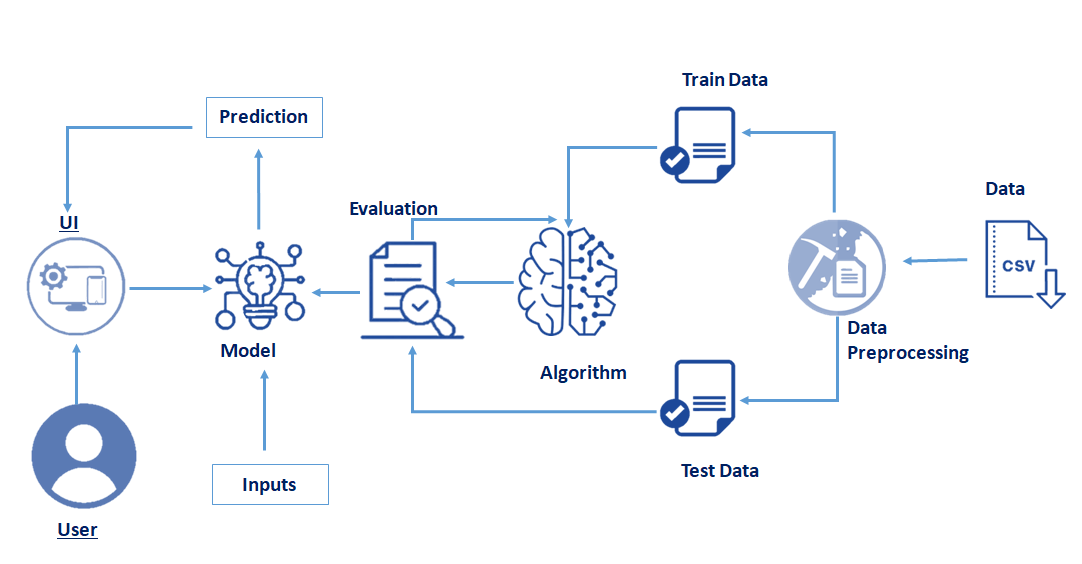
**House Rent Price Prediction**

**Project Description:**

Determining the house rent price of the house is very important nowadays as the price of the land and price of the house increases every year. So our future generation needs a simple technique to predict the house rent price in future. The price of rental house helps the people’s to know the rental price of the house based on different-different cities and also they are able to choose right place based on his budget. The right price of the rental house helps the people’s to select the house and go for that areas to bargaining of that house. There are several factors that affect the price of the rental house such as the physical condition, location, landmark etc. This project uses various regression techniques to predict the house rent such as Ridge, Lasso, ElasticNet regression, Decision tree, Random Forest techniques etc.

**Key Words:** House Price, Ridge, Lasso (Least Absolute Shrinkage and Selection Operator), Decision tree, Random Forest, Regression analysis.

**Technical Architecture:**

****

**Pre requisites:**

**To complete this project, you must require following software’s , concepts and packages**

* **Anaconda Application:**
  + Refer to the link below to download anaconda application

**Link : https://www.anaconda.com/products/individual**

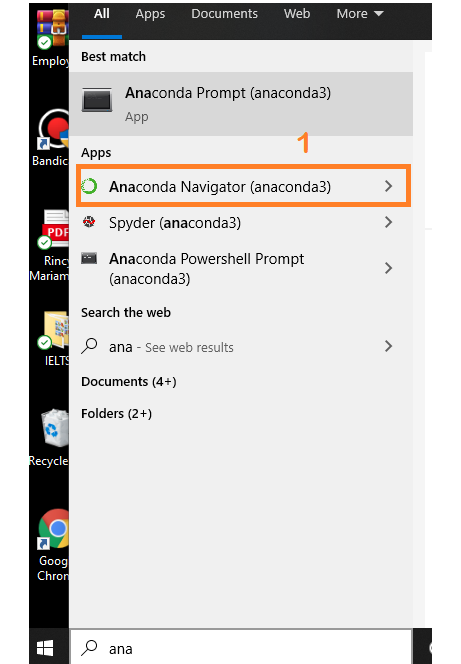
* **Python packages:**

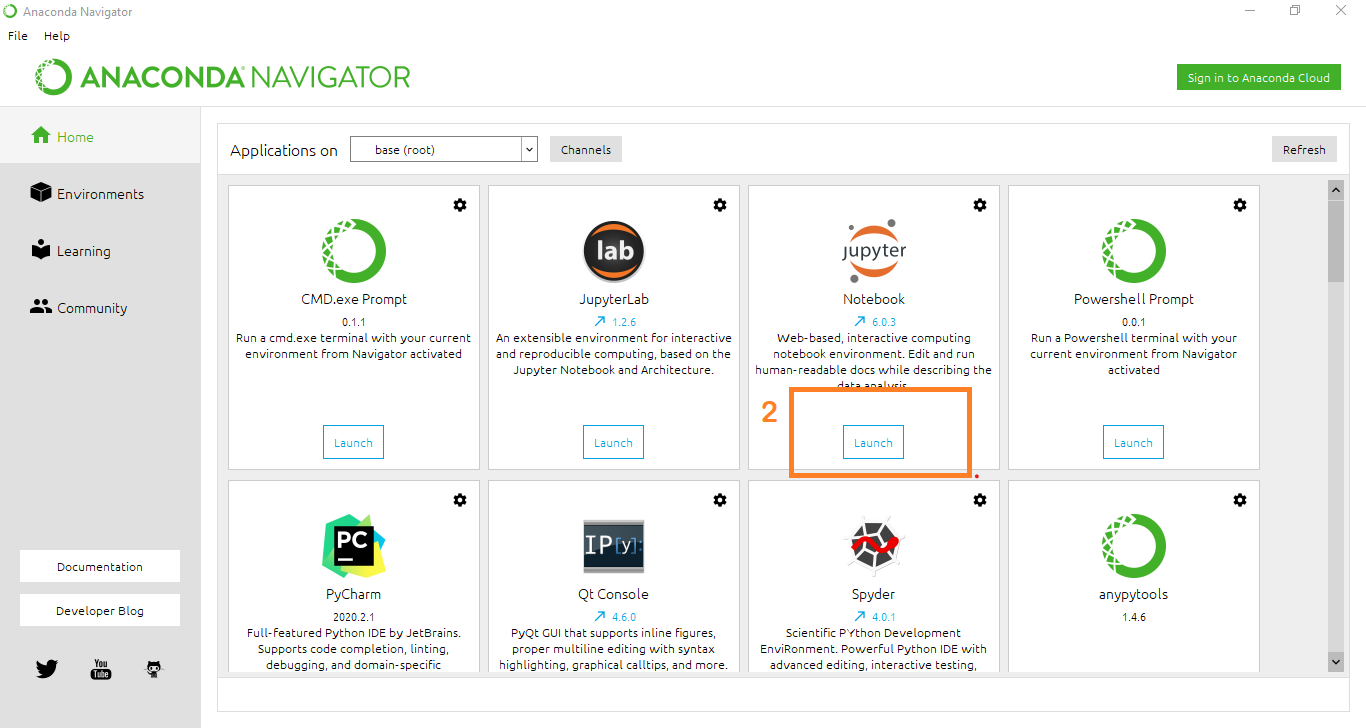
Open anaconda prompt as administrator.

* Type “pip install numpy” and click enter.
* Type “pip install pandas” and click enter.
* Type “pip install matplotlib” and click enter.
* Type “pip install scikit-learn” and click enter.
* Type “pip install Flask” and click enter.

