A PROJECT REPORT

on

House Price Prediction System

Submitted by

Ms. Simran Bhagirath Solanki

in partial fulfillment for the award of the degree

of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

under the guidance of

Prof. Vipul Saluja

Department of Computer Science

R. D. & S.H. National College & S. W. A. Science College

Bandra, Mumbai – 400050.

(Sem IV)

(2020 – 2021)

R.D. & S.H. National College & S. W.A. Science College,

Bandra, Mumbai – 400050.

Department of Computer Science

CERTIFICATE

This is to certify that Ms. Simran B Solanki of **M.Sc. – II (Sem IV)** class has satisfactorily completed the Project House Price Prediction System, to be submitted in the partial fulfillment for the award of **Master of Science** in **Computer Science** during the academic year **2020 – 2021**.

Project Guide **Co–ordinator,**

Department Computer Science

Date of Submission: **Signature of Examiner**

You Might not think that programmers are artists, but programming is an extremely creative profession. It's logic – based creativity.

- John Romero

Everybody in this country should learn to program a computer, because it teaches you how to think.

- Steve Jobs

DECLARATION

We, Nimish M Sane & Simran B Solanki, hereby declare that the project entitled "House Price Prediction System" submitted in the partial fulfillment for the award of **Master of Science** in **Computer Science** during the academic year **2020 – 2021** is our original work and the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Signature of the Student: 

Place: Mumbai

Date:

ACKNOWLEDGEMENT

We have taken efforts in the project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend my sincere thanks to all of them.

We wish to express our grateful thanks to the co-ordinator of Computer Science Department and project guide **Prof. Vipul Saluja** who gave us full support and valuable suggestions.

INDEX

CHAPTER 1:	**INTRODUCTION**	**1-2**
1.1	Motivation	1
1.2	Current System	1
1.3	Proposed System	2
CHAPTER 2:	**LITERATURE REVIEW**	**3-7**
2.1	Related Work	3
2.2	Reference	7
CHAPTER 3:	**OBJECTIVE**	**8**
CHAPTER 4:	**IMPLEMENTATION DETAILS**	**9-11**
CHAPTER 5:	**EXPERIMENTAL SET UP AND RESULTS**	**12-38**
5.1	Experimental Setup	12
5.2	Results of Exploratory Data Analysis	15
CHAPTER 6:	**IMPLEMENTATION DETAILS**	**39-47**
6.1	Analysis on Regression Models	39
6.2	Analysis on Classification Models	43
CHAPTER 7:	**CONCLUSION**	**48**
CHAPTER 8:	**FUTURE ENHANCEMENTS**	**49**
CHAPTER 9:	**SURVEY QUESTIONS**	**50-51**
CHAPTER 10:	**SCREEN LAYOUTS**	**52-63**
CHAPTER 11:	**SOURCE CODE**	**64-100**
CHAPTER 12:	**REFERENCES AND BIBLIOGRAPHY**	**101**

1 INTRODUCTION

1.1 Motivation:

Trends in housing prices indicate the current economic situation and also are a concern to the buyers and sellers.

Now a days everyone wishes for a house that suits their lifestyle and provides amenities according to their needs. House prices keep on changing very frequently which proves that house prices are often exaggerated. There are many factors that have an impact on house prices, such as the number of bedrooms and bathrooms. House price depends upon its location as well. A house with great accessibility to highways, schools, malls, employment opportunities, would have a greater price as compared to a house with no such accessibility. The prices change from place to place. Regardless of whether someone wants to sells or buy the house identifying the correct price is still a challenge.

The study on land price trend is deemed to be significant to support the decisions in urban planning. The real estate system is an unstable stochastic process. Investor's decisions are based on the market trends to reap maximum returns. In order to accurately estimate property prices, large amount of data that influences land price is required for analysis and modelling. The factors that affect the land price have to be studied and their impact on price has also to be modelled.

As the real estate is fast developing sector, the analysis and prediction of land prices using mathematical modelling and other scientific techniques is an immediate urgent need for decision making by all those concerned. The increase in population as well as the industrial activity is attributed to various factors, the most prominent being the recent sudden growth in the knowledge sector viz. Information Technology (IT) and Information technology enabled services.

Demand for land started of showing an upward trend and housing and the real estate activity started booming. All barren lands and paddy fields ceased their existence to pave way for multistore and high-rise buildings.

1.2 Current System:

In the current system if someone wants to buy or remove estimation of house price so they need to go and firstly find out an agent who will guide them, they also need to go in hunt from place to place and remove the estimation according

to their requirements and budget which is time consuming and there is a high chance of getting cheated if they do not know have knowledge. The customer also pays commission to the agent

1.3 Proposed System:

Due to this draw back there is a need of an autonomous system which will help people to find the correct house price based on their requirements. Such autonomous system can be build using various Machine learning algorithms and performing data analysis.

The proposed system will take different features such as location, carpet area, etc as input and various regression algorithms. The proposed system predicts house prices using a regression machine learning algorithm. In case you're going to sell or buy a house, you have to identify the accurate price. This regression model is built not only for predicting the price of the house which is ready for sale but also for houses that are under construction. Regression is a machine learning algorithm that encourages you to make predictions by taking input from the current measurable information, this current measurable information are different independent variables using which we will predict the dependent variable, in this case the dependent variable is the house price. As per this definition, a house's cost relies upon parameters, for example, the number of rooms, living region, area, and so forth. We will train the data using various regression model and then evaluate each model results by finding the error rate. The model having the least error rate will used for final prediction of the prices. By doing so we will be able to predict the most accurate prices of the house which can benefit both the buyers and sellers.

2 LITERATURE REVIEW

2.1 Related Work:

1) **The Paper proposed by Prof. Pradnya Patil**, In today's world, everyone wishes for a house that suits their lifestyle and provides amenities according to their needs. There are many factors that have to be taken into consideration for predicting house prices such as location, number of rooms, carpet area, how old the property is? and other basic local amenities. We will be using CatBoost algorithm along with Robotic Process Automation for real-time data extraction. Robotic Process Automation involves the use of software robots to automate the tasks of data extraction while machine learning algorithm is used to predict house prices with respect to the dataset. Keywords—Random Forest, CAT Boost, RPA, House Price Prediction.

Methodology Used:

In our proposed system, the initial step is data scraping. It is a technique with the help of which structured data can be extracted from the web or any application and saved to a database or spreadsheet or CSV file. After Data Extraction, we perform Data Cleaning. It refers to the modifications applied to the data before feeding it to the algorithm, with Random Forest Regression we can train the model efficiently for small amounts of data and can get pretty good result. It will, however quickly reach some extent where more samples won't improve the accuracy.

2) **The Paper Proposed by Science Direct** is commonly used to estimate the changes in housing price. Since housing price is strongly correlated to other factors such as location, area, population, it requires other information apart from HPI to predict individual housing price. There has been a considerably large number of papers adopting traditional machine learning approaches to predict housing prices accurately, but they rarely concern about the performance of individual models and neglect the less popular yet complex models. As a result, to explore various impacts of features on prediction methods, this paper will apply both traditional and advanced machine learning approaches to investigate the difference among several advanced models. This paper will also comprehensively validate multiple techniques in model implementation on regression and provide an optimistic result for housing price prediction.

Methodology:

Dataset used:- "Housing Price in Beijing" is a dataset containing more than 300, 000 data with 26 variables representing housing prices traded between 2009 and 2018. These variables, which served as features of the dataset, were then used to predict the average price per square meter of each house.

Operations performed on the above dataset:-

1) Data Pre processing
2) Data Analysis
3) Model Selection
   a) Random Forest :
      Random Forest is a kind of ensemble models that combines the prediction of multiple decision trees to create a more accurate final prediction.
   b) Extreme Gradient Boosting (XGBoost):
      XGBoost is a scalable machine learning system for tree boosting. The system is available as an open-source pack-age. The system has generated a significant impact and been widely recognized in various machine learning and data mining challenges.
   c) Light Gradient Boosting Machine (Light GBM):
      Light GBM is a gradient boosting framework that uses a tree-based learning algorithm. Light GBM has faster training speed with lower memory usage compare to XGBoost.
   d) Hybrid Regression:
      Hybrid Regression Model is a model that includes two or more different regression models.
   e) Stacked Generalization:
      The main idea of this method is to use the predictions of previous models as features for another model. This approach also utilizes the k-fold cross-validation technique to avoid overfitting.

3) **The Paper Proposed by NEELAM SHINDE & KIRAN GAWANDEI**, in this paper, we are predicting the sale price of the houses using various machine learning algorithms. Housing sales price are determined by numerous factors such as area of the property, location of the house, material used for construction, age of the property, number of bedrooms and garages and so on. This paper uses machine learning algorithms to build the prediction model for houses. Here, machine learning algorithms such as logistic regression and support vector

regression, Lasso Regression technique and Decision Tree are employed to build a predictive model. We have considered housing data of 3000 properties. Logistic Regression, SVM, Lasso Regression and Decision Tree show the R-squared value of 0.98, 0.96,0.81 and 0.99 respectively. Further, we have compared these algorithms based on parameters such as MAE, MSE, RMSE and accuracy. This paper also represents significance of our approach and the methodology.

Methodology:

Dataset Used: The dataset used in this project was an open source dataset from KaggleInc . It consists of 3000 records with 80 parameters that have the possibility of affecting the property prices.

Operations performed on the above dataset:-

1) Data Pre processing
2) Data Analysis
3) Application of Algorithms

Once the data is clean and we have gained insights about the dataset, we can apply an appropriate machine learning model that fits our dataset. We have selected four algorithms to predict the dependent variable in our dataset.

The algorithms that we have selected are basically used as classifiers but we are training them to predict the continuous values. The four algorithms are Logistic Regression, Support Vector Machine, Lasso Regression Technique and Decision Tree.

4) **The Paper Proposed by Nihar Bhagat, Ankit Mohokar, Shreyash Mane**, In this paper they have mentioned that people looking to buy a new home tend to be more conservative with their budgets and market strategies. The existing system involves calculation of house prices without the necessary prediction about future market trends and price increase. The goal of the paper is to predict the efficient house pricing for real estate customers with respect to their budgets and priorities. By analyzing previous market trends and price ranges, and also upcoming developments future prices will be predicted. The functioning of this paper involves a website which accepts customer's specifications and then combines the application of multiple linear regression algorithm of data mining. This application will help customers to invest in an estate without approaching an agent. It also decreases the risk involved in the transaction.

Methodology:

The proposed system works on Linear Regression Algorithm.

The database of property rates contains attributes like quarter, upper, average and lower, where each year from 2009 is divided into 4 quarters (q1: January-March, q2: April-June, q3: July- September, q4: October-December). The column upper consists of the average values of the houses that are high in prices, likewise average and lower column consists of average values of middle range and low range house . In order to use linear regression the quarter attribute is assigned on x-axis and the values of rates on y-axis. For each of the attribute linear regression is performed once. The x-axis being independent is the choice available to the user to select from a dropdown list.

5) **This paper proposed by Adyan Nur Alfiyatin, Hilman Taufiq, Ruth Ema Febrita & Wayan Firdaus Mahmudy** they have proposed that House prices increase every year, so there is a need for a system to predict house prices in the future. House price prediction can help the developer determine the selling price of a house and can help the customer to arrange the right time to purchase a house. There are three factors that influence the price of a house which include physical conditions, concept and location. This research aims to predict house prices based on NJOP houses in Malang city with regression analysis and particle swarm optimization (PSO). PSO is used for selection of affect variables and regression analysis is used to determine the optimal coefficient in prediction. The result from this research proved combination regression and PSO is suitable and get the minimum prediction error obtained which is IDR 14.186.

Methodology:

Dataset Used: In this research, we use house price data based on NJOP from Land and Building Tax (PBB) payment structure. Due to limited access to the data, this study used 9 houses data in time series scattered in Malang City area, within 2014-2017.

Operations performed on the above dataset:-

1) Regression analysis:
The prediction model used in this research is hedonic pricing, the suitable model using regression

2) Particle Swarm Optimization (PSO):
Particle Swarm Optimization (PSO) PSO is a stochastic optimization method that represents solutions as particle. Amount number of particles are generated randomly, where each particle consists of some dimensions of xi position and velocity vi. Each particle will measure its fitness value.

3) Testing Methods:
The model developed in this research will be tested using several methods such as Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE)
MAPE is calculated by making an average percentage of the absolute error of each predicted result. Thus, MAPE can indicate how much prediction error.
MAE calculate the average of absolute error for each predicted result. MAE is useful when measuring errors in certain units.
RMSE is used to calculate predicted performance by considering the prediction error of each data.

2.1 Reference:

https://www.irjet.net/archives/V7/i3/IRJET-V7I31123.pdf

https://www.sciencedirect.com/science/article/pii/S1877050920316318

http://www.iraj.in/journal/journal_file/journal_pdf/12-477-153396274234-40.pdf

https://www.ijcaonline.org/archives/volume152/number2/bhagat-2016-ijca-911775.pdf

https://thesai.org/Downloads/Volume8No10/Paper_42-Modeling_House_Price_Prediction_using_Linear_Regression.pdf

3 OBJECTIVE

The main objective of the project is to predict accurate house prices based on various features.

1. Create an effective price prediction model and Validate the model's prediction accuracy.

2. Uses a classical technique called regression and try to give an analysis of the results obtained. It helps establishes the relationship strength between dependent variable and other changing independent variable known as label attribute and regular attribute respectively. Regression displays continuous value of the dependent variable that is label attribute that is used for prediction.

3. The aim is to predict the efficient house pricing for real estate customers with respect to their budgets and priorities. By analyzing previous market trends and price ranges

4. Traditional house price prediction is based on cost and sale price comparison lacking of an accepted standard and a certification process. Therefore, the availability of a house price prediction model helps fill up an important information gap and improve the efficiency of the real estate mark

5. Analyzing the dataset and finding the features that strongly affect the prices of the houses.

6. Finding the best fit regression model that will have minimum error rate which will result in predicting accurate prices.

4 IMPLEMENTATION DETAILS

Data Collection method:

1. We will collect data from though google form and transfer the data into an excel sheet, using which we will perform further operations.

   Target Population:
   Age vise:- 25-65
   Area vise:- Western Line of Mumbai

   Sample Population: 200

Operations to be performed on Dataset:

1. **Data Cleaning**:
   Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. If data is incorrect, outcomes and algorithms are unreliable, even though they may look correct. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset.

2. **Exploratory Data Analysis**:
   Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

   It is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

   A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal modeling or hypothesis testing task.

3. **Feature Selection:**
   Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model. The data features that you use to train your machine learning models have a huge influence on the performance you can achieve.

Irrelevant or partially relevant features can negatively impact model performance.

Feature selection and Data cleaning should be the first and most important step of your model designing.

4. **Data Transformation:**
   It is rare to have collected data solely to make predictions. Consequently, the data you have available may not be in the right format or may require transformations to make it more useful. Data Transformation activities and techniques include:

   1) Categorical Encoding
      Label encoding and One Hot Encoding converts categorical variables to numerical representation, something that is machine-readable.

   2) Scaling
      Scaling is a method of transforming data into a particular range. This is important when using regression algorithms and algorithms using Euclidean distances (e.g. KNN, or K-Means) as they are sensitive to the variation in magnitude and range across features.

      The goal of scaling is to change the values of each numerical feature in the data set to a common scale. By doing so, changes in different features become more comparable. Scaling can be done with normalization (min-max scaling) or z-score standardization.

   3) Feature Engineering
      Feature engineering is the process of creating new features based upon knowledge about current features and the required task. It is important to have a clear understanding of the data to do this step, this may require you working with a subject matter expert.

5. **Splitting the dataset into the Training set and Test set:**
   The train-test split is a technique for evaluating the performance of a machine learning algorithm.
   It can be used for classification or regression problems and can be used for any supervised learning algorithm.

   The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the

input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset.

Train Dataset: Used to fit the machine learning model.
Test Dataset: Used to evaluate the fit machine learning model.

6. **Regression Model:**
   Regression Model is a predictive modelling technique that analyzes the relation between the target or dependent variable and independent variable in a dataset.

   The different types of regression techniques get used when the target and independent variables show a linear or non-linear relationship between each other, and the target variable contains continuous values. The regression technique gets used mainly to determine the predictor strength and in case of cause & effect relation.

7. **Evaluating Regression Model:**
   Model evaluation is very important in data science. It helps you to understand the performance of your model and makes it easy to present your model to other people.

   There are many different evaluation metrics out there but only some of them are suitable to be used for regression.

   Regression model will be evaluated and the model which has maximum accuracy will be selected for performing prediction

5 EXPERIMENTAL SET UP AND RESULTS

5.1 Experimental Set

A) Programming Language

1. Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Python supports modules and packages, which encourages program modularity and code reuse.
The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

We have chosen Python as it is easier for me to understand it compared to R Programming Language, it is widely used in data science and also supports wide set of libraries essential for Data Science purposes.

B) Tools:

1. Jupyter Notebook:

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.

Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

We have used Jupyter notebook for data preprocessing, visualizations and applying ml models.

2. Spyder

Spyder is an opensource cross-platform integrated development environment for scientific programming in the Python language.

We have used Spyder for combining all the data findings from Jupyter notebook and merge it into one, making functions for Streamlit, to run each function one by one and to display visualizations on the web app.

C) Libraries

1. Pandas

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the Data Frame.

Data Frames allow you to store and manipulate tabular data in rows of observations and columns of variables.

We have used Pandas for data cleaning, manipulation, translation and other preprocessing.

2. Numpy

NumPy stands for 'Numerical Python'. It is an open-source Python library used to perform various mathematical and scientific tasks. It contains multi-dimensional arrays and matrices, along with many high-level mathematical functions that operate on these arrays and matrices.

We have used numpy for performing operations on array

3. Matplotlib

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable opensource alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

We have used matplotlib for data visualization

4. Seaborn

Seaborn is a plotting library that offers a simpler interface, sensible defaults for plots needed for machine learning. It provides a high-level interface for drawing attractive and informative statistical graphics.

We have used seaborn for data visualization

5. Altair

Altair is a declarative statistical visualization library for Python, based on Vega and Vega-Lite, and the source is available on GitHub.

With Altair, you can spend more time understanding your data and its meaning. Altair's API is simple, friendly and consistent and built on top of the powerful Vega-Lite visualization. This elegant simplicity produces beautiful and effective visualizations with a minimal amount of code.

We have used altair for data visualization

6. Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python

We have used Sklearn for performing data transformation, implementing regression and classification models.

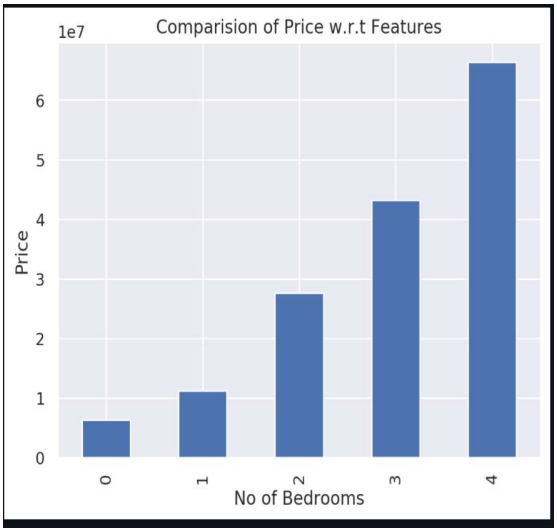
7. Streamlit

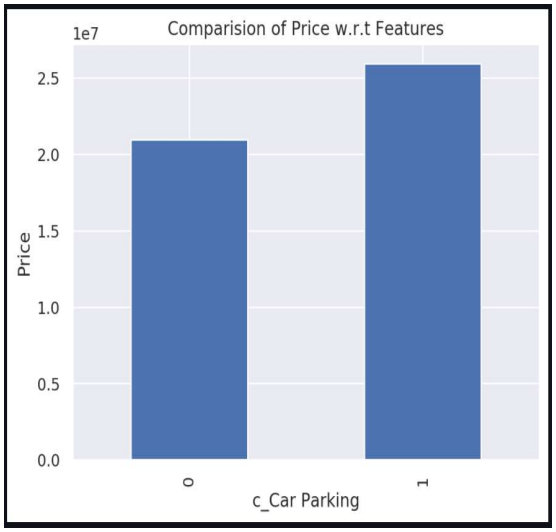
Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. Through streamlit you can build and deploy powerful data apps

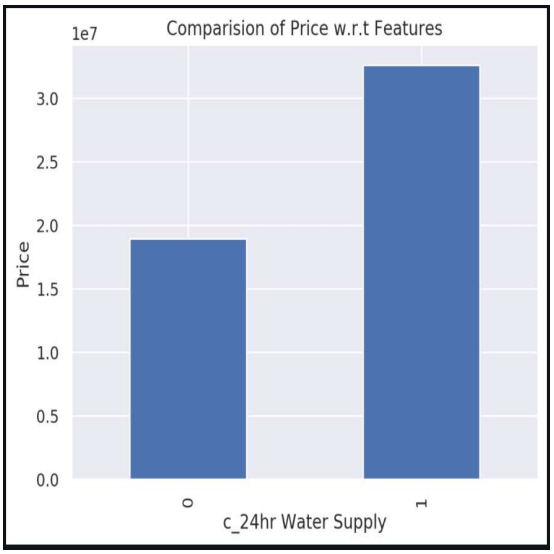
We have used streamlit to turn our work into a single interactive web application

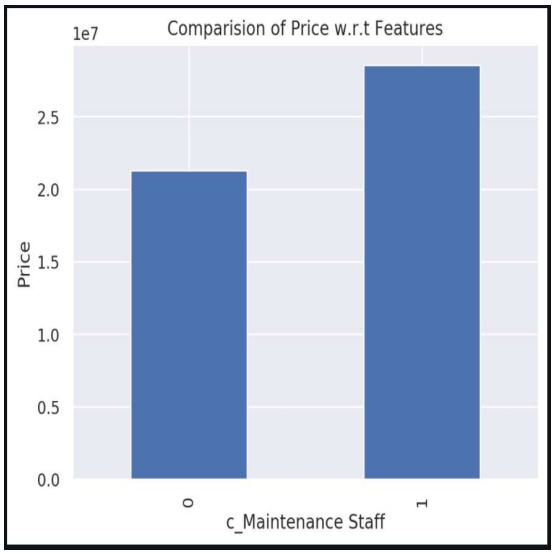
5.2 Results of Exploratory Data Analysis

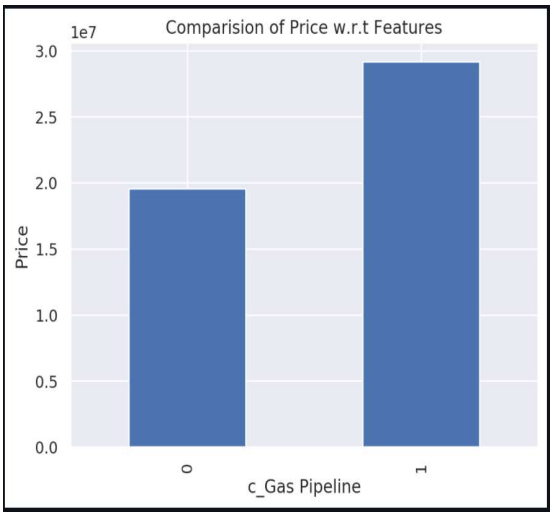
1) Bar graph to represent Comparison of House Price w.r.t Features

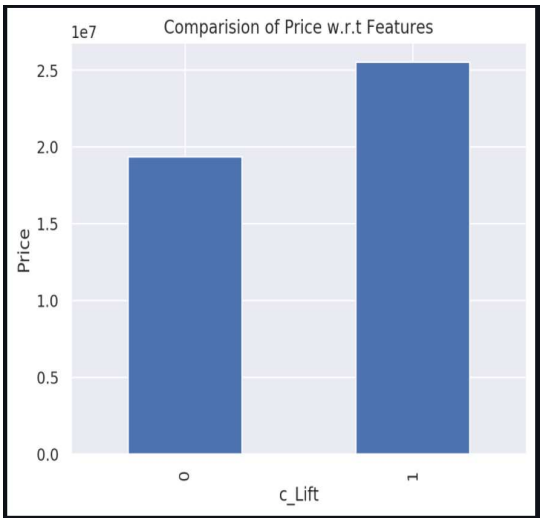


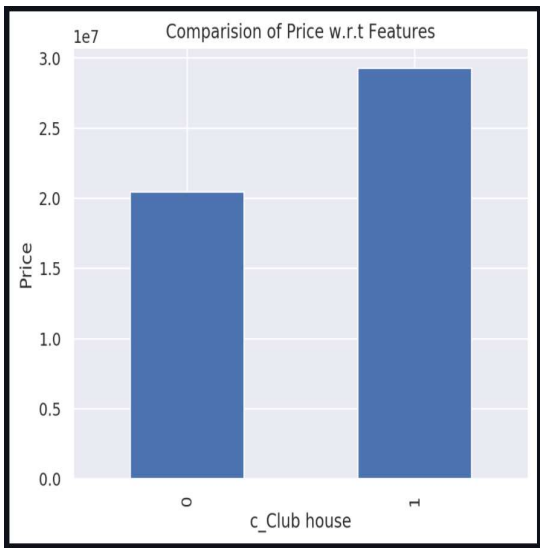


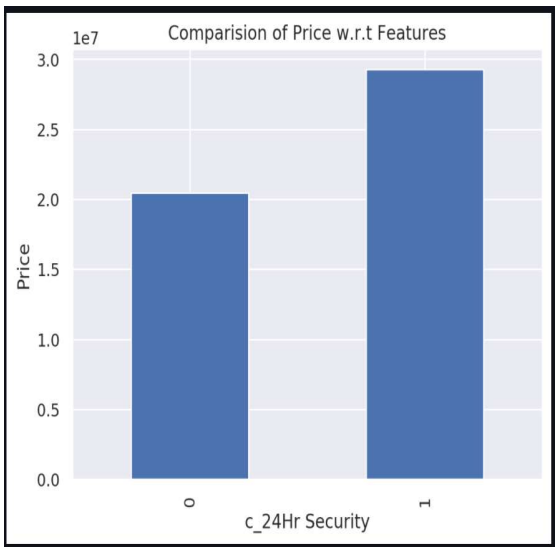


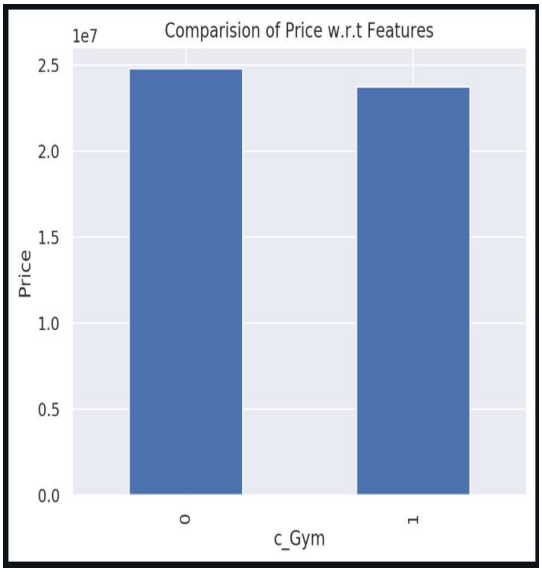


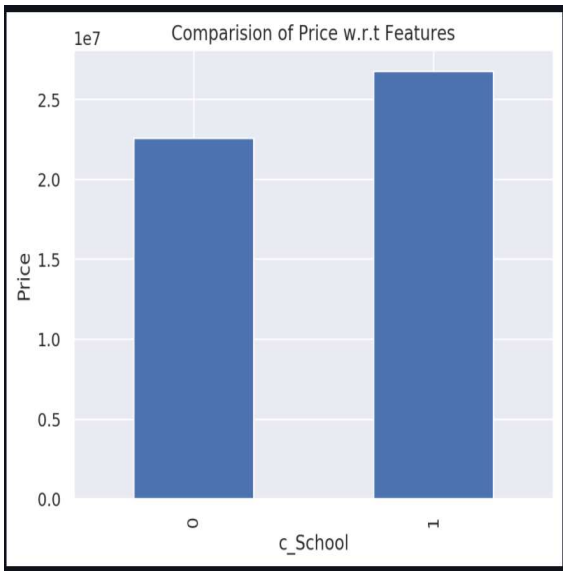

Comparision of Price w.r.t Features
1e7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Price
0
1
c_Gas Pipeline

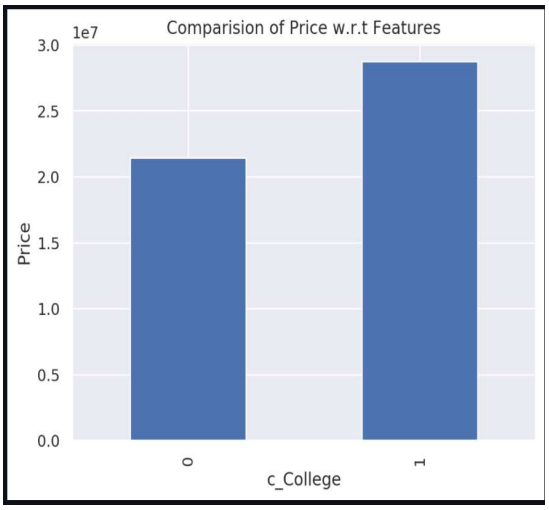

Comparision of Price w.r.t Features
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Price
0
1
c_Lift

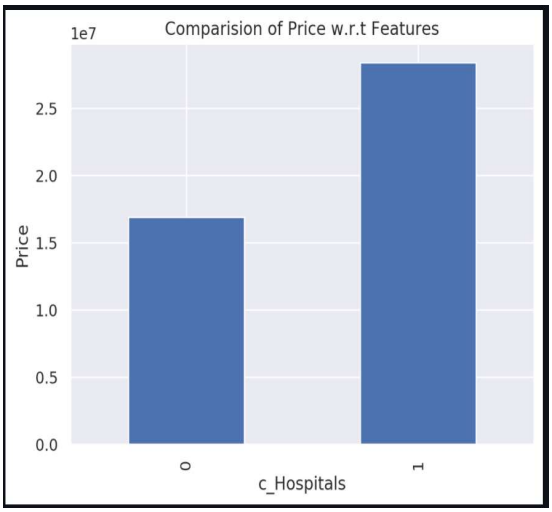

Comparision of Price w.r.t Features
1e7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Price
0
1
c_Club house

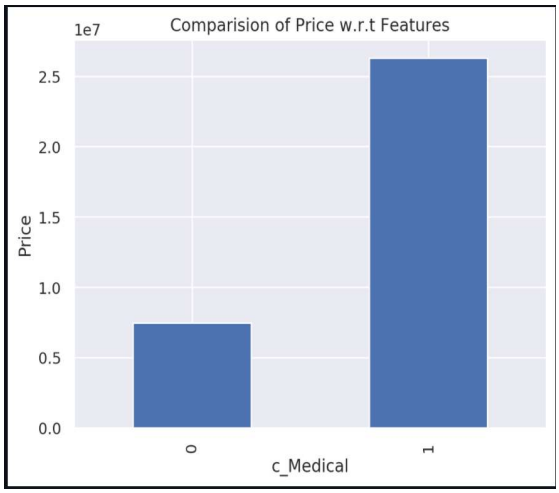

Comparision of Price w.r.t Features
1e7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Price
0
1
c_24Hr Security

