

Final Year Project Report

HOUSE PRICES PREDICUTION MODEDL

BS(CS) (Session: 2020-24)

Project Supervisor

Ms. Umber Noureen

Lecturer

University of Sahiwal

Submitted by

Student Name : Muhammad Waqas

Roll No : BSCS-M2-20-34



DEPARTMENT OF COMPUTER SCIENCE

PREFACE

In the realm of real estate, the ability to accurately predict house prices is of paramount importance. Whether it's for homeowners looking to sell their property at a fair market value, buyers seeking their dream home within a budget, or investors aiming to make informed decisions, accurate price prediction models serve as invaluable tools.

This project report delves into the development and evaluation of a house price prediction model, crafted with meticulous attention to detail and leveraging cutting-edge techniques in machine learning and data analysis. Our goal is to provide a comprehensive overview of the methodologies employed, the data utilized, the model construction process, and the results obtained.

Throughout this report, we aim to offer insights into the complexities of predicting house prices, exploring the multitude of factors that influence property values. From location and neighborhood characteristics to property size, amenities, and market trends, the variables at play are diverse and dynamic.

By documenting our approach, methodologies, challenges faced, and the outcomes achieved, we endeavor to contribute to the collective knowledge base in the field of real estate analytics and machine learning. Our hope is that this report serves as a valuable resource for researchers, practitioners, and enthusiasts alike, fostering deeper understanding and innovation in the domain of house price prediction.

We extend our gratitude to all those who contributed to this project, whether directly or indirectly, and we invite readers to explore the intricacies of our model and findings as we embark on this journey through the world of real estate analytics.

ACKNOWLEDGMENT

First of all, we thank Almighty Allah who gives us the strength and ability to think, work and deliver what we are assigned to do. Secondly, we must be grateful to our internal supervisor Ms.Umber Noureen, who guided us in this project. We also acknowledge our teachers who guided, taught and helped us during our study period. We would also like to thank all departmental staff and university staff, who had assisted us during our stay at the university.



UNIVERSITY OF SAHIWAL, SAHIWAL

University Online: www.uosahiwal.edu.pk

DEPARTMENT OF COMPUTER SCIENCE **PROJECT APPROVAL FORM**

CERTIFICATE OF COMPLETION

This is to certify that the following student

Student Name : Muhammad Waqas

Roll Number : BSCS-M2-20-34

have successfully completed their final year project titled

House Prices Prediction Model

in the partial fulfillment for the requirements of the Degree of Bachelor of Computer Science & Information Technology during the academic session 2020-2024.

Signatures:

Signatures:

Name of Advisor

Designation
Department of Computer Science
University of Sahiwal

Name of Chairman

Chairman
Department of Computer Science
University of Sahiwal

ABSTRACT

Usually, House price index represents the summarized price changes of residential housing. to make it more easier for a family to search for a house we have made it more precise by asking the required square feet, no of bedrooms and bathrooms required. With preloaded dataset and data features, a practical data pre-processing, creative feature engineering method is examined in this paper. The paper also proposes regression technique in machine learning to predict house price.

Keywords: House Price, Regression Technique, Machine Learning

TABLE OF CONTENTS

	ABSTRACT	v
	LIST OF FIGURES	viii
CHAPTER No.	TITLE	PAGE No.
1.	INTRODUCTION	1
	1.1 MACHINE LEARNING	1
	1.2 ADVANTAGES AND APPLICATIONS	2
2.	LITREATURE SURVEY	7
3.	AIM AND SCOPE OF PROJECT	13
	3.1 EXISTING SYSTEM	13
	3.2 PROPOSED SYSTEM	13
	3.3 FEASIBILITY STUDY	14
4.	EXPERIMENTAL METHODS AND ALGORITHMS	
	4.1 HARDWARE REQUIREMENTS	16
	4.2 SOFTWARE REQUIREMENTS	16
	4.3 PYTHON	17
	4.4 ANACONDA	19
	4.5 SYSTEM DESIGN	25
	4.6 USE CASE DIAGRAM	27
	4.7 SEQUENCE DIAGRAM	28
	4.8 ACTIVITY DIAGRAM	29

5.	RESULTS AND DISCUSSION	30
	5.1 MODULE IMPLEMENTATION	30
	5.2 SOFTWARE TESTION	35
	5.3 RESULTS	
6.	CONCLUSION AND FUTURE WORK	39
7.	REFERENCES	40
8.	APPENDIX	
	A. Model Implementation	42

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
4.1	ANACONDA	19
4.2	ANACONDA NAVIGATOR	19
4.3	SYSTEM DESIGN	25
4.4	USE CASE DIAGRAM	27
4.5	SEQUENCE DIAGRAM	28
4.6	ACTIVITY DIAGRAM	29

CHAPTER 1

INTRODUCTION:

Data is at the heart of technical innovations, achieving any result is now possible using predictive models. Machine learning is extensively used in this approach. Machine learning means providing valid dataset and further on predictions are based on that, the machine itself learns how much importance a particular event may have on the entire system supported its pre-loaded data and accordingly predicts the result. Various modern applications of this technique include predicting stock prices, predicting the possibility of an earthquake, predicting company sales and the list has endless possibilities.

Our aim is to predict a house price based on their needs and priorities.. By analyzing previous market trends and price ranges, and also upcoming developments future prices will be predicted. The functioning involves a website which accepts customers specifications and then combines the application of neuralnetwork.

Machine Learning

It is a subset of artificial intelligence (AI).It provides system the ability to automatically learn and improve by itself.It focuses on the development of computer programs that can access data learn by themselves. The process of learning begins with observations based on the examples that we provide. The aim is to make computers to learn by itself without the need of a human.

Machine Learning Methods

Machine learning can be classified into three types namely the supervised, unsupervised and reinforcement learning. **Supervised machine learning algorithms** can apply what has been learned in the past to new data predict future events. It analysis from a known training dataset, and produces a functions to predict outputs.

The system will provide outputs for inputs after training. The system will compare with the correct, intended output and find errors and modify it to make the model more practical and useful.

In contrast, **unsupervised machine learning algorithms** are the ones which does not require any supervision. It is used when when the sample data used to train is classified. As name suggests it, the model itself finds the hidden patterns and insights. The system may or may not produce right output, but it explores the data and can draw inferences from datasets by its own.

Semi-supervised machine learning algorithms is a combination of both supervised and unsupervised learning, In semi-supervised learning, an algorithm learns from a dataset that includes both labeled and unlabeled data, usually mostly unlabeled. Generally it is chosen when the sample data requires skilled resources in order to train from it. Otherwise, It doesn't require additional resources.

Reinforcement machine learning algorithms is a learning method that works based on feedback. Reinforcement learning differs from supervised learning in not needing labelled input/output pairs be presented. It is studied in various disciplines such as statistics, information theory etc.

Advantages of Machine Learning

It helps to manage a large amount of data. There is no need for human interference. It can also perform complex operations by its own. It is extremely useful for those who are in the field of e commerce or even healthcare. It is extremely useful in manufacturing industry.

While even experts often cannot be sure where and by which correlation a production error in a plant fleet arises, Machine Learning offers the possibility to identify the error early this saves down times and money. Machine learning are now used in the medical field. In the future, after collecting huge amounts of data apps will be able to warn in case his doctor wants to prescribe a drug that he cannot tolerate. The app can also suggest alternative options by taking into account the genetics of patient.

Applications of Machine Learning

1. Virtual Personal Assistants

There are many personal assistants available like apple's siri, google's google assistant and amazon's alexa. The only work for them is to find information when customers asks to find it over voice. To ask any questions we need to activate them and ask "What is the time in London?" or similar questions. To answer, it your personal assistant looks out for the information in browser, or collect it from phone apps. You can even ask your assistants for certain tasks like "Set a reminder for tomorrow", "Remind me to wish my friends birthday". Here the personal assistants uses machine learning to respond to users task or questions. These assistants are also integrated in various other devices such as televisions (smart tv) and speakers. These assistants make these devices a more smarter one.

2. Traffic Predictions :

Whenever we visit to a new place or when we are not sure about the route we generally use maps it shows the distance, the amount of time it takes to cover the distance and also it provides the information regarding traffic congestion. By making use of machine learning it predicts the traffic in the particular route by analyzing the previous days traffic on the route on the same time. Hence machine learning helps us in predicting traffic.

3. IDEO SURVEILLANCE:

A single person cannot be monitoring multiple cameras at single time that's where machine learning is used nowadays video cameras are powered by AI hence it helps us by tracking unusual behaviour for example if a person is standing motionless for long time or if a person is stumbling then it alerts the attendant who is looking after the camera .It has been used extensively in video surveillance and it has been extremely useful.

4.EMAIL SPAM AND FILTERING

Machine learning has been extensively used in checking spam and malware emails .It detects new malware and protects users against it.It can detect various malwares and can protect us .

5.online customer support:

Many websites are providing customers with a chatbox to answer their queries and doubts but most of the time there will not be any executive behind chatbox.These chatbox are powered by Ai and machine learning makes them to get better.These chatboxes gets better with time.

6.Search Engine Result Refining:

Whenever we search for anything in web the search engine for example if it is google then it will keep track of what users are opening after the results are shown.it checks whether the users are clicking the top search result or the bottom ones.Machine learning helps and makes the search engine better with time.

7. Product recommendations:

Every time when a product is recommended for you ,be it after you purchase a certain product from the website or it's a new product machine learning is the one that helps in recommending products to customers.

8. Online fraud detection:

It helps in detecting money fraud in online .many payment gateways have started to implement this technique to prevent fraud .company like paypal uses machine learning to detect fraud.

INTRODUCTION TO PROJECT

Housing is one of the most valuable economic assets an individual can purchase during his adult life. Hence we need to be extremely careful before buying a house we need to spend correct money to buy a house.

In the following, we explore different machine learning techniques and methodologies to predict house prices. The data contains a train and a test dataset. Our objective is, to predict house prices based on users requirements and needs .Our model predicts the price of a house from the sample data that has been given.

CHAPTER 2

LITERATURE SURVEY

Literature Survey

1. Housing Price Prediction Using Machine Learning Algorithms: The Case of Melbourne City, Australia

Author: The Danh Phan, 2018

House price Prediction is a crucial topic of land . The literature attempts to get useful knowledge from historical data of property markets. Machine learning techniques are applied to research historical property transactions in Australia to get useful models for house buyers and sellers. Revealed is the the high discrepancy between house prices within the costliest and most affordable places within Melbourne city. Moreover, experiments demonstrate that the mixture of Stepwise and Support Vector Machine that's supported mean squared error measurement may be a competitive approach.

2. Predicting Sales Prices of the Houses Using Regression Methods of Machine Learning

Authors: Parasich Andrey Viktorovich ; Parasich Viktor Aleksandrovich ; Kaftannikov Igor Leopoldovich ; Parasich Irina Vasilevna, 2018

This article we'll describe our solution for "House Prices: Advanced Regression Techniques" machine learning competition, which was persisted Kaggle platform. The goal is to predict house sale price by attributes like house area, year of building etc. In our solution, we use classic machine learning algorithms, and our original methods, which may be described here. At the highest of the competition, we took 18th place among 2124 participants from whole world.

3. Real Estate Value Prediction Using Linear Regression

Authors: Nehal N Ghosalkar ; Sudhir N Dhage, 2018

The real estate market may be a standout amongst the foremost focused regarding pricing and keeps fluctuating. It is one among the prime fields to use the ideas of machine learning on the way to enhance and foresee the prices with high accuracy. There are three factors that influence the price of a house which includes physical conditions, concepts and location. The current framework includes estimating the worth of homes with none expectations of market prices and price increment. The objective of the paper is prediction of residential prices for the purchasers considering their financial plans and wishes . By breaking down past market patterns and value ranges, and coming advancements future costs are going to be anticipated. This examination means to predict house prices in Mumbai city with Linear Regression. It will help clients to place resources into a gift without moving toward a broker. The result from this research proved linear regression gives minimum prediction error which is 0.3713.

