UNIVERSITY OF MUMBAI

PROJECT REPORT ON

**REAL ESTATE PRICE PREDICTION**

**PROJECT**

SUBMITTED BY

ASHISH RAMJANAM MALLAH

UNDER THE GUIDANCE OF

PROF.BHANUDAS SATAM & AQUILA SHAIKH



LATE BHAUSAHEB HIRAY SMARNIKA SAMITI TRUST  
**HIRAY GROUP OF INSTITUTES**

MUMBAI - 400051

MAHARASHTRA

MCA SEM I [ 2020-2021]



LATE BHAUSAHEB HIRAY S.S. TRUST’S INSTITUTE OF COMPUTER APPLICATION

ISO 90012008 CERTIFIED

**S.N. 341, Next to New English School, Govt. Colony,**

**Bandra (East), Mumbai – 400051,**

**Tel: 91-22-26570892/3181**

# Date:

**CERTIFICATE**

**This is to certify that Mr. ASHISH RAMJANAM MALLAH**

**----------------------------------------------------------------Roll No. 202114**

**is a student of MCA of 1th year Semester-I has completed successfully full-semester Mini-Project of subject “REAL STATE PRICE PREDICTION” for the academic year 2020 – 21.**

----------------- ---------------

**Subject In-Charge Director**

--------------------

**External Examiner**

# PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

**PNR No.:-** 2017016400250142 **SEAT No.:-**202178

**Name of the Student:-** ASHISH RAMJANAM MALLAH

**Title of the Project:**- REAL ESTATE PRICE PREDICTION

**Name of the Guide:-** Prof. AQUILA SHAIKH & BHANUDAS SATAM

**Teaching experience of the Guide:**

Is this your first submission? Yes No

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Signature of the Student** **Signature of the Guide**

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Signature of the Coordinator:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ACKNOWLEDGEMENT**

I extend my deepest appreciation to my esteemed guide,

**Prof. AQUILA SHAIKH & BHANUDAS SATAM** for providing me with the possibility to complete this project with the right guidance and advice.

Special gratitude I give to my respected head of the division

**PROF. VIKRAM PATALBANSI**, for allowing me to use the facilities available and also help me to coordinate my project Furthermore, I would also like to acknowledge with much appreciation the crucial role of faculty members on this occasion.

Last but not least, I would like to thank friends who help me to assemble the parts and gave a suggestion about the project.

# ----------------------

**Abstract:-**

Real estate is the least transparent industry in our ecosystem. Housing prices keep changing day in and day out and sometimes are hyped rather than being based on valuation.

Predicting housing prices with real factors is the main crux of our research project. Here we aim to make our evaluations based on every basic parameter that is considered while determining the price.

We use various regression techniques in this pathway, and our results are not sole determination of one technique rather it is the weighted mean of various techniques to give 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 get exact real-world valuations.

## **TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CONTENTS** |  | **PAGE NO.** |
| **INTRODUCTION**  1)Introduction and Importance  1.1) Topic Intro  1.2) Need and Motivation |  |  |
| **DATA SET**  2)Steps in Preparing Data for Model  2.1) Data Load and Cleaning  2.2) Outlier Detection and Removal  2.3) Feature Engineering  2.4) Dimensionality Reduction  2.5)Grid Search Cv  2.6) K-fold cross validation |  |  |
| **LANGUAGE AND MODELS USED**  3.1) Python  3.2) Numpy and Pandas for data cleaning  3.3) Matplotlib for data visualization  3.4) Sklearn for model building  3.5) Jupyter notebook,visual studio code & pycharm as IDE  3.6) Python flask for http server  3.7)Using PostMan Application  3.8) HTML/CSS/Javascript for UI  **3B) Models Used**  3B.1) Multiple Linear Regression  3B.2) Decision-Tree-Regressor  3B.3) Lasso Regression |  |  |
| **4) RESULTS AND DISCUSSIONS**  4.1) BEST SUITED MODEL  4.2) DEPLOYMENT APP |  |  |
| 5) ABBREVIATIONS |  |  |
| 6) CONCLUSION |  |  |
| 7) REFERENCE |  |  |

LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| CONTENT |  | PAGE N0. |
| 4.1)Banglore Home Price Dataset |  |  |
| 4.2) Transformed Dataset House |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Contents** |  | **Page No**. |
| 1. Bar Plots of various Parameters |  |  |
| 2. Correlation Heatmap |  |  |
| 3. Skewed Plot |  |  |
| 4. Real Vs Predicted Plots |  |  |
| 5. Comparison Bar Plot |  |  |
| 6. Website View |  |  |

**INTRODUCTION**

Real Estate Price Prediction Project

Investment is a business activity on which most people are interested in this globalization era. There are several objects that are often used for investment, for example, gold, stocks and property. In particular, property investment has increased significantly.

Housing price trends are not only the concern of buyers and sellers, but it also indicates the current economic situation. There are many factors which has impact on house prices, such as numbers of bedrooms and bathrooms. Even the nearby location, a location with a great accessibility to highways, expressways, schools, shopping malls and local employment opportunities contributes to the rise in house price.

Manual house predication becomes difficult, hence there are many systems developed for house price prediction. We have proposed an advanced house prediction system using linear regression. This system aim is to make a model which can give us a good house pricing prediction based on other variables. We are going to use Linear Regression for this dataset and hence it gives a good accuracy.

This house price prediction project has two modules namely, Admin and User. Admin can add location and view the location. Admin has authority to add density on the basis of per unit area. User can view the location and see the predicted housing price for the particular location.

## **1.2.Need and Motivation:-**

Having lived in India for so many years if there is one thing that I had been taking for granted, it’s that housing and rental prices continue to rise. Since the housing crisis of 2008, housing prices have recovered remarkably well, especially in major housing markets. However, in the 4th quarter of 2016, I was surprised to read that Bombay housing prices had fallen the most in the last 4 years. In fact, median resale prices for condos and coops fell 6.3%, marking the first time there was a decline since Q1 of 2017.

The decline has been partly attributed to political uncertainty domestically and abroad and the 2014 election. So, to maintain the transparency among customers and also the comparison can be made easy through this model. If customer finds the price of house at some given website higher than the price predicted by the model, so he can reject that house.

**REQUIREMENTS**

* **Hardware Requirement:**
* Processor –Core i3
* Hard Disk – 160 GB
* Memory – 1GB RAM
* **Software Requirement:**
* Windows 7 or higher
* Python
* python flask server
* **Advantages**
* Saves time
* Easy to access the system anywhere and anytime.
* **Limitation**
* Requires an active internet connection.
* **Application**
* This system can be used by the multiple peoples to get the counselling sessions online.
* **Modules:**

The system comprises of 3 major modules with their sub-modules as follows:

1. **Admin:**
   * **Add Location:** Admin can add locations.
   * **View Location:** Admin can View the added location.
   * **Add Density:** Admin can add density of the houses by per unit area.
2. **User:**
   * **View Location:** User can view the location.
   * **View Predicted housing price:** User can view the predicted price of house.

**2.DATA SET**

**Steps in Preparing Data for Model**

* **STEPS:**

This data science project walks through step by step process of how to build a real estate price prediction website.

We will first build a model using sklearn and linear regression using banglore home prices dataset from kaggle.com.

Second step would be to write a python flask server that uses the saved model to serve http requests.

Third component is the website built in html, css and javascript that allows user to enter home square ft area, bedrooms etc and it will call python flask server to retrieve the predicted price.

During model building we will cover almost all **data science concepts** such as :-

2.1. Data load and cleaning,

2.2. Feature engineering,

2.3. Outlier detection and removal,

2.4. Dimensionality Reduction

2.5. Grid Search Cv

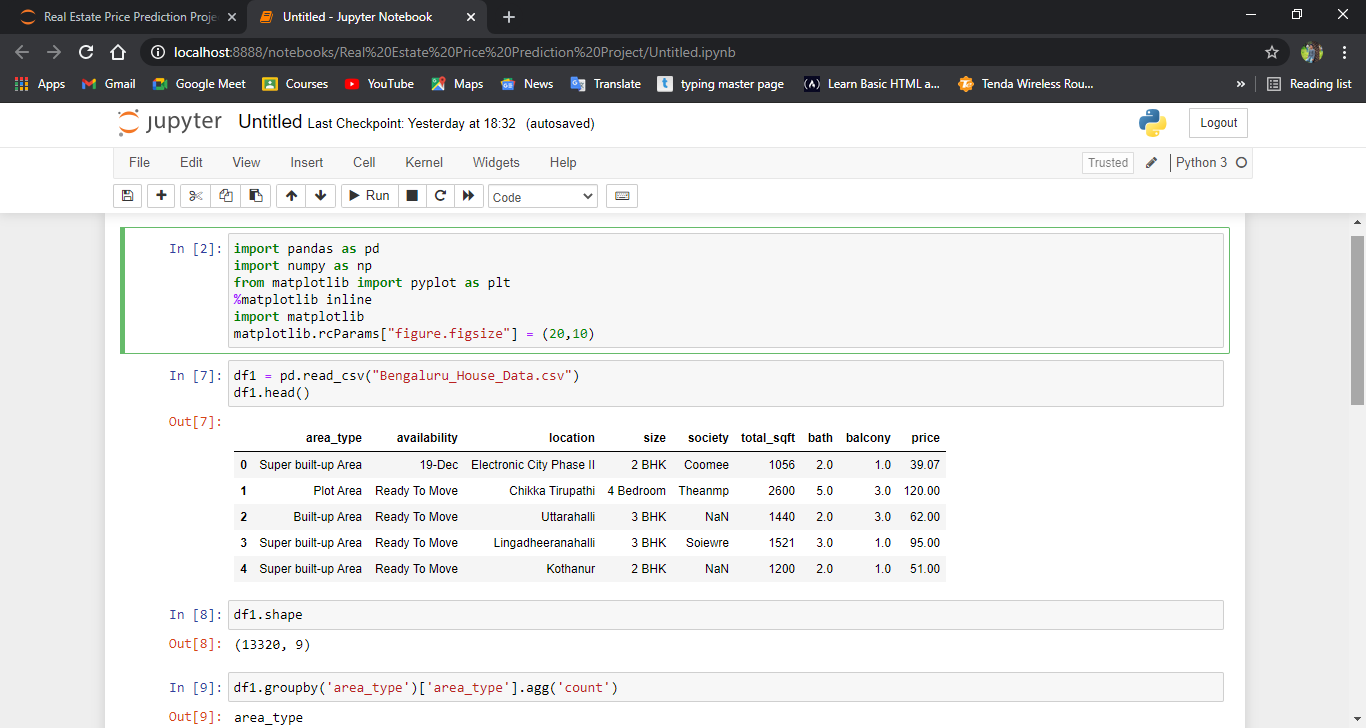
2.6. K-Fold Cross Validation

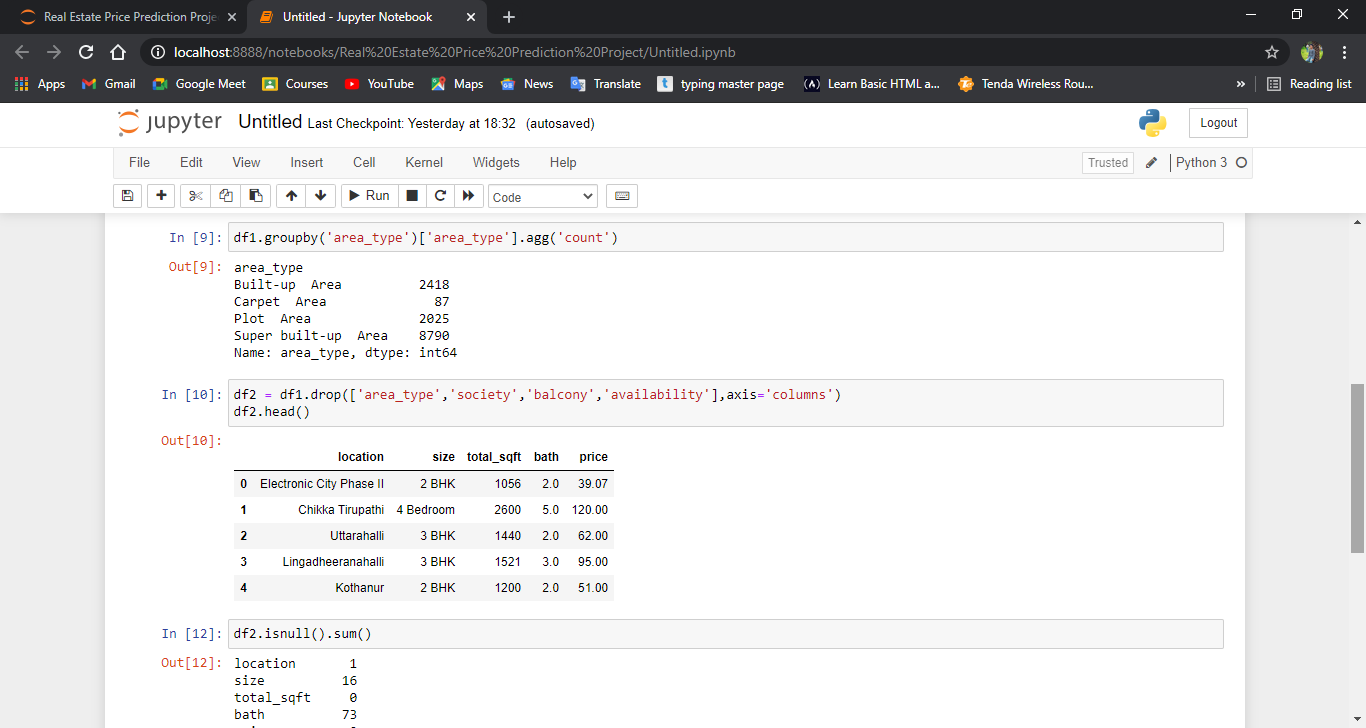
**2.1)Data Load and Cleaning:-**

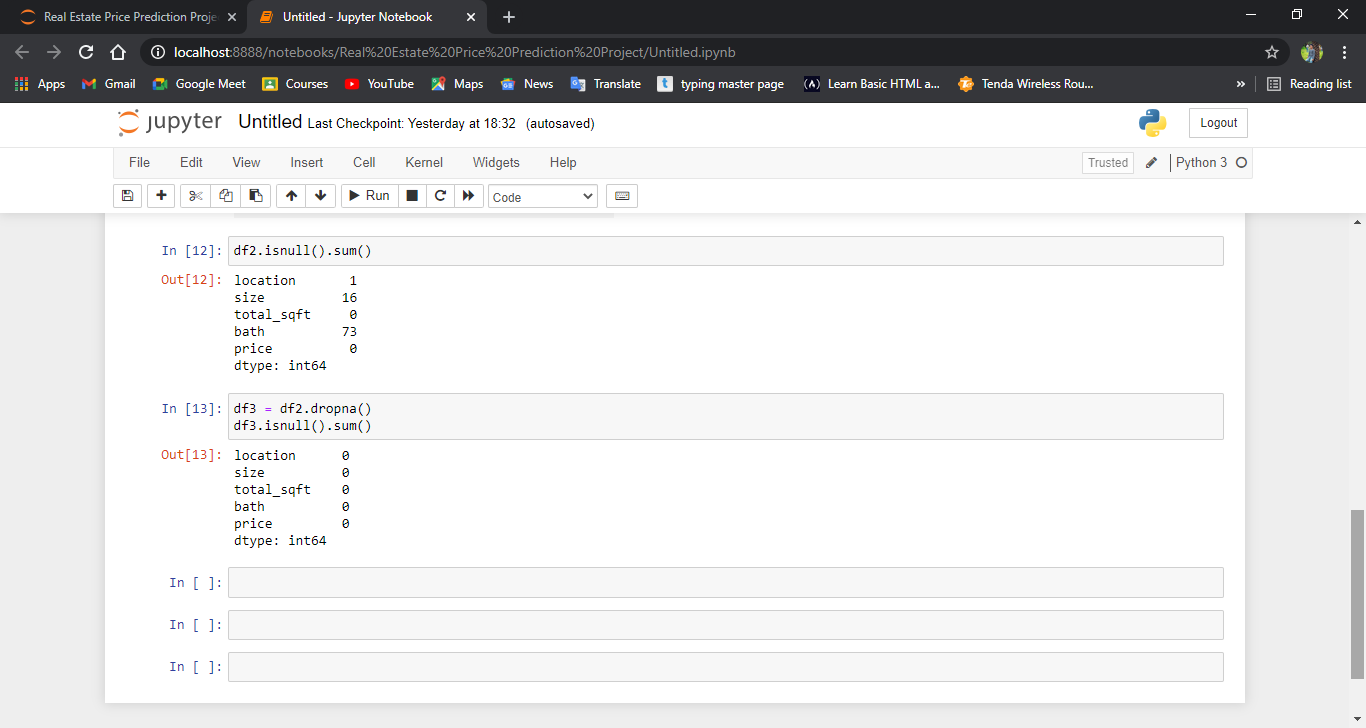
Data cleansing or data cleaning is the process of detecting and correcting corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data.

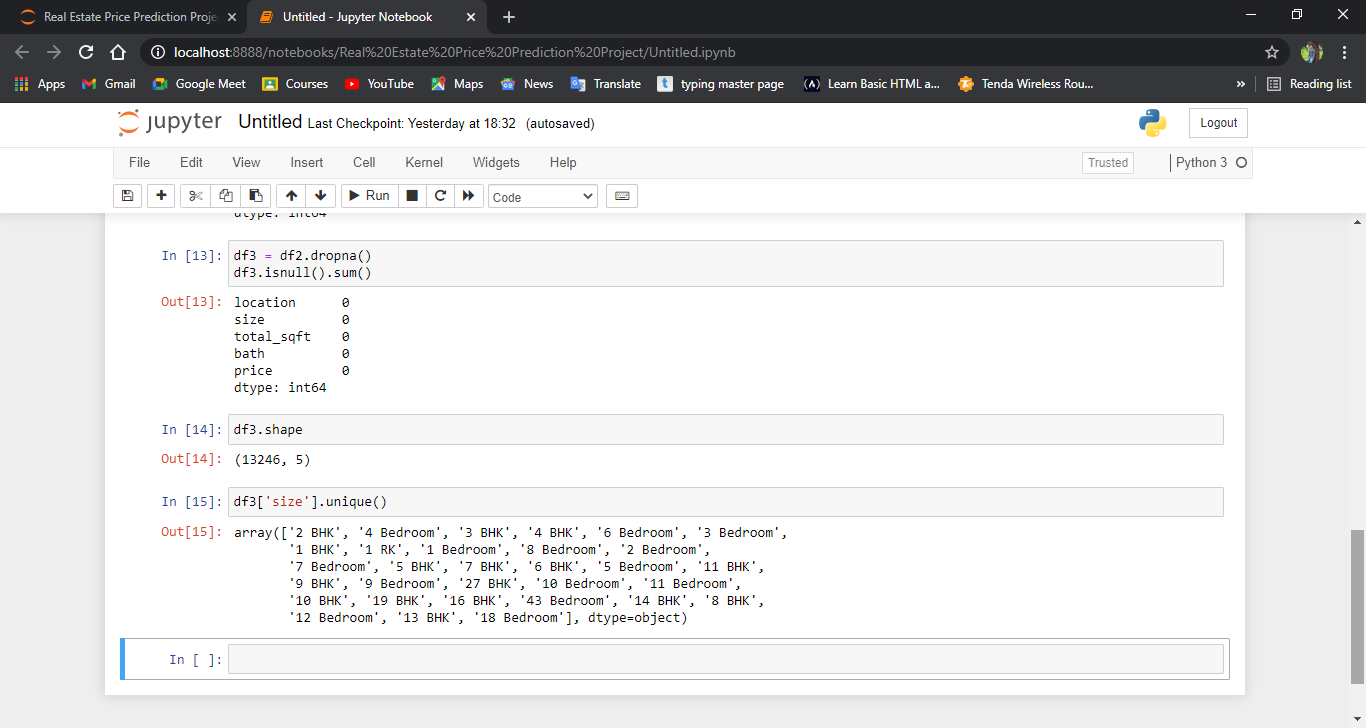
Data cleansing may be performed interactively with data wrangling tools, or as batch processing through scripting.