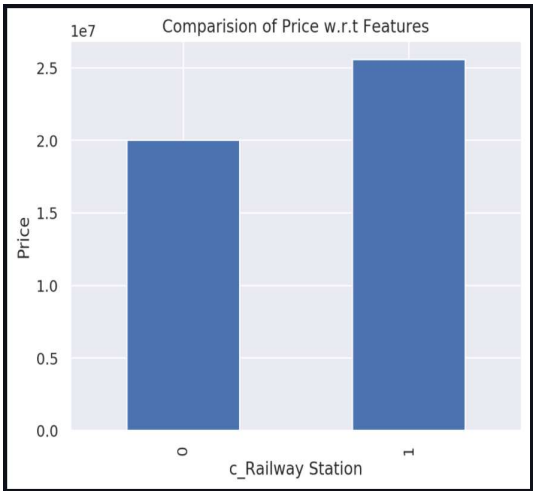

Comparision of Price w.r.t Features
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Price
0
1
c_Gym

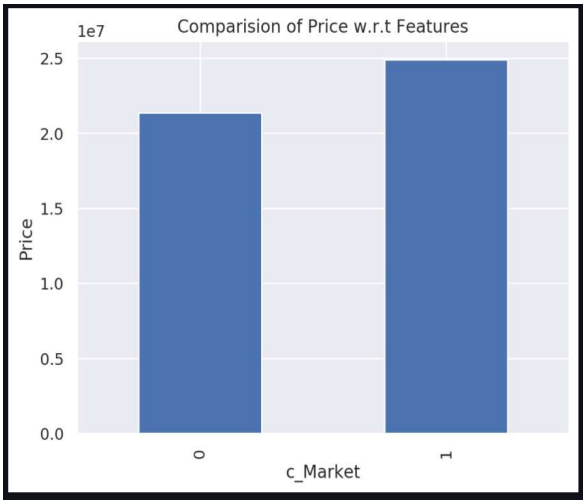

Comparision of Price w.r.t Features
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Price
0
1
c_School


Comparision of Price w.r.t Features
1e7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Price
0
1
c_College


Comparision of Price w.r.t Features
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Price
0
1
c_Hospitals


Comparision of Price w.r.t Features
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Price
0
1
c_Medical

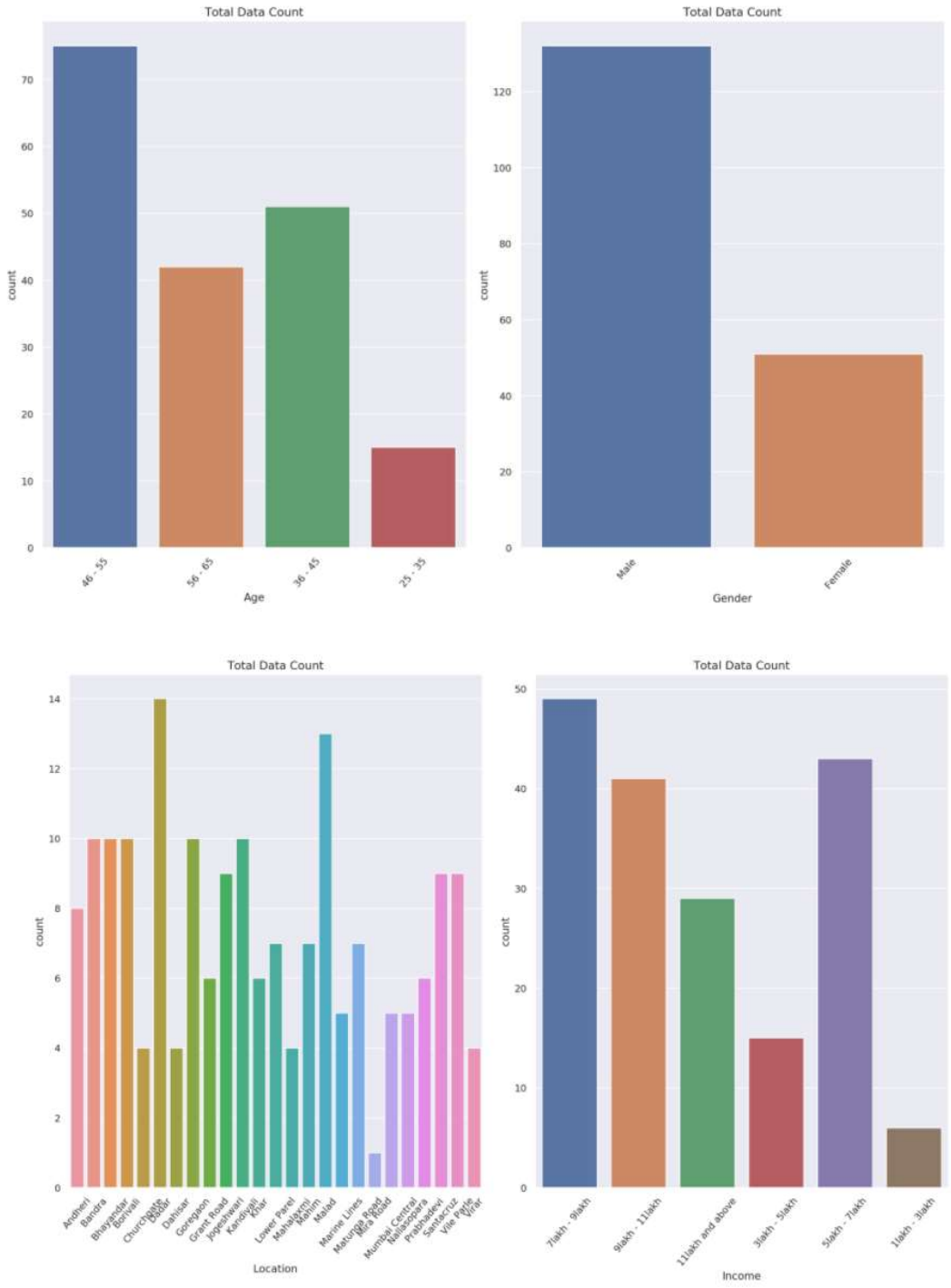

Comparision of Price w.r.t Features
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Price
0
1
c_Railway Station

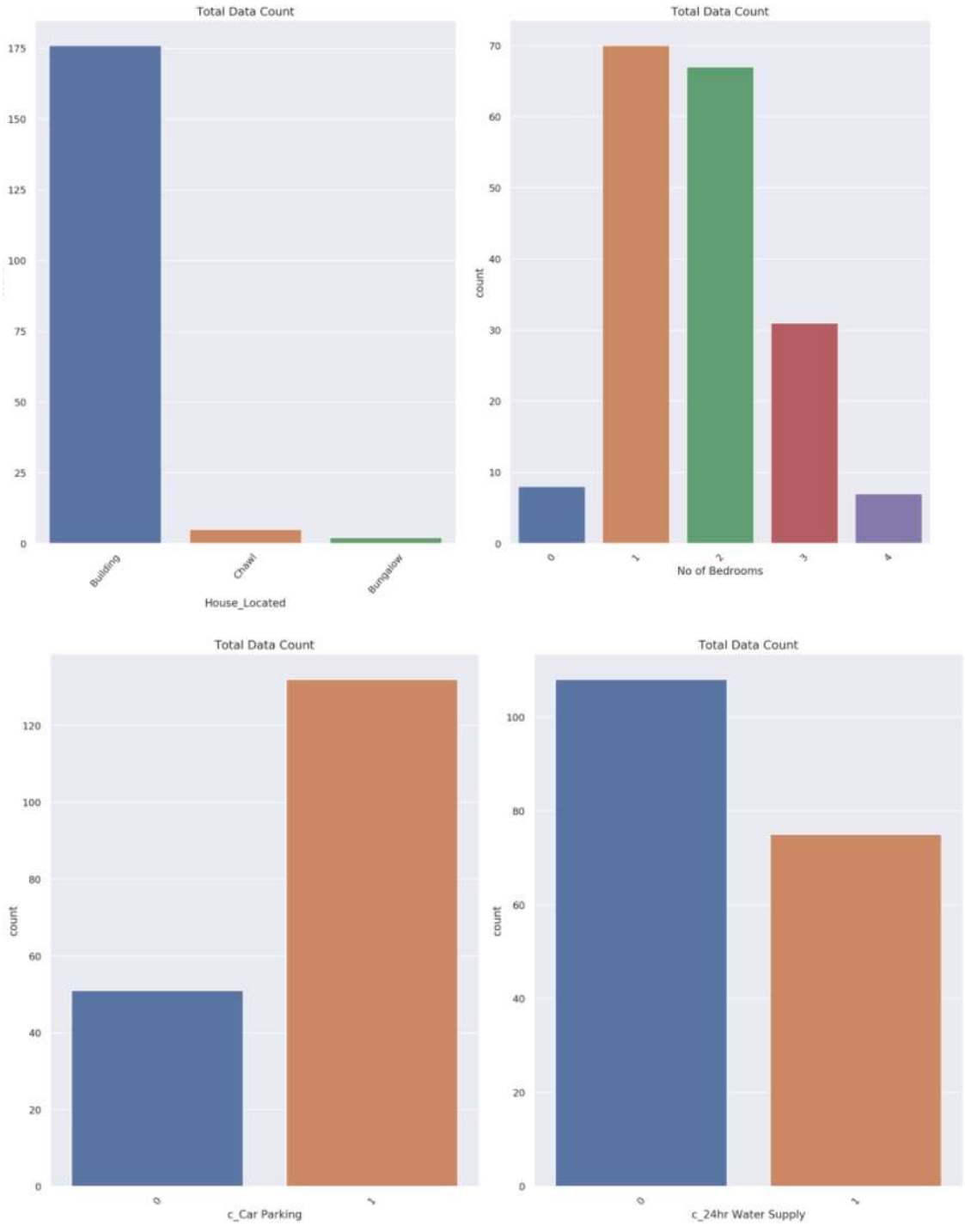

Comparision of Price w.r.t Features
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Price
0
1
c_Market

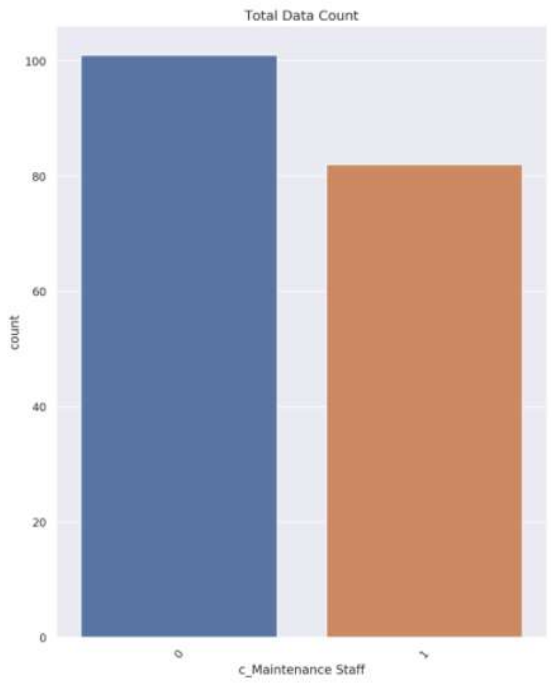
Conclusion:

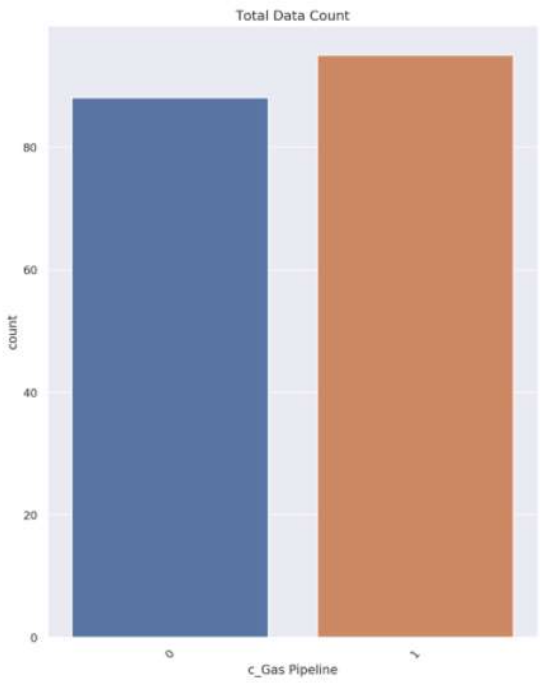
From the above graphs we can conclude that features such as No of BedRooms, 24Hr Water Supply, Gas Pipeline, Lift and medical are highly influenced for increase in price whereas other features does not affect price as much.

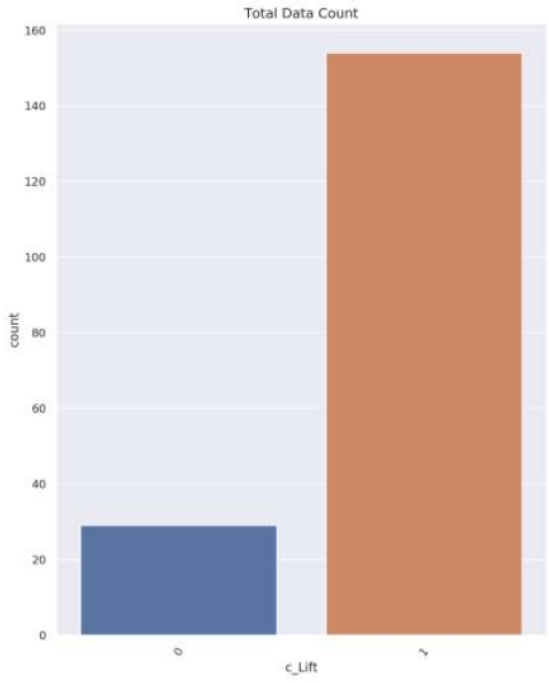
2) Count Plot to represent total data count

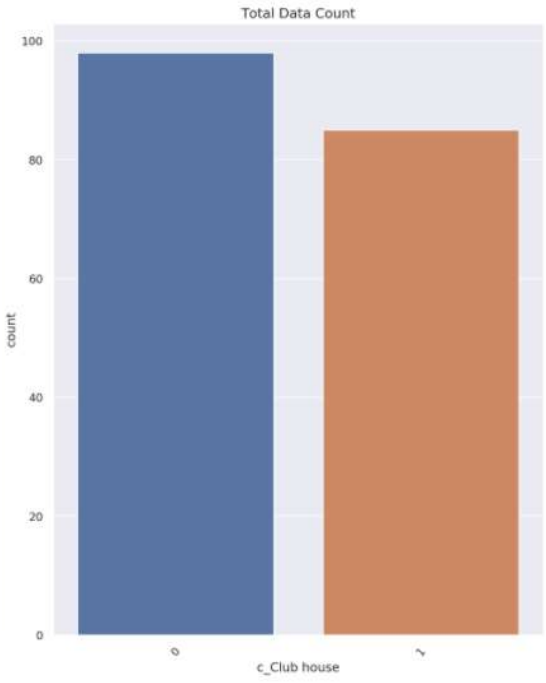


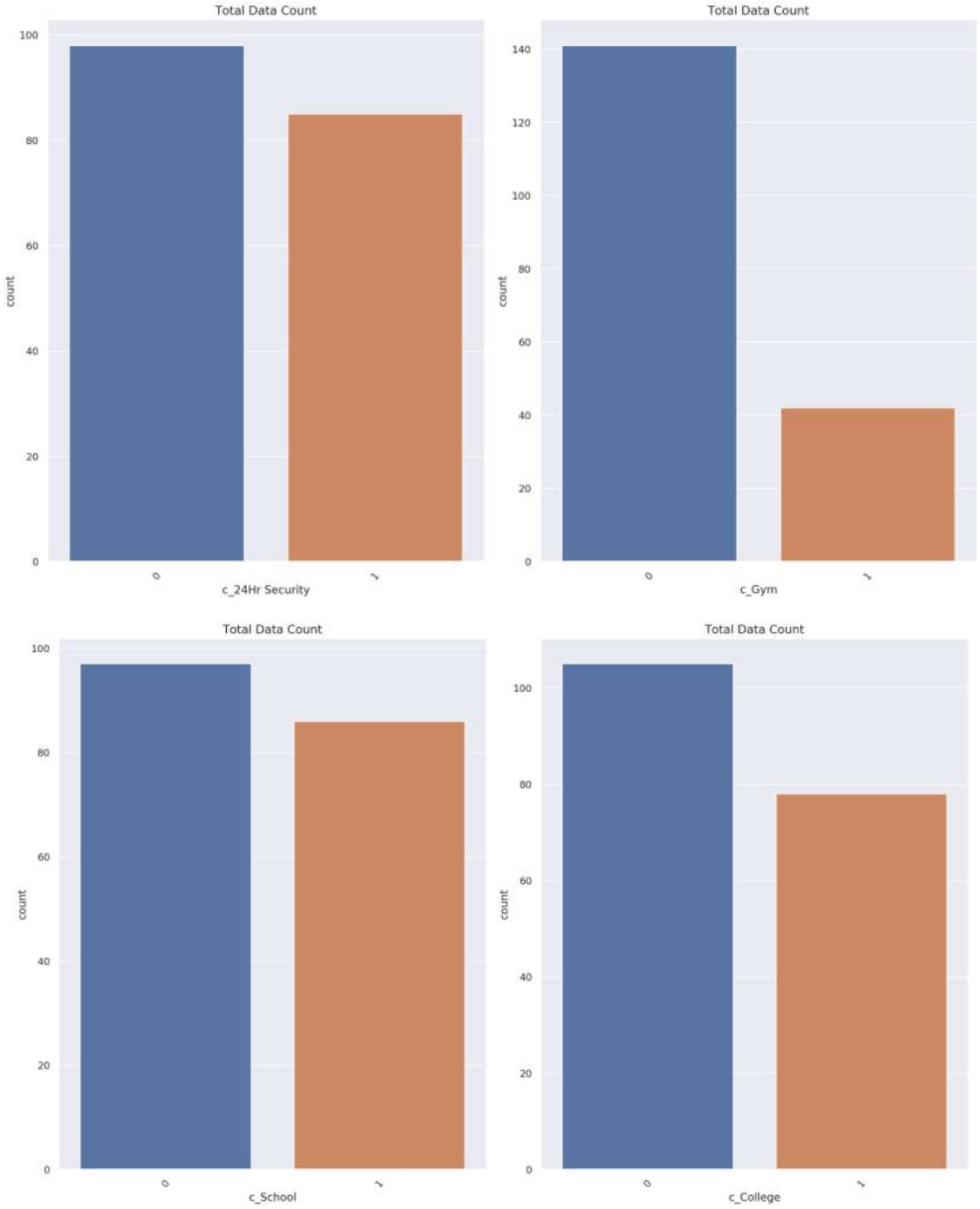

Total Data Count
175
150
125
100
75
50
25
0
Building
Chawl
Bungalow
House_Located
Total Data Count
70
60
50
40
30
20
10
0
count
0
1
2
3
4
No of Bedrooms
Total Data Count
120
100
80
60
40
20
0
count
0
1
c_Car Parking
Total Data Count
100
80
60
40
20
0
count
0
1
c_24hr Water Supply

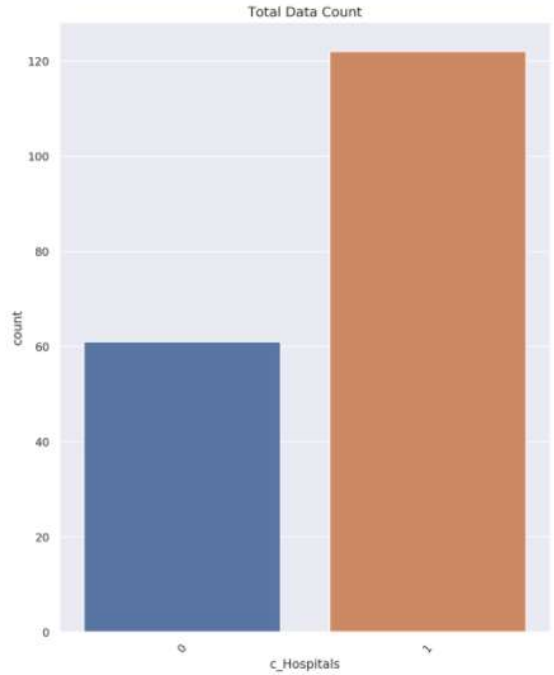

Total Data Count
count
c_Maintenance Staff

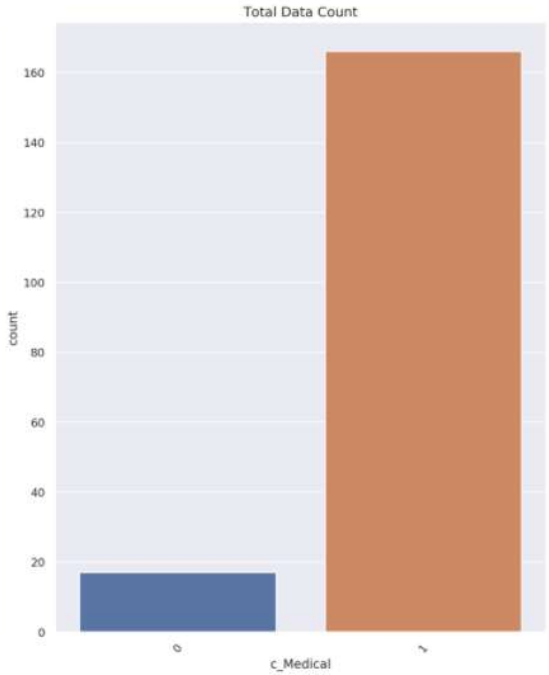

Total Data Count
count
c_Gas Pipeline

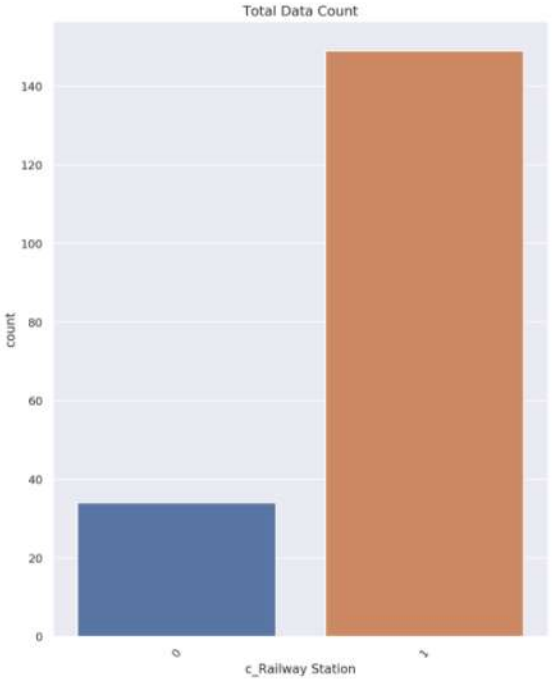

Total Data Count
count
c_Lift

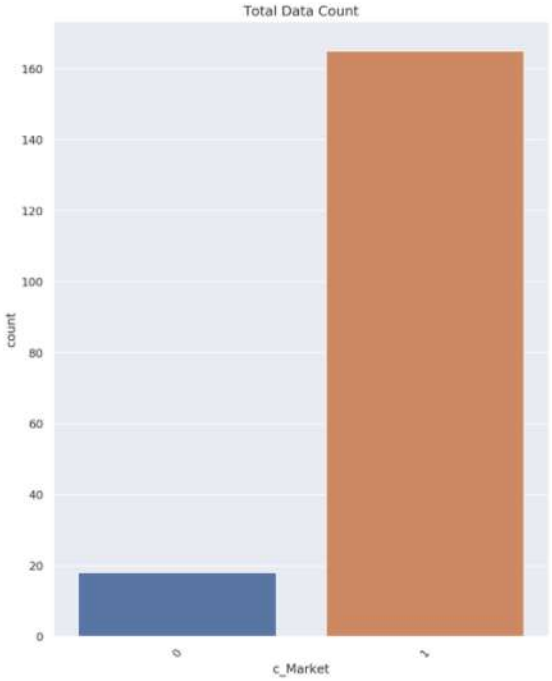

Total Data Count
count
c_Club house


Total Data Count
count
c_24Hr Security
Total Data Count
count
c_Gym
Total Data Count
count
c_School
Total Data Count
count
c_College

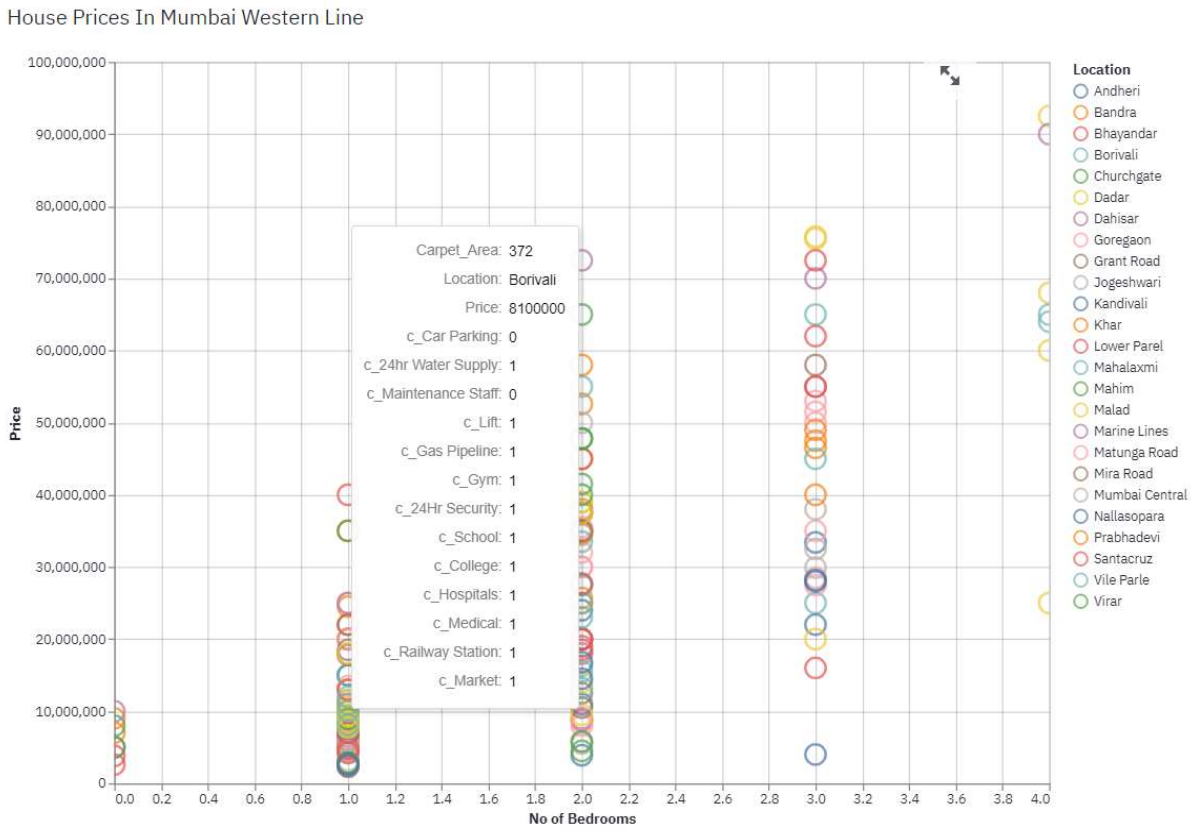

Total Data Count
count
c_Hospitals
0
1


Total Data Count
count
c_Medical
0
1

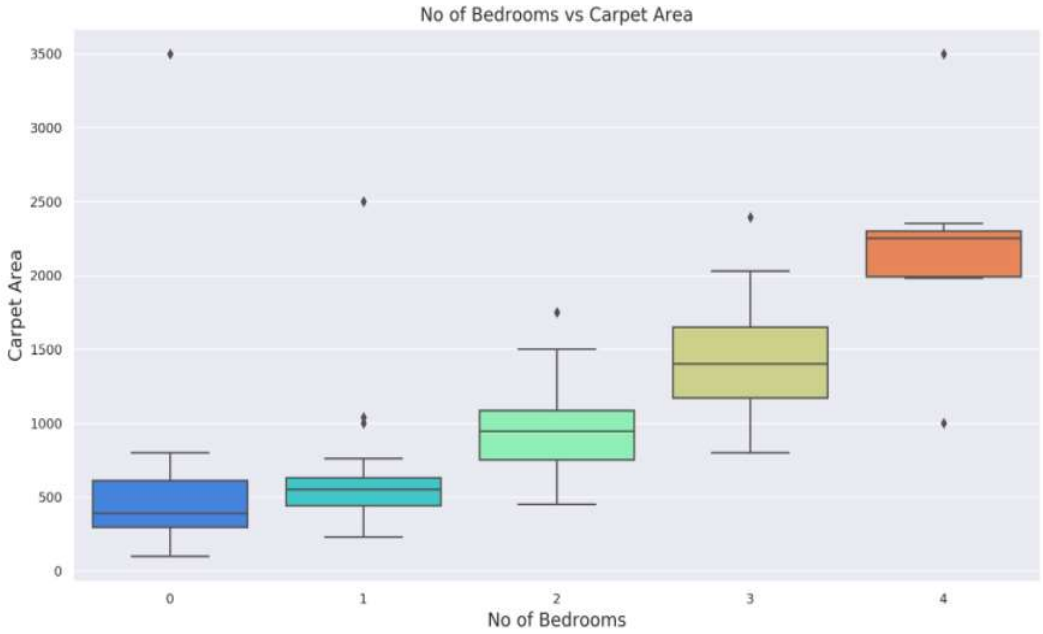

Total Data Count
count
c_Railway Station
0
1


Total Data Count
count
c_Market
0
1

3) A Chart to represent House Prices In Mumbai Western Line



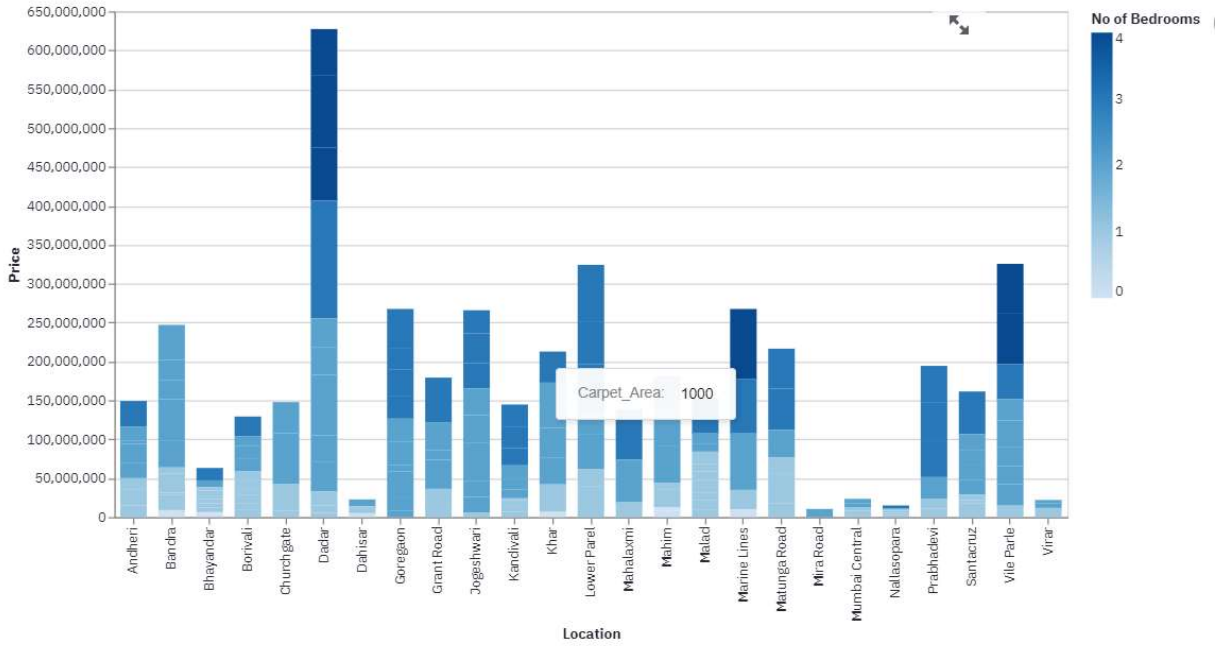
4) BoxPlot to represent Outlier w.r.t No of Bedrooms vs Carpet Area



Conclusion:

From the above graph we can observe that there are some outliers in carpet area with respect to rooms such as in 0 No. of BedRooms i.e 1 RK range of carpet area lies between 100 to 800 therefore house having carpet area 3500 is a outlier, in 1 BK range of carpet area lies between 200 and 760 therefore we have 3 higher outliers with carpet area 1000,1010 and 2500, in 2 BK range of carpet area lies between 480 and 1500 therefore we have 1 higher outliers with carpet area 1750, in 3 BK range of carpet area lies between 900 and 2050 therefore we have 1 higher outliers with carpet area 2400, in 4 BK range of carpet area lies between 1990 and 2350 therefore we have 1 higher outliers with carpet area 3500 and one lower outlier 100.

