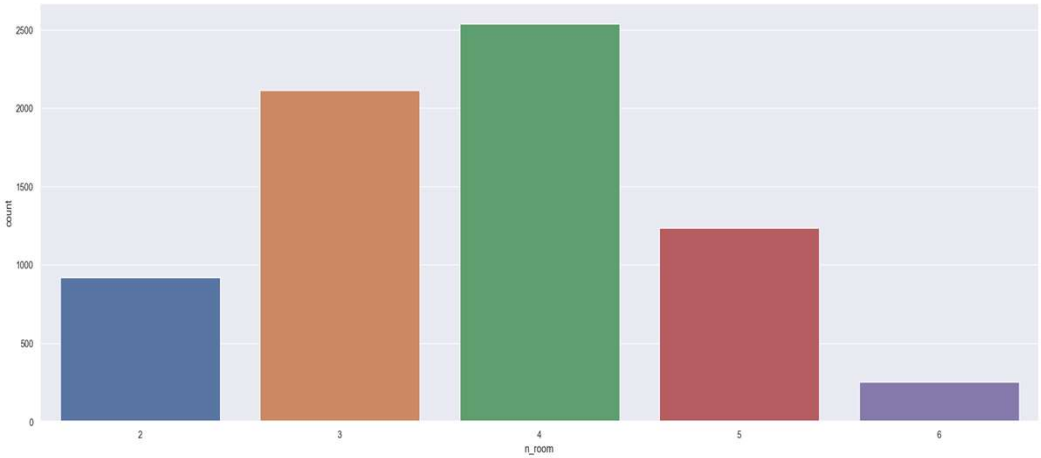
EXPLORATORY DATA ANALYSIS

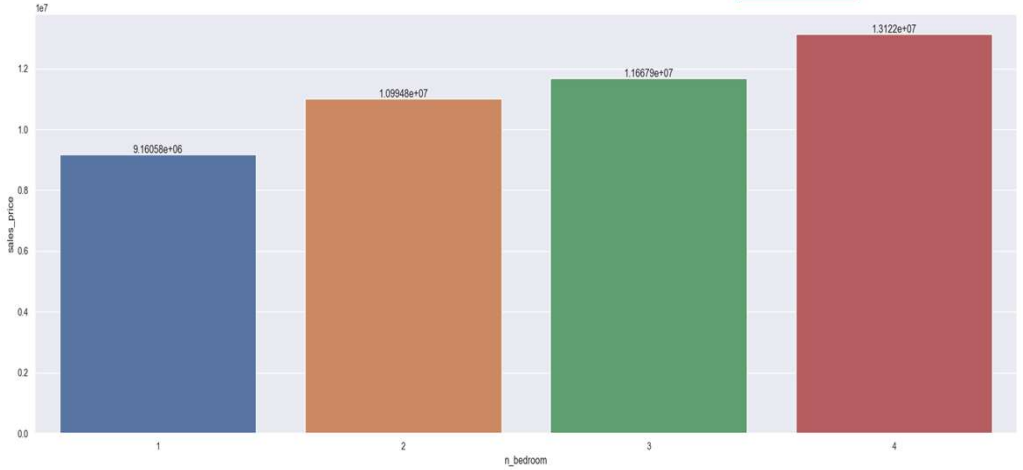
- My sale price normally distributed and right tail

