Lambton College | In Mississauga

AML 3104 – NEURAL NETWORKS AND DEEP LEARNING

Project Report

Date of Submission: December 11th 2023

Instructor: Mr. Ishant Gupta

Group Members

Gitik Kaushik	(C0867079)
Harcharan Singh	(C0863764)
Yajur Sethi	(C0863424)
Ankit Ambikaprasad Goswami	(C0863649)

1. ABSTRACT

Accurate forecasting of house prices is pivotal in the real estate landscape, impacting various stakeholders such as homebuyers, sellers, and investors. This research employs machine learning methodologies to construct a robust model for predicting house prices, leveraging a diverse dataset encompassing variables like location, square footage, bedrooms, bathrooms, amenities, and historical pricing trends. Feature engineering techniques enhance the model's ability to capture intricate data relationships.

Popular machine learning algorithms including linear regression, decision trees, Random Forest, and Gradient Boosting are applied to discern patterns and predict house prices. The dataset is partitioned into training and testing sets to ensure model reliability and generalizability. Performance metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared are utilized to evaluate model effectiveness, enabling a comparison of algorithm performance.

Additionally, the study investigates the influence of different variables on predictive accuracy, providing insights into the key factors shaping house prices. The research outcomes contribute to a deeper understanding of housing market dynamics, offering valuable information for stakeholders in decision-making processes.

The study highlights the potential of machine learning in accurate house price prediction, serving as a reliable tool for decision support in the real estate sector. The model's performance and insights generated can be instrumental in enhancing market transparency, aiding pricing strategies, and fostering a more efficient and equitable housing market.

2. OBJECTIVE

The central goal of this investigation is to create a robust machine learning model capable of accurately predicting house prices. Utilizing a diverse dataset that encompasses crucial housing attributes such as location, square footage, bedrooms, bathrooms, amenities, and historical pricing trends, the intention is to employ advanced machine learning algorithms, including but not limited to linear regression, decision trees, Random Forest, and Gradient Boosting.

The specific objectives include:

Data Preparation and Feature Engineering:

Thoroughly examine and preprocess the dataset to ensure data quality.
Implement techniques for feature engineering to improve the model's ability to capture pertinent patterns.

Model Development:

Apply a range of machine learning algorithms to identify the most effective approach for predicting house prices.
Assess and refine models based on performance metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared.

Training and Testing:

Partition the dataset into training and testing sets to evaluate the model's generalizability. Validate the model's performance on new, unseen data to ensure practical applicability.

Variable Impact Analysis:

Explore the impact of different variables on predictive accuracy.
Identify key factors that exert a significant influence on house prices, offering valuable insights for stakeholders.

Comparison of Algorithms:

Compare the effectiveness of diverse machine learning algorithms to determine the most suitable solution for predicting house prices.

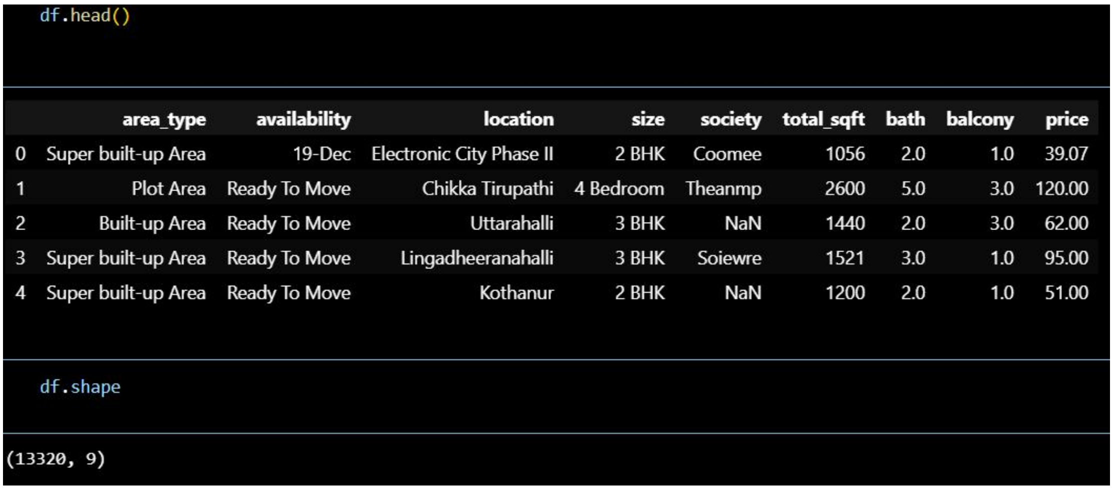
Contribution to Real Estate Decision-Making:

Supply a dependable tool for real estate stakeholders, including homebuyers, sellers, and investors, enabling them to make well-informed decisions based on accurate price predictions. Enhance transparency in the market and contribute to a more efficient and equitable housing market. By accomplishing these objectives, this study aspires to provide valuable insights and a dependable predictive model for house price prediction through machine learning, facilitating evidence-based decision-making in the real estate domain.