4. Predicting Housing Market Trends Using Twitter Data

Authors: Marlon Velthorst ; Cicek Güven, 2019

In this study, we attempt to predict the Dutch housing market trends using text mining and machine learning as an application of knowledge science methods in finance. Our main goal is to predict the short term upward or downward trend of the average house price in the Dutch market by using text data collected from Twitter. Twitter is widely used also and has been proven to be a helpful source of knowledge . However, Twitter, text mining (tokenization, bag-of-words, n-grams, weighted term frequencies) and machine learning (classification algorithms) have not been combined yet in order to predict the housing market trends in short term. In this study, tweets including predefined search words are collected counting on domain knowledge, and therefore the corresponding text is grouped by month as documents. Then words and word sequences are transformed into numerical values. These values served as attributes to predict whether the housing market moves up or down,

i.e. we approached this as a binomial classification problem relating text data of a month with (up or down) trends for the subsequent month.

Our main results reveal there's a correlation between the (weighted) frequency of words and short term housing trends, in other words, we were ready to make accurate predictions of trends in short term using multiple machine learning and text mining techniques combined.

5. House Price Prediction Using Machine Learning and Neural Networks

Authors: Ayush Varma ; Abhijit Sarma ; Sagar Doshi ; Rohini Nair, 2018

Real estate is that the least transparent industry in our ecosystem. Housing prices keep changing day in and outing and sometimes are hyped instead of being supported valuation. Predicting housing prices with real factors is that the main crux of our scientific research. Here we aim to form our evaluations supported every basic parameter that's considered while determining the worth. We use various regression techniques during this pathway, and our results aren't sole determination of 1 technique rather it's the weighted mean of varied techniques to offer most accurate results. The results proved that this approach yields minimum error and maximum accuracy than individual algorithms applied. We also propose to use real-time neighborhood details using Google maps to urge exact real-world valuations.

6. Forecasting house price index of China using dendritic neuron model

Authors: Ying Yu ; Shuangbao Song ; Tianle Zhou ; Hanaki Yachi ; Shangce Gao, 2016

The results of Chinese housing market continues to prosper or not is said to the event of China, and further it also has an impression on the planet finance. Thus forecasting the house price level is extremely important and challenging. during this paper we propose an unsupervised learnable neuron

model (DNM) by including the nonlinear interactions between excitation and inhibition on dendrites.

We use DNM to suit the House price level (HPI) data then forecast the trends of Chinese housing market. To verify the effectiveness of the DNM, we use a standard statistical model (i.e., the exponential smoothing (ES) model) to form a performance comparison. Three quantitative statistical metrics including normalized mean square error, absolute percentage of error, and coefficient of correlation are used to evaluate the forecasting performance of the 2 models. Experimental results demonstrate that the proposed DNM is best than ES altogether of the three quantitative statistical metrics.

7. Prediction of real estate price variation based on economic parameters

Authors: Li Li ; Kai-Hsuan Chu, 2017

It is documented that a lot of economic parameters may more or less influence the important estate price variation. Additionally, the banker and investor also are interesting to understand the important estate price future change. There had not appropriate model for including these factors for price prediction. Here, the influences of most macroeconomic parameters on land price variation are investigated before establishing the worth fluctuation prediction model. Here, back propagation neural network (BPN) and radial basis function neural network (RBF) two schemes are employed to determine the nonlinear model for real estates price variation prediction of Taipei, Taiwan supported leading and simultaneous economic indices. Those prediction results are compared with the general public Cathay House price level or the Sinyi Home price level. The mean absolute error and root mean square error two indices of the worth variation are selected because the performance index. The general public related data of Taipei, Taiwan land variation during 2005 ~ 2015 are adopted for analysis and prediction comparison.

8. Predicting house sale price using fuzzy logic, Artificial Neural Network and K-Nearest Neighbor

Authors: Muhammad Fahmi Mukhlisin ; Ragil Saputra ; Adi Wibowo, 2017

Determining the worth of land and residential are regularly determined at the earliest by the vendor , however determining the proper price within the sales process will affect the buyer's desire to elect and bid. Special characteristics in Indonesia, tax object value (NJOP) and site parameters are high influence to the worth . during this paper we proposed the prediction of land and house value using several methods. symbolic logic , Artificial Neural Network and K-Nearest Neighbor are compared during this paper to get the foremost appropriate method which will be used as a reference for determining the worth by the sellers. Google Maps is employed to represent the spatial data for prediction parameter. The variables that utilized in the methods are NJOP of land, the locations, the age, NJOP of house, and therefore the valuable location of the land. The experimental methods are tested by comparing between the important price transaction and therefore the prediction using MAPE formula.

9. Comprehensive Analysis of Housing Price Prediction in Pune Using Multi-Featured Random Forest Approach

Authors: Rushab Sawant ; Yashwant Jangid ; Tushar Tiwari ; Saurabh Jain ; Ankita Gupta, 2018

The housing sector in India has been predicted to grow at 30-35% over subsequent decade. In terms of employment provided, it's second only to the agricultural sector. Housing is one among the main domain of land . Pune is emerging together of the main metropolitan cities of India and has many prestigious Educational institutions and IT parks. This makes it a perfect place to shop for homes. Vagueness among the costs of homes makes it challenging for the customer to pick their dream house.

The interests of both buyer and seller should be satisfied in order that they are doing not overestimate or underestimate price. This housing price prediction model acts as a hand for buyer and seller or a true realtor to form a better-informed decision. to realize this, diverse features are selected as input from feature set and various algorithms are applied like Random Forest and Decision Tree.

10. Time-Aware Latent Hierarchical Model for Predicting House Prices

Authors: Fei Tan ; Chaoran Cheng ; Zhi Wei, 2017

It is widely acknowledged that the worth of a home is the mixture of an outsized number of characteristics. House price prediction thus presents a singular set of challenges in practice. While an outsized body of works are dedicated to the present task, their performance and applications are limited by the shortage of while span of transaction data, the absence of real-world settings and therefore the insufficiency of housing features. to the present end, a time-aware latent hierarchical model is introduced to capture underlying spatiotemporal interactions behind the evolution of house prices. The hierarchical perspective obviates the necessity for historical transaction data of exactly same houses when temporal effects are considered. The proposed framework is examined on a large-scale dataset of the property transaction in Beijing. the entire procedure strictly complies with the real-world scenario. The empirical evaluation results demonstrate the outperformance of our approach over alternative competitive methods.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT SYSTEM

EXISTING SYSTEM

Multi Linear Regression

Multiple Linear Regression. It shows the relationship between two or more explanatory variables and scalar response variable. Independent variable value is associated with dependent variable value

Limitations

The dependent variable y must be continuous.. The independent variables can be of any type. The dependent variable is usually affected by the independent variables.

Proposed System

Linear Regression is a technique that helps to identify the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression.

Advantages

- Space complexity is very low it just needs to save the weights at the end of training. hence it's a high latency algorithm.
- Its very simple to understand
- Good interpretability
- Feature importance is generated at the time model building. With the help of hyperparameter λ , you can handle features selection hence we can achieve dimensionality reduction

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. The feasibility study of the proposed system is carried out. It is carried out to ensure that the proposed system is not a burden to the company. Economic feasibility

1. Economical feasibility
2. Technical feasibility
3. Social feasibility

ECONOMICAL FEASIBILITY

This study is generally carried out to check whether right amount of funds are invested in the model. this study is done to eliminate excess amount of money poured into a single model. It makes sure whether the model is well within the budget. It is extremely important to spend only right amount of funds to a model.

TECHNICAL FEASIBILITY

It makes sure whether the technical requirements are limited to what we can offer. Any system developed should not have high demand on technical resources since it puts burden on client, It also checks the projects potential what it can do once developed.

SOCIAL FEASIBILITY

It is carried out check how a system acts with other systems. It checks the level of acceptance of the system by the user. It trains the user to use the system efficiently. it is a necessity. Since a client is the final user of the system he can criticize the system but it should be in a disciplined and meaningful manner.

CHAPTER 4

EXPERIMENTAL METHODS AND ALGORITHMS

HARDWARE REQUIREMENTS

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

1. PROCESSOR : PENTIUM IV
2. RAM : 8 GB
3. PROCESSOR : 2.4 GHZ
4. MAIN MEMORY : 8GB RAM
5. PROCESSING SPEED : 600 MHZ
6. HARD DISK DRIVE : 1TB
7. KEYBOARD :104 KEYS

SOFTWARE REQUIREMENTS

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

1. FRONT END :PYTHON
2. IDE : ANACONDA
3. OPERATING SYSTEM :WINDOWS 10

Python Language

- Python is an object-oriented programming language
- It is created by Guido Rossum in 1989.
- It is ideally designed for rapid prototyping of complex applications.
- It is extensible to C or C++.
- Companies like google and nasa also uses python language
- It is majorly used in AI

Python Programming Characteristics

- It provides rich data types
- Syntax is simple
- It is a platform independent scripted language
- Compared to other programming languages, it allows more run-time flexibility
- A module in Python may have one or more classes and free functions
- Libraries in Pythons can also run in Linux and Windows
- For building large applications, Python can be compiled to byte-code
- It supports functional and structured programming
- It supports interactive mode that allows interacting Testing and debugging of snippets of code
- In Python editing, debugging and testing is fast.

Applications of Python Programming

Web Applications

We can create web apps in python by using frameworks and CMS. We can create web applications using Django, Flask, Pyramid, Plone, Django CMS. Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

Scientific and Numeric Computing

There are many number of libraries in python that can be used for scientific and numeric computing . SciPy and NumPy that are used in general purpose computing. EarthPy is used for earth science, AstroPy is used for Astronomy and so on. It is also used in machine learning, data mining and deep learning.

Creating software Prototypes

Python is slow but is great for creating prototypes. For example: You can use Pygame which is used to create game prototype. If you are satisfied with the prototype then you can build the app using C or C++.

Good Language to Teach Programming

Python has been used by many students. There are several companies teaching python to their employees. It has a lot of features and capabilities. The syntax is simple and it is one of the easiest language to learn.

About Opencv Package

Python is a general purpose programming language started by Guido van Rossum, It became very popular because of its simplicity and code readability. It helps the programmer to express his ideas in fewer lines of code .

Compared to other languages like C/C++, Python is slower. Python can be easily extended with C/C++. We can write codes in C/C++ and create a python wrapper.

This gives us two advantages: first, our code is as fast as original C/C++ code and second, it is very easy to code in Python. Hence OpenCV-Python is a Python wrapper around original C++ implementation.

Python also supports Numpy. It gives a MATLAB-style syntax. The OpenCV array structures are converted to-and-from Numpy arrays. Whatever operations you can do in Numpy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this.