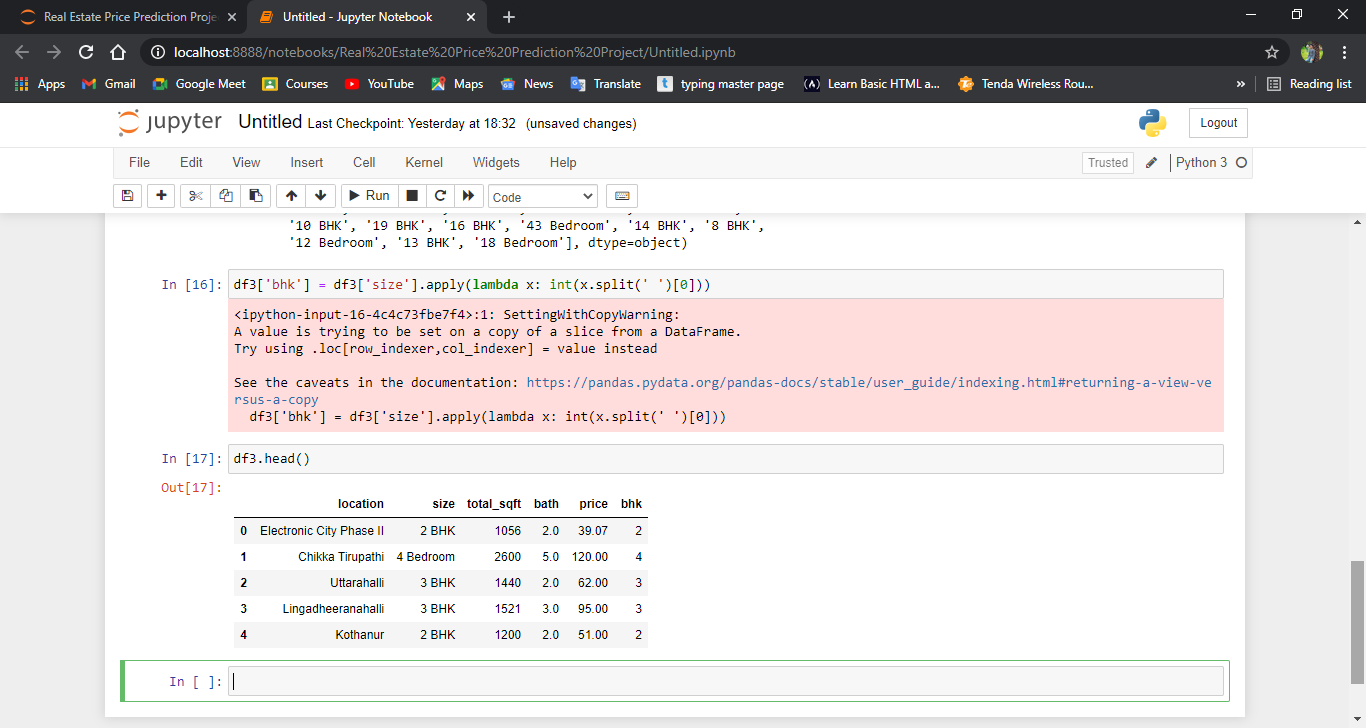
Data Cleaning Screen Shot :-

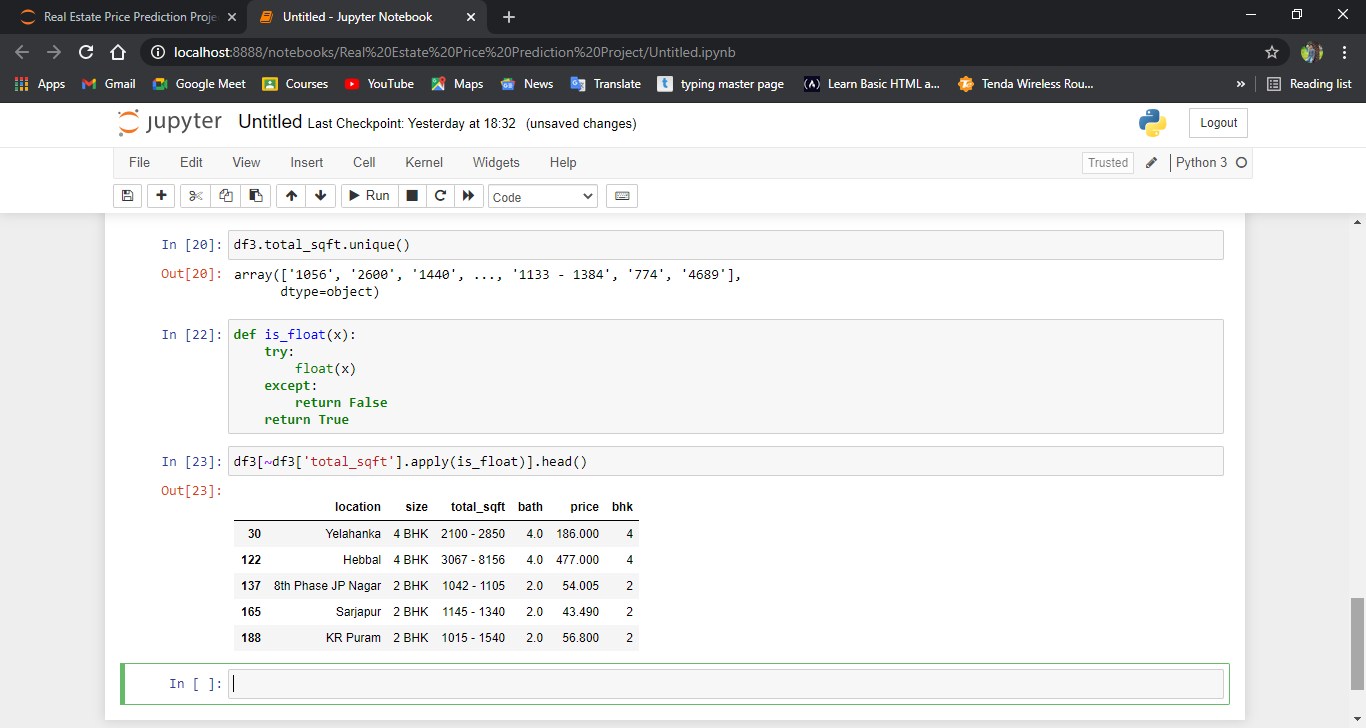


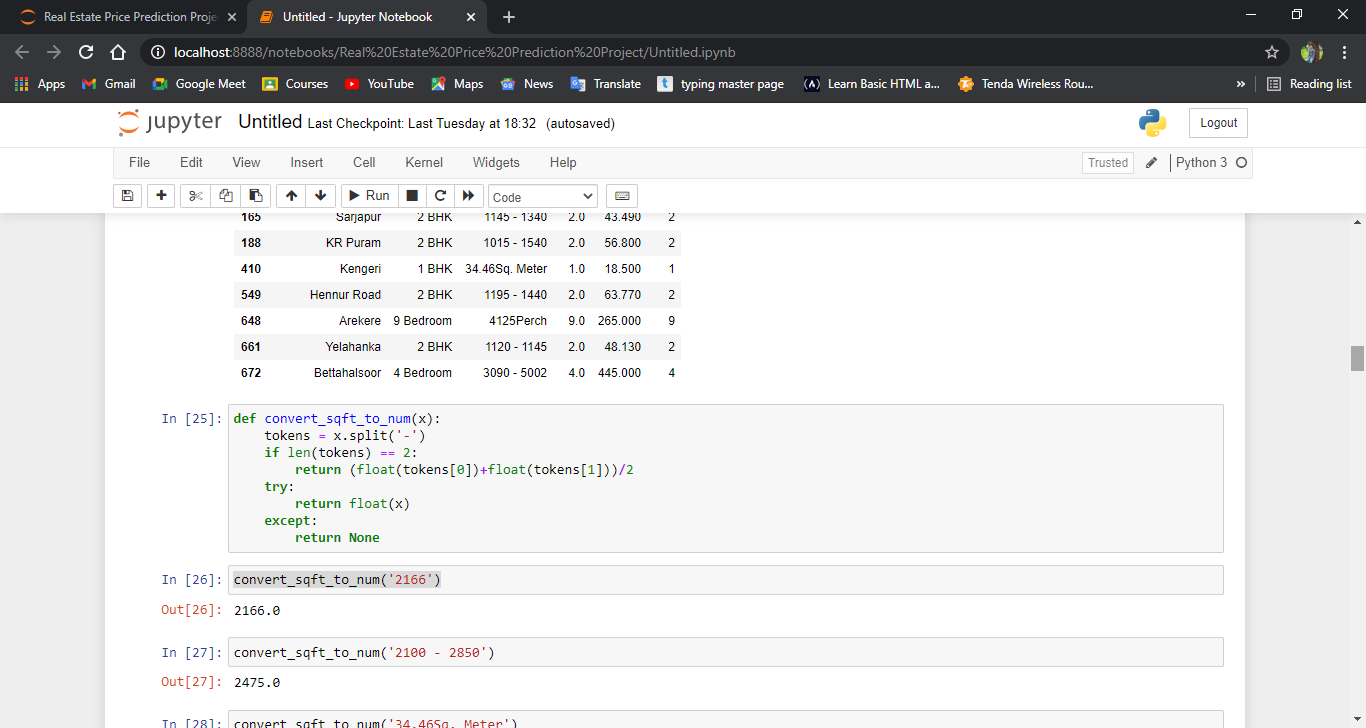


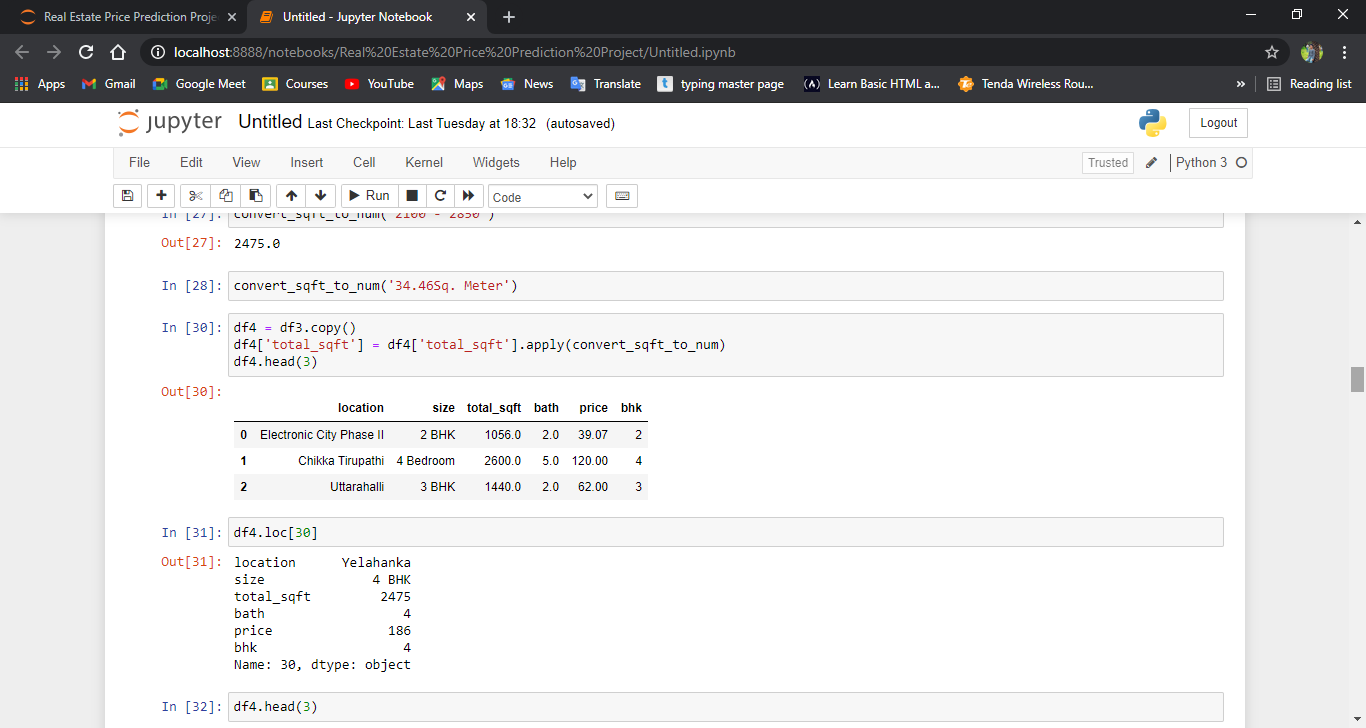










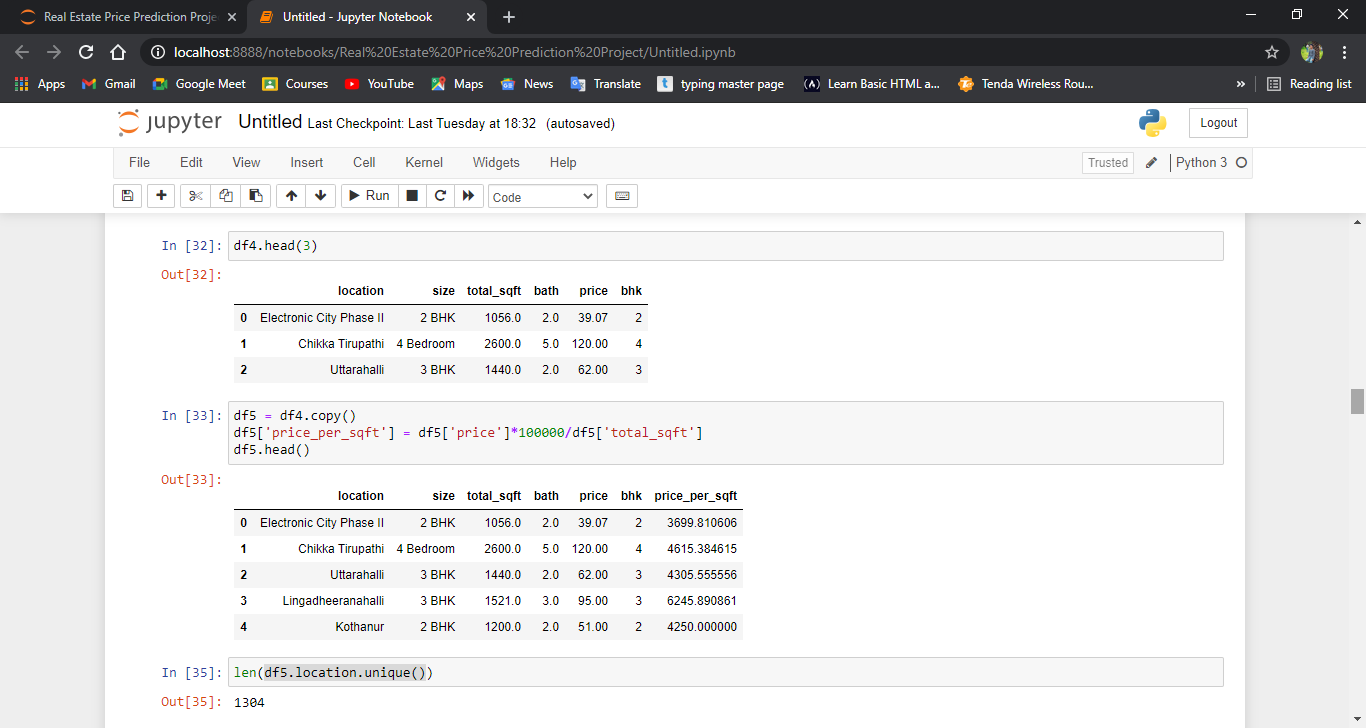


**2.2) Feature Engineering:-**

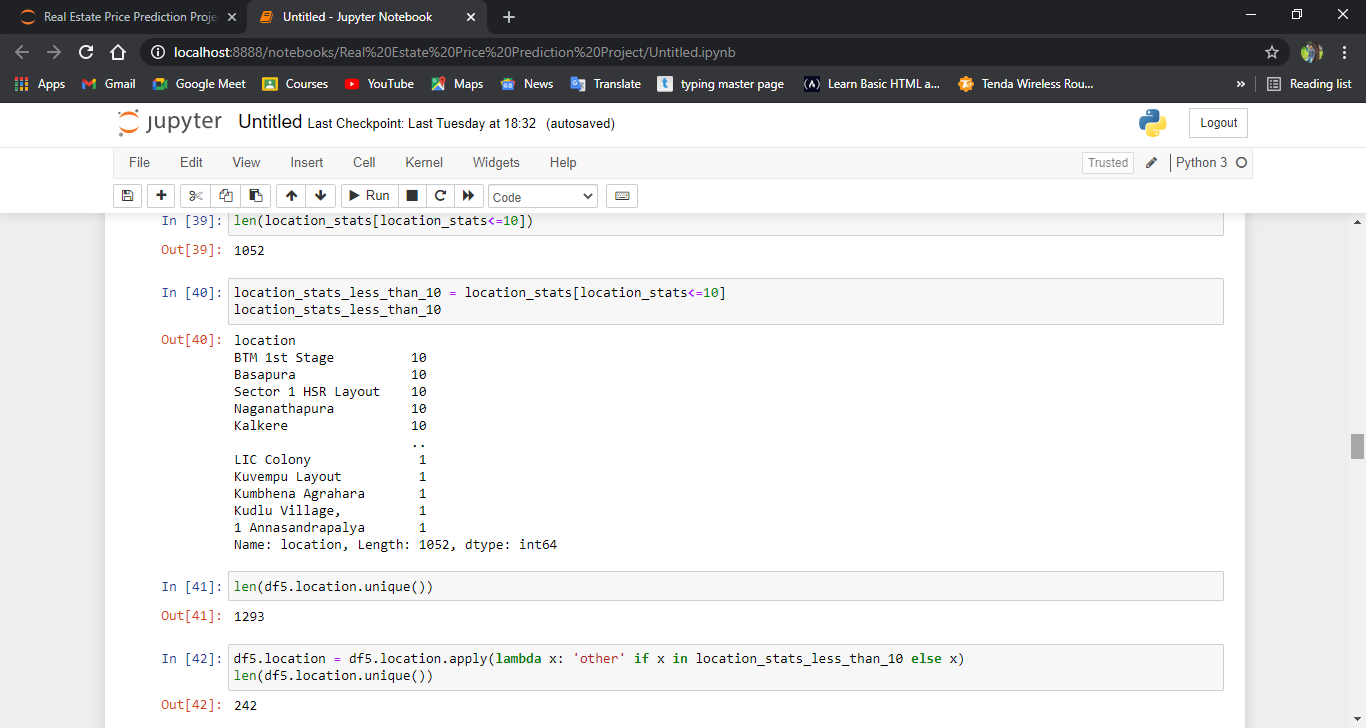
Feature engineering is**the process of using domain knowledge to extract features from raw data.**

These features can be used to improve the performance of machine learning algorithms. Feature engineering can be considered as applied machine learning itself.

