PREDICTING REAL ESTATE PRICE: A COMPARATIVE ANALYSIS OF LINEAR REGRESSION AND SUPPORT VECTOR MACHINE

A MINOR PROJECT REPORT

Submitted by

D.KARTHIK REDDY RA2011003020130

A.SAI VIKAS RA2011003020124

B.VISHNU SAI RA2011003020128

Under the guidance of

Mrs.M.S.BENNET PRABA

(Assistant Professor, Department of Computer Science and Engineering)

in partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

RAMAPURAM,CHENNAI - 600089

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University U/S 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled **"PREDICTING REAL ESTATE PRICE: A COMPARATIVE ANALYSIS OF LINEAR REGRESSION AND SUPPORT VECTOR MACHINE"** is the bonafide work of **D.KARTHIK REDDY [REG NO: RA2011003020130], A.SAI VIKAS [REG NO: RA2011003020124], B.VISHNU SAI [REG NO: RA2011003020128]** who carried out the project work under my supervision. No part of the report has been submitted for any degree, diploma, title or recognition before

SIGNATURE

Mrs.M.S.BENNET PRABA
Assistant Professor
Computer Science and Engineering,
SRM Institute of Science and
Technology,
Ramapuram, Chennai.

SIGNATURE

Dr.K.RAJA, M.E., Ph.D.,
HEAD OF THE DEPARTMENT
Computer Science and Engineering,
SRM Institute of Science and
Technology,
Ramapuram, Chennai.

Submitted for the Viva-Voce Examination held on ___________ at SRM Institute of Science and Technology, Ramapuram Campus, Chennai -600089.

INTERNAL EXAMINER 1

INTERNAL EXAMINER 2

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
RAMAPURAM, CHENNAI - 89

DECLARATION

We hereby declare that the entire work contained in this project report titled "**PREDICTING REAL ESTATE PRICE: A COMPARATIVE ANALYSIS OF LINEAR REGRESSION AND SUPPORT VECTOR MACHINE**" has been carried out by **D.KARTHIK REDDY** [REG NO: RA2011003020130], **A. SAI VIKAS** [REG NO: RA2011003020124], **B.VISHNU SAI** [REG NO: RA2011003020128] at SRM Institute of Science and Technology, Ramapuram Campus, Chennai- 600089, under the guidance of **Mrs.M.S.BENNET PRABA , Assistant Professor** , Department of Computer Science and Engineering.

Place : Chennai

Date :

D.Karthik Reddy

A.Sai Vikas

B.Vishnu Sai

ABSTRACT

The market for residential and commercial property is a dynamic and complex industry because the valuations of properties fluctuate depending on a variety of variables. Both buyers and sellers should prioritize accurate price forecasts. This work focuses on applying machine learning techniques to provide end users with a useful way to evaluate real estate according to their specific needs and specific locations. Data collection is the first step in the process, after which null values and unnecessary columns are removed using data cleaning techniques. The prediction model used in this study is a linear regression model. An algorithm can correctly predict assets based on a given data set through extensive training. This step is necessary to ensure accuracy. With this visual data analysis, users can understand how real estate prices are changing to help them make smart investment choices. This user interface gives users quick access to predictive model results and a visual breakdown of real estate price changes. People from different fields can use it to make decisions in the real estate market because it is easily accessible to everyone. Using the Kaggle dataset as a starting point, this research work provides an overview of the application of machine learning algorithms to real estate price forecasting. We want to provide consumers with accurate and useful forecasts, so we clean the data, remove outliers, and use a linear regression model and support vector machine (SVM).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER - 1

INTRODUCTION

1.1 INTRODUCTION

1.1.1 PROBLEM STATEMENT

The aim of this project is to develop a machine learning model that accurately predicts real estate property prices based on a set of relevant features or parameters. The real estate market is a complex and dynamic environment where property prices are influenced by a multitude of factors, including location, property size, number of bedrooms, amenities, economic conditions, and more. Accurate prediction of real estate prices can assist buyers, sellers, and investors in making informed decisions.

Given historical data on property attributes and their corresponding sale prices, the goal is to create a predictive model that can generalize to new, unseen properties.Analyze the importance of different features and perform feature selection if needed. Additionally, create new relevant features that might enhance the predictive power of the model

1.2 AIM OF THE PROJECT

Project's main aim is to develop a data model capable of predicting house prices based on user-provided parameters. To achieve this, we will train the linear regression model using a dataset that captures various attributes of houses. Users will input specific criteria, and this model will utilize its learning to estimate the price of a house that aligns with those criteria.

In addition to price prediction, this project emphasizes the importance of informed decision-making when purchasing a house. The plan incorporates data visualization techniques to present users with various graphical representations that illustrate how different parameters, such as location, size, and features, impact house prices. These visual insights will empower users to make more informed and well-reasoned decisions when navigating the real estate market.

By combining predictive modeling with data visualization, this project aims to provide a comprehensive solution for both price estimation and enhanced decision support, ultimately assisting users in finding their ideal homes.

1.3 PROJECT DOMAIN

The real estate and property markets are normally the focus of a linear regression-based real estate price forecast project. In this field, one statistical and machine learning technique that may be used to forecast property prices based on a variety of variables or factors is linear regression. An outline of a project in this field can be found here. In terms of metrics and performance, a linear regression model is contrasted with a support vector machine model.

1.4 SCOPE OF THE PROJECT

Ability to estimate house prices can be useful to both the developer and the customer in planning the best time to buy a home.The Sqft, BHK, Location are three main factors that affect the prices of the houses. Additionally, this model will also help by giving accurate predictions when compared with SVM model, for them to set the pricing and save them from a lot of hassle and save a lot of precious time and money. Correct real estate prices are the essence of the market and want to ensure that by using this model

1.5 METHODOLOGY

Data Collection and Preparation: Project commences with the collection of a comprehensive dataset containing information about various houses, encompassing attributes such as location, size, amenities, and historical sale prices. The first step in data preparation is data cleaning, which involves addressing issues like missing values, data formatting, and, significantly, the removal of outliers. Outliers, which can adversely affect model accuracy, are identified and carefully eliminated to enhance the overall dataset quality.

Linear Regression Model Development: Following meticulous data preparation, we advance to the development of a linear regression model. Linear regression is a supervised learning technique that aims to create a robust predictive model. This model is trained using the refined dataset as the training set, with house attributes as features and sale prices as the target variable. The goal is to establish a linear relationship between the features and prices, allowing the model to accurately estimate house prices based on user-provided parameters. The model undergoes fine-tuning for optimal performance and predictive accuracy, with the added benefit of improved resilience against outliers.

Data Visualization and User Interface: The user-friendly interface empowers users to input their criteria for house selection. The machine learning model incorporates outlier-resistant features, ensuring that predictions are reliable and free from undue influence. In addition, we integrate data visualization tools to generate various graphs and charts, facilitating data-driven decision-making by showcasing the relationships between parameters and house prices.

CHAPTER - 2

LITERATURE REVIEW

The application of a machine learning technique called Gradient Boosting (GB) to examine the real estate market using a range of institutions and indicators that have an impact on real estate trends It highlights how crucial assessment measures are for determining how well a model performs, particularly in terms of mean absolute error (MAE) and root mean square error (RMSE). It explains why, in contrast to the notion that a high RMSE value is preferable, a lower RMSE value denotes a more accurate model. These methods are helpful for modeling and forecasting market trends and real estate values.[1]

These nonlinear and nonparametric models can be used to estimate the value of a property at different points in time. ML is used to create real estate price indexes and predict measurement errors in price changes. This model has higher predictive accuracy compared to the linear model. However, the data output by ML algorithms is less reliable and smaller in size due to their high dependence on calibration data [2]. By demonstrating various methods based on local features for evaluating real estate prices in France, they contrasted two machine learning systems. Geocoding enhances the input of ML models with accurate location-specific features. [3]. The pricing and evaluating assets like office buildings, retail establishments, and industrial facilities, the use of machine learning (ML) algorithms and resources such as the NCREIF real estate index determines the degree of accuracy and bias in commercial real estate appraisals. By using MAPE and MPE for the best price prediction, manual valuation forecasts sales prices, assisting in the selection of the most accurate estimate for on-the-spot transactions.[4]. To determine the best model fit, algorithms, machine learning models, and linear hedonic techniques all use a totally data-driven methodology and use probabilistic criteria. The dataset used in this research was given by the National Council of Investment in Real Property Fiduciaries. Accuracy can be raised and structural bias of commercial assessment values can be decreased by fitting a single regression tree with a boosting strategy. [5]

Forecasting real estate rents in housing market analysis plays an important role in calculating the yield index. In our system, we analyze two ML algorithms for rent prediction like linear regression and SVM,which are used for high efficiency and scalability.

This process strategy chooses various features for predictive models based on uninterested learning algorithms, which leads to reduced prediction errors and more accuracy than lazy learning techniques. The accuracy of a predictive model across test and training data sets is estimated using benchmark evaluations for regression analysis and model evaluation, namely MAE and R-squared error [6].

To evaluate how macroeconomic uncertainty affected house price prediction, they used random forest-based machine learning and Bayesian dynamic factor modeling. In comparison to negative projections, this allowed them to assess the impact of predictor factors and nonlinearities on improving forecast accuracy, with macroeconomic uncertainty playing a role in this improvement.[7]

This study frequently focuses on machine learning techniques in real estate price prediction. Decision trees explore both classification and regression, whereas linear regression uses input features from the model to predict causal relationships between variables. Homogeneous industries are determined using his 2018 collection of UK property prices for property types with 17 characteristics. This is used to highlight issues and predict costs in real estate modeling decisions. To estimate the actual mean, Random Forest queries the provided data and calculates the mean, which is the average of all the means [8].The ability to modify pricing models, such as procedures, to achieve changes in trend estimates over time and attain a specific level of accuracy compared to the initial model, is what really makes real estate pricing models based on data mining and machine learning useful. It's about being flexible. They also use volatility parameters and average recovery rates that vary over time using quadratic exponential smoothing [9]. The prediction of real estate prices is done using linear regression and SVM.