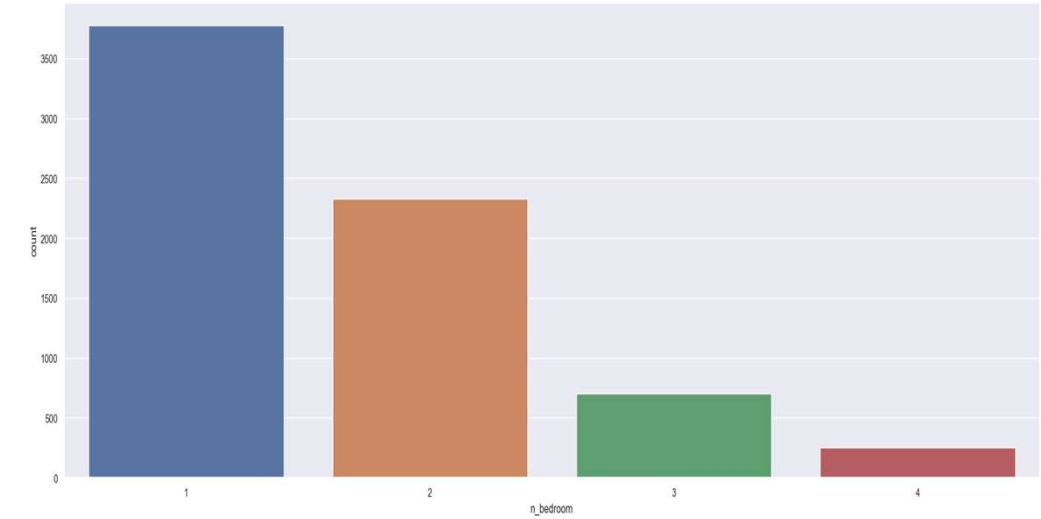
EXPLORATORY DATA ANALYSIS

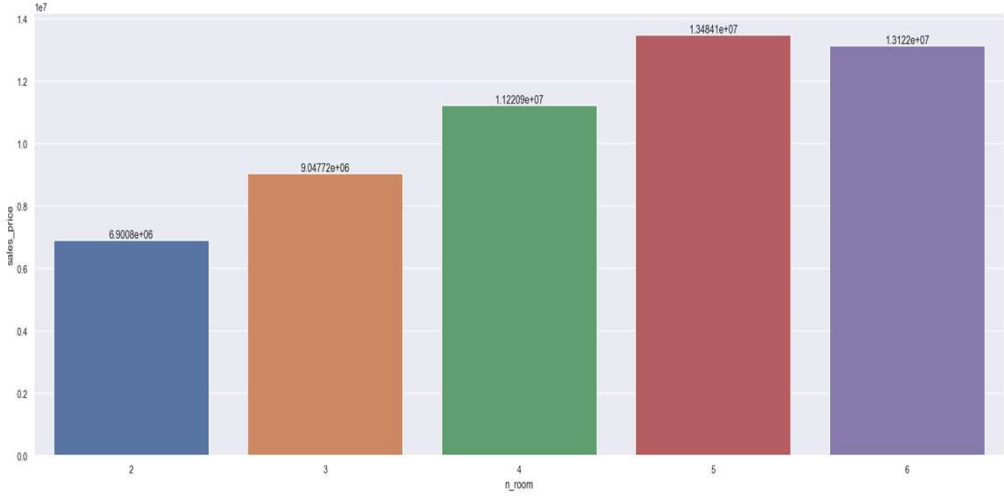




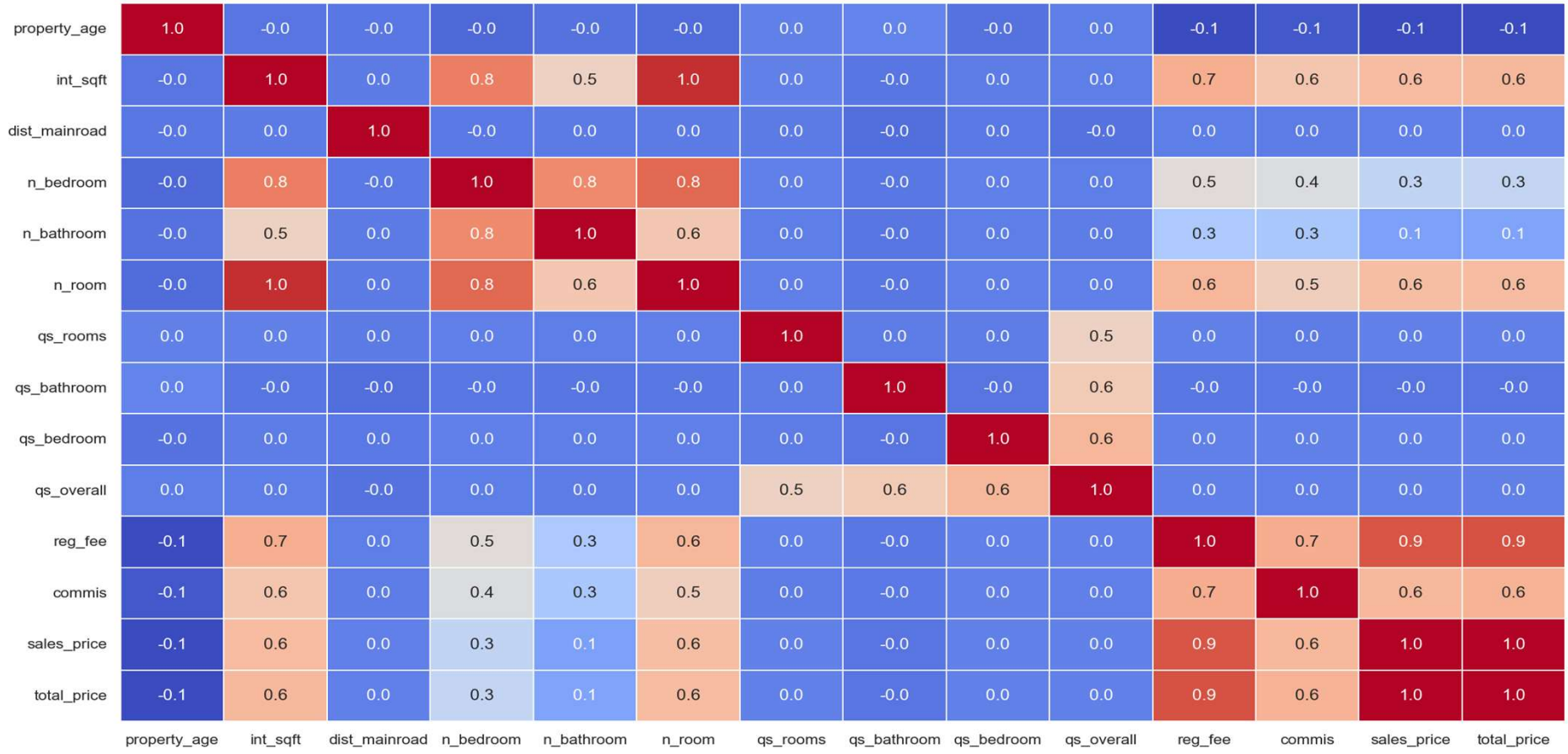

2500
2000
1500
1000
500
0
count
2
3
4
5
6
n_room


1e7
1.2
1.0
0.8
0.6
0.4
0.2
0.0
sales_price
9.16058e+06
1.09948e+07
1.16679e+07
1.3122e+07
1
2
3
4
n_bedroom


3500
3000
2500
2000
1500
1000
500
0
count
1
2
3
4
n_bedroom


1e7
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
sales_price
6.9008e+06
9.04772e+06
1.12209e+07
1.34841e+07
1.3122e+07
2
3
4
5
6
n_room

Heatmaps are very useful to find relations between two variables in a dataset. numerical columns there are some correlation between target and features which are shown in reddish colour.



DISTRIBUTION OF DATA COLUMN WISE

distribution of data in all the columns are normally distributed in most of the cases and in very few column the data is very slightly skewed.

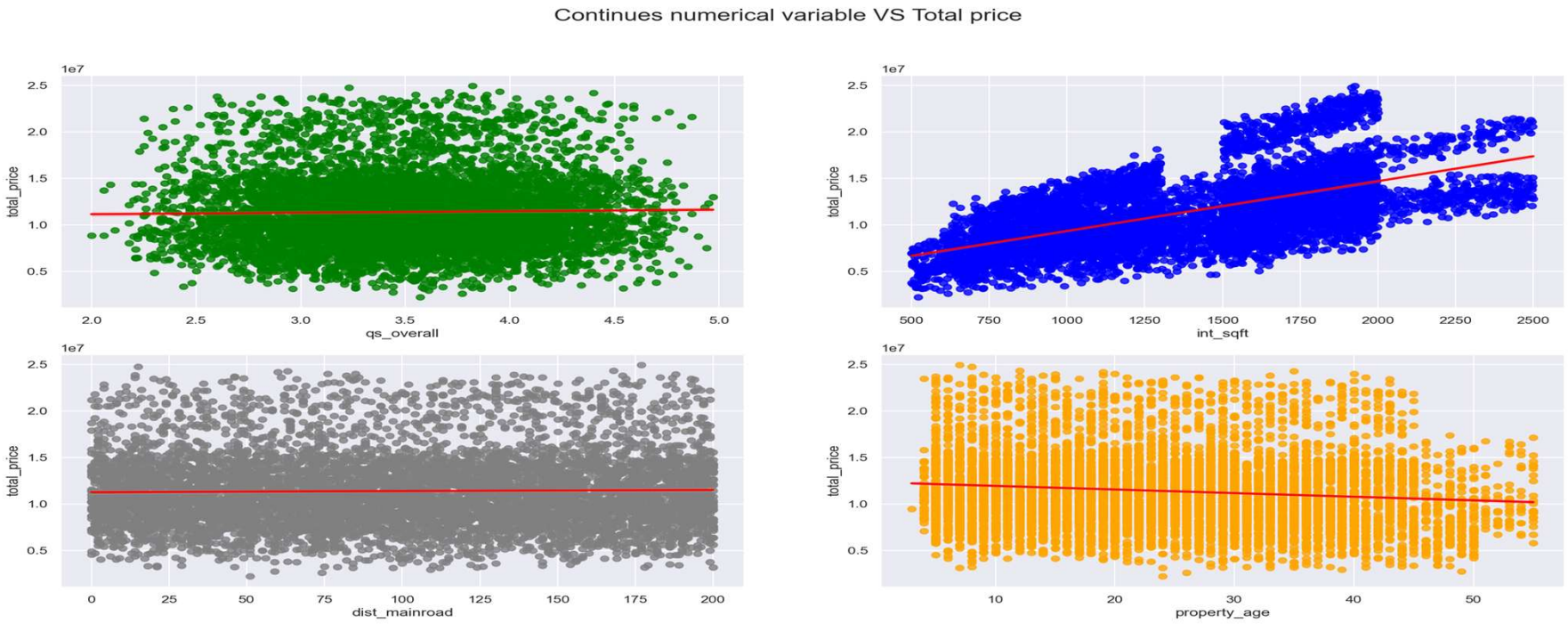
FEATURE ENGINEERING

- Feature engineering is the process of extracting features from raw data using domain knowledge and data mining techniques. These characteristics can help machine learning algorithms perform better. Feature engineering can be thought of as applied machine learning.
- In our dataset, dimensionality reduction techniques are used to remove rows that are insignificant in determining the house price.