5) Bar Chart to represent Location wise Prices



1 RK		
High	**Mid**	**Low**
Marine Lines	Mahim	Dahisar
Bandra	Khar	Bhayandar
1 BHK		
High	**Mid**	**Low**
Khar	Churchgate	Bhayandar
Lower Parel	Andheri	Borivali
	Marine Lines	Dahissar
	Grant Road	Jogeshwari
	Vile Parle	Kandivali
	Bandra	Mahalaxmi
	Prabhadevi	Malad
	Dadar	Mumbai Central
	Mahin	Nallasopara
	Virar	
	Santacruz	

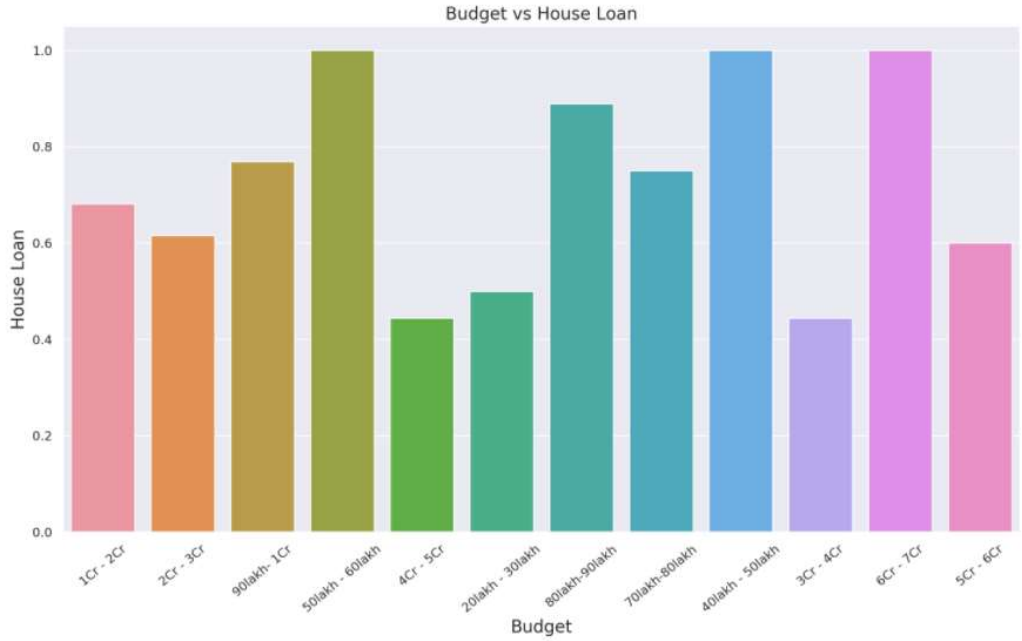
2 BHK		
High	**Mid**	**Low**
Marine Lines	Bandra	Bhayandar
Mahalaxmi	Dadar	Borivali
Churchagte	Grant Road	Dahisar
Khar	Jogeshwari	Kandivali
Mahim	Lower Parel	Malad
	Matunga Road	Miraroad
	Prabhadevi	Santacruz
	Vile Parle	Virar
	Goregaon	

3 BHK		
High	**Mid**	**Low**
Dadar	Andheri	Bhayandar
Lower Parel	Borivali	Malad
Mahalaxmi	Goregaon	Nallasopara
Grant Road	Jogeshwari	
Matunga Road	Kandivali	
Prabhadevi	Khar	
Santacruz		
Vile Parle		

4 BHK		
High	**Mid**	**Low**
Marine Lines	Dadar	Malad
	Vile Parle	

6) Countplot to represent Analysis on Budget

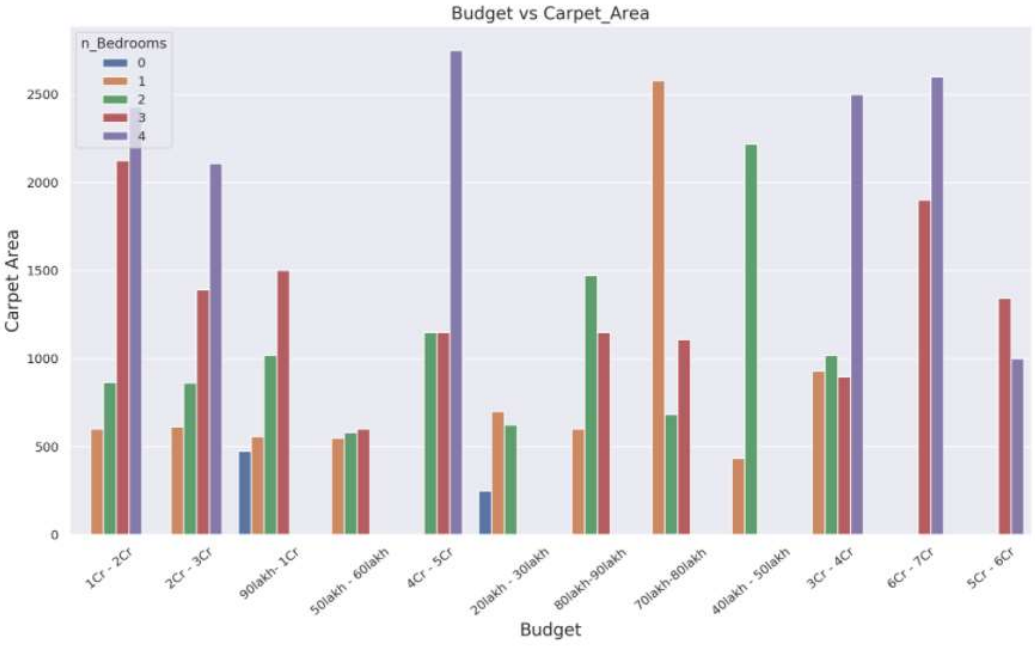
A) Budget Vs House Loan



Conclusion:

From the above graph we can observe that people have budget below 1 cr and above 6cr have the highest changes to take a house loan

B) Budget Vs Carpet Area



Conclusion:

From the above graph we conclude that:

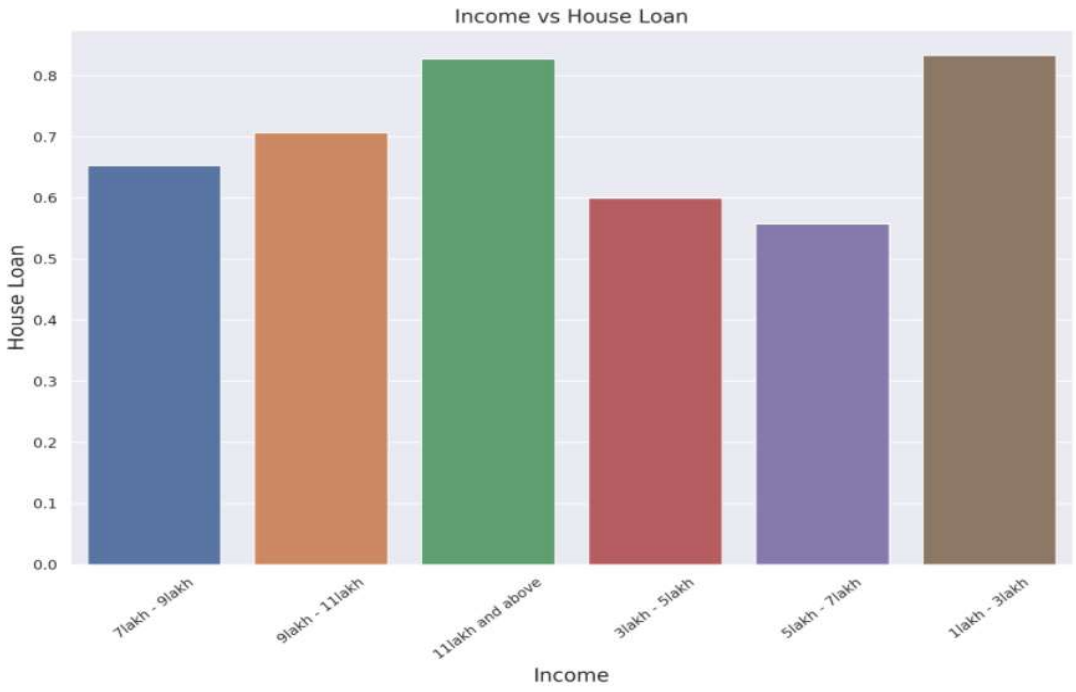
Budget greater than 1Cr target 1,2,3 and 4 bhk with higher carpet area,ecxept budget range between 5Cr-7Cr target 3 and 4bhk with higher carpet area

Budget range between 50lakh and 90Lakh target 1,2 and 3bhk with higher carpet area

Budget range less that 40lakh target 0,1 and 2bhk with medium carpet area

7) Countplot to represent Analysis on Income

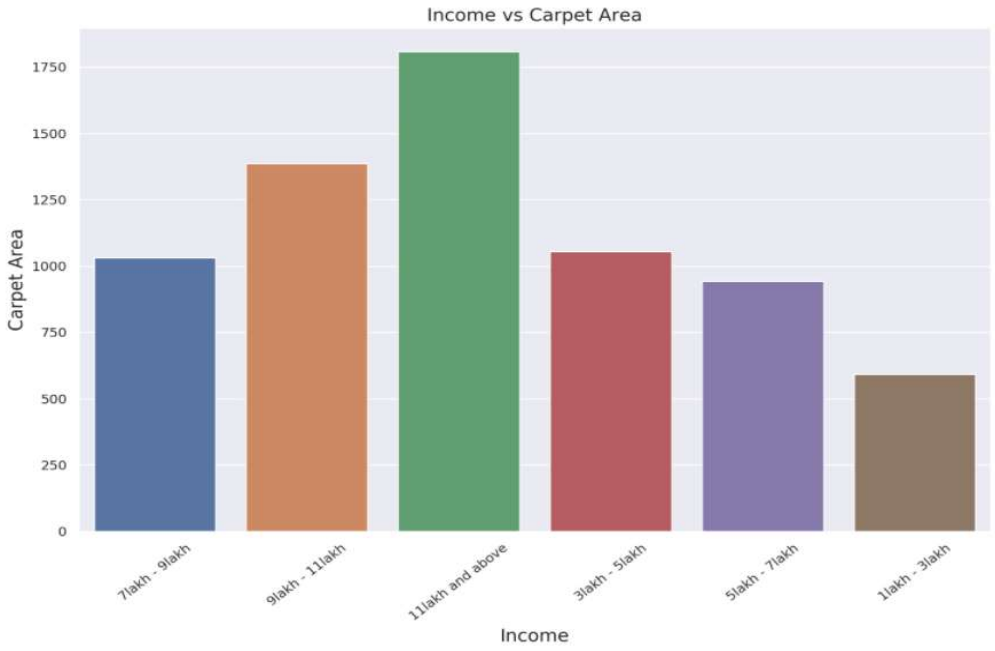
A) Income vs House Loan



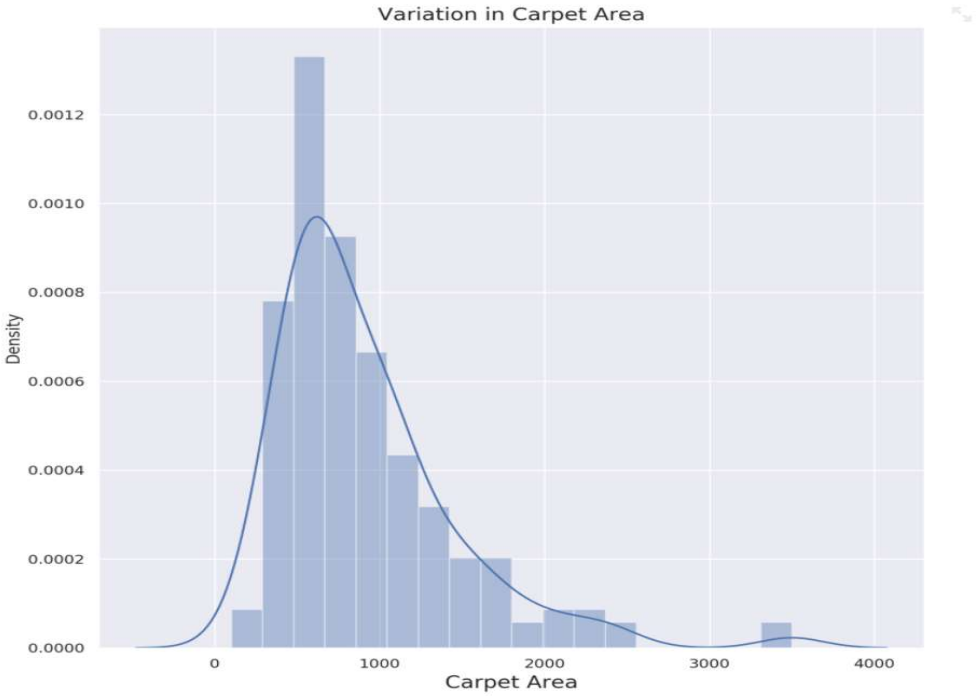
Conclusion:

From the above graph we can observe that people having income between 1 lakh - 3 lakh, 9 lakh - 11 lakh and 11 lakh and above have above 70% changes to take a house loan

B) Income vs Carpet Area



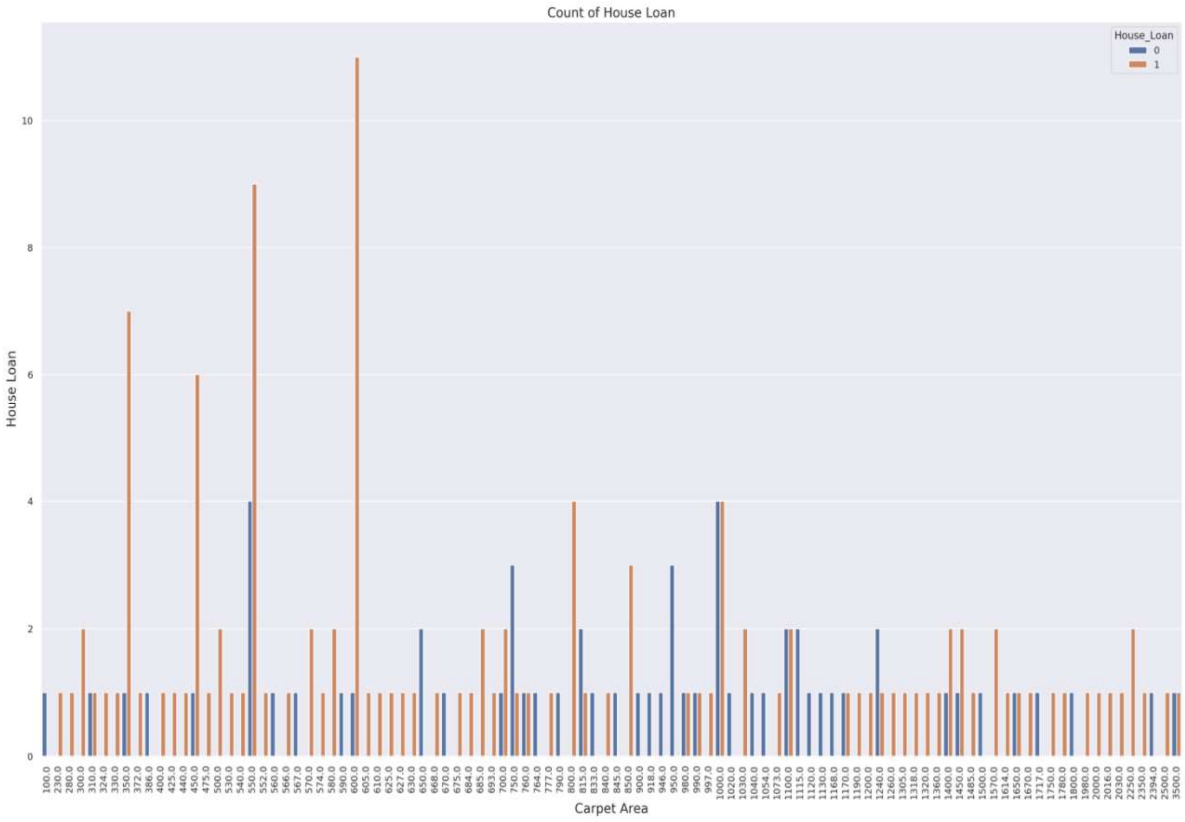
8) Histogram to represent Variation in Carpet Area



Conclusion:

From the above graph we can observe that the dataset contains carpet area majority between 300 to 1000 sq.ft.

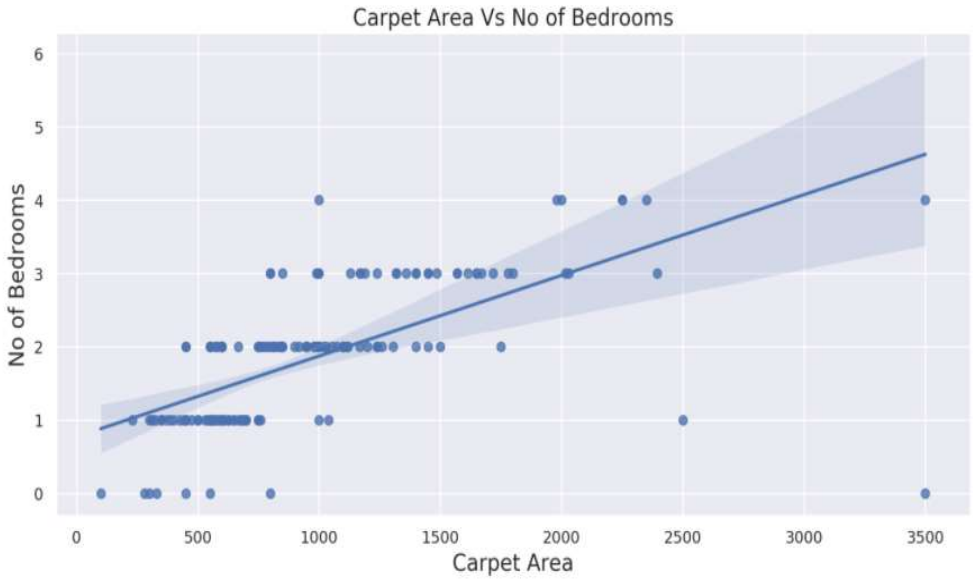
9) Countplot to represent Count of House Loan w.r.t Carpet



Conclusion:

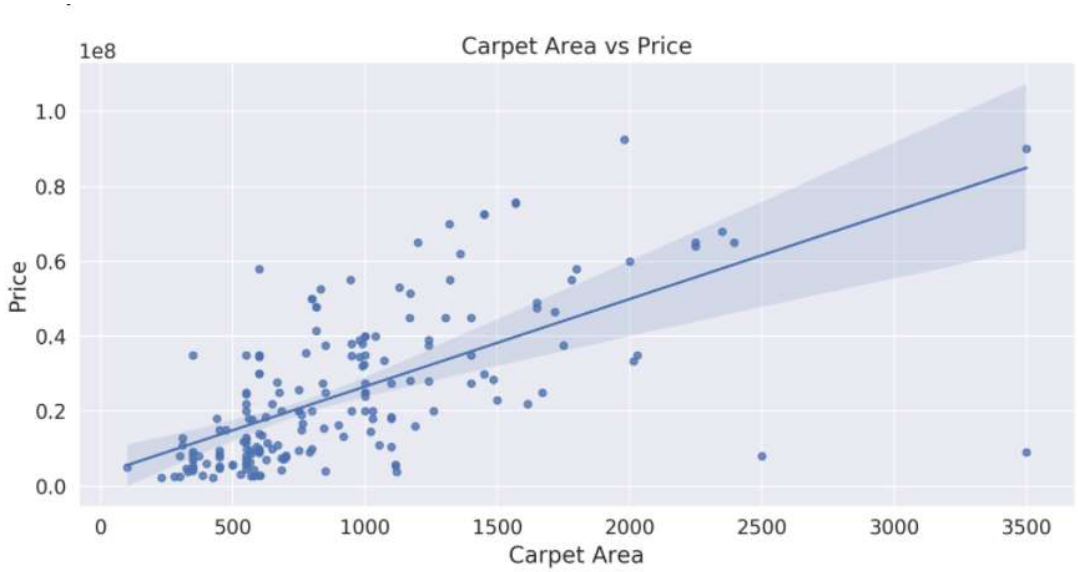
From the above graph we can observe that majority of the people were willing to take the house loan.

10) Graphs to represent Regression Analysis



Conclusion:

From the above graph we can conclude that there is a linear relation between the No. of Bedrooms and Carpet Area as No. of Bedrooms increases Carpet Area also increases but the points do not fit on the regression line.

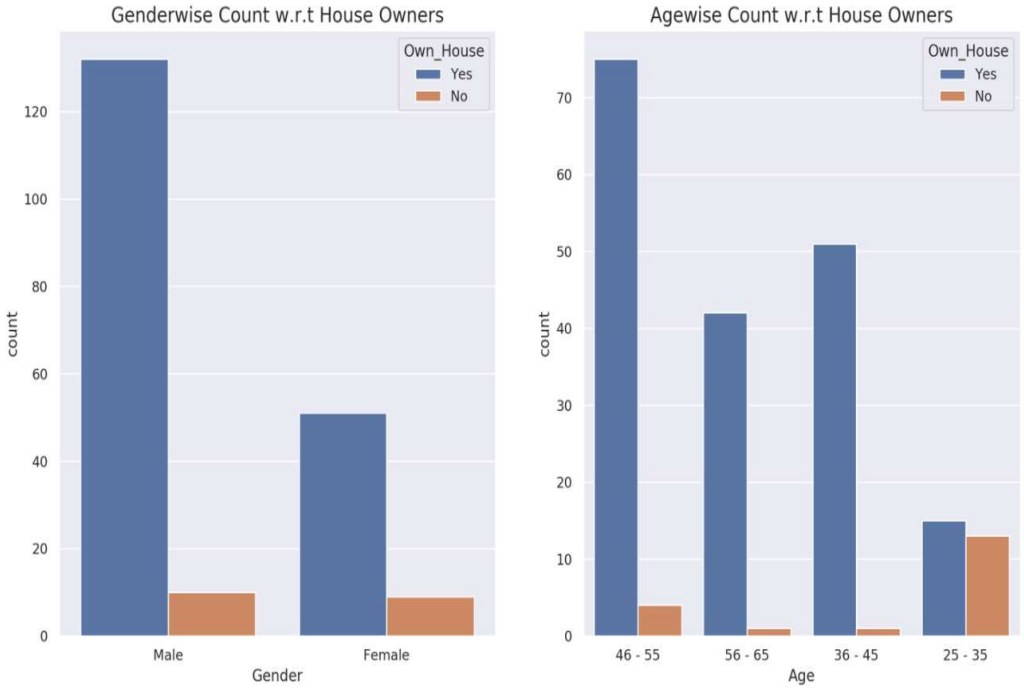


Conclusion:

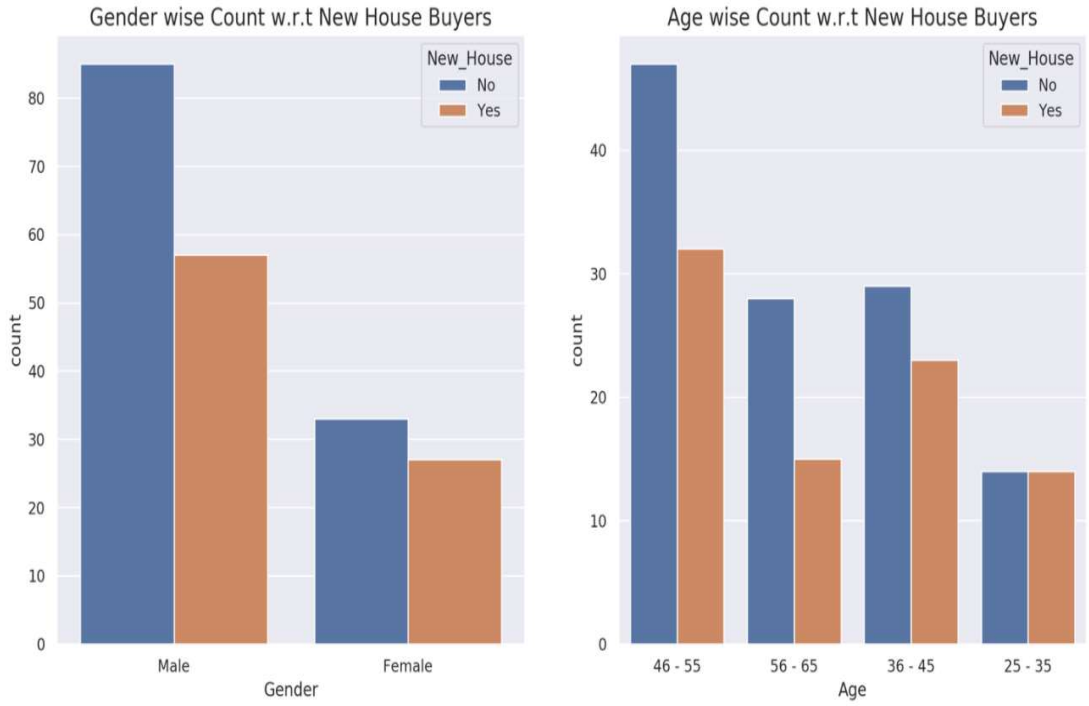
From the above graph we can conclude that there is a linear relation between the House Price and Carpet Area as House Price increases Carpet Area also increases but the points do not fit on the regression line.

