LONDON
METROPOLITAN
UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

Level 6 – Applied Machine Learning (CC6057NI)

Assessment Type: Coursework

Semester: 1st

2024/25 Autumn

Student Name: Asmita Basnet

London Met ID: 22085764

College ID: NP01AI4S230019

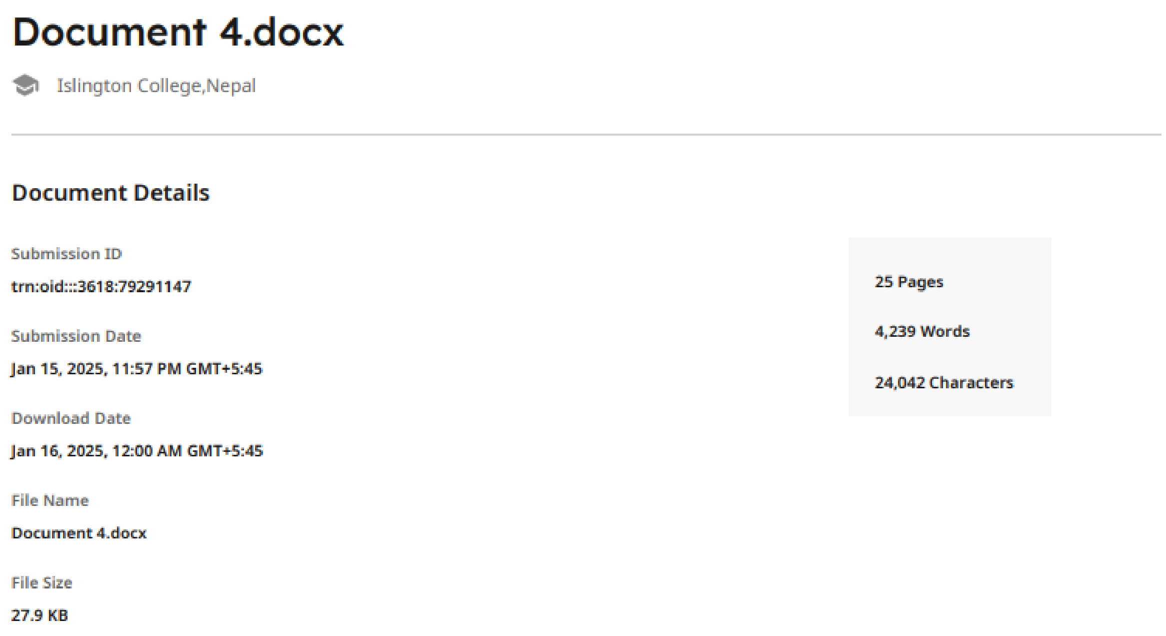
Assignment Due Date: Wednesday, January 15, 2025

Assignment Submission Date: Wednesday, January 15, 2025

Submitted To: Mahotsav Bhattarai

Word Count (Where Required):4306

I confirm that I understand my coursework needs to be submitted online via MST Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.