NUMERICAL COLUMNS

- Continues numerical variable Here we are plotting all our Continues numerical variable columns with total price of the house to figure out, is there any relation between Continues numerical variable features column and total sales.
- In qs_overall we didn't find any relation so this feature will be of no use or very less us for us so we will drop it.
- In int_sqft we find good relation so this feature will be very important for us so we will keep it.
- In dist_mainroad we didn't find any relation so this feature will be of no use or very less us for us so we will drop it.
- In property_age we find small relation so this feature will be useful for us so we will keep it.

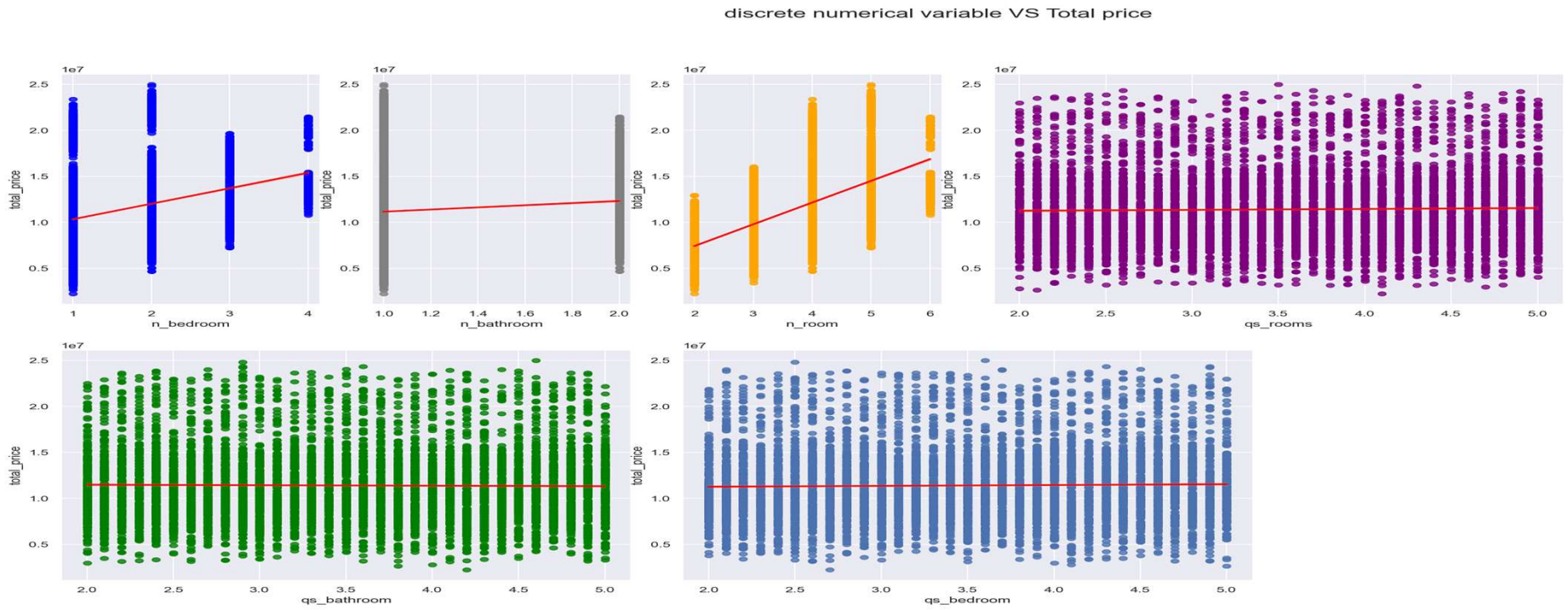
CONTINUES NUMERICAL VARIABLE VS TOTAL PRICE



DISCRETE NUMERICAL VARIABLE

- In n_bedroom we find good relation so this feature will be very important for us so we will keep it.
- In n_bathroom we find small relation so this feature will be useful for us so we will keep it.
- In n_room we find good relation so this feature will be very important for us so we will keep it.
- In qs_rooms we didn't find any relation so this feature will be of no use or very less us for us so we will drop it.
- In qs_bathroom we didn't find any relation so this feature will be of no use or very less us for us so we will drop it.
- In qs_bedroom we didn't find any relation so this feature will be of no use or very less us for us so we will drop it.