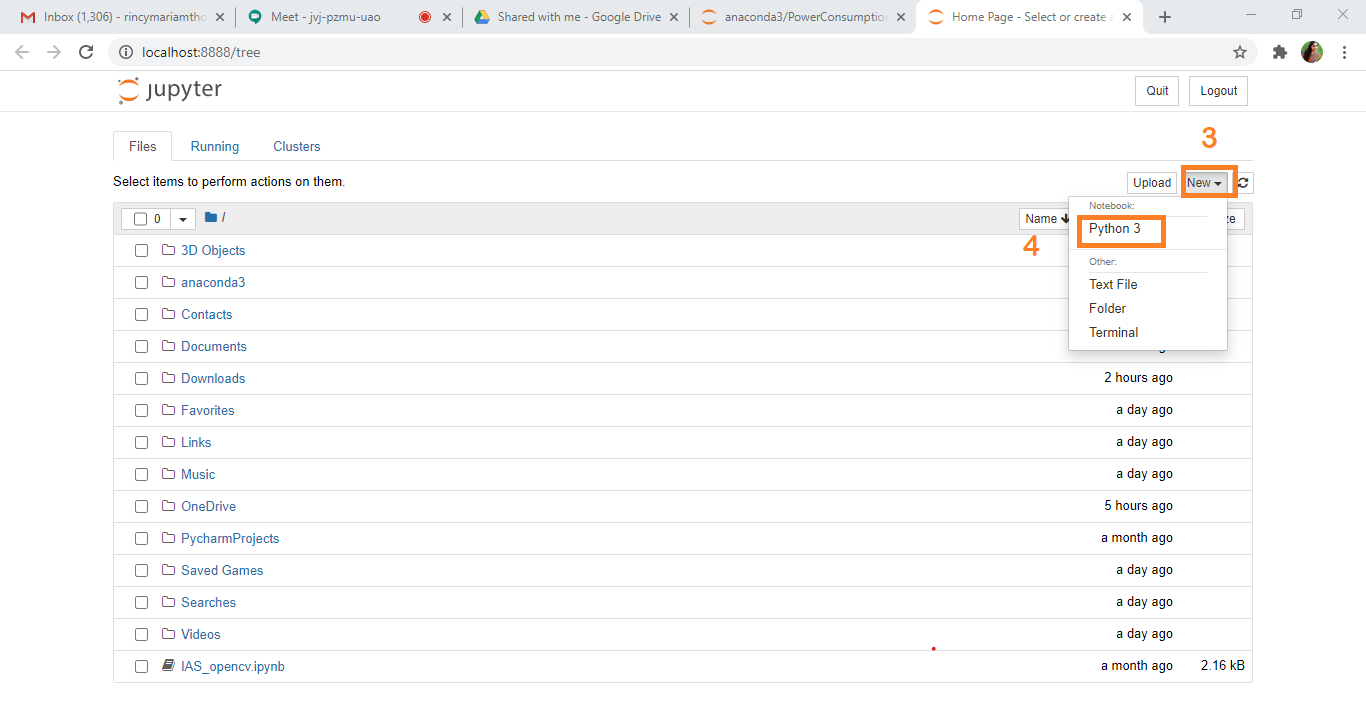
The above steps allow you to install the packages in the anaconda environment

* **Launch Jupyter**
  + Search for Anaconda Navigator and open Launch Jupyter notebook.





* Then you will be able to see that the jupyter notebook runs on local host:8888.
* To Create a new file Go to New 🡪Python3.The file in jupyter notebook is saved with .ipynb extension.



* + Flask Basics : <https://www.youtube.com/watch?v=lj4I_CvBnt0>

**Project Objectives:**

By the end of this project:

* You’ll be able to understand the problem to classify if it is a regression or a classification kind of problem.
* You will be able to know how to pre-process / clean the data using different data pre-processing techniques.
* You will able to analyse or get insights of data through visualization.
* Applying different algorithms according to dataset and based on visualization.
* You will able to know how to find accuracy of the model.
* You will be able to know how to build a web application using Flask framework.

**Project Flow:**

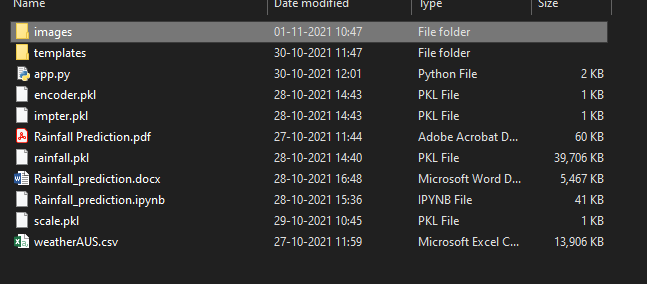
* User interacts with the UI (User Interface) to enter the input values
* Entered input values are analysed by the model which is integrated
* Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities and tasks listed below

* Data Collection.
* Collect the dataset or Create the dataset
* Exploratory Data Analysis (**EDA**).
  + Import the Libraries.
  + Importing the dataset.
  + Data Cleaning.
  + Data Visualization.
  + Outlier Treatment.
  + Data Transformation.
  + Encoding
  + Splitting Data into Train and Test.
* Model Building
  + Import the model building Libraries
  + Initializing the model
  + Training and testing the model
  + Evaluation of Model
  + Save the Model
* Application Building
  + Create an HTML file
  + Build a Python Code

**Project Structure:**

Create a Project folder which contains files as shown below

****

* A python file called app.py for server side scipting.
* We need the model which is saved and the saved model in this content is **Rainfall.pkl**
* Templates folder which contains index.HTML file, chance.HTML file, noChance.HTML file.
* Scale.pkl for scaling,encoder.pkl file for encoding the categorical data,imputer.pkl file for filling out the missing values

**Milestone 1: Data Collection:**

ML depends heavily on data, without data, it is impossible for an “AI” to learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training **data set.**It is the actual **data set** used to train the model for performing various actions.

**Activity1: Download The dataset**

Please refer to the link given below to download the data set and to know about the dataset

<https://drive.google.com/file/d/1PvFtYozEkP_duHKjB65Xnmh5H7oImZv9/view?usp=sharing>

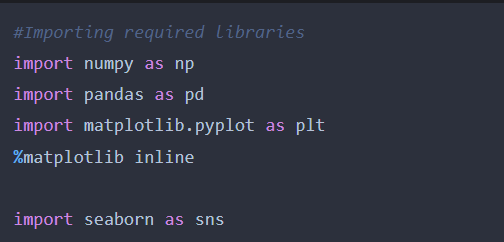
**Milestone 2: Data Pre-processing /**  **Exploratory Data Analysis** (**EDA**)

Data Pre-processing includes the following main tasks

* + Import the Libraries.
  + Importing the dataset.
  + Data Cleaning.
  + Data Visualization.
  + Outlier Treatment.
  + Data Transformation.
  + Encoding
  + Splitting Data into Train and Test.

**Activity 1: Import Necessary Libraries**

* + It is important to import all the necessary libraries such as pandas, numpy, matplotlib.
  + **Numpy**- It is an open-source numerical Python library. It contains a multi-dimensional array and matrix data structures. It can be used to perform mathematical operations on arrays such as trigonometric, statistical, and algebraic routines.
  + **Pandas**- It is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.
  + **Seaborn**- Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
  + **Matplotlib**- Visualisation with python. It is a comprehensive library for creating static,animated, and interactive visualizations in Python
  + **Sklearn** – which contains all the modules required for model building



**Activity 2: Importing the Dataset**

* You might have your data in .csv files, .excel files
* Let’s load a .csv data file into pandas using **read\_csv() function.**We will need to locate the directory of the CSV file at first (it’s more efficient to keep the dataset in the same directory as your program).
* If your dataset is in some other location ,Then

**Data=pd.read\_csv(r”File\_location/datasetname.csv”)**

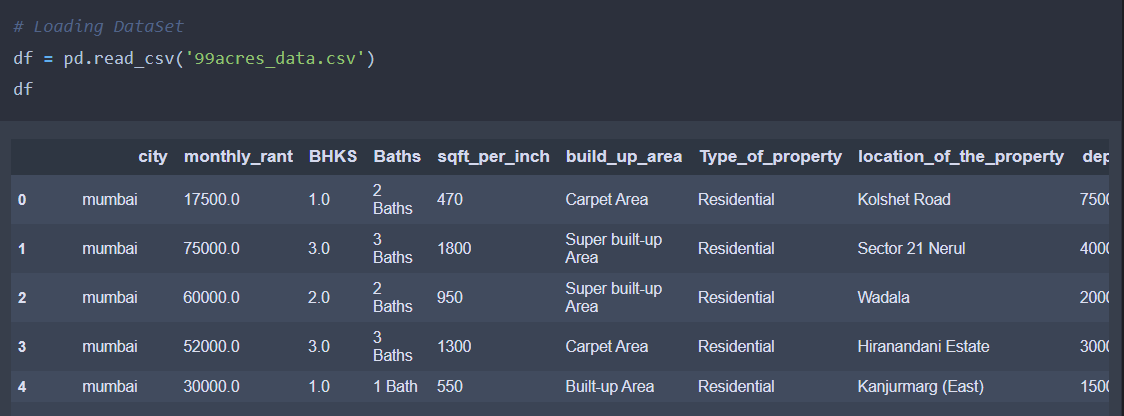
**Note:**r stands for "raw" and will cause backslashes in the string to be interpreted as actual backslashes rather than special characters.

* If the dataset in same directory of your program, you can directly read it, without giving raw as r.
* Our Dataset 99acres\_data.csv contains following Columns
* city, monthly\_rant, BHKS, sqft\_per\_inch, build\_up\_area,
* Type\_of\_property, deposit
* monthly\_rant – output column

The output column to be predicted is **House** **Rent Price** .Based on the input variables we predict the House rent. The predicted output gives them a fair idea about the rent of the house based on given cities.

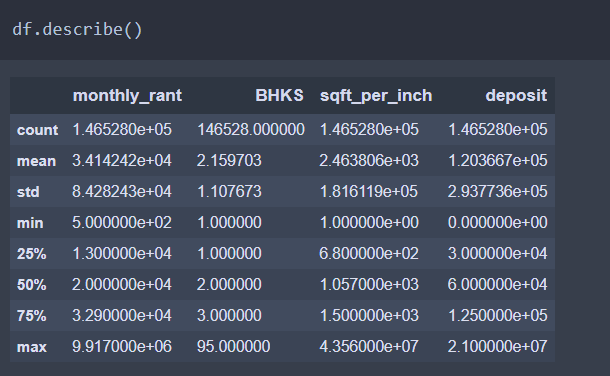
**Activity 3: Analyse the data**

* head() method is used to return top n (5 by default) rows of a DataFrame or series.