Document 4.docx

Islington College,Nepal

Document Details

Submission ID

trn:oid:::3618:79291147

Submission Date

Jan 15, 2025, 11:57 PM GMT+5:45

Download Date

Jan 16, 2025, 12:00 AM GMT+5:45

File Name

Document 4.docx

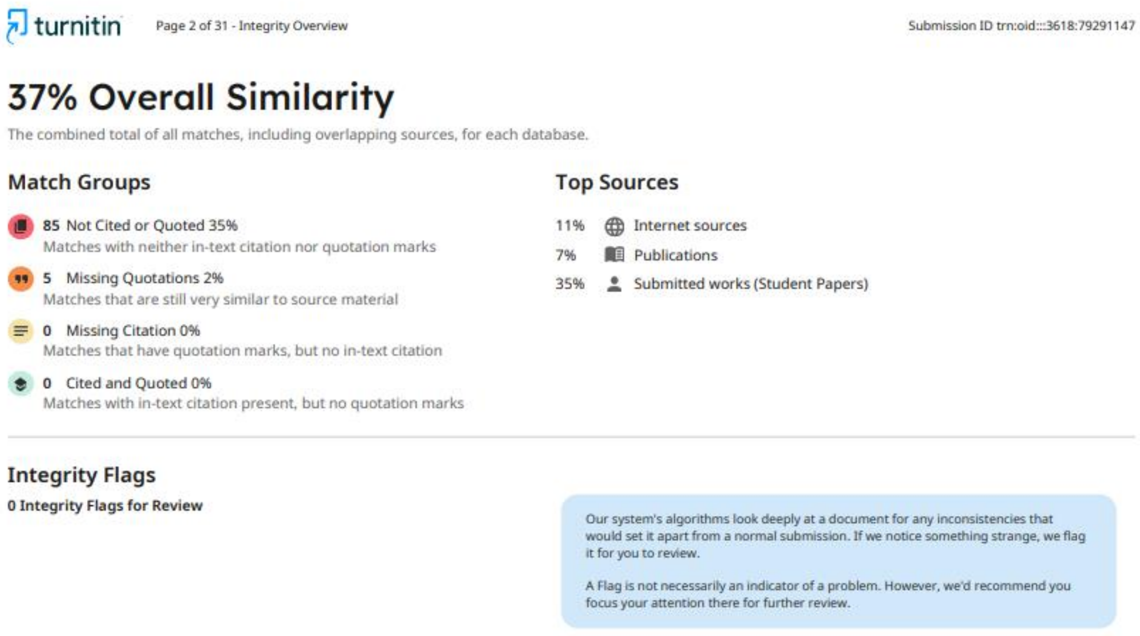
File Size

27.9 KB

25 Pages

4,239 Words

24,042 Characters



turnitin Page 2 of 31 - Integrity Overview Submission ID trn:oid:::3618:79291147

37% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- 85 Not Cited or Quoted 35%
  Matches with neither in-text citation nor quotation marks
- 5 Missing Quotations 2%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

11%	Internet sources
7%	Publications
35%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

1.1. Overview

The home price prediction model for Bengaluru predicts prices of Bengaluru real estate properties via machine learning algorithms. This report uses machine learning techniques to estimate the value of houses based on parameters like size and location. In major cities, such as Bengaluru, where real estate investments involve huge sums of money and very long-term debts, knowing how to ascertain and estimate home prices is critical. (Kumari Sandhya1, 2022)Many buyers have a hard time to determine the home price based on surroundings and listings; with the machine learning techniques, they can predict prices accurately. Property-related information can be modeled to yield high predictive power in forecasting prices. The model will be using Python programming along with data extracting, processing, and analysis tools like Pandas and NumPy.

1.1. Introduction To Machine Learning Concept

Machine learning is a subfield of artificial intelligence that allows systems to learn from experience and to improve their algorithms without programming explicitly for it. The ML model runs data where historical prices and property features will be processed to identify patterns helpful in predicting real estate prices. The main focus of this research was on supervised learning, where the model predicts house prices utilizing labeled data. Data science techniques, especially regression models, find extensive utility in predictive analysis across a multitude of domains. The primary focus of this study was to derive a robust regression model aimed at predicting house prices more accurately in dynamic Bengaluru real estate markets. Various models have been pitted against one another in a bid to track down the best approach toward precise price prediction. (Manasa, 2020) Regression is one of the basic tools in machine learning and statistical analysis that is used for modeling the relationship between one dependent variable (target) and one or more independent variables (features). Regressions are the main tools in regard to predicting and analyzing, with the aim of estimating continuous values. (Shinde, 2018)

The Common metrics for evaluation of regressive models in project include:

- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)

- Mean Absolute Error (MAE)
- R-squared (Coefficient of Determination)

1.2. Importance Of House Price Prediction

House price prediction is a key task in the real estate market as it helps buyers and sellers make wise decisions. Well-grounded predictions allow buyers to balance their budgets to reflect market trends and find their dream homes with economic constraints. Sellers can estimate a price possibly the most optimal for maximizing profit while staying competitive. Price predictions beyond an individual transaction can provide vital economic planning and policy-making insight, conducting market dynamic analysis and trend forecasting. Therefore, the use various regression techniques, emulating conventional and recent advances in modeling house prices, determined to reconcile the needs with a specific emphasis on real-time, data-informed insight provision in an ever-volatile pricing market. (Madhuri, 2019)

1.3. Role Of AI In House Price Prediction

AI significantly enhances the precision and reliability of house price predictions through advanced regression techniques and machine-learning models. Techniques applied to process huge datasets and capture complex relationships between predictors, such as location, physical attributes, and market conditions, include Multiple Linear Regression, Ridge, LASSO, Elastic Ne, Gradient Boosting, and AdaBoost. AI models provide an approach offering a systematic and data-driven modus operandi, scaling down the error margins and presenting insights conclusive in nature. The comparison in this work emphasizes that AI evolves as an efficient tool in selecting algorithms that assures accurate yet efficient forecasting. (Madhuri, 2019)