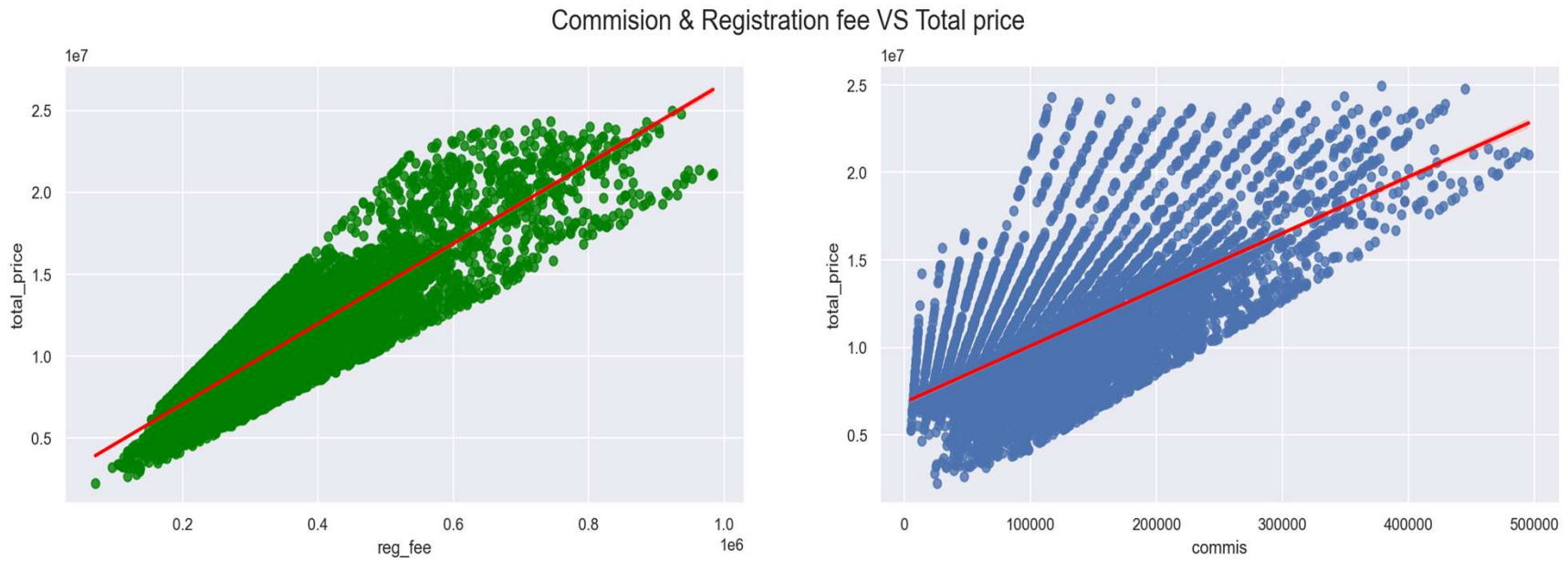
DISCRETE NUMERICAL VARIABLE VS TOTAL PRICE



COMMISSION AND REGISTRATION FEE COLUMNS

- On top of all numerical columns we are given registration fee and commission columns on which sales price don't depends on but these two columns are completely depended on sales column value. That means after determining the sales price the commission and registration fee are paid.
- Hence, these two columns don't directly contribute to determine the sale we can add value of these two column on sales column and try to predict total sale price. And at the end compare this result with only predicted sales price.

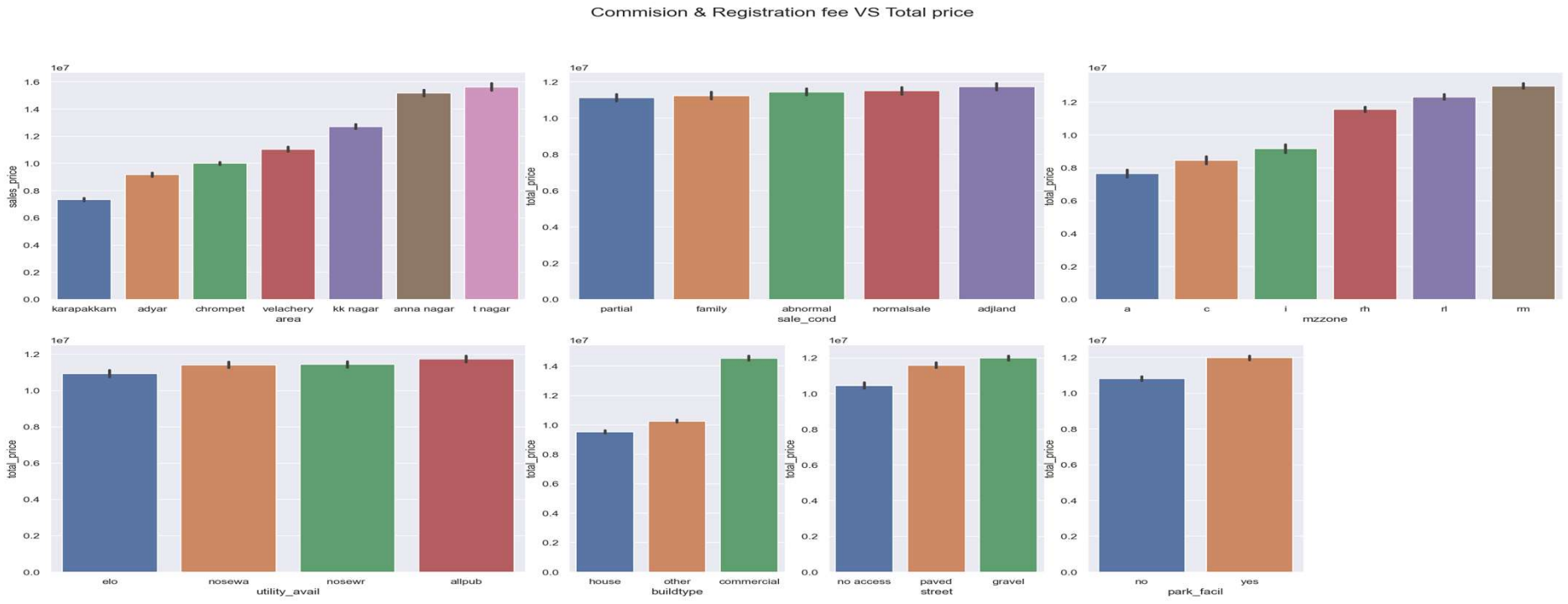
COMMISSION & REGISTRATION FEE VS TOTAL PRICE



CATEGORICAL COLUMNS

- In area when we sort column in respect of total_price we find good relation linear ordinal relation in this categorical feature and it will be important for us so we will keep it and do label encoding by preserving the order.
- In sale_cond when we sort column in respect of total_price we find good relation linear ordinal relation in this categorical feature and it will be important for us so we will keep it and do label encoding by preserving the order.
- In mzzone when we sort column in respect of total_price we find good relation linear ordinal relation in this categorical feature and it will be important for us so we will keep it and do label encoding by preserving the order.
- In utility_avain when we sort column in respect of total_price we find good relation linear ordinal relation in this categorical feature and it will be important for us so we will keep it and do label encoding by preserving the order.
- In buildtype when we sort column in respect of total_price we didn't find linear relation in this categorical feature but it may beimportant for us so we will keep it and do OneHotEncoding on this column data.
- In street when we sort column in respect of total_price we find good relation linear ordinal relation in this categorical feature and it will be important for us so we will keep it and do label encoding by preserving the order.
- In park_facil when we sort column in respect of total_price we find good relation linear ordinal relation in this categorical feature and it will be important for us so we will keep it and do label encoding by preserving the order.