**ScreenShot:-**







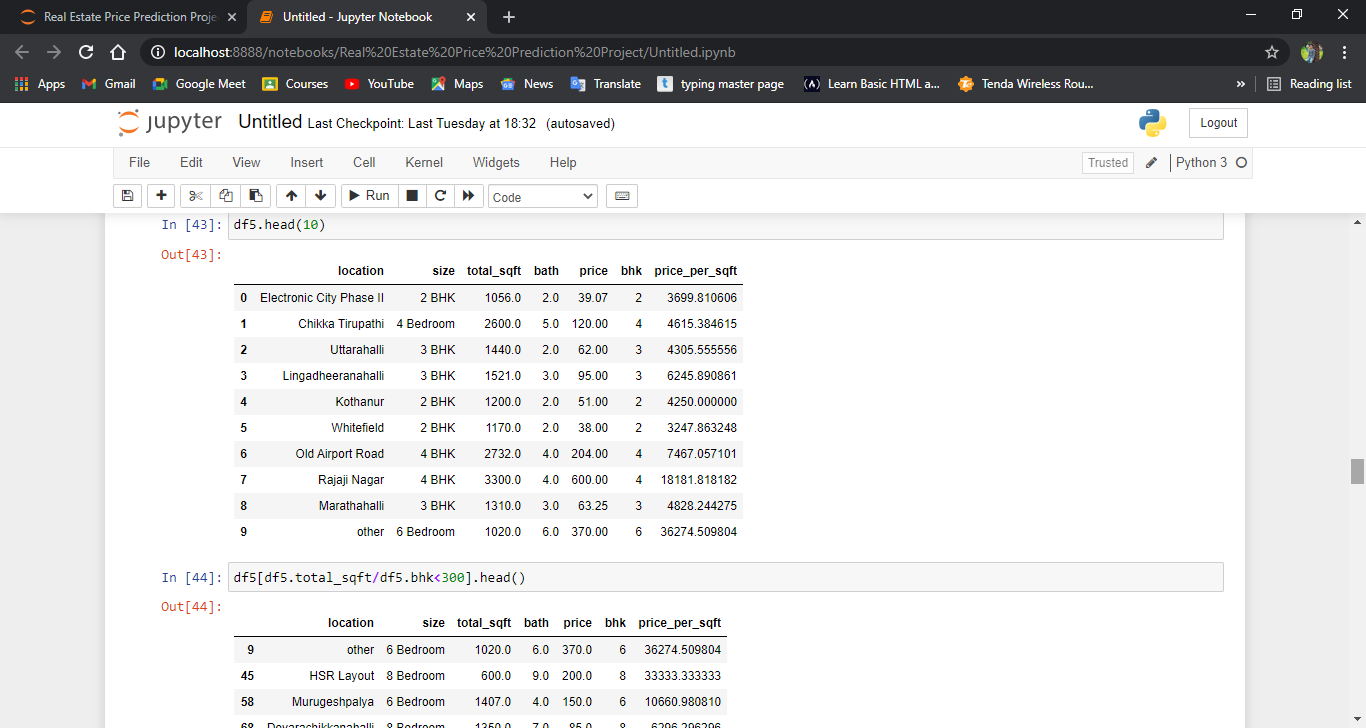
**2.3) Outlier detection and removal:-**

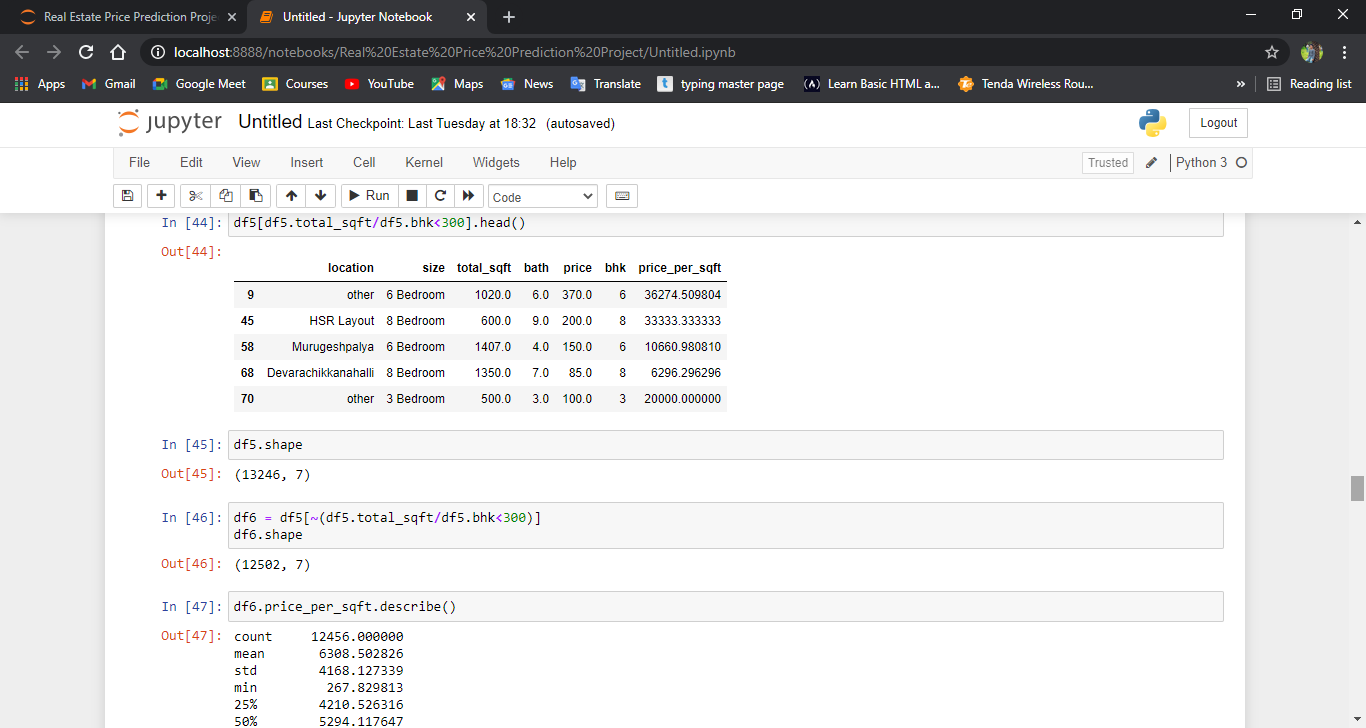
In statistics, an outlier is a data point that differs significantly from other observations.

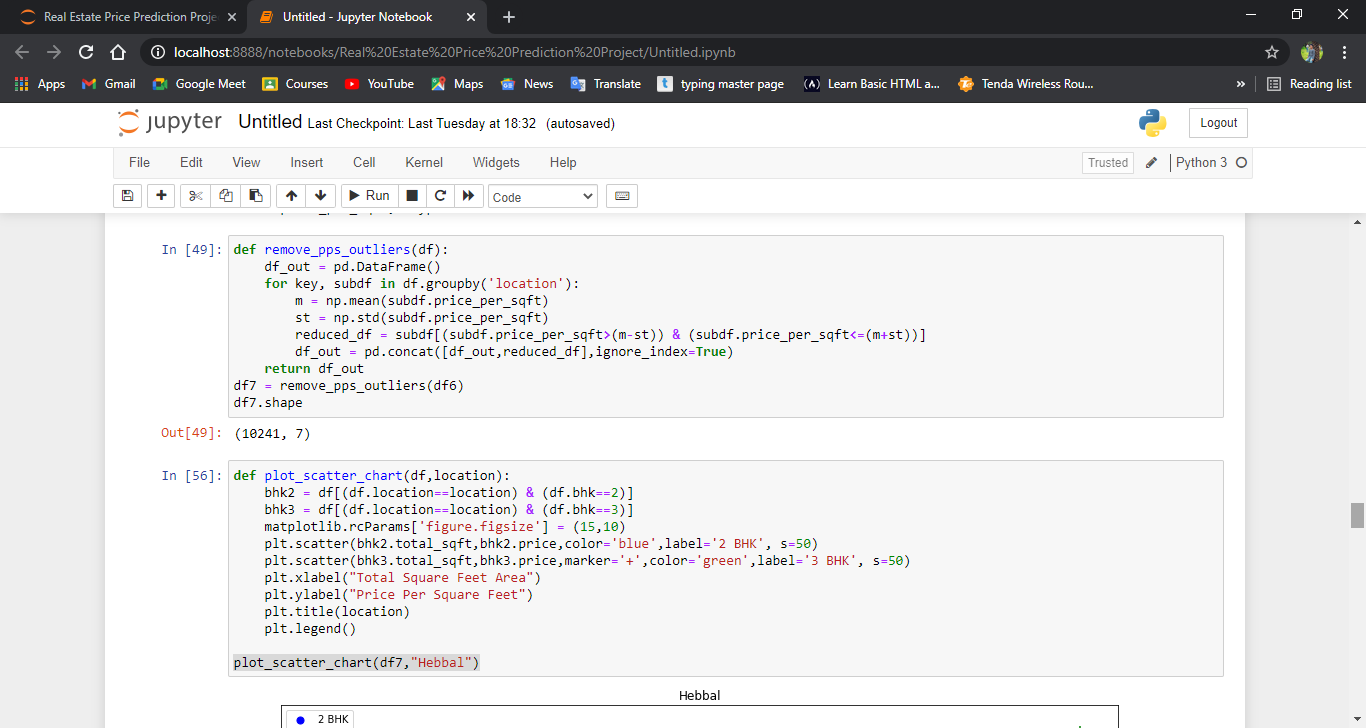
An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set.

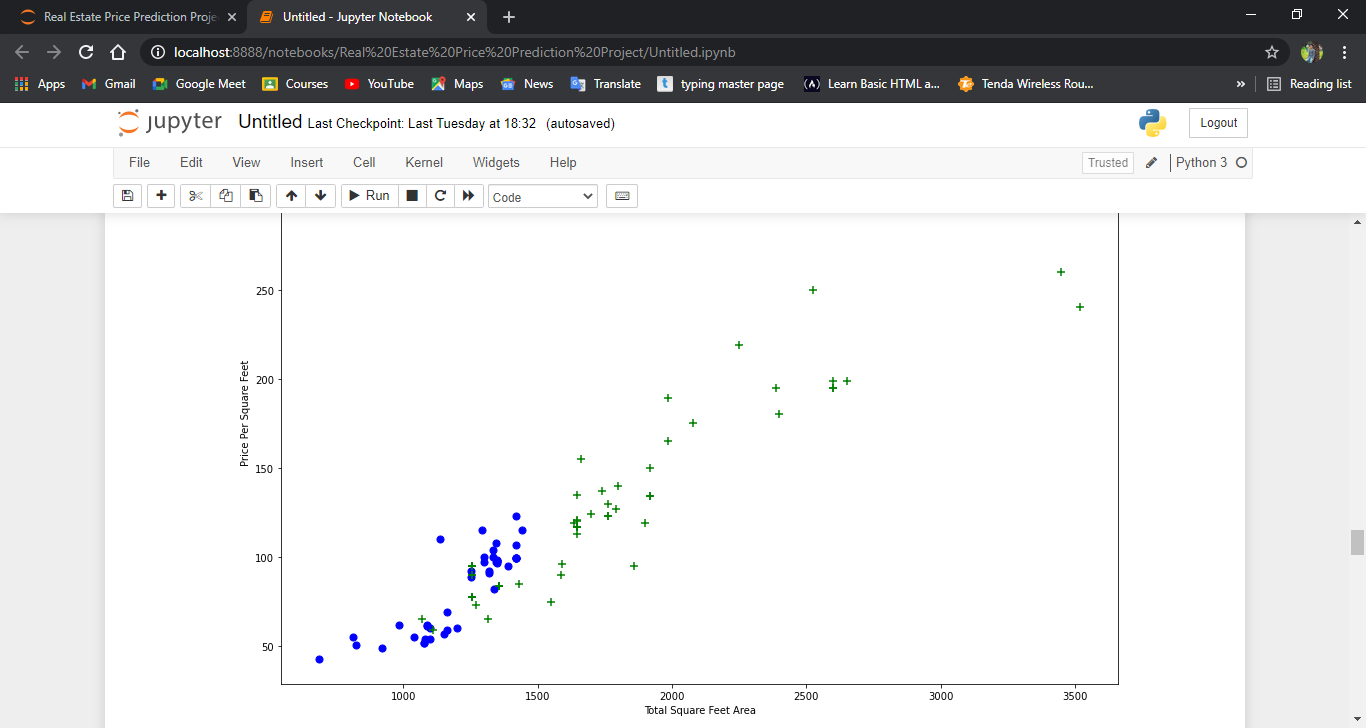
An outlier can cause serious problems in statistical analyses.