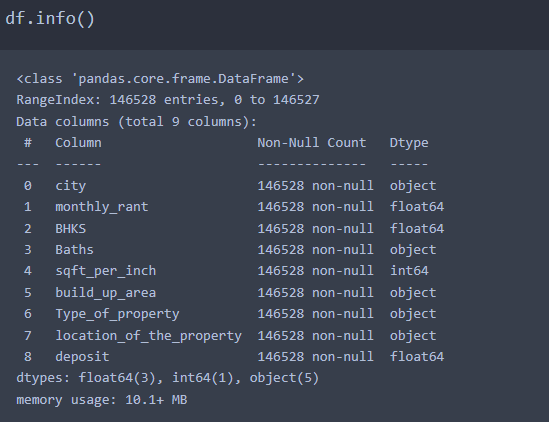
* describe() method computes a summary of statistics like count, mean, standard deviation, min, max and quartile values.

The output is as shown below



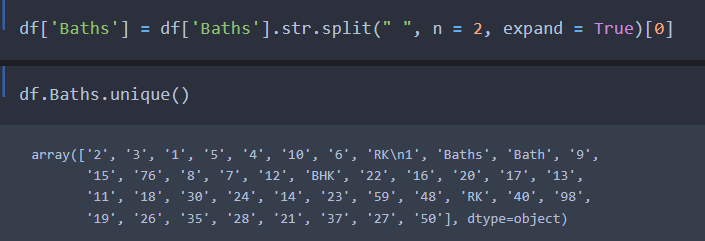
From the data we infer that there are only decimal values and no categorical values

* info() gives information about the data



**Activity 4: Data Cleaning**

1. We need to keep only meaningful information inside the data column.



Previously we have Numbers and Alphabets both are present in our column but now we have numbers but also some categories are present.

**Activity 5: Data Visualisation**

* Data visualization is where a given data set is presented in a graphical format. It helps the detection of patterns, trends and correlations that might go undetected in text-based data.
* Understanding your data and the relationship present within it is just as important as any algorithm used to train your machine learning model. In fact, even the most sophisticated machine learning models will perform poorly on data that wasn’t visualized and understood properly.
* To visualize the dataset we need libraries called Matplotlib and Seaborn.
* The Matplotlib library is a Python 2D plotting library which allows you to generate plots, scatter plots, histograms, bar charts etc.

Let’s visualize our data using Matplotlib and searborn library.

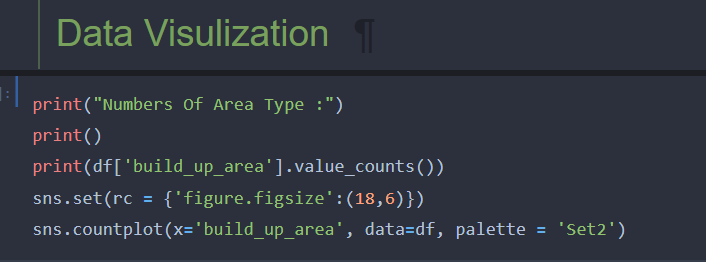
Before diving into the code, let's look at some of the basic properties we will be using when plotting.

**xlabel:** Set the label for the x-axis.

**ylabel:** Set the label for the y-axis.

**title:** Set a title for the axes.

**Legend:** Place a legend on the axes.



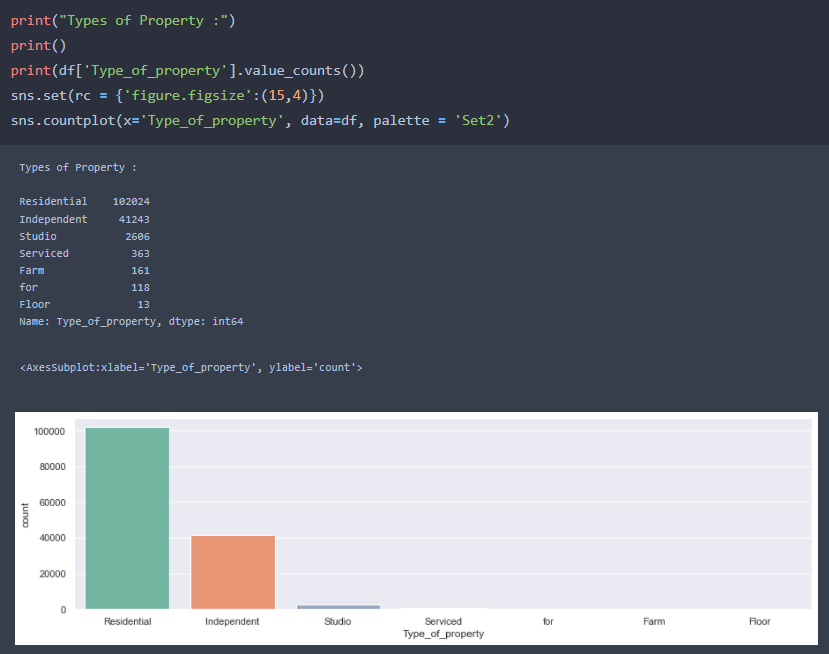
We are checking what is the no of count present in different-different **build\_up\_area**

Output :-

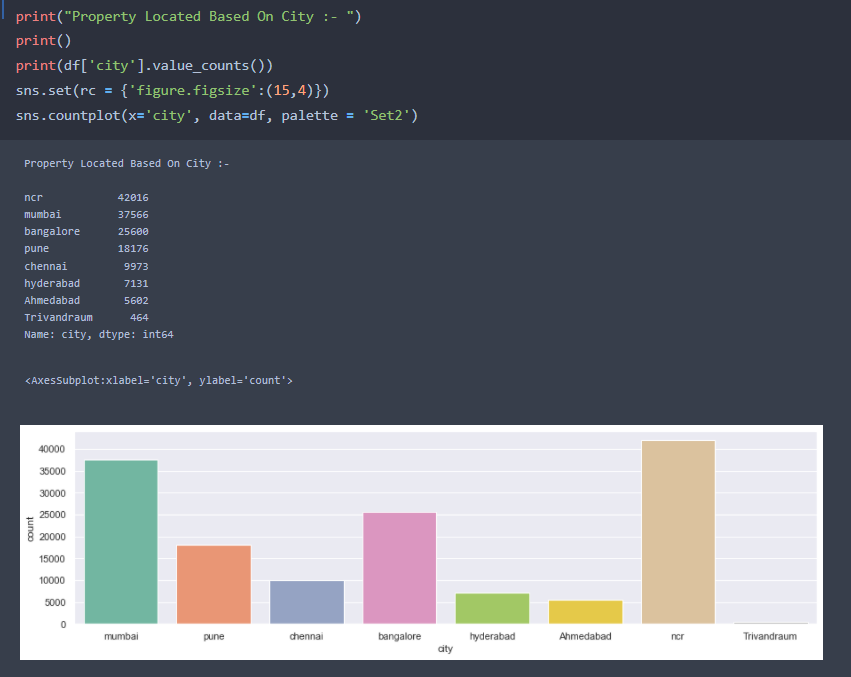


Visual representation helps us to easily analyse the values.

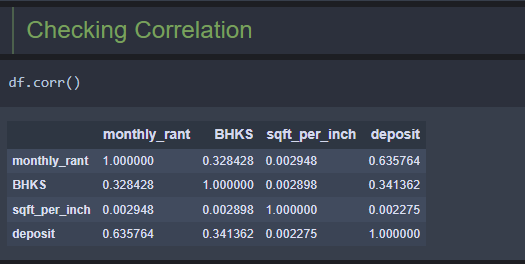
Similarly, we are checking what is the count of **Type\_of\_property**



And how many property located based on city.



**df.corr()** gives the correlation between the columns





* Correlation strength varies based on colour, lighter the colour between two variables, more the strength between the variables, darker the colour displays the weaker correlation
* We can see the correlation scale values on left side of the above image

**Pair Plot**: Plot pairwise relationships in a dataset.

* By default, this function will create a grid of Axes such that each numeric variable in data will by shared across the y-axes across a single row and the x-axes across a single column. The diagonal plots are treated differently: a univariate distribution plot is drawn to show the marginal distribution of the data in each column.
* We implement this using the below code
* We will able to see upto six dimensional data using pair plot.