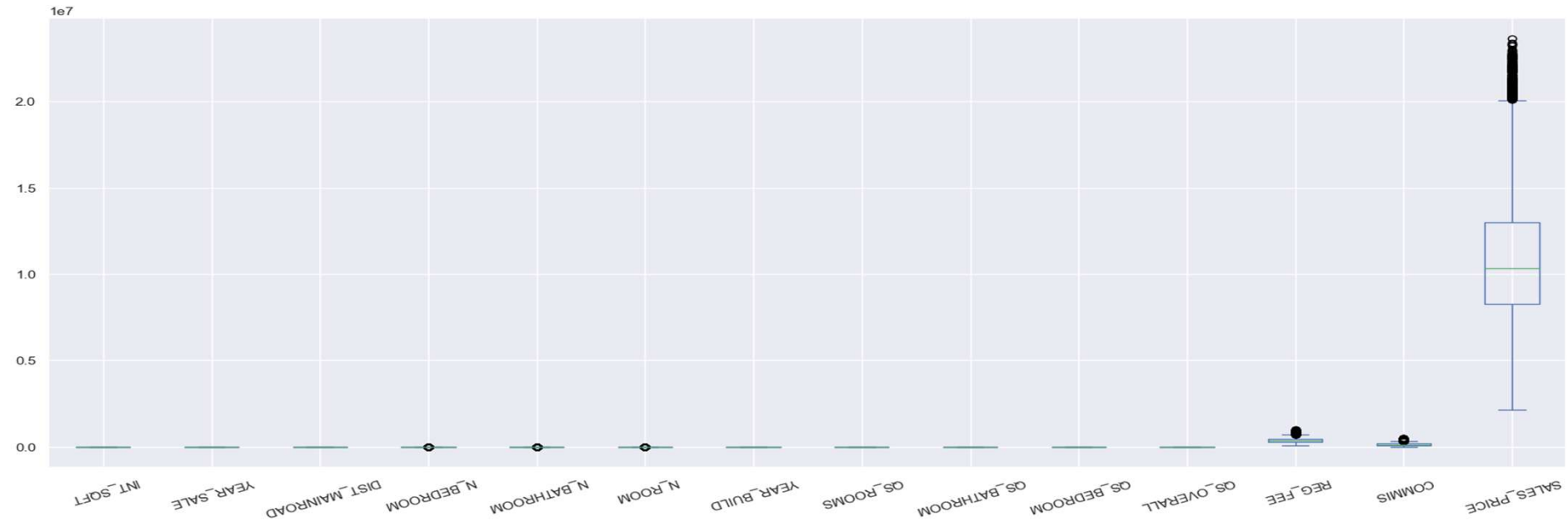
CATEGORICAL DATA VS TOTAL PRICE



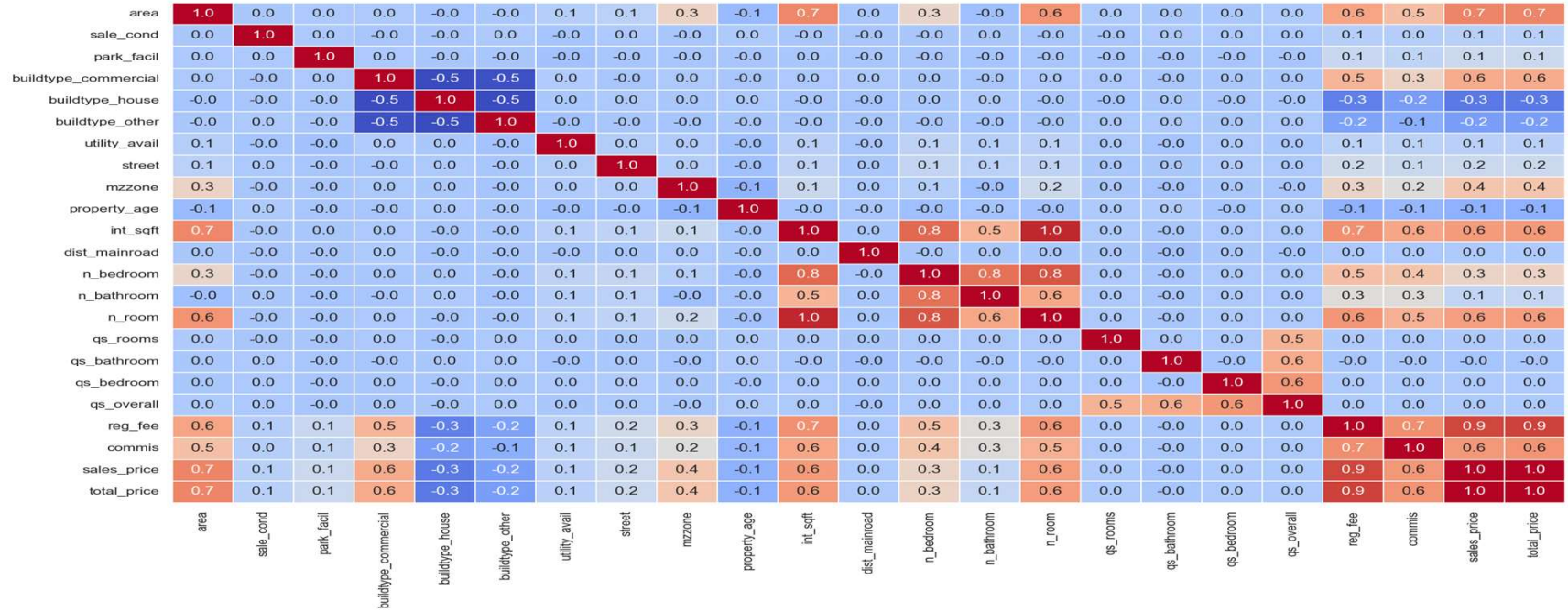
OUTLIER DETECTION

- An outlier is an observation that deviates from the overall pattern of a sample.
- Outlier detection techniques are including Z-Score or Extreme Value Analysis, probabilistic and statistical modelling, information theory models, standard deviation, and so on.
- To detect outliers in our dataset, we used simple domain knowledge of the real estate market.

OUTLIER DETECTION



CATEGORICAL COLUMNS VS TARGET



MODEL EVALUATION

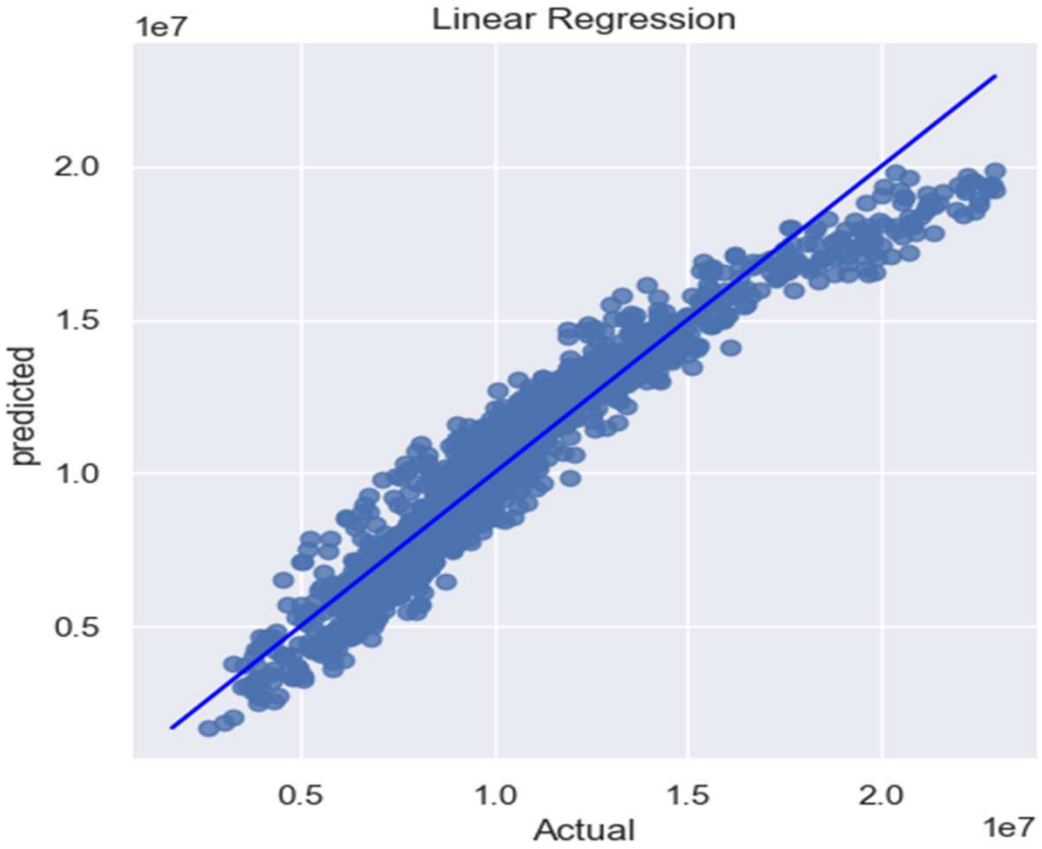
- Modelling is the process of training a machine learning algorithm to predict labels based on features.
- We have 80% training data 20% testing data
- We used Linear Regression, Gradient Boosting Regression, XGB Regression, Random Forest Regression, and the K Neighbours Regression algorithm to train and test the model.
- Our model has a 99% accuracy rate, which is quite good.