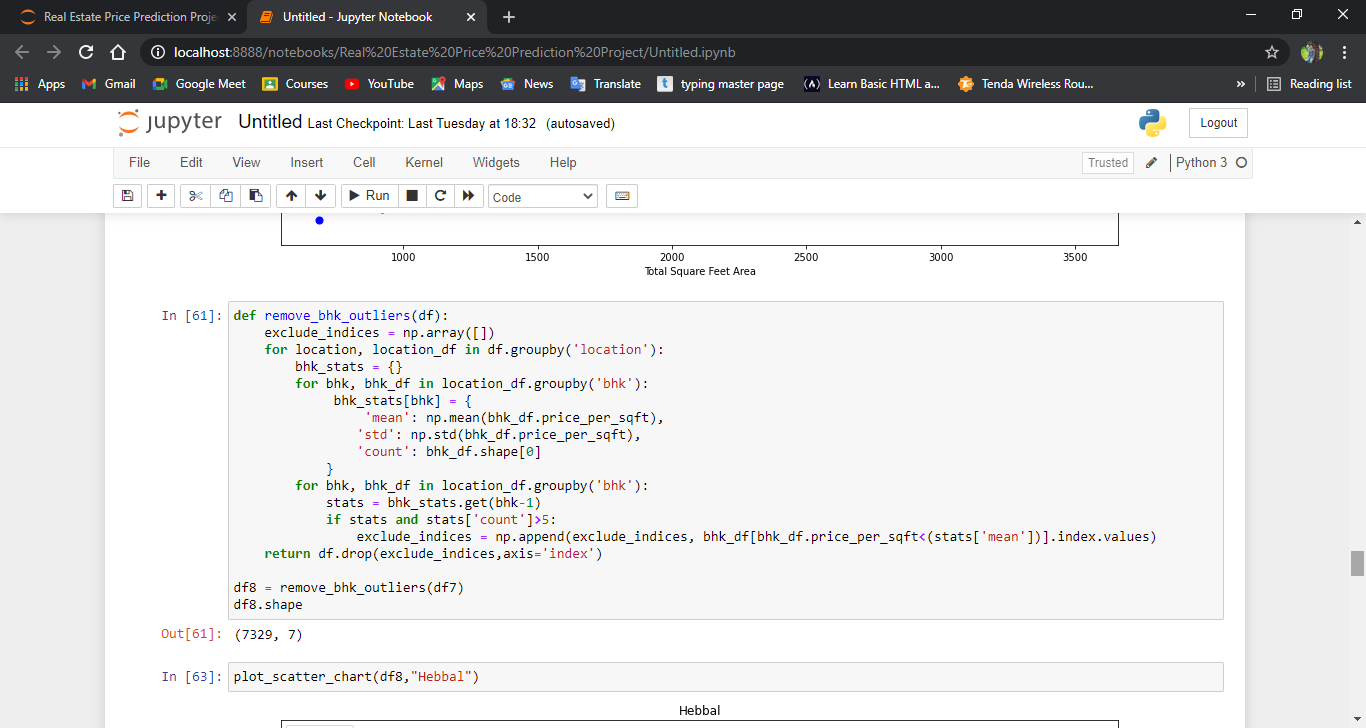
ScreenShot:-



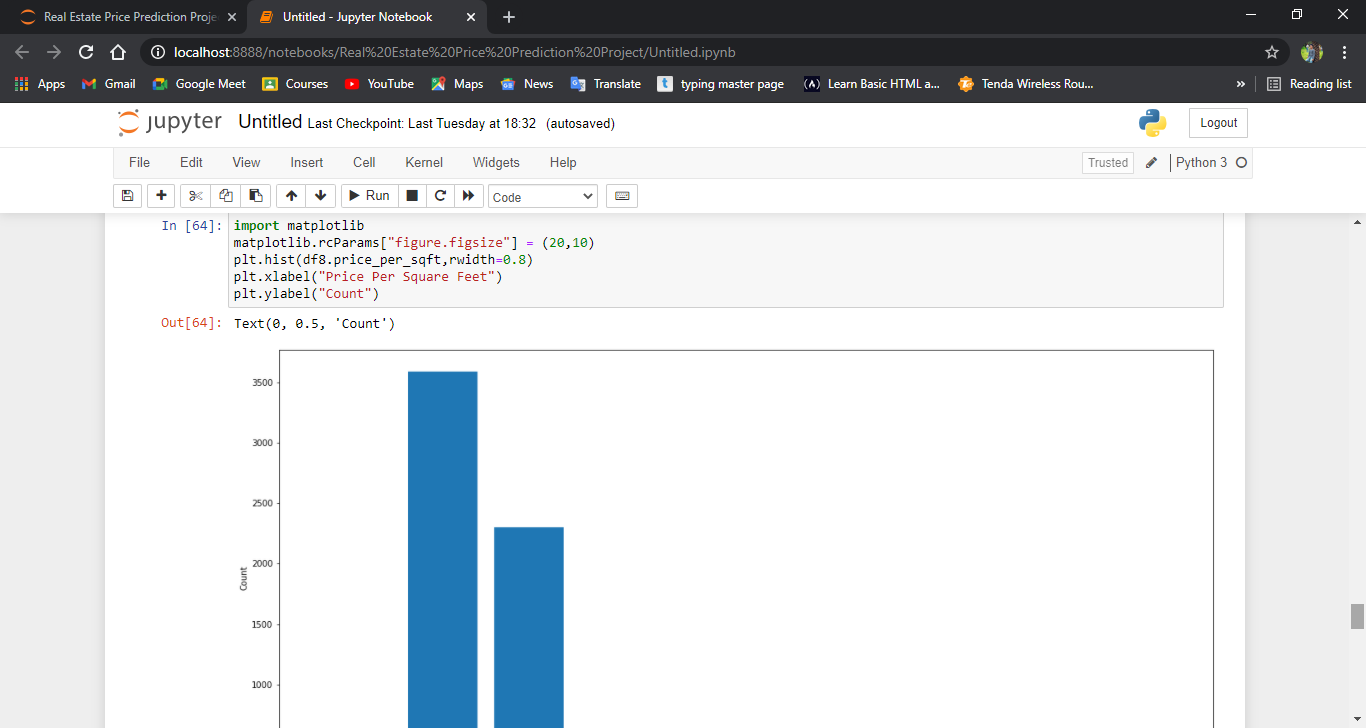


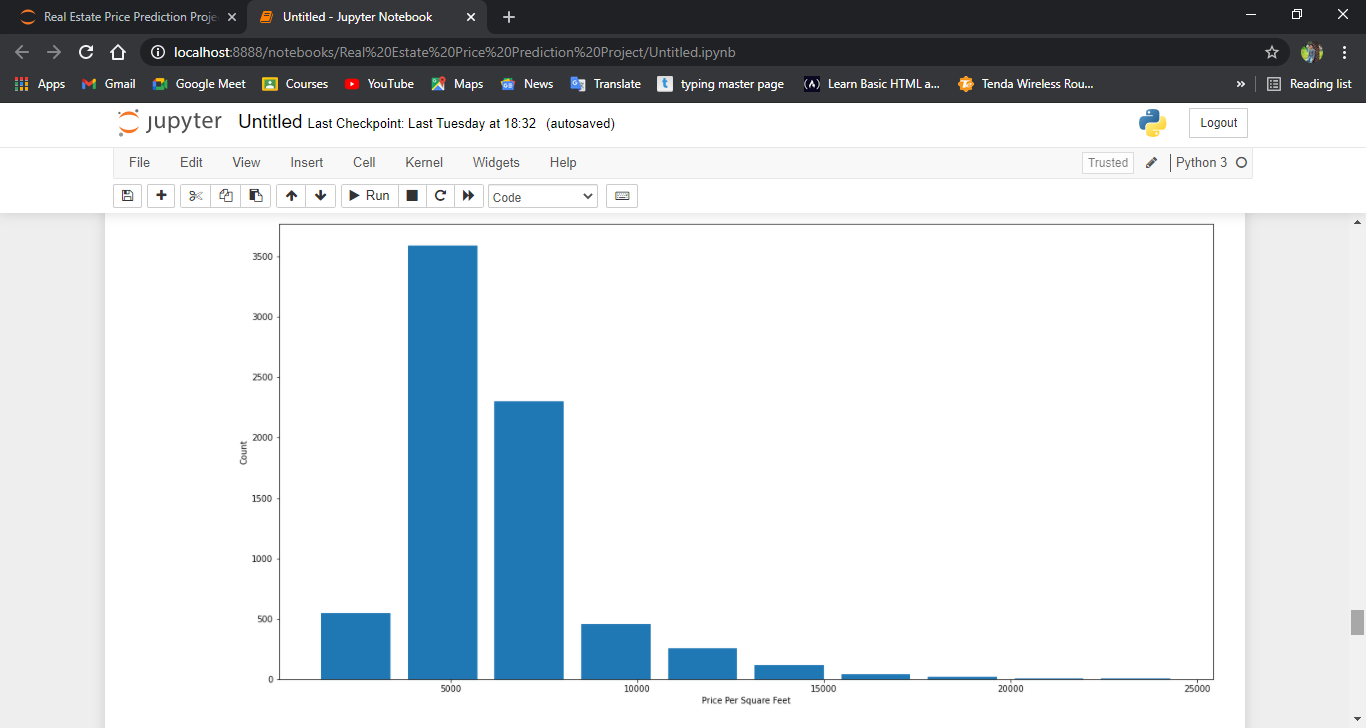


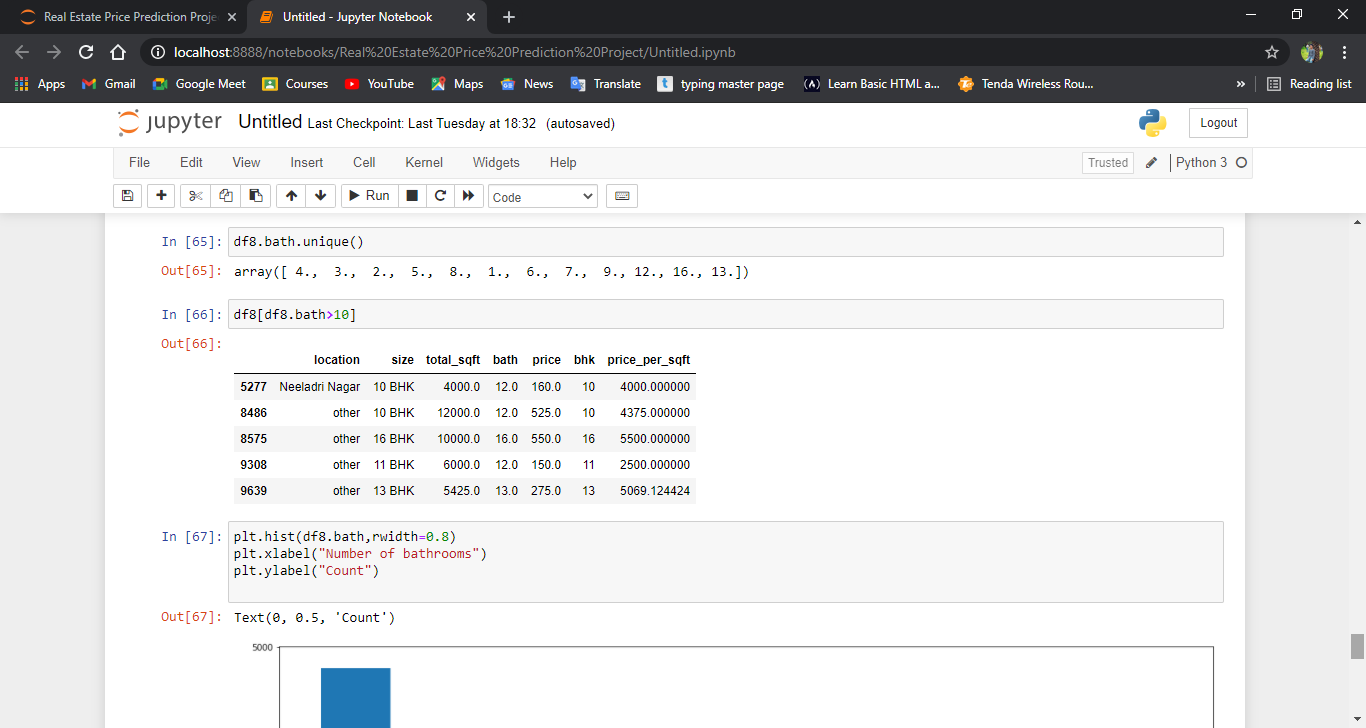


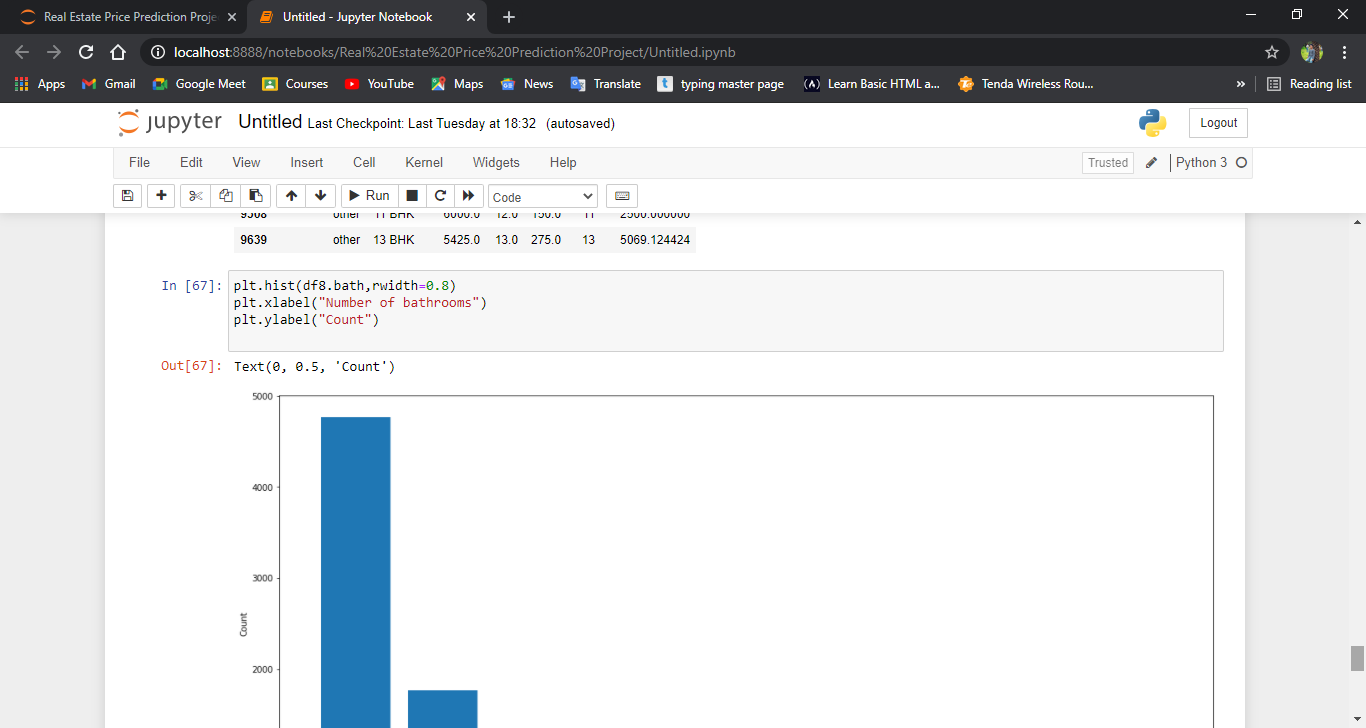




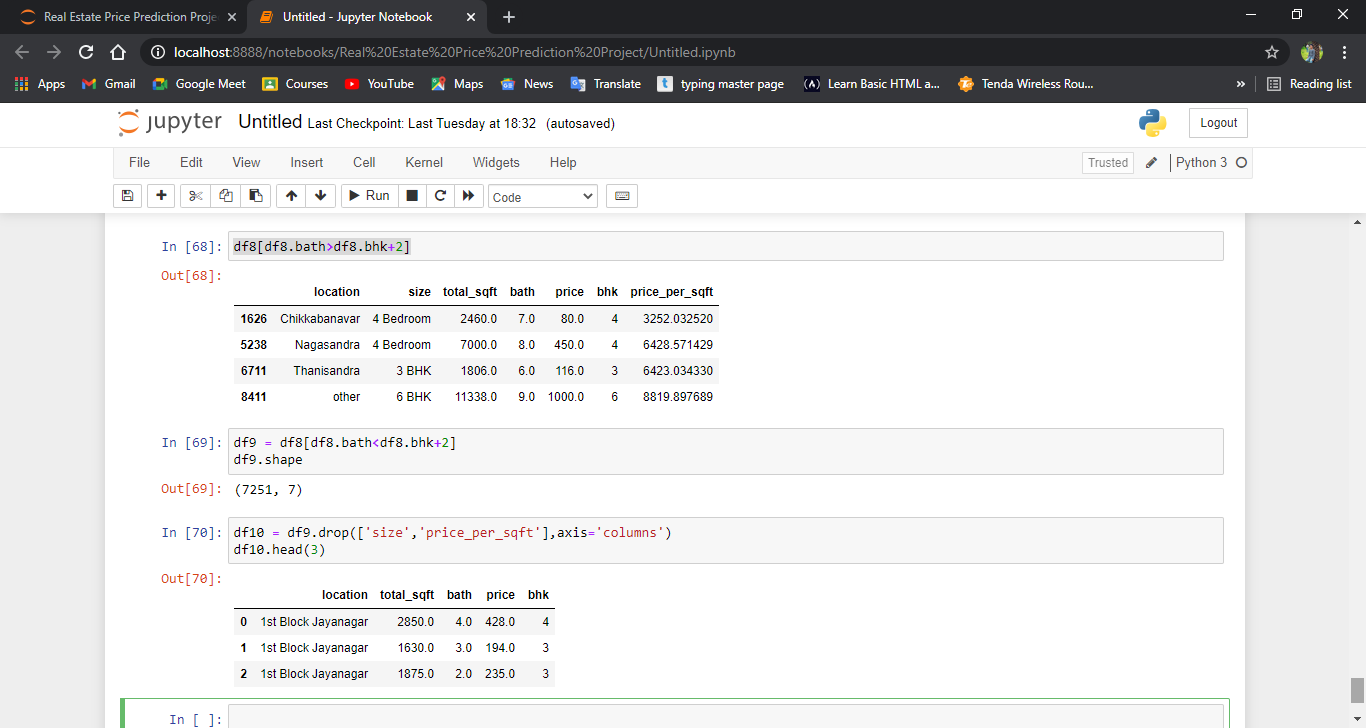








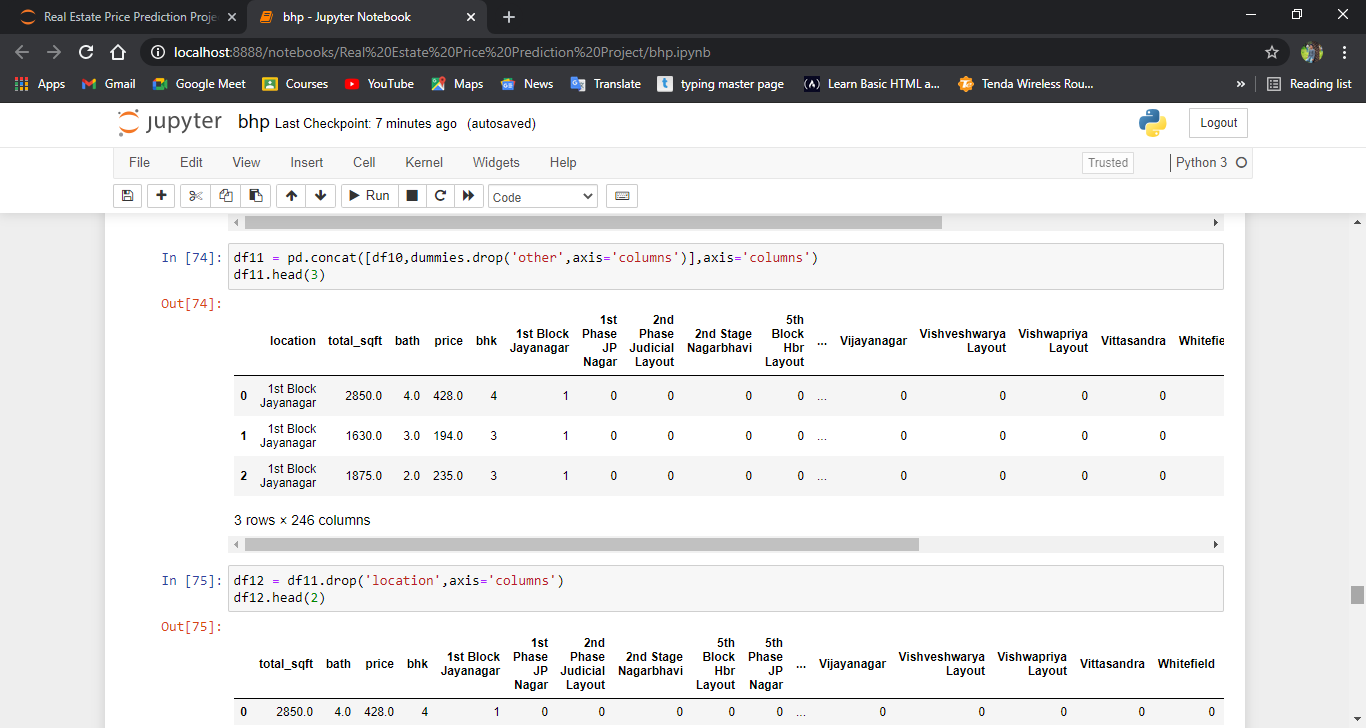


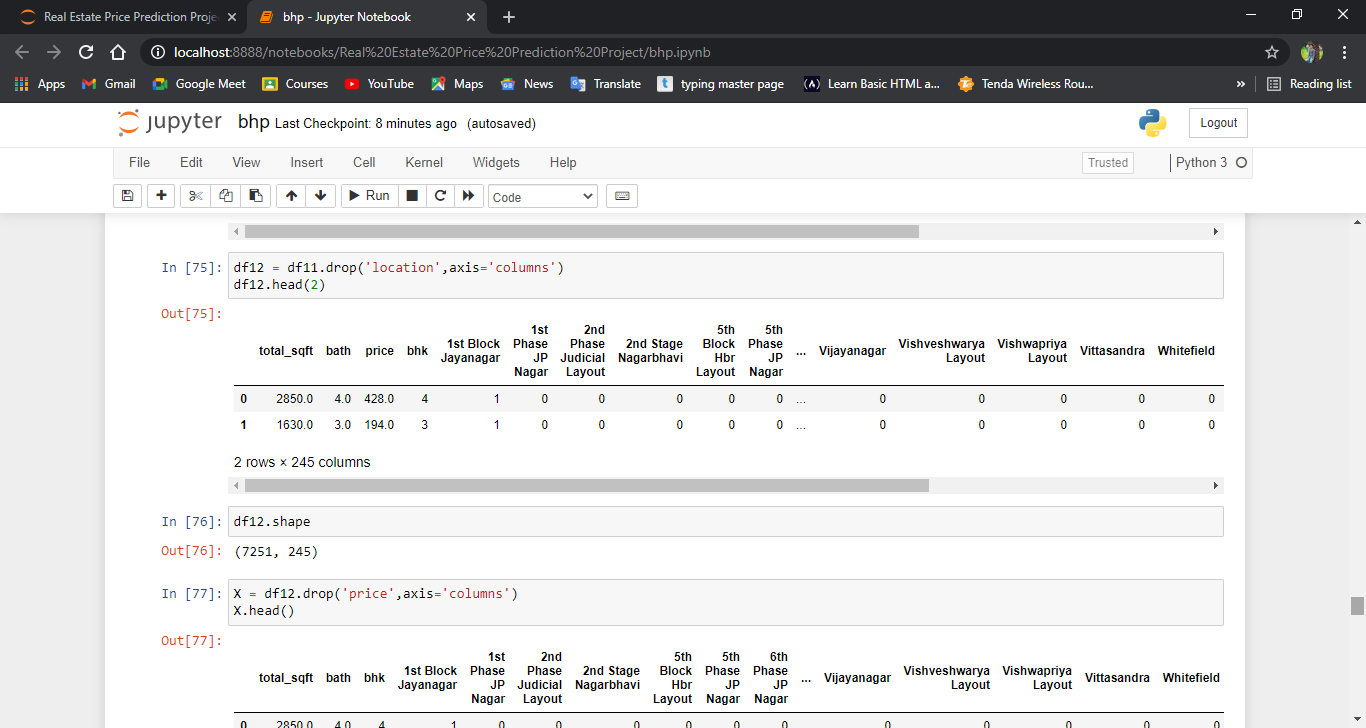


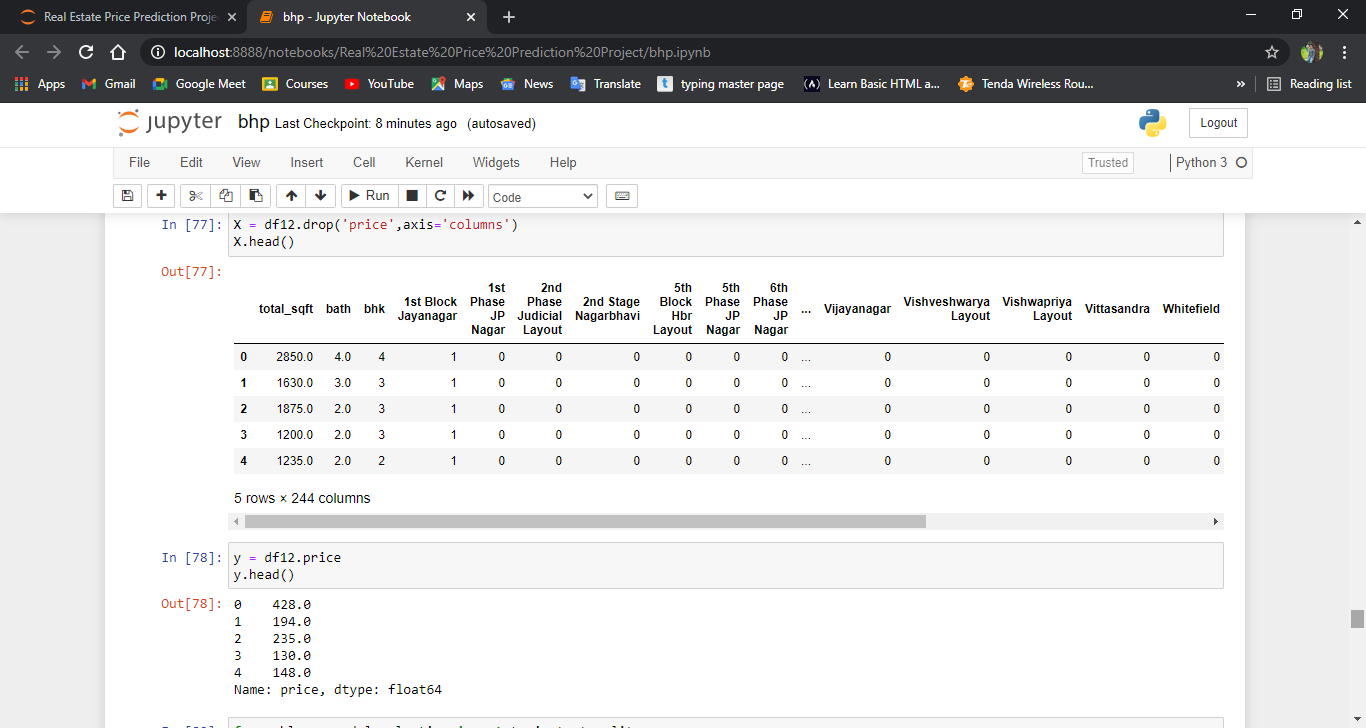
**2.4Dimensionality reduction:-**

Dimensionality reduction, or dimension reduction, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension. Working in high-dimensional spaces can be undesirable for many reasons; raw data are often sparse as a consequence of the curse of dimensionality, and analyzing the data is usually computationally intractable. Dimensionality reduction is common in fields that deal with large numbers of observations and/or large numbers of variables, such as signal processing, speech recognition, neuroinformatics, and bioinformatics.