**Code:- sns.pairplot(df)**

The output is as shown below



Pair plot usually gives pair wise relationships of the columns in the dataset

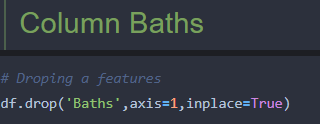
From the above pairplot we infer that

1.from the above plot we can draw inferences such as linearity and strength between the variables

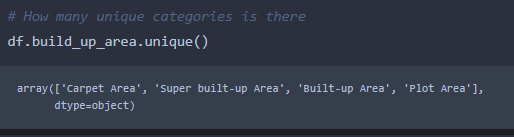
2.how features are correlated(positive, neutral and negative)

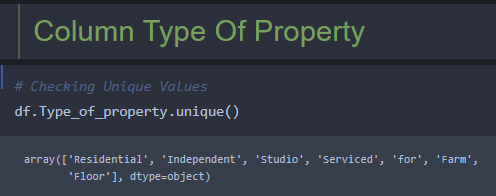
**Activity 6: Cleaning data (Dropping and Filtering)**

**Droping Baths columns because as compare to BHKS it contain almost same values.**

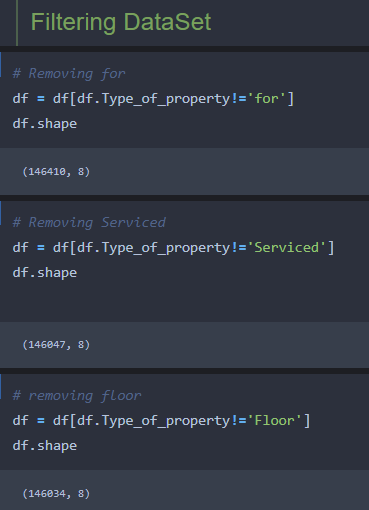


**Checking Unique values presents in columns**

****

****

In this column some values not making any sense so we need to filter it with checking percentage of the data.



**Activity 7: Outliers Treatment**

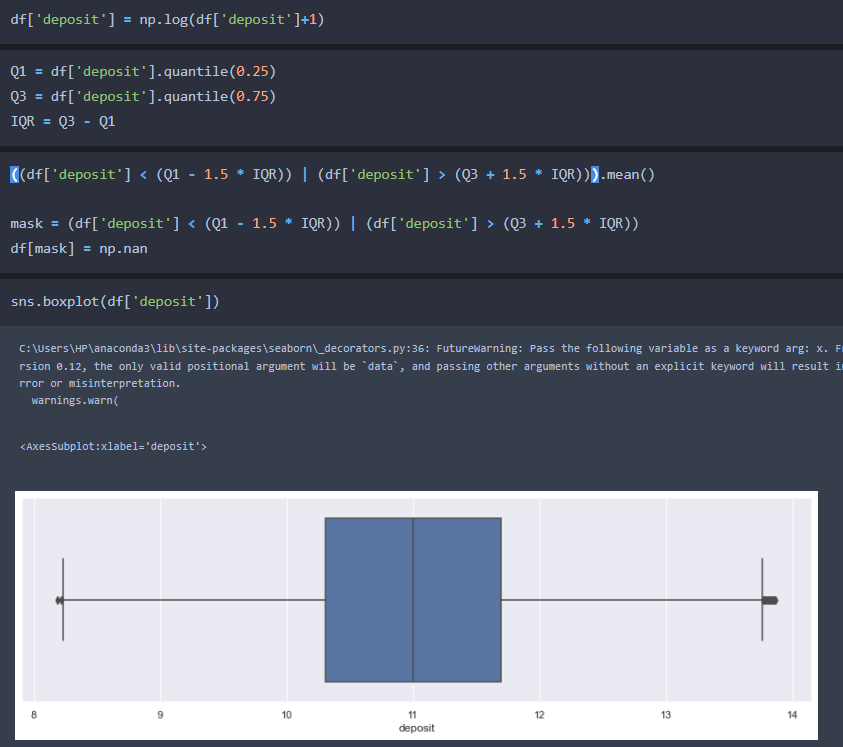
* In statistics, an outlier is **a data point that differs significantly from other observations**. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set. An outlier can cause serious problems in statistical analyses.
* ***Box Plot*** is the visual representation of the depicting groups of numerical data through their quartiles. Boxplot is also used for detect the outlier in data set. It captures the summary of the data efficiently with a simple box and whiskers and allows us to compare easily across groups. Boxplot summarizes a sample data using 25th, 50th and 75th percentiles. These percentiles are also known as the lower quartile, median and upper quartile.
* A box plot consist of 5 things.
* Minimum
* First Quartile or 25%
* Median (Second Quartile) or 50%
* Third Quartile or 75%
* Maximum

**If the data points present before minimum and after maximum value we considered as a outlier**

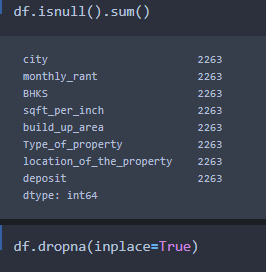
****

**We are removed that outlier with the help of IQR interquartile range.**

**For reference watch this video: -**  <https://youtu.be/A3gClkblXK8>



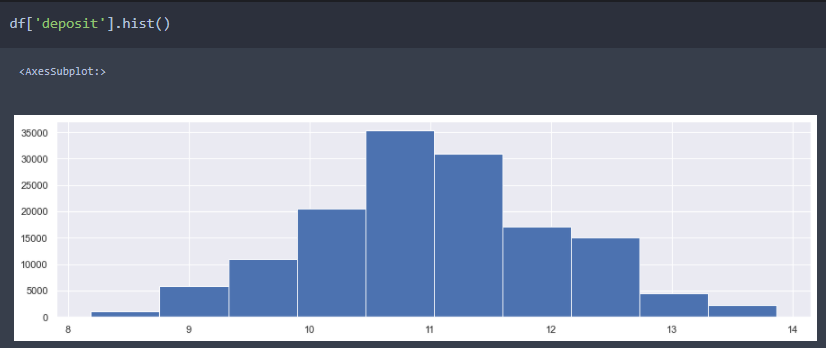
After removing the outlier we got sum null values. We are directly remove those because the percentage of data is very less.



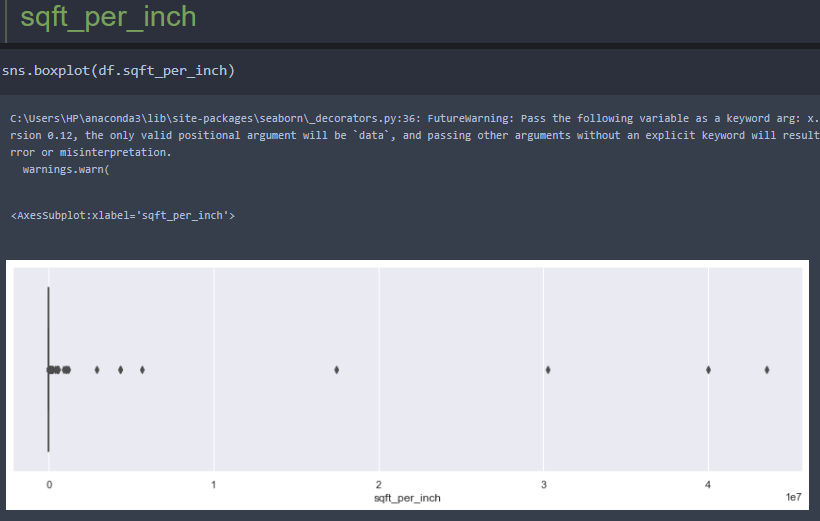
**Checking the distribution of the data:-**

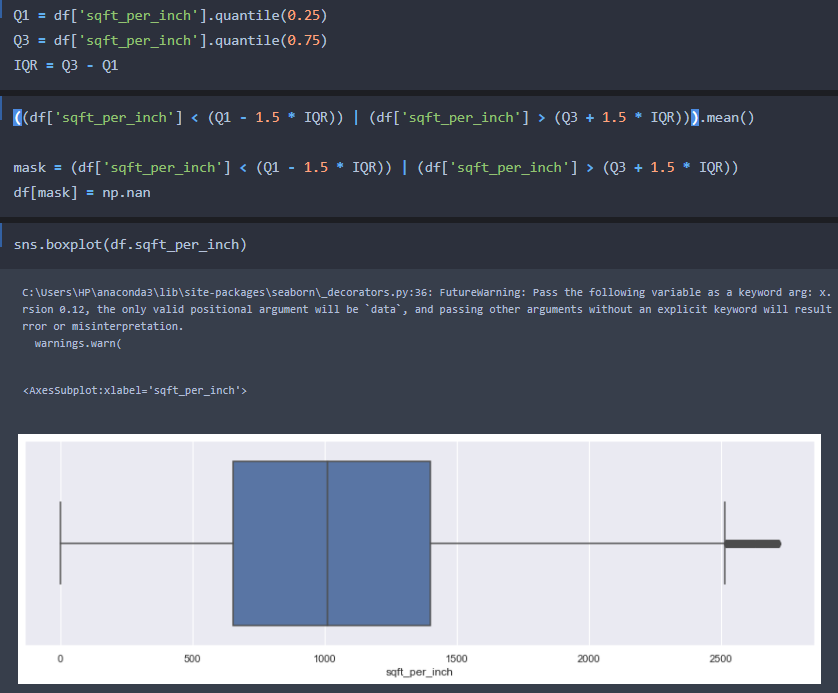
* Data is looking like almost normal distributed.
* If you want to know more about the normal distribution follow this link :