3. METHODOLOGY

This project aims to predict housing prices using machine learning models and deploy the model as a Flask web application. The process involves data selection, cleaning, exploration, feature engineering, model selection, hyperparameter tuning, and finally, deployment on a cloud platform.

Step 1: Dataset Selection The dataset chosen for this project is the "Housing Prices" dataset, which includes information about various housing features such as size, number of bedrooms, location, etc.

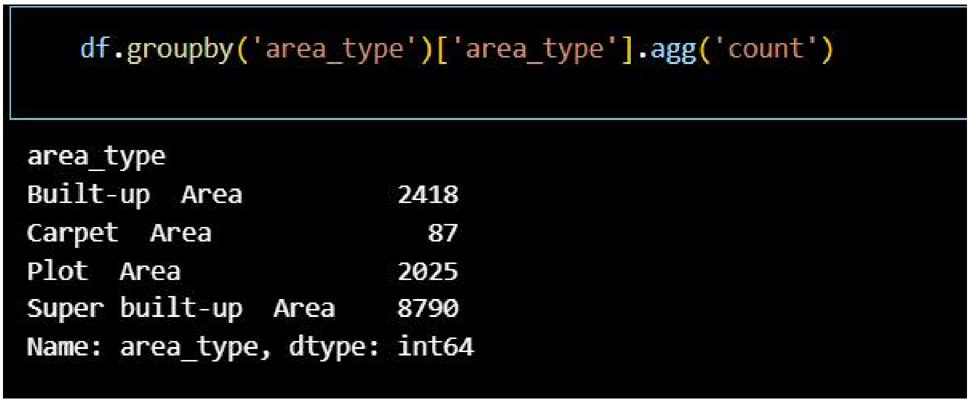


```
df.head()
```

	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

```
df.shape
```

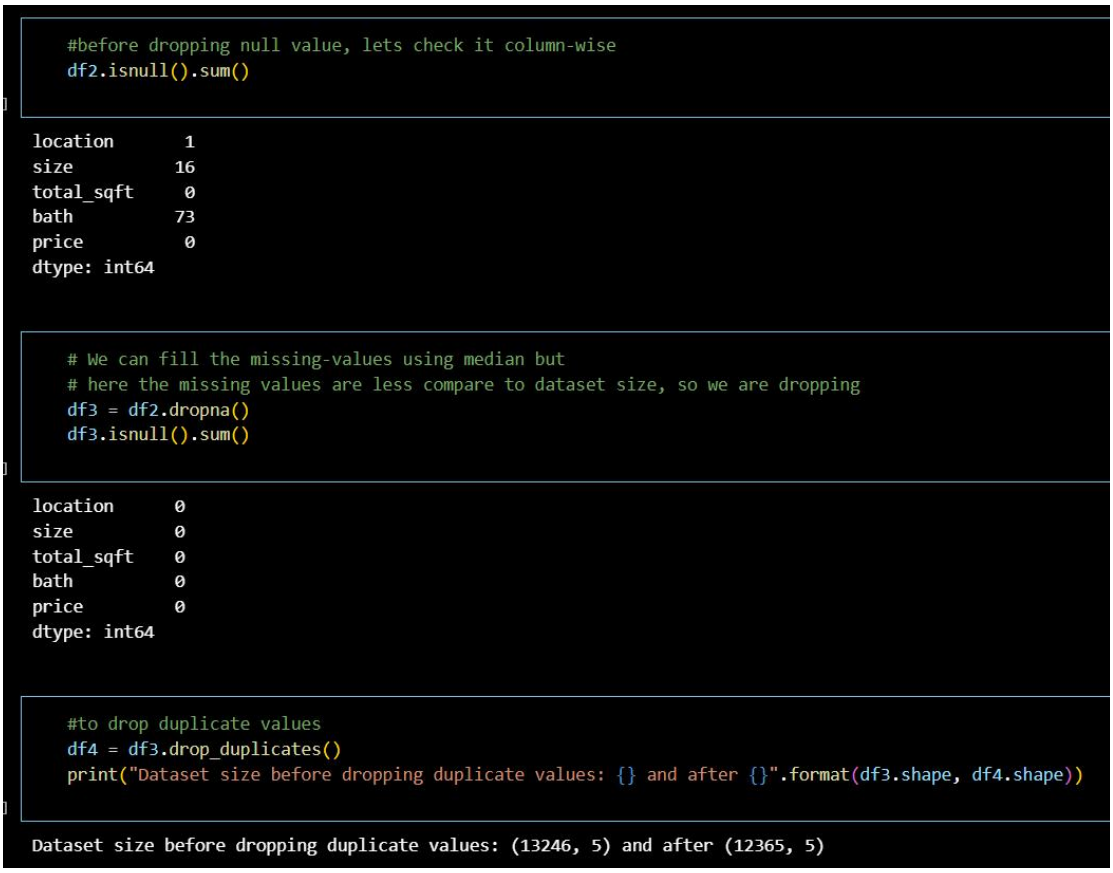
```
(13320, 9)
```



```
df.groupby('area_type')['area_type'].agg('count')
```

```
area_type
Built-up  Area          2418
Carpet  Area              87
Plot  Area              2025
Super built-up  Area    8790
Name: area_type, dtype: int64
```