***ScreenShot:-***



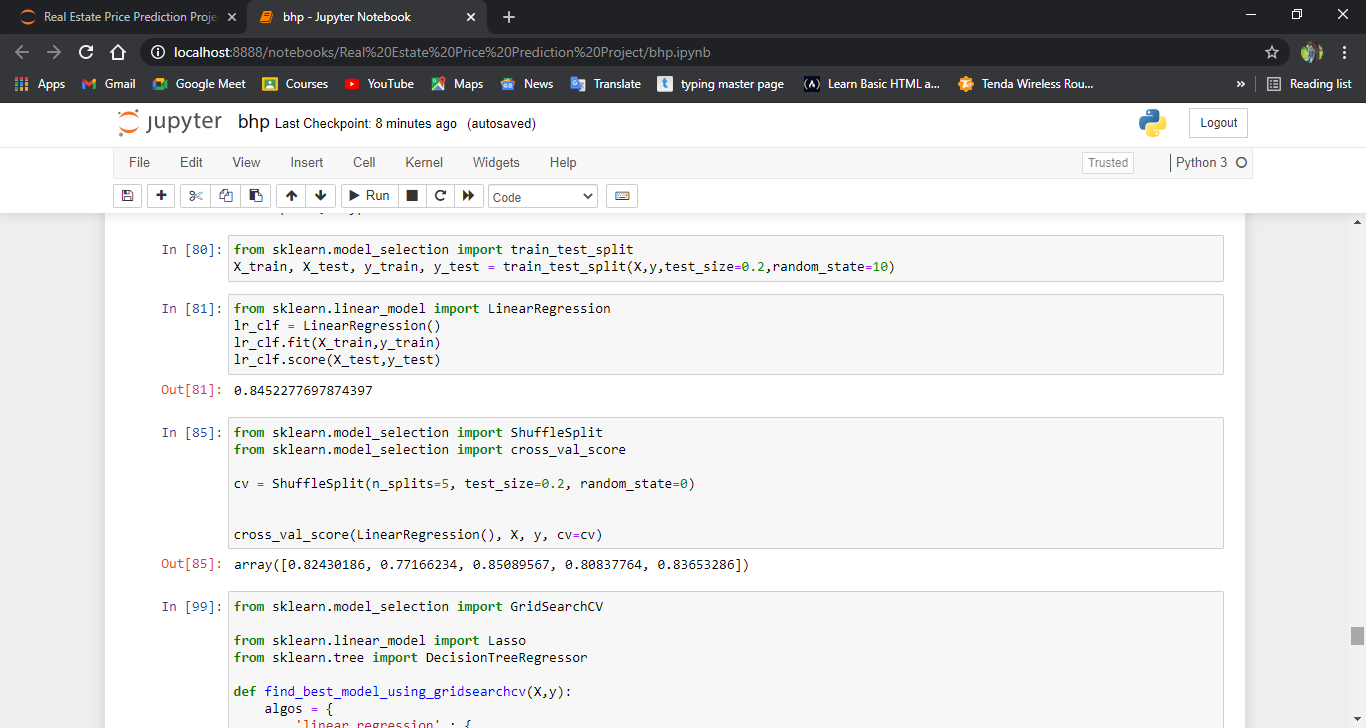


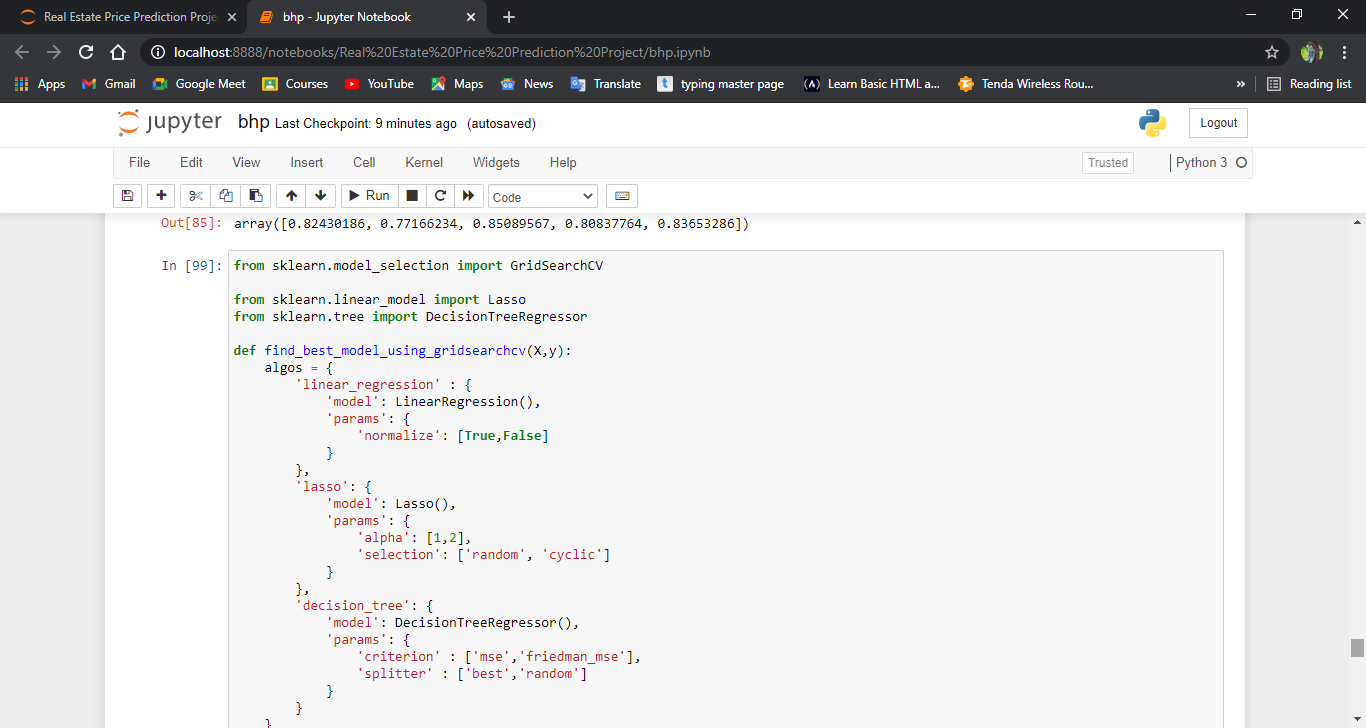


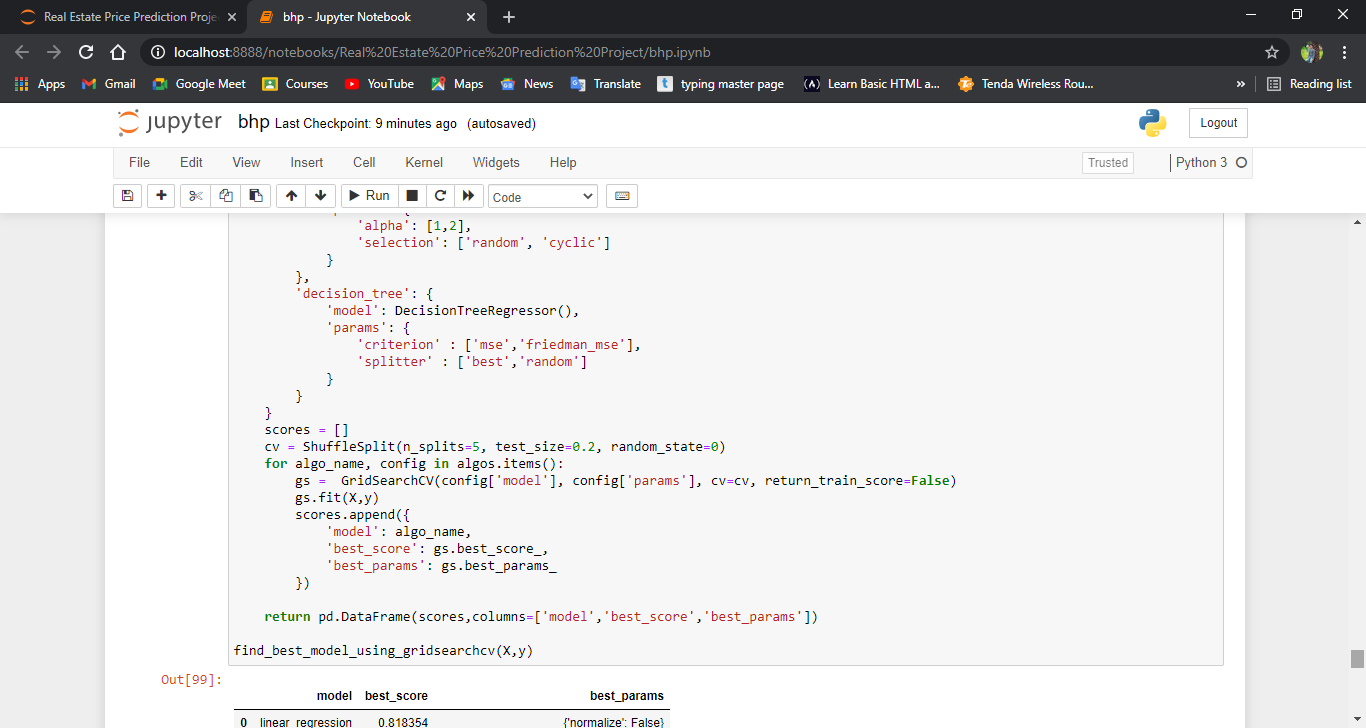
**2.5 Grid Search Cv:-**

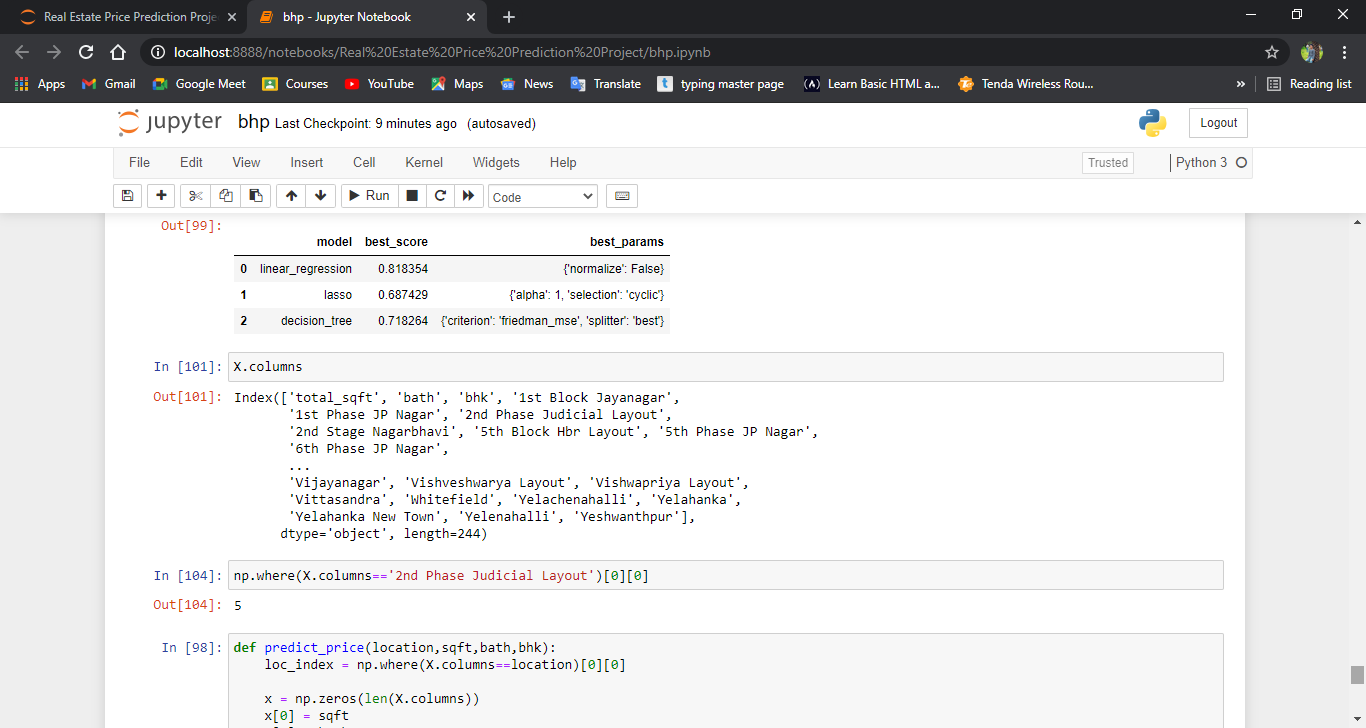
GridSearchCV is a method to search the candidate best parameters exhaustively from the grid of given parameters. Target estimator (model) and parameters for search need to be provided for this cross-validation searchmethod.

**ScreenShot:-**









**K-Fold Cross Validation:-**

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into.

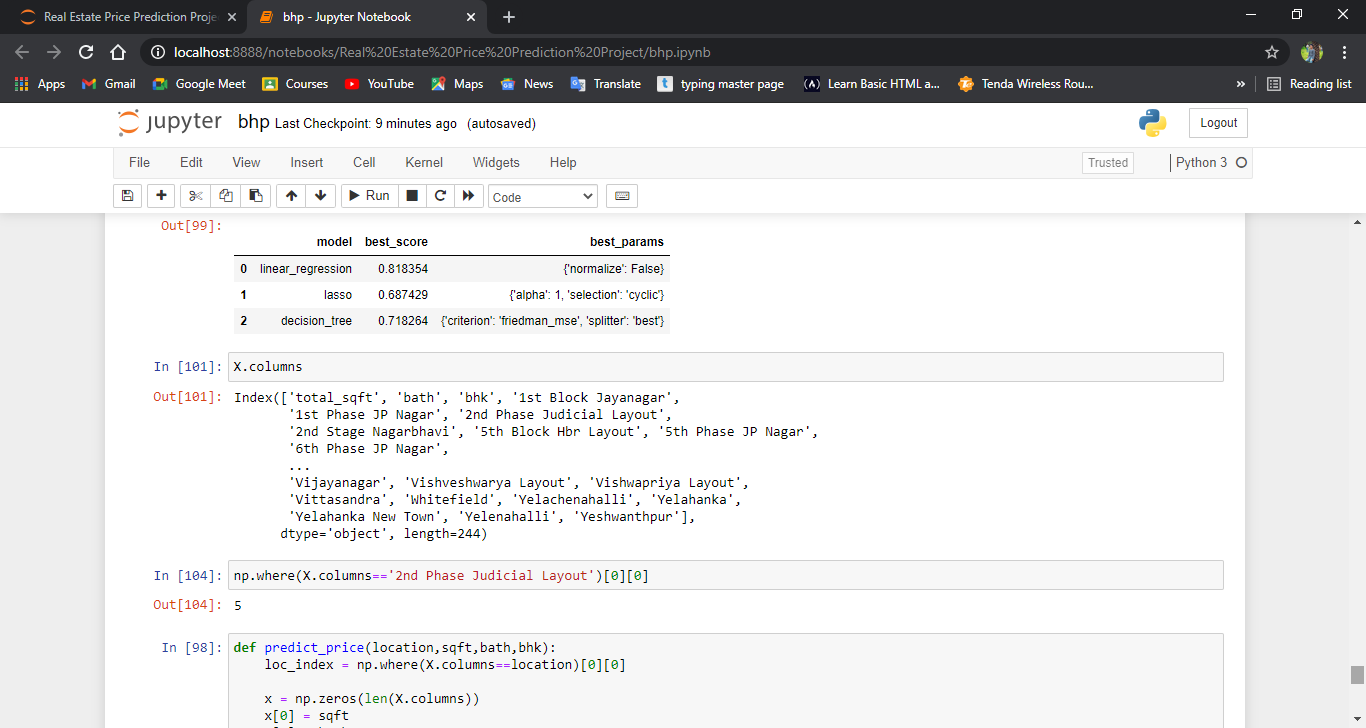
As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

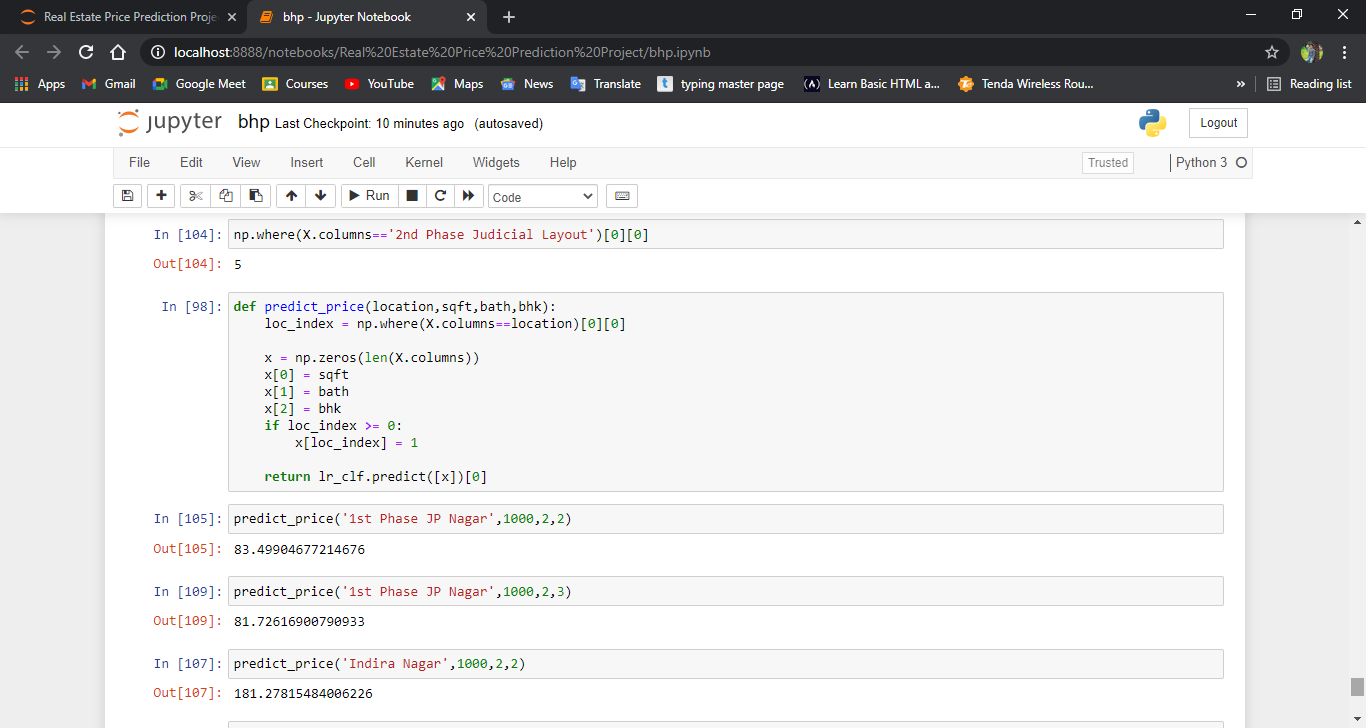
Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data.