So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

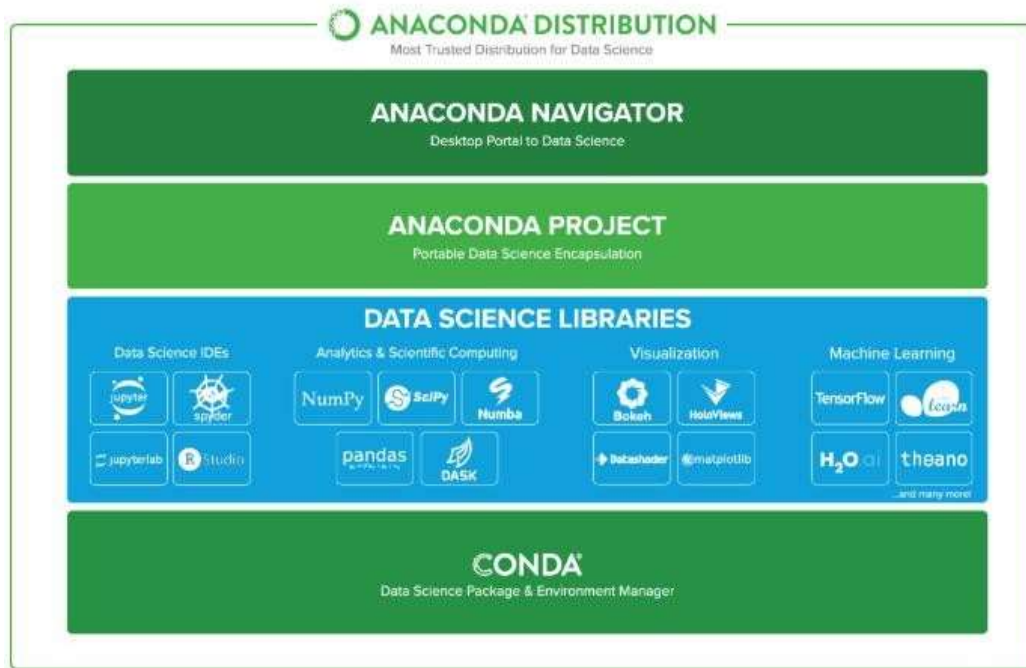
FEATURES OF ANACONDA NAVIGATOR

Anaconda is free

It is open source, easy to install distribution of Python and R programming languages.

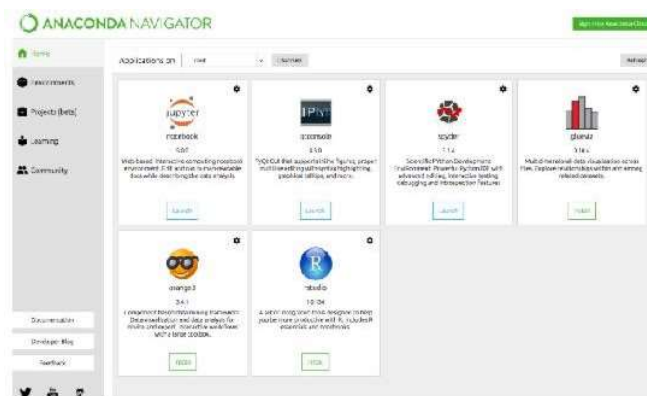
It is used for scientific computing, data science, statistical analysis and machine learning.

The latest distribution of Anaconda is Anaconda 5.3 .



What is Anaconda Navigator?

Anaconda Navigator may be a desktop graphical interface (GUI) included within the Anaconda distribution. It allows us to launch applications provided within the Anaconda distribution and simply manage conda packages, environments and channels without the utilization of command-line commands. It is available for Windows, macOS and Linux.



Anaconda Navigator

Applications Provided In Anaconda Distribution

The Anaconda distribution comes with the subsequent applications along side Anaconda Navigator.

1. JupyterLab
2. Jupyter Notebook
3. Qt Console
4. Spyder
5. Glueviz
6. Orange3
7. RStudio
8. Visual Studio Code

> JupyterLab: This is the extensible working environment for interactive and the reproducible computing, supported the Jupyter Notebook and Architecture.

>Jupyter Notebook: This is an web-based, interactive computing notebook environment. we will able to edit and runs in human-readable docs while describing the info analysis.

> Qt Console: It is an PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips and etc..,

Spyder: Spyder is an scientific Python Development Environment. It is a powerful Python IDE of advanced editing, interactive testing, debugging and the introspection features.

VS Code: It is an streamlined code editor within the support for development operations like debugging, task running and version control.

Glueviz: It is used for multidimensional data visualization across the files. It is explored in relationships within and among related datasets.

Orange 3: It is an component-based on data mining framework. it can be used for the data visualization and data analysis. The workflows under Orange 3 is very interactive and provides a large toolbox.

Rstudio: This is a set of integrated tools designed for help you to be more productive by R. Then it includes R essentials and notebooks.

New Features of Anaconda 5.3



Compiled by Latest Python release: Anaconda 5.3 is compiled by Python 3.7, taking advantage of Python's speed and feature improvements.

- **Better Reliability:** The reliability of Anaconda is improved in the latest release by capturing and storing the package metadata for the installed packages.

Users deploying Tensorflow can make usefull by MKL 2019 for Deep Neural Networks. These Python binary packages are provided to realize the high CPU performance.

- **New packages has been added:** These packages are over 230 packages which is updated and added in the new release.

- **add Progress:** there's a casting bug in Numpy with Python 3.7 but the team is currently performing on patching it until Numpy is updated.

Flask

Flask is an API of Python that permits to create up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and it is also easier to learn because it has the less base code to implement a simple web-Application.

A Web-Application Framework or Web Framework is the collection of modules and libraries that helps the developer to write down applications

without writing the low-level codes like protocols, thread management, etc.

Flask is predicated on WSGI(Web Server Gateway Interface) toolkit and Jinja2 template engine.

METHOD	DESCRIPTION
--------	-------------

GET	This is used to send the data in an without encryption of the form to the server.
-----	---

HEAD	provides response body to the form
------	------------------------------------

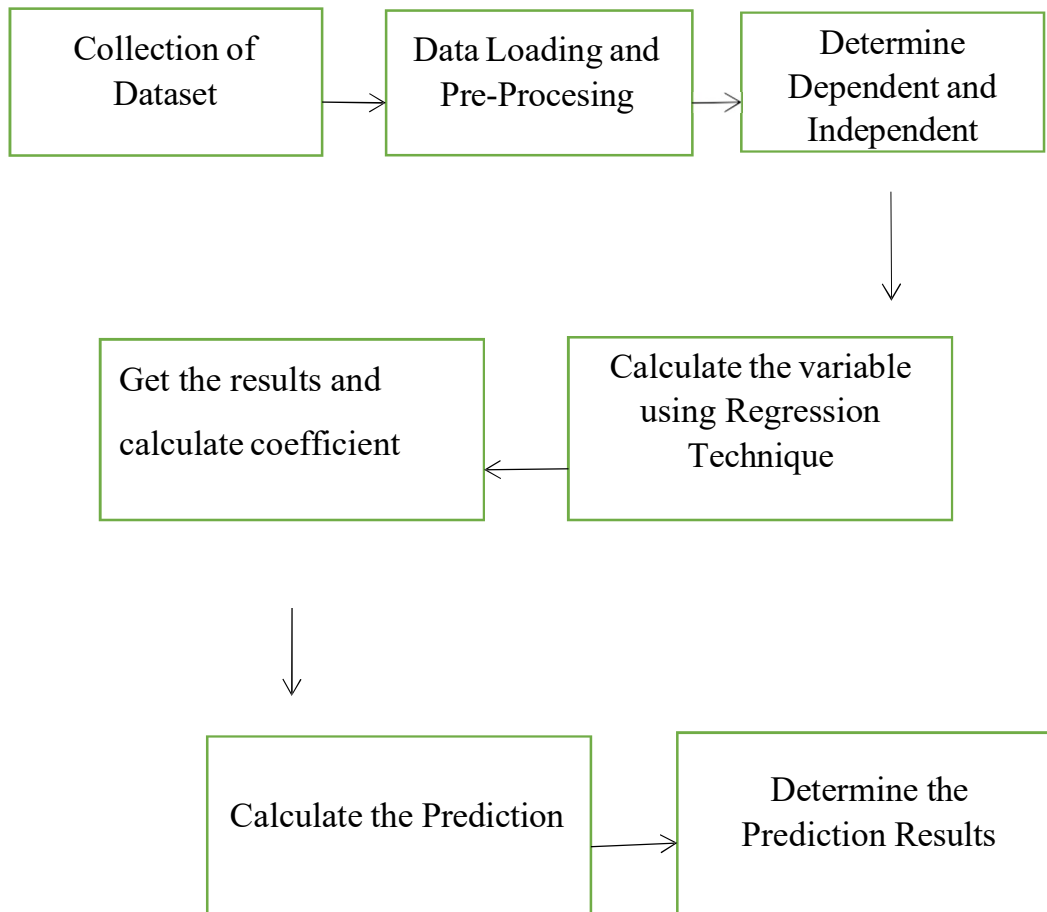
POST	Sends the form data to server. Data received by POST method is not cached by server.
------	--

PUT	Replaces current representation of target resource with URL.
-----	--

DELETE	Deletes the target resource of a given URL
--------	--

SYSTEM DESIGN

Architecture



UML DIAGRAMS

- UML stands for Unified Modeling Language.
- It is used in the field of object-oriented software engineering.
- The goal is for UML to become a common language for creating models of object oriented computer software.
- It consists of two components: a Meta-model and a notation..
- The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.
- It has been proven successful in the modeling of large and complex systems.
- The UML is a very important part of developing objects oriented software and the software development process. It uses graphical notations to show the design of software projects.

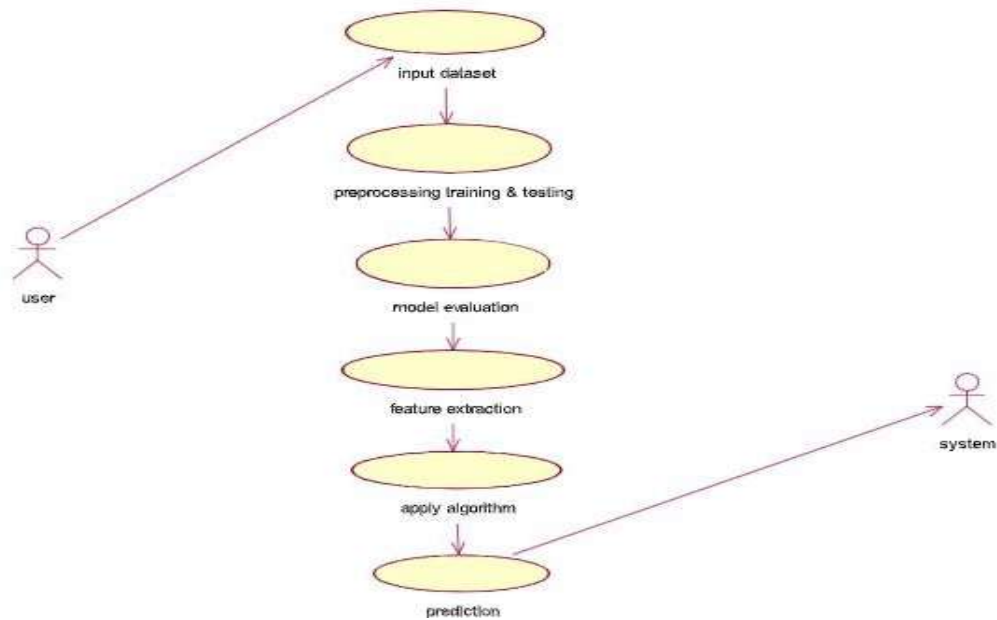
GOALS:

The Primary goals are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

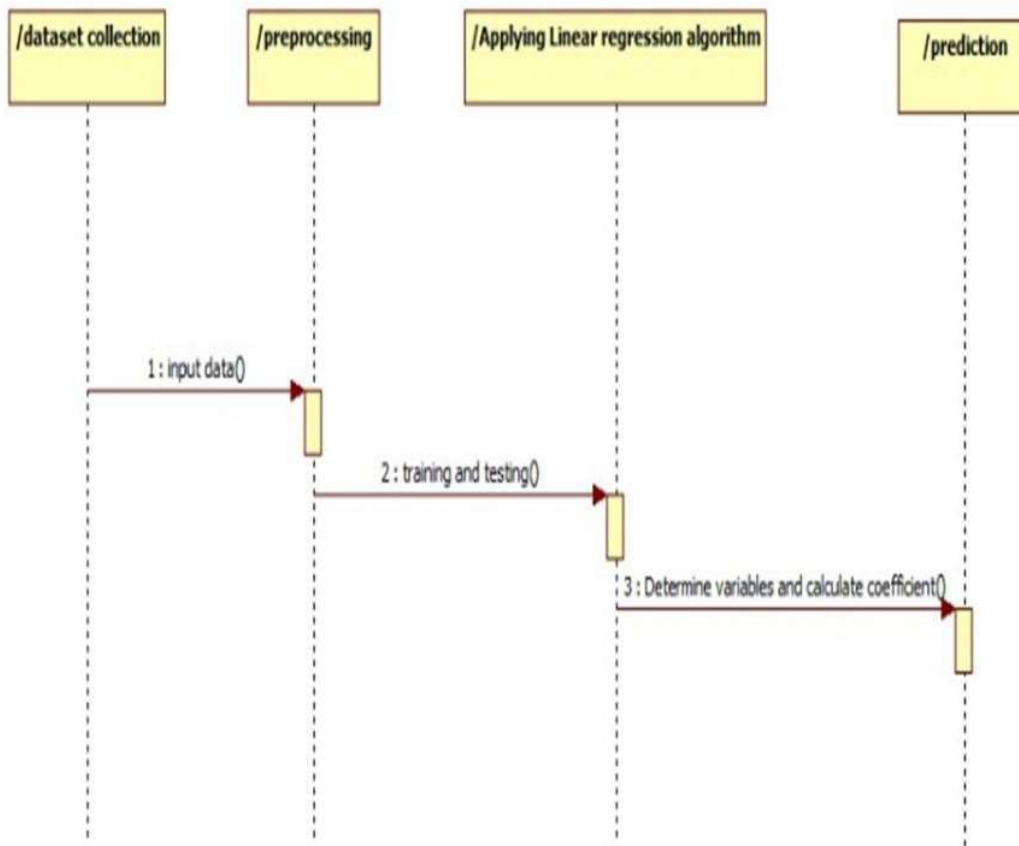
USE CASE DIAGRAM:

A use case diagram is a behavioural diagram. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



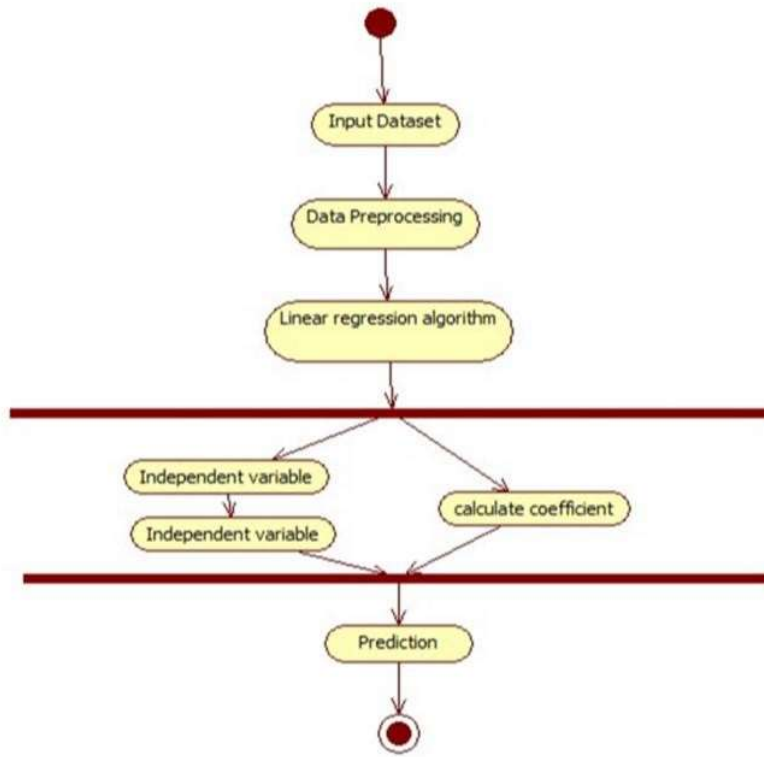
SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a interaction diagram that shows how processes operate with one another and in what order.. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. Activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



CHAPTER 5

RESULTS AND PERFORMANCE ANALYSIS

Module Implementation

Collection of Dataset

The dataset used in this project was Parameters such as Area in square meters, Location, no of bedrooms and no of bathrooms in that particular property. Selling price is a dependent variable on several other independent variables.

Data Preprocessing

It is a process of transforming the raw, complex data into systematic understandable knowledge. It will find out missing and redundant data in the dataset. Thus, this brings uniformity in the dataset. However in our dataset, there was no missing values .

Import Libraries

A library is a collection of modules the first step is to import the libraries that we require in our system. There are functions for them, which can be invoked without writing the required code. This is a list for most popular Python libraries for Data Science. We have imported pandas library and named it as pd.

Import the Dataset

A lot of datasets come in CSV formats. At first we have to locate the directory of the CSV file and read it using a method called `read_csv` which may be found in the library called `pandas`.

Encoding categorical data

Sometimes our data is in qualitative form, that is we have texts as our data. We can find categories in text form. Now it gets complicated for machines to know texts and process them, rather than numbers, since the models are based on mathematical equations and calculations. Therefore, we have to encode the categorical data.

Split Dataset into Training and Test Set

Now we should split our dataset into two sets — a Training set and a Test set. We will train our machine learning models on our training set, i.e. our machine learning models will try to understand any correlations in our training set and then we will test the models on our test set to check how accurately it can predict. In general we need to allocate 80% of the dataset to training set and the remaining 20% to test set.

Dependent and independent variable in regression

Regression analysis describes the relationship between independent variables and the dependent variable. It predicts the value of the dependent variable by analyzing the value of independent variables.

Regression coefficient

It is the same as the slope of the line of the regression equation.

Prediction

Prediction is nothing but the output of an algorithm after being trained on a dataset and applied to new data and predicts the output. Finally our model will predict the house price based on user inputs.

SOFTWARE TESTING

General

In a generalized way, we can say that the system testing is a type of testing in which the main aim is to make sure that system performs efficiently and seamlessly. The process of testing is applied to a program with the main aim to discover an unprecedented error, an error which otherwise could have damaged the future of the software. Test cases which brings up a high possibility of discovering and error is considered successful. This successful test helps to answer the still unknown errors.

TEST CASE

Testing, as already explained earlier, is the process of discovering all possible weak-points in the finalized software product. Testing helps to counter the working of sub-assemblies, components, assembly and the complete result. The software is taken through different exercises with the main aim of making sure that software meets the business requirement and user-expectations and doesn't fails abruptly. Several types of tests are used today. Each test type addresses a specific testing requirement.

Testing Techniques

A test plan is a document which describes approach, its scope, its resources and the schedule of aimed testing exercises. It helps to identify almost other test item, the features which are to be tested, its tasks, how will everyone do each task, how much the tester is independent, the environment in which the test is taking place, its technique of design plus the both the end criteria which is used, also rational of choice of theirs, and whatever kind of risk which requires emergency planning. It can be also referred to as the record of the process of test planning. Test plans are usually prepared with signification input from test engineers.

(I) UNIT TESTING

In unit testing, the design of the test cases is involved that helps in the validation of the internal program logic. The validation of all the decision branches and internal code takes place. After the individual unit is completed it takes place. Plus it is taken into account after the individual unit is completed before integration. The unit test thus performs the basic level test at its component stage and test the particular business process, system configurations etc. The unit test ensures that the particular unique path of the process gets performed precisely to the documented specifications and contains clearly defined inputs with the results which are expected.

(II) INTEGRATION TESTING

These tests are designed to test the integrated software items to determine whether if they really execute as a single program or application. The testing is event driven and thus is concerned with the basic outcome of field. The Integration tests demonstrate that the components were individually satisfaction, as already represented by successful unit testing, the components are apt and fine. This type of testing is specially aimed to expose the issues that come-up by the components combination.

(III) FUNCTIONAL TESTING

The functional tests help in providing the systematic representation that functions tested are available and specified by technical requirement, documentation of the system and the user manual.

(IV) SYSTEM TESTING

System testing, as the name suggests, is the type of testing in which ensure that the software system meet the business requirements and aim. Testing of the configuration is taken place here to ensure predictable result and thus analysis of it. System testing is relied on the description of process and its flow, stressing on pre driven process and the points of integration.

V) WHITE BOX TESTING

The white box testing is the type of testing in which the internal components of the system software is open and can be processed by the tester. It is therefore a complex type of testing process. All the data structure, components etc. are tested by the tester himself to find out a possible bug or error. It is used in situation in which the black box is incapable of finding out a bug. It is a complex type of testing which takes more time to get applied.

(VI) BLACK BOX TESTING

The black box testing is the type of testing in which the internal components of the software is hidden and only the input and output of the system is the key for the tester to find out a bug. It is therefore a simple type of testing. A programmer with basic knowledge can also process this type of testing. It is less time consuming as compared to the white box testing. It is very successful for software which are less complex are straight-forward in nature. It is also less costly than white box testing.

(V) ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also make sures that the system meets the functional requirement.

RESULTS:

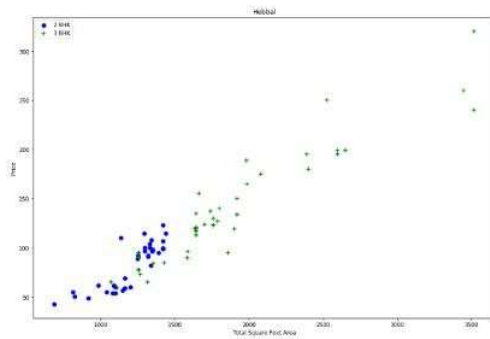
SAMPLE DATA SET:

area_type	availability	location	size	society	total_sqft	bath	balcony	price
Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2	1	39.07
Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5	3	120
Built-up Area	Ready To Move	Uttarahalli	3 BHK		1440	2	3	62
Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3	1	95
Super built-up Area	Ready To Move	Kothanur	2 BHK		1200	2	1	51
Super built-up Area	Ready To Move	Whitefield	2 BHK	DuenaTa	1170	2	1	38
Super built-up Area	18-May	Old Airport Road	4 BHK	Jaades	2732	4		204
Super built-up Area	Ready To Move	Rajaji Nagar	4 BHK	Brway G	3300	4		600
Super built-up Area	Ready To Move	Marathahalli	3 BHK		1310	3	1	63.25
Plot Area	Ready To Move	Gandhi Bazar	6 Bedroom		1020	6		370
Super built-up Area	18-Feb	Whitefield	3 BHK		1800	2	2	70
Plot Area	Ready To Move	Whitefield	4 Bedroom	Prrry M	2785	5	3	295
Super built-up Area	Ready To Move	7th Phase JP Nagar	2 BHK	Shncyes	1000	2	1	38
Built-up Area	Ready To Move	Gottigere	2 BHK		1100	2	2	40
Plot Area	Ready To Move	Sarjapur	3 Bedroom	Skityer	2250	3	2	148
Super built-up Area	Ready To Move	Mysore Road	2 BHK	PrntaEn	1175	2	2	73.5
Super built-up Area	Ready To Move	Bisuvanahalli	3 BHK	Prityel	1180	3	2	48
Super built-up Area	Ready To Move	Raja Rajeshwari Nagar	3 BHK	GrrvaGr	1540	3	3	60