<https://youtu.be/rzFX5NWojp0>

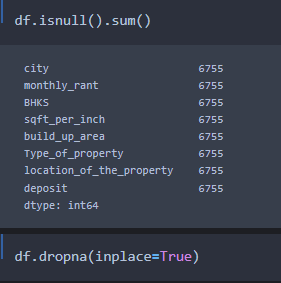
****

**Similarly,** We apply for another column

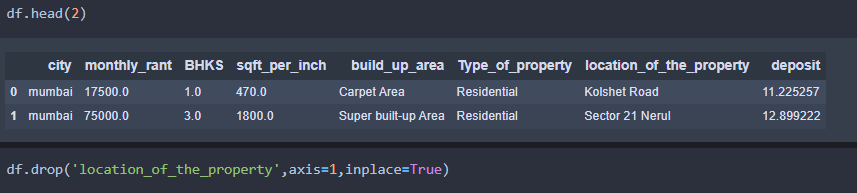




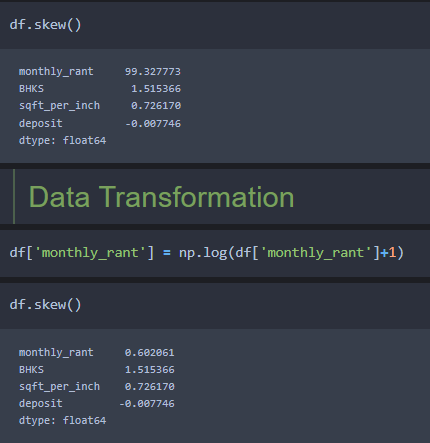
And dropping some rows



Also we need to drop one more column location\_of\_the\_property.



**Checking skewness of the column.**

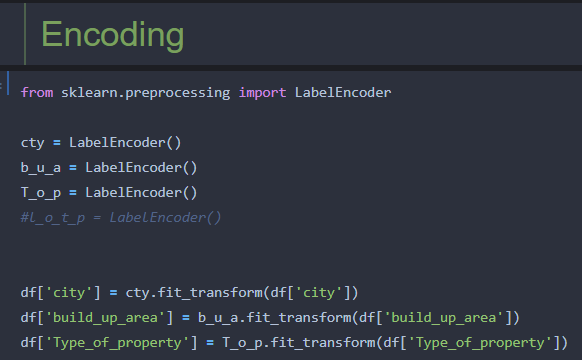
****

**Click this link to know more about skewness:-** [**https://youtu.be/0djtjjy12fI**](https://youtu.be/0djtjjy12fI)

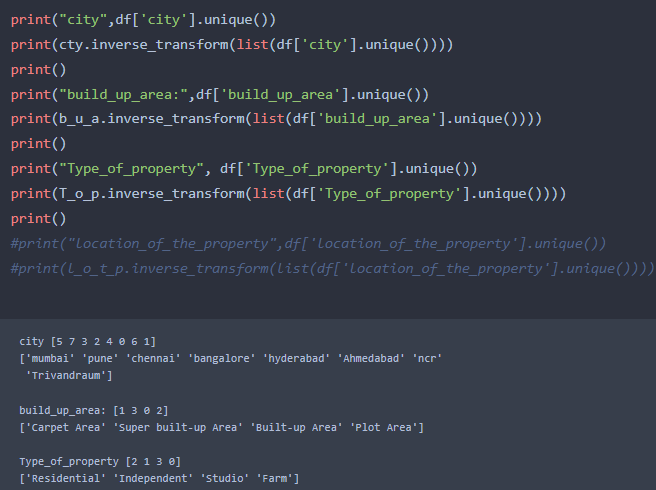
So the monthy\_rant is highly positive skewed

**For that we applied logarithmic transformation**

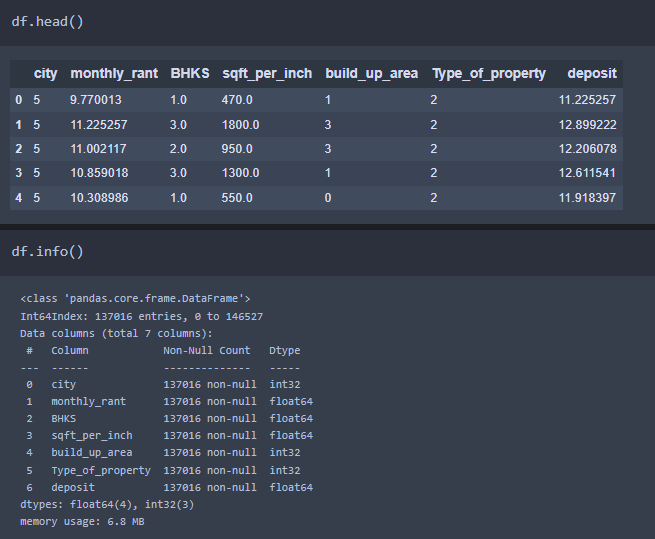
**Activity 8: Encoding: -** Encoding is a technique to convert categorical variable to numeric form. Click this link to know more about Encoding :- <https://youtu.be/OTPz5plKb40>



**Inverse\_transformation :-** It is a technique to see what value it was given to that particular category.

****

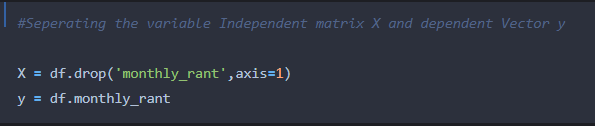
**After applied encoding dataset is looks like**

****

**Activity 6: Separating the Dataset into Dependent and Independent variable**

* In machine learning, the concept of dependent variable (y) and independent variables(x) is important to understand. Here, Dependent variable is nothing but output in dataset and independent variable is all inputs in the dataset.
* With this in mind, we need to split our dataset into the matrix of independent variables and the vector or dependent variable. Mathematically, Vector is defined as a matrix that has just one column.

Let’s split our dataset into independent and dependent variables.

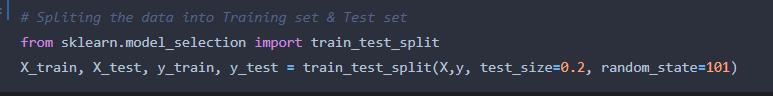


**Activity 8: Splitting the data into Train and Test**

* When you are working on a model and you want to train it, you obviously have a dataset. But after training, we have to test the model on some test dataset. For this, you will a dataset which is different from the training set you used earlier. But it might not always be possible to have so much data during the development phase. In such cases, the solution is to split the dataset into two sets, one for training and the other for testing.
* But the question is, how do you split the data? You can’t possibly manually split the dataset into two sets. And you also have to make sure you split the data in a random manner. To help us with this task, the Scikit-learn library provides a tool, called the Model Selection library. There is a class in the library which is,**‘[train\_test\_split](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html" \t "_blank).’** Using this we can easily split the dataset into the training and the testing datasets in various proportions.
* The train-test split is a technique for evaluating the performance of a machine learning algorithm.
* **Train Dataset**: Used to fit the machine learning model.
* **Test Dataset**: Used to evaluate the fit machine learning model.
* In general you can allocate 80% of the dataset to training set and the remaining 20% to test set.We will create 4 sets— X\_train (training part of the matrix of features), X\_test (test part of the matrix of features), Y\_train (training part of the dependent variables associated with the X train sets, and therefore also the same indices), Y\_test (test part of the dependent variables associated with the X test sets, and therefore also the same indices.
* There are a few other parameters that we need to understand before we use the class:
* **test\_size** — this parameter decides the size of the data that has to be split as the test dataset. This is given as a fraction. For example, if you pass 0.5 as the value, the dataset will be split 50% as the test dataset
* **train\_size** — you have to specify this parameter only if you’re not specifying the test\_size. This is the same as test\_size, but instead you tell the class what percent of the dataset you want to split as the training set.
* **random\_state** — here you pass an integer, which will act as the seed for the random number generator during the split. Or, you can also pass an instance of the Random\_state class, which will become the number generator. If you don’t pass anything, the Random\_state instance used by np.random will be used instead.
* Now split our dataset into train set and test using train\_test\_split class from scikit learn library.

**from sklearn import model\_selection**

**x\_train,x\_test,y\_train,y\_test=model\_selection.train\_test\_split(x,y,test\_size=0.2,random\_state =0)**

****

**Milestone 3: Model Building:**

Model building includes the following main tasks

* + Import the model building Libraries
  + Initializing the model
  + Training and testing the model
  + Evaluation of Model
  + Save the Model

**Activity 1: Training and Testing the Model**

* Once after splitting the data into train and test, the data should be fed to an algorithm to build a model.
* There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have it may be Classification algorithms are Regression algorithms.

**For Regression kind of problem some examples models are:-**

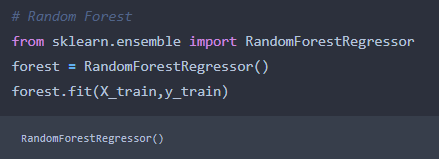
1. Linear Regression
2. Polynomial Regression
3. Ridge Regression
4. Lasso Regression
5. ElasticNet regression
6. Decision Tree Regressor
7. Random Forest Regressor

**For Classification kind of problem some examples models are:-**

* + 1. Logistic Regression
    2. Decision Tree Classifier
    3. Random Forest Classifier
    4. KNN
    5. svm
    6. xgboost

**Steps in Building the model:-**

* **Initialize the model**
* **Fit the models with x\_train and y\_train**
* **Predict the y\_train values and calculate the accuracy**
* **Predict the y\_test values and calculate the accuracy**

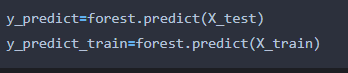
****

We’re going to use x\_train and y\_train obtained above in train\_test\_split section to train our Random Forest regression model. We’re using the fit method and passing the parameters as shown below.