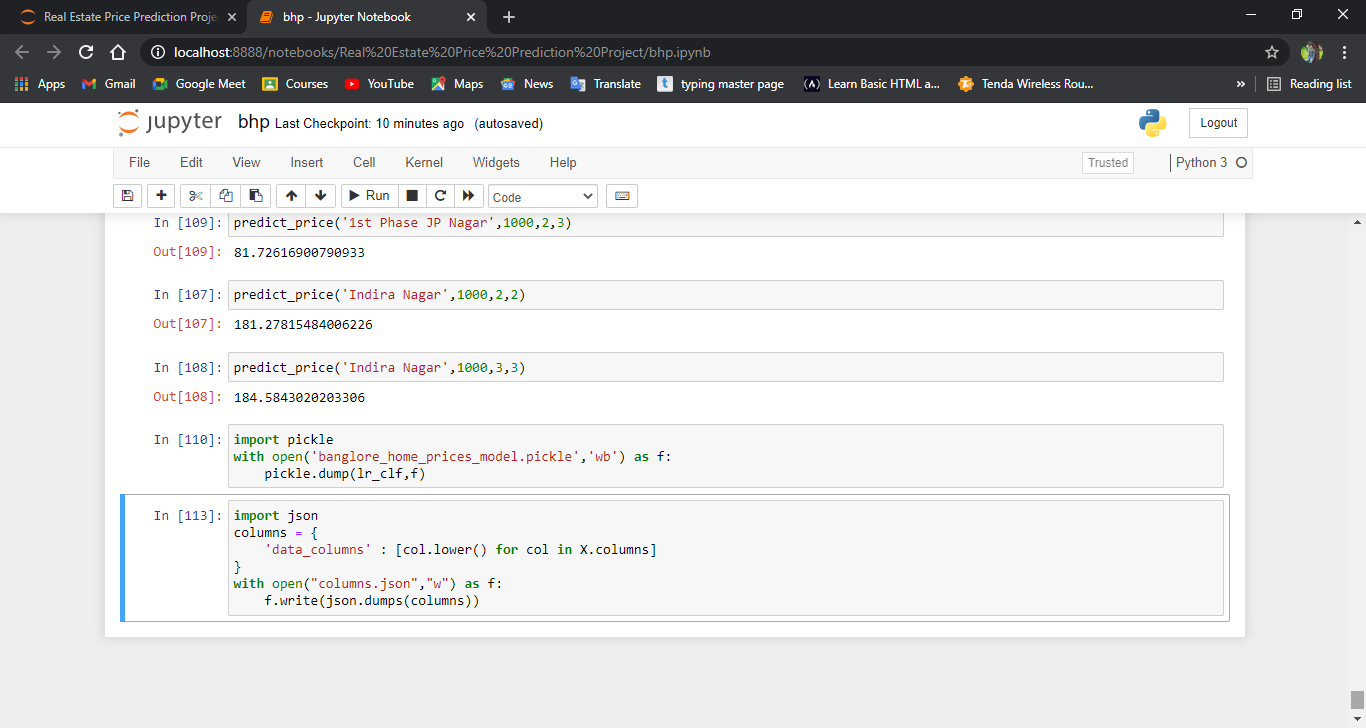
That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

**ScreenShot:-**







**3.LANGUAGE & MODELS USED:-**

**Technology and tools wise this project covers:-**

1. Python
2. Numpy and Pandas for data cleaning
3. Matplotlib for data visualization
4. Sklearn for model building
5. Jupyter notebook, visual studio code and pycharm as IDE
6. Python flask for http server
7. HTML/CSS/Javascript for UI

**Python**

Python is widely used in scientific and numeric computing:

* SciPy is a collection of packages for mathematics, science, and engineering.
* Pandas is a data analysis and modelling library.
* IPython is a powerful interactive shell that features easy editing and recording of a work session, and supports visualizations and parallel computing.
* The Software Carpentry Course teaches basic skills for scientific computing, running

bootcamps and providing open-access teaching materials.

**Libraries Used for this Project include –**

* Pandas
* NumPy
* Matplotlib
* Seaborn
* Scikit Learn

**Juypter Notebook:-**

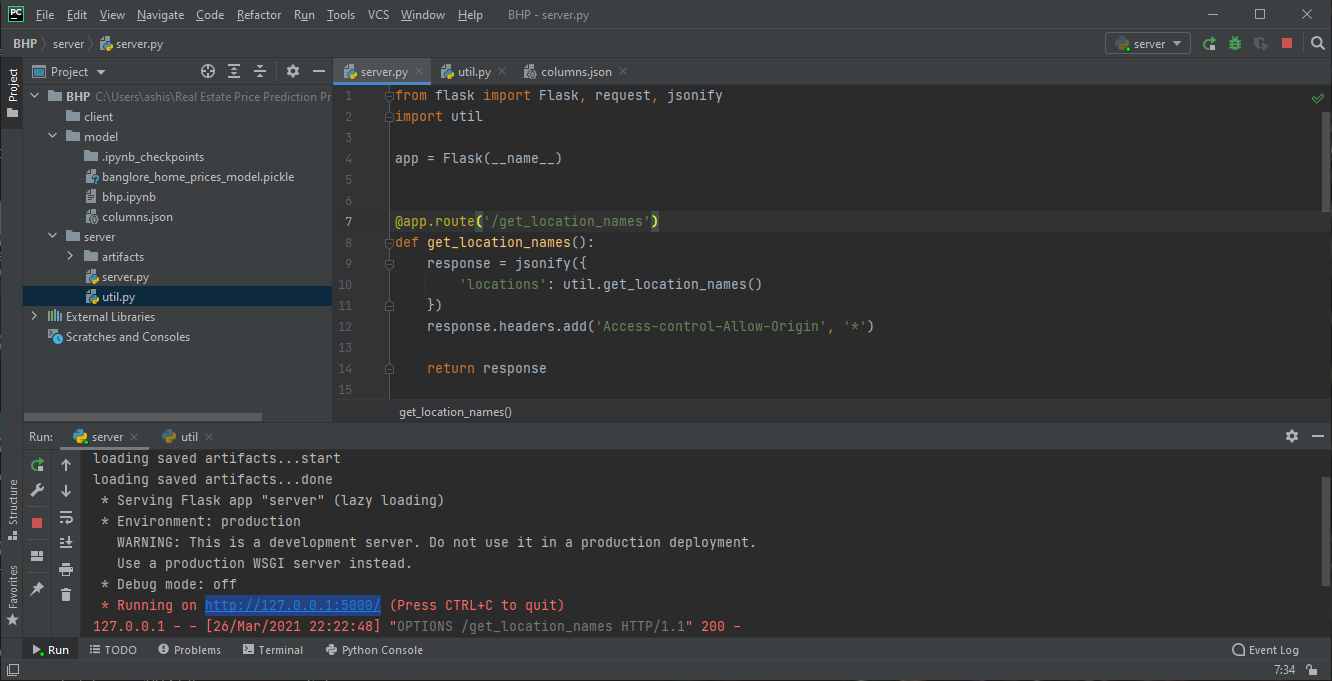
Project Jupyter is a project and community whose goal is to "develop open-source software, open-standards, and services for interactive computing across dozens of programming languages". It was spun off from IPython in 2014 by Fernando Pérez. Project Jupyter's name is a reference to the three core programming languages supported by Jupyter, which are Julia, Python and R, and also a homage to Galileo's notebooks recording the discovery of the moons of Jupiter. Project Jupyter has developed and supported the interactive computing products Jupyter Notebook, JupyterHub, and JupyterLab.