LINEAR REGRESSION

R2 Score : 0.920545

Mean Squared Error : 1.127599

Adjusted R2 Score : 0.920362

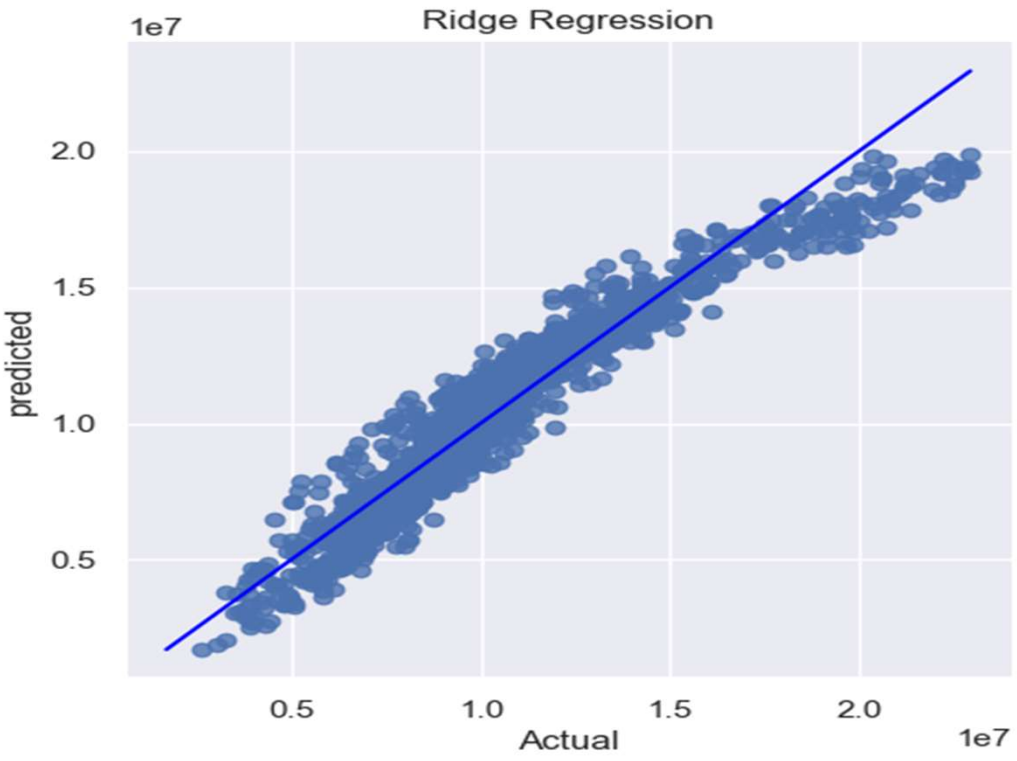


RIDGE REGRESSION

R2 Score : 0.920554

Mean Squared Error : 1.127484

Adjusted R2 Score : 0.920370

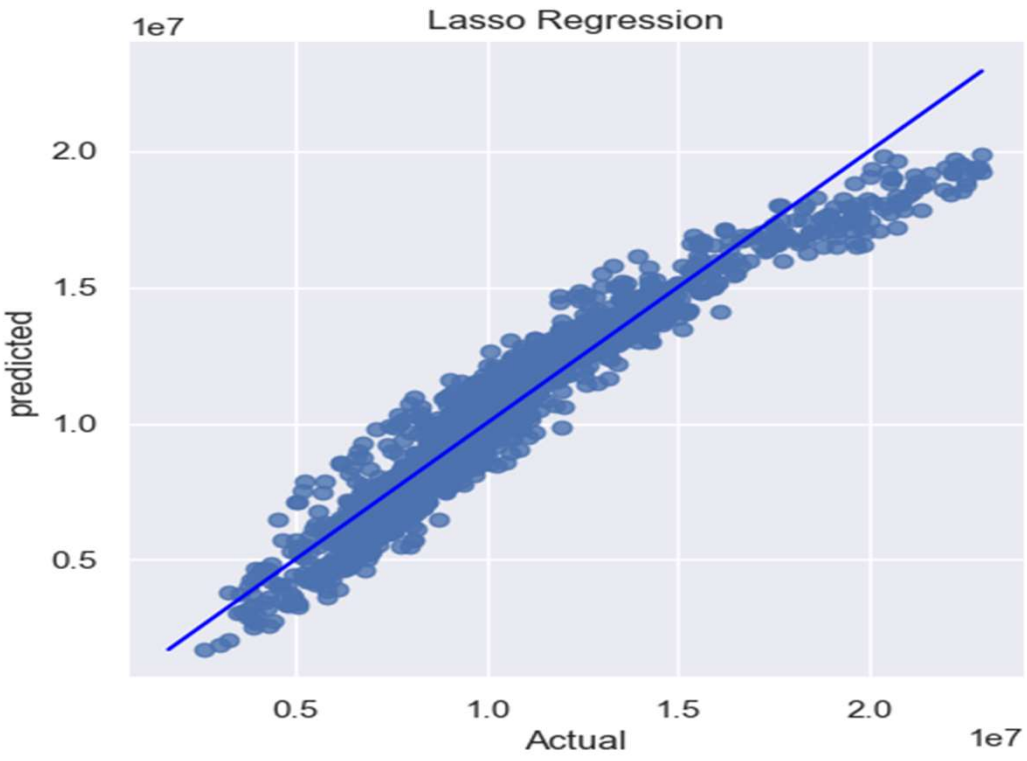


LASSO REGRESSION

R2 Score : 0.920545

Mean Squared Error : 1.127599

Adjusted R2 Score : 0.920362

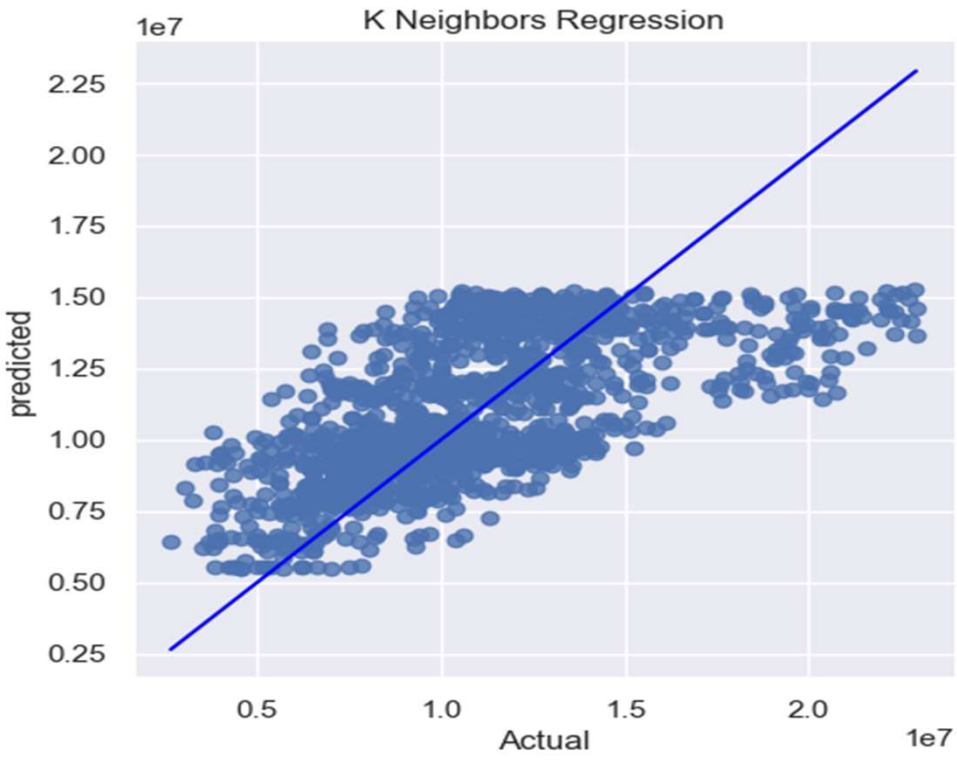


K NEAREST NEIGHBOUR

R2 Score : 0.432962

Mean Squared Error : 8.047259

Adjusted R2 Score : 0.431653

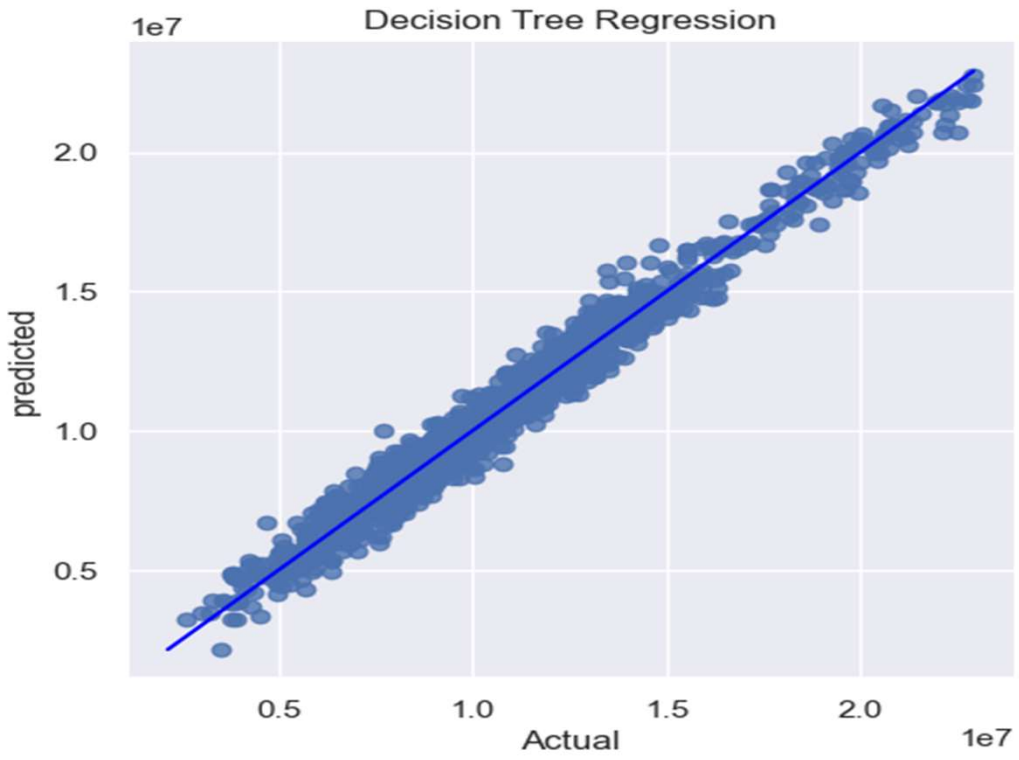


DECISION TREE

R2 Score : 0.974923

Mean Squared Error : 3.558925

Adjusted R2 Score : 0.974865

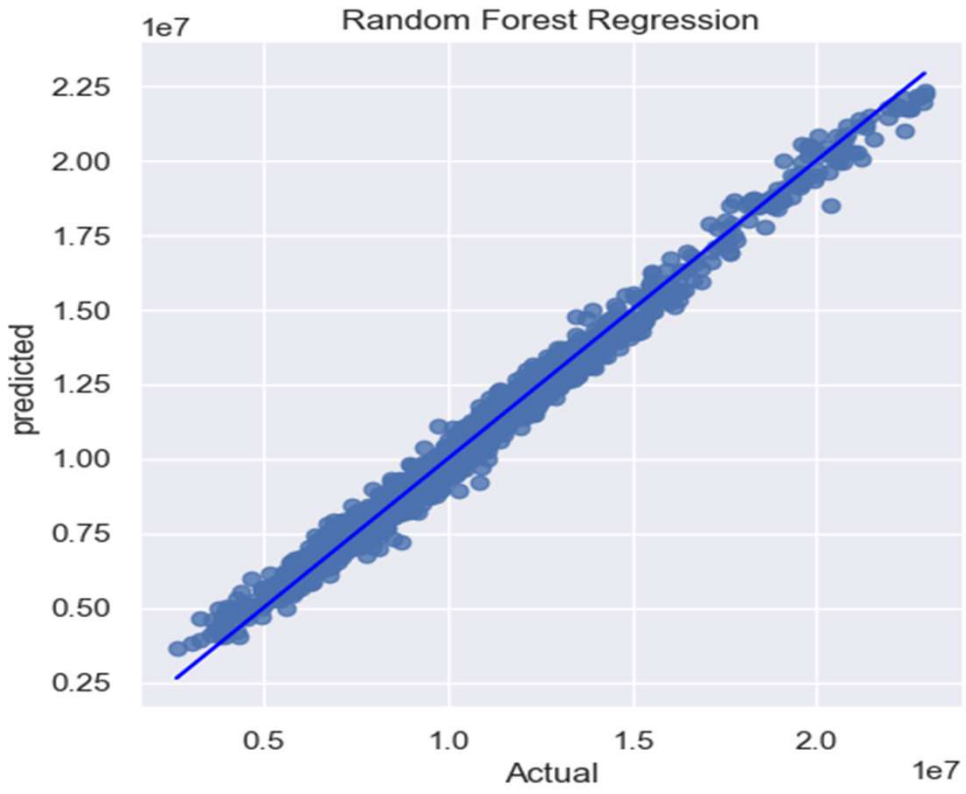


RANDOM FOREST

R2 Score : 0.987564

Mean Squared Error : 1.764950

Adjusted R2 Score : 0.987535

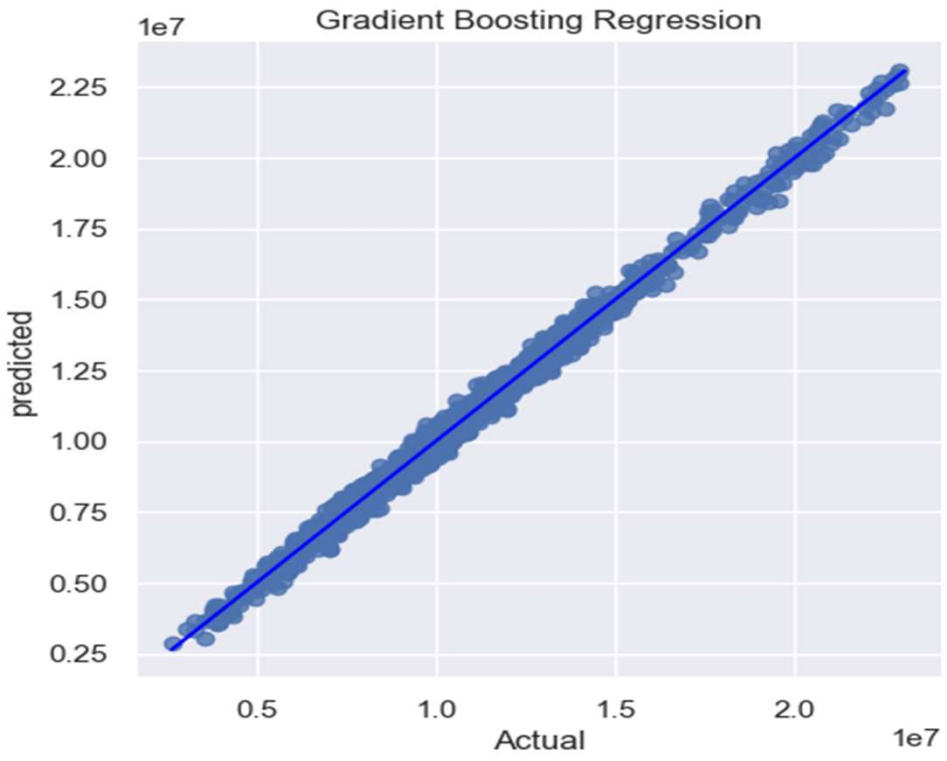