11) Countplot to represent House Owner's and House Buyer's strength

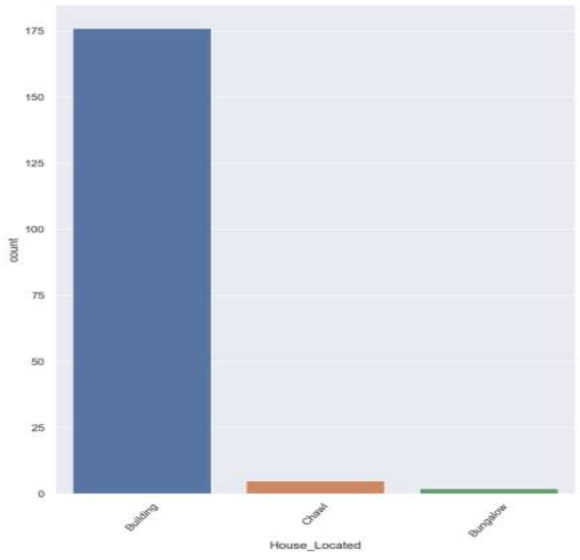
A) Strength of House Owners w.r.t Age and Gender

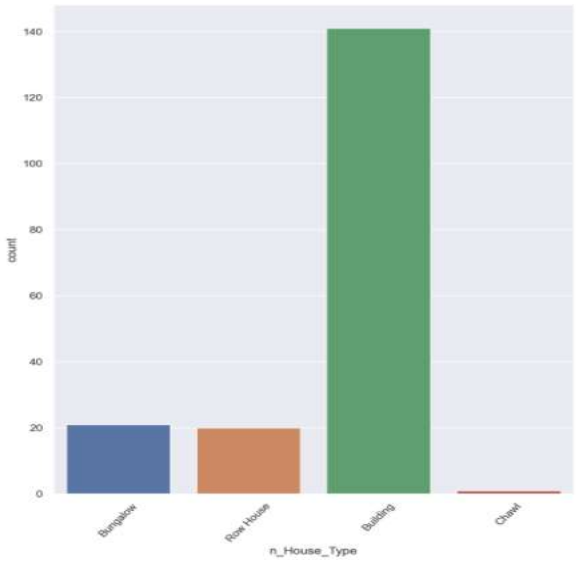


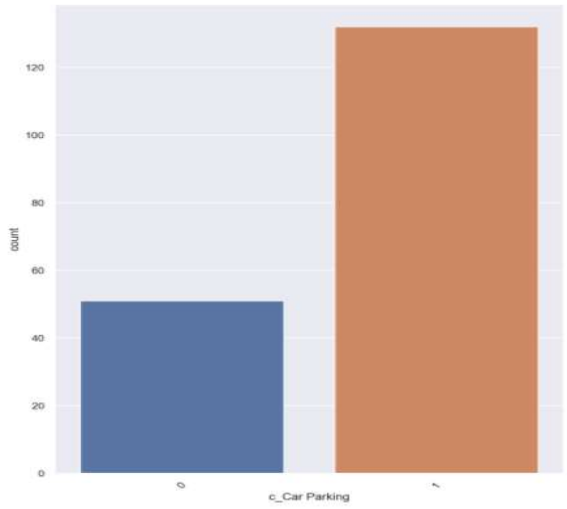
B) Strength of New House Buyers w.r.t Age and Gender

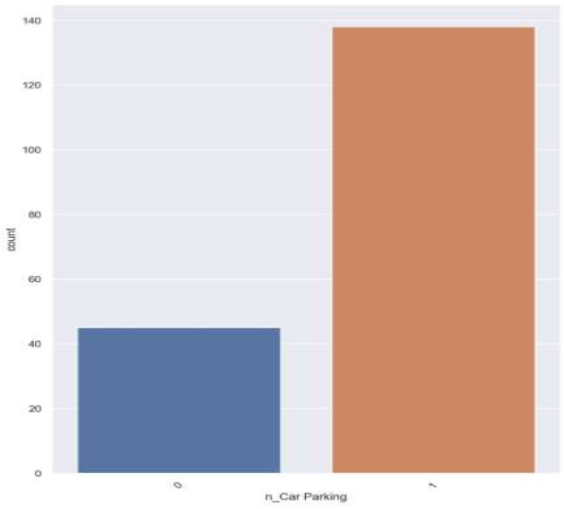


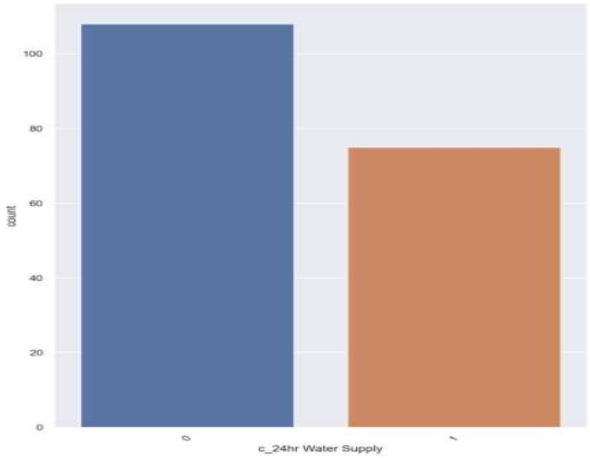
12) Countplot to represent Comparison between Current Features and New Features

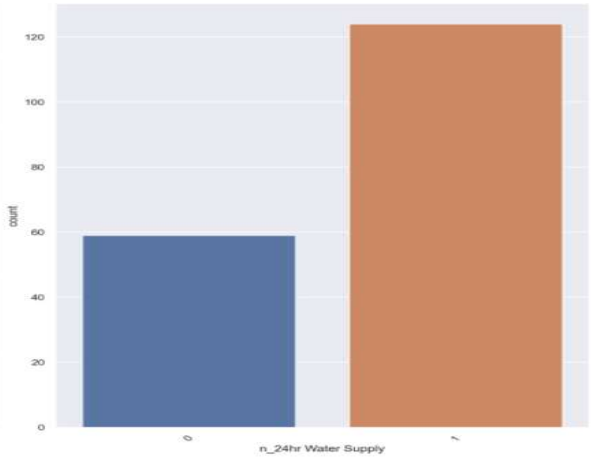


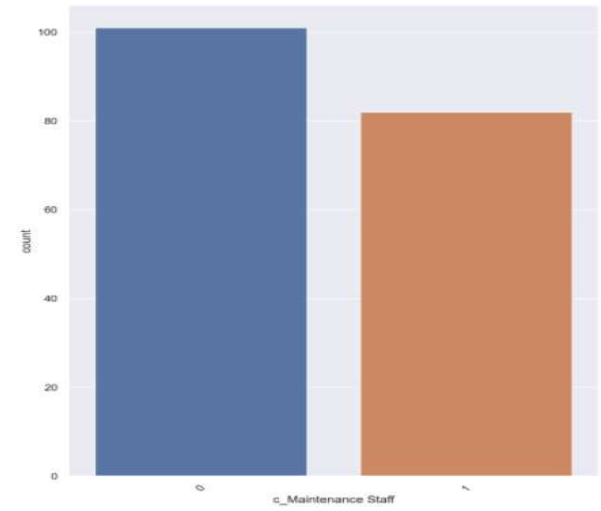


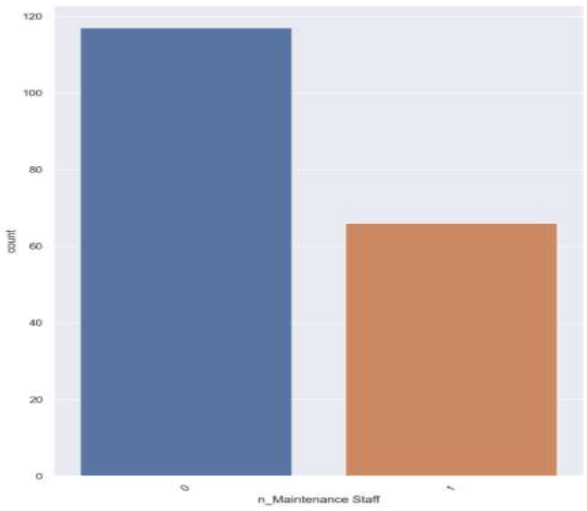


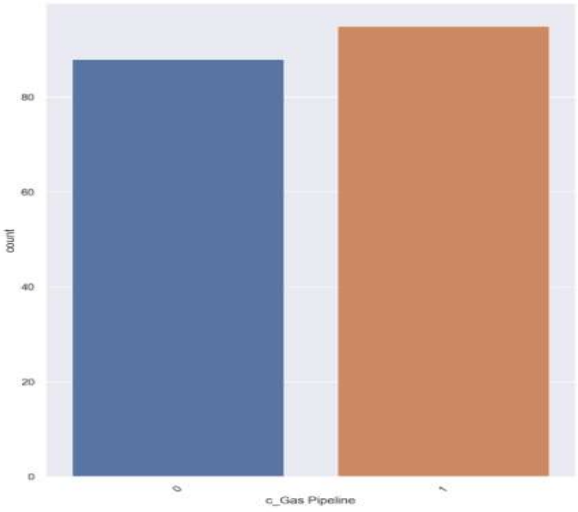


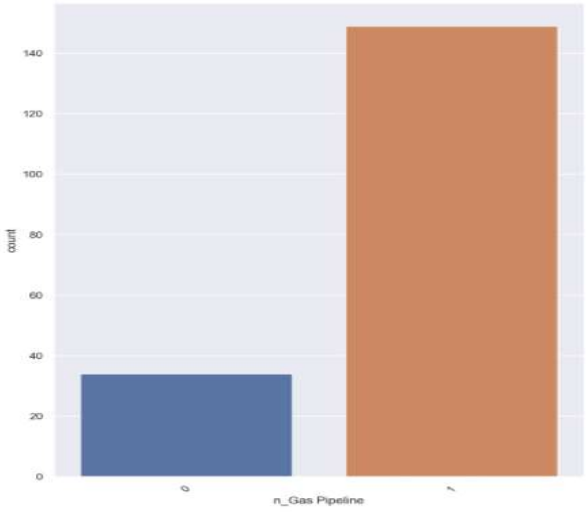


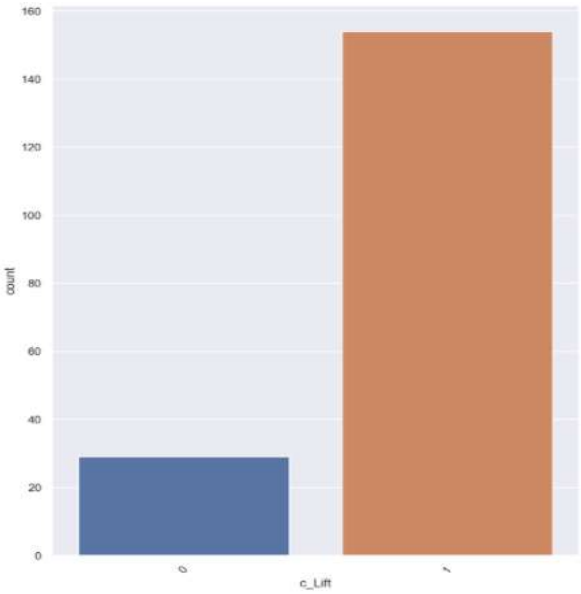


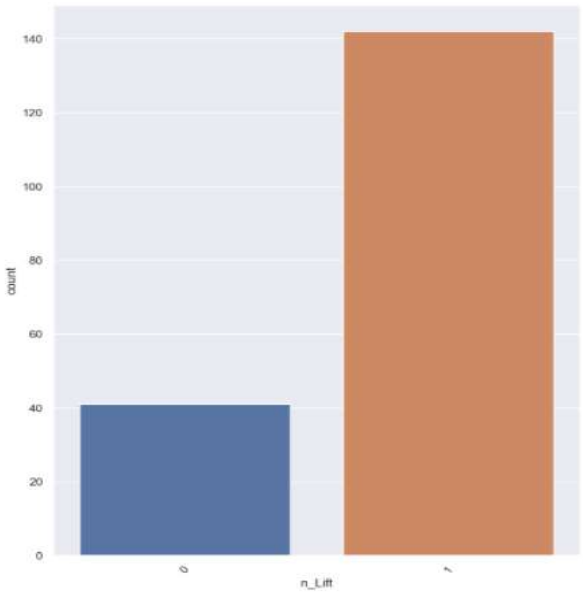

count
c_Maintenance Staff

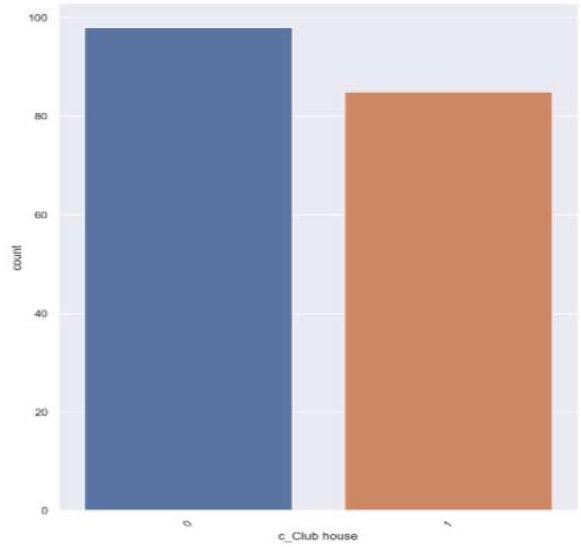

count
n_Maintenance Staff

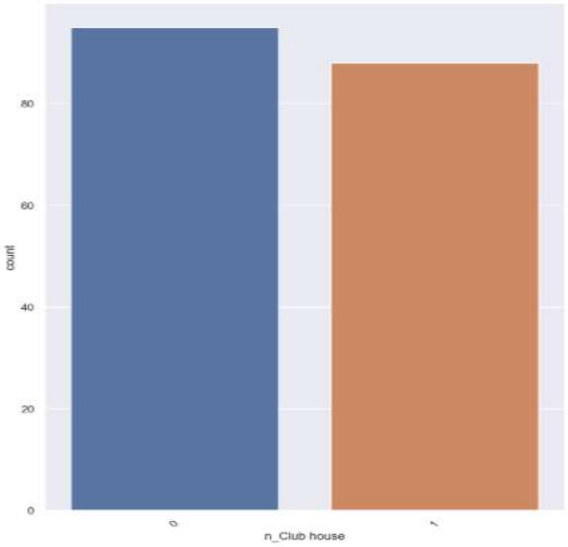

count
c_Gas Pipeline

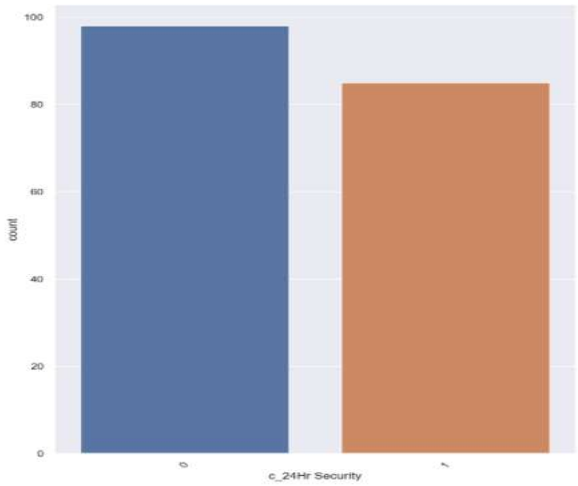

count
n_Gas Pipeline

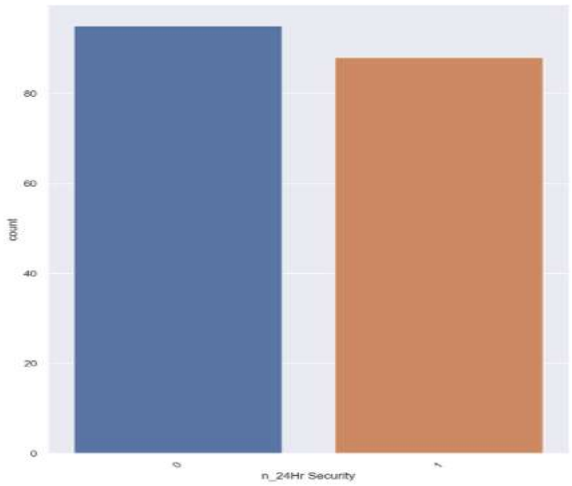

count
c_Lift

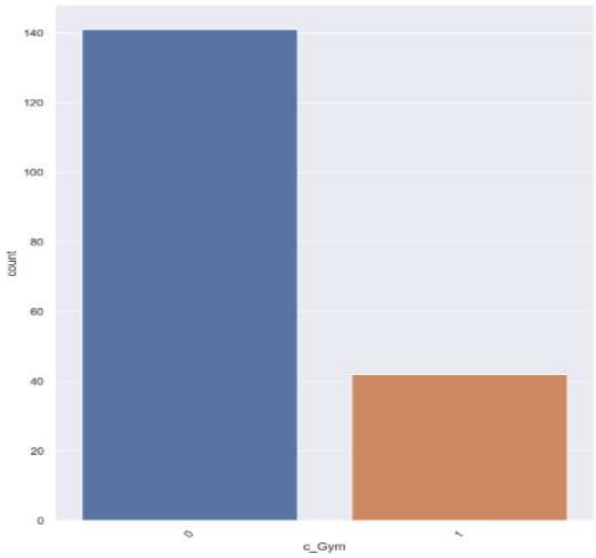

count
n_Lift

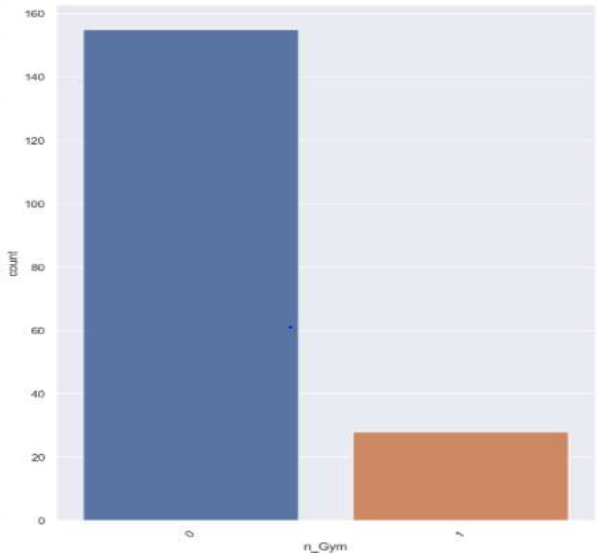

count
c_Club house
0
1

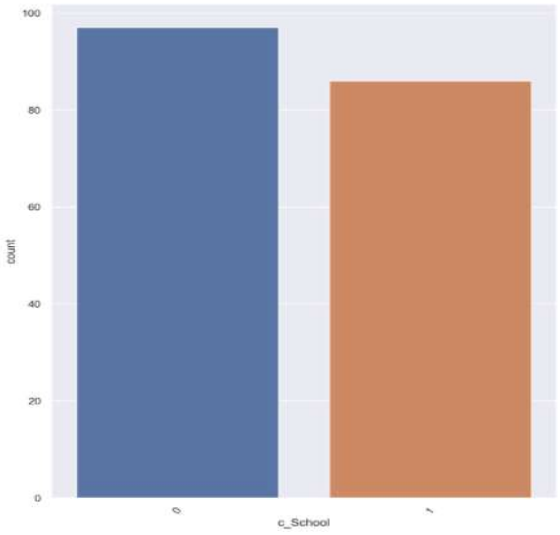

count
n_Club house
0
1

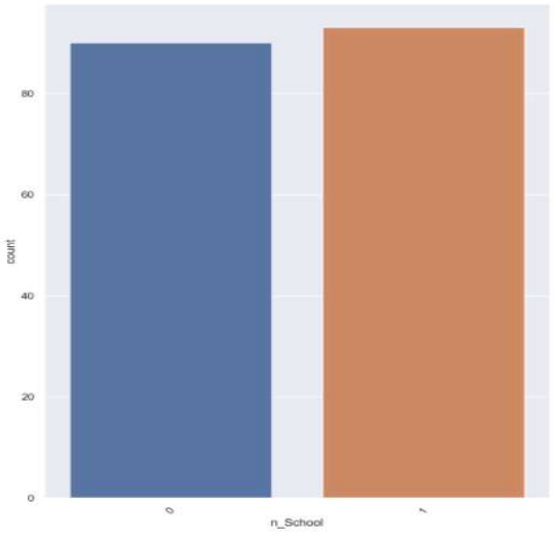

count
c_24Hr Security
0
1

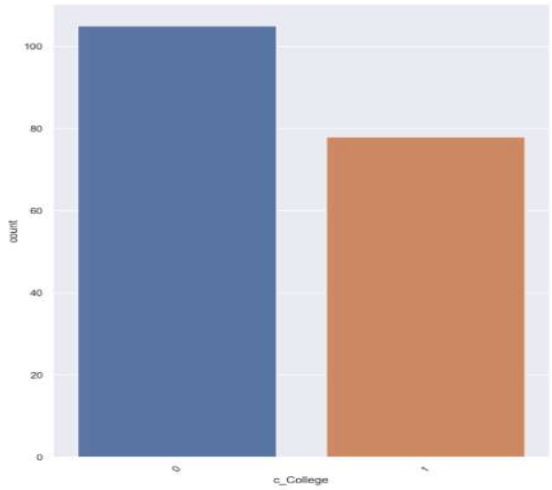

count
n_24Hr Security
0
1

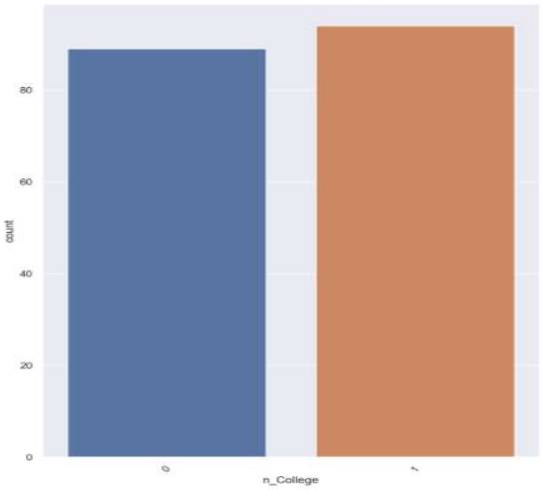

count
c_Gym
0
1

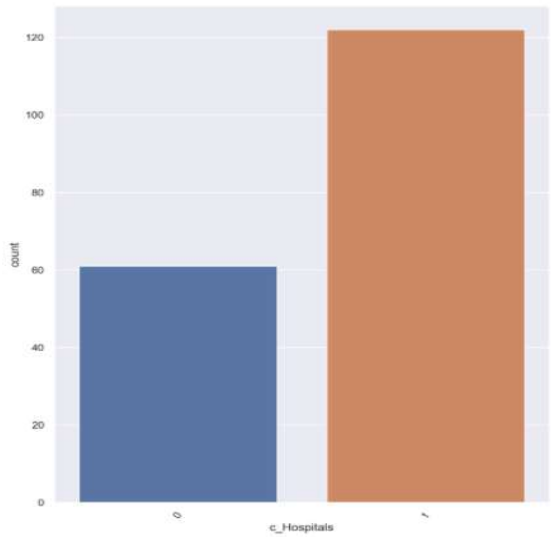

count
n_Gym
0
1

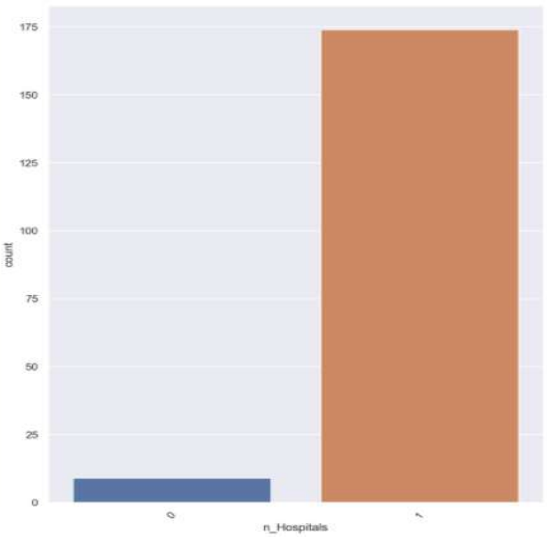

100
80
60
40
20
0
count
0
1
c_School

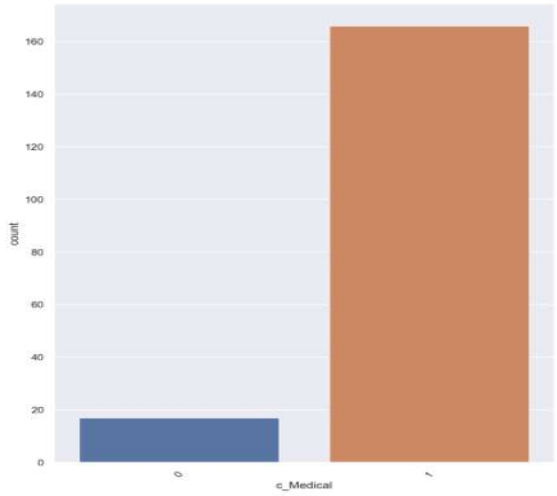

80
60
40
20
0
count
0
1
n_School

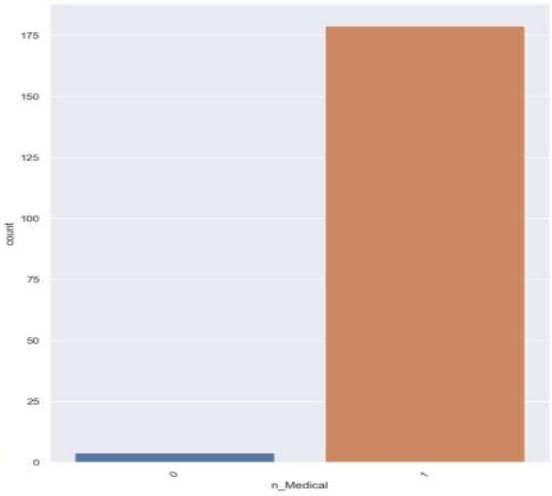

100
80
60
40
20
0
count
0
1
c_College

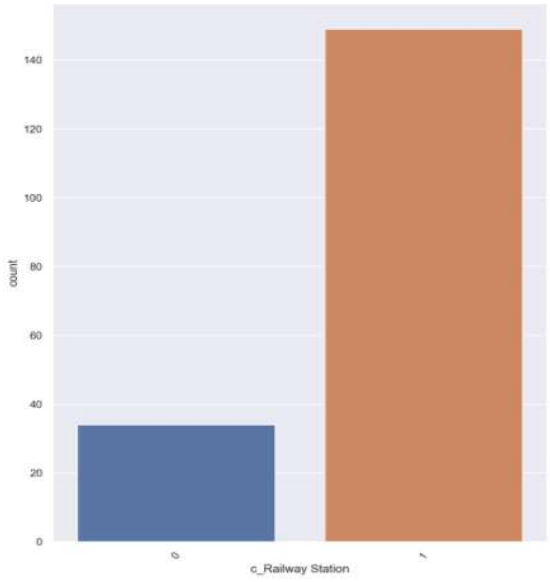

80
60
40
20
0
count
0
1
n_College

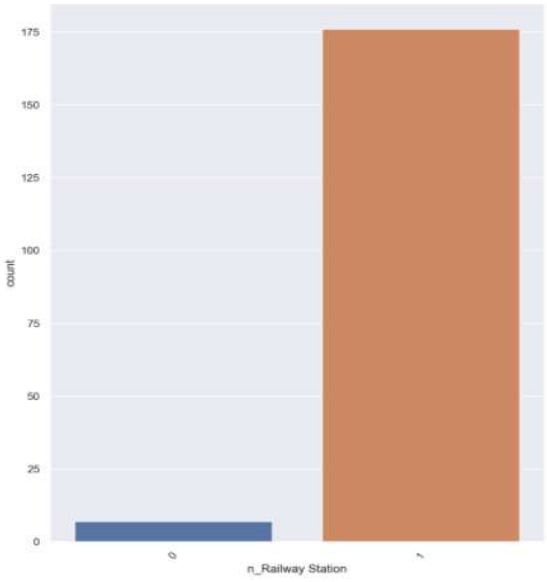

120
100
80
60
40
20
0
count
0
1
c_Hospitals

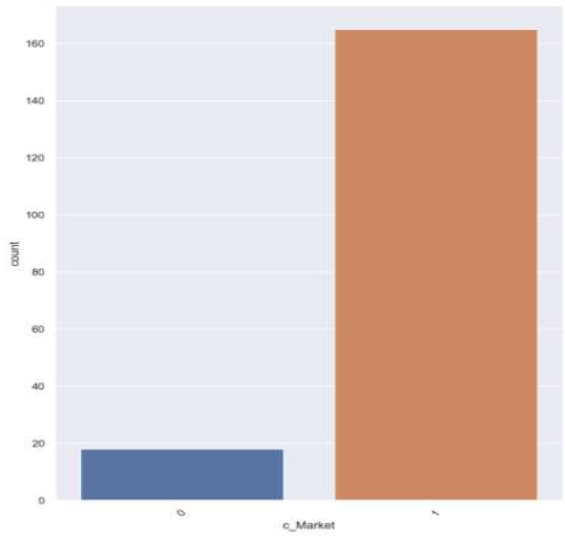

175
150
125
100
75
50
25
0
count
0
1
n_Hospitals

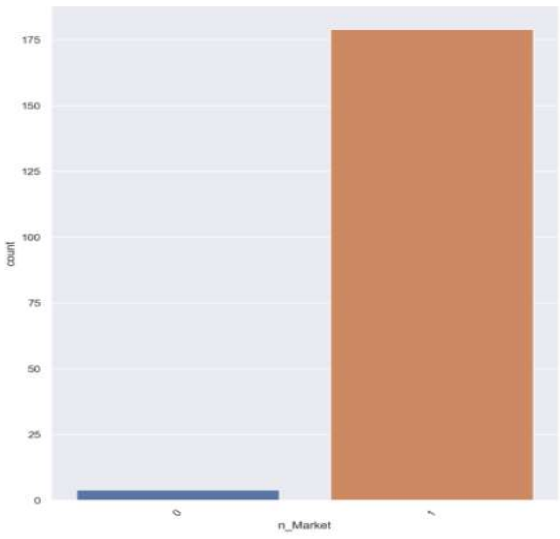

count
c_Medical
0
1


count
n_Medical
0
1


count
c_Railway Station
0
1


count
n_Railway Station
0
1

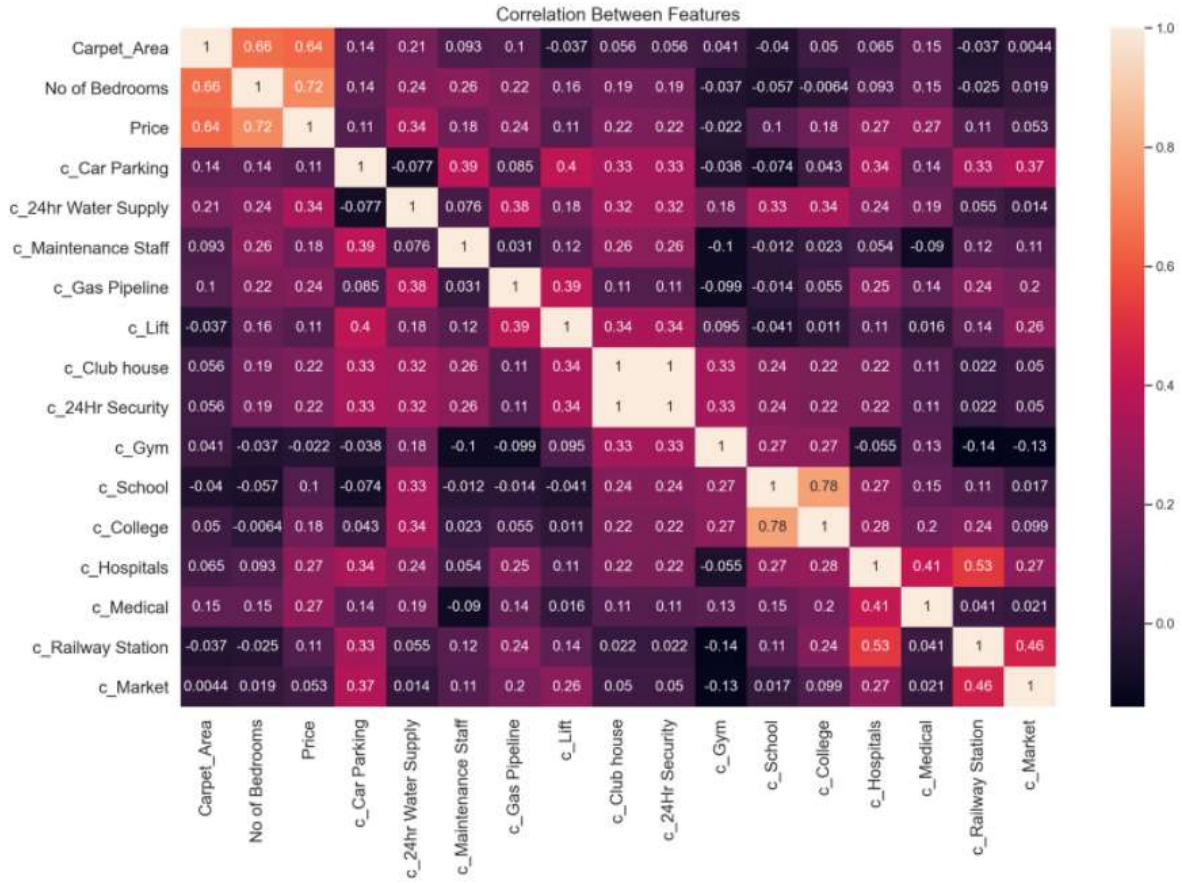

count
c_Market
0
1


count
n_Market
0
1

Conclusion:

From the above graphs we can conclude that except Gym and Club House majority of the people wanted all the rest of the features.