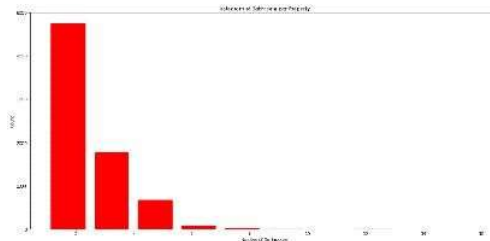
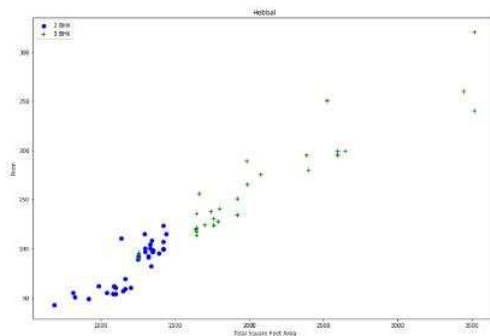
These are the sample for preloaded data sets in our model

Graph:

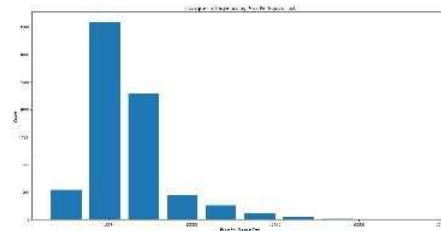
Before deleting anomalies:



After deleting anomalies (we don't have any unwanted data):



This graph shows bathrooms per property



This graph represents property price by square feet

Importing libraries:

We use pandas library to read the train and test files.

```
import pandas as pd ( used for data analysis)
```

```
import numpy as np (Used for computations)
```

```
import matplotlib.pyplot as plt ( used to plot values in graph)
```

Data preprocessing :

It gets the count of area type in dataset and removes unwanted columns

Encoding categorical data:

	weather	..		is_sunny	is_rainy	..
0	sunny	..	0	1	0	..
1	rainy	..	1	0	1	..
2	rainy	..	2	0	1	..
3	sunny	..	3	1	0	..

OH encoding

Splitting dataset into train and test data:

We are taking 80% of our data as training data and 20% as test data.

```
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)
```

Dependent and independent variable in regression:

Eg:

Locality	Area	Bedrooms	Bathrooms	Price
Electronic City	1056	2	1	40
Whitefield	1170	2	1	38

Dependent variable in our model is price(since it relies on other factors for its value)

Independent variables in our model are locality,Area,Bedrooms and bathrooms since it doesn't depend on other variables for its value.

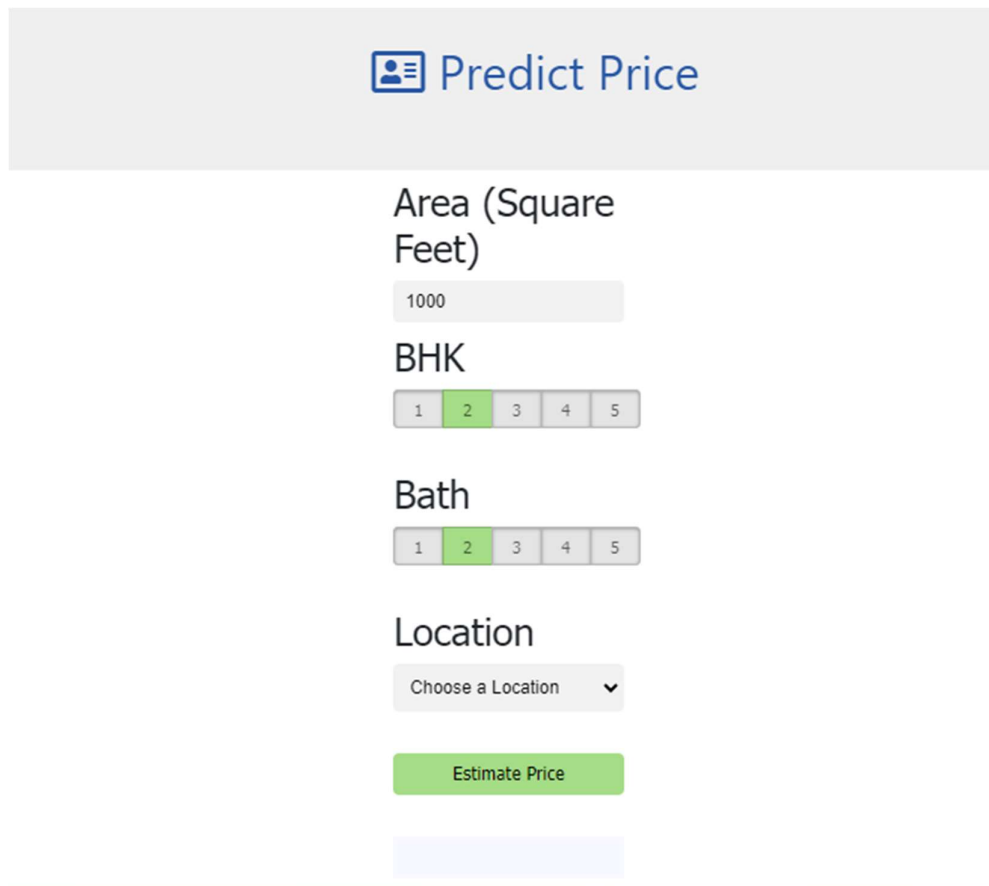
Using linear regression it predicts the value of output.

Linear regression:

It predicts the result value from user defined datasets

```
def predict_price(location,sqft,bath,bhk):  
    loc_index = np.where(X.columns == location)[0][0]  
  
    x = np.zeros(len(X.columns))  
    x[0] = sqft  
    x[1] = bath  
    x[2] = bhk  
    if loc_index >= 0:  
        x[loc_index] = 1  
    return regressor.predict([x])[0]
```

Screenshot:



The screenshot displays a web interface for predicting house prices. At the top, there is a header with a user icon and the text 'Predict Price'. Below this, the form is organized into sections: 'Area (Square Feet)' with a text input field containing '1000'; 'BHK' with a row of five buttons (1, 2, 3, 4, 5) where button '2' is highlighted in green; 'Bath' with a similar row of five buttons (1, 2, 3, 4, 5) where button '2' is also highlighted in green; and 'Location' with a dropdown menu showing 'Choose a Location' and a downward arrow. At the bottom of the form is a green button labeled 'Estimate Price'. Below the button is a light blue rectangular box, likely intended for the predicted price output.

Fig: The final output of our model

CHAPTER 6

Conclusion and Future Work

In this paper, several tests have been performed using linear regression algorithm to perform house price prediction. This algorithm is to predict prices of new properties that are going to be listed by taking some input variables and predicting the correct and justified price. It was a great learning experience building this predictive Sale Price model. In Future Using different methods that match the time-series data will be used in the research to obtain smaller error prediction values and using more data to get the better result.

References

1. Housing Price Prediction Using Machine Learning Algorithms: The Case of Melbourne City, Australia, The Danh Phan, 2018 International Conference on Machine Learning and Data Engineering (iCMLDE)
2. Predicting Sales Prices of the Houses Using Regression Methods of Machine Learning, Parasich Andrey Viktorovich ; Parasich Viktor Aleksandrovich ; Kaftannikov Igor Leopoldovich ; Parasich Irina Vasilevna, 2018 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC)
3. Real Estate Value Prediction Using Linear Regression, Nehal N Ghosalkar ; Sudhir N Dhage, 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)
4. Predicting Housing Market Trends Using Twitter Data, Marlon Velthorst ; Cicek Güven, 2019 6th Swiss Conference on Data Science (SDS)
5. House Price Prediction Using Machine Learning and Neural Networks, Ayush Varma ; Abhijit Sarma ; Sagar Doshi ; Rohini Nair, 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)
6. Forecasting house price index of China using dendritic neuron model, Ying Yu ; Shuangbao Song ; Tianle Zhou ; Hanaki Yachi ; Shangce Gao, 2016 International Conference on Progress in Informatics and Computing (PIC)

7. Prediction of real estate price variation based on economic parameters, Li Li ; Kai-Hsuan Chu, 2017 International Conference on Applied System Innovation (ICASI)
8. Predicting house sale price using fuzzy logic, Artificial Neural Network and K-Nearest Neighbor, Muhammad Fahmi Mukhlishin ; Ragil Saputra ; Adi Wibowo, 2017 1st International Conference on Informatics and Computational Sciences (ICICoS)
9. Comprehensive Analysis of Housing Price Prediction in Pune Using Multi-Featured Random Forest Approach, Rushab Sawant ; Yashwant Jangid ; Tushar Tiwari ; Saurabh Jain ; Ankita Gupta, 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)
- 10.** Time-Aware Latent Hierarchical Model for Predicting House Prices, Fei Tan ; Chaoran Cheng ; Zhi Wei, 2017 IEEE International Conference on Data Mining (ICDM)

Model Implementation:

Data Science Regression Project: Predicting Home Prices in Pakistan

Dataset is downloaded from here: <https://www.kaggle.com/datasets/jillanisoftech/pakistanhouse-price-dataset>

```
[76]: import pandas as pd

from matplotlib import pyplot as plt
%matplotlib inline
import matplotlib
matplotlib.rcParams["figure.figsize"] = (20, 10)

import numpy as np
```

Data Load: Load pakistan home prices into a dataframe

```
[77]: df1 = pd.read_csv("pakistan_house_prices.csv")
df1 . head()
```

C:\Users\waqas\AppData\Local\Temp\ipykernel_15996\2334963485.py:1: DtypeWarning: Columns (0,1,3,4,5) have mixed types. Specify dtype option on import or set low_memory=False. df1 = pd.read_csv("pakistan_house_prices.csv")

```
[77]:
```

	area_type	availability	location	size	society \
0	Super built-up Area	19-DecG-10	2 BHK Coomee		
1	Plot Area	Ready To Move	E-11 4 Bedroom Theanmp		
2	Built-up Area	Ready To Move	G-15	3 BHK	NaN
3	Super built-up Area	Ready To Move	Bani Gala	3 BHK	Soiewre
4	Super built-up Area	Ready To Move	DHA Defence	2 BHK	NaN

	total_sqft	bath	balcony	price
0	1056	2.0	1.0	39.07
1	2600	5.0	3.0	120.00
2	1440	2.0	3.0	62.00
3	1521	3.0	1.0	95.00
4	1200	2.0	1.0	51.00

```
[78]: df1 . shape
```

```
[78]: (168446, 9)
```

```
[79]: df1 . columns
```

```
[79]: Index(['area_type', 'availability', 'location', 'size', 'society',
        'total_sqft', 'bath', 'balcony', 'price'], dtype='object')
```

```
[80]: df1[ 'area_type' ] . unique()
```

```
[80]: array(['Super built-up Area', 'Plot Area', 'Built-up Area',
        'Carpet Area', nan], dtype=object)
```

```
[81]: df1[ 'area_type' ] . value_counts()
```

```
[81]: area_type
Super built-up Area    8790
Built-up Area
```

```

2418
Plot Area          2025
Carpet Area        87
Name: count, dtype: int64

```

Drop features that are not required to build our model

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

```
[82]: (168446, 5)
```

Data Cleaning: Handle NA values

```
[83]: df2.isnull().sum()
```

```
[83]: location 0 size 155142 total_sqft
      155126 bath 155199
      price 155126
      dtype: int64
```

```
[84]: df2.shape
```

```
[84]: (168446, 5)
```

```
[85]: df3 = df2.dropna()
      df3.isnull().sum()
```

```
[85]: location 0 size 0 total_sqft
      0
      bath 0 price 0
      dtype: int64
```

```
[86]: df3.shape
```

```
[86]: (13247, 5)
```

Feature Engineering

Add new feature(integer) for bhk (Bedrooms Hall Kitchen)