The data contains 11 different parameters like area type, location, availability, BHK size, society, total sq ft, price, latitude, longitude, number of bathrooms, and balcony. To get accurate pricing, compare two ML algorithms, such as linear regression and SVM, and compare these two algorithms based on their accuracy scores. investigated this data and used Plotly to present the data in the form of graphs. One of the most important regression algorithms, multiple linear regression, models a linear connection between a single continuous dependent variable and multiple independent variables.

Multiple predictor variables are required to predict the response variable. SVM is used for linearly separable data sets that can be divided into two classes using a single straight line. The data was predicted accurately. The training and test data values can be used to evaluate the performance of the ML algorithm in adapting to the input data by evaluating metrics such as RMSE. A value less than zero means the ML algorithm cannot fit the data

CHAPTER - 3

PROJECT DESCRIPTION

3.1 EXISTING SYSTEM

Market segmentation techniques divide the housing market into discrete submarkets as a means of addressing spatial variation. Submarkets are physical regions or noncontiguous collections of homes with comparable attributes and/or hedonic pricing. Price estimates for the entire market are done either worldwide or individually for each submarket by adding spatial indicators, including dummy variables for submarkets. The goal is to split the market in a way that makes accurate house value estimations possible, rather than necessarily defining relatively homogeneous submarkets made up of interchangeable homes. The input explanatory variables always play a major role with regard to the relevance of automated parametric or nonparametric models. Several types of explanatory variables are commonly used to estimate the prices of properties, such as the following: physical characteristic variables (e.g., living area, number of rooms), accessibility variables (e.g., proximity to amenities such as schools), neighborhood socioeconomic variables (e.g., local unemployment rates), and environmental variables. In metric values, the linear regression model has a low r2 score and mean absolute error values.

3.2 PROPOSED SYSTEM

Our proposed research aims to develop an accurate real estate price prediction system that takes into account user-specified characteristics such as square footage, number of bedrooms, and presence or absence of a balcony. Our approach provides more accurate predictions compared to existing models by carefully addressing data quality issues such as removing null values and outliers from a data set containing 13,000 data points. We created a data model for the project based on the principles of linear regression and SVM. This model uses a clustering strategy to improve prediction accuracy. The slope values generated from these clusters can be used to determine the price. Clusters are formed based on the range of input parameters.

This new method takes into account changes in housing costs across multiple clusters, allowing for more precise calculations. For classification tasks, SVM is a powerful machine learning technique. It can be applied as a regression model to estimate real estate value and determine the cost of a home depending on factors such as square footage, number of bedrooms, and location. SVM is effective at managing complex,high-dimensional data and nonlinear relationships. In this case, price ranges act as separate classes, and the goal is to create a hyperplane that maximizes the margin between these data points while minimizing prediction error. SVM is especially useful when there are outliers or when the relationship between the features and the target variable is nonlinear. Considering that linear regression is a simple but common method for estimating real estate prices, its use is logical. When calculating property values using data, it is assumed that there is a linear relationship between the input characteristics and the intended conclusion. The objective of linear regression is to find the straight line (or linear equation) that minimizes the sum of the squares of the difference between the predicted price and the actual price. Although it may not be as effective as SVM in capturing complex nonlinear relationships, it is transparent and easy to understand, making it a useful tool for simple price estimation tasks. Depending on your data type and the specific requirements of your task, you can use linear regression or SVM to predict real estate prices.

SVM is more adaptive and can capture nonlinear correlations, making it more suitable for complex datasets with advanced feature interactions. In contrast, linear regression is easier to use, easier to read, and generally has a higher success rate when there is a fairly obvious linear relationship between the features and the target variable. Additionally, overfitting is less likely to occur with small datasets. Therefore, if your data has a predominantly linear pattern, linear regression can be a faster and more efficient solution. Linear regression is expected to be more accurate than SVM in predicting real estate prices because the relationship between input factors and target variables mostly follows a linear pattern. In such cases, linear regression provides a clearer and easier to understand solution, making it easier-to-understand which features influence the price estimate and how changes in those features affect the results. Become. Additionally, linear regression typically has lower processing costs than SVM and is less prone to overfitting on small data sets, both of which are beneficial in real estate applications where data may be limited. Although SVM is better at managing complex nonlinear interactions when the data follows linear assumptions, linear regression is the preferred method due to its simplicity and effectiveness.

Section 3.1 is about linear regression. Section 3.2 explains SVM.

3.2.1 Linear regression

A linear relationship between one or more independent variables (x) and a dependent variable (y) is represented by the linear regression process (also known as linear regression). Because this is a linear relationship, you can use linear regression to determine how the value of the dependent variable changes around the value of the independent variable. The main purpose of using linear regression is to find the best-fitting straight line, defined as the error between predicted and actual values,and this best-fitting straight line will have the least error[3]

There are two types of linear regression models which are simple and multiple linear regression models. We are using a simple linear regression model due to its higher efficiency for a smaller dataset. Multiple linear regression is more suitable for high scalability,which will be added in future work.

The price prediction formula for a single feature (such as square footage) is expressed equ(1) as:

price = $\alpha_0 + \alpha_1 * Square\ Feet + \varepsilon$ —-Equ(1)

where Price is the predicted house price.

- Square footage is the square footage of the property.
- α_0 is the intercept.
- α_1 is the coefficient (slope) representing the relationship between square footage and price.
- ε represents the error term

Pseudo code of linear regression

```
1. start
2. Read the total dataset(n)
3. Initialize all variables as zero
4. For a=1 to n:
     read Xa=total sq ft and Ya=prices
     next a
5. For a=1 to n:
        totalX+=Xa
      // the sum of all total sq ft values
```

```
            totalX2+=Xa*Xa
            // the sum of the squares of total sqft values
            totalY+=Ya
            // the sum of all prices of flats
            totalXY+=Xa*Ya
            // the sum of the product of Xa and Ya
            next a
6.  Constants b and c
          y=b+mx
          m=(n*totalXY-totalX*totalY)/
             (n*totalX2-totalX*totalX)
         b=(totalY-m*totalX)/n
7.  Shows value b and c
```

It begins with the provision of an initial dataset containing essential parameters such as square footage (sq ft) and property prices. From these dataset parameters, the linear regression algorithm extracts valuable insights. The primary objective is to establish a cumulative linear line that represents the relationship between the two main variables: price and sqft.In the world of linear regression, this essential relationship is captured through a linear equation, commonly expressed as y = mx + b. Here, 'y' represents the predicted price, 'x' stands for sq ft, 'm' denotes the slope of the line, and 'b' signifies the y-intercept, the point where the line intersects the y-axis. The slope m' is a crucial component in this equation. It quantifies how the price changes concerning the sqft. When 'm' is positive, it indicates that as sq ft increases, the price tends to rise, reflecting a positive correlation. Conversely, when 'm' is negative, it suggests that as sqft increases, the price typically decreases, illustrating a negative correlation. The cumulative linear line, known as the regression line, embodies this linear equation and is a visual representation of the sq ft-price relationship. In this representation, the x-axis corresponds to price, while the y-axis represents sq ft. The regression line captures the overall trend and pattern within the dataset. By fitting this line optimally to the data points, the algorithm discerns the underlying relationship, enabling accurate predictions. Now, when new data points emerge, linear regression proves its predictive prowess. If we are given a property's sq ft value and aim to predict its price, we turn to the cumulative regression line. By placing the known sqft value on the x-axis, we can trace upward to where it intersects the regression line. The corresponding point on the y-axis provides the predicted price. This prediction is grounded in the observed trend and relationship within the dataset. In essence, linear regression offers a comprehensive approach to real estate price prediction. It leverages the dataset's sqft and price parameters to derive a linear equation that captures the relationship, allowing for the creation of the regression line. This line, defined by the slope 'm,' serves as a dependable tool for predicting property prices based on square foot values. It exemplifies the power of data-driven insights and mathematical modeling in making informed predictions in the real estate market.

3.2.2 Support vector machine

SVM is an effective supervised method that performs best on complex but smaller datasets. Although support vector machines, often known as SVMs, are useful for classification as well as regression applications, their performance is generally greatest in the former. The margin in SVM is calculated using the points that are closest to the hyperplane (support vectors); we don't need to worry about additional observations; in logistic regression, on the other hand, the classifier is defined over all of the points. As a result, SVM naturally speeds up. Important terms in SVM are support vectors and margins.

The points that are closest to the hyperplane are known as support vectors. These data points will be used to define a separation line. The distance between the hyperplane and the observations (support vectors) that are closest to it is known as the margin. A big margin is regarded as a good margin in SVM. Hard margin and soft margin are the two different categories of margins in SVM. SVMs are capable of handling classification issues that are either linear or non-linear since they employ various kernel functions to convert data into a higher-dimensional space. In order to guarantee that the decision boundary is well defined and less susceptible to the beginning conditions, the optimization problem in SVM seeks to identify the global optimal solution. Because support vectors nearest to the decision boundary are the focus of SVMs, they are less impacted by outliers in the dataset. Additionally, the theoretical foundation of statistical learning theory gives SVMs a strong theoretical foundation for comprehending their performance and generalization.

3.3 ADVANTAGES

The project, focused on developing a linear regression model for house price prediction with outlier removal and data visualization, offers several key benefits. It provides accurate price estimations, enhances resilience to outliers, and empowers users to make informed decisions through data visualization. The user-friendly interface promotes efficiency and transparency in real estate transactions, saving time and resources. Additionally, the project's versatility allows it to adapt to various property types and locations, making it a valuable tool for a wide range of real estate markets.