GRADIENT BOOSTING

R2 Score : 0.993724

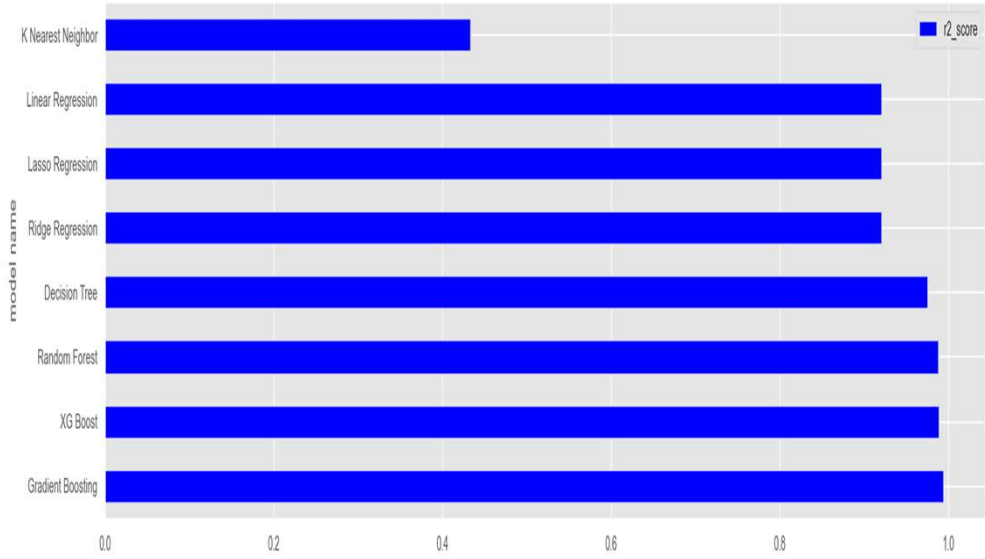
Mean Squared Error : 8.906226

Adjusted R2 Score : 0.993710



IMPLEMENTATION

	Model Name	R2 Score	Mean Squared Error	Ajusted R2
0	Linear Regression	0.920545	1.127599e+12	0.920362
1	Ridge Regression	0.920554	1.127484e+12	0.920370
2	Lasso Regression	0.920545	1.127599e+12	0.920362
3	K Nearest Neighbor	0.432962	8.047259e+12	0.431653
4	Decision Tree	0.974923	3.558925e+11	0.974865
5	Random Forest	0.987564	1.764950e+11	0.987535
6	Gradient Boosting	0.993724	8.906226e+10	0.993710
7	XG Boost	0.988278	1.663608e+11	0.988251



K-FOLD CROSS VALIDATION

- Cross-validation is a statistical method for estimating machine learning model skill.
- It is commonly used in applied machine learning to compare and select a model for a given predictive modelling problem because it is simple to understand, simple to implement, and produces skill estimates with lower bias than other methods.
- After applying the k-fold cross-validation method to our final dataset, we find that our accuracy rate is always greater than 98%.

RESULT/OUTPUT

- We developed a function to forecast house prices.
- Our function will be "predict price(location, sqft, bath, bhk)".
- When we enter the values into our function, it will calculate the house price for us.

Drawback

- It doesn't predict future prices of the houses mentioned by the customer.
- Due to this, the risk in investment in an apartment or an area increases considerably.
- To minimize this error, customers tend to hire an agent which again increases the cost of the process.
- This leads to the modification and development of the existing system.

THANK YOU