13) Heat Map to represent Highly Co-related Data



Conclusion:

From the graph we conclude that there is strong correlation between Club House and 24Hr Security so we drop Club House since club house is not a important feature to select.

6 ANALYSIS OF THE RESULTS

6.1 Analysis on Regression Models

a) Multiple Linear Regression:

Multiple regression is an extension of simple linear regression. It is used when we want to predict the value of a variable based on the value of two or more other variables. The variable we want to predict is called the dependent variable (or sometimes, the outcome, target or criterion variable)

Formula for calculating multiple linear regression:
$y = \beta 0 + \beta 1x1 + \beta 2x2 + \cdots \beta kxk$

Actual Vs Predicted Value

0 stands for actual value and 1 stands for predicted value

	0	1
0	9300000	219,143.0503
1	3800000	-3,983,229.1488
2	37500000	40,902,566.7639
3	18000000	30,523,569.2445
4	15500000	22,509,664.8228
5	60000000	60,237,008.7161
6	18000000	15,505,348.4107
7	27500000	30,807,973.1164
8	50000000	24,815,916.2528
9	40000000	69,424,534.3476
10	24000000	21,991,048.5486

From the above observation we conclude that there is a huge difference between the actual and predicted value.

R^2 Score: 0.6604280706669454
The model score is 66% that means 66% of the data fit the regression model,since the r-squared score is not close to 1, so the model does not fit best.

b) Support Vector Regression

Supervised Machine Learning Models with associated learning algorithms that analyze data for classification and regression analysis are known as Support Vector Regression. SVR is built based on the concept of Support Vector Machine or SVM.

The problem of regression is to find a function that approximates mapping from an input domain to real numbers on the basis of a training sample.

Actual Vs Predicted Value

0 stands for actual value and 1 stands for predicted value

	0	1
0	9300000	12,156,985.1662
1	3800000	5,396,704.6159
2	37500000	37,468,401.5591
3	18000000	10,842,869.1138
4	15500000	21,776,179.6703
5	60000000	44,543,161.1022
6	18000000	32,433,922.1986
7	27500000	32,434,146.4141
8	50000000	19,873,879.7091
9	40000000	28,379,937.1198
10	24000000	28,378,199.4315

From the above observation we conclude that there is a huge difference between the actual and predicted value.

R^2 Score: 0.518815148933865

The model score is 51% that means only 51% of the data fit the regression mode, since the r-squared score is not close to 1, so the model does not fit best.

c) Decision Tree Regressor

Decision Tree is one of the most commonly used, practical approaches for supervised learning. It can be used to solve both Regression and Classification tasks with the latter being put more into practical application.

It is a tree-structured classifier with three types of nodes. The Root Node is the initial node which represents the entire sample and may get split further into further nodes. The Interior Nodes represent the features of a data set and the branches represent the decision rules. Finally, the Leaf Nodes represent the outcome.

Actual Vs Predicted Value

0 stands for actual value and 1 stands for predicted value

	0	1
0	9300000	9300000
1	3800000	8000000
2	37500000	39000000
3	18000000	25000000
4	15500000	52600000
5	60000000	92500000
6	18000000	33500000
7	27500000	18000000
8	50000000	35000000
9	40000000	16299999
10	24000000	58000000

From the above observation we conclude that there is a huge difference between the actual and predicted value.

R^2 Score: 0.45316105389854877

The model score is 45% that means only 45% of the data fit the regression model, the model score is less than 50% ,since the r-squared score is not close to 1, so the model does not fit best.

d) Random Forest Regressor

Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees.

Actual Vs Predicted Value

0 stands for actual value and 1 stands for predicted value

	0	1
0	9300000	12585000
1	3800000	5951000
2	37500000	38,435,000.0000
3	18000000	13,745,000.0000
4	15500000	24,629,999.5000
5	60000000	83795000
6	18000000	12,660,000.0000
7	27500000	19140000
8	50000000	29200000
9	40000000	43055000
10	24000000	26370000

From the above observation we conclude that there is a less difference between the actual and predicted value.

R^2 Score: 0.755300315635314

The model score is 75% that means 75% of the data fit the regression model, since the r-squared score is close to 1 the model fits best.

Accuracy of the Regression Models:

To measure the accuracy of the regression model we have used R-squared (R^2) score.

R-squared (R^2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model

Regression Model	R^2 Score
Multiple Linear Regression	0.6604280706669454
Support Vector Regression	0.518815148933865
Decision Tree Regressor	0.45316105389854877
Random Forest Regressor	0.755300315635314

By comparing the R^2 of the regressions model we conclude that the Random Forest Regressor have more accuracy in prediction when compared to the others regression model, it has the highest R^2 Score i.e 0.755300315635314

We have used Random Forest Regressor for predicting house prices.

6.2 Analysis on Classification Models

a) Random Forest Classifier

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes become our model's prediction

Classification Report:

A Classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report as shown below.

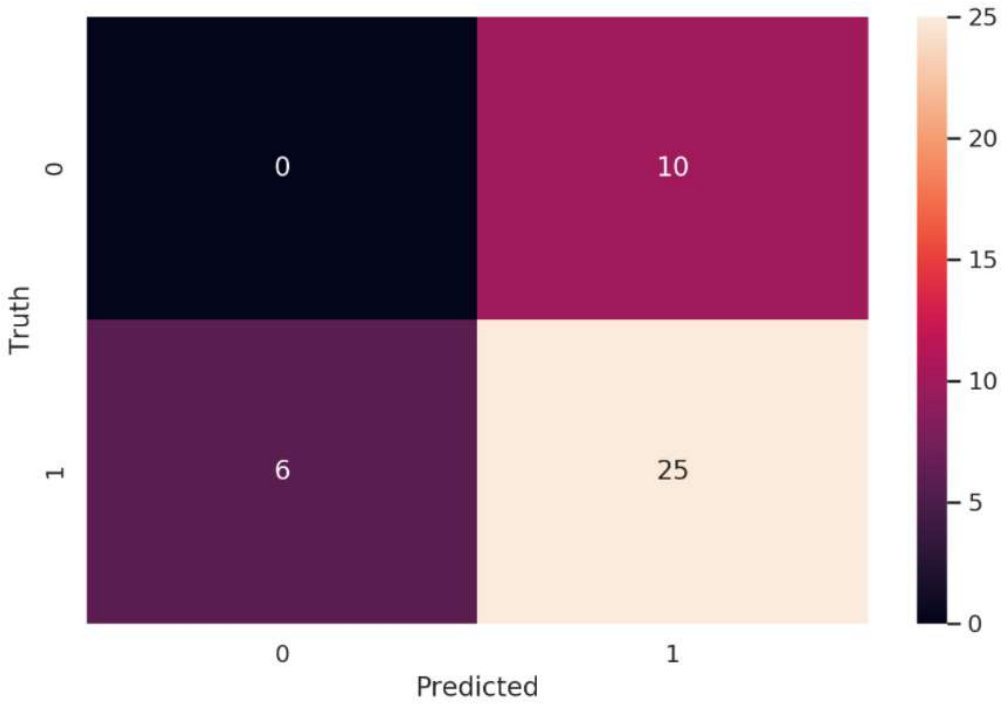
precision recall f1-score support

```
          0       0.00      0.00      0.00        10
          1       0.71      0.81      0.76        31

   accuracy                           0.61        41
```

The report shows the main classification metrics precision, recall and f1-score
Precision: When it predicts yes, how often it predict yes
Recall: Number of positive returned by the model
F Score: This the weighted average of the true positive rate(recall and precision)
Accuracy: Overall model score

HeatMap to represent confusion matrix



Conclusion:

From the above analysis result we conclude that since the overall accuracy score of the model is 0.61 which is not that close to 1 so the model does not fit best

2) K-Nearest Neighbors Classifier

K-Nearest Neighbors (KNN) is one of the simplest algorithms used in Machine Learning for regression and classification problem. KNN algorithms use data and classify new data points based on similarity measures (e.g distance function). Classification is done by a majority vote to its neighbors

Classification Report:

A Classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report as shown below.

precision recall f1-score support

0	0.00	0.00	0.00	10
1	0.74	0.94	0.83	31
accuracy			0.71	41

The report shows the main classification metrics precision, recall and f1-score

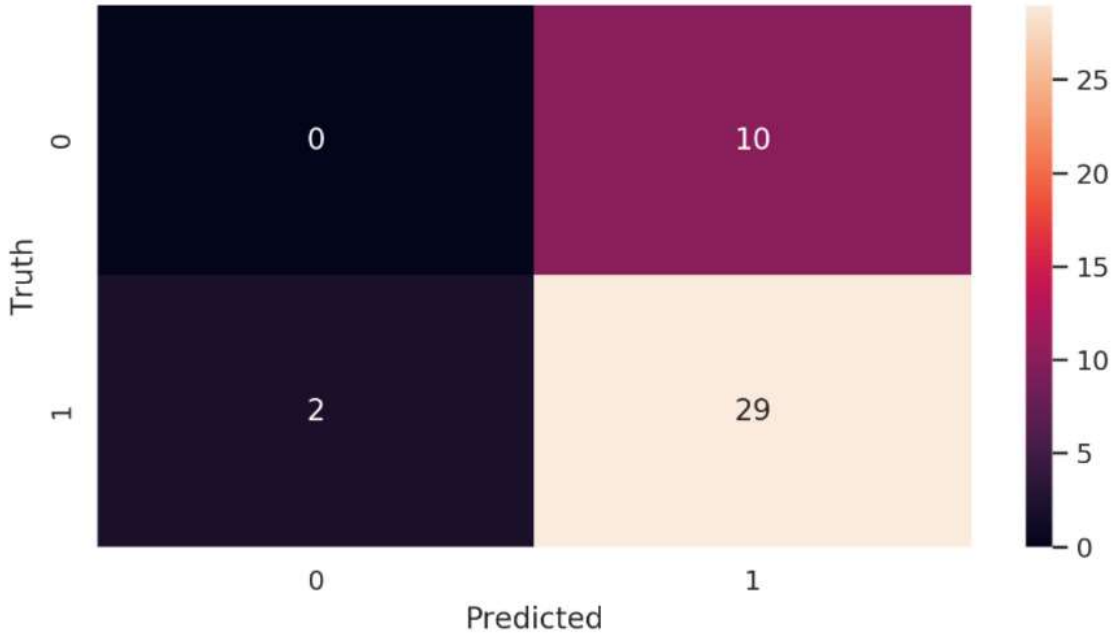
Precision: When it predicts yes, how often it predict yes

Recall: Number of positive returned by the model

F Score: This the weighted average of the true positive rate(recall and precision)

Accuracy: Overall model score

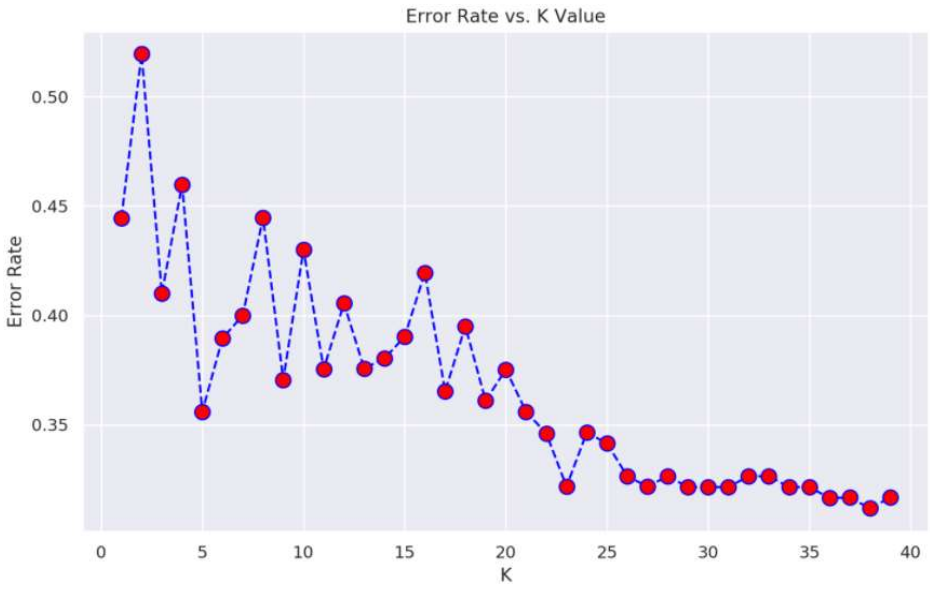
Heat Map to represent confusion matrix



Conclusion:

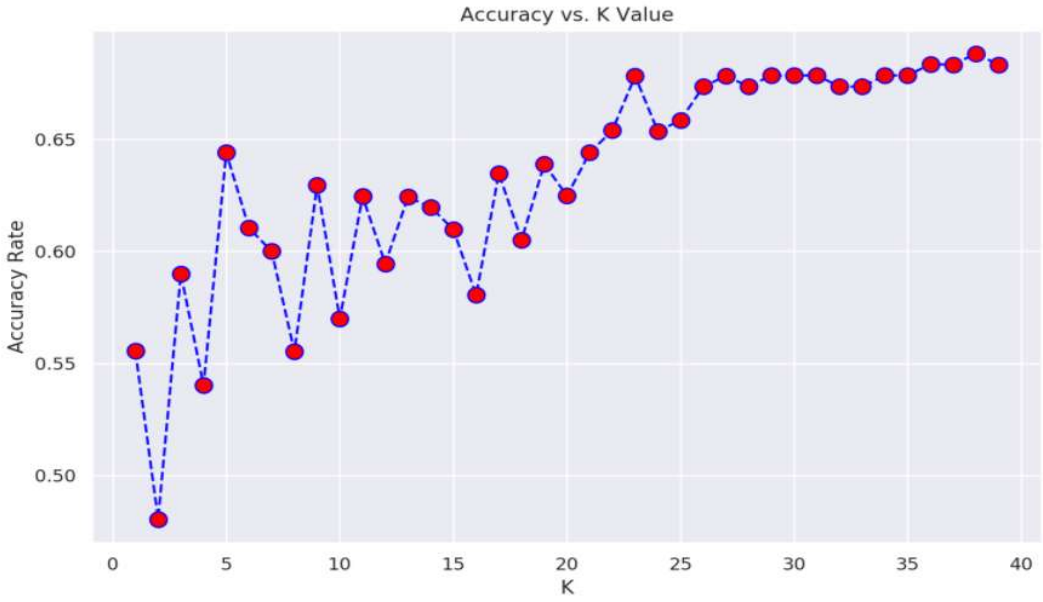
From the above analysis result we conclude that since the overall accuracy score of the model is 0.71 which is close to 1 so the model fits best

Choosing the K value



From the above graph we conclude that since the error rate does not fluctuate after k=23 so we choose k value as K=23

Checking the accuracy of K = 23



From the graph we conclude that since the accuracy of K value at k=23 neither increases nor decreases the k value chosen is accurate4

Accuracy of Classification Model

Classification Model	Accuracy Score
Random Forest Classifier	0.5121951219512195
K-Nearest Neighbor Classifier	0.7073170731707317

By comparing the accuracy of the classification model we conclude that the K-Nearest Neighbor Classifier have more accuracy in prediction when compared to the others classification model, it has the highest Accuracy Score i.e 0.7073170731707317

We have used K-Nearest Neighbor Classifier for predicting whether the user will take a house loan or not take a house loan to buy a house.

7 CONCLUSION

The main of this project is to determine the prediction of house prices which have successfully done using different machine learning algorithms like Multiple Linear Regression, Support Vector Regressor, Decision Tree Regressor and Random Forest Regressor, so after the analysis it was clear that Random Forest Regressor have more accuracy in prediction as compared to other regression models

Maximum house owners are between the age group of 46-65 and male house owners are more as compared to female

Maximum new house buyers are between the age group of 36-55 and minimum new house buyers are between the age group of 25-35 and 56-45 male buyers are more as compared to female

The house price is highly dependent on the following features (No of Bed Rooms, Carpet Area, Location, 24Hr Water Supply, Gas Pipeline, Lift and medical)

Most of the buyers take housing loan to buy a new house

Since we considered western line house prices so people staying between Borivali to Bandra have moderate house price as compared to people staying in between Mahim to Churchgate have very high prices and people staying in between Dahisar to Virar have less house price

8 FUTURE ENHANCEMENT

We had less time to collect data so we managed to collect only 200 entries of Mumbai considering only western line but in future a large dataset can be collected through survey or by some any other platform and all over Mumbai data can be taken into consideration

If there is a large dataset then the regression model will also be more accurate and there will be not much difference between the actual and predicted prices

If the dataset is large we can perform more operations on the dataset like calculating the k-fold cross value score finding the hyper parameter through grid search performing model selection technique for example XG Boost, and checking the model performance

Forecasting can be done that is in future, what will be house price, will the price trend be same or change, If change, so through what percentage will the rate increase in the coming year, and what other factors can affect the house rate

Some other prediction can also be taken into consideration like based on the house price how much saving a buyer must do before taking his future dream house, loan amount should be also brought into consideration like what will be the monthly EMI and the loan clearance period

9 Survey Questions

Q1) What is your age?

25-35 36-45 46-55 56-65

Q2) What is your gender?

Male Female Other

Q3) In which area do you stay in western line?
Note: Drop of area in western line

Q4) Your yearly income in INR
1lakh-3lakh 3lakh-5lakh 5lakh-7lakh 7lakh-9lakh
9lakh-11lakh 11lakh and above

Q5) Do you own a house ?
Yes No

Q6) Where is your house located ?
Chawl Building Bungalow Row House

Q7) What is the carpet area of your current house in square feet ?

Q8) How many rooms do you have
1Rk 1Bhk 2Bhk 3Bhk 4Bhk

Q9) What is the current price of your house in INR?

Q10) What facilities are available in your current society?

Car Parking	24hr Water Supply	Maintenance Staff
Gas Pipeline	Lift	Club house
24Hr Security	Gym	None of the above

Q11) What all facilities are currently available in your nearby area?

School	College	Hospitals
Medical	Railway Station	Market

Q12) Are you planning to buy a new house in Mumbai(Western Line)
Yes No

Q13) Would you take a housing loan if you had to buy a house?

Yes No

Q14) In which area you would like to buy a new house in Mumbai(Western Line) if you had to?

Note: Dropdown list of area in western line

Q15) What type of house you would look for if you had to buy a house?

Flat Row House Bungalow Chawl System

Q16) What would be your budget if you had to buy a house?

20lakh -30lakh	30lakh-40lakh	40lakh-50lakh	50lakh-60lakh
70lak-80lakh	80lakh-1Cr	1Cr-3Cr	3Cr-5Cr
5Cr-7Cr	7Cr-9Cr	More than 9Cr	

Q17) In which of western line you would like to buy a new house

Note: Dropdown list

Q18) How much carpet area would you want(square feet) if you had to buy a house?

Q19) How many rooms do you want if you had to buy a house?

1Rk 1Bhk 2Bhk 3Bhk 4Bhk

Q20) If you had to buy a house what would you buy?

New House Resale House

Q21) What facilities you would like a society to feature ?

Car Parking	24hr Water Supply	Maintenance Staff
Gas Pipeline	Lift	Club house
24Hr Security	Gym	

Q22) What all facilities you want nearby your area ?

School College Hospitals Medical

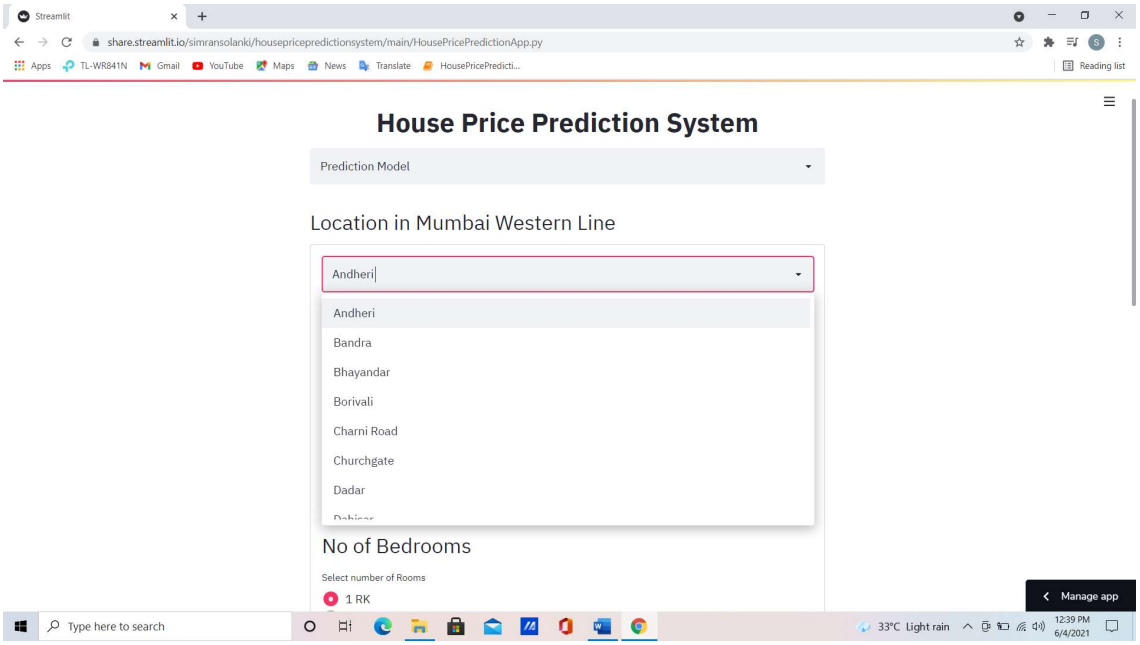
Railway Station Market

10 SCREEN LAYOUTS

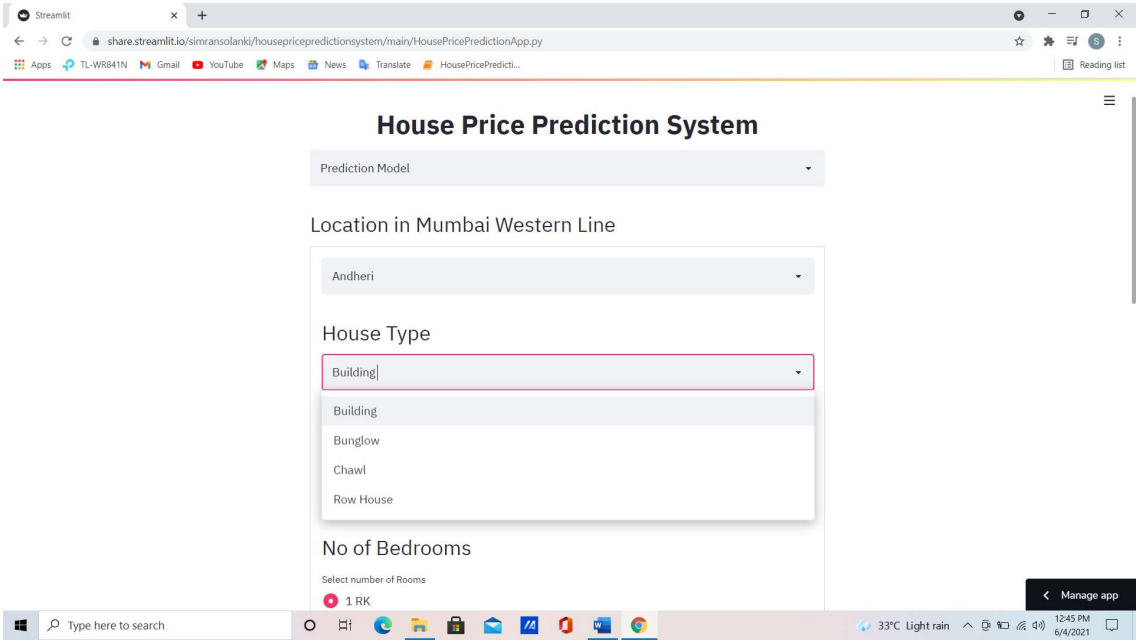
A) House Price Prediction Model

Random Forest Regression Model was used for predicting the House Prices

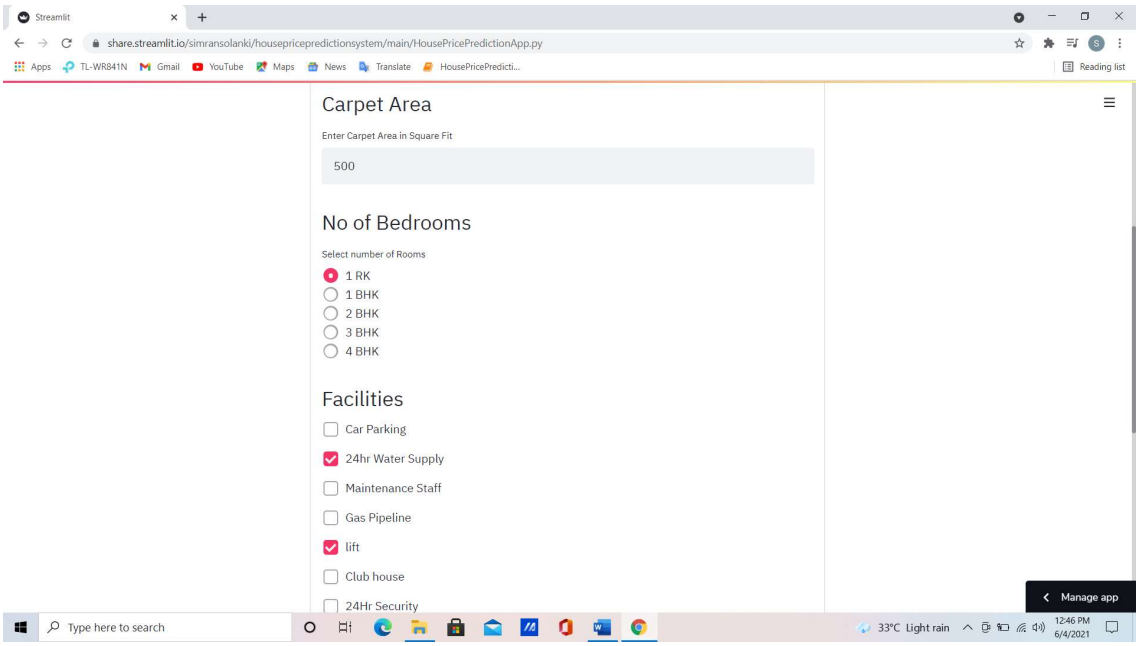
1) The user selects the location from the dropdown list



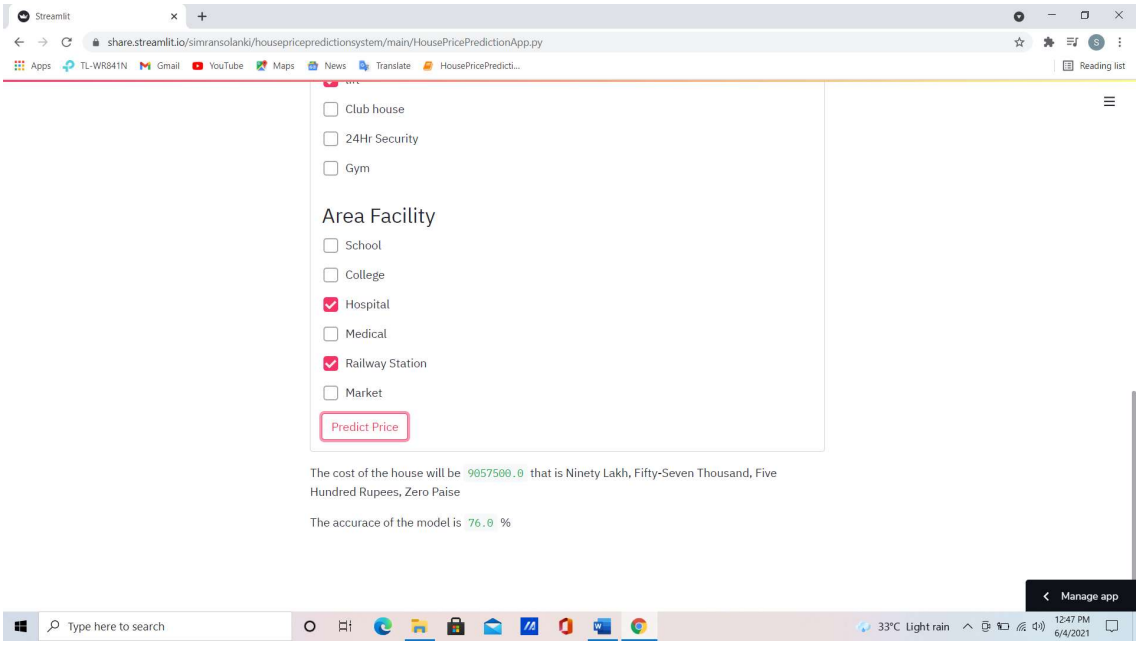
2) The user selects house type from the dropdown list



3) Enter the carpet area
Select No of Bedrooms
Select the facilities

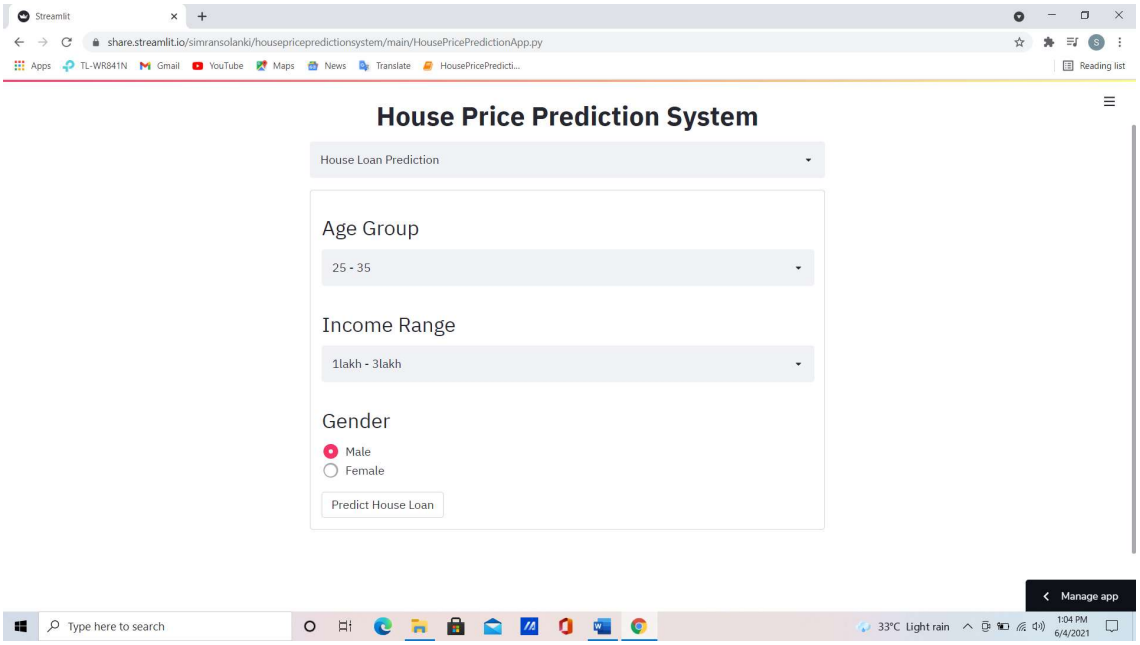


4) Select area facility
Click on the Predict Price Button (The predicted house price will be displayed)