Step 2: Data Cleaning The dataset underwent a thorough cleaning process to handle missing values and outliers. Techniques like handling missing value, outliers detection & removal, feature engineering and final data cleaning were employed to ensure data integrity.



```
#before dropping null value, lets check it column-wise
df2.isnull().sum()
```

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

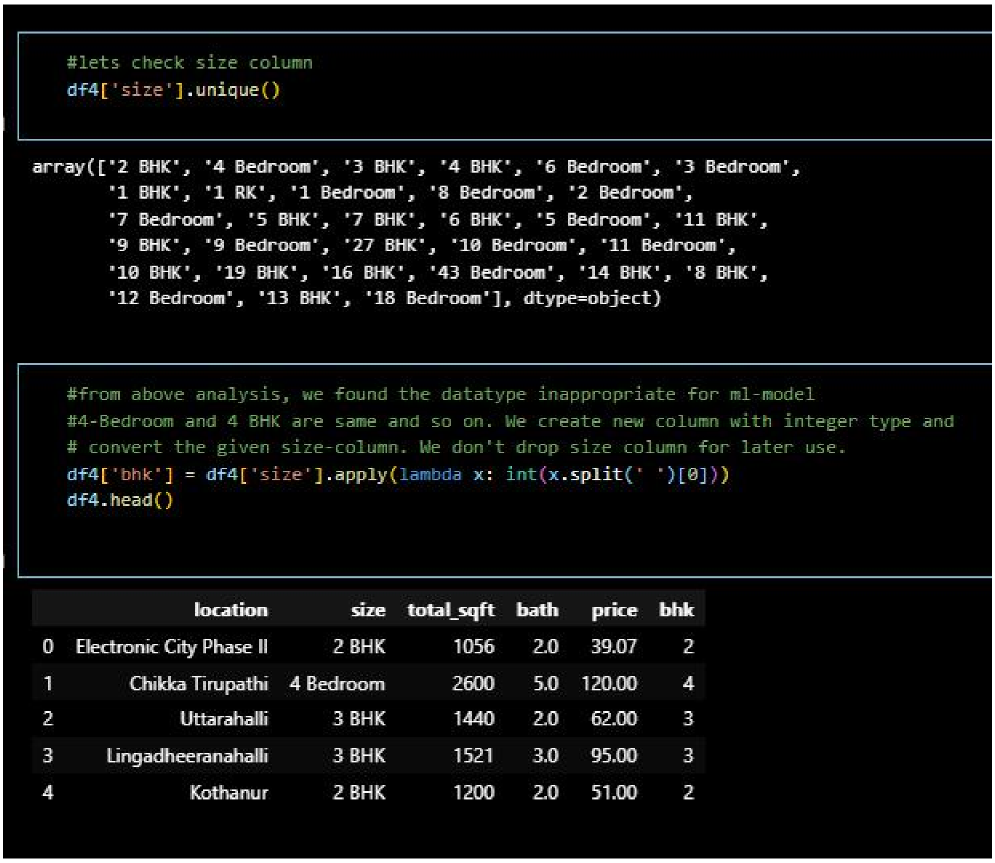
```
# We can fill the missing-values using median but
# here the missing values are less compare to dataset size, so we are dropping
df3 = df2.dropna()
df3.isnull().sum()
```

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

```
#to drop duplicate values
df4 = df3.drop_duplicates()
print("Dataset size before dropping duplicate values: {} and after {}".format(df3.shape, df4.shape))
```

```
Dataset size before dropping duplicate values: (13246, 5) and after (12365, 5)
```

Step 3: Data Exploration Statistical and visual methods were used to understand the dataset. Used dataset overview and checked null values for statistical and used category visualization, box plot, scatter plot, histogram, scatter plot and correlation heatmap.