3.4 FEASIBILITY STUDY

One essential part of maintaining consistency is the evaluation of the literature. It's a crucial step in the development process that must be finished. Software development requires the resources to be accurate and readily available. This section helps to both identify the developed content and provide guidance on how to use it in the contemporary context. Development is primarily driven by two factors: the strength of the product and the economy. It's important to monitor and evaluate the flow of resources and assistance after innovation has reached the building stage. This is also known as the research phase because all of the research required to finish the flow is completed here.

3.5 SYSTEM SPECIFICATIONS

3.5.1 HARDWARE REQUIREMENTS

- Processor - Intel i5-8250 CPU @1.60GHz 1.80GHz
- 512GB SSD
- NVIDIA GEFORCE RTX
- CPU QUAD CORES 3.4.2

3.5.2 SOFTWARE REQUIREMENTS

- ANACONDA
- ANACONDA PROMPT
- PYTHON
- VISUAL STUDIO

CHAPTER - 4

MODULE DESCRIPTION

4.1 GENERAL ARCHITECTURE

There are four phases,as shown in the architecture diagram above. Inputs to the process are covered in the first phase of the architecture diagram. In this case, a dataset obtained from Kaggle was provided as input. Data preprocessing is described in the second phase of the architecture diagram. Here, we first remove null values and remove outliers to improve the accuracy of data collection. Process categorical data, engineer functions from datasets, perform visualizations using dataset parameters and information from various graphs, and process categorical data. Regression modeling is described in the third phase of the architecture diagram in fig. 1.

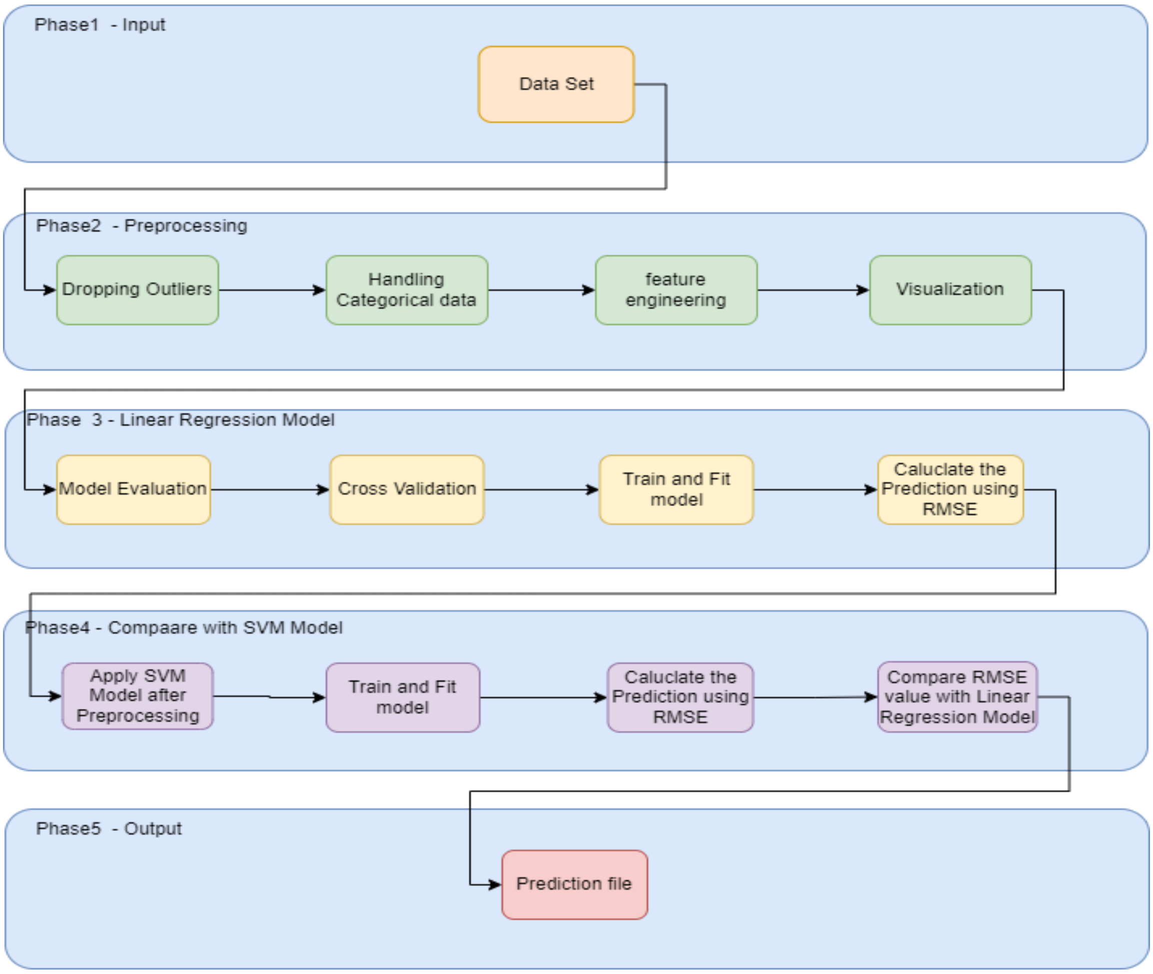

Phase1 - Input
Data Set
Phase2 - Preprocessing
Dropping Outliers
Handling Categorical data
feature engineering
Visualization
Phase 3 - Linear Regression Model
Model Evaluation
Cross Validation
Train and Fit model
Caluclate the Prediction using RMSE
Phase4 - Compaare with SVM Model
Apply SVM Model after Preprocessing
Train and Fit model
Caluclate the Prediction using RMSE
Compare RMSE value with Linear Regression Model
Phase5 - Output
Prediction file

Fig - 4.1 Architecture diagram

4.2 MODULE DESCRIPTION

4.2.1 Data Collection and Preprocessing:

- Collection of a dataset containing relevant features such as square footage, number of bedrooms, location, etc., and their corresponding house prices.
- Cleaning and preprocessing the data to handle missing values, outliers, and ensuring the data is in a suitable format for analysis.

4.2.2 Model Selection:

- Choosing an appropriate regression algorithm for the task, such as Linear Regression, Ridge Regression, Lasso Regression, or more advanced methods like Random Forest Regression, Gradient Boosting, or Neural Networks.

4.2.3 Training the Model:

- Splitting the dataset into training and testing sets to evaluate the model's performance.
- Training the chosen regression model using the training data.

4.2.4 Model Evaluation:

- Evaluating the model's performance using appropriate metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (coefficient of determination).
- Comparing the model's performance against a baseline model or other regression algorithm

4.2.5 Prediction and Interpretation:

- Using the trained model to predict house prices on new, unseen data.
- Interpreting the coefficients or feature importance to understand which factors have the most significant impact on house prices.

4.2.6 Visualization:

- Creating visualizations to illustrate the model's predictions compared to the actual prices.
- Plotting features importance to showcase the factors driving house prices.

4.2.7 Documentation and Reporting:

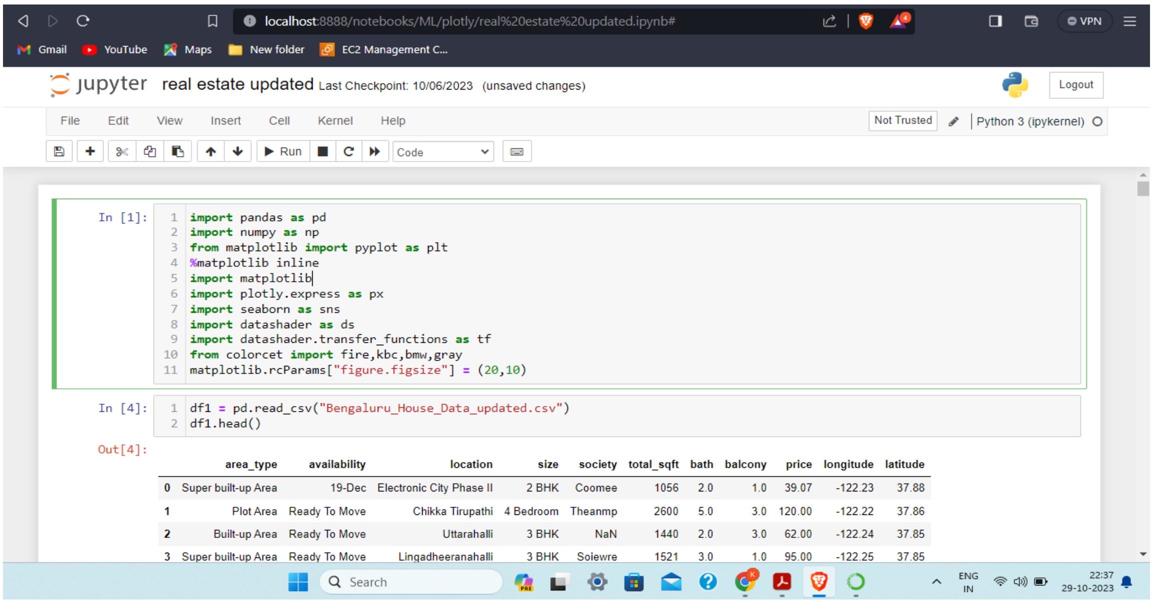
- Documenting the entire process, including data preprocessing steps, model selection criteria, hyperparameters used, and evaluation results.
- Summarizing the key findings and insights from the analysis.