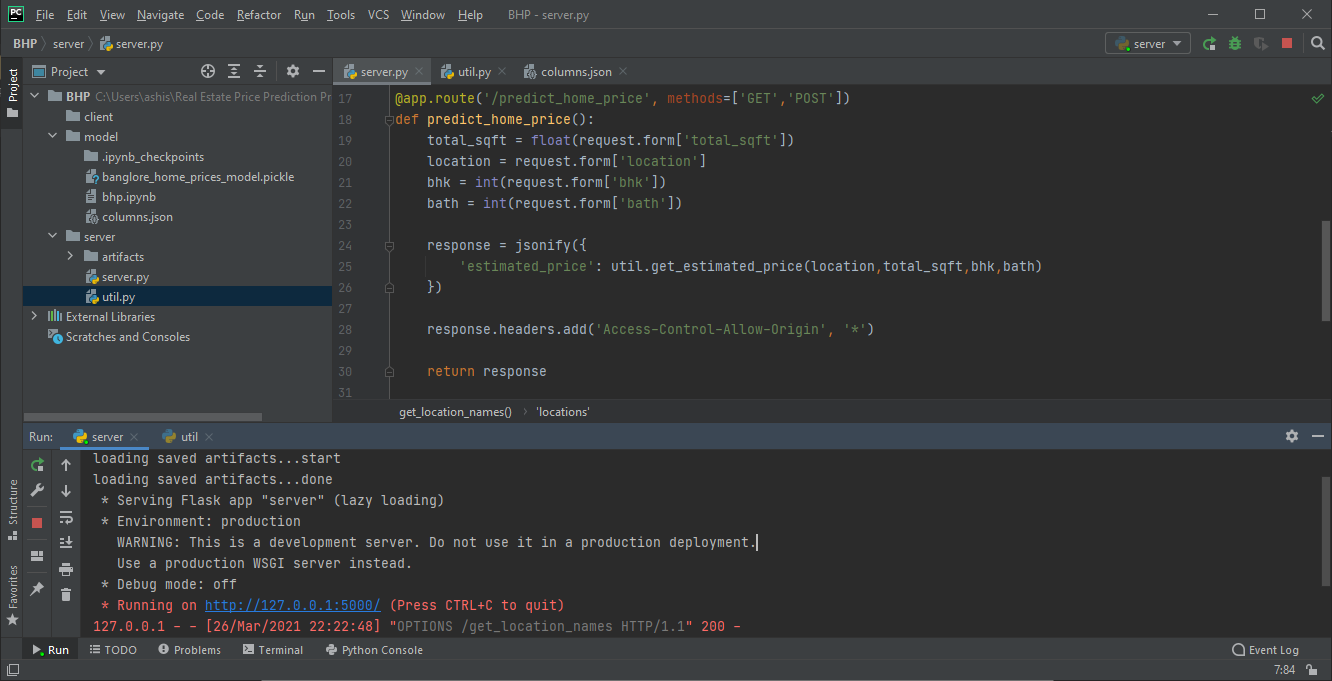
**Python Flask Server:-**

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

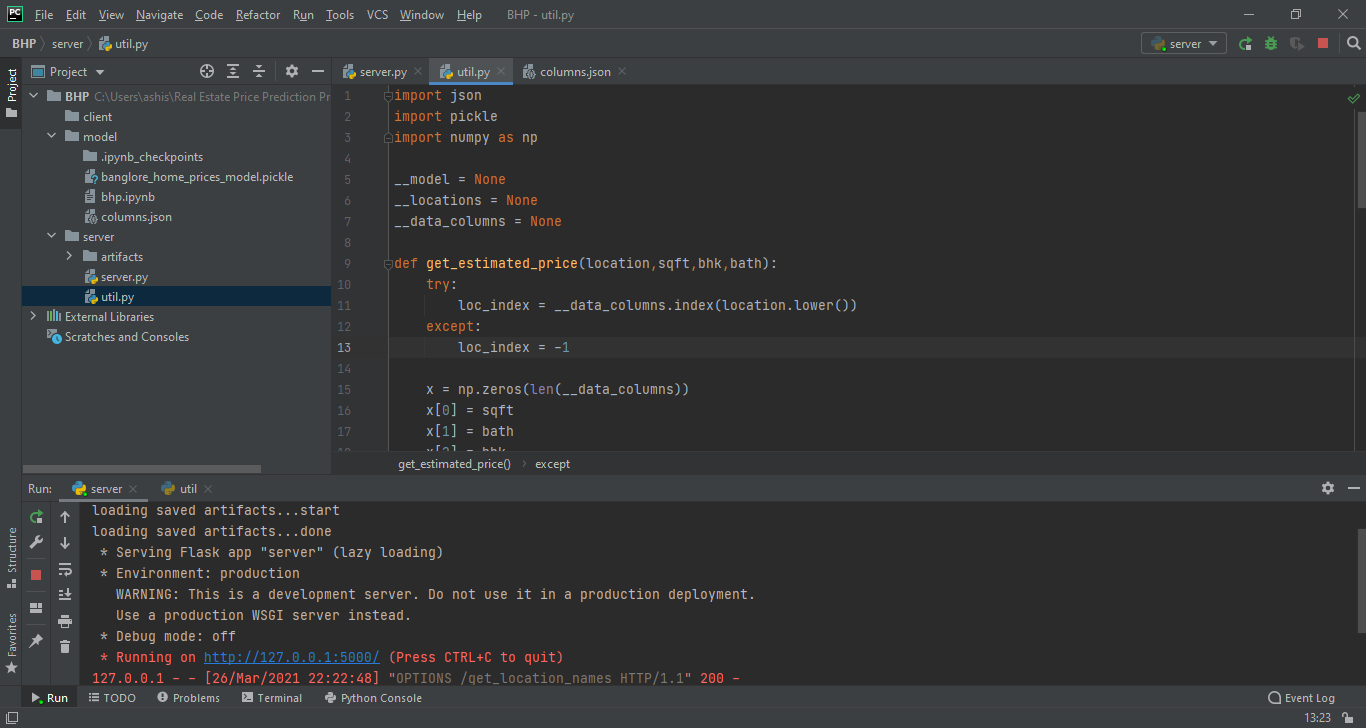
Python Flask Server:-

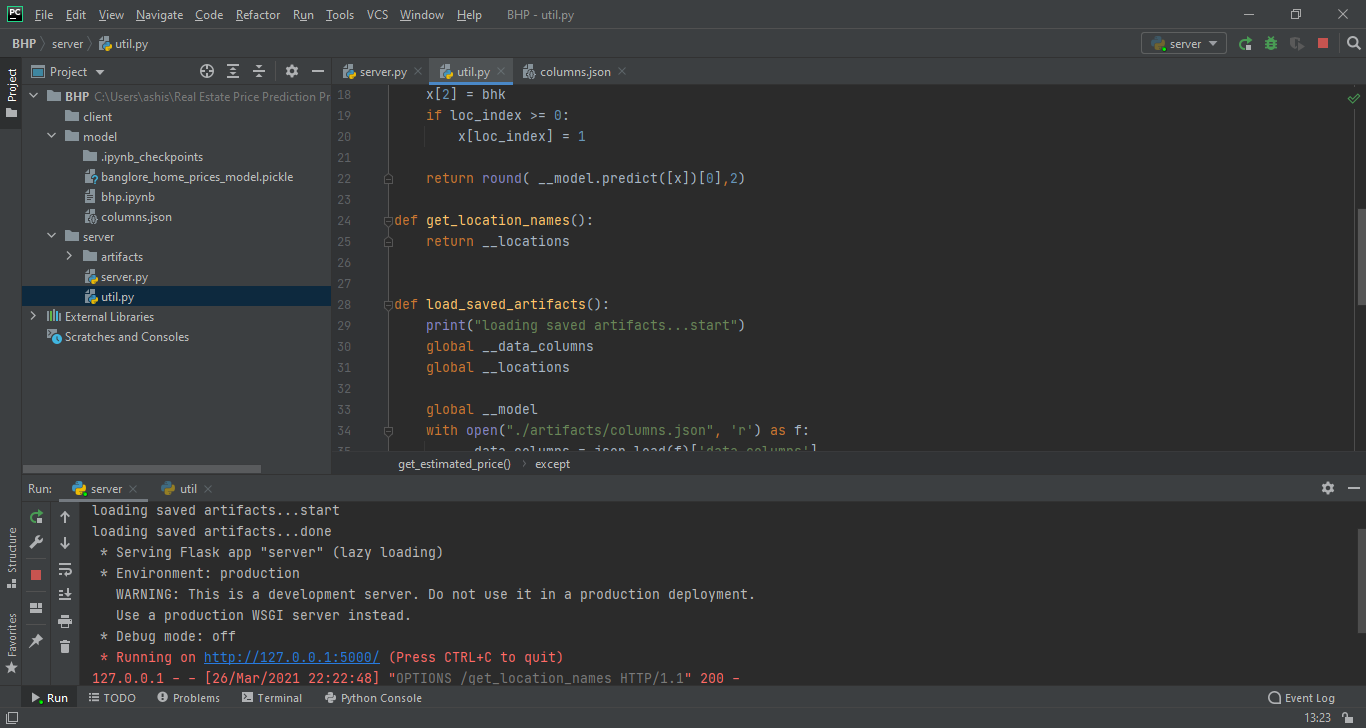
Server.py

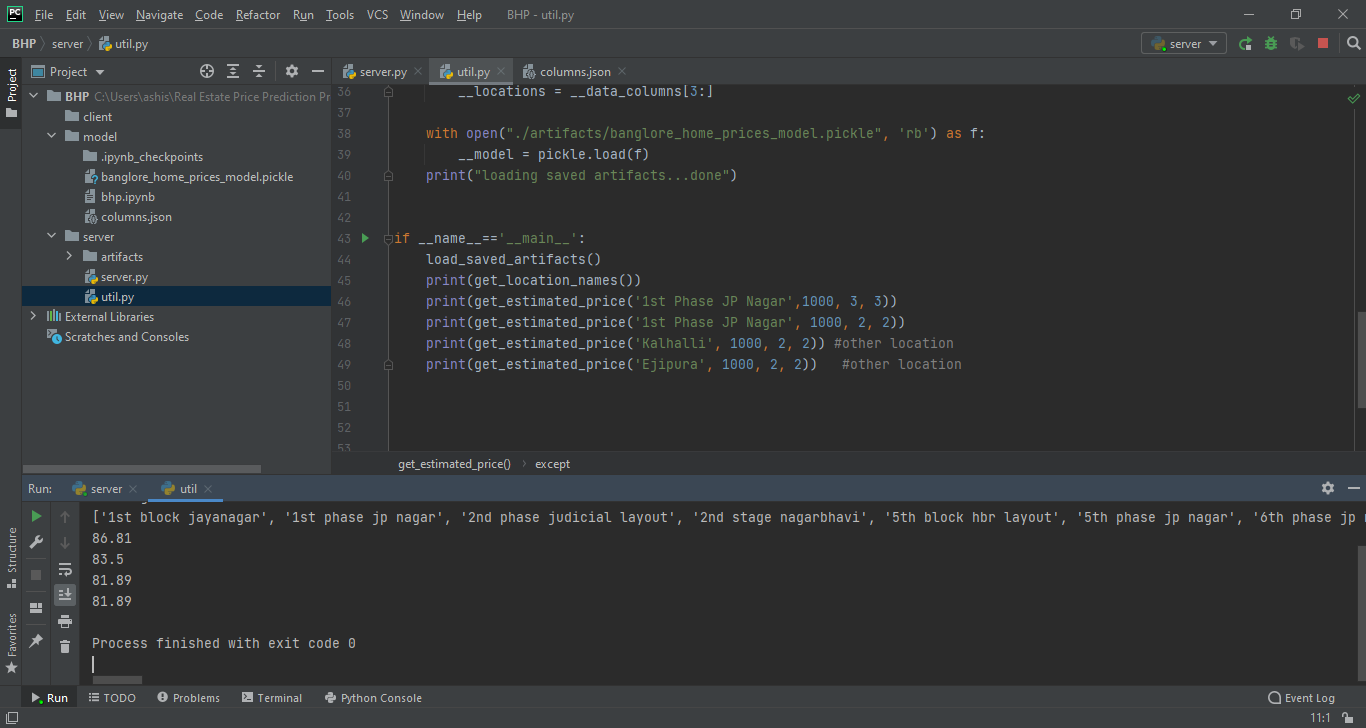




util.py







Python Flask Server Code:-

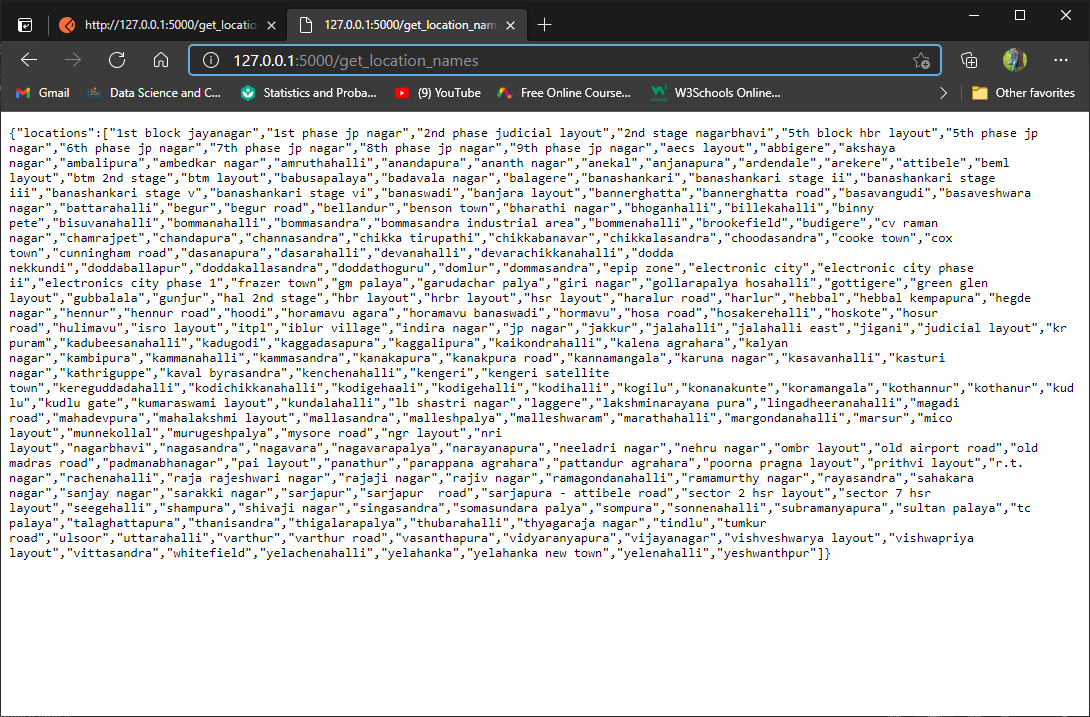
Server.py code:-

from flask import Flask, request, jsonify  
import util  
  
app = Flask(\_\_name\_\_)  
  
@app.route('/get\_location\_names', methods=['GET'])  
def get\_location\_names():  
 response = jsonify({  
 'locations': util.get\_location\_names()  
 })  
 response.headers.add('Access-Control-Allow-Origin', '\*')  
  
 return response  
  
@app.route('/predict\_home\_price', methods=['GET', 'POST'])  
def predict\_home\_price():  
 total\_sqft = float(request.form['total\_sqft'])  
 location = request.form['location']  
 bhk = int(request.form['bhk'])  
 bath = int(request.form['bath'])  
  
 response = jsonify({  
 'estimated\_price': util.get\_estimated\_price(location,total\_sqft,bhk,bath)  
 })  
 response.headers.add('Access-Control-Allow-Origin', '\*')  
  
 return response  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 print("Starting Python Flask Server For Home Price Prediction...")  
 util.load\_saved\_artifacts()  
 app.run()

util.py :-

import pickle  
import json  
import numpy as np  
  
\_\_locations = None  
\_\_data\_columns = None  
\_\_model = None  
  
def get\_estimated\_price(location,sqft,bhk,bath):  
 try:  
 loc\_index = \_\_data\_columns.index(location.lower())  
 except:  
 loc\_index = -1  
  
 x = np.zeros(len(\_\_data\_columns))  
 x[0] = sqft  
 x[1] = bath  
 x[2] = bhk  
 if loc\_index>=0:  
 x[loc\_index] = 1  
  
 return round(\_\_model.predict([x])[0],2)  
  
  
def load\_saved\_artifacts():  
 print("loading saved artifacts...start")  
 global \_\_data\_columns  
 global \_\_locations  
  
 with open("./artifacts/columns.json", "r") as f:  
 \_\_data\_columns = json.load(f)['data\_columns']  
 \_\_locations = \_\_data\_columns[3:] # first 3 columns are sqft, bath, bhk  
  
 global \_\_model  
 if \_\_model is None:  
 with open('./artifacts/banglore\_home\_prices\_model.pickle', 'rb') as f:  
 \_\_model = pickle.load(f)  
 print("loading saved artifacts...done")  
  
def get\_location\_names():  
 return \_\_locations  
  
def get\_data\_columns():  
 return \_\_data\_columns  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 load\_saved\_artifacts()  
 print(get\_location\_names())  
 print(get\_estimated\_price('1st Phase JP Nagar',1000, 3, 3))  
 print(get\_estimated\_price('1st Phase JP Nagar', 1000, 2, 2))  
 print(get\_estimated\_price('Kalhalli', 1000, 2, 2)) # other location  
 print(get\_estimated\_price('Ejipura', 1000, 2, 2)) # other location

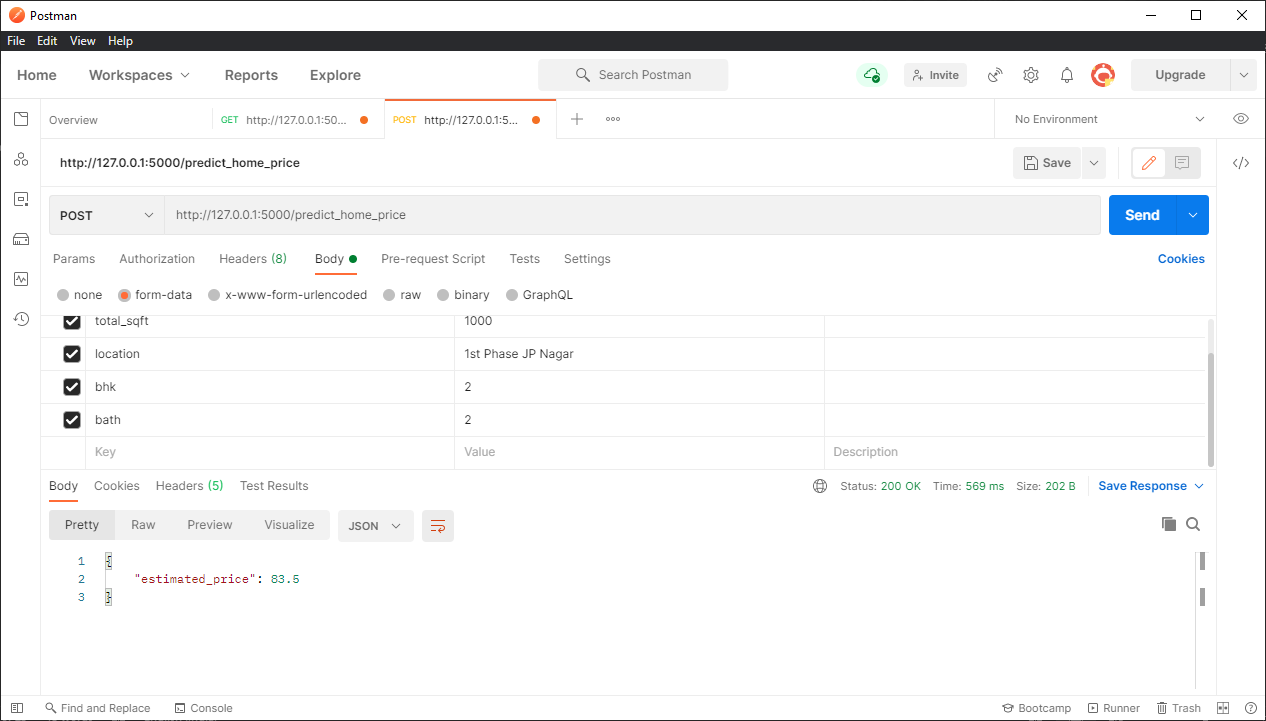
Server.py output:-



**Using PostMan Application for GET & POST:-**

Postman is a scalable API testing tool that quickly integrates into CI/CD pipeline. It started in 2012 as a side project by Abhinav Asthana to simplify API workflow in testing and development. API stands for Application Programming Interface which allows software applications to communicate with each other via API calls.





**3.7 HTML/CSS/JAVASCRIPT FOR USER INTERFACE:-**

Visual Studio Code:-

App.html :-

<!DOCTYPE html>

<html>

<head>

    <title>Banglore Home Price Prediction</title>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

    <script src="app.js"></script>

    <link rel="stylesheet" href="app.css">

</head>

<body>

<div class="img"></div>

<form class="form">

    <h2>Area (Square Feet)</h2>

    <input class="area"  type="text" id="uiSqft" class="floatLabel" name="Squareft" value="1000">

    <h2>BHK</h2>

    <div class="switch-field">

        <input type="radio" id="radio-bhk-1" name="uiBHK" value="1"/>

        <label for="radio-bhk-1">1</label>

        <input type="radio" id="radio-bhk-2" name="uiBHK" value="2" checked/>

        <label for="radio-bhk-2">2</label>

        <input type="radio" id="radio-bhk-3" name="uiBHK" value="3"/>

        <label for="radio-bhk-3">3</label>

        <input type="radio" id="radio-bhk-4" name="uiBHK" value="4"/>

        <label for="radio-bhk-4">4</label>

        <input type="radio" id="radio-bhk-5" name="uiBHK" value="5"/>

        <label for="radio-bhk-5">5</label>

    </div>

    </form>

<form class="form">

    <h2>Bath</h2>

    <div class="switch-field">

        <input type="radio" id="radio-bath-1" name="uiBathrooms" value="1"/>

        <label for="radio-bath-1">1</label>

        <input type="radio" id="radio-bath-2" name="uiBathrooms" value="2" checked/>

        <label for="radio-bath-2">2</label>

        <input type="radio" id="radio-bath-3" name="uiBathrooms" value="3"/>

        <label for="radio-bath-3">3</label>

        <input type="radio" id="radio-bath-4" name="uiBathrooms" value="4"/>

        <label for="radio-bath-4">4</label>

        <input type="radio" id="radio-bath-5" name="uiBathrooms" value="5"/>

        <label for="radio-bath-5">5</label>

    </div>

        <h2>Location</h2>

    <div>

  <select class="location" name="" id="uiLocations">

    <option value="" disabled="disabled" selected="selected">Choose a Location</option>

        <option>Electronic City</option>

        <option>Rajaji Nagar</option>

  </select>

</div>

    <button class="submit" onclick="onClickedEstimatePrice()" type="button">Estimate Price</button>

    <div id="uiEstimatedPrice" class="result">  <h2></h2> </div>

</body>

</html>

app.css :-

@import url(https://fonts.googleapis.com/css?family=Roboto:300);

.switch-field {

    display: flex;

    margin-bottom: 36px;

    overflow: hidden;

}

.switch-field input {

    position: absolute !important;

    clip: rect(0, 0, 0, 0);

    height: 1px;

    width: 1px;

    border: 0;

    overflow: hidden;

}

.switch-field label {

    background-color: #e4e4e4;

    color: rgba(0, 0, 0, 0.6);

    font-size: 14px;

    line-height: 1;

    text-align: center;

    padding: 8px 16px;

    margin-right: -1px;

    border: 1px solid rgba(0, 0, 0, 0.2);

    box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.3), 0 1px rgba(255, 255, 255, 0.1);

    transition: all 0.1s ease-in-out;

}

.switch-field label:hover {

    cursor: pointer;

}

.switch-field input:checked + label {

    background-color: #a5dc86;

    box-shadow: none;

}

.switch-field label:first-of-type {

    border-radius: 4px 0 0 4px;

}

.switch-field label:last-of-type {

    border-radius: 0 4px 4px 0;

}

.form {

    max-width: 270px;

    font-family: "Lucida Grande", Tahoma, Verdana, sans-serif;

    font-weight: normal;

    line-height: 1.625;

    margin: 8px auto;

    padding-left: 16px;

    z-index: 2;

}

h2 {

    font-size: 18px;

    margin-bottom: 8px;

}

.area{

  font-family: "Roboto", sans-serif;

  outline: 0;

  background: #f2f2f2;

  width: 76%;

  border: 0;

  margin: 0 0 10px;

  padding: 10px;

  box-sizing: border-box;

  font-size: 15px;

  height: 35px;

  border-radius: 5px;

}

.location{

  font-family: "Roboto", sans-serif;

  outline: 0;

  background: #f2f2f2;

  width: 76%;

  border: 0;

  margin: 0 0 10px;

  padding: 10px;

  box-sizing: border-box;

  font-size: 15px;

  height: 40px;

  border-radius: 5px;

}

.submit{

  background: #a5dc86;

  width: 76%;

  border: 0;

  margin: 25px 0 10px;

  box-sizing: border-box;

  font-size: 15px;

    height: 35px;

    text-align: center;

    border-radius: 5px;

}

.result{

        background: #dcd686;

        width: 76%;

        border: 0;

        margin: 25px 0 10px;

        box-sizing: border-box;

        font-size: 15px;

        height: 35px;

        text-align: center;

}

.img {

  background: url('https://images.unsplash.com/photo-1564013799919-ab600027ffc6?ixlib=rb-1.2.1&auto=format&fit=crop&w=1350&q=80');

    background-repeat: no-repeat;

  background-size: auto;

  background-size:100% 100%;

  -webkit-filter: blur(5px);

  -moz-filter: blur(5px);

  -o-filter: blur(5px);

  -ms-filter: blur(5px);

  filter: blur(15px);

  position: fixed;

  width: 100%;

  height: 100%;

  top: 0;

  left: 0;

  z-index: -1;

}

body, html {

  height: 100%;

}

App.js :-

function getBathValue() {

  var uiBathrooms = document.getElementsByName("uiBathrooms");

  for(var i in uiBathrooms) {

    if(uiBathrooms[i].checked) {

        return parseInt(i)+1;

    }

  }

  return -1; // Invalid Value

}

function getBHKValue() {

  var uiBHK = document.getElementsByName("uiBHK");

  for(var i in uiBHK) {

    if(uiBHK[i].checked) {

        return parseInt(i)+1;

    }

  }

  return -1; // Invalid Value

}

function onClickedEstimatePrice() {

  console.log("Estimate price button clicked");

  var sqft = document.getElementById("uiSqft");

  var bhk = getBHKValue();

  var bathrooms = getBathValue();

  var location = document.getElementById("uiLocations");

  var estPrice = document.getElementById("uiEstimatedPrice");

  var url = "http://127.0.0.1:5000/predict\_home\_price"; //Use this if you are NOT using nginx which is first 7 tutorials

   // Use this if  you are using nginx. i.e tutorial 8 and onwards

  $.post(url, {

      total\_sqft: parseFloat(sqft.value),

      bhk: bhk,

      bath: bathrooms,

      location: location.value

  },function(data, status) {

      console.log(data.estimated\_price);

      estPrice.innerHTML = "<h2>" + data.estimated\_price.toString() + " Lakh</h2>";

      console.log(status);

  });

}

function onPageLoad() {

  console.log( "document loaded" );

  var url = "http://127.0.0.1:5000/get\_location\_names"; // Use this if you are NOT using nginx which is first 7 tutorials

  //var url = "/api/get\_location\_names"; // Use this if  you are using nginx. i.e tutorial 8 and onwards

  $.get(url,function(data, status) {

      console.log("got response for get\_location\_names request");

      if(data) {

          var locations = data.locations;

          var uiLocations = document.getElementById("uiLocations");

          $('#uiLocations').empty();

          for(var i in locations) {

              var opt = new Option(locations[i]);

              $('#uiLocations').append(opt);

          }

      }

  });

}

window.onload = onPageLoad;

**MODELS USED**

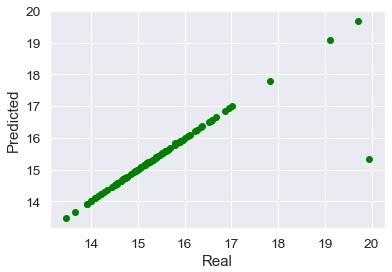
**R egression Model:-**

Linear Regression is a machine learning algorithm based on supervised learning.

