HOUSE PRICE PREDICTION USING MACHINE LEARNING

Date of Submission:- 17 Sep 2022

This Photo by Unknown Author is licensed under CC BY-SA

Sr No.	Team Members	Github Link
1	Shreyas Gonjari	https://github.com/harshyad/house_pred
2	Dhanush Thokala	https://github.com/harshyad/house_pred
3	Harsh Yadav	https://github.com/harshyad/house_pred

1. Problem Statement:-

In India, there are multiple real estate classified websites where properties are listed for sell/buy/rent purposes such as 99acres, housing, common floor, magic bricks and more. However, on each of these websites, we can see a lot of inconsistencies in terms of pricing of an apartment and there are some cases when similar apartments are priced differently thus there is a lot of in-transparency. Sometimes the consumers may feel the pricing is not justified for a particular listed apartment but there is no way to confirm that either. Proper and justified prices of properties can bring a lot of transparency and trust back to the real estate industry, which is very important as for most consumers especially in India the transaction prices are quite high and addressing this issue will help both the customers and the real estate industry in the long run. Prices of real estate properties are indirectly linked to our economy. Despite this, we do not have accurate measures of housing prices based on the vast amount of data available.

This project aims to use machine learning techniques for predicting house prices. We propose to use machine learning and artificial intelligence techniques to develop an algorithm that can predict housing prices based on certain input features.

2. MARKET/CUSTOMER/BUSINESS NEED ASSESSMENT

1) MARKET NEED

In the real estate market, several agents predict the estimated prices of the houses they are selling and sometimes the estimated cost seems too high to the customers hence losing the customers of the company and sometimes the surrounding conditions of the house do not match with the price of the house which again causes the loss of customers.

As a result of which customer lost their trust in the real estate agents

Hence making the real estate market grow down.

But this problem can be solved using machine learning techniques

And thus, the lost trust can also be regained.

2) CUSTOMER NEED

House is a very emotional part of the person's life they feel attached to it so they want the best house at the price they are paying as they devote a big amount of their life earning so they want the best.

And due to errors in the estimation price of the agents and estate companies, this can be a very harsh experience for the customer.

Hence Machine learning techniques can be very helpful for increasing - customers over real estate companies.

3) BUSINESS NEED

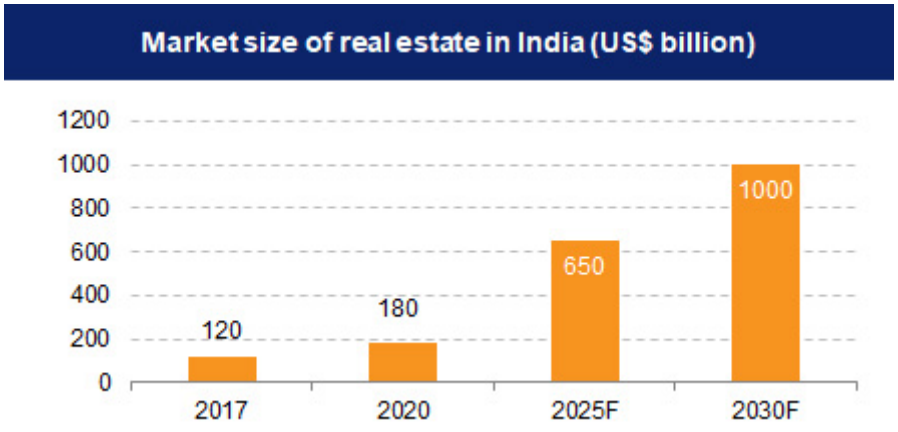
Using this technology, we will develop a website which will help customers and real estate companies by predicting the prices of the houses they want and for further on we will collaborate with the real estate companies so that they can use this website and we can charge a 10 or 20 per cent commission on the house price of each house sold by the use of the website and thus we can start earning money using our website.

And we can also start our own Real estate company using this website as a base and thus expand our business further by increasing the computational

power of the website and by increasing the area on which our website can predict the prices of houses thus making a multinational company.

3. TARGET SPECIFICATION AND CHARACTERIZATIONS

We will target the real estate companies in India as we know their market is increasing on a very high scale and we can also see that with the following graph: -



As we can see that the market size of real estate companies is going to be near about 1000 billion USD, hence this can be a good market to target.

And our main task would be to provide the best-estimated price range of the houses according to their location and locality.

And our proposed system will allow users to choose the location and set the attributes (e.g., No. of rooms, Sq. Ft, etc) which will then estimate the price range for that house and help the users decide whether to consider this house or not.

4. EXTERNAL SEARCH (information sources/references)

I HAVE USED THE BANGLORE HOUSES DATASET. THE DATASET CAN BE FOUND HERE: -
https://www.kaggle.com/datasets/saipavansaketh/pune-house-data

The dataset is named Pune-house-dataset but actually, it contains three cites datasets (Delhi, Bangalore, and Pune) I have used the Bangalore dataset for this project.

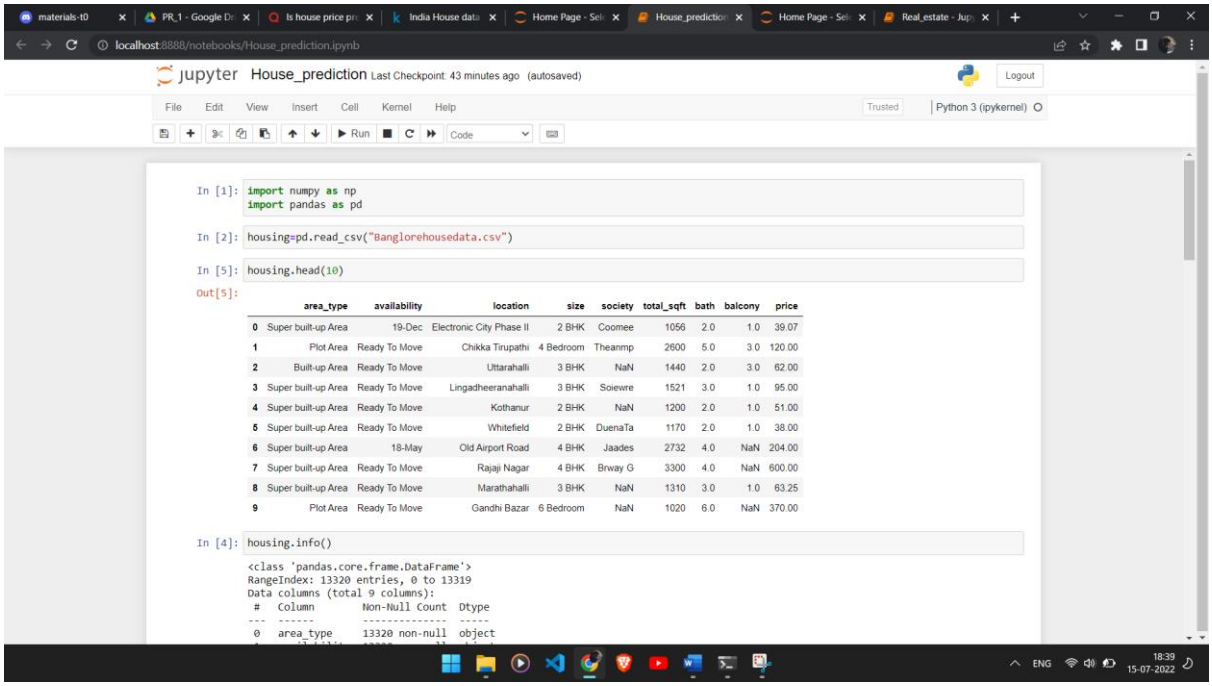
This dataset contains 9 features based on which the price of the houses can be predicted and it contains data of 13321 houses in Bangalore which will help us to evaluate the price of the houses in the Bangalore region.

I have read some articles from Quora: -
https://www.quora.com/Is-house-price-prediction-machine-learning-used-in-business

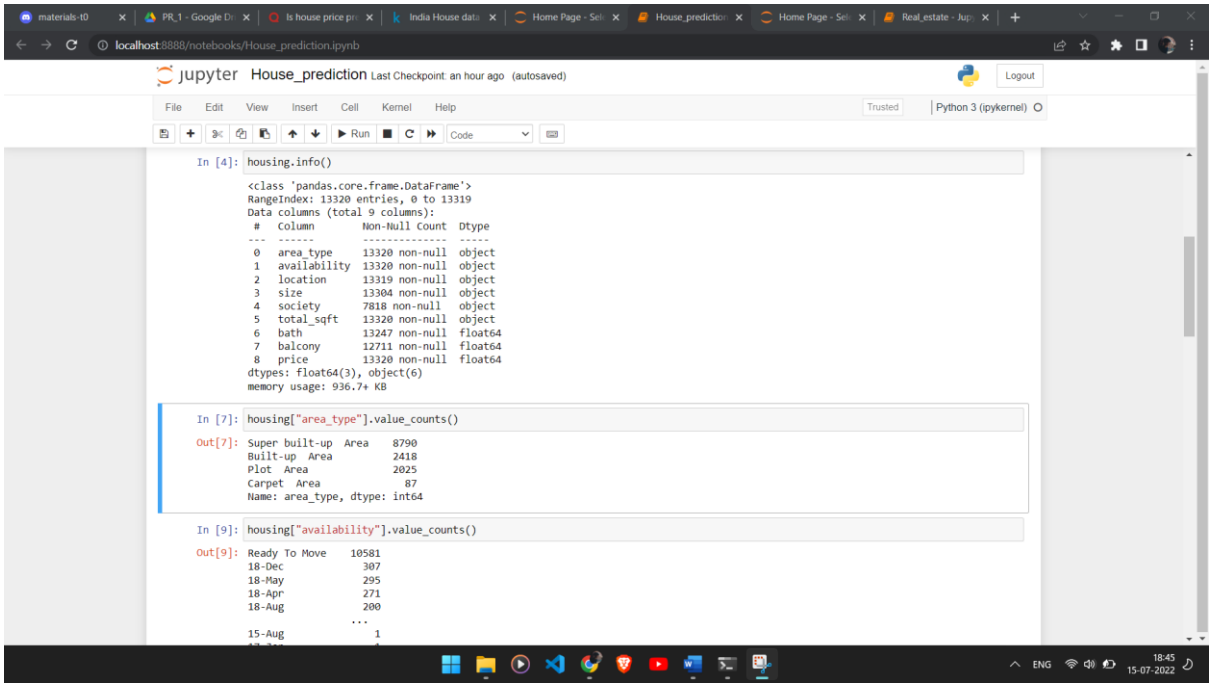
And I have written most of the content by myself and to understand the points I have taken help from the sample reports being provided and from our project mentor.

Here you can get the link to the license of the dataset by the open knowledge foundation: -
https://opendatacommons.org/licenses/dbcl/1-0/

Now let us have a view of our dataset: -



Some more information on our dataset: -



5. BENCHMARKING

Now let us see some correlation between the data points of our dataset.

Correlation between the datapoints

```
In [143]: housing.corr()
```

Out[143]:

	total_sqft	bath	price	BHK	price_per_sqft
total_sqft	1.000000	0.529650	0.583921	0.518814	0.206911
bath	0.529650	1.000000	0.527121	0.864710	0.334102
price	0.583921	0.527121	1.000000	0.480079	0.696377
BHK	0.518814	0.864710	0.480079	1.000000	0.298167
price_per_sqft	0.206911	0.334102	0.696377	0.298167	1.000000

As we can see that the **price** is highly correlated to the **total_sqft** and **price_per_sqft** features of the dataset.

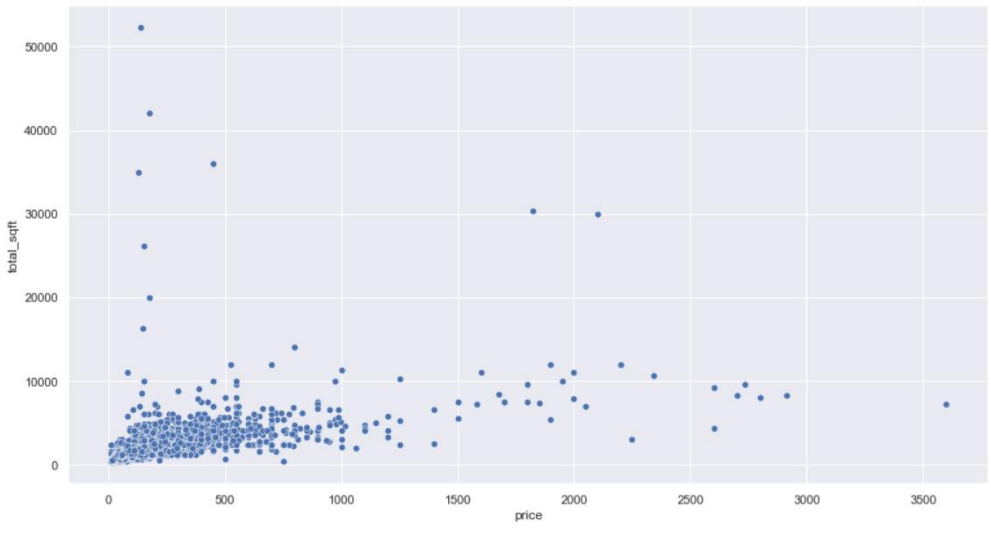
Hence these two features are very important for predicting the price of the house.

Now let us see the correlation heatmap of the features of dataset:-

```
In [146]: plt.figure(figsize = (15,10))
          heatmap=sns.heatmap(housing.corr(),cmap='RdYlGn',annot=True)
```

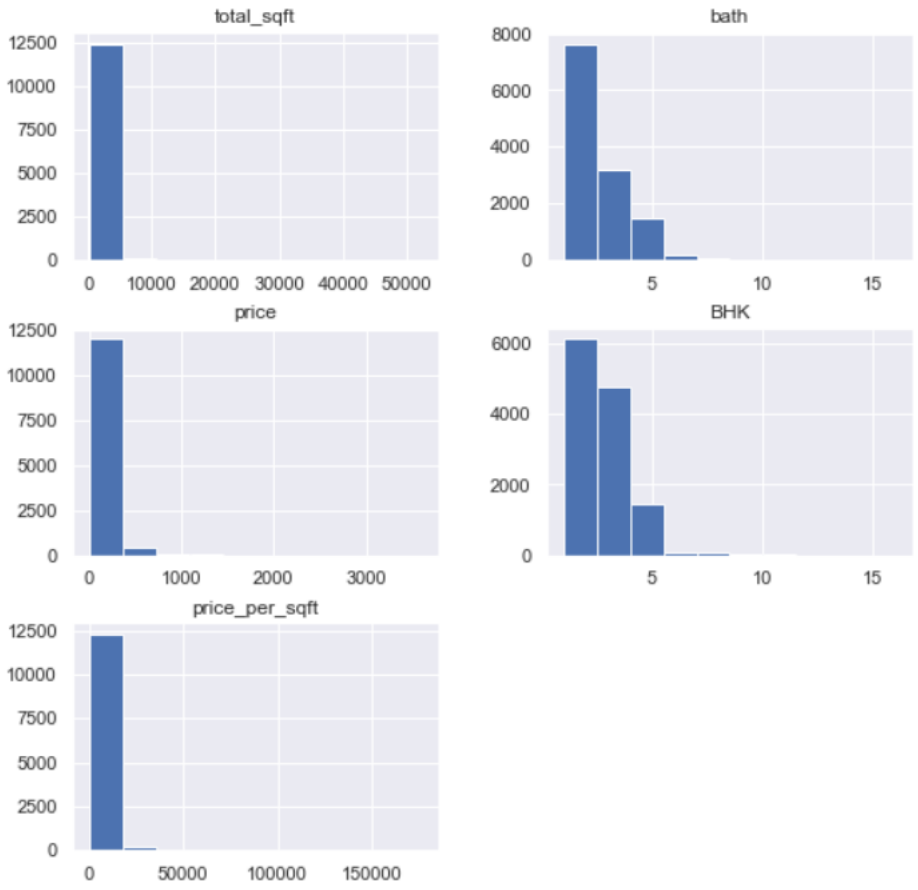
In this, above fig., we find the Correlation of all the columns. I use the matplotlib to resize the output of the image and using seaborn heatmap find a correlation between each of the columns

```
In [69]: plt.figure(figsize=(15,8))
         sns.set(style='darkgrid')
         sns.scatterplot(x=housing['price'], y=housing['total_sqft'])
Out[69]: <AxesSubplot:xlabel='price', ylabel='total_sqft'>
```



In the above figure we see insights of the dataset and get to know what the important prices related to the price of the house. We have seen the different houses in the different localities of Bangalore and their varying prices according to the area around them.

```
In [71]: housing.hist(figsize=(9,9))
         plt.show()
```



We will observe every feature of the dataset and will try to find the relation between each and every feature of the dataset.

7. APPLICABLE REGULATIONS

The patents mentioned above might claim the technology used if the algorithms are not developed and optimised individually and for our requirements.
Using a pre-existing model is off the table if it incurs a patent claim.

1. Must provide access to the 3rd party websites to audit and monitor the authenticity and behaviour of the service.
2. Enabling open-source, academic and research communities to audit the Algorithms and research the efficiency of the product.
3. Must be responsible for the scraped data: It is quite essential to protect the privacy and intention with which the data was extracted.

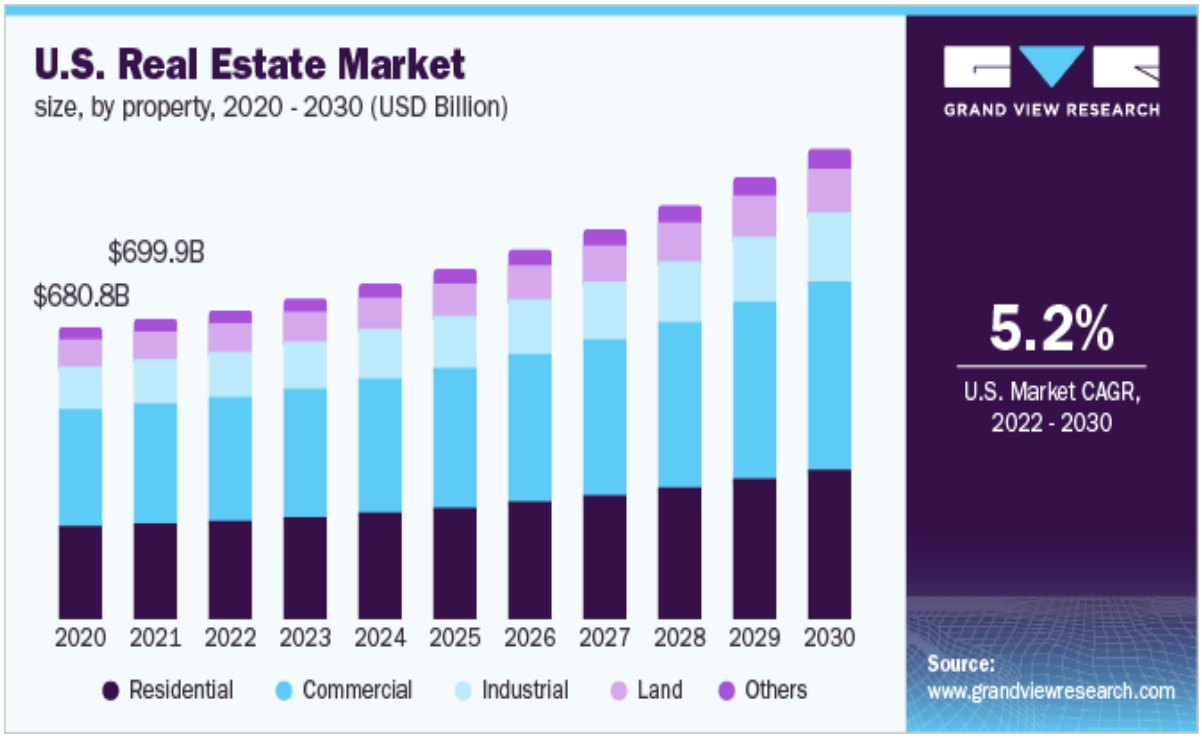
8. APPLICABLE CONSTRAINTS

- Continuous data collection and Maintainance
- Taking care of rarely bought products
- The use of a cloud platform to store the data gathered over the set
- Using the sci-kit learn library to clean and transform the data.
- Using the seaborn library for visualizing the data in the form of 2D and 3D plots.
- For modelling: Using the Supervised learning and the regression approach
- Using NumPy and Pandas library for the mathematical implementation of the data features.

9. BUSINESS OPPORTUNITYY

The businesss of real estate is a serious business and today, it has almost become a gold mine. Truth be told, many entrepreneurs have tapped into it and are making millions. People are always on the hook out to buy houses and buildings and are either looking to buy or rent.

As of 2019, there are 3.7 million square feet of commercial land used for the real estate business. Also, according to some reports, the real estate industry will be worth over $1 trillion by 2030. It is for this reason that this industry stands with a lot of business opportunities



As we can see from the above analysis that the real state market in the United States of America the market size of the country is going to increase by 5.2% CAGR U.S. (2022-2030).

10. CONCEPT GENERATION

This product requires the tool of machine learning models to be written from scratch to suit our needs. Tweaking these models for our use is daunting more than coding them up from scratch.

A well-trained model can either be repurposed or built. But building a model with the resources and data we have is dilatory but possible. The customer might want to spend the least amount of time giving input data.

This accuracy will take a little effort to nail because it's imprudent to rely purely on the Classic Machine Learning algorithm.

1. FIRST WE CLEAN THE DATA

```
In [8]: # Checking the null values
        housing.isnull().sum()
```

```
Out[8]: area_type          0
        availability       0
        location           1
        size              16
        society         5502
        total_sqft         0
        bath              73
        balcony          609
        price              0
        dtype: int64
```

```
In [9]: housing=housing.drop(['availability','area_type','society','balcony'], axis=1)
        housing.head()
```

Out[9]:

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

We check the null values of each feature in the dataset and we will remove the unwanted features of the dataset we will remove the null values by using the median strategy for the numeric values, and for the categorical value, s we will fill the most occurring value to fill the null values.

2. SPLIT THE DATA IN X, Y VARIABLE

In [55]:
```
X=housing.drop(['price'],axis=1)
Y=housing['price']
X.head()
```

Out[55]:

	location	total_sqft	bath	BHK
0	Electronic City Phase II	1056.0	2.0	2
2	Uttarahalli	1440.0	2.0	3
3	Lingadheeranahalli	1521.0	3.0	3
4	Kothanur	1200.0	2.0	2
6	Old Airport Road	2732.0	4.0	4

In [56]:
```
Y
```

Out[56]:
```
0        3.665355
2        4.127134
3        4.553877
4        3.931826
6        5.318120
           ...
13314    4.718499
13316    5.991465
13317    4.094345
13318    6.190315
13319    2.833213
Name: price, Length: 8694, dtype: float64
```

3. TRAIN_TEST_SPLIT DATA INTO X_TRAIN, X_TEST, Y_TRAIN, Y_TEST

In [57]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression,Lasso,Ridge
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import OneHotEncoder,StandardScaler
from sklearn.compose import make_column_transformer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score
```

In [58]:
```
X_train,X_test,y_train,y_test = train_test_split(X,Y,test_size=0.2,random_state=3)
print(X_train.shape)
print(X_test.shape)
```

```
(6955, 4)
(1739, 4)
```

WE WILL USE THE THREE DIFFERENT MODELS FOR THIS PROBLEM AND WE WILL FINALIZE THIS MODEL WHICH WILL GIVE THE GOOD ACCURACY.

1. LINEAR REGRESSION

Some of the functions for pipeline transformation

```
In [59]: column_trans=make_column_transformer((OneHotEncoder(sparse=False),['location']),
                                              remainder='passthrough')
```

```
In [60]: scaler=StandardScaler()
```

Applying the Linear Regression

```
In [61]: lr=LinearRegression()
```

```
In [62]: pipe=make_pipeline(column_trans,scaler,lr)
```

```
In [63]: pipe.fit(X_train,y_train)
```

```
Out[63]: Pipeline(steps=[('columntransformer',
                          ColumnTransformer(remainder='passthrough',
                                            transformers=[('onehotencoder',
                                                           OneHotEncoder(sparse=False),
                                                           ['location'])])),
                         ('standardscaler', StandardScaler()),
                         ('linearregression', LinearRegression())])
```

```
In [64]: y_pred_lr=pipe.predict(X_test)
```

```
In [65]: r2_score(y_test,y_pred_lr)
```

```
Out[65]: 0.7835904845877358
```

2. DECESION TREE REGRRESOR

Applying the Decision tree regressor

```
In [66]: dt=DecisionTreeRegressor()
```

```
In [67]: pipe=make_pipeline(column_trans,scaler,dt)
```

```
In [68]: pipe.fit(X_train,y_train)
```

```
Out[68]: Pipeline(steps=[('columntransformer',
                          ColumnTransformer(remainder='passthrough',
                                            transformers=[('onehotencoder',
                                                           OneHotEncoder(sparse=False),
                                                           ['location'])])),
                         ('standardscaler', StandardScaler()),
                         ('decisiontreeregressor', DecisionTreeRegressor())])
```

```
In [69]: y_pred_dt=pipe.predict(X_test)
```

```
In [70]: r2_score(y_test,y_pred_dt)
```

```
Out[70]: 0.7997477381083633
```

3. RANDOM FOREST REGRESSOR

Applying the Random Forest Regressor

```
In [71]: model = RandomForestRegressor()

In [72]: pipe=make_pipeline(column_trans,scaler,model)

In [73]: pipe.fit(X_train,y_train)

Out[73]: Pipeline(steps=[('columntransformer',
                          ColumnTransformer(remainder='passthrough',
                                            transformers=[('onehotencoder',
                                                           OneHotEncoder(sparse=False),
                                                           ['location'])])),
                         ('standardscaler', StandardScaler()),
                         ('randomforestregressor', RandomForestRegressor())])

In [74]: y_pred=pipe.predict(X_test)

In [75]: r2_score(y_test,y_pred)

Out[75]: 0.8413836068199805
```

NOW WE WILL ANALYZE ALL THE MODELS ACCORDING TO THEIR SCORE

1. **LINEAR REGRESSION:→ 0.7835904845877358**

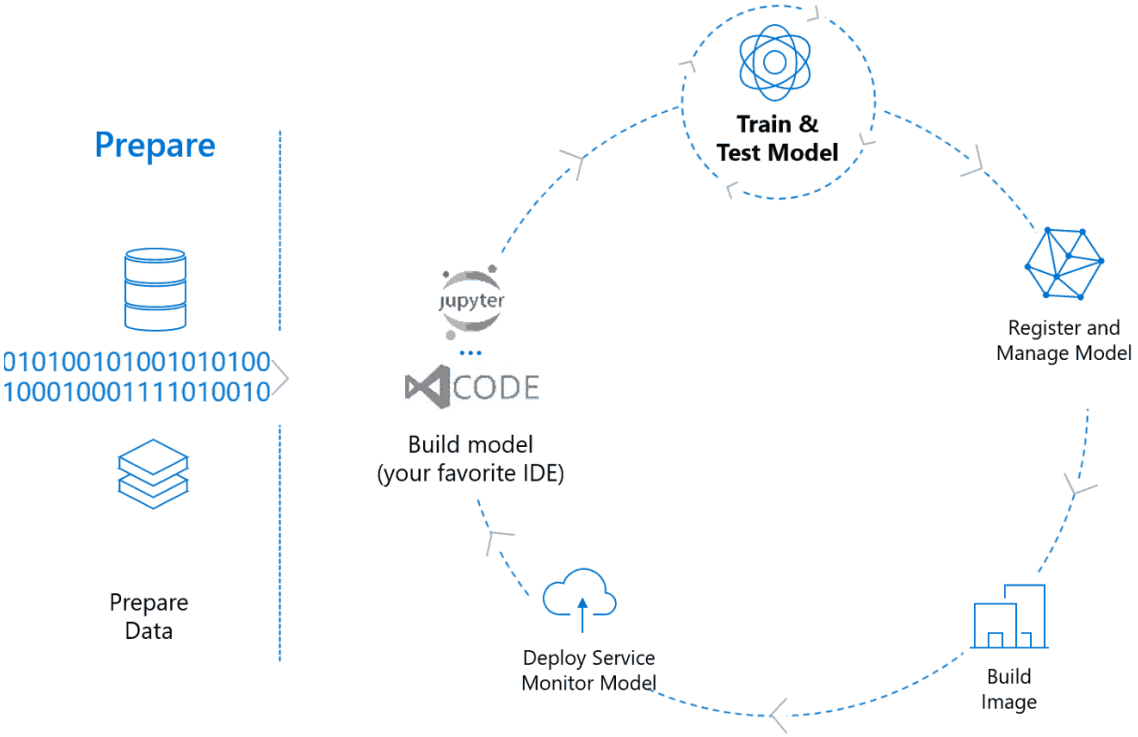
2. **DECDECISION TREE REGRESOR:→ 0.7997477381083633**

3. **RANDOM FOREST REGRESSOR:→ 0.8413836068199805**

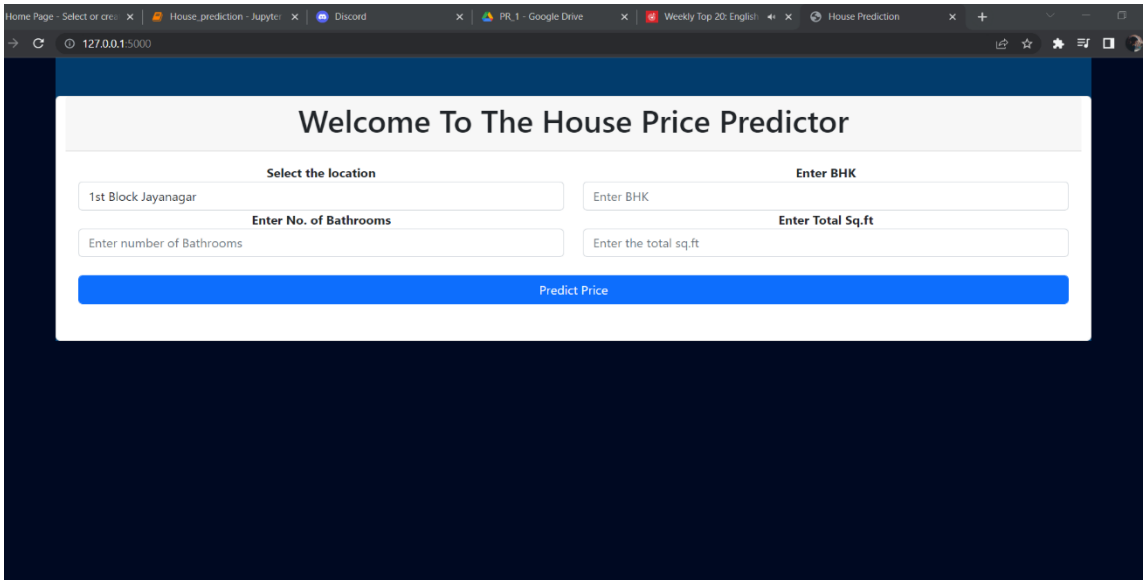
As we can see that random forest is providing us with the best R_2 score Hence, we will use a random forest algorithm for our model implementation.

11. CONCEPT DEVELOPMENT

The concept can be developed using the appropriate API (Flask in this case and using Django as a framework for the same and its development, The cloud services have to be chosen accordingly to the need.



I created a web-based service for predicting the house prices in the Bangalore Depending on the location, bhk, no. of bathrooms, total_sqft.



12. FINAL PRODUCT PROTOTYPE

The product takes the following functions to perfect and provide a good result

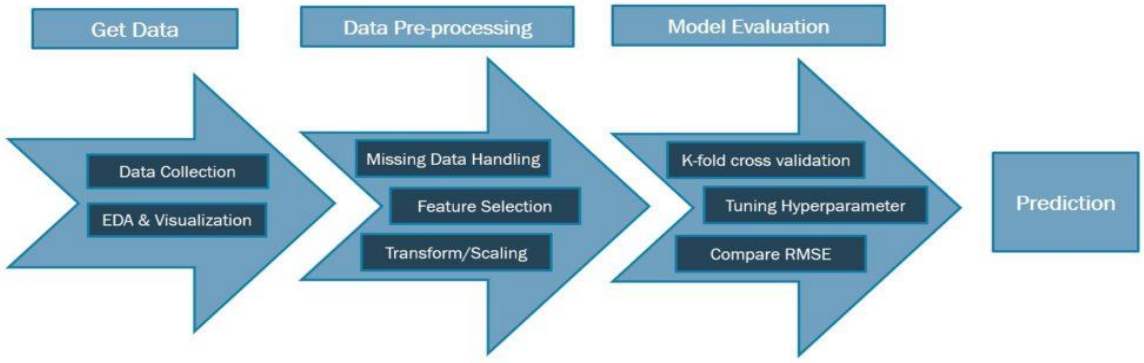
Back-end

Model or Webapp Development: This must be done before releasing the service. A lot of manual supervised machine learning must be performed to optimize the automated tasks

- Performing EDA to realize the dependent and independent features.
- Algorithm training and optimization must be done to minimize overfitting of the model and hyperparameter tuning

Front End

- Different user interfaces: The user must be given many options to choose from in terms of parameters. This can only be optimized after a lot of testing and analysis of all the edge cases.
- Interactive visualization of the data extracted from the trained models will return raw and inscrutable data. This must be present in an aesthetic and an “easy to read” style.
- Feedback system: A valuable feedback system must be developed to understand the customer’s needs that have not been met. This will help us train the models constantly.

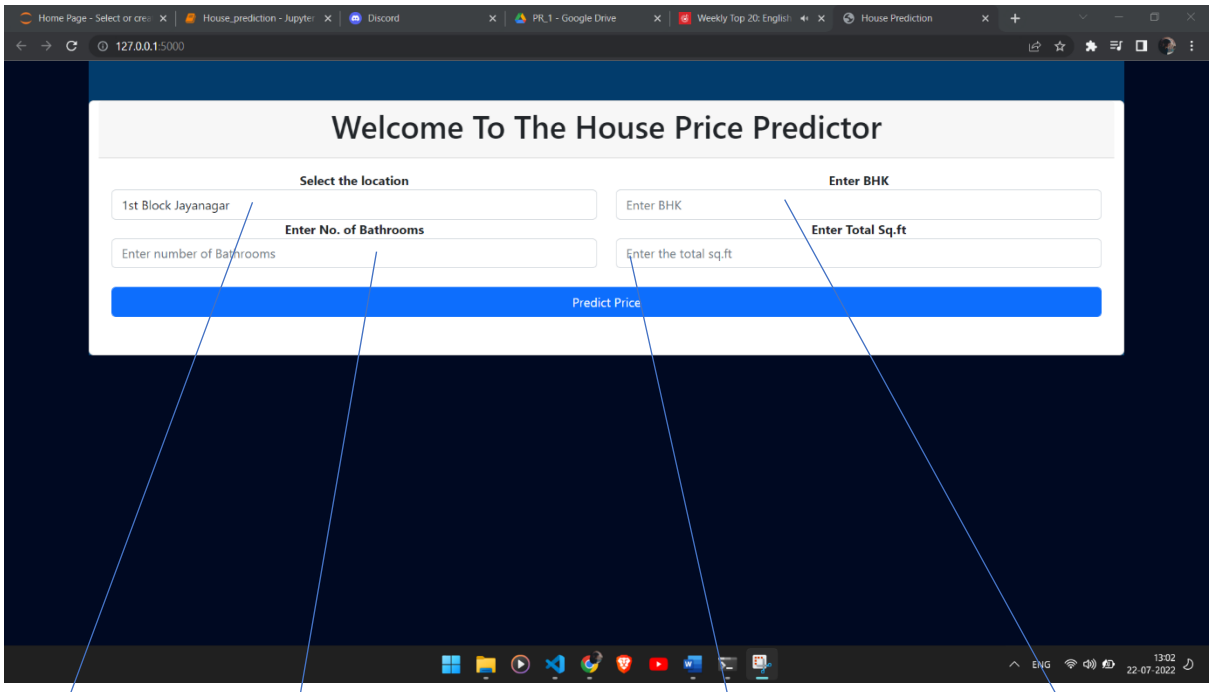


13. PRODUCT DETAILS

- HOW DOES IT WORK

I created a web app using the Flask framework and use a random forest regressor model for training the model on the training dataset and deploying this web app on the Heroku cloud platform.

Here's the view of our web product:→

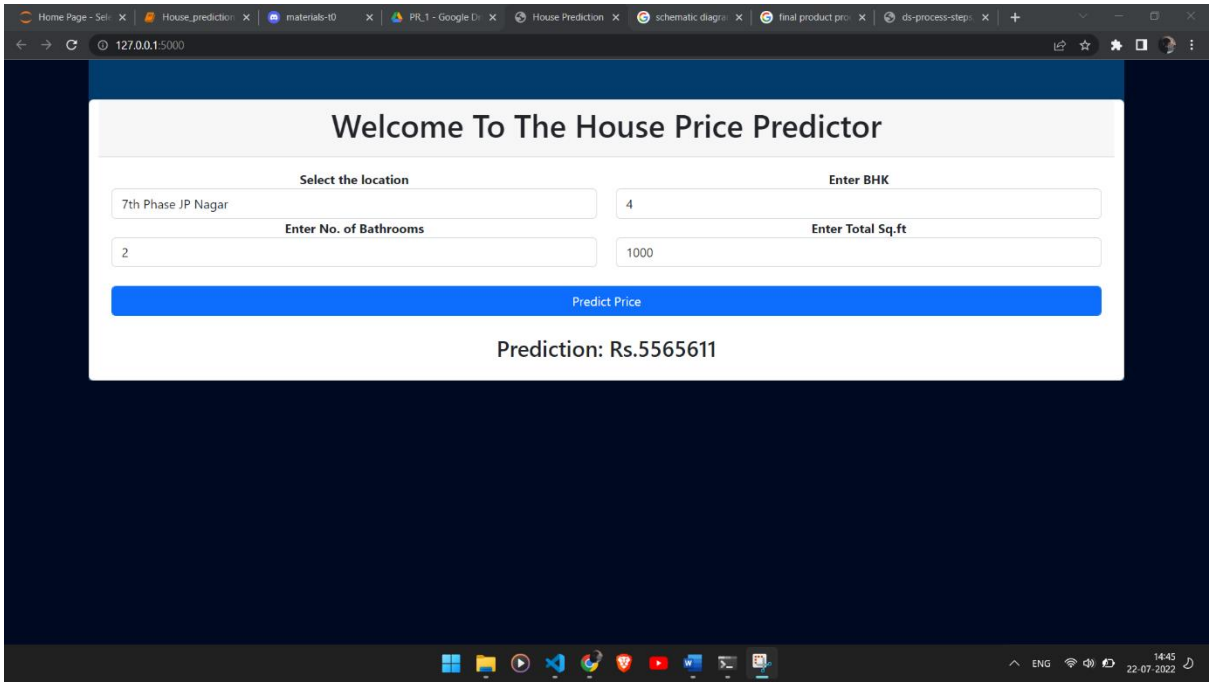


Step-1 **Step-2**

Step-3 **Step-4**

Step-1: Enter the location
Step-2: Enter the BHK
Step-3: Enter the No. of Bathrooms
Step-4: Enter the total square feet
Step-5: Click on the Predict price button to predict the price

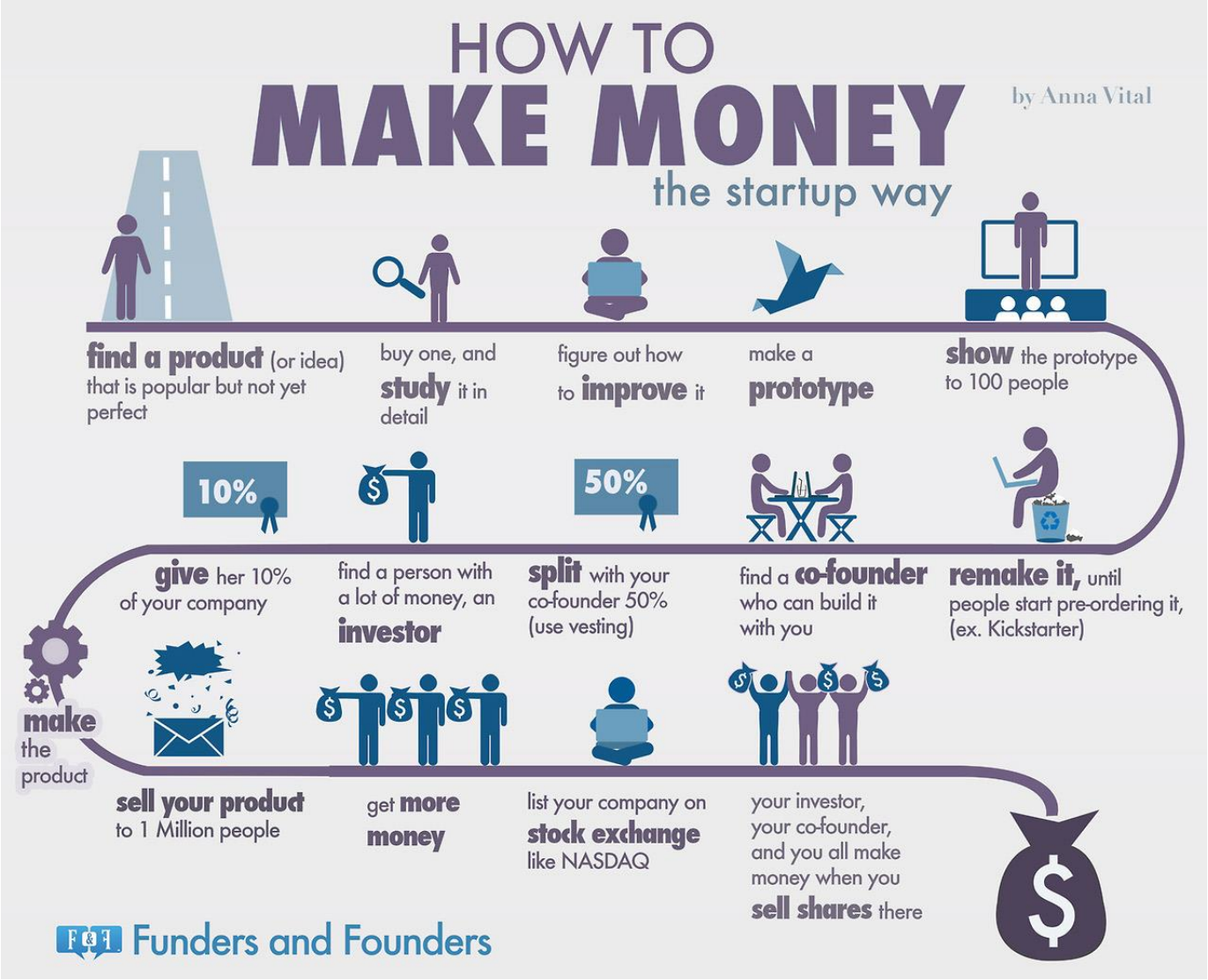
RESULT:



Here we can see that we enter:
Location: 7th phase Jp Nagar,
BHK: 4,
Enter no. of bathrooms: 2,
Enter Total Square feet: 1000,
And we got the price: Rs.55,65,611

14. Business Model

For this service, it is beneficial to use a Subscription Based Model, where initially some features will be provided for free to engage customer retention and increase our customer count. Later it will be charged a subscription fee to use the service further for their business. In the subscription business model, customers pay a fixed amount of money on fixed time intervals to get access to the product or service provided by the company. The major problem is user conversion; how to convert the users into paid users.



15. Financial Modelling

Let's assume that a team with 1 Machine learning engineers, 1 full stack developer, 1 android developer and 3 non-technical are required.

Profit=y
Percent of the house = 15%
Price of the house = x
Production and maintenance cost = 1 ml + 1 fs + 1 ad + 3 nt
Financial Equation will be as follows –

$$Y = (15\% \, x) - (1 \text{ ml} + 1 \text{ fs} + 1 \text{ ad} + 3 \text{ nt})$$

Here y is the profit.

16. Conclusion

While this project can be made more accurate by using more advanced machine learning techniques hence increasing the efficiency of the model.
There are many real estate companies in the real estate market but they have a huge margin of errors due to which they are losing customers with the help of machine learning algorithms we can reduce the margin of errors and thus increase the number of customers and more importantly making them believe in our company or website so that they can recommend it to further people which in turn increases the number of customers hence increasing the net profit.

So, our Aim is achieved as we have successfully ticked all our parameters as mentioned in our Aim Column. It is seen that circle rate is the most effective attribute in predicting the house price and that the Random Forest is the most effective model for our Dataset with
R2 score of 0.841383606199805