CHAPTER - 5

IMPLEMENTATION AND TESTING

5.1 INPUT AND OUTPUT

5.1.1 Input:

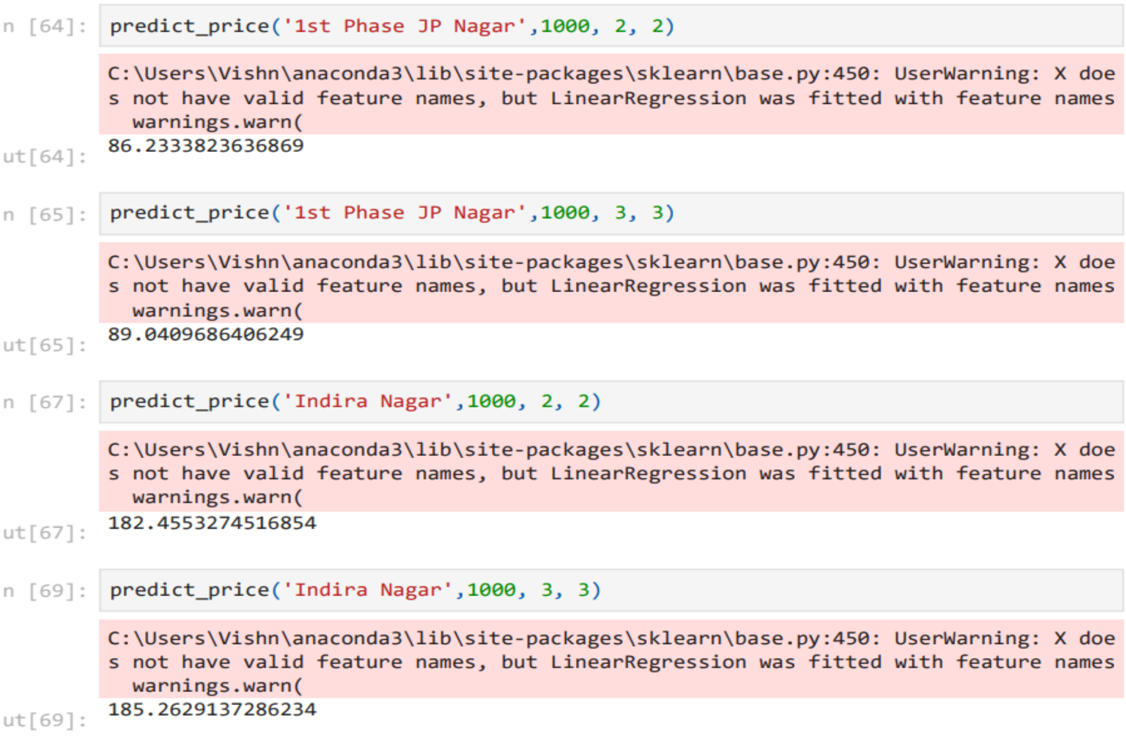


```
In [1]:  import pandas as pd
         import numpy as np
         from matplotlib import pyplot as plt
         %matplotlib inline
         import matplotlib
         import plotly.express as px
         import seaborn as sns
         import datashader as ds
         import datashader.transfer_functions as tf
         from colorcet import fire,kbc,bmw,gray
         matplotlib.rcParams["figure.figsize"] = (20,10)

In [4]:  df1 = pd.read_csv("Bengaluru_House_Data_updated.csv")
         df1.head()

Out[4]:
```

	area_type	availability	location	size	society	total_sqft	bath	balcony	price	longitude	latitude
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07	-122.23	37.88
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00	-122.22	37.86
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00	-122.24	37.85
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00	-122.25	37.85

Fig - 5.1 Input

5.1.2 Output:



```
n [64]:  predict_price('1st Phase JP Nagar',1000, 2, 2)

         C:\Users\Vishn\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X doe
         s not have valid feature names, but LinearRegression was fitted with feature names
           warnings.warn(
ut[64]:  86.2333823636869

n [65]:  predict_price('1st Phase JP Nagar',1000, 3, 3)

         C:\Users\Vishn\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X doe
         s not have valid feature names, but LinearRegression was fitted with feature names
           warnings.warn(
ut[65]:  89.0409686406249

n [67]:  predict_price('Indira Nagar',1000, 2, 2)

         C:\Users\Vishn\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X doe
         s not have valid feature names, but LinearRegression was fitted with feature names
           warnings.warn(
ut[67]:  182.4553274516854

n [69]:  predict_price('Indira Nagar',1000, 3, 3)

         C:\Users\Vishn\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X doe
         s not have valid feature names, but LinearRegression was fitted with feature names
           warnings.warn(
ut[69]:  185.2629137286234
```

5.2 TESTING

5.2.1 TYPES OF TESTING

Detecting mistakes is the aim of the testing. The process of testing a work product involves attempting to find every potential flaw or vulnerability. It offers guidance on how to view the functionality of individual parts, assemblies, subassemblies, and/or final products. It's a technique for testing software to make sure it satisfies user requirements and expectations and doesn't malfunction in an unacceptable way. Different types of exams exist. Every sort of test responds to a certain testing requirement. Below is a list of the several kinds of testing that will come next.

5.2.2 UNIT TESTING

The process of planning test cases for unit testing ensures that the internal logic of the program is operating correctly and that program inputs result in legitimate outputs. Validation should be done on all internal code flows and decision branches. It is the appliance's individual software modules being tested. Prior to integration, it is completed following the conclusion of a private unit. This is frequently an intrusive structural test that depends on understanding how it was built. Unit tests evaluate a chosen application, system configuration, or business process at the component level.

```
def plot_scatter_chart(df,location):
    bhk2 = df[(df.location==location) & (df.bhk==2)]
    bhk3 = df[(df.location==location) & (df.bhk==3)]
    matplotlib.rcParams['figure.figsize'] = (15,10)
    plt.scatter(bhk2.total_sqft,bhk2.price,color='blue',label='2 BHK', s=50)
    plt.scatter(bhk3.total_sqft,bhk3.price,marker='+',color='green',label='3 BHK', s=50)
    plt.xlabel("Total Square Feet Area")
    plt.ylabel("Price Per Square Feet")
    plt.title(location)
    plt.legend()
    #fig9=px.scatter(bhk2.total_sqft,bhk2.price,color=bhk2.total_sqft)
    #fig9=px.scatter(bhk3.total_sqft,bhk3.price,color=bhk3.total_sqft)
    #fig9.show()
```

```
plot_scatter_chart(df,"Hebbal")
```

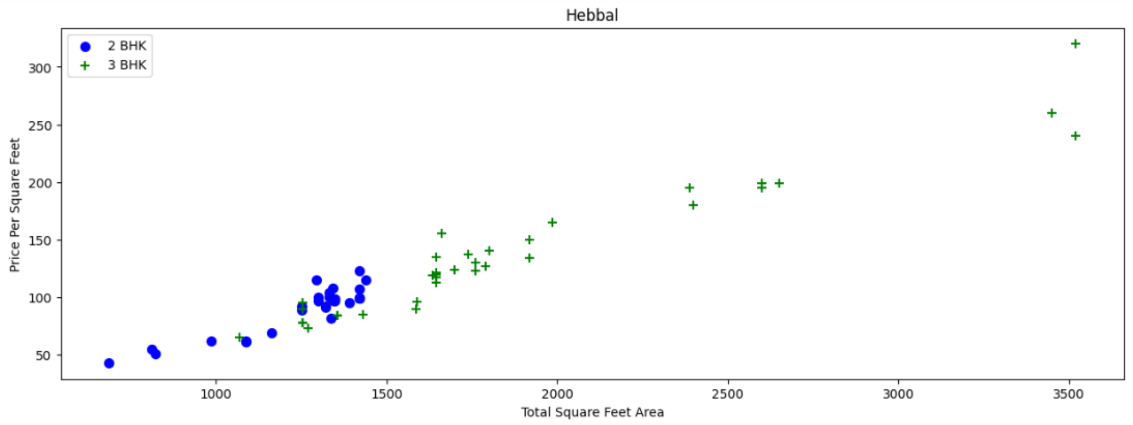


Fig - 5.2.2 Scatter Plot for 2bhk and 3 bhk

5.2.3 INTEGRATION TESTING

The purpose of integration tests is to verify that software components that are integrated actually function as a unit. The primary focus of event-driven testing is on the crucial results of fields or screens. Successful unit testing indicates that each component was individually gratifying, but integration tests show that the combination of components is accurate and consistent. The purpose of integration testing is to identify any problems that may result from the combination of components.

```
def remove_bhk_outliers(df):
    exclude_indicies = np.array([])
    for location, location_df in df.groupby('location'):
        bhk_stats = {}
        for bhk, bhk_df in location_df.groupby('bhk'):
            bhk_stats[bhk] = {
                'mean': np.mean(bhk_df.price_per_sqft),
                'std': np.std(bhk_df.price_per_sqft),
                'count': bhk_df.shape[0]
            }
        for bhk, bhk_df in location_df.groupby('bhk'):
            stats = bhk_stats.get(bhk-1)
            if stats and stats['count']>5:
                exclude_indices = np.append(exclude_indicies, bhk_df[bhk_df.price_per_sqft<(stats['mean'])].index.values)
    return df.drop(exclude_indices,axis='index')


df  = remove_bhk_outliers(df)
df.shape
```

5.2.4 VALIDATION TESTING

Validation testing is the process of ensuring that the tested and developed software satisfies the client. The business requirement logic or scenarios need to be tested intimately. All the critical functionalities of an application must be tested here.

```
from sklearn.preprocessing import LabelEncoder
```

```
# Assuming 'y' is your target variable with categorical labels

label_encoder = LabelEncoder()

y_encoded = label_encoder.fit_transform(y1)
```

5.2.5 SYSTEM TESTING

System testing, whether it be for hardware or software, evaluates how well an integrated system functions as a whole and whether its criteria are being met. System testing is a subset of recorder testing and should not necessitate an understanding of the inner workings of the code or logic. Typically, all "integrated" software components that have completed integration testing as well as the program itself combined with any suitable hardware system(s) are the inputs used by system testing. Testing that focuses on finding flaws in the "inter-assemblages" as well as the system as a whole may be more constrained than other types of testing. System testing is performed on the whole system within the context of a functional requirement specification (FRS) and/or a system requirement specification (SRS).