We are using the algorithm from Scikit learn library to build the model as shown below,

Once the model is trained, it’s ready to make predictions. We can use the **predict** method on the model and pass **x\_test** as a parameter to get the output as **y\_pred.**

Notice that the prediction output is an array of real numbers corresponding to the input array.



**Activity 2: Model Evaluation**

After training the model, the model should be tested by using the test data which is been separated while splitting the data for checking the functionality of the model.

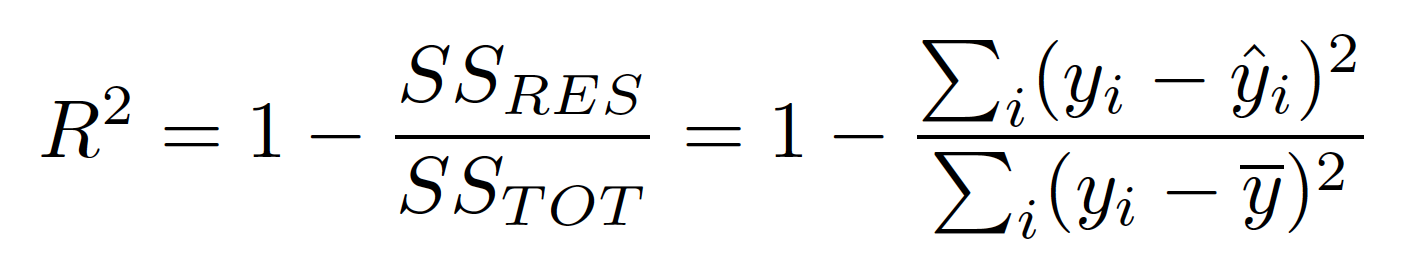
**Regression Evaluation Metrics:**

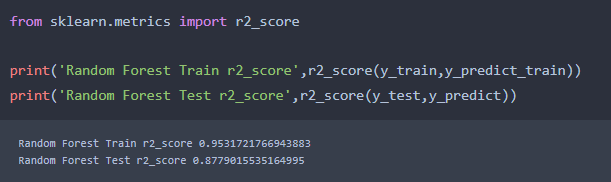
These model evaluation techniques are used to find out the accuracy of models built in classification type of machine learning models. We have three types of evaluation methods.

* R2-score
* MSE & MAE
* RMSE

1. R2 score

It is the ratio of number of correct predictions to the total number of input samples.



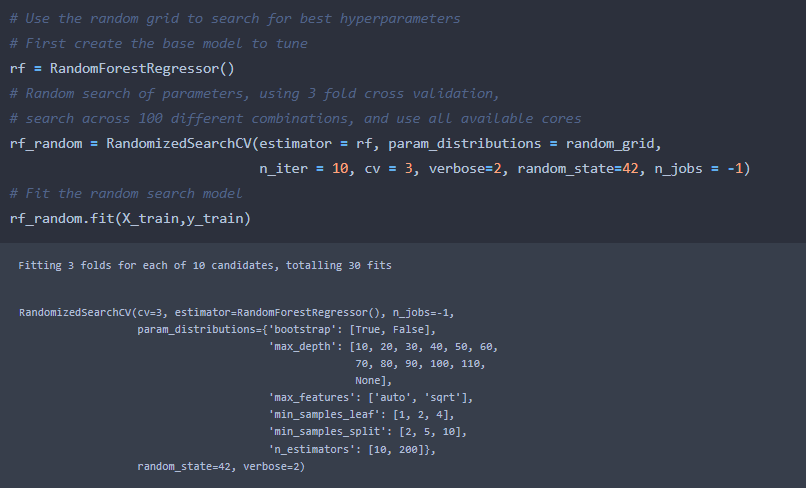


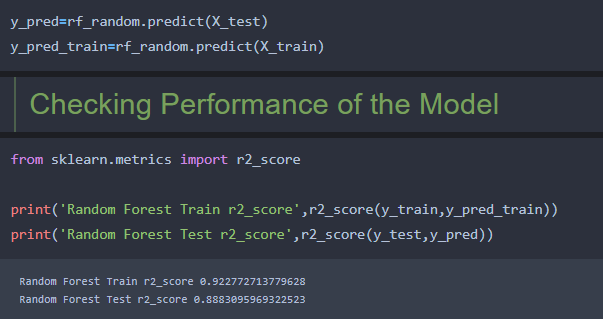
We need to select good parameters to get good accuracy of the model. For that we have some technique.

One of the technique called **hyper parameter tuning**

**For that we are using RandomizeSearchCV**

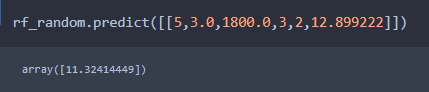






**Now we can see the model accuracy is increased little bit.**

**Predicting Using Model**

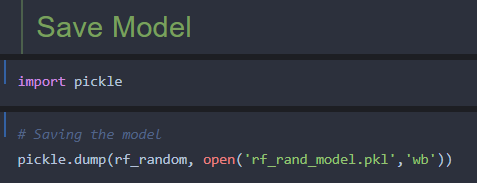


**Activity 3: Save the Model**

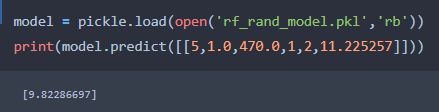
After building the model we have to save the model.

**Pickle** in **Python** is primarily **used** in serializing and deserializing a **Python** object structure. In other words, it's the process of converting a **Python** object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network. wb indicates write method and rd indicates read method.

This is done by the below code

****

**Predicting By Loading model**

****

**Milestone 4: Application Building**

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

* Building HTML Pages
* Building server side script

**Activity 1: Build HTML Code**

* + In this HTML page, we will create the front end part of the web page. In this page we will accept input from the user and Predict the values.

For more information regarding HTML

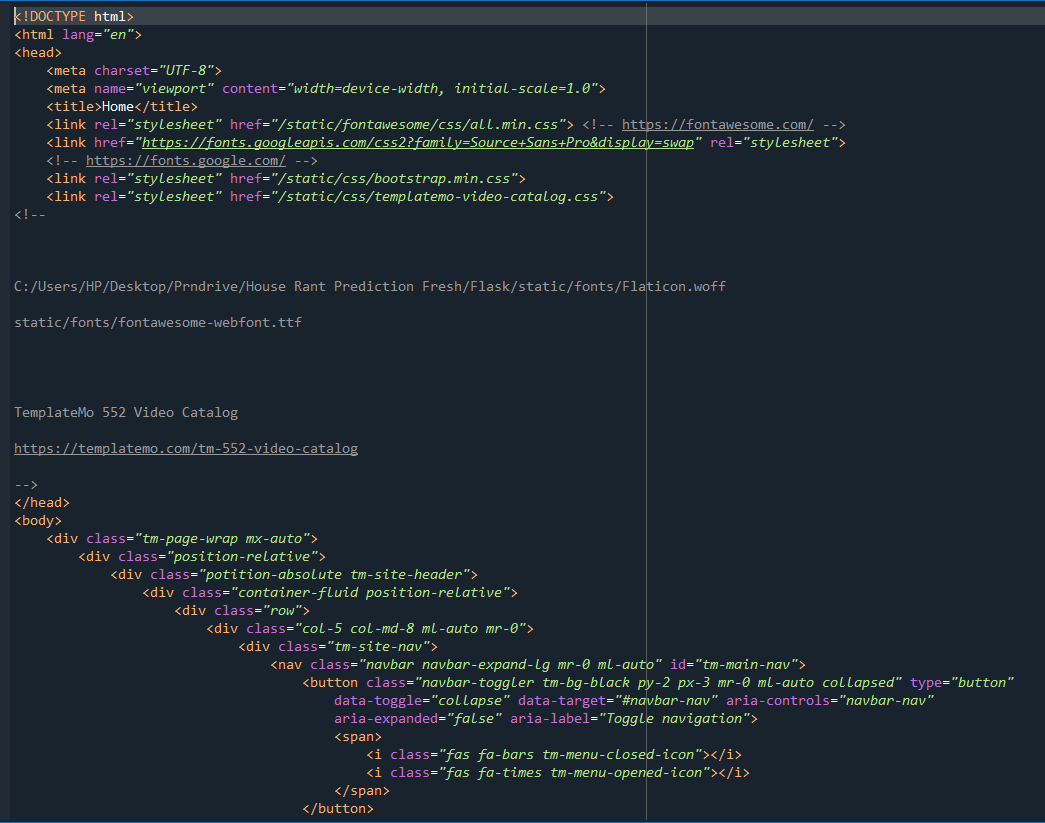
[**https://www.w3schools.com/html/**](https://www.w3schools.com/html/)

