

PROJECT REPORT

on

REAL ESTATE PRICE PREDICTION

(CSE V SEMESTER MINI PROJECT)

2022-2023

By

Ranvir Singh Rawat

20011241



DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

CERTIFICATE

This is to certify that the project report entitled “Real Estate Price Prediction using ML” is a bonafide project work carried out by Ranvir Singh Rawat, roll no- 20011241, in partial fulfillment of award of degree of B-tech of Graphic Era Hill University, Dehradun during the academic year 2022-2023. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated. The project has been approved as it satisfies the academic requirements associated with the degree mentioned.

ACKNOWLEDGEMENT

Here by ,I am submitting the project report on “**Real Estate Price Prediction Using ML**” as per the scheme of Graphic Era Hill University, Dehradun.

I consider it mine cardinal duty to express the deepest sense of gratitude to **Mr. Anmol Sir** , Asst. Professor, Department of Computer Science and Application for the invaluable guidance extended at every stage and in every possible way.

Finally I am very much thankful to all the faculty members of the Department of Computer Science and Technology, friends and our parents for their constant encouragement, support and help throughout the period of project conduction.

Mr. Ranvir Singh Rawat

Roll No.- 2018624

CSE-F-V-Sem

Session: 2022-2023

GEHU, Dehradun

ABSTRACT

Real Estate prices fluctuate on a daily basis and are sometimes exaggerated rather than based on worth. I propose to implement a house price prediction model of Bangalore, India.

The major focus of this project is on predicting home prices using genuine factors. Here basic intend is to base an evaluation on every basic criterion that is taken into account when establishing the pricing.

It's a Machine Learning model which integrates Data Science and Web Development. The goal of this project is to learn Python and get experience in Data Analytics, Machine Learning, and AI. and also build a website UI using HTML, CSS and Javascript.

TABLE OF CONTENTS

1. INTRODUCTION
2. LITERATURE SURVEY
3. LANGUAGES AND LIBRARIES USED
4. METHODOLOGY
 - 4.1 DATA CLEANING
 - 4.2 FEATURE ENGINEERING
 - 4.3 OUTLIER REMOVAL
 - 4.4 MODEL BUILDING
5. MACHINE LEARNING MODEL USED
6. RESULT AND DISCUSSIONS
 - 6.1 BEST SUITED MODEL
 - 6.2 DEPLOYMENT APP
7. CONCLUSION
8. REFERENCES

INTRODUCTION

Project Aim

Create an effective price prediction model

Validate the model's prediction accuracy

Identify the important home price attributes which feed the model's predictive power

Need And Motivation

What are the things that a potential home buyer considers before purchasing a house? The location, the size of the property, vicinity to offices, schools, parks, restaurants, hospitals or the stereotypical white picket fence? What about the most important factor — the price?

Now with the lingering impact of demonetization, the enforcement of the Real Estate (Regulation and Development) Act (RERA), and the lack of trust in property developers in the city, housing units sold across India in 2017 dropped by 7 percent. In fact, the property prices in Bengaluru fell by almost 5 percent in the second half of 2017, said a study published by property consultancy Knight Frank.

Buying a home, especially in a city like Bengaluru, is a tricky choice. While the major factors are usually the same for all metros, there are others to be considered for the Silicon Valley of India. With its help millennial crowd, vibrant culture, great climate and a slew of job opportunities, it is difficult to ascertain the price of a house in Bengaluru.

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 our model, so he can reject that house .

LITERATURE SURVEY

Real Estate Property is not only a person's primary desire, but it also reflects a person's wealth and prestige in today's society. Real estate investment typically appears to be lucrative since property values do not drop in a choppy fashion. Changes in the value of the real estate will have an impact on many home investors, bankers, policymakers, and others. Real estate investing appears to be a tempting option for investors. As a result, anticipating the important estate price is an essential economic indicator. According to the 2011 census, the Asian country ranks second in the world in terms of the number of households, with a total of 24.67 crores. However, previous recessions have demonstrated that real estate costs cannot be seen. The expenses of significant estate property are linked to the state's economic situation. Regardless, we don't have accurate standardized approaches to live the significant estate property values.

House price prediction is a vast topic, which is implemented through a variety of Computer Science Methods. Like Machine Learning, Linear Regression, Decision Tree, Deep Learning, Fuzzy Logic, ANFIS (Adaptive-Neuro Fuzzy Inference System), and Linear performance pricing.

In proposed model of Machine Learning, the dataset is divided into two parts: Training and Testing. 80% of data is used for training purpose and 20% used for testing purpose. The training set include target variable. The model is trained by using various machine learning algorithms, out of which Random forest regressions predict better results. For implementing the Algorithms, they have used Python Libraries NumPy and Pandas.

In another paper based on Machine Learning has used the multivariate linear regression model to perform the prediction. Also, it is compared with other Machine Learning models like Lasso, LassoCV, Ridge, RidgeCV and decision tree regressor. Multivariate linear regression performs the best with 84.5% accuracy.

LANGUAGES AND LIBRARIES USED

PYTHON

It is an advanced programming language intended to be simple to understand and implement. It is open-source software, which implies that it may be used for free, even in commercial applications. Python is available for Mac, Windows, and Unix computers, and it has also been adapted to Java and .NET virtual machines. It is also supported by various 2D and 3D imaging tools, allowing users to utilize Python to construct custom plug-ins and extensions.

LIBRARIES USED :-

NumPY- used for linear algebra , matrices ,etc.

PANDAS - used for creating data frame

MATPLOTLIB - used for data visualization

SKLEARN - used for model building

FOR CREATING WEBSITE UI:-

HTML - used to structure a web page and its content

CSS - used to style and layout web pages

JAVASCRIPT - used for programming the webpage

PYTHON FLASK SERVER - A built-in server capable of handling requests from http on one or more configured websites

IDE USED:-

JUPYTER NOTEBOOK

PYCHARM

VS CODE

METHODOLOGY

The processed data with the highest accuracy will be picked for house price estimation, and the forecast process on the web application will be monitored. Python will execute the flask server in Microsoft Visual Studio Code, and a local server will be enabled to evaluate home prices in a particular location.

SELECTING DATASET

The data is the most crucial part of a machine-learning project and should be carefully considered. Here I have taken Bangalore House Price DataSet from kaggle.com.

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

```
df1.shape
```

```
(13320, 9)
```

The above figure includes data on 13320 instances and 9 characteristics.

DATA CLEANING

It is a process of removing irrelevant values like area_type , society ,null values and fixing some independent variables like total_sqft.

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

```
(13320, 5)
```

```
df2.isnull().sum()
```

```
location      1
size          16
total_sqft     0
bath          73
price          0
dtype: int64
```

FEATURE ENGINEERING

Feature engineering is the art of formulating useful features from existing data following the target to be learned and the machine learning model used.

It involves transforming data to forms that better relate to the underlying target to be learned. When done right, feature engineering can augment the value of your existing data and improve the performance of your machine learning models. It is helpful for outlier detection and removal.

For example ,we have to create a new variable for model building i.e price_per_sqft.

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

Below you can see that a new column is being generated for price_per_sqft.

	location	size	total_sqft	bath	price	bhk	price_per_sqft
0	Electronic City Phase II	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	Chikka Tirupathi	4 Bedroom	2600.0	5.0	120.00	4	4615.384615
2	Uttarahalli	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Lingadheeranahalli	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	Kothanur	2 BHK	1200.0	2.0	51.00	2	4250.000000

OUTLIER DETECTION AND REMOVAL

Outlier is an unusual occurrence in the input data that causes a machine learning model to provide false results, which is overfitting. They represent the extreme variations of data values.

I have removed some outliers by calculating upper boundary and lower boundary by taking 1 standard deviation from the mean of the values. As 68% of the data falls within 1 standard deviation and mean. For example in price_per_sqft,

```
df6.shape
```

```
(12456, 7)
```

```
df6.price_per_sqft.describe()
```

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

Before there were around 12502 data points .

```
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
```

```
(10242, 7)
```

After Removing outliers now we have 10242 data points.

The figure consists of two side-by-side scatter plots, both titled "Electronic City". Both plots have "Price (Indian Lakh Rupees)" on the y-axis and "Total Square Feet Area" on the x-axis. The y-axis ranges from 20 to 110, and the x-axis ranges from 750 to 2000.

Legend for both plots:

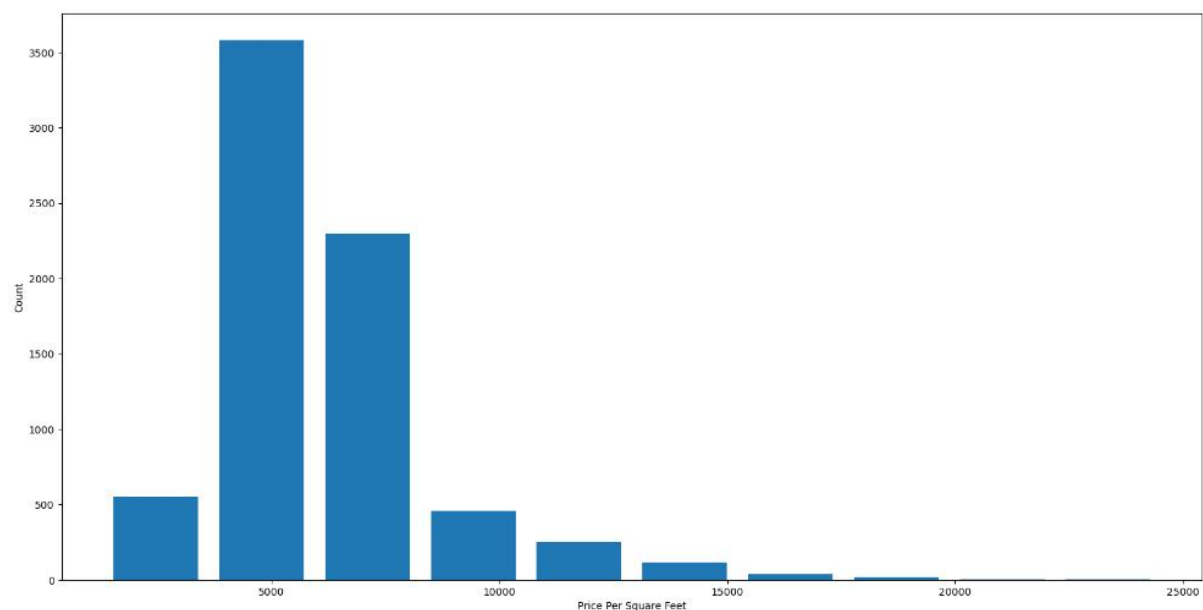
- 2 BHK (Blue dots)
- 3 BHK (Green plus signs)

The left plot shows a large red ellipse highlighting a cluster of 2 BHK houses. This cluster is centered around a price of 50-60 Lakhs and a total square feet area of 1000-1250.

The right plot is a zoomed-in view of the 2 BHK cluster highlighted in the left plot. It shows individual data points more clearly, with prices ranging from approximately 40 to 90 Lakhs and total square feet area ranging from 800 to 1400.

Electronic City Outliers

Now I have plotted a histogram in which I can see that Number of property per sqft area. After running that function we can see that the dataset has a normal distribution



As a machine learning model cannot interpret text data so we have convert it into numeric column And one of the way for conversion is one hot encoding also known as dummies so I have used pandas dummies method.

	location	total_sqft	bath	price	bhk
0	1st Block Jayanagar	2850.0	4.0	428.0	4
1	1st Block Jayanagar	1630.0	3.0	194.0	3
2	1st Block Jayanagar	1875.0	2.0	235.0	3

```
get dummies()
```

[illegible]

MODEL BUILDING

After dataset has been cleaned now we can move to training and testing. So basically we divide the dataset into two dataset first for model training and second for evaluating the model performance by testing.

Now taking our X and Y variable from the dataset for train and testing.

X will contain all the independent variables like total_sqft , bath , etc and Y will have the dependent variable i.e price.

Training set consists of 80% of the dataset and the testing set has 20% of the dataset.

Now import train_test_split from sklearn for testing and training.

```
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)
```

After that we will be using different machine learning models for testing and training.

For that we use a method called gridsearchCV provided by sklearn which can run any model on different regressor and different parameters.

GridsearchCv not only provides the best score among different algorithms but also it tells the best parameters in them.

```
from sklearn.model_selection import GridSearchCV

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

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

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

find_best_model_using_gridsearchcv(X,y)
```

In the above image you can see that I am using 3 different algorithm which are linear regression, lasso and decision tree regressor.

MACHINE LEARNING MODELS USED

Linear Regression

Linear regression aims to predict the relationship between two variables by fitting a equation to observable data. One variable is considered an explanatory variable, while the other is considered a dependent variable. A modeler, for example, could wish to apply a linear regression model to match people's pounds to their measurements.

Lasso

The LASSO approach regularizes model parameters by decreasing part of the regression coefficients to zero. Following the shrinkage, the feature selection phase follows, in which every non-zero value is chosen to be incorporated into the model. This strategy helps reduce prediction mistakes that are typical in statistical models.

DecisionTree Regressor

Decision tree learning applies a divide-and-conquer technique by undertaking a greedy search to determine the best split points inside a tree. This dividing procedure is then continued in a top-down, recursive way until all or the majority of records have been categorized under particular class labels. The decision tree's complexity determines whether or not all data points are classified as homogeneous sets.

RESULT AND DISCUSSIONS

BEST SUITED MODEL

According to the results, the Linear Regression Model obtained the most remarkable accuracy, whereas the other two algorithms produced comparable accuracy lower than the higher-attaining accuracy.

Finally, a linear regression model will be implemented in the web application to supervise the estimating process and calculate the price of the building in that specific location. Linear Regression has achieved an accuracy of 0.847796 while lasso and decision_tree are having score 0.726738 and 0.726238 respectively.

	model	best_score	best_parameter
0	linear_regression	0.847796	{'normalize': False}
1	lasso	0.726738	{'alpha': 2, 'selection': 'cyclic'}
2	decision_tree	0.726238	{'criterion': 'friedman_mse', 'splitter': 'ran...

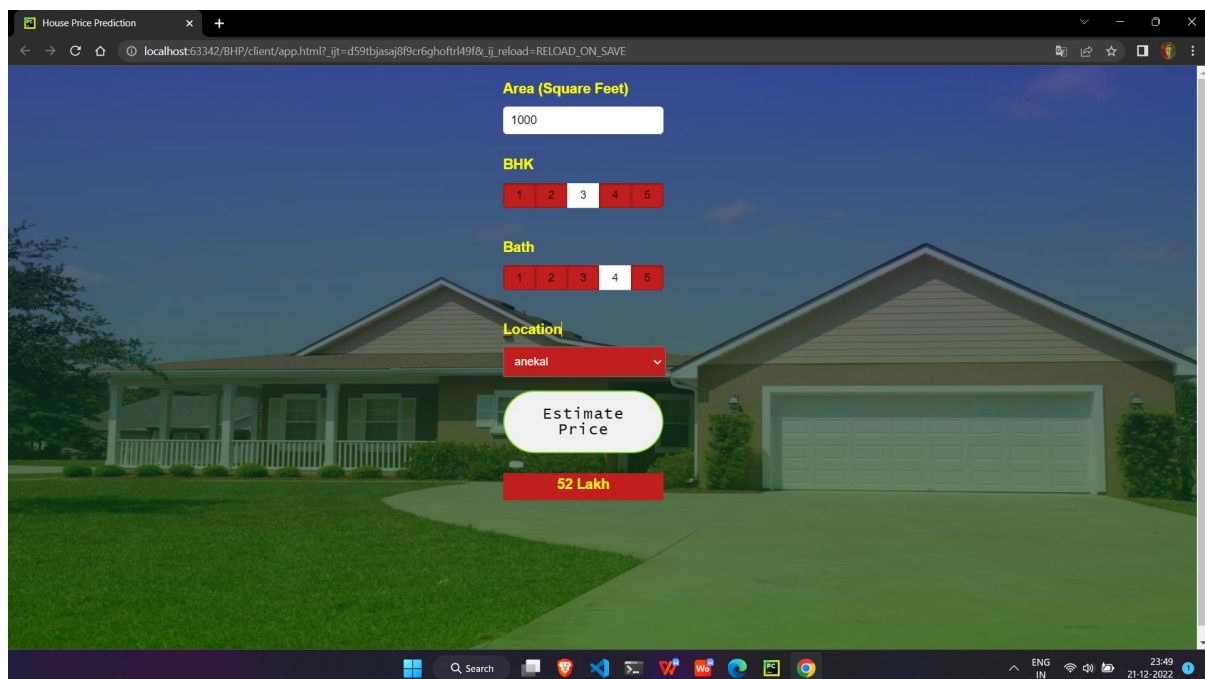
DEPLOYMENT APP

Now, we need to export our model as a pickle file , which transforms Python objects into a character stream. Also, in order to interact with the locations(columns) from the frontend, we must export them into a JSON (columns.json) file.

The Model is deployed through Python Web App Flask which can serve the https requests made from the UI .

I have deployed this website into the local server.

The front end is built up of straightforward HTML. To receive an estimated pricing, the user may fill-up the form with the number of square feet, BHK, bathrooms, and location and click the 'ESTIMATE PRICE' button.



The screenshot shows a web browser window titled "House Price Prediction" with a URL of `localhost:63342/BHP/client/app.html?_ijt=d591bjasa8f9c6ghoftr49f8&_ij_reload=RELOAD_ON_SAVE`. The background of the form is a house image. The form fields are:

- Area (Square Feet)**: A text input field containing "1000".
- BHK**: A row of five radio buttons, with the third button (value 3) selected.
- Bath**: A row of five radio buttons, with the fourth button (value 4) selected.
- Location**: A dropdown menu showing "anekal".
- Estimate Price**: A button with a green border.
- 52 Lakh**: A red button displayed below the "Estimate Price" button, showing the predicted price.

The Windows taskbar at the bottom shows the time as 23:49 on 21-12-2022.

Web Application Interface

CONCLUSION

So my aim is achieved as I have successfully ticked all the parameters as mentioned in the aim section.

With several characteristics, the suggested method predicts the property price in Bangalore. I experimented with different Machine Learning algorithms to get the best model. When compared to all other algorithms, the Decision Tree Algorithm achieved the lowest loss and the greatest R-squared. Flask was used to create the website.

Let's see how the project pans out. Open the HTML web page we generated and run the app.py file in the backend. Input the property's square footage, the number of bathrooms, and the location, then click 'ESTIMATE PRICE.' We forecasted the cost of what may be someone's ideal home.

REFERENCES

1. Dataset: <https://www.kaggle.com/datasets/amitabhajoy/bengaluru-house-price-data>
2. Repository: "Web Application" <https://github.com/Amey-Thakur/BANGALOREHOUSE-PRICE-PREDICTION>
3. Pickle "Documentation"
4. A. Varma, A. Sarma, S. Doshi and R. Nair, "House Price Prediction Using Machine Learning and Neural Networks," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 1936-1939, doi: 10.1109/ICICCT.2018.8473231.
5. Furia, Palak, and Anand Khandare. "Real Estate Price Prediction Using Machine Learning Algorithm." eConference on Data Science and Intelligent Computing. 2020.
6. Musciano, Chuck, and Bill Kennedy. HTML & XHTML: The Definitive Guide: The Definitive Guide. " O'Reilly Media, Inc.", 2002.
7. Aggarwal, Shalabh. Flask framework cookbook. Packt Publishing Ltd, 2014.
8. Grinberg, Miguel. Flask web development: developing web applications with python. " O'Reilly Media, Inc.", 2018.