CHAPTER - 6

RESULTS AND DISCUSSIONS

6.1 EFFICIENCY OF THE PROPOSED SYSTEM

Model	Root mean square error (RMSE)	Mean absolute error (MAE)	R2 score
Linear Regression	36.43	22.50	0.68
Support Vector Machine	295.37	229.37	0.54

Table - 1

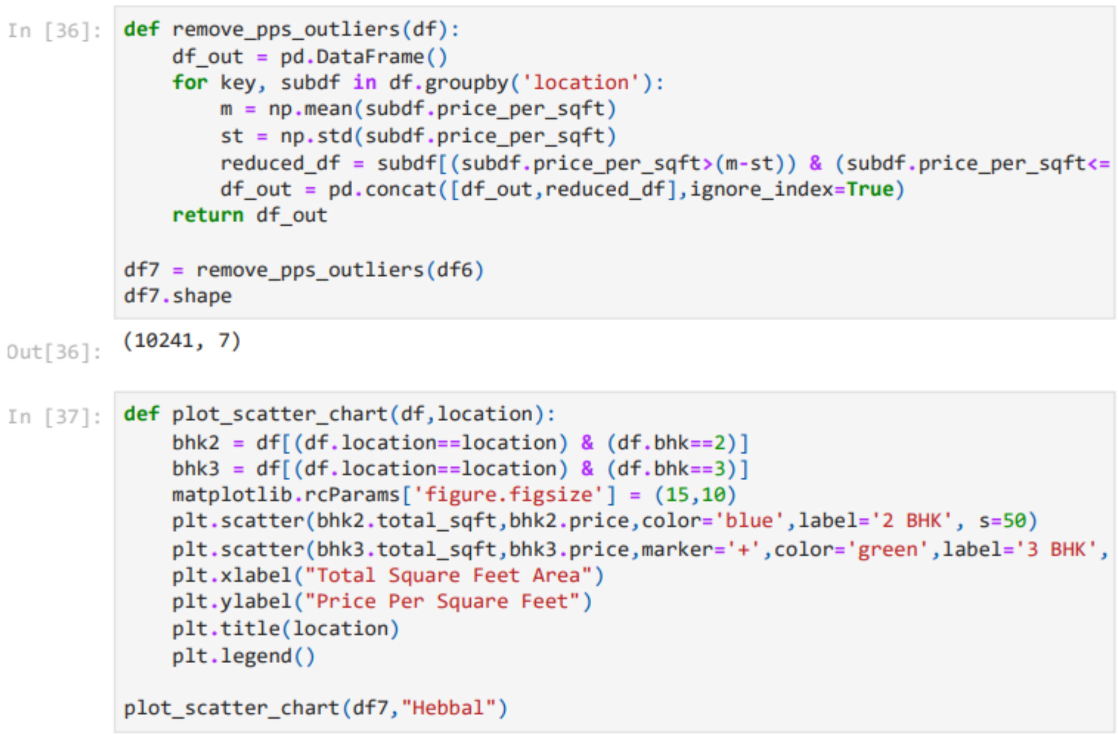
Table 1 shows the linear regression model has an RMSE of 36.43; it's performing better than the SVM model, which has 295.37, so linear regression has a good fit. Linear regression with an R-squared score of 0.68 means it accounts for 68% of the variance in the target variable, and SVM has a score of 0.54, meaning it accounts for 54% of the variance in the target variable, which is slightly weaker than the linear regression model. The SVM model with an MAE of 229.37, shows the predictions, and the linear regression model with 22.5. The linear regression model is performing better in terms of MAE because it has a much smaller MAE value.

6.2 COMPARISON OF EXISTING AND PROPOSED SYSTEMS

In the existing system, the support vector machine model has a higher efficiency than the linear regression model by using metric values. Their major limitation is the linearity assumption between the outcome variable and the explanatory variables. The SVM method suffers from a lack of transparency in its results; graphical visualizations can be used to facilitate the interpretation of the results. The performances of the models are different for each city. Except for the R2 metric, real estate price predictions are more accurate for cities with medium costs of living in terms of real estate prices (e.g., Toulouse, Montpellier, and Nantes) and are less accurate for cities with high costs of living (e.g., Paris, Bordeaux, and Nice).

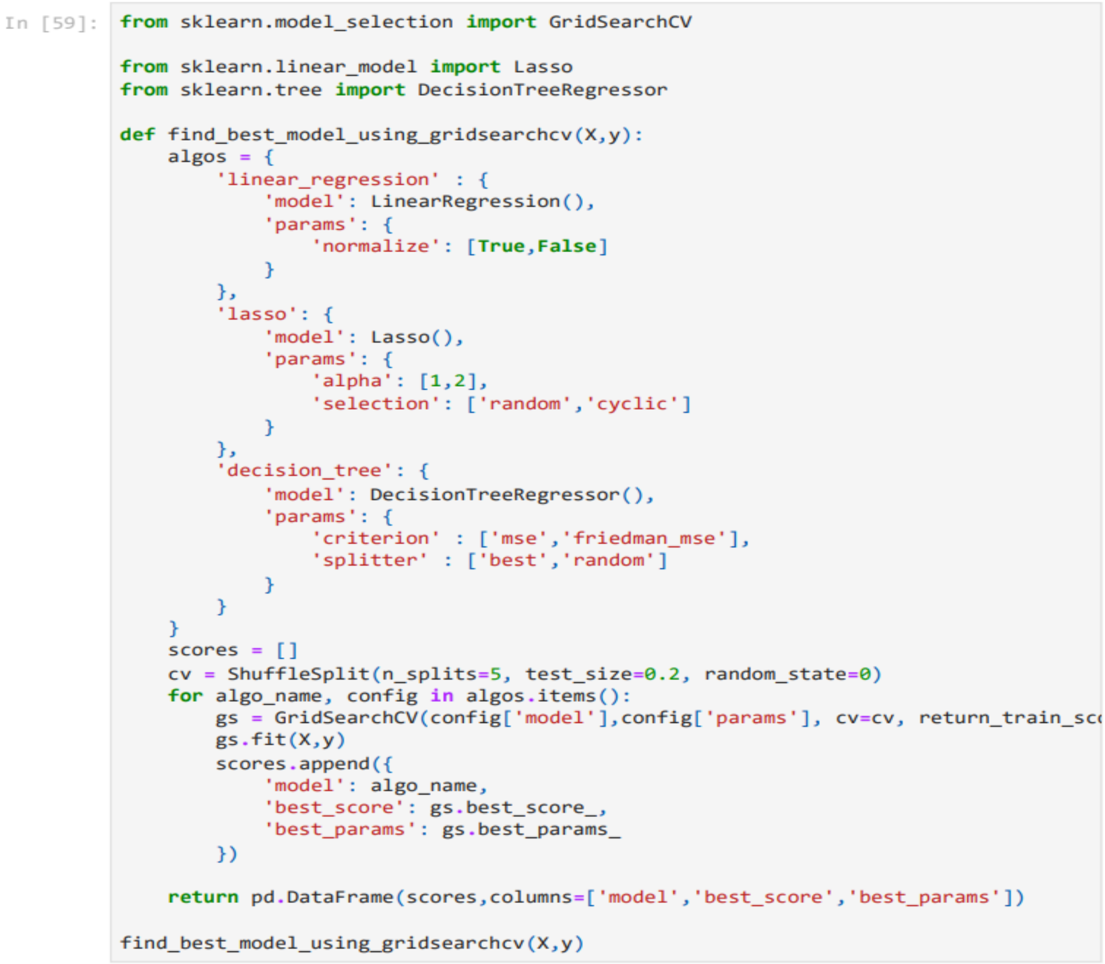
In the proposed system, the linear regression model shows a better fit than the support vector machine, and the linear regression model has low transparency in its results. Visualizations can be used to facilitate the interpretation of the results. The linear regression model predicts the prices accurately compared to SVM. SVM shows the prices inaccurately.

6.3 SAMPLE CODE



```
In [36]: def remove_pps_outliers(df):
             df_out = pd.DataFrame()
             for key, subdf in df.groupby('location'):
                 m = np.mean(subdf.price_per_sqft)
                 st = np.std(subdf.price_per_sqft)
                 reduced_df = subdf[(subdf.price_per_sqft>(m-st)) & (subdf.price_per_sqft<=
                 df_out = pd.concat([df_out,reduced_df],ignore_index=True)
             return df_out

         df7 = remove_pps_outliers(df6)
         df7.shape

Out[36]: (10241, 7)
```

```
In [37]: def plot_scatter_chart(df,location):
             bhk2 = df[(df.location==location) & (df.bhk==2)]
             bhk3 = df[(df.location==location) & (df.bhk==3)]
             matplotlib.rcParams['figure.figsize'] = (15,10)
             plt.scatter(bhk2.total_sqft,bhk2.price,color='blue',label='2 BHK', s=50)
             plt.scatter(bhk3.total_sqft,bhk3.price,marker='+',color='green',label='3 BHK',
             plt.xlabel("Total Square Feet Area")
             plt.ylabel("Price Per Square Feet")
             plt.title(location)
             plt.legend()

         plot_scatter_chart(df7,"Hebbal")
```



```
In [59]: from sklearn.model_selection import GridSearchCV

         from sklearn.linear_model import Lasso
         from sklearn.tree import DecisionTreeRegressor

         def find_best_model_using_gridsearchcv(X,y):
             algos = {
                 'linear_regression' : {
                     'model': LinearRegression(),
                     'params': {
                         'normalize': [True,False]
                     }
                 },
                 'lasso': {
                     'model': Lasso(),
                     'params': {
                         'alpha': [1,2],
                         'selection': ['random','cyclic']
                     }
                 },
                 'decision_tree': {
                     'model': DecisionTreeRegressor(),
                     'params': {
                         'criterion' : ['mse','friedman_mse'],
                         'splitter' : ['best','random']
                     }
                 }
             }
             scores = []
             cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
             for algo_name, config in algos.items():
                 gs = GridSearchCV(config['model'],config['params'], cv=cv, return_train_sc
                 gs.fit(X,y)
                 scores.append({
                     'model': algo_name,
                     'best_score': gs.best_score_,
                     'best_params': gs.best_params_
                 })

             return pd.DataFrame(scores,columns=['model','best_score','best_params'])

         find_best_model_using_gridsearchcv(X,y)
```