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

C:\Users\waqas\AppData\Local\Temp\ipykernel_15996\2716584372.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-versus-a-copy
df3['bhk'] = df3['size'].apply(lambda x: int(x.split(' ')[0]))

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

Explore total_sqft feature

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

```
[89]: 2+3
```

[89]: 5

```
[90]: df3[~df3['total_sqft'].apply(is_float)].head(10)
```

```
[90]:
```

	location	size	total_sqft	bath	price	bhk
30	E-11	4 BHK	2100 - 2850	4.0	186.000	4
122	Askari 13	4 BHK	3067 - 8156	4.0	477.000	4
137	Askari 13	2 BHK	1042 - 1105	2.0	54.005	2
165	Scheme 33	2 BHK	1145 - 1340	2.0	43.490	2
188	Malir	2 BHK	1015 - 1540	2.0	56.800	2
410	DHA Defence	1 BHK	34.46Sq. Meter	1.0	18.500	1
549	Clifton	2 BHK	1195 - 1440	2.0	63.770	2
648	Cantt 9 Bedroom		4125Perch	9.0	265.000	9
661	Valencia Housing Society	2 BHK	1120 - 1145	2.0	48.130	2
672	Bahria Town Rawalpindi	4 Bedroom	3090 - 5002	4.0	445.000	4

Above shows that total_sqft can be a range (e.g. 2100-2850). For such case we can just take average of min and max value in the range. There are other cases such as 34.46Sq. Meter which one can convert to square ft using unit conversion. I am going to just drop such corner cases to keep things simple

```
[91]: def convert_sqft_to_num (x):
      tokens = x.split( '-' )
      if len ( tokens ) == 2:
          return ( float ( tokens [ 0] ) + float ( tokens [ 1] )) / 2
      try:
          return float ( x )
      except:
          return None
```

```
[92]: df4 = df3 . copy()
      df4 . total_sqft = df4 . total_sqft . apply(convert_sqft_to_num)
      df4 = df4[df4 . total_sqft . notnull()]
      df4 . head( 2)
```

```
[92]:
```

	location	size	total_sqft	bath	price	bhk
0	G-10	2 BHK	1056.0	2.0	39.07	2
1	E-11	4 Bedroom	2600.0	5.0	120.00	4

For below row, it shows total_sqft as 2475 which is an average of the range 2100-2850

```
[93]: df4 . loc[ 30]
```

```
[93]: location      E-11
      size          4 BHK
      total_sqft    2475.0
      bath          4.0
      price         186.0
      bhk           4
      Name: 30, dtype: object
```

```
[94]: ( 2100+2850 ) / 2
```

```
[94]: 2475.0
```

Feature Engineering

Add new feature called price per square feet

```
[95]: df5 = df4 . copy()
      df5[ 'price_per_sqft' ] = df5[ 'price' ] * 100000 / df5[ 'total_sqft' ]
      df5 . head()
```

```
[95]:
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	G-10	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	E-11 4 Bedroom		2600.0	5.0	120.00	4	4615.384615
2	G-15	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Bani Gala	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	DHA Defence	2 BHK	1200.0	2.0	51.00	2	4250.000000

```
[96]: df5_stats = df5[ 'price_per_sqft' ] . describe()
      df5_stats
```

```
[96]: count      1.320100e+04
      mean      7.920566e+03
      std       1.067231e+05
      min       2.678298e+02
      25%       4.267782e+03
      50%       5.438066e+03
      75%       7.317073e+03
      max       1.200000e+07
      Name: price_per_sqft, dtype: float64
```

```
[97]: df5 . to_csv( "bhp.csv ",index =False)
```

Examine locations which is a categorical variable. We need to apply dimensionality reduction technique here to reduce number of locations

```
[98]: df5.location = df5.location.apply(lambda x: x.strip()) location_stats =
      df5['location'].value_counts(ascending=False) location_stats
```

```
[98]: location
```

```
      DHA Defence      1387
```

Bahria Town Rawalpindi	648
Gulshan-e-Iqbal Town	430
Gulistan-e-Jauhar	425
Bahria Town	344
...	
C-18	1
Dhoke Munshi Khan	1
Wazir Town	1
Islamabad Farm Houses	1
Orchard Scheme	1

Name: count, Length: 969, dtype: int64

```
[99]: location_stats . values . sum()
```

```
[99]: 13201
```

```
[100]: len ( location_stats[location_stats >10])
```

```
[100]: 196
```

```
[101]: len ( location_stats )
```

```
[101]: 969
```

```
[102]: len ( location_stats[location_stats <=10])
```

```
[102]: 773
```

Dimensionality Reduction

Any location having less than 10 data points should be tagged as “other” location. This way number of categories can be reduced by huge amount. Later on when we do one hot encoding, it will help us with having fewer dummy columns

```
[103]: location_stats_less_than_10 = location_stats[location_stats<=10] location_stats_less_than_10
```

```
[103]: location
      Sea View Apartments      10
      Ghaziabad             10
      9th Avenue            10
      Bahria Nasheman    10 Manawan  10
      ..
      C-18                  1
      Dhoke Munshi Khan     1
      Wazir Town            1
      Islamabad Farm Houses 1
      Orchard Scheme        1
      Name: count, Length: 773, dtype: int64
```

```
[104]: len ( df 5. location . unique())
```

```
[104]: 969
```

```
[105]: df5.location = df5.location.apply(lambda x: 'other' if x in_
      location_stats_less_than_10 else x) len(df5.location.unique())
```

```
[105]: 197
```

```
[106]: df5 . head( 10)
```

```
[106]:
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	G-10	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	E-11	4 Bedroom	2600.0	5.0	120.00	4	4615.384615
2	G-15	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Bani Gala	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	DHA Defence	2 BHK	1200.0	2.0	51.00	2	4250.000000
5	Ghauri Town	2 BHK	1170.0	2.0	38.00	2	3247.863248
6	Korang Town	4 BHK	2732.0	4.0	204.00	4	7467.057101
7	E-11	4 BHK	3300.0	4.0	600.00	4	18181.818182
8	DHA Defence	3 BHK	1310.0	3.0	63.25	3	4828.244275
9	F-11	6 Bedroom	1020.0	6.0	370.00	6	36274.509804

Outlier Removal Using Business Logic

As a data scientist when you have a conversation with your business manager (who has expertise in real estate), he will tell you that normally square ft per bedroom is 300 (i.e. 2 bhk apartment is minimum 600 sqft. If you have for example 400 sqft apartment with 2 bhk than that seems suspicious and can be removed as an outlier. We will remove such outliers by keeping our minimum threshold per bhk to be 300 sqft

```
[107]: df5[df5 . total_sqft / df5 . bhk <300] . head()
```

```
[107]:
```

	location	size	total_sqft	bath	price	bhk	price_per_sqft
9	F-11	6 Bedroom	1020.0	6.0	370.0	6	36274.509804
45	E-11	8 Bedroom	600.0	9.0	200.0	8	33333.333333
58	other	6 Bedroom	1407.0	4.0	150.0	6	10660.980810
68	Chungi Amar Sadhu	8 Bedroom	1350.0	7.0	85.0	8	6296.296296
70	Askari	3 Bedroom	500.0	3.0	100.0	3	20000.000000

Check above data points. We have 6 bhk apartment with 1020 sqft. Another one is 8 bhk and total sqft is 600. These are clear data errors that can be removed safely

```
[108]: df5 . shape
```

```
[108]: (13201, 7)
```

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

```
[109]: (12457, 7)
```

Outlier Removal Using Standard Deviation and Mean

```
[110]: df6 . price_per_sqft . describe()
```

```
[110]: count    12457.000000
mean         6308.427888
std          4167.968413
min           267.829813
25%          4210.526316
50%          5294.117647
75%          6916.666667
max         176470.588235
Name: price_per_sqft, dtype: float64
```

Here we find that min price per sqft is 267 rs/sqft whereas max is 12000000, this shows a wide variation in property prices. We should remove outliers per location using mean and one standard deviation

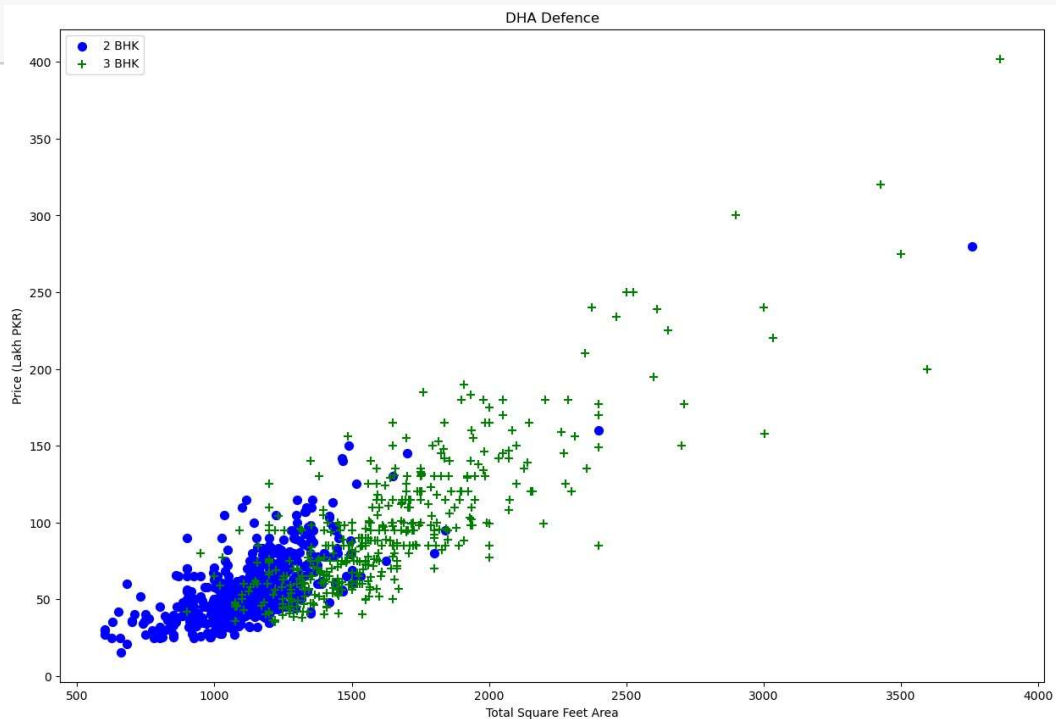
```
[111]: def remove_pps_outliers (df):
df_out = pd.DataFrame()
for key, subdf in df.groupby( 'location '):
m = np.mean(subdf . price_per_sqft)
st = np.std(subdf . price_per_sqft)
reduced_df = subdf[(subdf . price_per_sqft > (m-st)) & ( subdf .
price_per_sqft <= ( m+st))]
df_out = pd.concat([df_out,reduced_df],ignore_index =True)
return df_out
df7 = remove_pps_outliers(df6)
df7 . shape
```

[111]: (10862, 7)