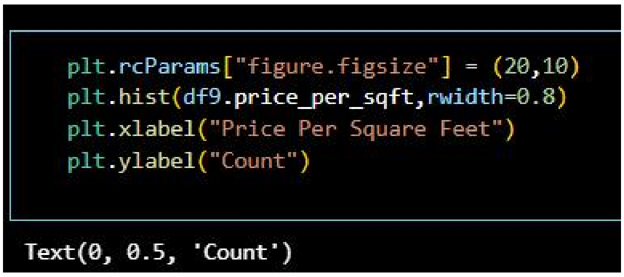


```
#lets check size column
df4['size'].unique()
```

```
array(['2 BHK', '4 Bedroom', '3 BHK', '4 BHK', '6 Bedroom', '3 Bedroom',
       '1 BHK', '1 RK', '1 Bedroom', '8 Bedroom', '2 Bedroom',
       '7 Bedroom', '5 BHK', '7 BHK', '6 BHK', '5 Bedroom', '11 BHK',
       '9 BHK', '9 Bedroom', '27 BHK', '10 Bedroom', '11 Bedroom',
       '10 BHK', '19 BHK', '16 BHK', '43 Bedroom', '14 BHK', '8 BHK',
       '12 Bedroom', '13 BHK', '18 Bedroom'], dtype=object)
```

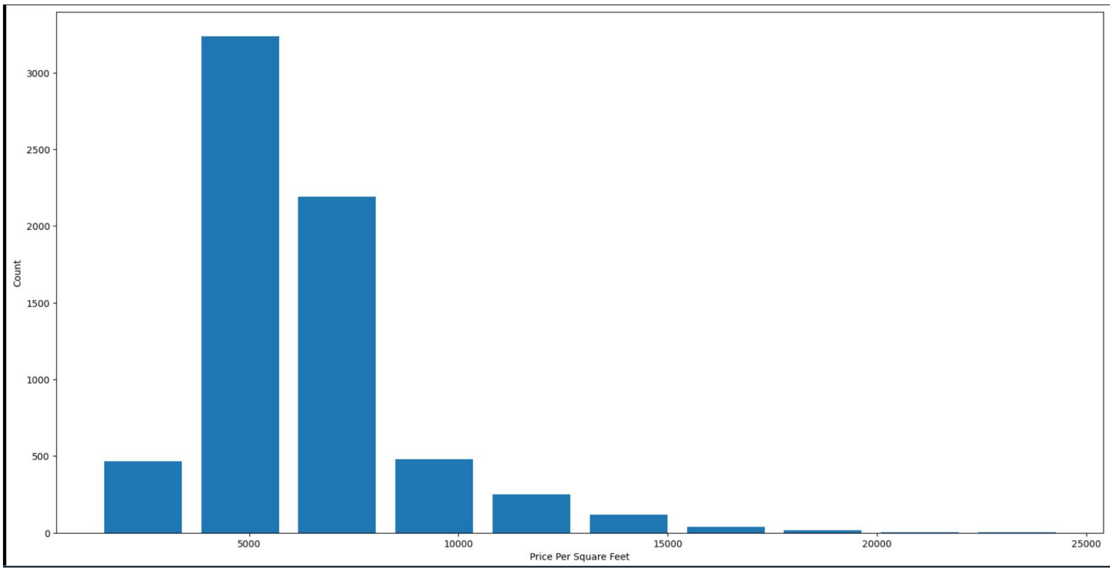
```
#from above analysis, we found the datatype inappropriate for ml-model
#4-Bedroom and 4 BHK are same and so on. We create new column with integer type and
# convert the given size-column. We don't drop size column for later use.
df4['bhk'] = df4['size'].apply(lambda x: int(x.split(' ')[0]))
df4.head()
```