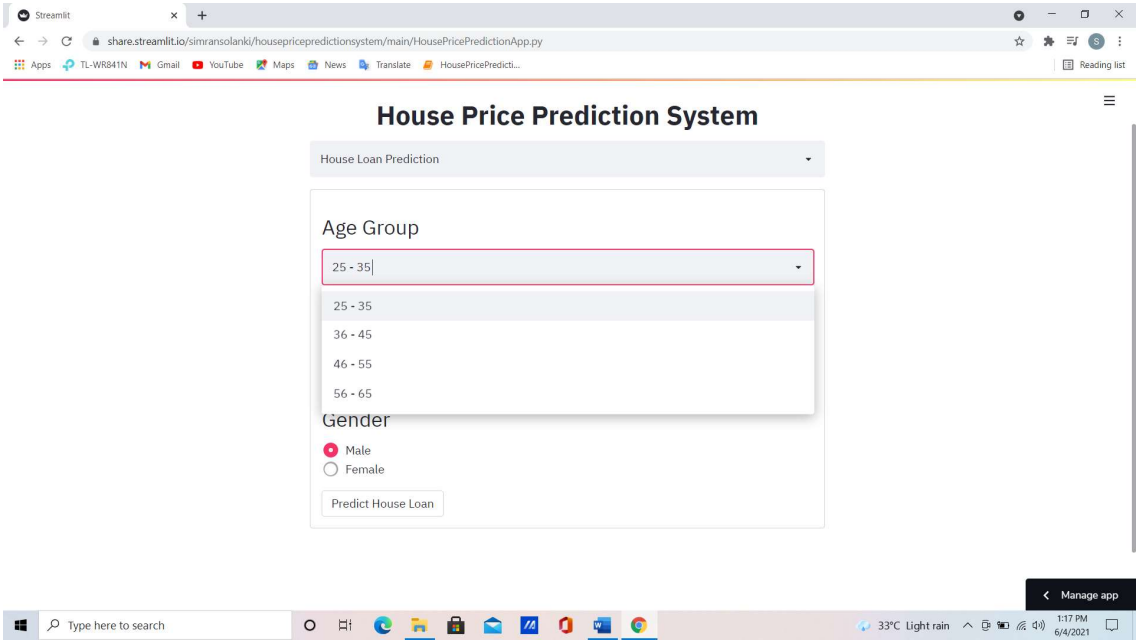


B) House Loan Prediction Model

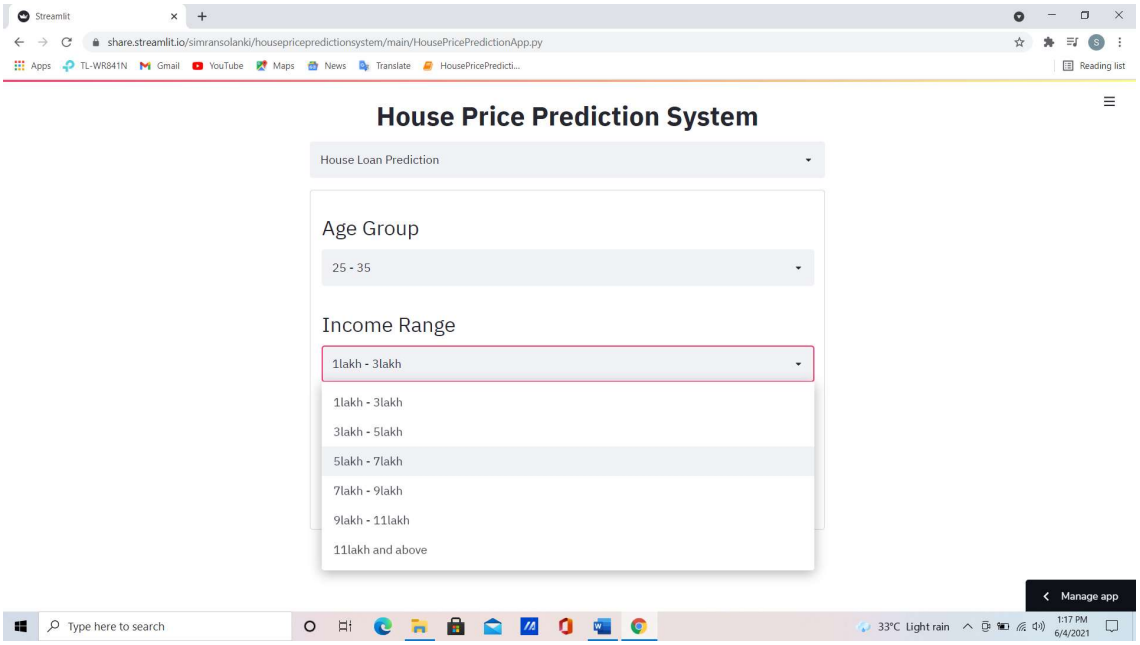
K-Nearest Neighbor Classification model was used for predicting whether the buyer will take House Loan or not



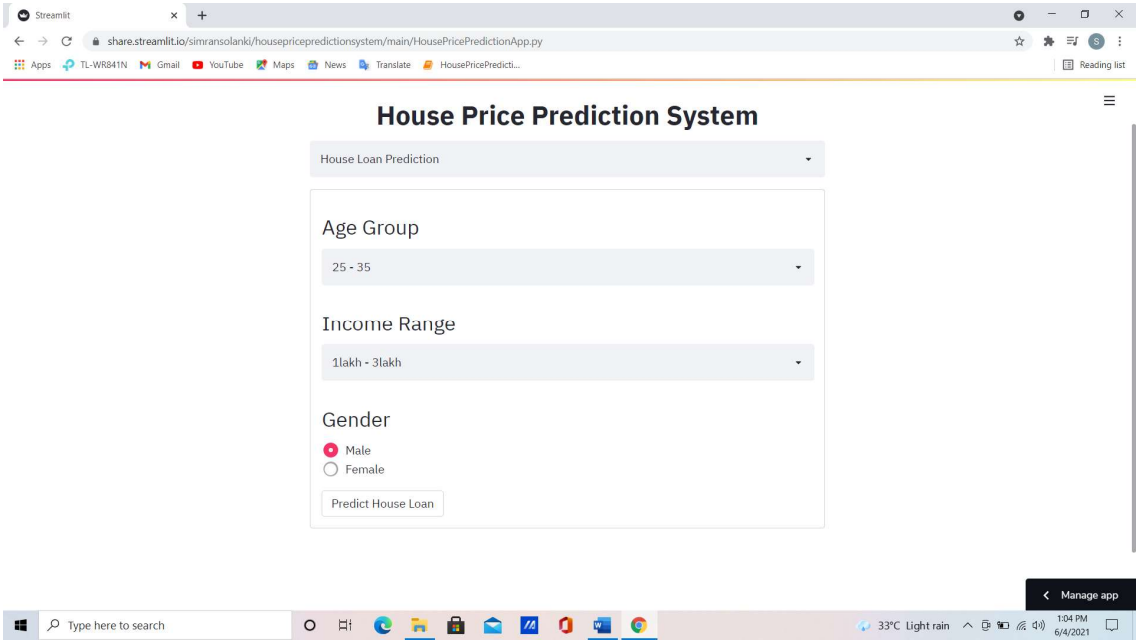
1) Select the age group from the dropdown list



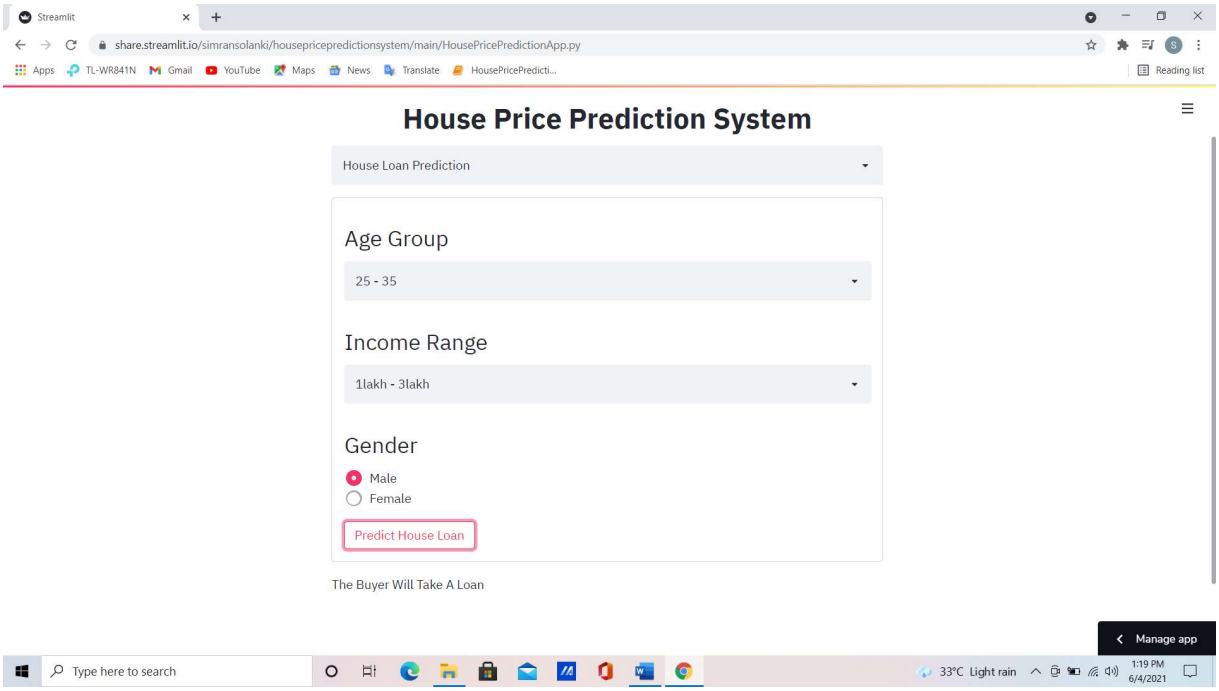
2) Select the income range from the dropdown list



3) Select Gender

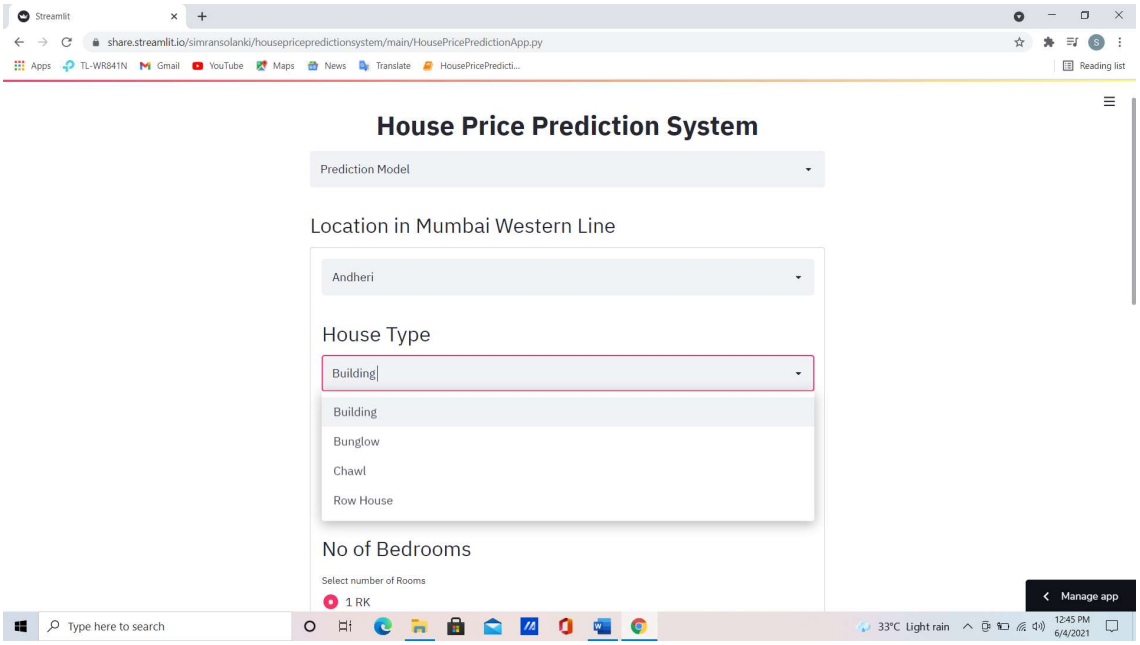


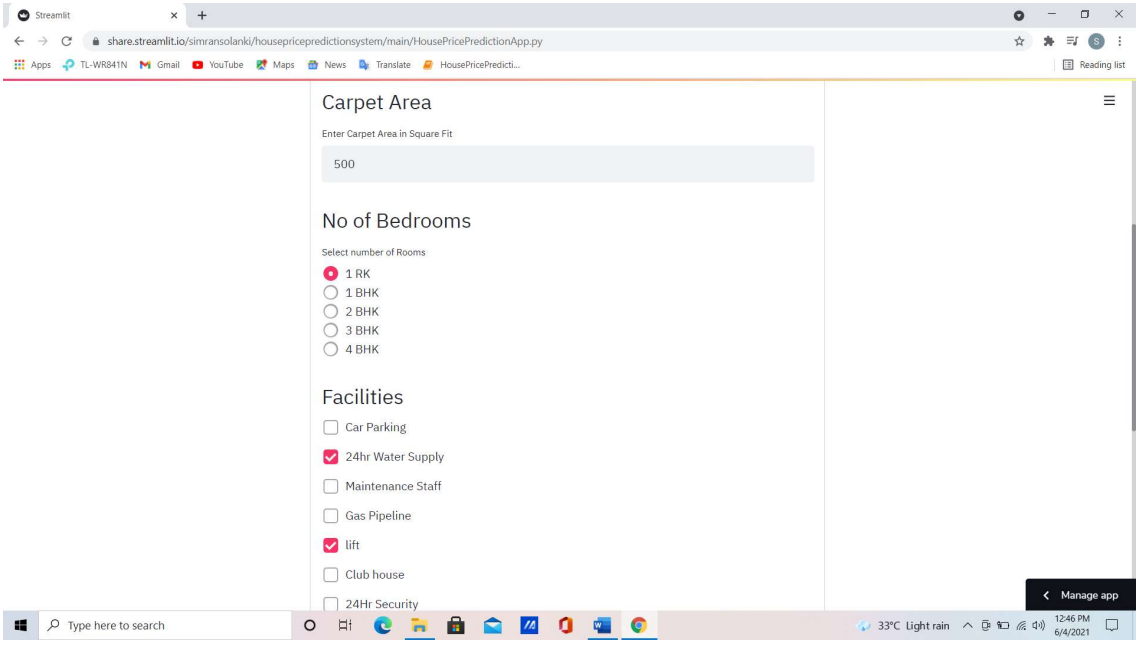
4) Click on the Predict House Loan button (This will display whether the buyer will take house loan or not)

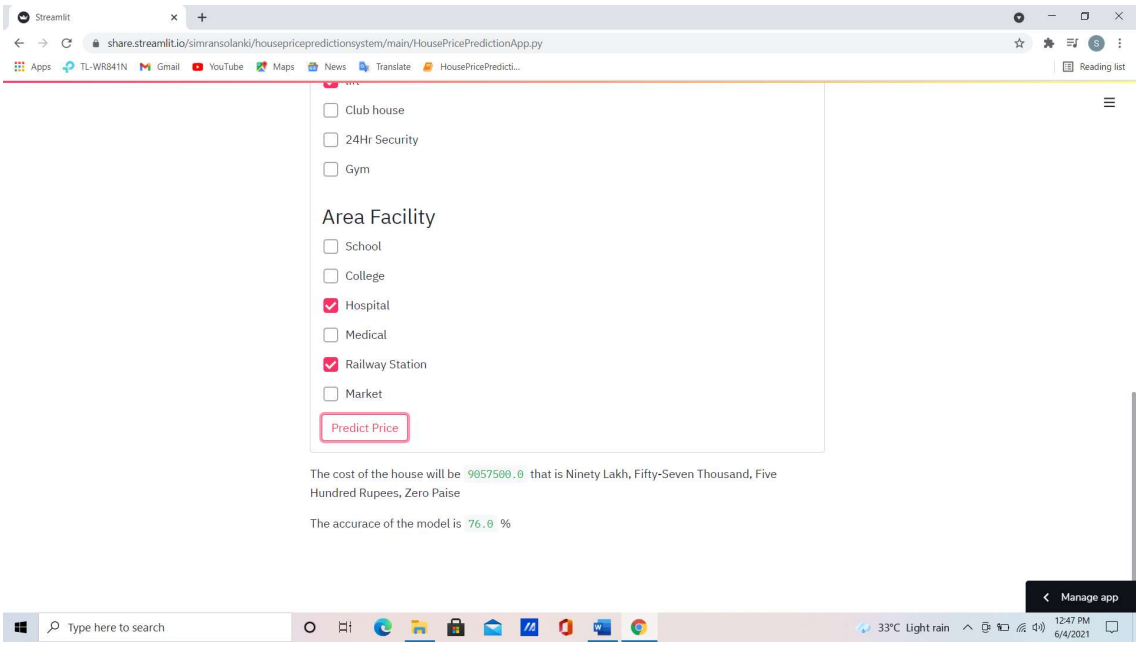


Do add some heading over hear

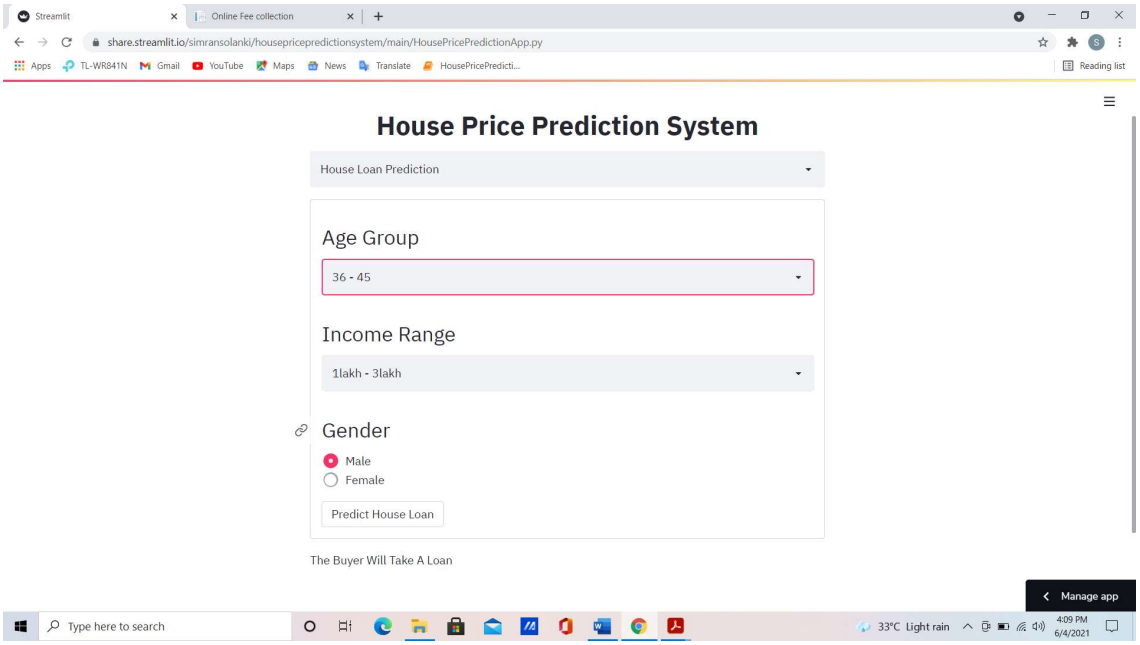
1) Prediction Model



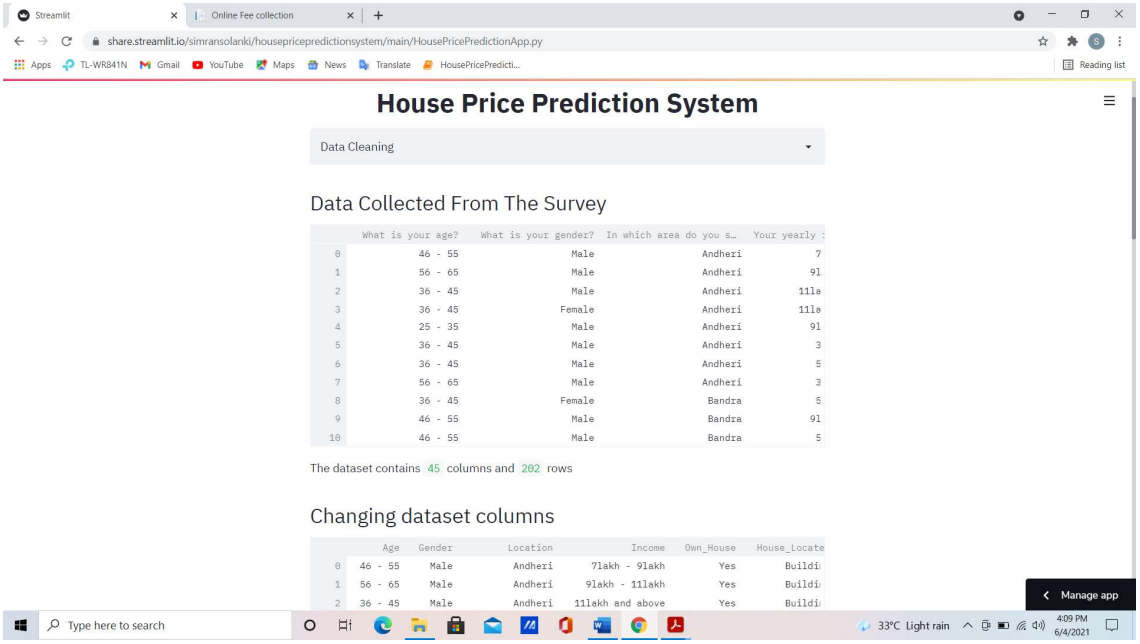

Streamlit
share.streamlit.io/simransolanki/housepricepredictionsystem/main/HousePricePredictionApp.py
Carpet Area
Enter Carpet Area in Square Fit
500
No of Bedrooms
Select number of Rooms
1 RK
1 BHK
2 BHK
3 BHK
4 BHK
Facilities
Car Parking
24hr Water Supply
Maintenance Staff
Gas Pipeline
lift
Club house
24Hr Security
Manage app
Type here to search
33°C Light rain
12:46 PM
6/4/2021

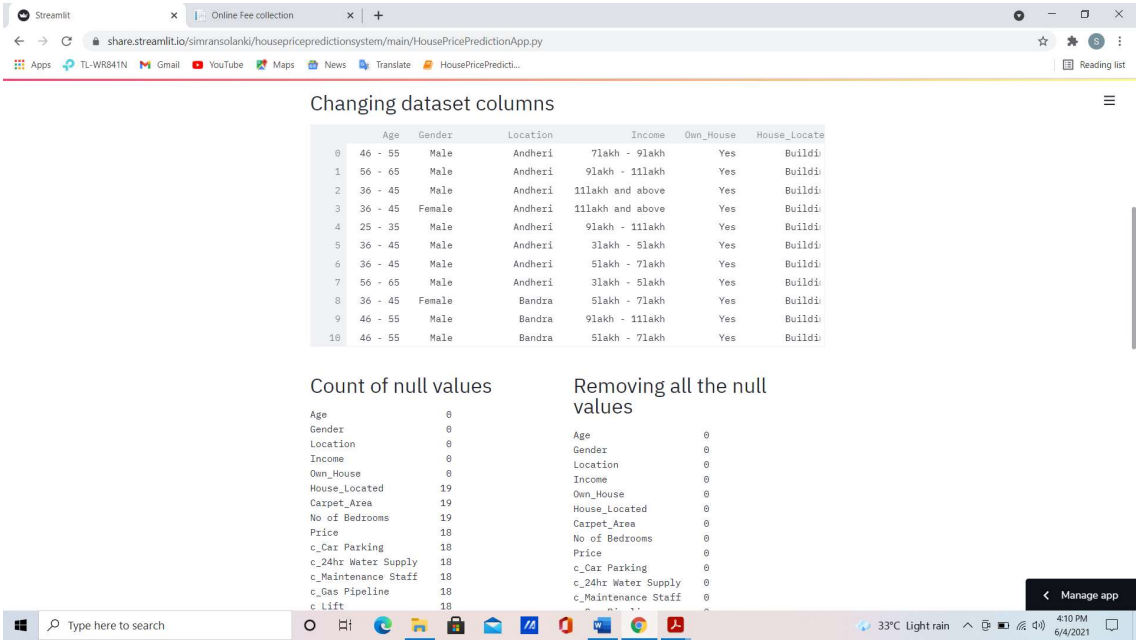

Streamlit
share.streamlit.io/simransolanki/housepricepredictionsystem/main/HousePricePredictionApp.py
Club house
24Hr Security
Gym
Area Facility
School
College
Hospital
Medical
Railway Station
Market
Predict Price
The cost of the house will be 9057500.0 that is Ninety Lakh, Fifty-Seven Thousand, Five Hundred Rupees, Zero Paise
The accurace of the model is 76.0 %
Manage app
Type here to search
33°C Light rain
12:47 PM
6/4/2021

2) House Loan Prediction



3) Data Cleaning

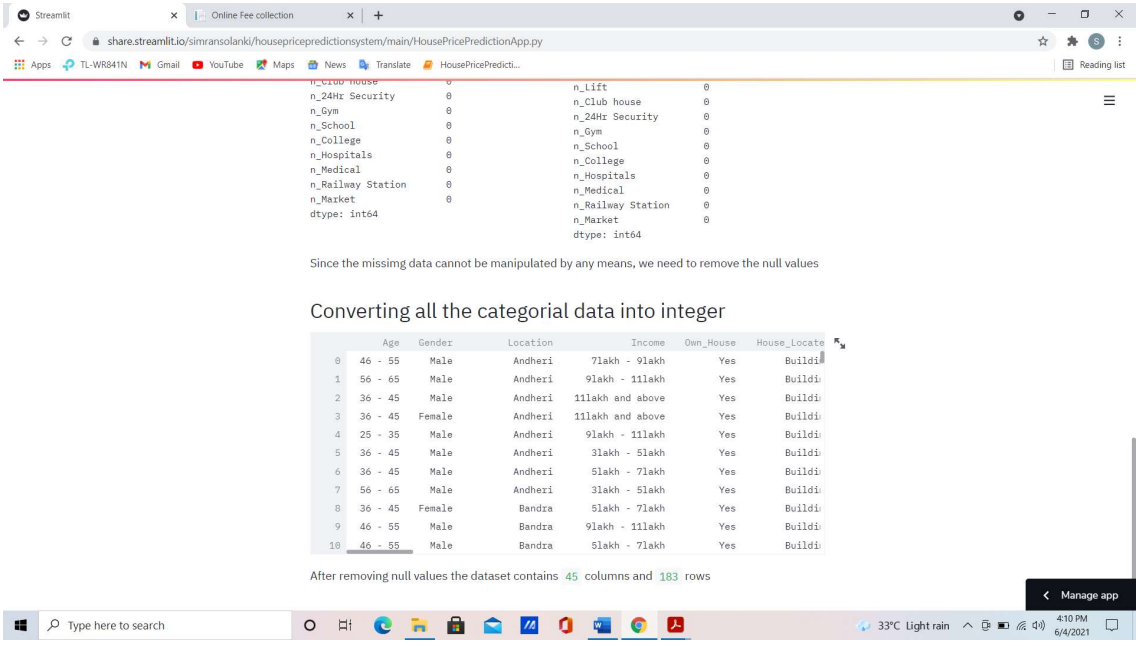




	Age	Gender	Location	Income	Own_House	House_Locate
0	46 - 55	Male	Andheri	7lakh - 9lakh	Yes	Buildi
1	56 - 65	Male	Andheri	9lakh - 11lakh	Yes	Buildi
2	36 - 45	Male	Andheri	11lakh and above	Yes	Buildi
3	36 - 45	Female	Andheri	11lakh and above	Yes	Buildi
4	25 - 35	Male	Andheri	9lakh - 11lakh	Yes	Buildi
5	36 - 45	Male	Andheri	3lakh - 5lakh	Yes	Buildi
6	36 - 45	Male	Andheri	5lakh - 7lakh	Yes	Buildi
7	56 - 65	Male	Andheri	3lakh - 5lakh	Yes	Buildi
8	36 - 45	Female	Bandra	5lakh - 7lakh	Yes	Buildi
9	46 - 55	Male	Bandra	9lakh - 11lakh	Yes	Buildi
10	46 - 55	Male	Bandra	5lakh - 7lakh	Yes	Buildi

Age	0
Gender	0
Location	0
Income	0
Own_House	0
House_Located	19
Carpet_Area	19
No of Bedrooms	19
Price	18
c_Car Parking	18
c_24hr Water Supply	18
c_Maintenance Staff	18
c_Gas Pipeline	18
c_Lift	18

Age	0
Gender	0
Location	0
Income	0
Own_House	0
House_Located	0
Carpet_Area	0
No of Bedrooms	0
Price	0
c_Car Parking	0
c_24hr Water Supply	0
c_Maintenance Staff	0