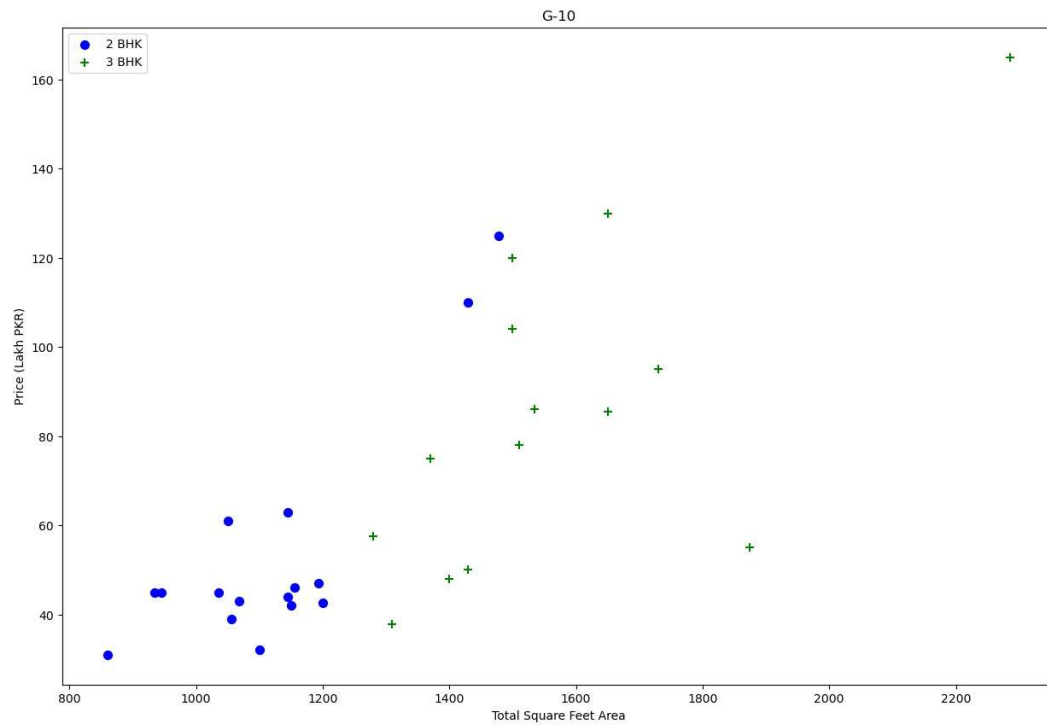
Let's check if for a given location how does the 2 BHK and 3 BHK property prices look like

```
[112]: 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 (Lakh PKR)")
plt.title(location) plt.legend()

plot_scatter_chart(df7,"DHA Defence")
```



```
[113]: plot_scatter_chart(df7, "G-10")
```



We should also remove properties where for same location, the price of (for example) 3 bedroom apartment is less than 2 bedroom apartment (with same square ft area). What we will do is for a given location, we will build a dictionary of stats per bhk, i.e.

```
{
    '1': {
        'mean': 4000,
        'std': 2000,
        'count': 34
    },
    '2': {
        'mean': 4300,
        'std': 2300,
        'count': 22
    },
}
```

Now we can remove those 2 BHK apartments whose price_per_sqft is less than mean

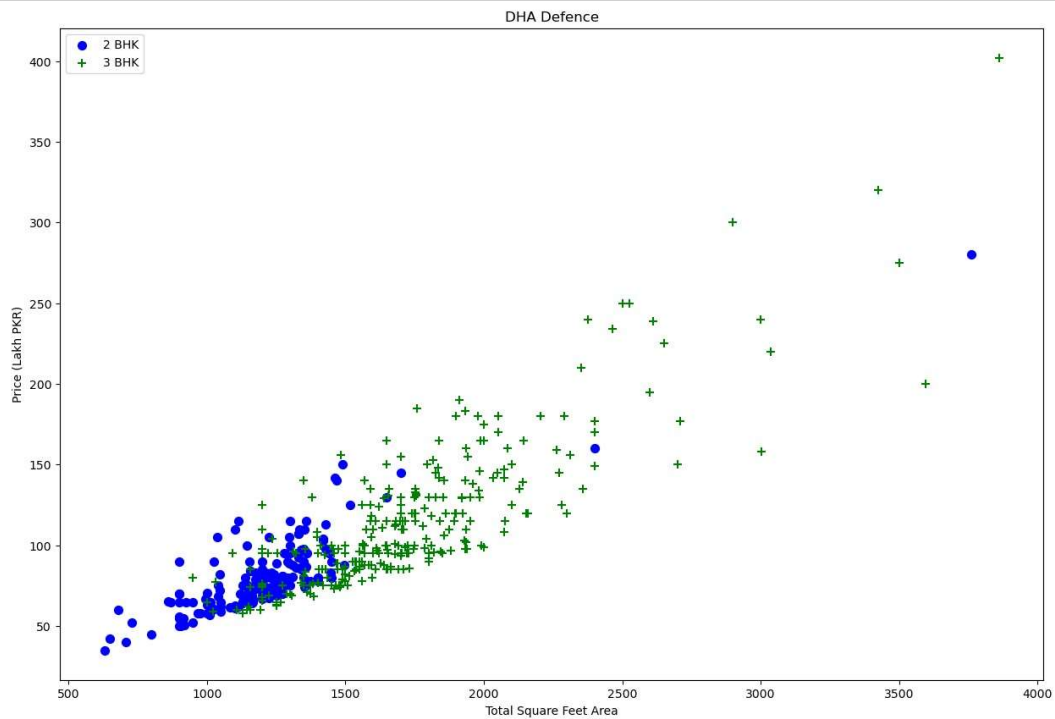
price_per_sqft of 1 BHK apartment

```
[114]: def remove_bhk_outliers (df):
        exclude_indices = 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_indices, bhk_df[bhk_df
                    .price_per_sqft < (stats [ 'mean' ])] .index .values)
            return df . drop(exclude_indices,axis = 'index ' )
df8 = remove_bhk_outliers(df7)
# df8 = df7.copy()
df8 . shape
```

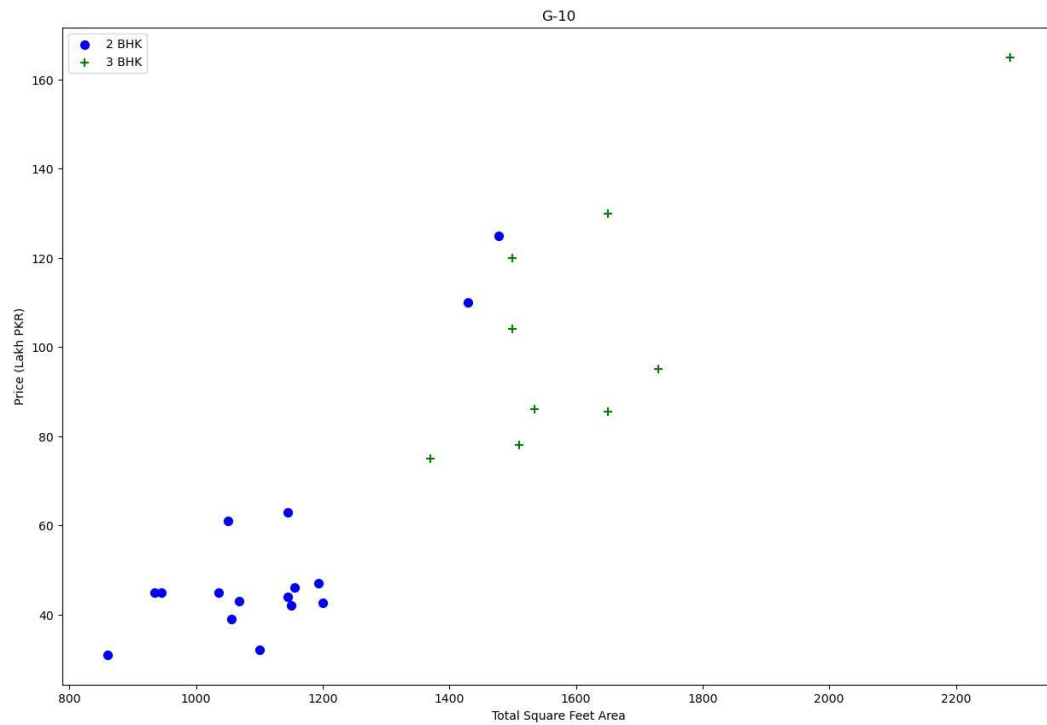
[114]: (7069, 7)

Plot same scatter chart again to visualize price_per_sqft for 2 BHK and 3 BHK properties

```
[115]: plot_scatter_chart(df8, "DHA Defence ")
```



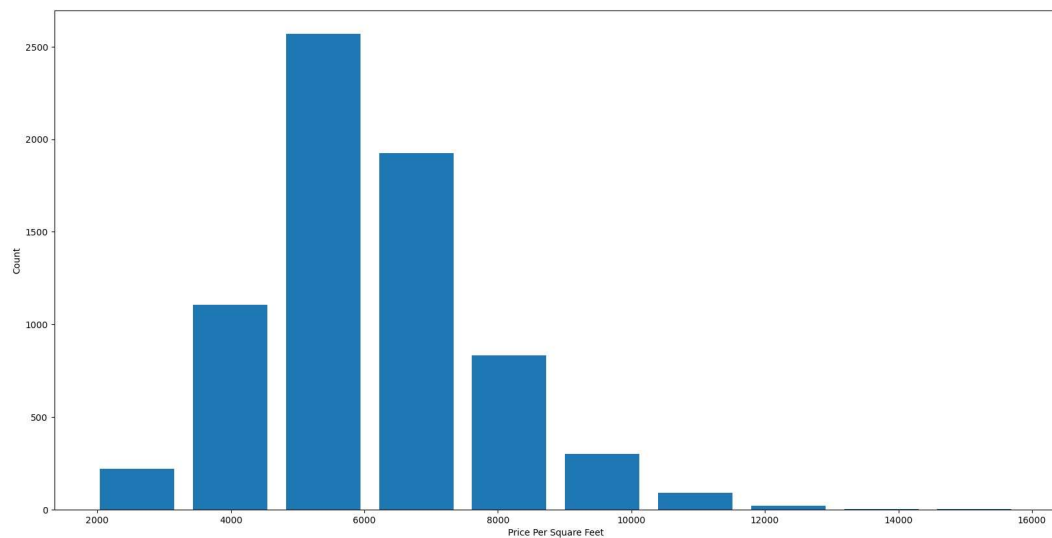
```
[116]: plot_scatter_chart(df8, "G-10 ")
```



Based on above charts we can see that data points highlighted in red below are outliers and they are being removed due to remove_bhk_outliers function

```
[117]: import matplotlib
matplotlib.rcParams["figure.figsize"] = (20, 10)
plt.hist(df8.price_per_sqft, rwidth=0.8)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")
```

```
[117]: Text(0, 0.5, 'Count')
```



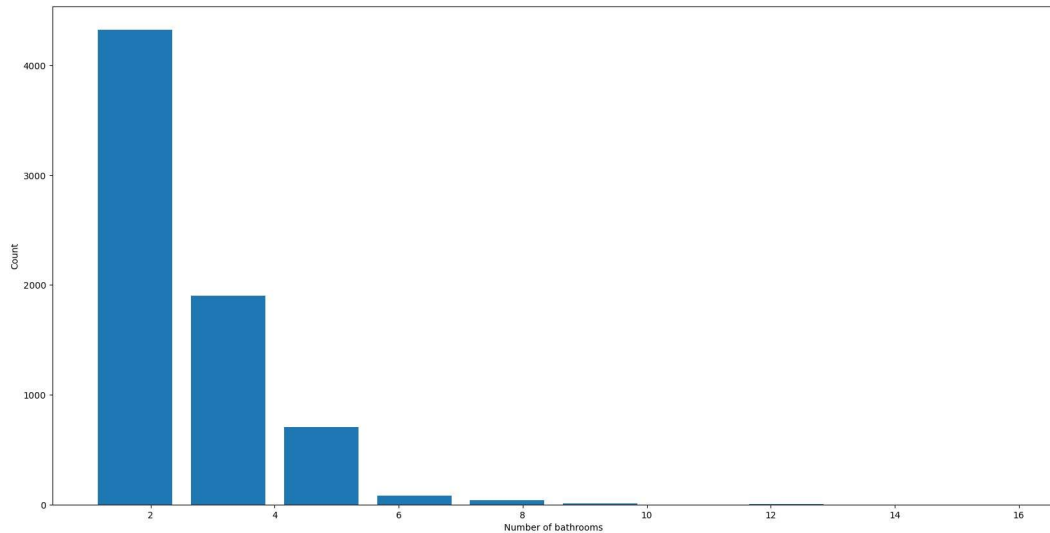
Outlier Removal Using Bathrooms Feature

```
[118]: df8 . bath . unique()
```

```
[118]: array([ 2., 3., 4., 1., 5., 8., 6., 12., 7., 9., 13., 16.]
```

```
[119]: plt . hist(df8 . bath,rwidth =0.8 )
plt . xlabel( " Number of bathrooms " )
plt . ylabel( " Count " )
```

```
[119]: Text(0, 0.5, 'Count')
```



```
[120]: df8[df8 . bath >10]
```

```
[120]:
```

	location	size	total_sqft	bath	price	bhk \
356	Askari 10 BHK	12000.0	12.0	525.0	10	
1483	Bahria Town Rawalpindi 13 BHK	5425.0	13.0	275.0	13	
4778	G-11 10 BHK	4000.0	12.0	160.0	10	
4936	G-15 16 BHK	10000.0	16.0	550.0	16	

```

price_per_sqft
356 4375.000000 1483
5069.124424
4778 4000.000000
4936 5500.000000