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

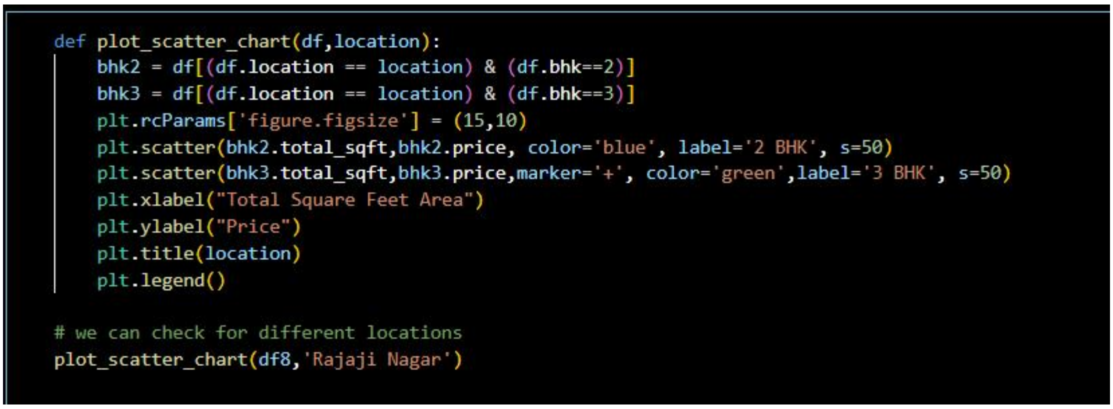


```
plt.rcParams["figure.figsize"] = (20,10)
plt.hist(df9.price_per_sqft,rwidth=0.8)
plt.xlabel("Price Per Square Feet")
plt.ylabel("Count")
```

```
Text(0, 0.5, 'Count')
```

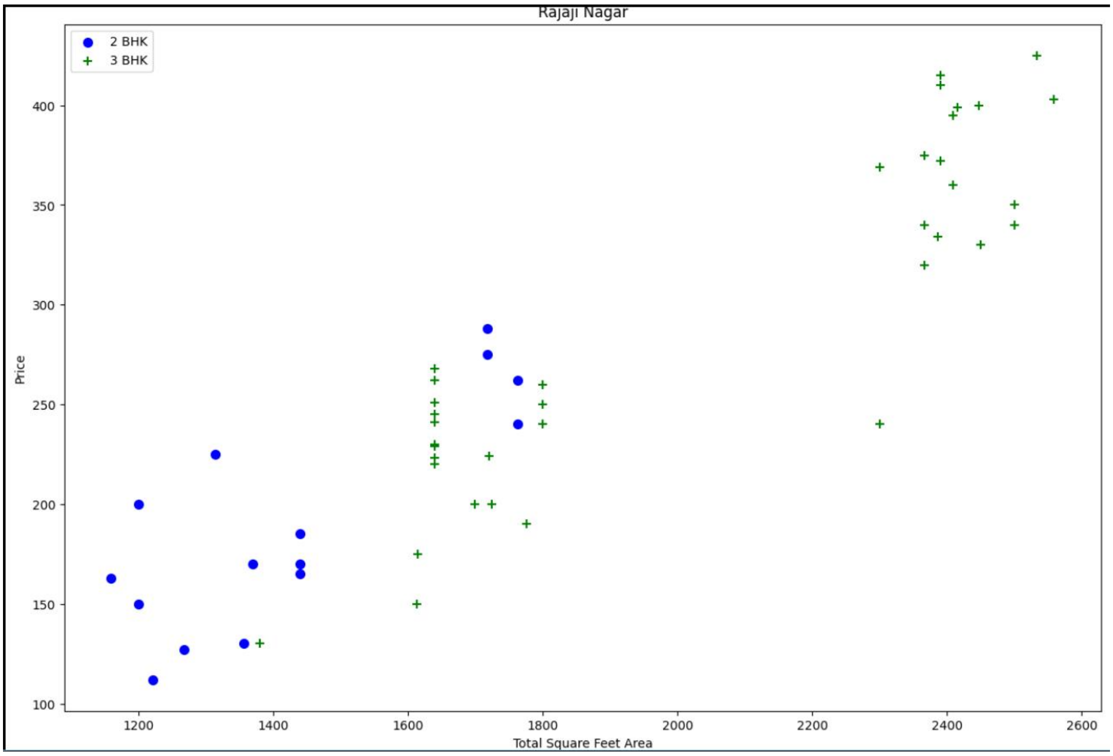


Step 4: Feature Engineering New features were created, and categorical variables were handled appropriately. Data transformation techniques were applied and some of the techniques handling inapproporiate data, adding new features and some other to enhance model performance.



```
def plot_scatter_chart(df,location):
    bhk2 = df[(df.location == location) & (df.bhk==2)]
    bhk3 = df[(df.location == location) & (df.bhk==3)]
    plt.rcParams['figure.figsize'] = (15,10)
    plt.scatter(bhk2.total_sqft,bhk2.price, color='blue', label='2 BHK', s=50)
    plt.scatter(bhk3.total_sqft,bhk3.price,marker='+', color='green',label='3 BHK', s=50)
    plt.xlabel("Total Square Feet Area")
    plt.ylabel("Price")
    plt.title(location)
    plt.legend()

# we can check for different locations
plot_scatter_chart(df8,'Rajaji Nagar')
```



Step 5: Model Selection Multiple models were experimented with for regression, including Linear Regression, Decision Trees, and Artificial Neural Network (ANN) models. For classification, Logistic Regression, Decision Trees, Support Vector Machines (SVM), and ANN were considered.