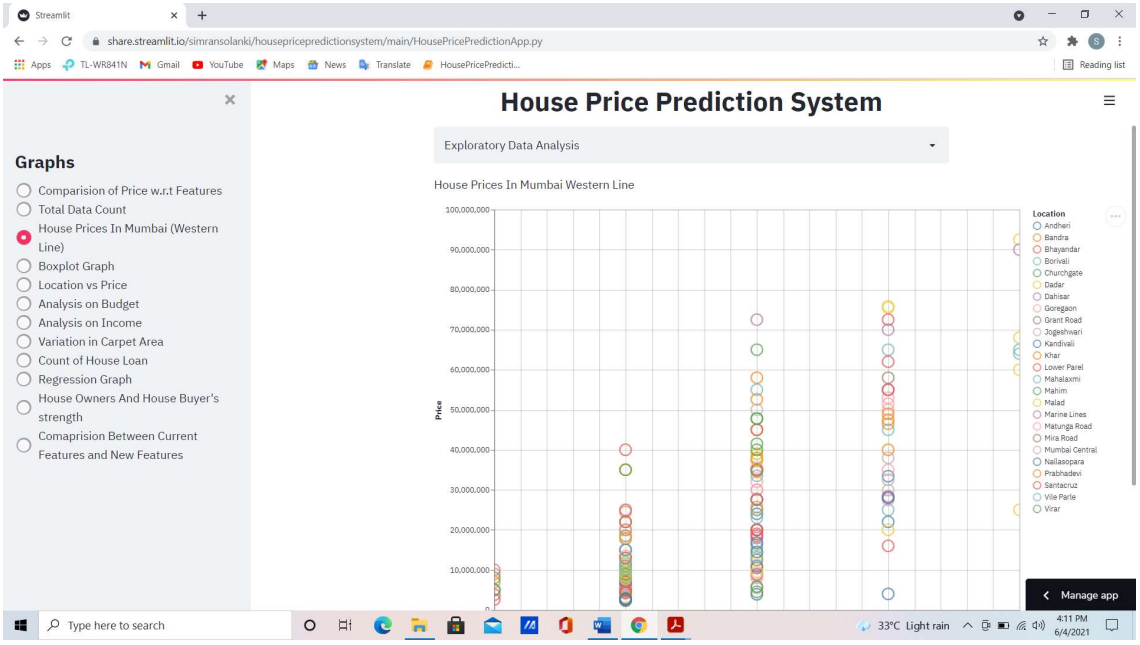


n_24Hr Security	0
n_Gym	0
n_School	0
n_College	0
n_Hospitals	0
n_Medical	0
n_Railway Station	0
n_Market	0
dtype: int64	

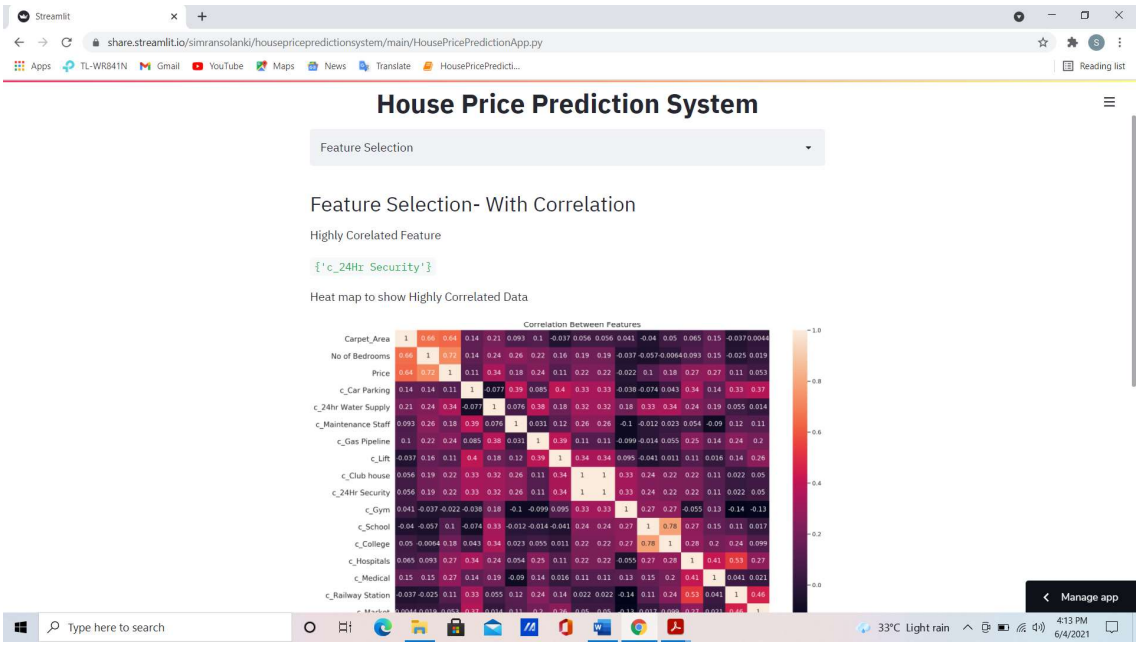
n_Lift	0
n_Club house	0
n_24Hr Security	0
n_Gym	0
n_School	0
n_College	0
n_Hospitals	0
n_Medical	0
n_Railway Station	0
n_Market	0
dtype: int64	

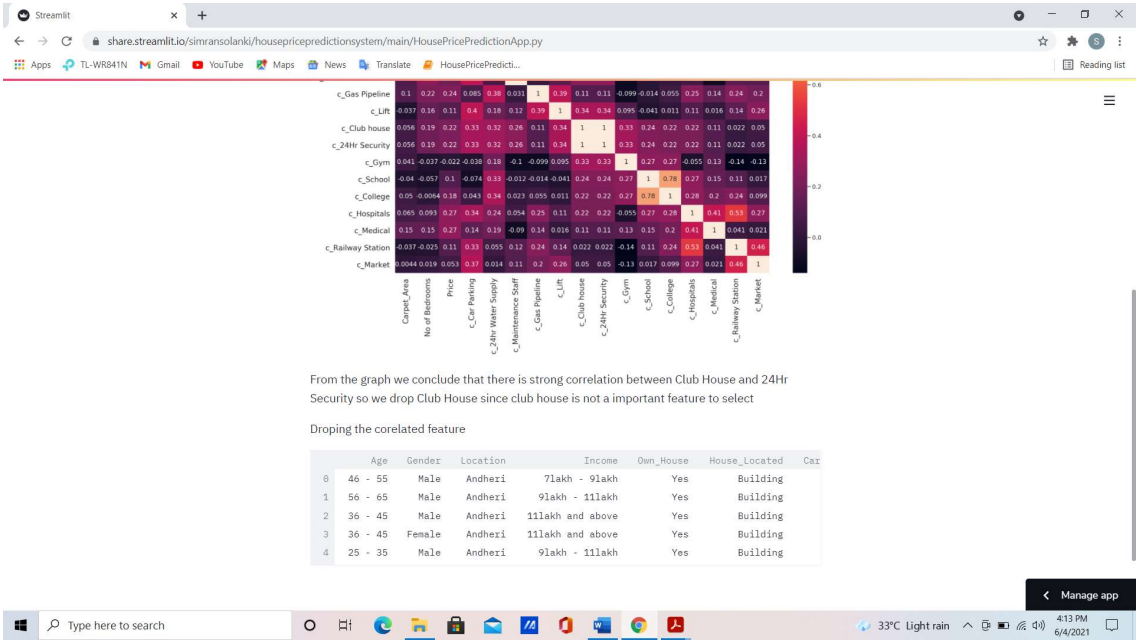
	Age	Gender	Location	Income	Own_House	House_Locate
0	46 - 55	Male	Andheri	7lakh - 9lakh	Yes	Buildi
1	56 - 65	Male	Andheri	9lakh - 11lakh	Yes	Buildi
2	36 - 45	Male	Andheri	11lakh and above	Yes	Buildi
3	36 - 45	Female	Andheri	11lakh and above	Yes	Buildi
4	25 - 35	Male	Andheri	9lakh - 11lakh	Yes	Buildi
5	36 - 45	Male	Andheri	3lakh - 5lakh	Yes	Buildi
6	36 - 45	Male	Andheri	5lakh - 7lakh	Yes	Buildi
7	56 - 65	Male	Andheri	3lakh - 5lakh	Yes	Buildi
8	36 - 45	Female	Bandra	5lakh - 7lakh	Yes	Buildi
9	46 - 55	Male	Bandra	9lakh - 11lakh	Yes	Buildi
10	46 - 55	Male	Bandra	5lakh - 7lakh	Yes	Buildi

4) Exploratory Data Analysis



5) Feature Selection



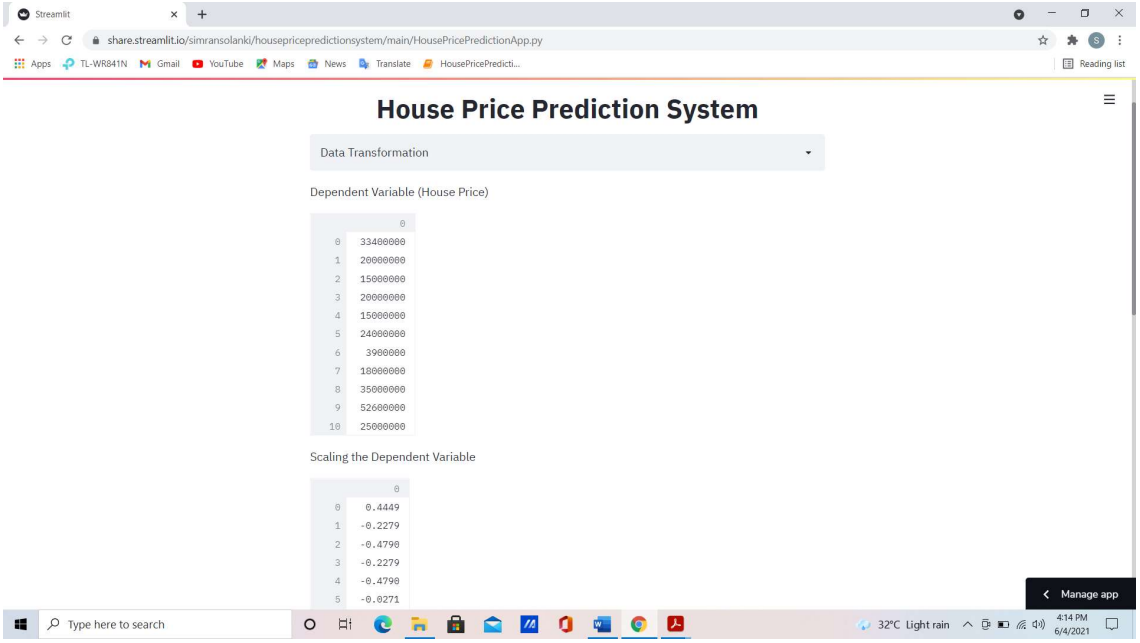


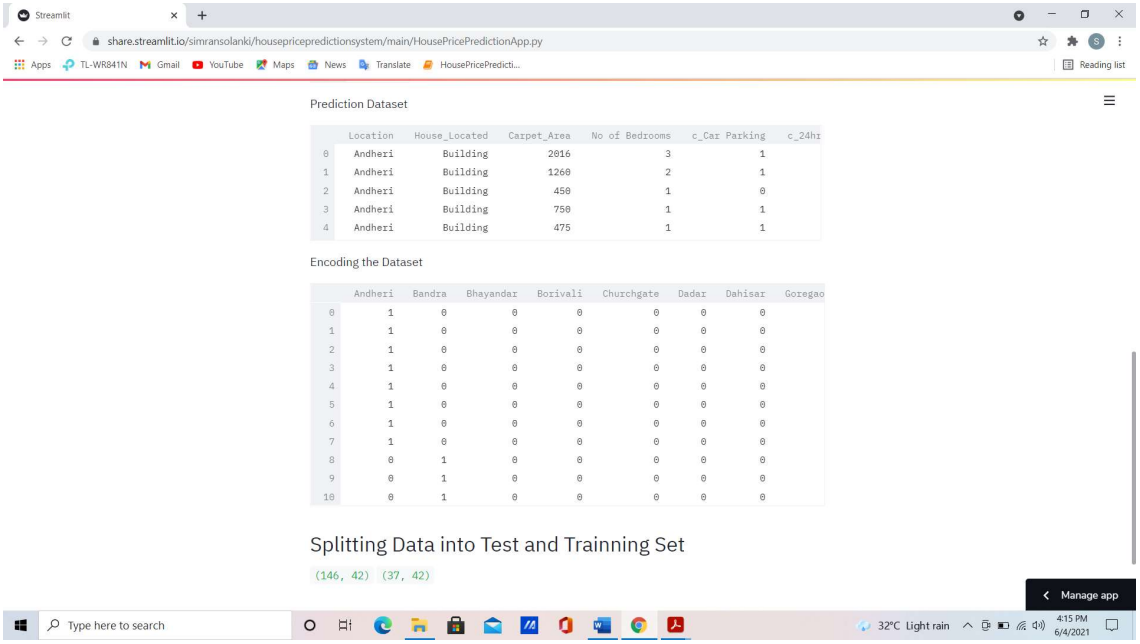
From the graph we conclude that there is strong correlation between Club House and 24Hr Security so we drop Club House since club house is not a important feature to select

Droping the corelated feature

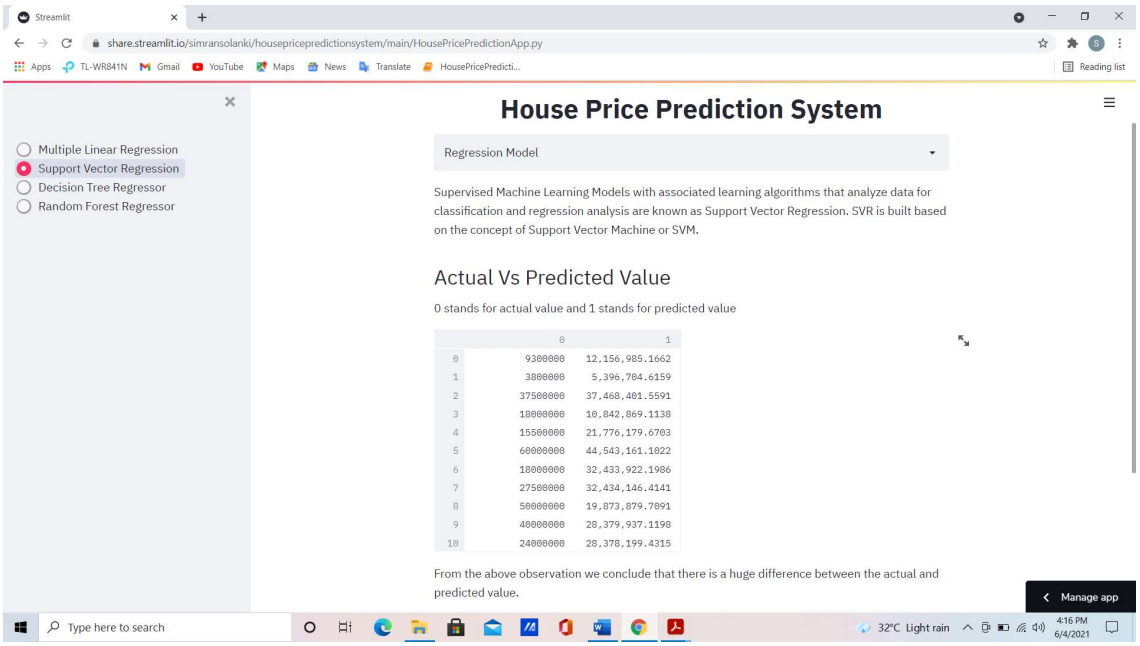
	Age	Gender	Location	Income	Own_House	House_Located	Car
0	46 - 55	Male	Andheri	7lakh - 9lakh	Yes	Building	
1	56 - 65	Male	Andheri	9lakh - 11lakh	Yes	Building	
2	36 - 45	Male	Andheri	11lakh and above	Yes	Building	
3	36 - 45	Female	Andheri	11lakh and above	Yes	Building	
4	25 - 35	Male	Andheri	9lakh - 11lakh	Yes	Building	

6 Data Transformation

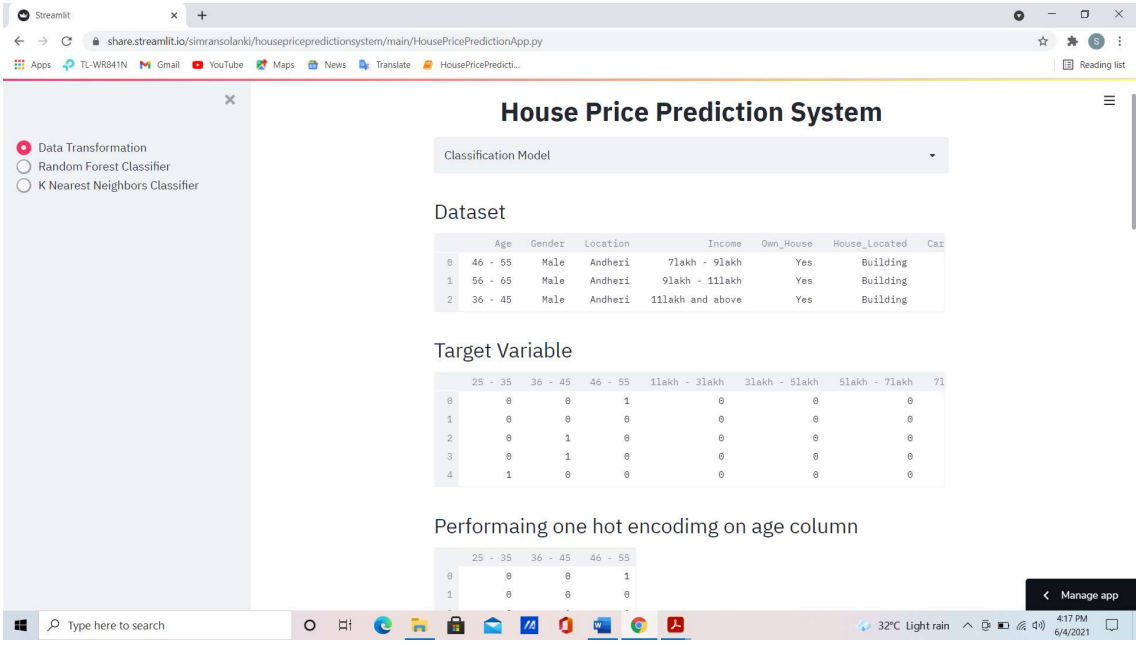




7) Regression Models



8) Classification Model



11 SOURCE CODE

```
import streamlit as st
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import preprocessing
from sklearn.svm import SVR
from sklearn import metrics
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import cross_val_score
from sklearn.metrics import confusion_matrix,classification_report,accuracy_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from num2words import num2words
import altair as alt
sns.set(color_codes=True)
og_dataset = pd.read_csv('dataset/Maindataset.csv')
pd.options.display.max_columns = None

def display_original_dataset():
    st.write(og_dataset)
    st.write('The dataset contains ',og_dataset.shape[1],' columns and ',og_dataset.shape[0],' rows')
```

```
dataset=og_dataset.rename({'What is your age?':'Age',
        'What is your gender?':'Gender',
        'In which area do you stay in western line in Mumbai?':'Location',
       'Your yearly income in INR':'Income',
        'Do you own a house?': 'Own_House',
        'Where is your house located?':'House_Located',
        'What is the carpet area of your current house in square feet?':'Carpet_Area',
        'What is the current price of your house in INR?':'Price',
        'How many rooms do you have?':'No of Bedrooms',
        'Are you planning to buy a new house in Mumbai(Western Line)?':'New_House',
        'Would you take a housing loan if you had to buy a house?':'House_Loan',
      'In which area you would like to buy a new house in Mumbai(Western Line) if you had to?':'n_Location',
      'What type of house you would look for if you had to buy a house?':'n_House_Type',
      'What would be your budget if you had to buy a house?':'Budget',
      'How much carpet area would you want(square feet) if you had to buy a house?':'n_Carpet_Area',
      'How many rooms do you want if you had to buy a house?':'n_Bedrooms',
      'If you had to buy a house what would you buy?':'New/Resale'},axis=1)


st.markdown("<h1 style='text-align: center;'>House Price Prediction System</h1>", unsafe_allow_html=True)
option = st.selectbox("",["Prediction Model","House Loan Prediction","Data Cleaning","Exploratory Data Analysis","Feature Selection","Data Transformation","Regression Model","Classification Model"])
dataset1=dataset.copy()
dataset_with_null = dataset
dataset = dataset.dropna()
data = dataset


def impute_carparking(cols):
   CarParking = cols[0]
   if("Car Parking" in CarParking):
      return 1
```

```
    else:
        return 0


def impute_watersupply(cols):
    watersupply = cols[0]
    if("24hr Water Supply" in watersupply):
        return 1
    else:
        return 0
def impute_maintenance_staff(cols):
    maintenance_staff = cols[0]
    if("Maintenance Staff" in  maintenance_staff ):
        return 1
    else:
        return 0
def impute_Lift(cols):
    lift = cols[0]
    if("Lift" in lift):
        return 1
    else:
        return 0
def impute_gaspipeline(cols):
    gas_pipeline = cols[0]
    if("Gas Pipeline" in gas_pipeline ):
        return 1
    else:
        return 0
def impute_gym(cols):
    gym = cols[0]
    if("Gym" in gym):
        return 1
    else:
```

```
        return 0
    def impute_club_house(cols):
        gym = cols[0]
        if("Club house" in gym):
            return 1
        else:
            return 0
    def impute_24hr_security(cols):
        security = cols[0]
        if("24Hr Security" in security):
            return 1
        else:
            return 0

    def impute_school(cols):
        school = cols[0]
        if("School" in school):
            return 1
        else:
            return 0

    def impute_college(cols):
        college = cols[0]
        if("College" in college):
            return 1
        else:
            return 0

    def impute_medical(cols):
        medical = cols[0]
        if("Medical" in medical):
            return 1
```

```
    else:
        return 0

def impute_hospitals(cols):
    hospitals = cols[0]
    if("Hospitals" in hospitals):
        return 1
    else:
        return 0

def impute_railwayStation(cols):
    railway = cols[0]
    if("Railway Station" in railway):
        return 1
    else:
        return 0

def impute_market(cols):
    market = cols[0]
    if("Market" in market):
        return 1
    else:
        return 0

def changeRooms(cols):
    rooms = cols
    if("BHK" in rooms):
        return int(rooms.split(' ')[0])
    else:
        return int(0)

data['c_Car Parking'] = data[['c_Car Parking']].apply(impute_carparking,axis=1)
```

```
data['c_24hr Water Supply'] = data[['c_24hr Water Supply']].apply(impute_watersupply,axis=1)
data['c_Maintenance Staff'] = data[['c_Maintenance Staff']].apply(impute_maintenance_staff,axis=1)
data['c_Lift'] = data[['c_Lift']].apply(impute_Lift,axis=1)
data['c_Gas Pipeline'] = data[['c_Gas Pipeline']].apply(impute_gaspipeline,axis=1)
data['c_Gym'] = data[['c_Gym']].apply(impute_gym,axis=1)
data['c_24Hr Security'] = data[['c_24Hr Security']].apply(impute_24hr_security,axis=1)
data['c_Club house'] = data[['c_Club house']].apply(impute_24hr_security,axis=1)

data['c_School'] = data[['c_School']].apply(impute_school,axis=1)
data['c_College'] = data[['c_College']].apply(impute_college,axis=1)
data['c_Medical'] = data[['c_Medical']].apply(impute_medical,axis=1)
data['c_Hospitals'] = data[['c_Hospitals']].apply(impute_hospitals,axis=1)
data['c_Railway Station'] = data[['c_Railway Station']].apply(impute_railwayStation,axis=1)
data['c_Market'] = data[['c_Market']].apply(impute_market,axis=1)
data['n_Car Parking'] = data[['n_Car Parking']].apply(impute_carparking,axis=1)
data['n_24hr Water Supply'] = data[['n_24hr Water Supply']].apply(impute_watersupply,axis=1)
data['n_Maintenance Staff'] = data[['n_Maintenance
Staff']].apply(impute_maintenance_staff,axis=1)
data['n_Lift'] = data[['n_Lift']].apply(impute_Lift,axis=1)
data['n_Gas Pipeline'] = data[['n_Gas Pipeline']].apply(impute_gaspipeline,axis=1)
data['n_Gym'] = data[['n_Gym']].apply(impute_gym,axis=1)
data['n_24Hr Security'] = data[['n_24Hr Security']].apply(impute_24hr_security,axis=1)
data['n_Club house'] = data[['n_Club house']].apply(impute_24hr_security,axis=1)
data['n_School'] = data[['n_School']].apply(impute_school,axis=1)
data['n_College'] = data[['n_College']].apply(impute_college,axis=1)
data['n_Medical'] = data[['n_Medical']].apply(impute_medical,axis=1)
data['n_Hospitals'] = data[['n_Hospitals']].apply(impute_hospitals,axis=1)
data['n_Railway Station'] = data[['n_Railway Station']].apply(impute_railwayStation,axis=1)
data['n_Market'] = data[['n_Market']].apply(impute_market,axis=1)
data['No of Bedrooms'] =  data['No of Bedrooms'].apply(changeRooms)
#data['No of Bedrooms'].apply(changeRooms)
data['n_Bedrooms'] = data['n_Bedrooms'].apply(changeRooms)
```

```
predictionDataset = data.iloc[:,2:-22]
predictionDataset.drop({'Price','Income','Own_House'},axis=1,inplace=True)

dependent_var = data.iloc[:,8].values
x = predictionDataset.iloc[:,:].values
dependent_var = dependent_var.reshape(len(dependent_var),1)
sc_y = StandardScaler()
y = sc_y.fit_transform(dependent_var)

dummies_location = pd.get_dummies(data.Location)
dummies_house_located = pd.get_dummies(data.House_Located)
dummies_location = dummies_location.drop(['Virar'],axis='columns')
dummies_house_located = dummies_house_located.drop(['Bungalow'],axis='columns')
dummies = pd.concat([dummies_location,dummies_house_located],axis="columns")

x = pd.concat([dummies,predictionDataset],axis='columns')
x = x.drop(['Location','House_Located'],axis='columns')
x = pd.DataFrame(x)

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 0)

feature = ['Age', 'Gender', 'Location', 'Income','House_Located', 'No of Bedrooms',  'c_Car Parking',
        'c_24hr Water Supply', 'c_Maintenance Staff', 'c_Gas Pipeline',
        'c_Lift', 'c_Club house', 'c_24Hr Security', 'c_Gym', 'c_School',
        'c_College', 'c_Hospitals', 'c_Medical', 'c_Railway Station',
        'c_Market', ]

list(enumerate(feature))
# data1 = data.copy()
def totalCount():
    plt.figure(figsize = (15, 100))
    for i in enumerate(feature):
```

```
        data1 = data.copy()
        plt.subplot(10, 2,i[0]+1).set_title("Total Data Count",fontsize=13)
        sns.countplot(i[1], data = data1)
        plt.xticks(rotation = 50)
        plt.tight_layout()
    st.pyplot()

def locationPrice():
    #st.write("House Prices In Mumbai Western Line")
    c= alt.Chart(data).mark_bar(size=20).encode(
    x='Location',
    y='Price',
    color='No of Bedrooms',
    tooltip = ['Carpet_Area'],
    order=alt.Order(
     # Sort the segments of the bars by this field
     'No of Bedrooms',
     sort='ascending'
    )
    ).properties(width = 900,height = 500)
    st.altair_chart(c)

def locationCarpetArea():
    fig = plt.figure(figsize = (19,8))
    sns.barplot(x="Carpet_Area", y="Price",data=data,ci=None)
    plt.title('Location vs Price',fontsize=20)
    plt.xticks(rotation = 90)
    plt.ticklabel_format(style='plain', axis='y')
    st.pyplot(fig)

le = LabelEncoder()
data['House_Loan'] = le.fit_transform(data['House_Loan'])
```

```
def budgetLoan():
    fig = plt.figure(figsize = (15,8))
    sns.barplot(x='Budget',y='House_Loan',data=data,ci=None)
    plt.title('Budget vs House Loan',fontsize=15)
    plt.xlabel("Budget",fontsize=15)
    plt.ylabel("House Loan",fontsize=15)
    plt.xticks(rotation = 40);
    st.pyplot(fig)

def budgetCarpet():
    fig = plt.figure(figsize = (15,8))
    sns.barplot(x='Budget',y='n_Carpet_Area',hue='n_Bedrooms',data=data,ci=None)
    plt.title('Budget vs Carpet_Area',fontsize=15)
    plt.xlabel("Budget",fontsize=15)
    plt.ylabel("Carpet Area",fontsize=15)
    plt.xticks(rotation = 40);
    st.pyplot(fig)

def incomeLoan():
    fig = plt.figure(figsize = (13,8))
    sns.barplot(x='Income',y='House_Loan',data=data,ci=None)
    plt.title('Income vs House Loan',fontsize=15)
    plt.xlabel("Income",fontsize=15)
    plt.ylabel("House Loan",fontsize=15)
    plt.xticks(rotation = 40);
    st.pyplot(fig)

def incomeCarpet():
    sns.set_context(font_scale=1.5)
    fig = plt.figure(figsize = (13,8))
    sns.barplot(x='Income',y='n_Carpet_Area',data=data,ci=None)
    plt.title('Income vs Carpet Area',fontsize=15)
```

```
    plt.xlabel("Income",fontsize=15)
    plt.ylabel("Carpet Area",fontsize=15)
    plt.xticks(rotation = 40);
    st.pyplot(fig)

carpet_area = data['Carpet_Area'].unique()
np.sort(carpet_area)
def CarpetArea():
    sns.set_context(font_scale=1.5)
    fig = plt.figure(figsize=(10,10))
    sns.distplot(data['Carpet_Area'])
    plt.xlabel("Carpet Area",fontsize=15)
    plt.title('Variation in Carpet Area',fontsize=15)
    st.pyplot(fig)

def countLoan():

    fig = plt.figure(figsize = (25,15))
    sns.set_context(font_scale=1.5)
    sns.countplot(x='Carpet_Area',hue='House_Loan',data=data)
    plt.xlabel("Carpet Area",fontsize=15)
    plt.ylabel("House Loan",fontsize=15)
    plt.title('Count of House Loan',fontsize=15)
    plt.xticks(rotation = 90);
    st.pyplot(fig)

def carpetNbedrooms():
    sns.set_context(font_scale=1.5)
    fig = sns.lmplot(x='Carpet_Area',y='No of Bedrooms',data=data,aspect=2)
    plt.xlabel("Carpet Area",fontsize=15)
    plt.ylabel("No of Bedrooms",fontsize=15)
    plt.title('Carpet Area Vs No of Bedrooms',fontsize=15)
```

```
    st.pyplot(fig)

def BedroomsCarpetArea():
    sns.set_context(font_scale=1.5)
    fig = plt.figure(figsize=(15,8))
    sns.boxplot(x='No of Bedrooms',y='Carpet_Area',data=data,palette='rainbow')
    plt.xlabel("No of Bedrooms",fontsize=15)
    plt.ylabel("Carpet Area",fontsize=15)
    plt.title('No of Bedrooms vs Carpet Area',fontsize=15)
    st.pyplot(fig)

def CarpetAreaPrice():
    sns.set_context('paper',font_scale=1.5)
    fig = sns.lmplot(x='Carpet_Area',y='Price',data=data,aspect=2)
    plt.xlabel("Carpet Area",fontsize=15)
    plt.title('Carpet Area vs Price',fontsize=15)
    st.pyplot(fig)

def HouseOwners():
    fig = plt.figure(figsize = (15,8))
    plt.subplot(1,2,1)
    plt.title('Genderwise Count w.r.t House Owners',fontsize=15)
    sns.countplot(x="Gender", data=dataset1, hue='Own_House')
    plt.subplot(1,2,2)
    plt.title('Agewise Count w.r.t House Owners',fontsize=15)
    sns.countplot(x="Age", data=dataset1, hue='Own_House')
    st.pyplot(fig)

def NewOwners():
    fig = plt.figure(figsize = (15,7))
    plt.subplot(1,2,1)
    plt.title('Gender wise Count w.r.t New House Buyers',fontsize=15)
```

```
    sns.countplot(x="Gender", data=dataset1, hue='New_House')
    plt.subplot(1,2,2)
    plt.title('Age wise Count w.r.t New House Buyers',fontsize=15)
    sns.countplot(x="Age", data=dataset1, hue='New_House')
    st.pyplot(fig)


compare_features=['House_Located','n_House_Type','c_Car Parking','n_Car Parking','c_24hr Water Supply','n_24hr Water Supply',
        'c_Maintenance Staff','n_Maintenance Staff','c_Gas Pipeline','n_Gas Pipeline',
        'c_Lift','n_Lift', 'c_Club house','n_Club house', 'c_24Hr Security','n_24Hr Security',
        'c_Gym','n_Gym', 'c_School', 'n_School','c_College','n_College' ,'c_Hospitals','n_Hospitals',
        'c_Medical','n_Medical' ,'c_Railway Station','n_Railway Station','c_Market','n_Market']


def OldFeatureNewFeature(compare_features):
    fig = plt.figure(figsize = (15,140))
    for i in enumerate(compare_features):
        data1 = data.copy()
        plt.subplot(15, 2,i[0]+1)
        sns.countplot(i[1], data = data1).set_title("Comaprision Between Current Features and New Features",fontsize=18)
        plt.xticks(rotation = 50)
        plt.tight_layout()
    st.pyplot(fig)


FinalData = data.iloc[:,6:23]
def FinalDataCorr():
    fig = plt.figure(figsize=(16,10))
    sns.heatmap(FinalData.corr(), annot=True)
    sns.set_style('white')
    plt.title('Correlation Between Features',fontsize=15)
    plt.xticks(fontsize=14)
    plt.yticks(fontsize=14)
    st.pyplot(fig)
```

```
def correlation(dataset,threshold):
    col_corr = set()
    corr_matrix = dataset.corr()
    for i in range(len(corr_matrix.columns)):
        for j in range(i):
            if(abs(corr_matrix.iloc[i, j]) > threshold):
                colname = corr_matrix.columns[i]
                col_corr.add(colname)
    return col_corr

# Mutiple Linear Regression
linear_regressor = LinearRegression()
linear_regressor.fit(X_train, y_train)

# SVR
y_train1 = y_train.flatten()
svr_regressor = SVR(kernel = 'rbf')
svr_regressor.fit(X_train,y_train1)

y_pred_svr = svr_regressor.predict(X_test)

# decision tree regression
tree_regressor = DecisionTreeRegressor(random_state = 1)
tree_regressor.fit(X_train, y_train)

#Random forest regression
forest_regressor = RandomForestRegressor(n_estimators = 20, random_state = 1)
forest_regressor.fit(X_train, y_train1)

y_pred_forest = forest_regressor.predict(X_test)
```

```
y_pred_tree = tree_regressor.predict(X_test)


#Classification
target_var = dataset1.iloc[:,0:4]
target_var = target_var.drop('Location',axis=1)
target_var1 = target_var
dummies_age = pd.get_dummies(target_var.Age)
dummies_age = dummies_age.drop(['56 - 65'],axis='columns')
dummies_income = pd.get_dummies(target_var.Income)
dummies_income = dummies_income.drop(['11lakh and above'],axis='columns')
target_dummies = pd.concat((dummies_age,dummies_income),axis='columns')
le = LabelEncoder()
target_var['Gender'] = le.fit_transform(target_var['Gender'])
target_var = pd.concat((target_dummies,target_var),axis='columns')
target_var = target_var.drop(['Age','Income'],axis='columns')

dataset1['House_Loan'] = le.fit_transform(dataset1['House_Loan'])
y_c = dataset1.iloc[:,24].values

X_train_c, X_test_c, y_train_c, y_test_c =
train_test_split(target_var,y_c,test_size=0.2,random_state=0)

knn_classifier = KNeighborsClassifier(n_neighbors=23)
knn_classifier.fit(X_train_c,y_train_c)
knn_pred = knn_classifier.predict(X_test_c)

knn_accuracy_rate = []
for i in range(1,40):
   knn = KNeighborsClassifier(n_neighbors=i)
   score=cross_val_score(knn,target_var,y_c,cv=10)
   knn_accuracy_rate.append(score.mean())

knn_error_rate = []
```

```
for i in range(1,40):
    knn = KNeighborsClassifier(n_neighbors=i)
    score=cross_val_score(knn,target_var,y_c,cv=10)
    knn_error_rate.append(1-score.mean())
randomforest_Classifier = RandomForestClassifier(n_estimators=20)
randomforest_Classifier.fit(X_train_c, y_train_c)
y_predicted_random = randomforest_Classifier.predict(X_test_c)
cm = confusion_matrix(y_test_c, y_predicted_random)


def predict_price(location,house_located,capet_area,No_of_Bedrooms,
c_Car_Parking,c_24hr_Water_Supply,
      c_Maintenance_Staff, c_Gas_Pipeline, c_Lift, c_Club_house,
      c_24Hr_Security, c_Gym, c_School, c_College, c_Hospitals,
      c_Medical, c_Railway_Station, c_Market):
    l = []
    loc = dummies_location.columns==location
    rooms = 0
    for i in loc:
     if(i):
       l.append(1)
     else:
       l.append(0)
    house_loc = dummies_house_located.columns==house_located
    for i in house_loc:
     if(i):
       l.append(1)
     else:
       l.append(0)
    l.append(capet_area)

    if(No_of_Bedrooms == "1 RK"):
       l.append(0)
    else:
```

```
    if("BHK" in No_of_Bedrooms):
        rooms=int(No_of_Bedrooms.split(' ')[0])
        l.append(rooms)

if(c_Car_Parking):
    l.append(1)
else:
    l.append(0)

if(c_24hr_Water_Supply):
    l.append(1)
else:
    l.append(0)

if(c_Maintenance_Staff):
    l.append(1)
else:
    l.append(0)

if(c_Gas_Pipeline):
    l.append(1)
else:
    l.append(0)

if(c_Lift):
    l.append(1)
else:
    l.append(0)

if(c_Club_house):
    l.append(1)
else:
```

```
    l.append(0)

if(c_24Hr_Security):
    l.append(1)
else:
    l.append(0)

if(c_Gym):
    l.append(1)
else:
    l.append(0)

if(c_School):
    l.append(1)
else:
    l.append(0)

if(c_College):
    l.append(1)
else:
    l.append(0)

if(c_Hospitals):
    l.append(1)
else:
    l.append(0)

if(c_Medical):
    l.append(1)
else:
    l.append(0)
```

```
    if(c_Railway_Station):
        l.append(1)
    else:
        l.append(0)

    if(c_Market):
        l.append(1)
    else:
        l.append(0)

    predicted_value = sc_y.inverse_transform(forest_regressor.predict([l]))
    return round(predicted_value[0],2)

def predict_knn(age,salary,gender):
    l = []
    age_list = dummies_age.columns==age
    sal_list = dummies_income.columns==salary

    for i in age_list:
      if(i):
        l.append(1)
      else:
        l.append(0)

    for i in sal_list:
      if(i):
        l.append(1)
      else:
        l.append(0)

    if(gender == "Male"):
        l.append(0)
```

```
    else:
        l.append(0)

    pred = knn_classifier.predict([l])

    return pred[0]

if(option == "Data Cleaning"):
    st.write("## Data Collected From The Survey")
    display_original_dataset()

    st.write("## Changing dataset columns")
    st.write(dataset)

    col1, col2 = st.beta_columns(2)

    col1.write("## Count of null values")
    col1.text(dataset_with_null.isnull().sum())

    col2.write("## Removing all the null values")
    col2.text(dataset.isnull().sum())

    st.write("Since the missimg data cannot be manipulated by any means, we need to remove the
null values")

    st.write("## Converting all the categorial data into integer")
    st.write(data)

    st.write('After removing null values the dataset contains ',dataset.shape[1],' columns and
',dataset.shape[0],' rows')

elif(option == "Exploratory Data Analysis"):
    st.sidebar.title("Graphs")
```

```
selectbox = st.sidebar.radio(label="", options=["Comparision of Price w.r.t Features", "Total Data Count", "House Prices In Mumbai (Western Line)",
"Boxplot Graph","Location vs Price","Analysis on Budget","Analysis on Income","Variation in Carpet Area","Count of House Loan",
"Regression Graph","House Owners And House Buyer's strength",
"Comaprision Between Current Features and New Features"])
if selectbox == "Comparision of Price w.r.t Features":
    numerical_features = [feature for feature in dataset.columns if dataset[feature].dtypes != 'O']
    data[numerical_features].head()
    c_features = data[numerical_features].iloc[:,:-16]
    c_features.drop(['Price','Carpet_Area','House_Loan'],axis='columns',inplace=True)
    c_features.head()
    def hp_features():
        st.set_option('deprecation.showPyplotGlobalUse', False)
        for feature in c_features:
            data1=data.copy()
            data1.groupby(feature)['Price'].mean().plot.bar()
            plt.xlabel(feature)
            plt.ylabel('Price')
            plt.title("Comparision of Price w.r.t Features")
            st.pyplot()
    hp_features()

    st.write("From the above graphs we can conclude that features such as No of BedRooms, 24Hr Water Supply, Gas Pipeline, Lift and medical are highly influenced for increase in price whereas other features does not affect price as much ")

elif selectbox == "Total Data Count":

    st.write("Count Plot shows the counts of observations in each categorical data using bars")
    totalCount()

elif selectbox == "House Prices In Mumbai (Western Line)":
```

```
    numerical_features = [feature for feature in dataset.columns if dataset[feature].dtypes != 'O']
    data[numerical_features].head()
    scatter_features = data[numerical_features].iloc[:,:-16]
    scatter_features['Location'] = data['Location']
    scatter_features.head()
    def housePrice():
      st.write("House Prices In Mumbai Western Line")
      c = alt.Chart(scatter_features).mark_point(size = 240).encode(
        x='No of Bedrooms', y='Price',color = 'Location',
tooltip=['Carpet_Area','Location','Price','c_Car Parking','c_24hr Water Supply','c_Maintenance
Staff','c_Lift','c_Gas Pipeline','c_Gym',
                'c_24Hr Security','c_School','c_College','c_Hospitals','c_Medical','c_Railway
Station','c_Market']).interactive().properties(width = 900,height= 600)
      st.altair_chart(c)
    housePrice()


  elif selectbox == "Location vs Price":


    locationPrice()


    st.markdown("""<style>
            table {
              border-collapse: collapse;
              width: 100%;
            }
            th, td {
              text-align: left;
              padding: 8px;
            }
            th {
              background-color: #FCFCFC;
              color: black;
            }
            #th_border th {
```

```
                border-color: black;
              }
            </style>""",unsafe_allow_html=True)
    st.markdown("<h2 style='text-align:center';>Analysis on Location with respect to Price</h2>",unsafe_allow_html=True)
    st.markdown("""<table style='width:100%' id='th_border'>
      <tr>
        <th colspan="3" style='text-align: center'>1 RK</th>
      </tr>
      <tr>
        <th>High</th>
        <th>Mid</th>
        <th>Low</th>
      </tr>
      <tr>
        <td>Marine Lines</td>
        <td>Mahim</td>
        <td>Dahisar</td>
      </tr>
       <tr>
        <td>Bandra</td>
        <td>Khar</td>
        <td>Bhayandar</td>
      </tr>
      <tr>
        <th colspan="3" style='text-align: center'>1 BHK</th>
      </tr>
      <tr>
        <th>High</th>
        <th>Mid</th>
        <th>Low</th>
      </tr>
       <tr>
```

```
    <td>Khar</td>
    <td>Churchgate</td>
    <td>Bhayandar</td>
</tr>
 <tr>
    <td>Lower Parel</td>
    <td>Andheri</td>
    <td>Borivali</td>
</tr>
 <tr>
    <td rowspan ="9"></td>
    <td>Marine Lines</td>
    <td>Dahissar</td>
</tr>
<tr>
    <td>Grant Road</td>
    <td>Jogeshwari</td>
<tr>
    <td>Vile Parle</td>
    <td>Kandivali</td>
</tr>
<tr>
    <td>Bandra</td>
    <td>Mahalaxmi</td>
</tr>
<tr>
    <td>Prabhadevi</td>
    <td>Malad</td>
</tr>
<tr>
    <td>Dadar</td>
    <td>Mumbai Central</td>
```

```
</tr>
<tr>
  <td>Mahin</td>
  <td>Nallasopara</td>
</tr>
<tr>
  <td>Virar</td>
  <td rowspan="2"></td>
</tr>
<tr>
  <td>Santacruz</td>
</tr>
<tr>
  <th colspan="3" style='text-align: center'>2 BHK</th>
</tr>
  <tr>
    <th>High</th>
    <th>Mid</th>
    <th>Low</th>
  </tr>
   <tr>
    <td>Marine Lines</td>
    <td>Bandra</td>
    <td>Bhayandar</td>
  </tr>
   <tr>
    <td>Mahalaxmi</td>
    <td>Dadar</td>
    <td>Borivali</td>
  </tr>
   <tr>
    <td>Churchagte</td>
```

```
      <td>Grant Road</td>
      <td>Dahisar</td>
    </tr>
    <tr>
      <td>Khar</td>
      <td>Jogeshwari</td>
      <td>Kandivali</td>
    </tr>
    <tr>
      <td>Mahim</td>
      <td>Lower Parel</td>
      <td>Malad</td>
    </tr>
  <tr>
    <td rowspan ="4">
    <td>Matunga Road</td>
    <td>Miraroad</td>
  </tr>
    <tr>
    <td>Prabhadevi</td>
    <td>Santacruz</td>
    </tr>
    <tr>
    <td>Vile Parle</td>
    <td>Virar</td>
    </tr>
    <tr>
    <td>Goregaon</td>
    <td></td>
    </tr>
    <tr>
    <th colspan="3" style='text-align: center'>3 BHK</th>
```

```
</tr>
<tr>
  <th>High</th>
  <th>Mid</th>
  <th>Low</th>
</tr>
<tr>
  <td>Dadar</td>
  <td>Andheri</td>
  <td>Bhayandar</td>
</tr>
<tr>
  <td>Lower Parel</td>
  <td>Borivali</td>
  <td>Malad</td>
</tr>
<tr>
  <td>Mahalaxmi</td>
  <td>Goregaon</td>
  <td>Nallasopara</td>
</tr>
<tr>
  <td>Grant Road</td>
  <td>Jogeshwari</td>
  <td rowspan="5"></td>
</tr>
<tr>
  <td>Matunga Road</td>
  <td>Kandivali</td>
</tr>
<tr>
  <td>Prabhadevi</td>
```

```
            <td>Khar</td>
          </tr>
          <tr>
            <td>Santacruz</td>
            <td rowspan="2"></td>
          </tr>
          <tr>
            <td>Vile Parle</td>
          </tr>
          <tr>
          <th colspan="3" style='text-align: center'>4 BHK</th>
          </tr>
          <tr>
            <th>High</th>
            <th>Mid</th>
            <th>Low</th>
          </tr>
          <tr>
            <td>Marine Lines</td>
            <td>Dadar</td>
            <td>Malad</td>
          </tr>
          <tr>
            <td colspan="1" >
            <td>Vile Parle</td>
            <td></td>
          </tr></table>""",unsafe_allow_html=True)
    # locationCarpetArea()

elif selectbox == "Analysis on Budget":

    budgetLoan()
```

```
        st.write("From the above graph we can observe that people have budget below 1 cr and above 6cr have the highest changes to take a house loan")


        budgetCarpet()

        st.write("From the above graph we conclude that:")

        st.write("Budget greater than 1Cr target 1,2,3 and 4 bhk with higher carpet area,ecxept budget range between 5Cr-7Cr target 3 and 4bhk with higher carpet area")

        st.write("Budget range between 50lakh and 90Lakh target 1,2 and 3bhk with higher carpet area ")

        st.write("Budget range less that 40lakh target 0,1 and 2bhk with medium carpet area")


    elif selectbox == "Analysis on Income":

        st.write("## Income vs House Loan")

        incomeLoan()

        st.write("From the above graph we can observe that people having income between 1 lakh - 3 lakh, 9 lakh - 11 lakh and 11 lakh and above have above 70% changes to take a house loan")


        st.write("## Income vs Carpet Area")

        incomeCarpet()


    elif selectbox == "Variation in Carpet Area":


        CarpetArea()

        st.write("From the above graph we can observe that the dataset contains carpet area majority between  300 to 1000 sq.ft")


    elif selectbox == "Count of House Loan":


        countLoan()

        st.write("From the above graph we can observe that majority of the people where willing to take the house loan")


    elif selectbox == "Boxplot Graph":

        BedroomsCarpetArea()
```

```
        st.write("From the above graph we can observe that there are some outliers in carpet area with respect to rooms such as in 0 No. of BedRooms i.e 1 RK range of carpet area lies between 100 to 800 therefore house having carpet area 3500 is a outlier, in 1 BK range of carpet area lies between 200 and 760 therefore we have 3 higher outliers with carpet area 1000,1010 and 2500, in 2 BK range of carpet area lies between 480 and 1500 therefore we have 1 higher outliers with carpet area 1750, in 3 BK range of carpet area lies between 900 and 2050 therefore we have 1 higher outliers with carpet area 2400, in 4 BK range of carpet area lies between 1990 and 2350 therefore we have 1 higher outliers with carpet area 3500 and one lower outlier 100.")


    elif selectbox == "Regression Graph":

        st.write("No of Bedrooms vs Carpet Area")


        carpetNbedrooms()

        st.write("From the above graph we can conclude that there is a linear relation between the No. of Bedrooms and Carpet Area as No. of Bedrooms increases Carpet Area also increases but the points do not fit on the regression line.")




        st.write("## Carpet Area vs Price")

        CarpetAreaPrice()

        st.write("From the above graph we can conclude that there is a linear relation between the House Price and Carpet Area as House Price increases Carpet Area also increases but the points do not fit on the regression line.")

    elif selectbox == "House Owners And House Buyer's strength":

        HouseOwners()

        NewOwners()

    elif selectbox == "Comaprision Between Current Features and New Features":

        OldFeatureNewFeature(compare_features)


        st.write("From the above graphs we can conclude that except Gym and Club House majority of the people wanted all the rest of the features")


elif(option == "Feature Selection"):

    st.write("## Feature Selection- With Correlation")

    corr_features = correlation(predictionDataset,0.85)

    st.write(" Highly Corelated Feature")

    corr_features
```

```
    predictionDataset.drop('c_Club house',axis=1)
    st.write("Heat map to show Highly Correlated Data")
    FinalDataCorr()
    st.write("From the graph we conclude that there is strong correlation between Club House and 24Hr Security so we drop Club House since club house is not a important feature to select")
    st.write("Droping the corelated feature")
    st.write(data.head())


elif(option == "Data Transformation"):
    st.write(" Dependent Variable (House Price)")
    st.write(dependent_var)
    st.write(" Scaling the Dependent Variable")
    st.write(y)
    st.write(" Prediction Dataset")
    st.write(predictionDataset.head())
    st.write(" Encoding the Dataset")
    st.write(x.head(15))
    st.write("## Splitting Data into Test and Trainning Set")
    X_train.shape , X_test.shape



elif(option == "Regression Model"):
    selectbox = st.sidebar.radio(label="", options=["Multiple Linear Regression","Support Vector Regression",
            "Decision Tree Regressor","Random Forest Regressor"])
    if(selectbox == "Multiple Linear Regression"):


        st.write("Multiple regression is an extension of simple linear regression. It is used when we want to predict the value of a variable based on the value of two or more other variables. The variable we want to predict is called the dependent variable (or sometimes, the outcome, target or criterion variable)")
        linear_y_pred = linear_regressor.predict(X_test)


        np.set_printoptions(precision=2)
```

```
        st.write("## Actual Vs Predicted Value")
        st.write("0 stands for actual value and 1 stands for predicted value")

st.write(np.concatenate((sc_y.inverse_transform(y_test).reshape(len(y_test),1),sc_y.inverse_transform(linear_y_pred).reshape(len(linear_y_pred),1)),axis=1))
        st.write("From the above observation we conclude that there is a huge difference between the actual and predicted value.")
        st.write('R^2 Score:', metrics.r2_score(y_test,linear_y_pred))
        st.write("The model score is 60% that means 60% of the data fit the regression model,since the r-squared score is not close to 1, so the model does not fit best")

    elif(selectbox == "Support Vector Regression"):
        #st.write("")
        st.write("Supervised Machine Learning Models with associated learning algorithms that analyze data for classification and regression analysis are known as Support Vector Regression. SVR is built based on the concept of Support Vector Machine or SVM.")

        np.set_printoptions(precision=2)
        st.write("## Actual Vs Predicted Value")
        st.write("0 stands for actual value and 1 stands for predicted value")

st.write(np.concatenate((sc_y.inverse_transform(y_test).reshape(len(y_test),1),sc_y.inverse_transform(y_pred_svr).reshape(len(y_pred_svr),1)),axis=1))
        st.write("From the above observation we conclude that there is a huge difference between the actual and predicted value.")
        st.write('R^2 Score:', metrics.r2_score(y_test,y_pred_svr))
        st.write("The model score is 50% that means only 50% of the data fit the regression mode, since the r-squared score is not close to 1, so the model does not fit best")

    elif(selectbox == "Decision Tree Regressor"):

        tree_regressor = DecisionTreeRegressor(random_state = 1)
        tree_regressor.fit(X_train, y_train1)
        y_pred_tree = tree_regressor.predict(X_test)
```

```
        np.set_printoptions(precision=2)

        st.write("## Actual Vs Predicted Value")

        st.write("0 stands for actual value and 1 stands for predicted value")


st.write(np.concatenate((sc_y.inverse_transform(y_test).reshape(len(y_test),1),sc_y.inverse_transfo
rm(y_pred_tree).reshape(len(y_pred_tree),1)),axis=1))

        st.write("From the above observation we conclude that there is a huge difference between the
actual and predicted value.")

        #st.write('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred_tree))

        st.write('R^2 Score:', metrics.r2_score(y_test,y_pred_tree))

        st.write("The model score is 40% that means only 40% of the data fit the regression model, the
model score is less than 50%,since the r-squared score is not close to 1, so the model does not fit
best ")

    elif(selectbox == "Random Forest Regressor"):


        st.write("Random Forest Regression is a supervised learning algorithm that uses ensemble
learning method for regression. A Random Forest operates by constructing several decision trees
during training time and outputting the mean of the classes as the prediction of all the trees.")

        np.set_printoptions(precision=2)

        st.write("## Actual Vs Predicted Value")

        st.write("0 stands for actual value and 1 stands for predicted value")


st.write(np.concatenate((sc_y.inverse_transform(y_test).reshape(len(y_test),1),sc_y.inverse_transfo
rm(y_pred_forest).reshape(len(y_pred_forest),1)),axis=1))

        st.write("From the above observation we conclude that there is a less difference between the
actual and predicted value.")

        #st.write('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred_forest))

        st.write('R^2 Score:', metrics.r2_score(y_test,y_pred_forest))

        st.write("The model score is 75% that means 75% of the data fit the regression model, since the
r-squared score is close to 1 the model fits best")

elif(option == "Classification Model"):

    radio = st.sidebar.radio(label = "", options = ["Data Transformation","Random Forest Classifier","K
Nearest Neighbors Classifier"])

    if(radio == "Data Transformation"):

        st.write("## Dataset")

        st.write(dataset1.head(3))
```

```
        st.write("## Target Variable")
        st.write(target_var.head())
        st.write("## Performaing one hot encodimg on age column")
        st.write(dummies_age.head())
        st.write("## Performing one hot encoding on income column")
        st.write(dummies_income.head())
        st.write("## Concatenationg the columns")
        st.write(target_dummies)
        st.write("## Label Encoding on Gender Column")
        st.write(target_var)
        st.write("## Displaying the final Dataset")
        target_var
        st.write("## Label Encoding on House Loan Column")
        st.write(y_c)
    y = dataset1.iloc[:,24].values
    if(radio == "Random Forest Classifier"):
        st.write("# Random Forest Classifier")
        st.write("Random forest, like its name implies, consists of a large number of individual decision
trees that operate as an ensemble. Each individual tree in the random forest spits out a class
prediction and the class with the most votes becomes our model’s prediction")
        st.write("## Accuracy Score")
        st.write(accuracy_score(y_test_c,y_predicted_random))


        st.write("## Classification Report")
        st.write(classification_report(y_test_c,y_predicted_random))


        st.write("## Heat Map to show Confusion Matirx")
        fig = plt.figure(figsize=(8,5))
        sns.heatmap(cm, annot=True)
        plt.xlabel('Predicted')
        plt.ylabel('Truth')
        st.pyplot(fig)
```

```
    if(radio == "K Nearest Neighbors Classifier"):
        st.write("# K Nearest Neighbors Classifier")
        st.write("K-Nearest Neighbors (KNN) is one of the simplest algorithms used in Machine Learning for regression and classification problem. KNN algorithms use data and classify new data points based on similarity measures (e.g. distance function). Classification is done by a majority vote to its neighbors")
        st.write("## Accuracy Score")
        st.write(accuracy_score(y_test_c, knn_pred))

        st.write("## Classification Report")
        st.write(classification_report(y_test_c,knn_pred))

        st.write("## Heat Map to show Confusion Matirx")
        fig = plt.figure(figsize=(8,4))
        sns.heatmap(confusion_matrix(y_test_c,knn_pred), annot=True)
        plt.xlabel('Predicted')
        plt.ylabel('Truth')
        st.pyplot(fig)

        st.write("## Choosing the K value")
        st.write("K Value = 23")
        fig = plt.figure(figsize=(10,6))
        plt.plot(range(1,40),knn_error_rate,color='blue', linestyle='dashed', marker='o',
                 markerfacecolor='red', markersize=10)
        plt.title('Error Rate vs. K Value')
        plt.xlabel('K')
        plt.ylabel('Error Rate')
        st.pyplot(fig)

        st.write("## Checking the accuracy of K = 23")
        fig = plt.figure(figsize=(10,6))
        plt.plot(range(1,40),knn_accuracy_rate,color='blue', linestyle='dashed', marker='o',
                 markerfacecolor='red', markersize=10)
        plt.title('Accuracy vs. K Value')
```

```
        plt.xlabel('K')
        plt.ylabel('Accuracy Rate')
        st.pyplot(fig)

elif (option == "Prediction Model"):
    st.write("## Location in Mumbai Western Line")
    with st.form(key='prediction_form'):
        locations = ["Marine Lines","Churchgate","Charni Road","Grant Road","Mumbai
Central","Mahalaxmi","Lower Parel",
                                "Prabhadevi","Dadar","Matunga Road","Mahim
Junction","Bandra","Khar Road","Santacruz","Vile Parle","Andheri","Jogeshwari","Ram Mandir",
                                "Goregaon","Malad","Borivali","Dahisar","Mira
Road","Bhayandar","Naigaon","Vasai Road","Nallasopara","Virar"]
        locations.sort()
        location = st.selectbox(label="", options=locations)
        st.write("## House Type")
        house_located = st.selectbox(label="", options=["Building","Bunglow","Chawl","Row House"])

        st.write("## Carpet Area")
        carpet_area = st.text_input(label = "Enter Carpet Area in Square Fit")
        st.write("## No of Bedrooms")
        No_of_Bedrooms = st.radio(label = "Select number of Rooms", options=["1 RK","1 BHK","2
BHK","3 BHK","4 BHK"])

        st.write("## Facilities")
        c_Car_Parking = st.checkbox("Car Parking")
        c_24hr_Water_Supply = st.checkbox("24hr Water Supply")
        c_Maintenance_Staff = st.checkbox("Maintenance Staff")
        c_Gas_Pipeline = st.checkbox("Gas Pipeline")
        c_Lift = st.checkbox("lift")
        c_Club_house = st.checkbox("Club house")
        c_24Hr_Security = st.checkbox("24Hr Security")
        c_Gym = st.checkbox("Gym")
```

```
        st.write("## Area Facility")
        c_School = st.checkbox("School")
        c_College = st.checkbox("College")
        c_Hospitals = st.checkbox("Hospital")
        c_Medical = st.checkbox("Medical")
        c_Railway_Station = st.checkbox("Railway Station")
        c_Market = st.checkbox("Market")
        submit = st.form_submit_button('Predict Price')
    if(submit):
        try:
            if(not carpet_area):
                st.write("Please enter the carpet area")
            elif(int(carpet_area)>=100 and int(carpet_area)<=6000):
                predicted_price = predict_price(location,house_located,carpet_area,No_of_Bedrooms,
c_Car_Parking,c_24hr_Water_Supply,
                c_Maintenance_Staff, c_Gas_Pipeline, c_Lift, c_Club_house,
                c_24Hr_Security, c_Gym, c_School, c_College, c_Hospitals,
                c_Medical, c_Railway_Station, c_Market)

                price_in_words = num2words(predicted_price, to='currency', lang='en_IN')

                st.write("The cost of the house will be ",predicted_price," that is
"+price_in_words.title().replace("Euro", "Rupees").replace("Cents", "Paise"))
                st.write('The accurace of the model is ',
round(metrics.r2_score(y_test,y_pred_forest)*100),"%")
            else:
                st.write("Please enter a valid carpet area")
        except:
            st.write("Please enter a valid carpet area")

elif(option == "House Loan Prediction"):
    with st.form(key='classification_form'):
```

```
    age_grp_list = dummies_age.columns.values
    age_grp_list = np.append(age_grp_list,'56 - 65')

    income_list = dummies_income.columns.values
    income_list = np.append(income_list,'11lakh and above')

    st.write("## Age Group")
    age = st.selectbox(label="",options=age_grp_list)

    st.write('## Income Range')
    salary = st.selectbox(label='', options=income_list)

    st.write("## Gender")
    check_gender = st.radio(label="", options=["Male","Female"])

    submit = st.form_submit_button('Predict House Loan')
if(submit):
  pred = predict_knn(age,salary,check_gender)
  if(pred == 1):
     st.write("The Buyer Will Take A Loan")
  else:
     st.write("The Buyer Will Not Take A Loan")
```

12 REFERENCES AND BIBLIOGRAPHY

- https://scikit-learn.org/0.21/documentation.html
- https://pandas.pydata.org/docs/
- https://seaborn.pydata.org/introduction.html
- https://altair-viz.github.io/
- https://docs.streamlit.io/en/stable/
- https://matplotlib.org/stable/contents.html
- https://www.youtube.com/channel/UCNU_lfiiWBdtULKOw6X0Dig
- https://www.youtube.com/channel/UCh9nVJoWXmFb7sLApWGcLPQ