```

It is unusual to have 2 more bathrooms than number of bedrooms in a home

```
[121]: df8[df8 . bath >df8 . bhk +2]
```

```
[121]:
```

	location	size	total_sqft	bath	price	bhk \
545	B-17 3 BHK	1806.0	6.0	116.0	3	
3255	DHA Defence 4 Bedroom	7000.0	8.0	450.0	4	
5580	Gulistan-e-Jauhar 6 BHK	11338.0	9.0	1000.0	6	

```

            price_per_sqft
545 6423.034330 3255
6428.571429
5580      8819.897689

```

Again the business manager has a conversation with you (i.e. a data scientist) that if you have 4 bedroom home and even if you have bathroom in all 4 rooms plus one guest bathroom, you will have total bath = total bed + 1 max. Anything above that is an outlier or a data error and can be removed

```

[122]: df9 = df8[df8 . bath <df8 . bhk +2]
df9 . shape

```

```

[122]: (6993, 7)

```

```

[123]: df9 . head( 2)

```

```

[123]: location      size total_sqft bath price bhk price_per_sqft 0 7th Avenue 2 BHK
      1089.0 2.0 43.55      2      3999.081726
      1 7th Avenue 3 BHK      1700.0      3.0 95.00      3      5588.235294

```

```

[124]: df10 = df9 . drop([ ' size ', ' price_per_sqft ' ],axis =' columns ' )
df10 . head( 3)

```

```

[124]: location total_sqft bath price bhk 0 7th Avenue 1089.0
      2.0 43.55 2
      1 7th Avenue      1700.0 3.0 95.00      3
      2 7th Avenue      1500.0 2.0 88.00      2

```

Use One Hot Encoding For Location

```

[125]: dummies = pd. get_dummies(df10 . location)
dummies. head( 3)

```

```

[125]:      7th Avenue Aashiana Road Abul Hassan Isphani Road Adiala Road \
0      True      False False False
1      True      False False False
2      True      False False False

```

```

      Afshan Colony Airport Housing Society Al Rehman Garden \
0      False False False False
1      False False False False
2      False False False False

```

```

      Alfalah Town Ali Pur ... Thokar Niaz Baig Township University Road \
0      False False ...      False False False
1      False False ...      False False False
2      False False ...      False False False

```

```

      Valencia Housing Society Walton Road Wapda Town Wassanpura Westridge \
0      False      False      False      False      False
1      False      False      False      False      False
2      False      False      False      False      False

```

```
0
1
   Zamzama other False
   False
2   False False
```

[3 rows x 197 columns]

```
[126]: df11 = pd.concat([df10,dummies.drop('other',axis='columns')],axis='columns') df11.head()
```

```
[126]:      location total_sqft bath price bhk 7th Avenue Aashiana Road \
0 7th Avenue      1089.0    2.0 43.55    2      True      False
1 7th Avenue      1700.0    3.0 95.00    3      True      False
2 7th Avenue      1500.0    2.0 88.00    2      True      False
3 7th Avenue      1020.0    2.0 48.00    2      True      False
4 7th Avenue      1007.0    2.0 67.00    2      True      False
   Abul Hassan Isphani Road Adiala Road Afshan Colony ... Tariq Road \
0      False      False      False ...      False
1      False      False      False ...      False
2      False      False      False ...      False
3      False      False      False ...      False
4      False      False      False ...      False
   Thokar Niaz Baig Township University Road Valencia Housing Society \
0      False      False      False      False      False
1      False      False      False      False      False
2      False      False      False      False      False
3      False      False      False      False      False
4      False      False      False      False      False
   Walton Road Wapda Town Wassanpura Westridge Zamzama
0      False      False      False      False      False
1      False      False      False      False      False
2      False      False      False      False      False
3      False      False      False      False      False
4      False      False      False      False      False
```

[5 rows x 201 columns]

```
[127]: df12 = df11 . drop( ' location ' ,axis = ' columns ' )
df12 . head( 2 )
```

```
[127]:      total_sqft bath price bhk 7th Avenue Aashiana Road \
0      1089.0 2.0 43.55    2      True  False
1      1700.0 3.0 95.00    3      True  False
   Abul Hassan Isphani Road Adiala Road Afshan Colony Airport ... \
```

```

0
1
False False False False ...
False False False False ...

```

```

Tariq Road Thokar Niaz Baig Township University Road \
0 False False False False
1 False False False False

```

```

Valencia Housing Society Walton Road Wapda Town Wassanpura Westridge \
0 False False False False False
1 False False False False False

```

```

Zamzama
0 False
1 False

```

[2 rows x 200 columns]

Build a Model Now...

```
[128]: df12 . shape
```

```
[128]: (6993, 200)
```

```
[129]: X = df12 . drop([ ' price ' ],axis =' columns ' )
X.head( 3)
```

```
[129]:
total_sqft bath bhk 7th Avenue Aashiana Road Abul Hassan Isphani Road \
0 1089.0 2.0 2 True False False
1 1700.0 3.0 3 True False False
2 1500.0 2.0 2 True False False

```

```

Adiala Road Afshan Colony Airport Airport Housing Society ... \
0 False False False False ...
1 False False False False ...
2 False False False False ...

```

```

Tariq Road Thokar Niaz Baig Township University Road \
0 False False False False
1 False False False False
2 False False False False

```

```

Valencia Housing Society Walton Road Wapda Town Wassanpura Westridge \
0 False False False False False
1 False False False False False
2 False False False False False

```

```
0
1
   Zamzama
   False
   False
2   False
```

```
[3 rows x 199 columns]
```

```
[130]: X.shape
```

```
[130]: (6993, 199)
```

```
[131]: y = df12 . price
        y.head( 3)
```

```
[131]: 0 43.55 1
        95.00
        2   88.00
        Name: price, dtype: float64
```

```
[132]: len ( y)
```

```
[132]: 6993
```

```
[133]: 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)
```

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

```
[134]: 0.8084274685808824
```

Use K Fold cross validation to measure accuracy of our LinearRegression model

```
[135]: 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)
```

```
[135]: array([0.77667826, 0.80828867, 0.80263425, 0.8117566 , 0.77330778])
```

We can see that in 5 iterations we get a score above 80% all the time. This is pretty good but we want to test few other algorithms for regression to see if we can get even better score. We will use GridSearchCV for this purpose

Find best model using GridSearchCV

```
[136]: from sklearn.model_selection import GridSearchCV, ShuffleSplit
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.tree import DecisionTreeRegressor
import pandas as pd

def find_best_model_using_gridsearchcv(X, y):
    algos = {
        'linear_regression': {
            'model': LinearRegression(),
            'params': {
                'fit_intercept': [True, False],
                'copy_X': [True, False],
                'positive': [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_
        })

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

# Assuming you have X and y defined somewhere
# Call the function to find the best model
find_best_model_using_gridsearchcv(X, y)
```

C:\Users\waqas\anaconda3\Lib\sitepackages\sklearn\model_selection_validation.py:378:
FitFailedWarning:

10 fits failed out of a total of 20.

The score on these train-test partitions for these parameters will be set to nan.

If these failures are not expected, you can try to debug them by setting `error_score='raise'`.

Below are more details about the failures:

-----10 fits failed with the following error:

Traceback (most recent call last):

```
File "C:\Users\waqas\anaconda3\Lib\sitepackages\sklearn\model_selection\_validation.py", line 686, in _fit_and_score estimator.fit(X_train, y_train, **fit_params)
```

```
File "C:\Users\waqas\anaconda3\Lib\site-packages\sklearn\tree\_classes.py", line 1247, in fit super().fit()
```

```
File "C:\Users\waqas\anaconda3\Lib\site-packages\sklearn\tree\_classes.py", line 177, in fit self._validate_params()
```

```
File "C:\Users\waqas\anaconda3\Lib\site-packages\sklearn\base.py", line 600, in _validate_params validate_parameter_constraints()
```

```
File "C:\Users\waqas\anaconda3\Lib\sitepackages\sklearn\utils\_param_validation.py", line 97, in validate_parameter_constraints raise InvalidParameterError(
```

sklearn.utils._param_validation.InvalidParameterError: The 'criterion' parameter of DecisionTreeRegressor must be a str among {'squared_error', 'poisson', 'absolute_error', 'friedman_mse'}. Got 'mse' instead.

```
warnings.warn(some_fits_failed_message, FitFailedWarning)
```

```
C:\Users\waqas\anaconda3\Lib\sitepackages\sklearn\model_selection\_search.py:952:
```

```
UserWarning: One or more of the test scores are non-finite: [      nan      nan 0.66258837 0.71582284] warnings.warn(
```

```
[136]:      model best_score \
0      linear_regression 0.794533
1      lasso            0.791111
2      decision_tree     0.715823
```

```
best_params
0      {'copy_X': True, 'fit_intercept': True, 'posit...
1      {'alpha': 2, 'selection': 'random'}
2      {'criterion': 'friedman_mse', 'splitter': 'ran...
```

Based on above results we can say that LinearRegression gives the best score. Hence we will use that.

Test the model for few properties

```
[137]: def predict_price ( location,sqft,bath,bhk ):
loc_index = np.where(X.columns==location)[0][0]

x = np.zeros( len ( X.columns))
x[0] = sqft
x[1] = bath
x[2] = bhk
if loc_index >= 0:
    x[loc_index] = 1

return lr_clf . predict([x])[0]
```

```
[138]: predict_price( ' Bani Gala ', 1000, 2, 2)
```

C:\Users\waqas\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
warnings.warn(

[138]: 55.552545545794175

[139]: predict_price(' Korang Town ', 1000, 3, 3)

C:\Users\waqas\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
warnings.warn(

[139]: 44.75194098757295

[140]: predict_price(' F-7 ', 1000, 2, 2)

C:\Users\waqas\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
warnings.warn(

[140]: 45.75925059020437

[141]: predict_price(' Gulberg ', 1000, 3, 3)

C:\Users\waqas\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
warnings.warn(

[141]: 49.3274558199345

Export the tested model to a pickle file

[142]:

```
import pickle
with open('banglore_home_prices_model.pickle', 'wb') as f:
    pickle.dump(lr_clf, f)
```

Export location and column information to a file that will be useful later on in our prediction application

[143]:

```
import json
columns = {
    'data_columns': [col.lower() for col in X.columns]
}
with open("columns.json", "w") as f:
    f.write(json.dumps(columns))
```

[]: