



# VIT<sup>®</sup>

**Vellore Institute of Technology**

(Deemed to be University under section 3 of UGC Act, 1956)

**ITA5007-DATA MINING AND BUSINESS INTELLIGENCE**

**TITLE- REAL ESTATE PRICE PREDICTION**

**SUBMITTED TO**  
**Prof. Jagadeesan S, SITE**

**TEAM MEMBERS:**

ADARSH KUMAR (22MCA0081)

ASHUTOSH KUMAR (22MCA0349)

NEHA KUMARI (22MCA0389)

Vellore Institute of Technology

School of Information Technology & Engineering (SITE)

## **ABSTRACT:-**

This project aims to develop a real estate price prediction model using machine learning techniques. The model will be trained on a dataset of historical real estate transaction records, including various property features such as location, square footage, number of bedrooms and bathrooms, and other relevant attributes. The objective is to build a model that can accurately predict the sale price of a given property based on its characteristics. This project has potential applications for real estate investors, agents, and anyone interested in buying or selling property. The accuracy of the model will be evaluated using various performance metrics such as mean squared error and R-squared. The results of this project could provide insights into the factors that affect real estate prices and help individuals make more informed decisions in the real estate market.

## **INTRODUCTION:**

The real estate market is a complex and dynamic industry that can be influenced by numerous factors, including economic conditions, supply and demand, location, property characteristics, and more. As a result, predicting real estate prices can be a challenging task. However, with the advent of machine learning and artificial intelligence techniques, it has become possible to develop predictive models that can help to estimate property prices accurately.

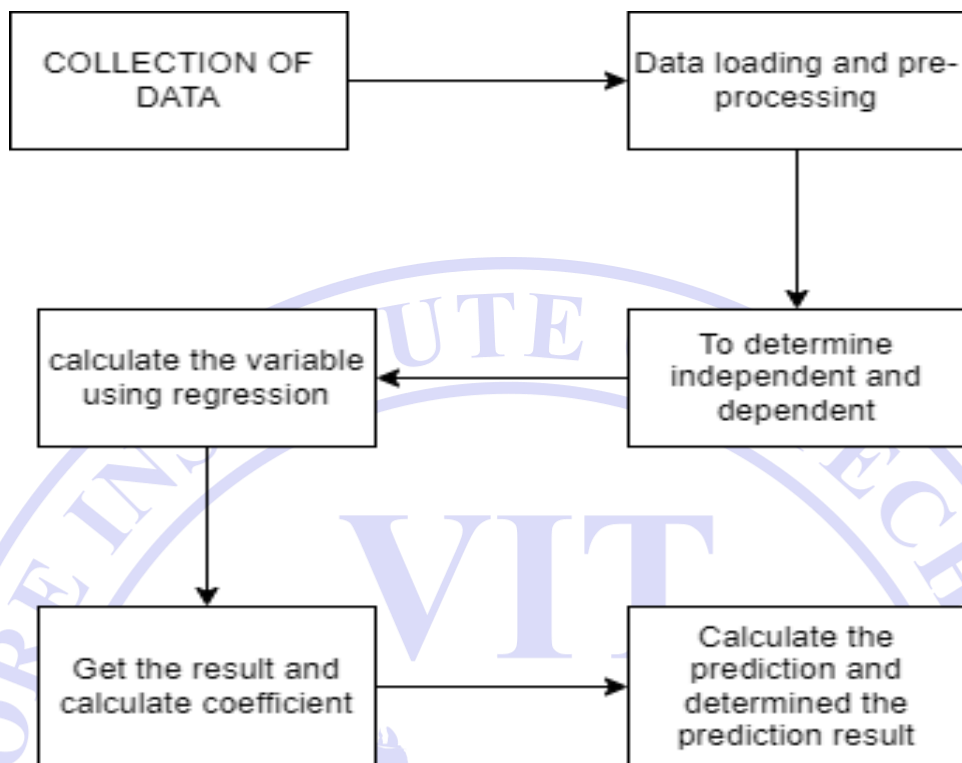
In this project, we aim to develop a real estate price prediction model using machine learning algorithms. The model will be trained on a dataset of historical real estate transaction records, which will include various property features such as location, square footage, number of bedrooms and bathrooms, and other relevant attributes. By analyzing the data and building a predictive model, we hope to create a tool that can accurately predict the sale price of a given property based on its characteristics.

The objective of this project is to provide insights into the factors that affect real estate prices and help individuals make more informed decisions in the real estate market. The potential applications of this model are vast and could be of great benefit to real estate investors, agents, and anyone interested in buying or selling property. By accurately predicting property prices, individuals can make better investment decisions, negotiate more effectively, and ultimately achieve their real estate goals more efficiently.

In the following sections, we will provide a detailed methodology of how we plan to develop this real estate price prediction model, including the data sources, feature selection, model selection, and performance evaluation. Ultimately, we hope to develop a reliable and accurate tool that can help to predict real estate prices and provide valuable insights into the real estate market.

உழைப்பே உயர்வு தரும்

## ARCHITECTURE DIAGRAM



### Data Mining Functionalities focused & Software Platform used: -

- Regression
- Data Characterization
- Data Discrimination
- Prediction
- Outlier Analysis

Google Colaboratory (in which we used PYTHON-3)

### Library used in python:-

- Matplot
- Numpy
- Pandas
- Sklearn

### Data Set used & its Description:-

- We have taken 'Bengaluru\_House\_Data' from kaggle to perform our prediction task from kaggle. (<https://www.kaggle.com/code/mohaiminul101/bengaluru-house-price-prediction/data>)
- Dataset is having 9 attributes which are:-
  - area\_type

- Availability
- Location
- size
- society
- total\_sqft
- bat,balcony
- price

## IMPLEMENTATION APPROACH

1. **Importing Required Libraries:** This step involves importing the necessary Python libraries such as pandas, numpy, matplotlib, and sklearn. These libraries provide various functionalities for data manipulation, analysis, visualization, and machine learning.

```
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
%matplotlib inline
import matplotlib
matplotlib.rcParams["figure.figsize"] = (20,10)
```

2. **Loading the Dataset:** The dataset is loaded using the `pd.read_csv` function from the pandas library. The dataset contains information about property prices and their features like location, total square feet area, number of bedrooms, number of bathrooms, etc.

```
[ ] from google.colab import files
    uploaded = files.upload()
```

Choose Files Bengaluru\_...se\_Data.csv

• Bengaluru\_House\_Data.csv(text/csv) - 938020 bytes, last modified: 9/29/2019 - 100% done  
Saving Bengaluru\_House\_Data.csv to Bengaluru\_House\_Data.csv

```
[ ] import pandas as pd
    import io
    df1 = pd.read_csv(io.BytesIO(uploaded['Bengaluru_House_Data.csv']))
    df1.head()
```

	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

3. **Data Cleaning and Preprocessing:** Data cleaning and preprocessing steps are performed to handle missing values, remove irrelevant columns, and convert categorical variables into numerical representations. In this code snippet, the 'size' column is processed to extract the number of bedrooms as an integer value.

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

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

```
[ ] df2.shape
```

```
(13320, 5)
```

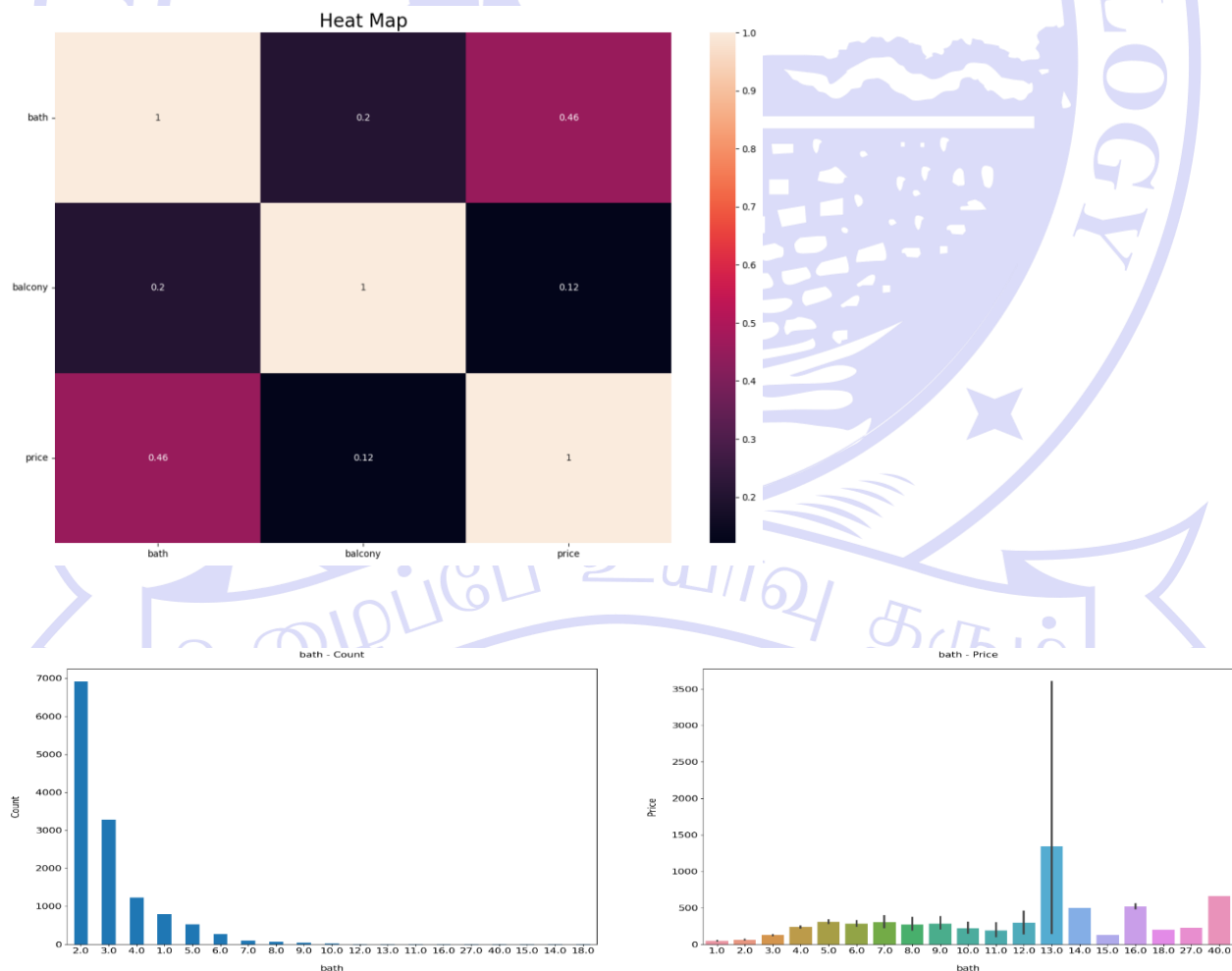
```
[ ] df3 = df2.dropna()  
df3.isnull().sum()
```

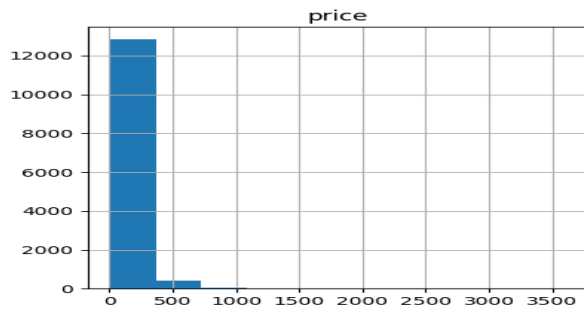
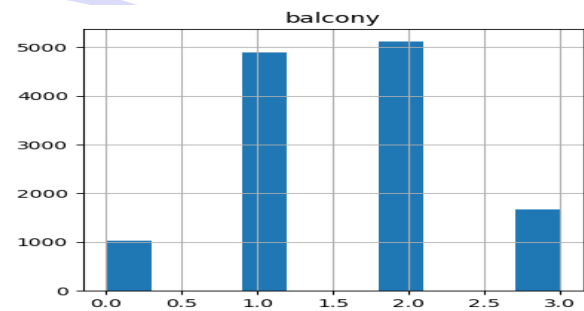
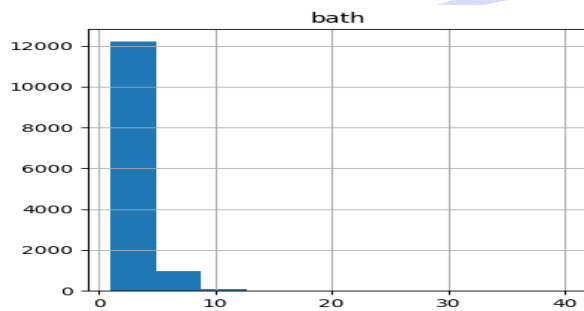
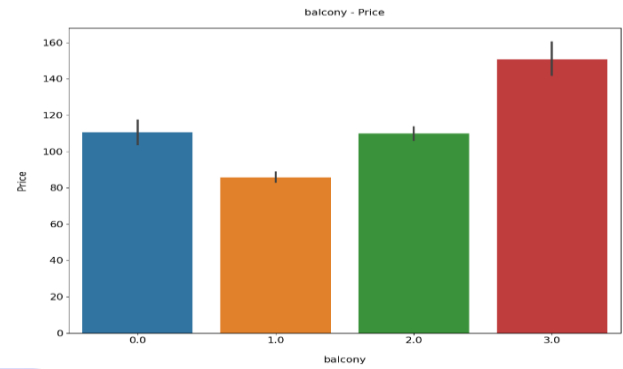
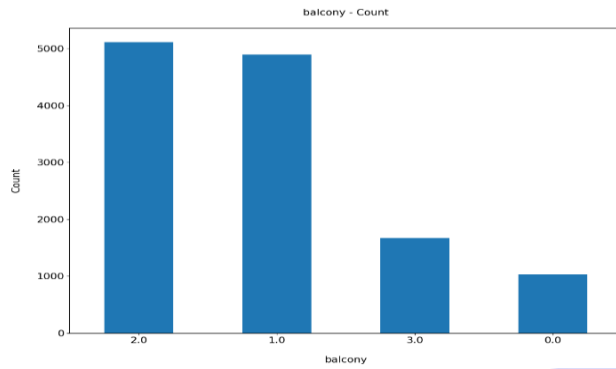
```
location    0  
size        0  
total_sqft  0  
bath        0  
price       0  
dtype: int64
```

```
[ ] df3.shape
```

```
(13246, 5)
```

4. **Data Visualization:** Data visualization techniques are used to gain insights into the dataset. This includes plotting histograms, scatter plots, and other visualizations to understand the relationships between variables.





5. **Feature Engineering:** Feature engineering involves creating new features or transforming existing ones to improve the model's performance. In this code snippet, a new feature 'price\_per\_sqft' is created by dividing the price by the total square feet area.

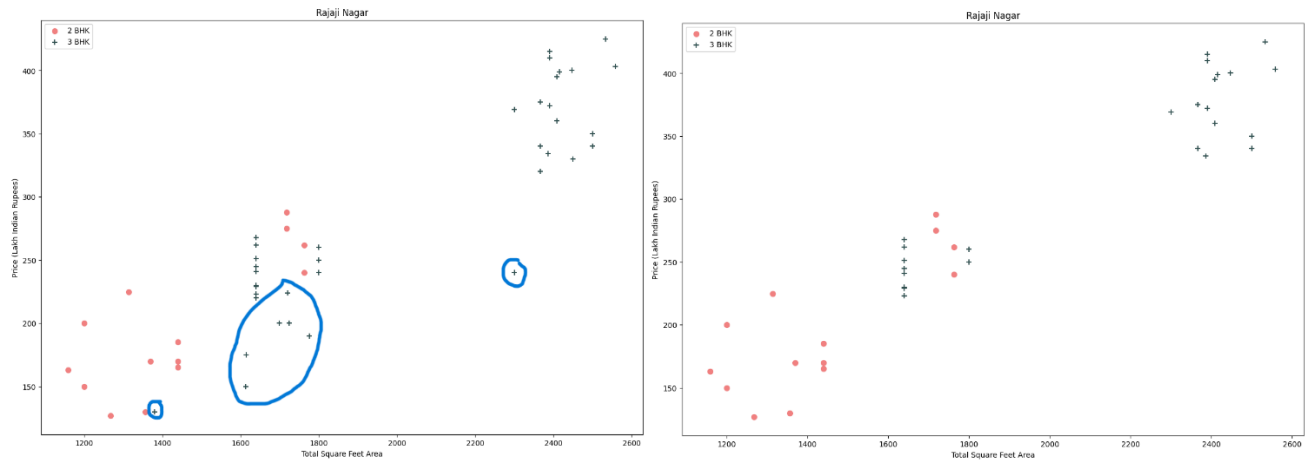
Add new feature called price per square feet

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

	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

6. **Outlier Removal:** The code uses outlier removal techniques to improve the accuracy of regression models by eliminating unrealistic or erroneous data points. Business logic and statistical methods (using mean and standard deviation) are employed to identify and remove outliers from the dataset. Additionally, specific outlier removal is performed for certain property configurations to ensure more accurate modeling.





7. **Splitting Data into Features and Target Variables:** The dataset is divided into two parts: features (independent variables) and the target variable (dependent variable). The features include all columns except for the 'price' column, which is the target variable we want to predict.
8. **Model Selection and Evaluation:** This step involves selecting machine learning models and evaluating their performance. In the provided code, the Linear Regression model (`LinearRegression()` from `sklearn.linear_model`) is used for evaluation. Linear regression is a simple and widely used regression algorithm that predicts the target variable by fitting a linear equation to the independent variables.

The model is evaluated using three methods:

- a. **Cross-Validation using ShuffleSplit:** Cross-validation is performed using the `ShuffleSplit` function from `sklearn.model_selection`. The dataset is randomly shuffled and split into train and test sets multiple times. The `cross_val_score` function calculates the R-squared scores for each split, indicating the goodness of fit of the model. The higher the score, the better the model's performance.
  - b. **Grid Search with Cross-Validation:** `GridSearchCV` is used to tune hyperparameters of the model. It exhaustively searches for the best combination of hyperparameters by evaluating the model's performance on each combination. The `GridSearchCV` function from `sklearn.model_selection` is used, and the model's performance is evaluated using cross-validation. The function returns the best score and corresponding best parameters for each model. In this code, grid search is performed for **linear regression, lasso regression, and decision tree regression models**.
  - c. **Prediction of Property Prices:** The `predict_price` function takes inputs such as location, total square feet area, number of bathrooms, and number of bedrooms. It uses the trained `LinearRegression` model to predict the property price based on these inputs. The function creates a feature array with a one-hot encoded location and passes it to the model's `predict` method to obtain the predicted price.
9. **Saving the Model and Feature Scaling:** After selecting the best model based on evaluation, the next step involves fitting the model on the entire dataset and preparing it for deployment. Before that, it's important to perform feature scaling if necessary. Feature scaling ensures that all features have a similar scale, which can help improve the performance of certain models.

In this code snippet, `MinMaxScaler` from `sklearn.preprocessing` is used to scale the features. `MinMaxScaler` scales each feature to a specified range (by default, between 0 and 1) based on the minimum and maximum values in the feature. This ensures that all features are on a similar scale and prevents any single feature from dominating the learning process.

10. **Saving the Model:** Once the model is trained and ready for deployment, it is saved using the joblib library (from sklearn.externals). The **dump** function is used to save the model object to a file, which can be later loaded and used for making predictions without retraining the model.

### Data Mining algorithms used:-

```
"""# **Find best model using GridSearchCV**"""  
  
import pandas as pd  
from sklearn.model_selection import GridSearchCV, ShuffleSplit, cross_val_score  
from sklearn.linear_model import LinearRegression, Lasso, Ridge  
from sklearn.tree import DecisionTreeRegressor  
from sklearn.preprocessing import StandardScaler  
  
def find_best_model_using_gridsearchcv(X,y):  
  
    algos = {  
        'linear_regression': {  
            'model': LinearRegression(),  
            'params': {  
                'copy_X': [True, False],  
                'fit_intercept': [True, False],  
                'n_jobs': [None, -1],  
                'positive': [False],  
                # '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)

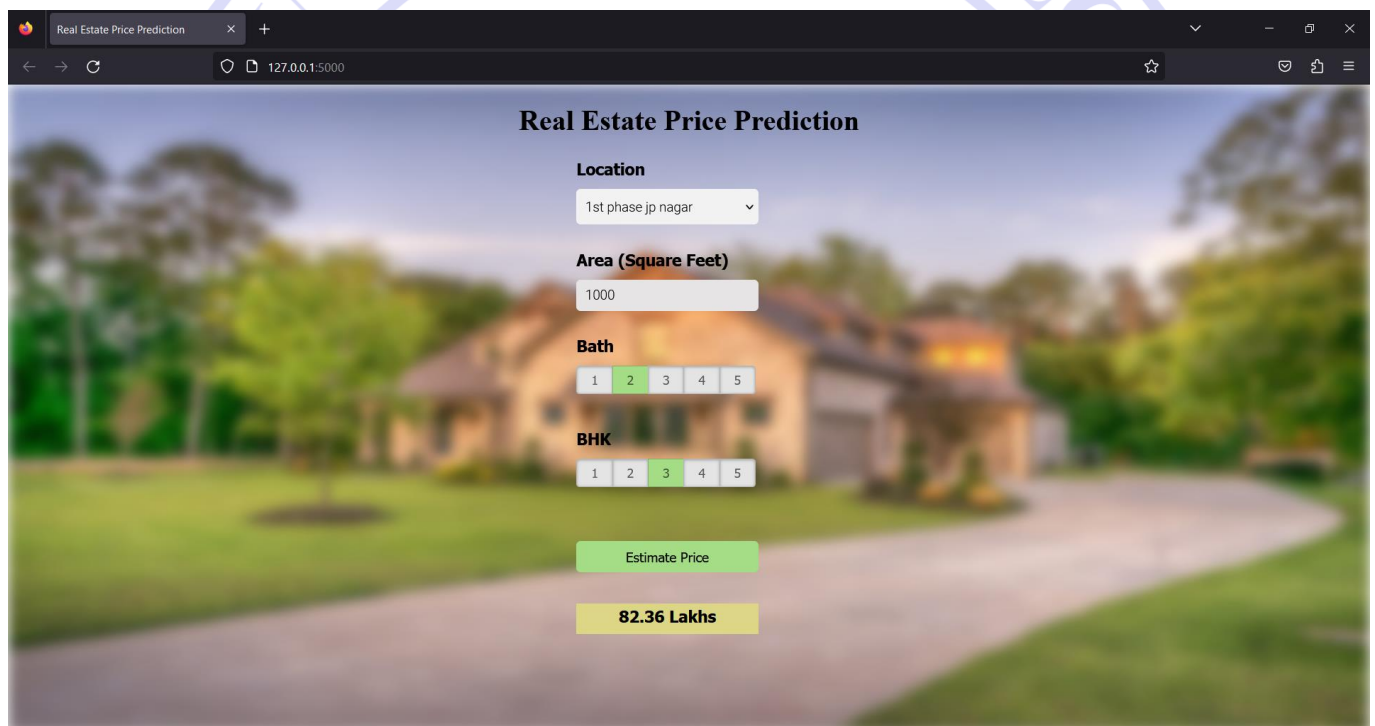
```

## RESULTS:-

	model	best_score	best_params
0	linear_regression	0.847951	{'copy_X': True, 'fit_intercept': False, 'n_jo...
1	lasso	0.726823	{'alpha': 2, 'selection': 'random'}
2	decision_tree	0.708385	{'criterion': 'friedman_mse', 'splitter': 'best'}

## ▼ Test the model for few properties

```
def predict_price(location,sqft,bath,bhk):  
    loc_index = np.where(X.columns==location)[0][0]  
  
    x = np.zeros(len(X.columns))  
    x[0] = sqft  
    x[1] = bath  
    x[2] = bhk  
    if loc_index >= 0:  
        x[loc_index] = 1  
  
    return lr_clf.predict([x])[0]  
  
[ ] print(predict_price('1st Phase JP Nagar', 1000, 2, 3).round(3),'Lakhs')  
  
82.364 Lakhs  
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning:  
warnings.warn()
```



The screenshot shows a web browser window with the title 'Real Estate Price Prediction'. The page has a background image of a house. The form contains the following fields and values:

- Location:** A dropdown menu showing '1st phase jp nagar'.
- Area (Square Feet):** A text input field containing '1000'.
- Bath:** A row of five buttons labeled 1, 2, 3, 4, 5. The button '2' is highlighted in green.
- BHK:** A row of five buttons labeled 1, 2, 3, 4, 5. The button '3' is highlighted in green.
- Estimate Price:** A green button labeled 'Estimate Price'.
- Result:** A yellow button labeled '82.36 Lakhs'.

உழைப்பே உயர்வு தரும்

## **LITERATURE SURVEY**

- [1] Gampala et al. developed a real estate price prediction system using machine learning techniques. They used a dataset of housing features and trained various regression models to predict the sale price of houses. The authors found that the XGBoost algorithm performed the best in terms of accuracy.
- [2] Dey and Urolagin used data mining techniques to develop a real estate price prediction model. They used a dataset of housing features and applied various regression techniques to predict the sale price of houses. The authors found that the Support Vector Regression algorithm outperformed other algorithms in terms of accuracy.
- [3] Matey et al. developed a real estate price prediction model using supervised learning techniques. They used a dataset of housing features and applied various regression techniques to predict the sale price of houses. The authors found that the Random Forest algorithm outperformed other algorithms in terms of accuracy.
- [4] Anand et al. proposed a real estate price prediction model based on various housing features. They applied a regression technique to predict the sale price of houses. The authors found that the Random Forest algorithm performed the best in terms of accuracy.
- [5] Panchal et al. developed a real estate price prediction model called MakanSETU using machine learning techniques. They used a dataset of housing features and applied various regression techniques to predict the sale price of houses. The authors found that the Gradient Boosting algorithm outperformed other algorithms in terms of accuracy.
- [6] Cekic et al. proposed an artificial intelligence approach for modeling house price prediction. They used a dataset of housing features and applied various regression techniques to predict the sale price of houses. The authors found that the Random Forest algorithm outperformed other algorithms in terms of accuracy.
- [7] Gao et al. developed a multi-task learning approach for location-centered house price prediction. They used a dataset of housing features and applied various regression techniques to predict the sale price of houses in different locations. The authors found that their approach outperformed other methods in terms of accuracy.
- [8] Ahtesham et al. developed a house price prediction model using machine learning techniques in the context of the Karachi city in Pakistan. They used a dataset of housing features and applied various regression techniques to predict the sale price of houses. The authors found that the Gradient Boosting algorithm outperformed other algorithms in terms of accuracy.
- [9] Li and Chu developed a real estate price variation prediction model based on economic parameters. They applied a regression technique to predict the price changes of houses. The authors found that the model could accurately predict the price changes of houses.
- [10] Xue developed a real estate price prediction model based on a Back Propagation Neural Network (BPNN) algorithm. They used a dataset of housing features and applied the BPNN algorithm to predict the sale price of houses. The author found that the model could accurately predict the sale price of houses.

## **REFERENCES**

- [1] V. Gampala, N. Y. Sai and T. N. Sai Bhavya, "Real-Estate Price Prediction System using Machine Learning," 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2022, pp. 533-538, doi: 10.1109/ICAAIC53929.2022.9793177.
- [2] S. K. Dey and S. Urolagin, "Real Estate Price Prediction using Data Mining Techniques," 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON), Kuala Lumpur, Malaysia, 2021, pp. 1-4, doi: 10.1109/GUCON50781.2021.9573829.
- [3] V. Matey, N. Chauhan, A. Mahale, V. Bhistannavar and A. Shitole, "Real Estate Price Prediction using Supervised Learning," 2022 IEEE Pune Section International Conference (PuneCon), Pune, India, 2022, pp. 1-5, doi: 10.1109/PuneCon55413.2022.10014818.
- [4] S. Anand, P. Yadav, A. Gaur and I. Kashyap, "Real Estate Price Prediction Model," 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2021, pp. 541-543, doi: 10.1109/ICAC3N53548.2021.9725772.
- [5] Y. Panchal, M. Mer and A. Ghosh, "Residential Property Price Prediction Using Machine Learning: MakanSETU," 2022 International Seminar on Application for Technology of Information and Communication (iSemantic), Semarang, Indonesia, 2022, pp. 257-262, doi: 10.1109/iSemantic55962.2022.9920395.
- [6] M. Cekic, K. N. Korkmaz, H. Müküs, A. A. Hameed, A. Jamil and F. Soleimani, "Artificial Intelligence Approach for Modeling House Price Prediction," 2022 2nd International Conference on Computing and Machine Intelligence (ICMI), Istanbul, Turkey, 2022, pp. 1-5, doi: 10.1109/ICMI55296.2022.9873784.
- [7] Guangliang Gao, Zhifeng Bao, Jie Cao, A. K. Qin, and Timos Sellis. 2022. Location-Centered House Price Prediction: A Multi-Task Learning Approach. ACM Trans. Intell. Syst. Technol. 13, 2, Article 32 (April 2022), 25 pages. <https://doi-org.egateway.vit.ac.in/10.1145/3501806>
- [8] M. Ahtesham, N. Z. Bawany and K. Fatima, "House Price Prediction using Machine Learning Algorithm - The Case of Karachi City, Pakistan," 2020 21st International Arab Conference on Information Technology (ACIT), Giza, Egypt, 2020, pp. 1-5, doi: 10.1109/ACIT50332.2020.9300074.
- [9] L.Li and K. -H. Chu, "Prediction of real estate price variation based on economic parameters," 2017 International Conference on Applied System Innovation (ICASI), Sapporo, Japan, 2017, pp. 87-90, doi: 10.1109/ICASI.2017.7988353.
- [10] H. Xue, "The Prediction on Residential Real Estate Price Based on BPNN," 2015 8th International Conference on Intelligent Computation Technology and Automation (ICICTA), Nanchang, China, 2015, pp. 1008-1013, doi: 10.1109/ICICTA.2015.256.