Fig 6.3 Sample Code

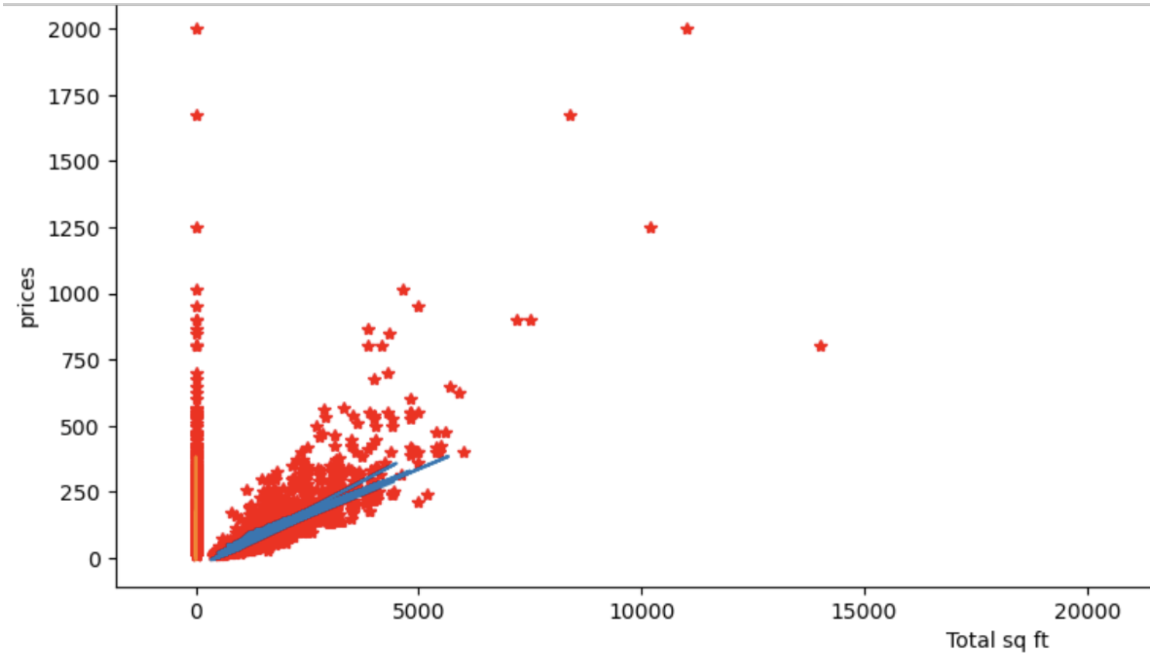


Fig 6.3.1 Performance Graph

CHAPTER - 7

CONCLUSION & FUTURE ENHANCEMENT

7.1 CONCLUSION

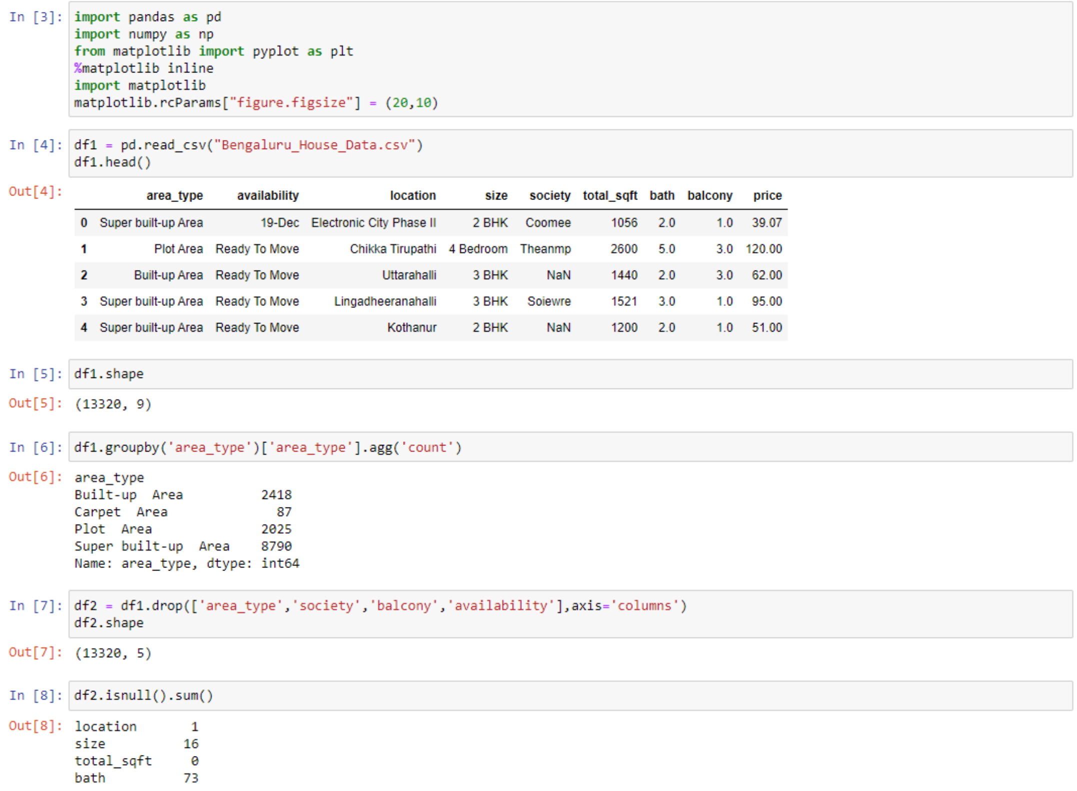
In this research, we investigate the use of machine learning models to predict real estate prices. In comparison to SVM, the R2_score and RMSE of the linear regression model performed better. SVM provides a strong fit for complicated data, whereas linear regression provides a good fit for small datasets. Both linear regression and SVM are used for the prediction of small and complex data. According to the trial findings, real estate outperformed all other machine learning prediction models with an R2_score measure of 0.68.

7.2 FUTURE ENHANCEMENT

For this real estate price prediction project, we're currently using a linear regression model, a straightforward method that estimates property prices based on features like bedrooms, square footage, and location. In the future, the aim is to enhance this project by incorporating more advanced techniques, such as ensemble methods, to capture complex patterns in the dataset in training the data model. This will also focus on improving user-friendliness and providing personalized property recommendations, with the help of graphs using data visualization tools, making it easier for users to make informed decisions when buying or selling real estate. This project will continue to evolve, aiming to deliver more accuracy when compared to the sum model and user-friendly predictions.

CHAPTER - 8
SOURCE CODE

8.1 SAMPLE CODE



In [3]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import matplotlib
matplotlib.rcParams["figure.figsize"] = (20,10)
```

In [4]:
```
df1 = pd.read_csv("Bengaluru_House_Data.csv")
df1.head()
```

Out[4]:

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00

In [5]:
```
df1.shape
```

Out[5]:
```
(13320, 9)
```

In [6]:
```
df1.groupby('area_type')['area_type'].agg('count')
```

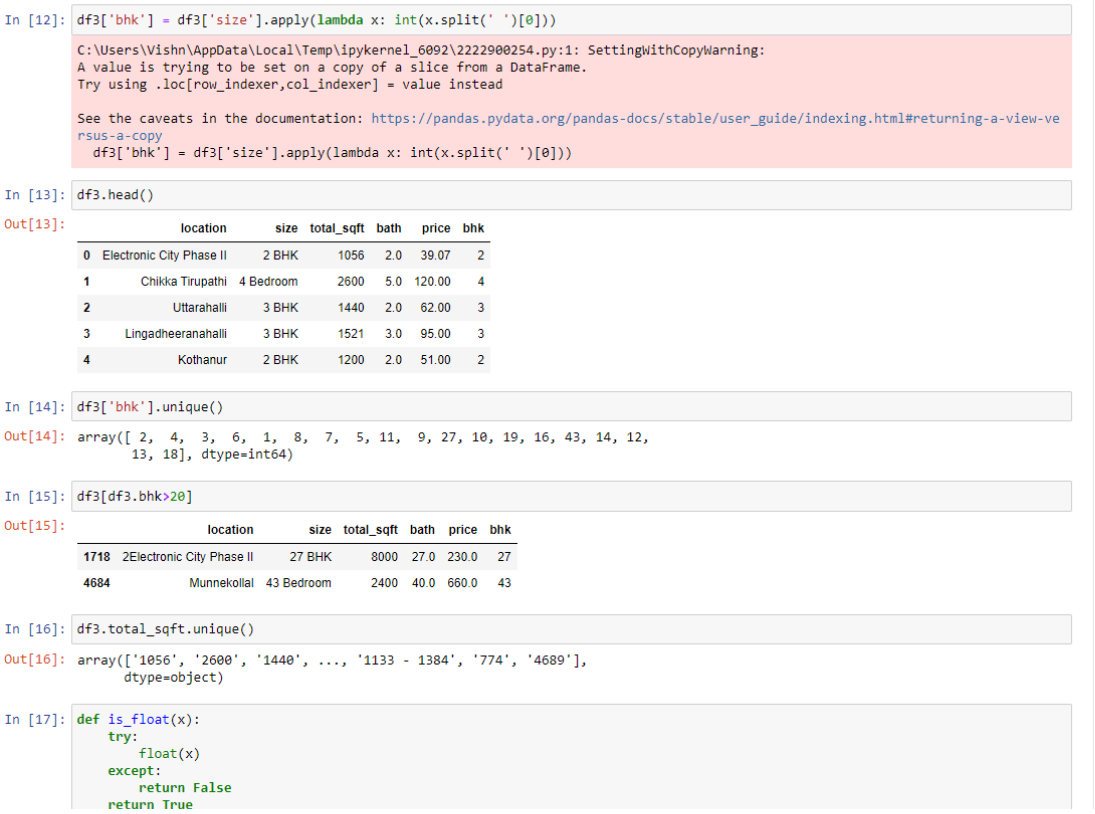
Out[6]:
```
area_type
Built-up  Area          2418
Carpet  Area              87
Plot  Area              2025
Super built-up  Area    8790
Name: area_type, dtype: int64
```

In [7]:
```
df2 = df1.drop(['area_type','society','balcony','availability'],axis='columns')
df2.shape
```

Out[7]:
```
(13320, 5)
```

In [8]:
```
df2.isnull().sum()
```

Out[8]:
```
location       1
size          16
total_sqft     0
bath          73
```



In [12]:
```
df3['bhk'] = df3['size'].apply(lambda x: int(x.split(' ')[0]))
```

```
C:\Users\Vishn\AppData\Local\Temp\ipykernel_6092\2222900254.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-ve
rsus-a-copy
  df3['bhk'] = df3['size'].apply(lambda x: int(x.split(' ')[0]))
```

In [13]:
```
df3.head()
```

Out[13]:

	location	size	total_sqft	bath	price	bhk
0	Electronic City Phase II	2 BHK	1056	2.0	39.07	2
1	Chikka Tirupathi	4 Bedroom	2600	5.0	120.00	4
2	Uttarahalli	3 BHK	1440	2.0	62.00	3
3	Lingadheeranahalli	3 BHK	1521	3.0	95.00	3
4	Kothanur	2 BHK	1200	2.0	51.00	2

In [14]:
```
df3['bhk'].unique()
```

Out[14]:
```
array([ 2,  4,  3,  6,  1,  8,  7,  5, 11,  9, 27, 10, 19, 16, 43, 14, 12,
       13, 18], dtype=int64)
```

In [15]:
```
df3[df3.bhk>20]
```

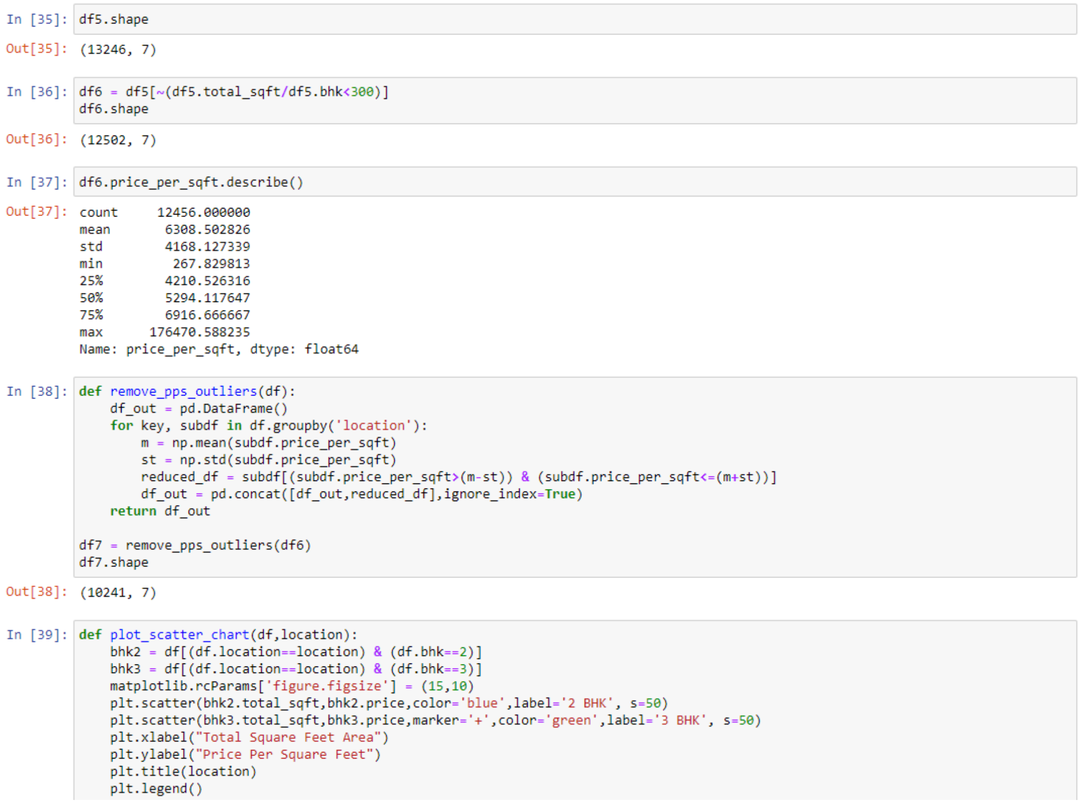
Out[15]:

	location	size	total_sqft	bath	price	bhk
1718	2Electronic City Phase II	27 BHK	8000	27.0	230.0	27
4684	Munnekollal	43 Bedroom	2400	40.0	660.0	43

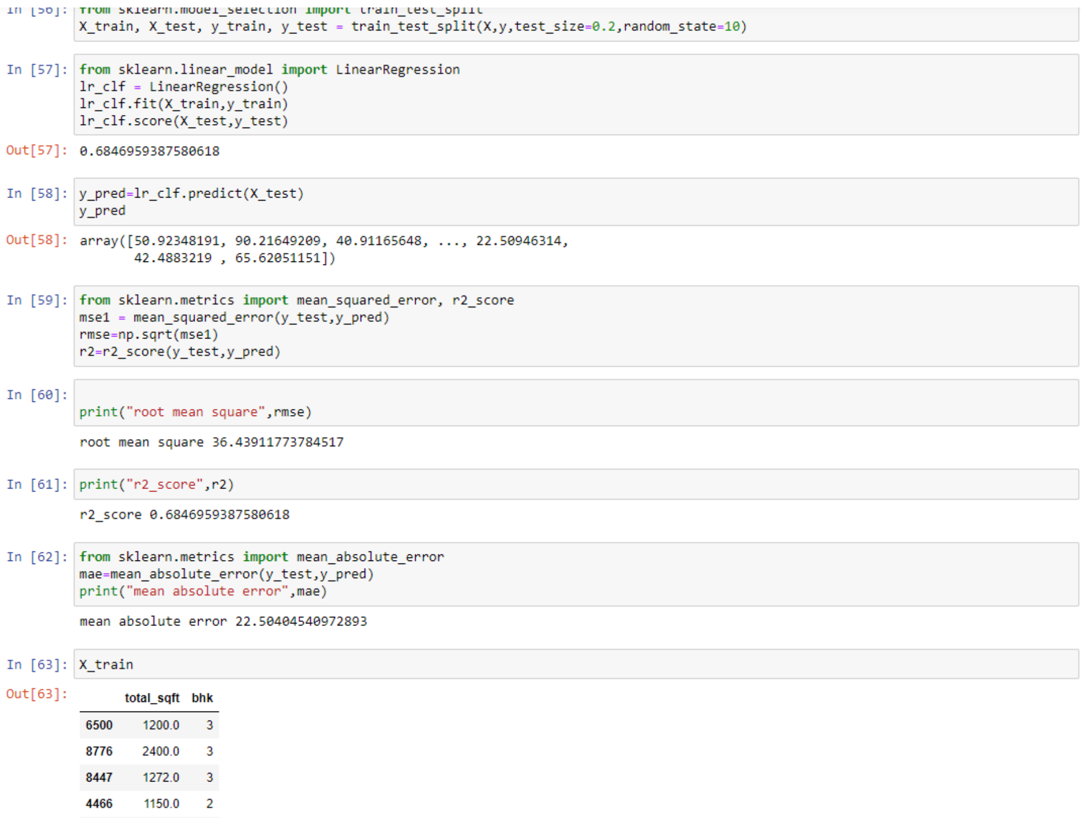
In [16]:
```
df3.total_sqft.unique()
```

Out[16]:
```
array(['1056', '2600', '1440', ..., '1133 - 1384', '774', '4689'],
      dtype=object)
```

In [17]:
```
def is_float(x):
    try:
        float(x)
    except:
        return False
    return True
```

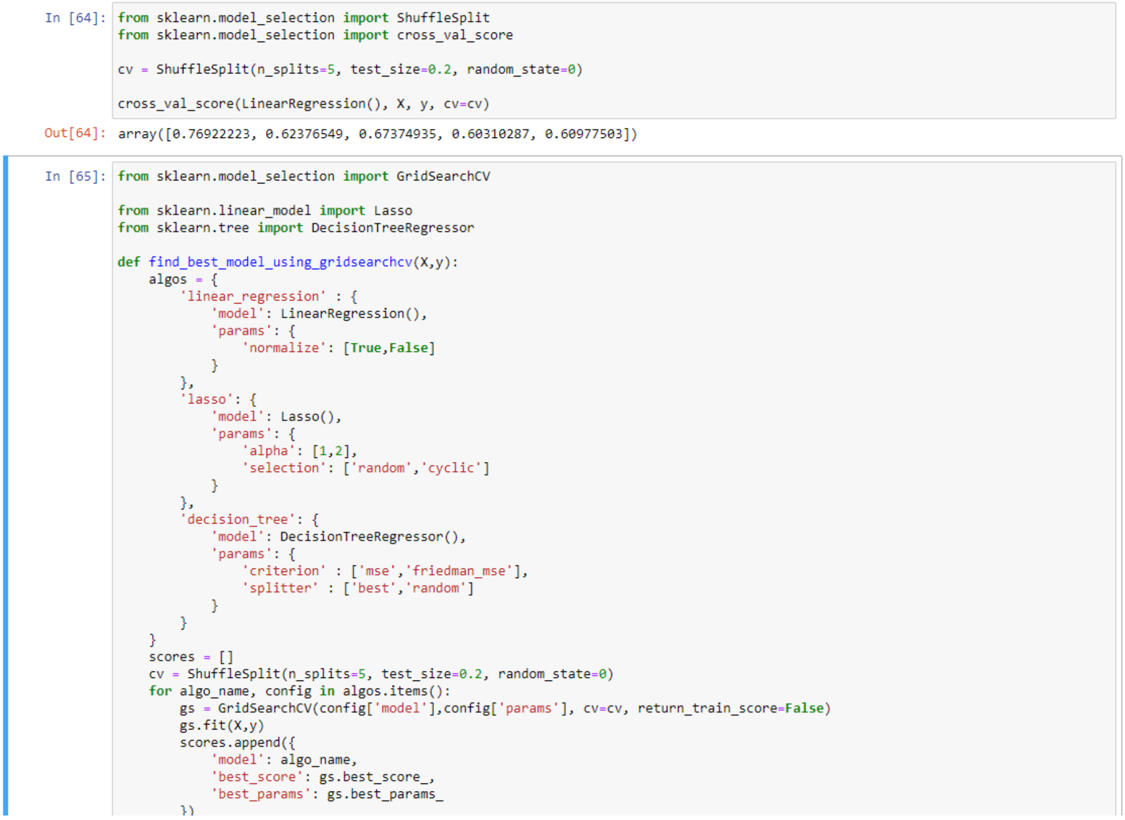


```
In [35]: df5.shape

Out[35]: (13246, 7)

In [36]: df6 = df5[~(df5.total_sqft/df5.bhk<300)]
         df6.shape

Out[36]: (12502, 7)

In [37]: df6.price_per_sqft.describe()

Out[37]: count     12456.000000
         mean       6308.502826
         std        4168.127339
         min         267.829813
         25%        4210.526316
         50%        5294.117647
         75%        6916.666667
         max      176470.588235
         Name: price_per_sqft, dtype: float64

In [38]: def remove_pps_outliers(df):
             df_out = pd.DataFrame()
             for key, subdf in df.groupby('location'):
                 m = np.mean(subdf.price_per_sqft)
                 st = np.std(subdf.price_per_sqft)
                 reduced_df = subdf[(subdf.price_per_sqft>(m-st)) & (subdf.price_per_sqft<=(m+st))]
                 df_out = pd.concat([df_out,reduced_df],ignore_index=True)
             return df_out

         df7 = remove_pps_outliers(df6)
         df7.shape

Out[38]: (10241, 7)

In [39]: def plot_scatter_chart(df,location):
             bhk2 = df[(df.location==location) & (df.bhk==2)]
             bhk3 = df[(df.location==location) & (df.bhk==3)]
             matplotlib.rcParams['figure.figsize'] = (15,10)
             plt.scatter(bhk2.total_sqft,bhk2.price,color='blue',label='2 BHK', s=50)
             plt.scatter(bhk3.total_sqft,bhk3.price,marker='+',color='green',label='3 BHK', s=50)
             plt.xlabel("Total Square Feet Area")
             plt.ylabel("Price Per Square Feet")
             plt.title(location)
             plt.legend()
```



```
In [56]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=10)

In [57]: from sklearn.linear_model import LinearRegression
         lr_clf = LinearRegression()
         lr_clf.fit(X_train,y_train)
         lr_clf.score(X_test,y_test)

Out[57]: 0.6846959387580618

In [58]: y_pred=lr_clf.predict(X_test)
         y_pred

Out[58]: array([50.92348191, 90.21649209, 40.91165648, ..., 22.50946314,
                42.4883219 , 65.62051151])

In [59]: from sklearn.metrics import mean_squared_error, r2_score
         mse1 = mean_squared_error(y_test,y_pred)
         rmse=np.sqrt(mse1)
         r2=r2_score(y_test,y_pred)

In [60]:
         print("root mean square",rmse)

         root mean square 36.43911773784517

In [61]: print("r2_score",r2)

         r2_score 0.6846959387580618

In [62]: from sklearn.metrics import mean_absolute_error
         mae=mean_absolute_error(y_test,y_pred)
         print("mean absolute error",mae)

         mean absolute error 22.50404540972893

In [63]: X_train

Out[63]:
```

	total_sqft	bhk
6500	1200.0	3
8776	2400.0	3
8447	1272.0	3
4466	1150.0	2



```
In [64]: from sklearn.model_selection import ShuffleSplit
         from sklearn.model_selection import cross_val_score

         cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)

         cross_val_score(LinearRegression(), X, y, cv=cv)

Out[64]: array([0.76922223, 0.62376549, 0.67374935, 0.60310287, 0.60977503])
```

```
In [65]: from sklearn.model_selection import GridSearchCV

         from sklearn.linear_model import Lasso
         from sklearn.tree import DecisionTreeRegressor

         def find_best_model_using_gridsearchcv(X,y):
             algos = {
                 'linear_regression' : {
                     'model': LinearRegression(),
                     'params': {
                         'normalize': [True,False]
                     }
                 },
                 'lasso': {
                     'model': Lasso(),
                     'params': {
                         'alpha': [1,2],
                         'selection': ['random','cyclic']
                     }
                 },
                 'decision_tree': {
                     'model': DecisionTreeRegressor(),
                     'params': {
                         'criterion' : ['mse','friedman_mse'],
                         'splitter' : ['best','random']
                     }
                 }
             }
             scores = []
             cv = ShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
             for algo_name, config in algos.items():
                 gs = GridSearchCV(config['model'],config['params'], cv=cv, return_train_score=False)
                 gs.fit(X,y)
                 scores.append({
                     'model': algo_name,
                     'best_score': gs.best_score_,
                     'best_params': gs.best_params_
                 })
```

Fig 8.1

REFERENCES

1. Bandar Almaslukh. (2020). A Gradient Boosting Method for Effective Prediction of Housing Prices in Complex Real Estate Systems. *2020 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, (p. 6).

2. D.Calainho, F., Minne, A. M., & K.Francke, M. (2022). A Machine Learning Approach to Price Indices: Applications in Commercial Real Estate. *The Journal of Real Estate Finance and Economics*, 30.

3. Dieudonné Tchuente, & Serge Nyawa. (2021). Real estate price estimation in French cities using geocoding and machine learning. *Annals of Operations Research*, 38.

4. Dupre, D. (2020). Urban and socio-economic correlates of property prices in Dublin's area. *International Conference on Data Science and Advanced Analytics*, (p. 7).

5. Juergen Deppner; AhlefeldtDehn, Benedict von; Eli Beracha; Wolfgang Schaefers;. (2023). Boosting the Accuracy of Commercial Real Estate Appraisals: An Interpretable Machine Learning Approach. *The Journal of Real Estate Finance and Economics*, 38.

6. Maryam Heidari, Samira Zad, & Setareh Rafatirad. (2021). Ensemble of Supervised and Unsupervised Learning Models to Predict a Profitable Business Decision. *Electronics and Mechatronics Conference* (p. 6). IEEE Xplore.

7. Rangan Gupta, Hardik A. Marfatia, Christian Pierdzioch, & Afees A. Salisu. (2021). Machine Learning Predictions of Housing Market Synchronization across US States: The Role of Uncertainty. *Journal of Real Estate Finance & Economics*, 23.

8. Syafiqah Jamil, Thuraiya Mohd, Suraya Masrom, & Norbaya Ab Rahim. (2020). Machine Learning Price Prediction on Green Building Prices. *IEEE Xplore*, (p. 6).

9. Yanliang Yu, Jingfu Lu, Dan Shen, & Binbing Chen. (2020). Research on real estate pricing methods based on data mining and machine learning. *Neural Computing and Applications*,

10. Gan Srirutchataboon, Saranpat Prasertthum, Ekapol Chuangsuwanich, Ploy N. Pratanwanich, & Chotirat Ratanamahatana. (2021). Stacking Ensemble Learning for Housing Price Prediction: a Case Study in Thailand. *Conference on Knowledge and Smart Technology*, (p. 5).

11. Iva´n Garcı´a-Magarino, Carlos Medrano, & Jorge Delgado. (2019). Estimation of missing prices in real-estate market agent-based simulations with machine learning and dimensionality reduction methods. *Neural Computing and Applications Springer*, 18.

12. NINA RIZUN, & ANNA BAJ-ROGOWSKA. (2021). Can Web Search Queries Predict Prices Change on the Real Estate Market? *Department of Informatics in Management* (p. 23). IEEE Access.

13. J, M., Radha Gupta, & Narahari N S. (2020). Machine Learning based Predicting House Prices using Regression Techniques. *IEEE Xplore Part Number: CFP20K58-ART; ISBN: 978-1-7281-4167-1*, (p. 7).

14. Khosravi, M., Sadman Bin Arif, Ali Ghaseminejad, Hamed Tohidi, & Hanieh Shabanian. (2022). Performance Evaluation of Machine Learning Regressors for Estimating Real Estate House Prices. 18.

15. Zulkifley, N. H., Shuzhlina Abdul Rahman, Nor Hasbiah Ubaidullah, & Ismail Ibrahim. (2020). House Price Prediction using a Machine Learning Model: A Survey of Literature. *Modern Education and Computer Science* (p. 9). MECS.

PAPER NAME

Realestate Report.docx

WORD COUNT

4587 Words

CHARACTER COUNT

25497 Characters

PAGE COUNT

37 Pages

FILE SIZE

546.9KB

SUBMISSION DATE

Nov 8, 2023 08:28 PM GMT+5:30

REPORT DATE

Nov 8, 2023 08:30 PM GMT+5:30

● 8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 8% Internet database
- 7% Publications database
- Crossref database
- Crossref Posted Content database