It performs a regression task. Regression models a target prediction value based on independent variables.

It is mostly used for finding out the relationship between variables and forecasting.

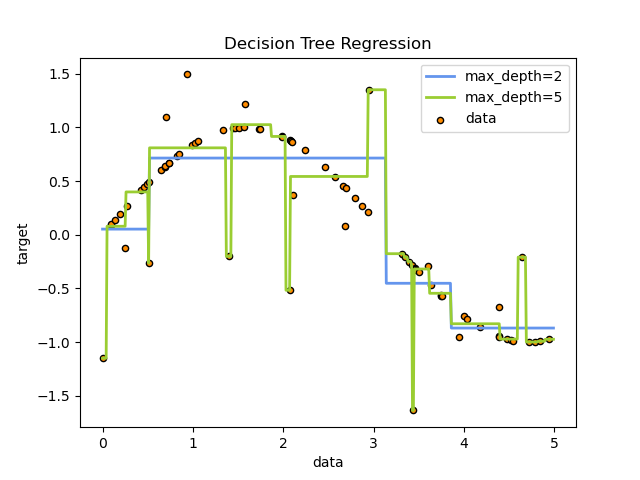
## Real Vs Predicted



**Decision Tree Regressor:-**

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

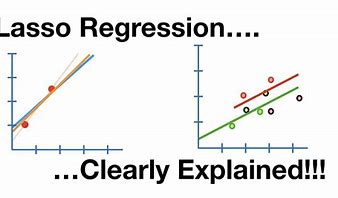
It is one way to display an algorithm that only contains conditional control statements.



**Lasso Regression:-**

In statistics and machine learning, lasso is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model.

It was originally introduced in geophysics, and later by Robert Tibshirani, who coined the term.



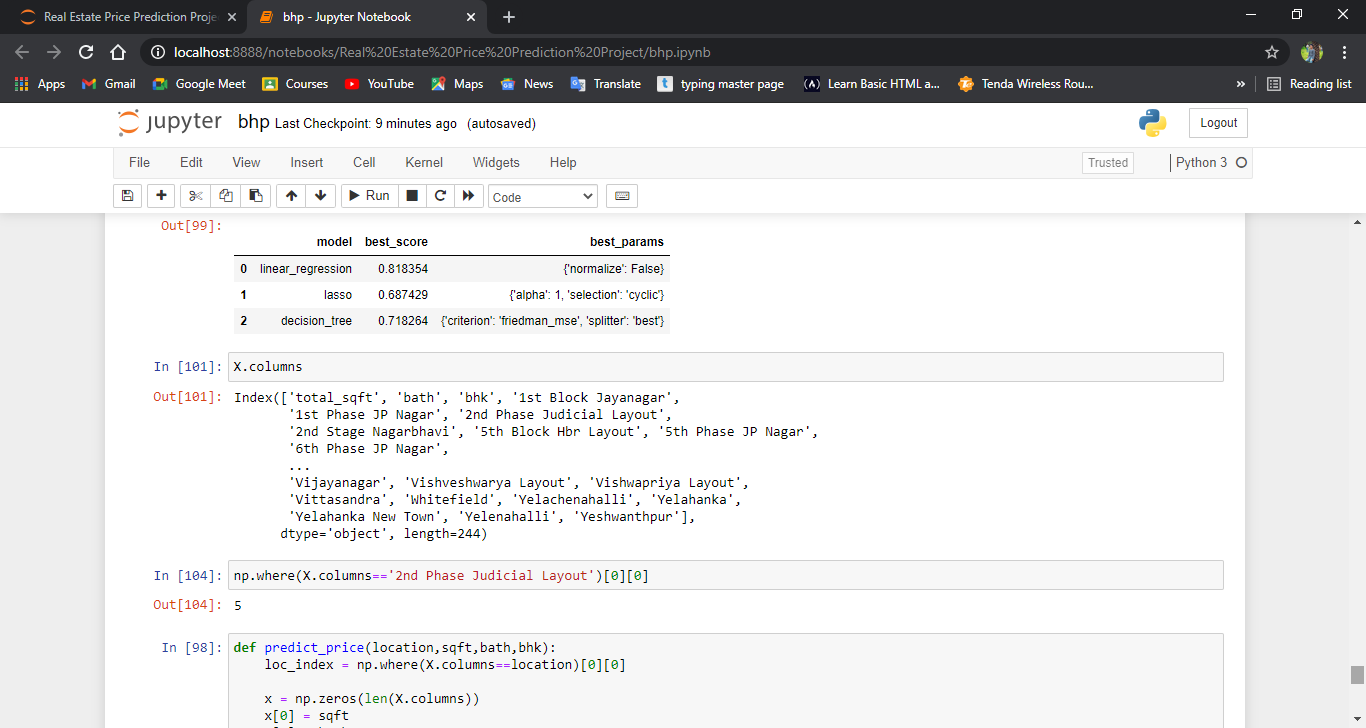
**RESULTS AND DISCUSSIONS**

**Best Suited Model**

So, our study showed that……..

Linear Regression displayed the best performance for this Dataset and can be used for deploying purposes.

Decision Tree Regressor and Lasso Regressor are far behind, so can’t be recommended for further deployment purposes.



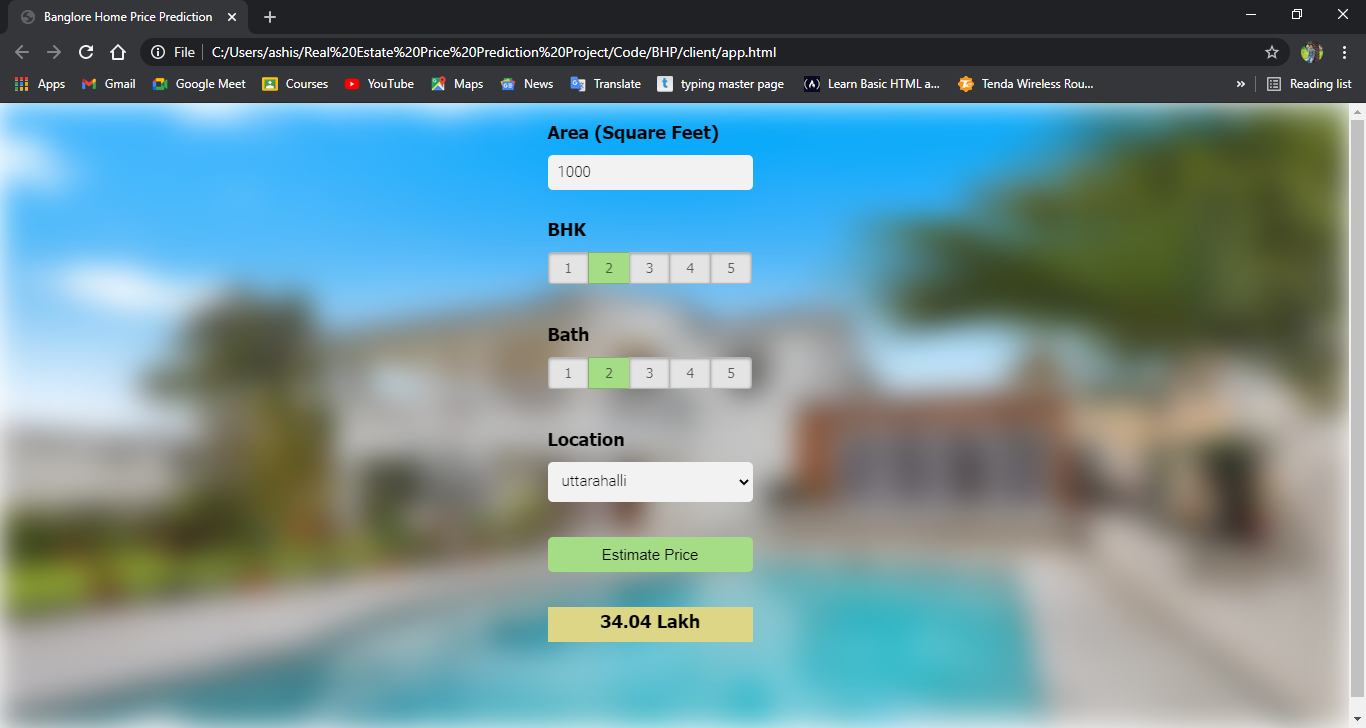
**Deployment App**

The Model is deployed through Python Web App Flask in

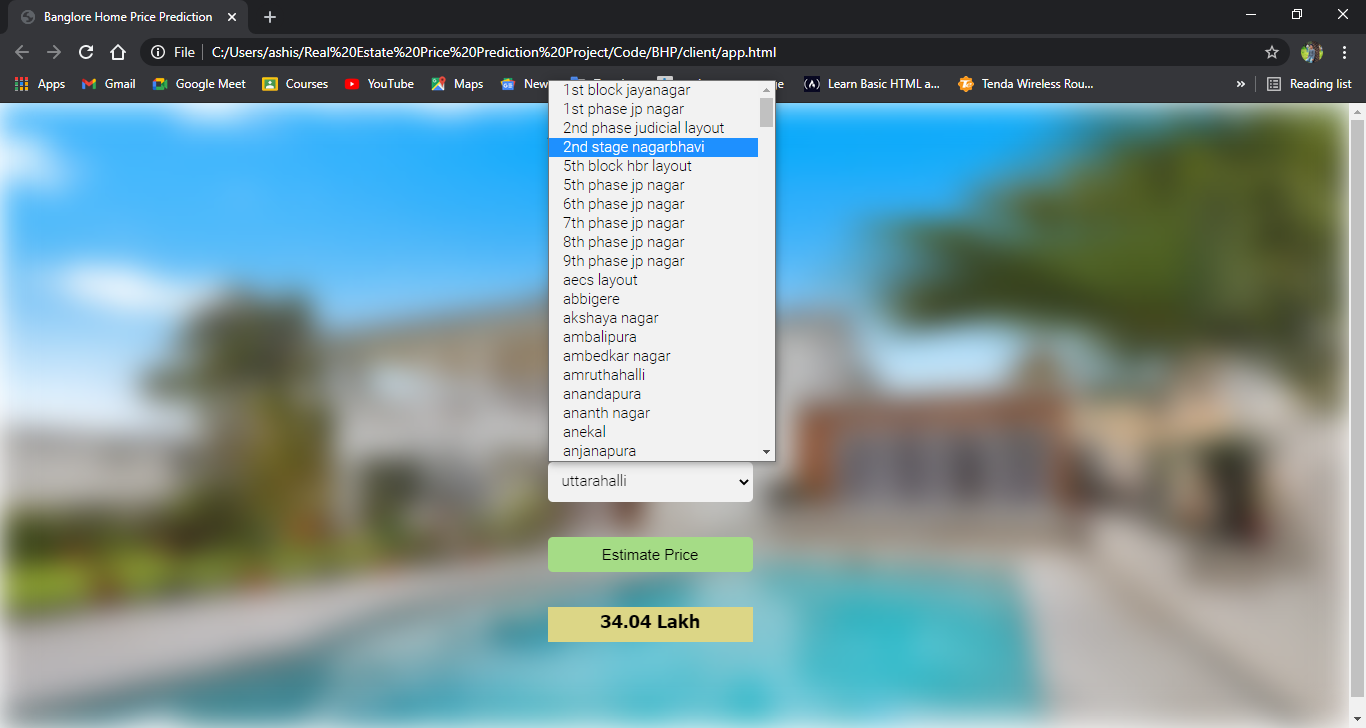
collaboration with HTML and CSS.

Output screenshoot:-

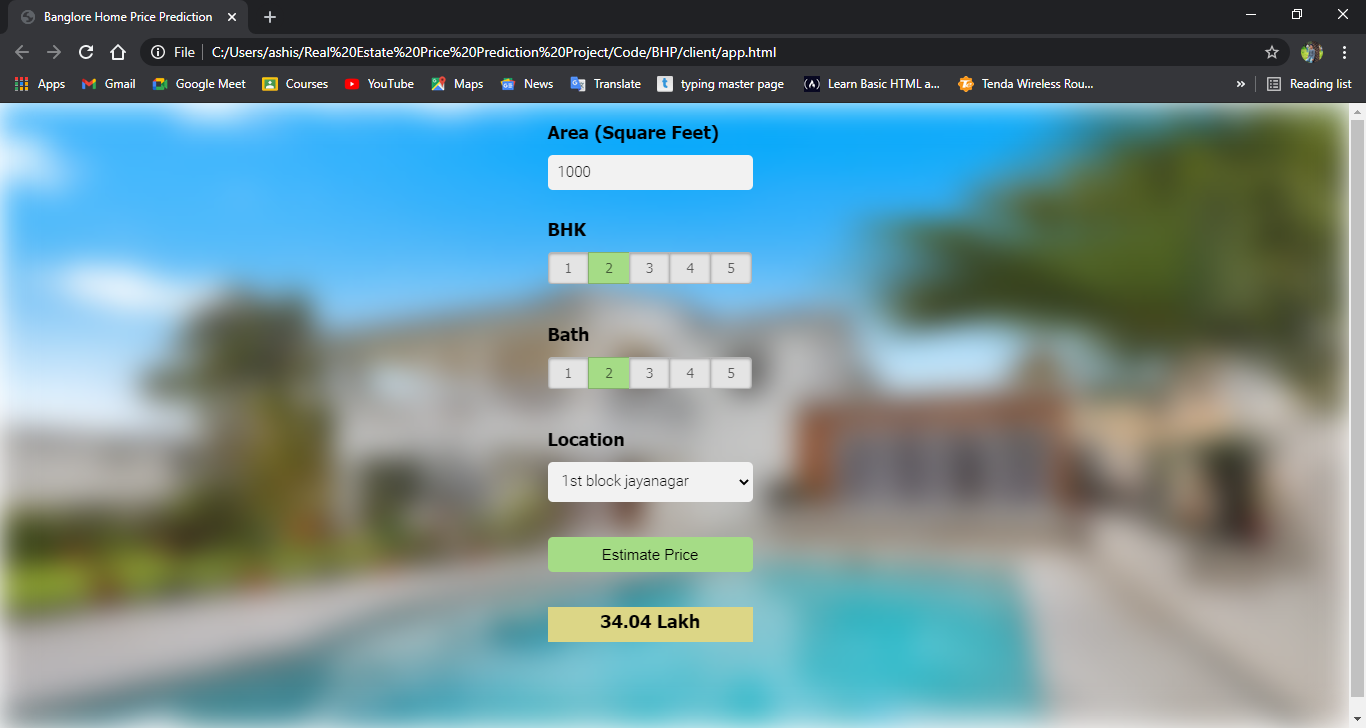
1)web Page:-



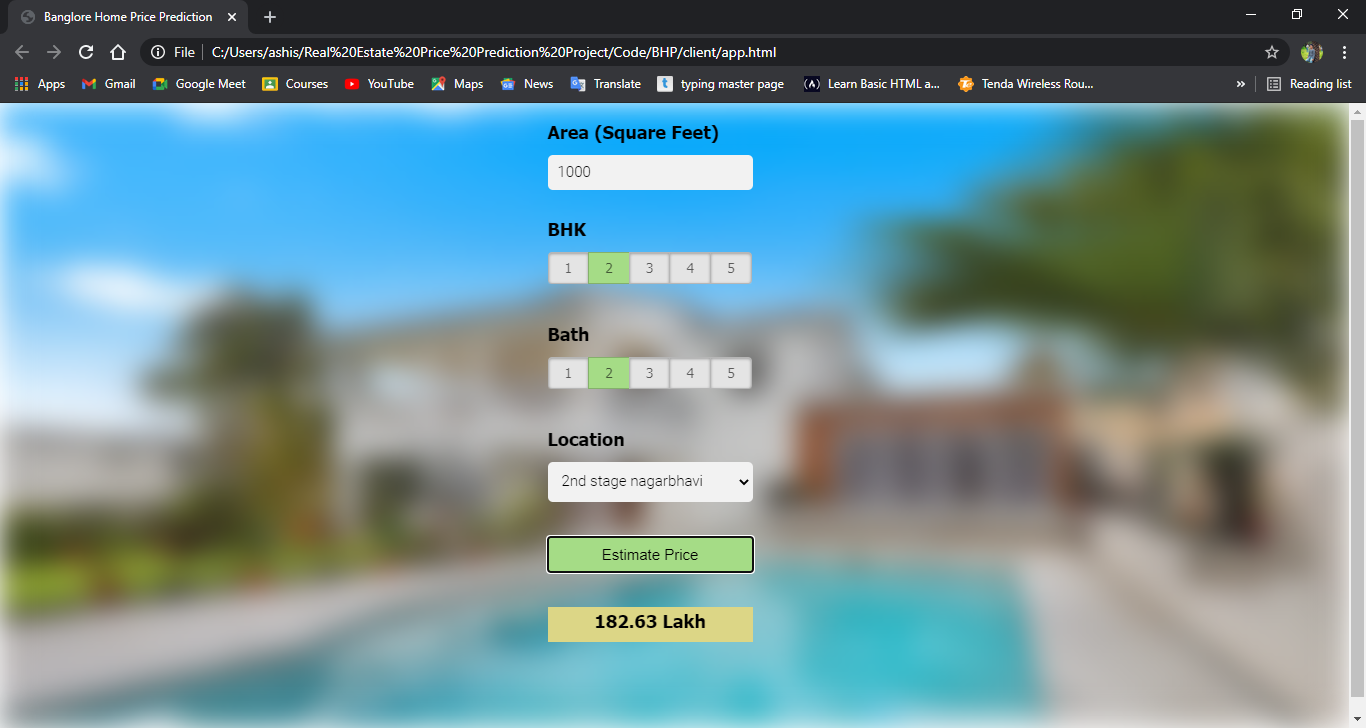
2) All Location scroll:-



3)Selection of sqft, bhk, bath:-



4)Price Prediction Depends upon bhk, bath,sqft, different location:-



CONCLUSION

So, our Aim is achieved as we have successfully ticked all our parameters as mentioned in our CONTEXT Column.

We use different model to predict best price for home in their particular location.

It is seen that circle rate is the most effective attribute in predicting the house price and that the Linear Regression is the most effective model for our Dataset with BEST score of 0.818354.

* **Bibliography**
* [Find Open Datasets and Machine Learning Projects | Kaggle](https://www.kaggle.com/datasets)
* [Towards Data Science](https://towardsdatascience.com/)
* [RxJS, ggplot2, Python Data Persistence, Caffe2, PyBrain, Python Data Access, H2O, Colab, Theano, Flutter, KNime, Mean.js, Weka, Solidity (tutorialspoint.com)](https://www.tutorialspoint.com/index.htm)
* [Learn R, Python & Data Science Online | DataCamp](https://www.datacamp.com/)
* [YouTube](https://www.youtube.com/)
* <https://www.khanacademy.org>
* Analyticsvidhya.com
* Machinelearningplus.com
* Simplilearn.com
* www.tutorialpoint.com