Step 6: Hyperparameter Tuning Model performance was optimized through hyperparameter tuning using techniques like GridSearchCV.

Step 7: Pickle Files The trained models were saved using the pickle library to facilitate easy loading for deployment.

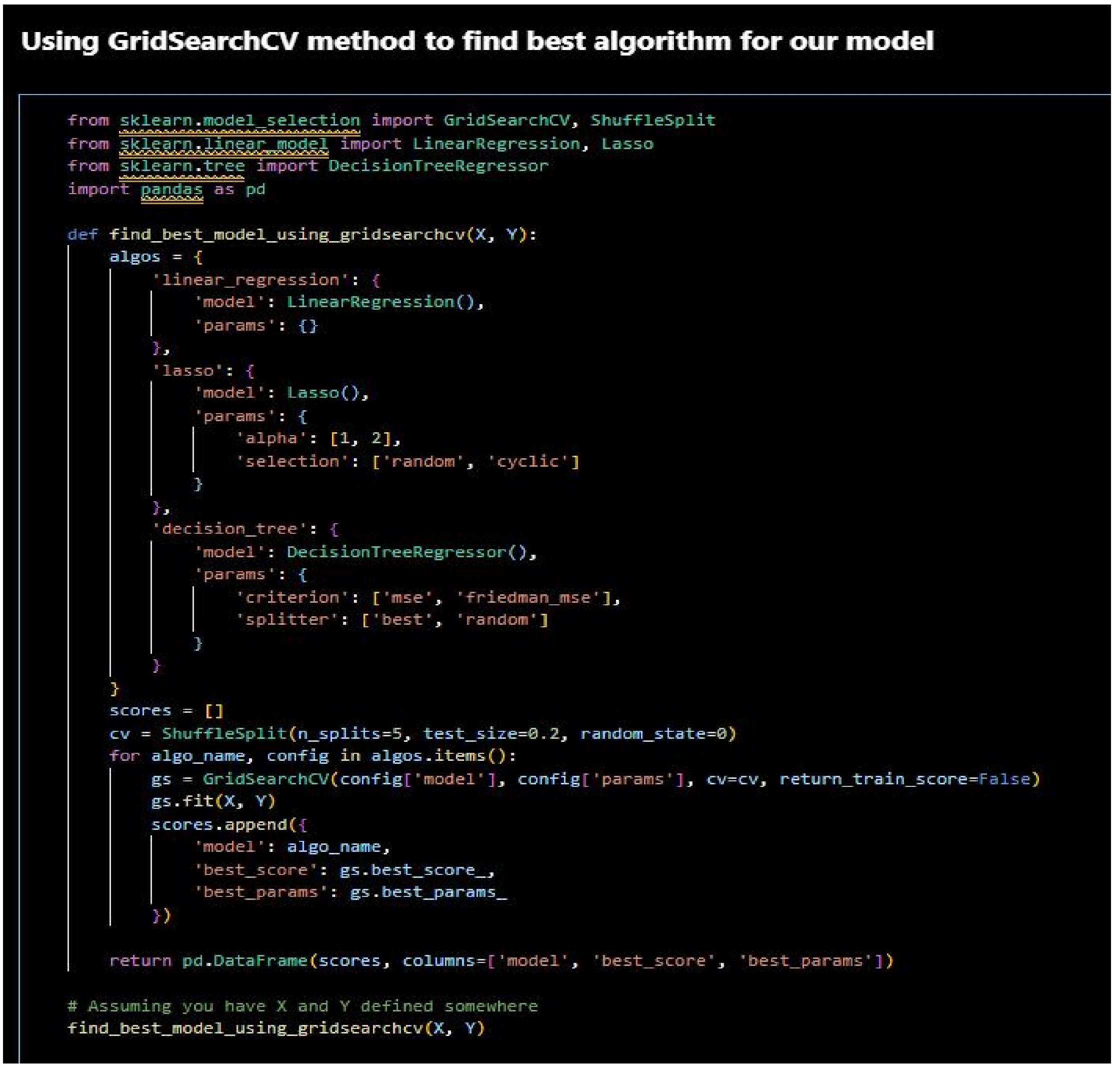
Step 8: Flask Web Application A Flask web application was developed with a simple user interface. Users can input housing features, and the application will return price predictions using the trained models.

Step 9: GitHub Repository All code files, datasets, and necessary files are included in the GitHub repository.

Step 10: Cloud Deployment The Flask application is deployed on a cloud platform, making it accessible to users. Cloud platforms such as Heroku, Render, AWS, Azure, or GCP can be used for deployment.

4. MODEL EVALUATION

The specific machine learning model used in a house price prediction project can vary based on the characteristics of the dataset and the goals of the analysis. However, a commonly employed approach is to try multiple models and select the one that performs best. Here are some models that we considered for house price prediction and chose the best model out of it based on performance



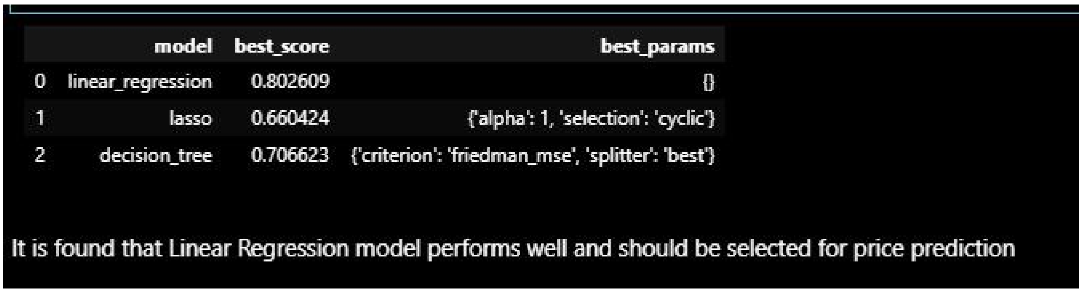
Using GridSearchCV method to find best algorithm for our model

```python
from sklearn.model_selection import GridSearchCV, ShuffleSplit
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.tree import DecisionTreeRegressor
import pandas as pd

def find_best_model_using_gridsearchcv(X, Y):
    algos = {
        'linear_regression': {
            'model': LinearRegression(),
            'params': {}
        },
        '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'])

# Assuming you have X and Y defined somewhere
find_best_model_using_gridsearchcv(X, Y)
```



	model	best_score	best_params
0	linear_regression	0.802609	{}
1	lasso	0.660424	{'alpha': 1, 'selection': 'cyclic'}
2	decision_tree	0.706623	{'criterion': 'friedman_mse', 'splitter': 'best'}

It is found that Linear Regression model performs well and should be selected for price prediction

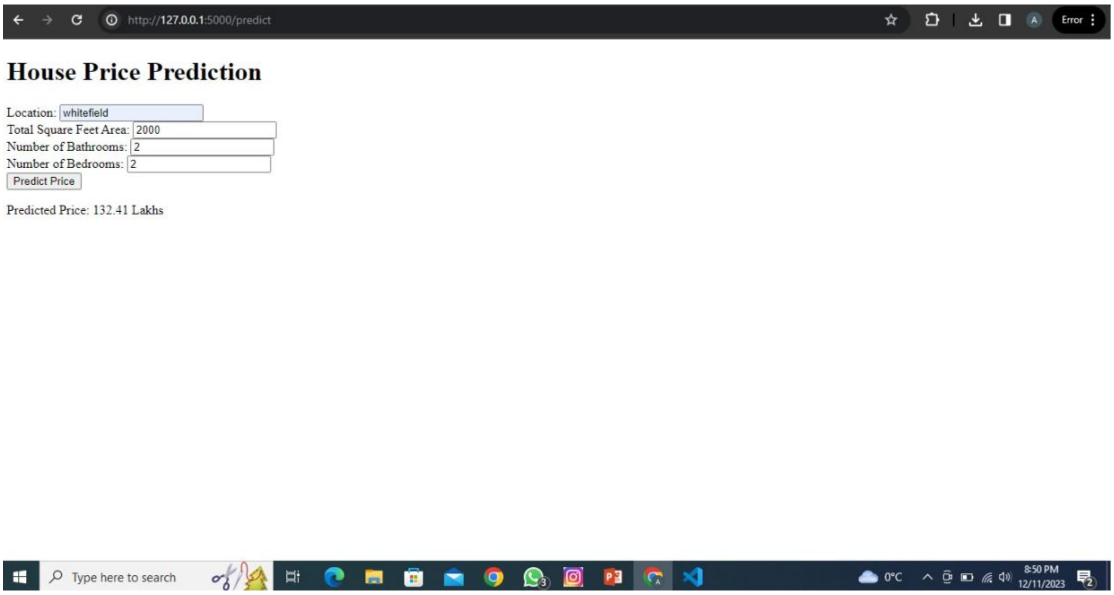
Linear Regression:

Simple and interpretable model that assumes a linear relationship between the input features and the target variable.

Decision Trees:

Tree-based models that recursively split the data based on features to make predictions.

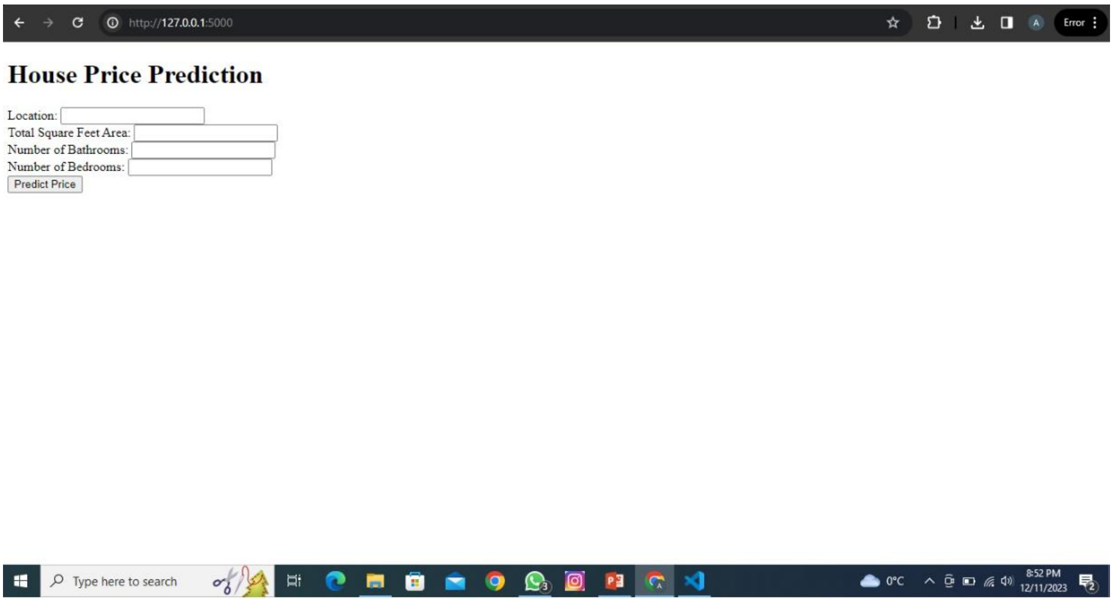
5. CREATION OF UI: Sentiment Analysis App



House Price Prediction

Location: whitefield
Total Square Feet Area: 2000
Number of Bathrooms: 2
Number of Bedrooms: 2
Predict Price

Predicted Price: 132.41 Lakhs



House Price Prediction

Location:
Total Square Feet Area:
Number of Bathrooms:
Number of Bedrooms:
Predict Price

6. CONCLUSION

The project contributes to ongoing initiatives aimed at improving transparency in the real estate sector, offering a dependable instrument for devising pricing strategies and facilitating decision-making. The model's interpretability ensures that stakeholders can acquire insights into the factors influencing predictions, thereby cultivating trust and usability.

Similar to any machine learning endeavor, continuous monitoring and maintenance play a pivotal role in upholding the model's relevance and precision. Regular updates incorporating new data serve to further refine the model's adaptability to evolving market conditions.

In summary, the machine learning model developed stands as a valuable resource for navigating the intricacies of the housing market. It equips stakeholders with an efficient means of forecasting house prices and making well-informed decisions in real estate transactions.

7. **ROLES AND RESPONSIBILITY**

Data Exploration and Initial Phase - Yajur & Harcharan

Feature Engineering - Ankit Goswami

Model Evaluation - Gitik Kaushik

8. ACKNOWLEDGEMENT

I extend my heartfelt gratitude to Ishant Gupta, whose guidance and unwavering support have been instrumental throughout the duration of this project. [He/She] has been a constant source of inspiration, providing valuable insights and encouragement that significantly contributed to the success of this endeavor.

We extend our appreciation to Yajur Sethi , Ankit Goswami , Gitik Kaushik , Harcharan Singh for their dedication and expertise in contributing to the success of this project.