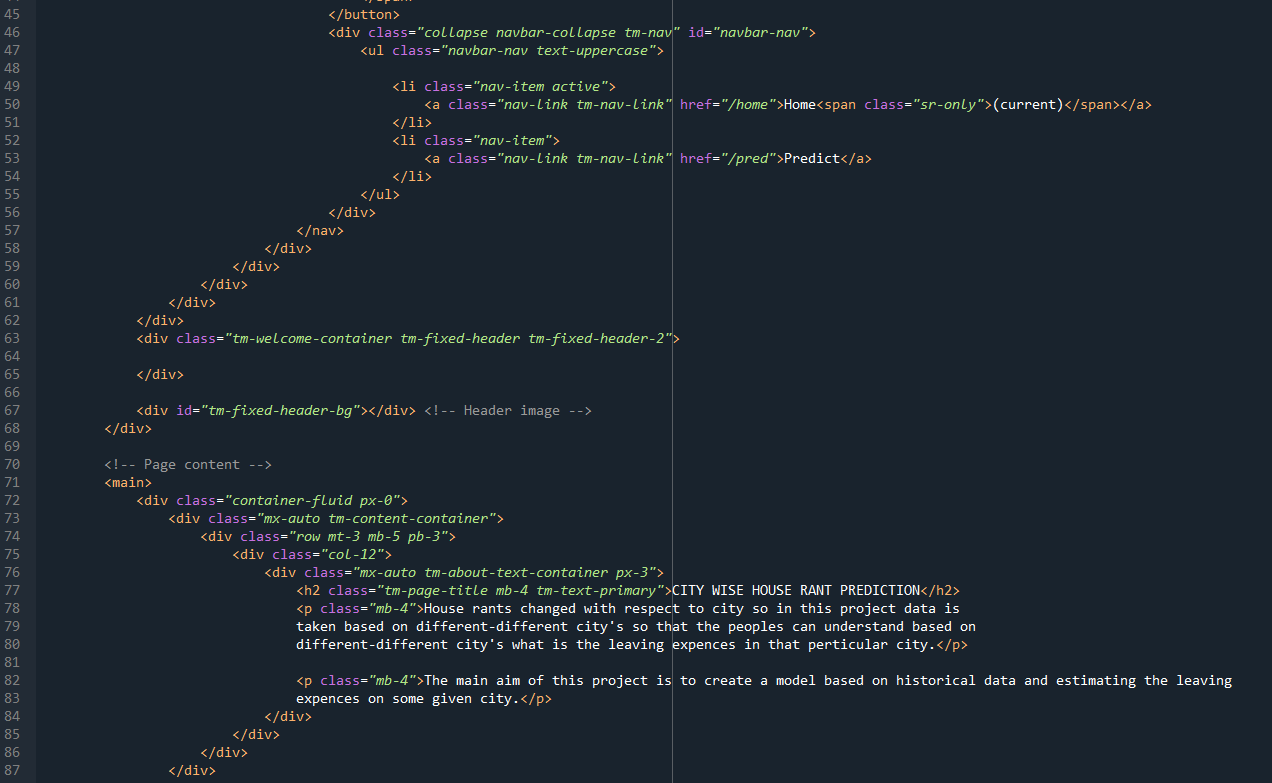
In our project we have 3 HTML files ,they are

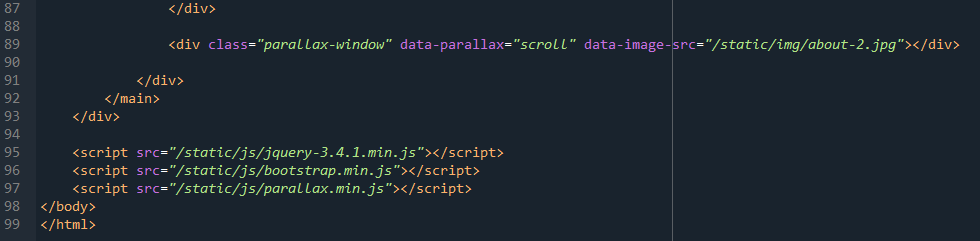
1.home.html

2.upload.html

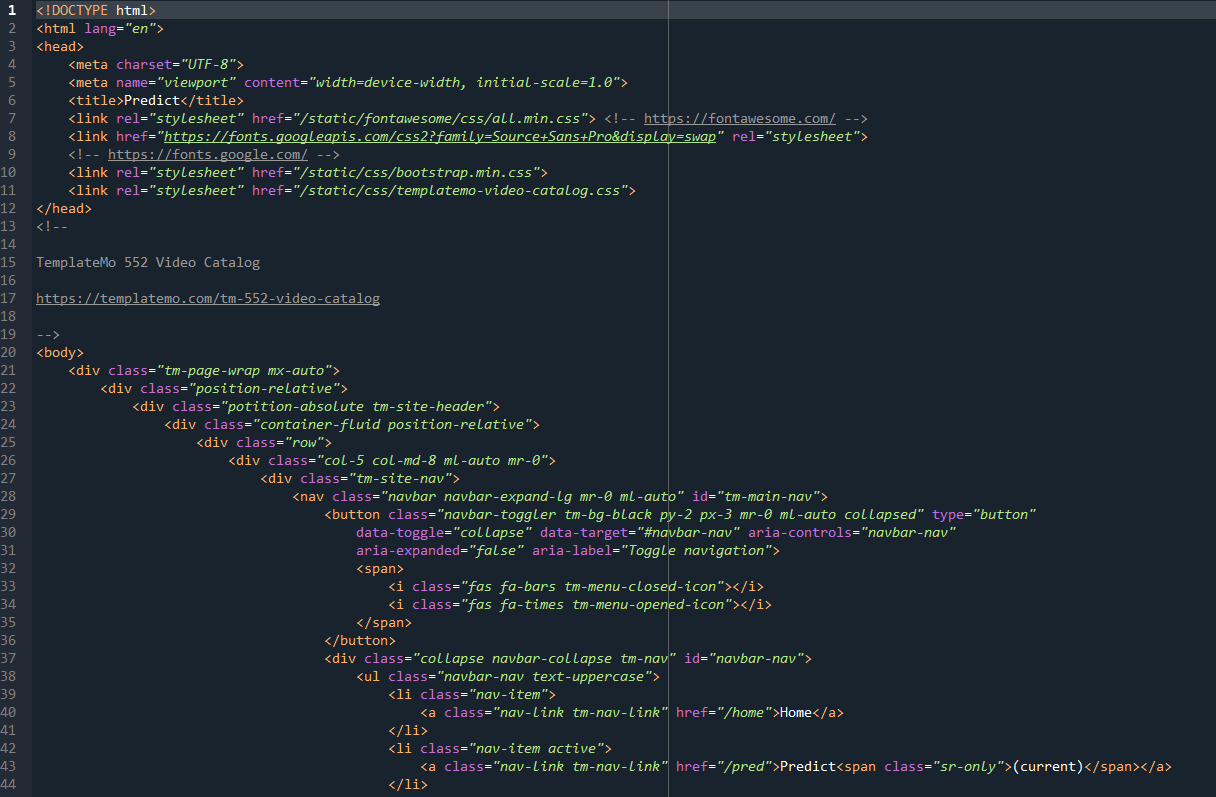
**home.html**

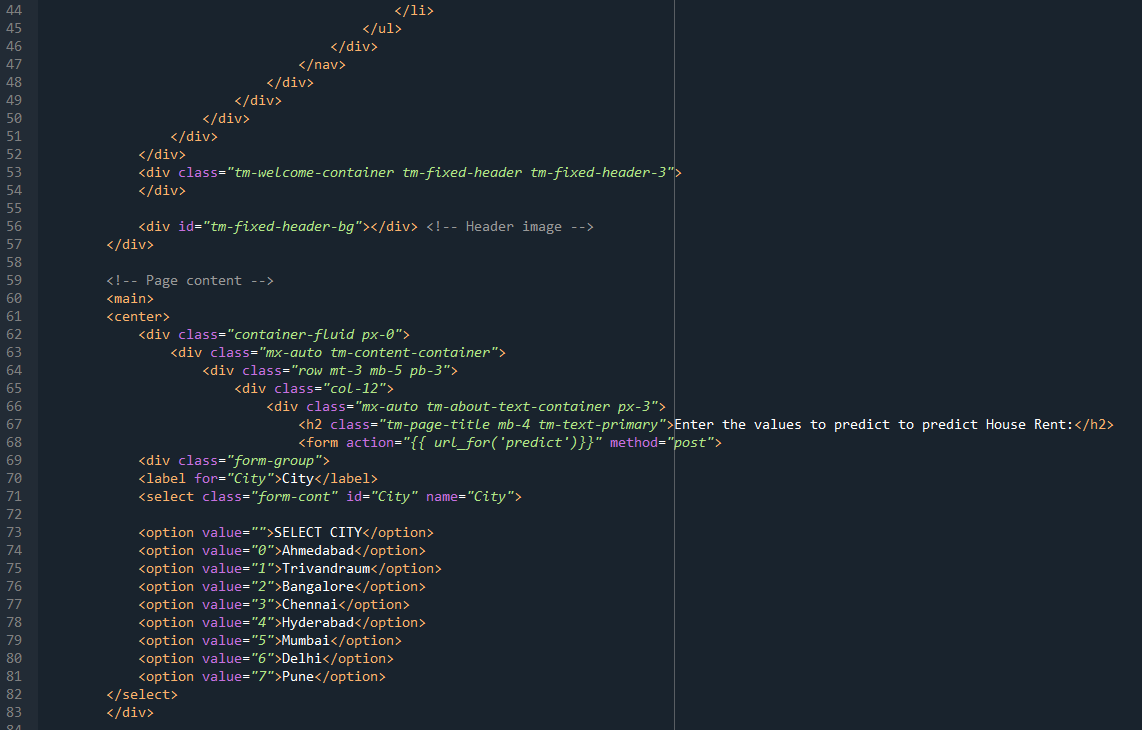
****

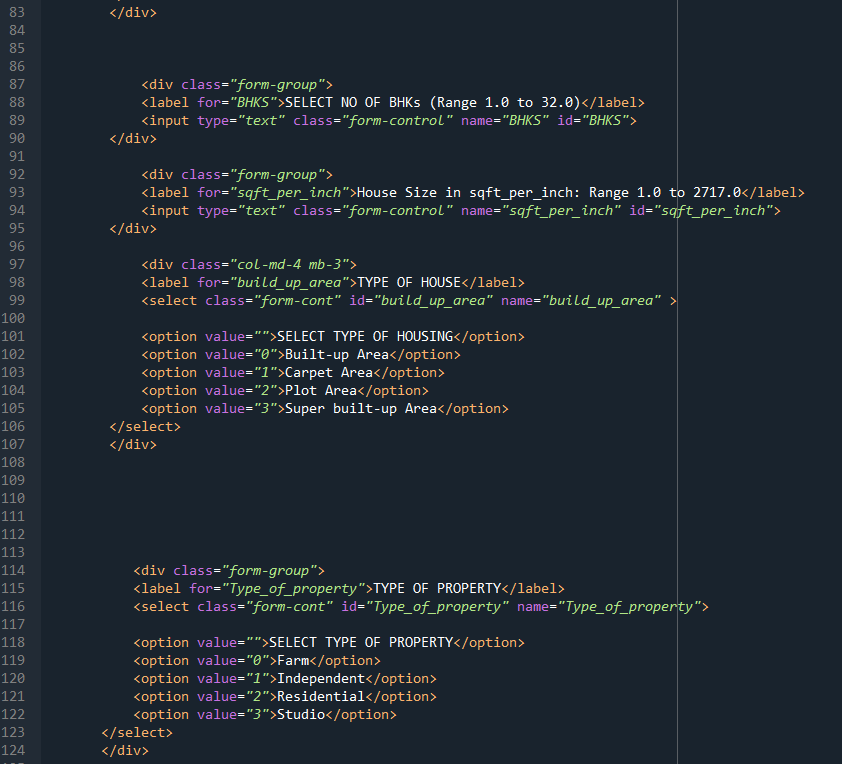
****

****

**Upload.html**

****

****

****

****

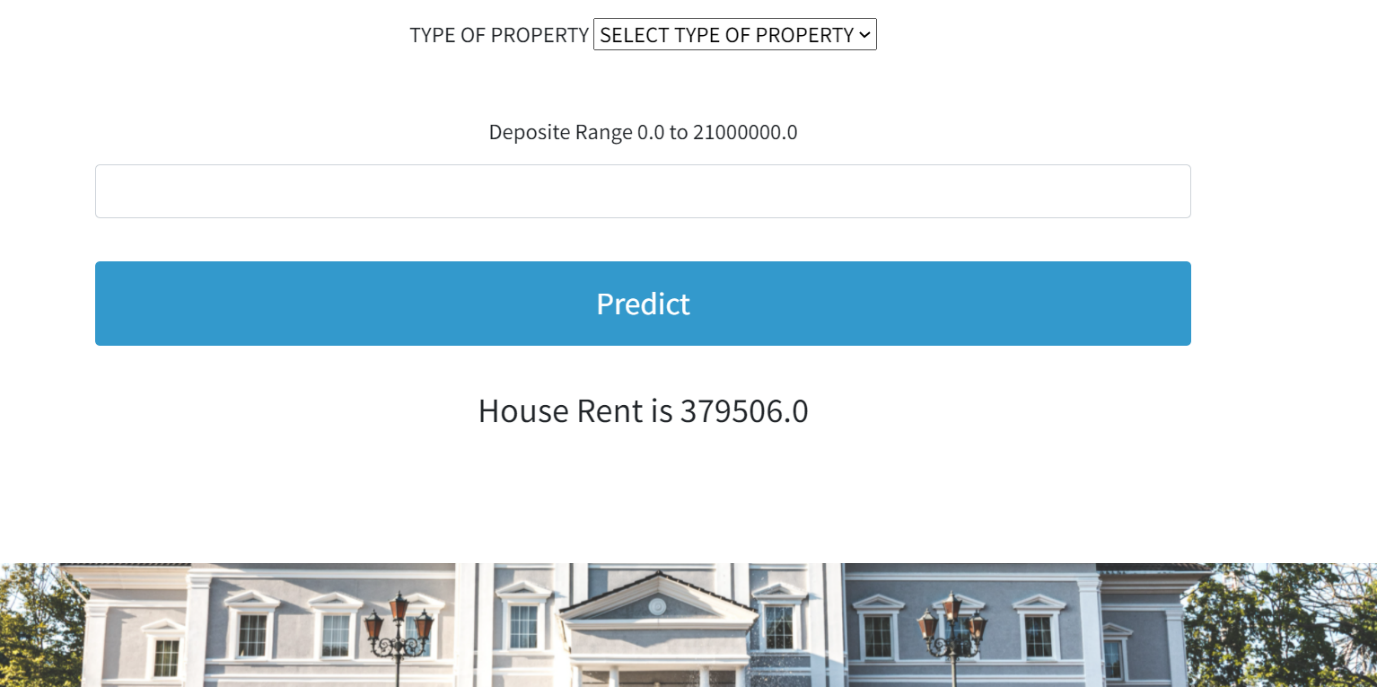
**The home.html page looks like**

****

**upload.html**

****

**Final Output**

****

**Activity 2: Main Python Script**

Let us build app.py flask file which is a web framework written in python for server-side scripting. Let’s see step by step procedure for building the backend application.

In order to develop web api with respect to our model, we basically use Flask framework which is written in python.

Line 1-10 we are importing necessary libraries like Flask to host our model request

Line 12 Initialise the Flask application

Line 13 loading the model using pickle

Line 15 Routes the api url

Line 16-20 rendering the template. This helps to redirect to home page. In this home page, we give our input and ask the model to predict

Line 22 we are taking the inputs from the form

Line 27-28 converting our inputs to float and storing it in a array

Line 32 predicting the values given by the user

Line 37 prediction output rendering to upload template

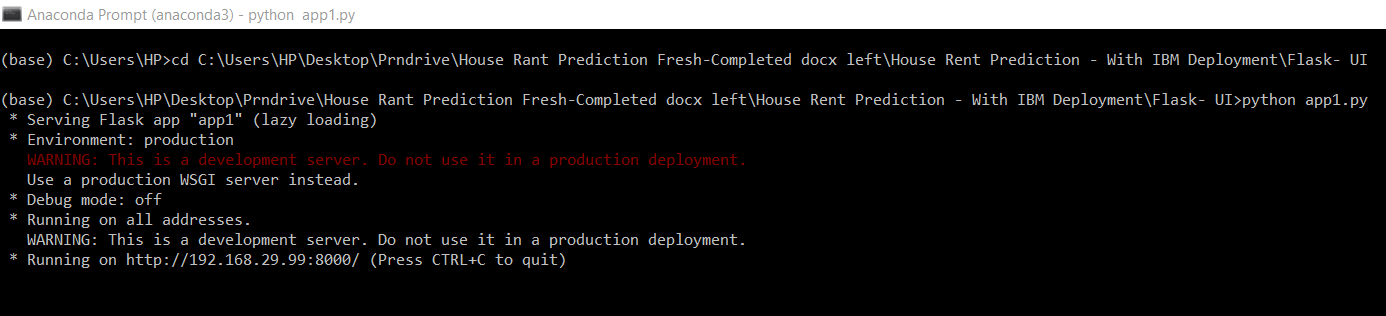
Line 31 the value of \_\_name\_\_ is set to \_\_main\_\_ when module run as main program otherwise it is set to name of the module

****

**Activity 3: Run the App**

* + Open anaconda prompt from the start menu
  + Navigate to the folder where your python script is.
  + Now type “python app.py” command

Navigate to the localhost where you can view your web page, Then it will run on local host:8000

****

**Activity 4:**

* Copy the http link and paste it in google link tab,it will display the form page
* Enter the values as per the form and click on predict buttion
* It will redirect to the page based on prediction output