2. Problem Domain

2.1. Detailed Explanation Of Problem

Predicting house sale prices in a city like Bengaluru becomes one of the most challenging tasks with a whole array of interaction of several variables. These characteristics of the properties have localities, environments, and other various factors directly or indirectly correlated with the sale price of properties. If that's not enough, real estate also has another dimension of dynamism. This aims to construct a predictive model developed for different housing prices according to their prime influencing variables. Using publicly accessible datasets, the objective is to analyze and assess the numerous factors affecting housing values in Bangalore and thus develop an efficient prediction model. (Manasa, 2020)

2.2. Literature Review

In this section, we cover an overview of the datasets, techniques, and procedures used to address the issue of property price prediction. Various datasets and analytical methods reported upon give results of prior studies. Literature review gives us grounds for finding successful methods, their limitations, and relating our research to a wider context. It is structured in two sections-a Dataset Review, itself involving explanations of the data used, and an Article Review, which describes the methods and findings of important papers on related issues.

2.2.1. Dataset Review

Choosing the right dataset is crucial for machine learning projects since the model results to a large extent depend upon the data quality and variety of features. Learning speed depends on several aspects such as reliability, source presentation format, its consistency, and presence of outliers. Generalized steps in data preparation include acquisition, cleaning, and transformation; each of these steps can be spiced up with mini features. All of them have been realized correctly in this project to make the model reliable and predictable. (Thakur, 2021)

After long and intense searches, the dataset-the housing price dataset in Bengaluru-could be found on the official site of Kaggle. The dataset was just right to provide detailed and, at the same time, realizable information that was pared down to embed several features. It allows enough leeway to explore various features and also meets all other demands in building a good practical model for prediction. This was very competent to build a good ground for this project in research and modeling.

The model is built using the following method:

1. **Dataset Description**

   Source: Kaggle (Link To Dataset)

   File name: Bengaluru_House_Data.csv'

   Dataset Size: 9 columns

Features	**Descriptions**
area_type	A categorical feature denoting the type of property area (e.g. Super built-up Area, Carpet Area).
availability	A categorical feature denoting the availability status of the property (e.g. Ready to Move, Launch Date).
location	A text/string feature providing the property's location in Bengaluru.
size	A categorical feature denoting the size of the property, e.g., 2 BHK, 4 Bedroom.
society	A text/string feature giving the name of the society or the residential colony. It has missing values.
total_sqft	A mixed feature showing the total area of the property in square feet, e.g., "2100-3000".
bath	A numerical feature that describes the number of bathrooms in the property.
balcony	Numerical feature indicating the number of balconies in the property since missing values exist.
price	The numerical target variable that represents the house price

Table 1:Dataset Description Table

2. Use Case For the Dataset

The process begins with an exploratory data analysis (EDA) to identify any patterns or relationships among these features and prices. In this, EDA will shine a light on factors that will have great impact on prices-in this case, the pricing based on location and the size of the property. Following this, feature engineering will allow new and significant variables to be created that will enhance predictive capability in the model. The data set also contains missing and inconsistent values which makes it a good candidate for the practice of data cleaning and preprocessing techniques. Visualization plays an important part in the understanding of data and communicating findings as in location-wise price trends, and size or area effect on property values, which can be shown using intuitive and informative plots. This method, from initiation to completion, will allow for the creation of meaningful predictive models and a great understanding of housing market dynamics.

Challenges:

- Dealing with missing values: Columns like society and balcony have some missing values.
- Data cleaning: The total_sqft column has combined values ("2100", "2600-3000") that need pre-processing.
- Outlier detection: The price column can have one or more extremely high values.
- Categorical Data Encoding: Columns must be encoded so that predictive modelling can take place.

2.2.2. Article Review

- Article 1:

The study, termed "Bangalore House Price Prediction," by Thakur in the year 2021, examined how machine learning techniques can predict prices of houses prevailing in Bengaluru, India. Algorithms like XGBoost, Random Forest, Decision Trees, and Linear Regression were applied to the Bengaluru housing price dataset to validate the various methods for reliably predicting real estate prices. Random Forest and XGBoost were found to be the best for prediction accuracy, as they have higher R-squared values and lower MSE values. It demonstrated that relevant features like location, area, and the number of bedrooms play a significant role in determining the price. It also outlined the need for feature selection and proper data preprocessing for better model performance. The findings noted that Random Forest and XGBoost have become practical tools for property price prediction, mostly applied in the area of providing information for administration within Bengaluru. (Amey Thakur1, 2021)

- Article 2:

In the research paper "House Price Prediction Using Machine Learning" (2022), various machine learning techniques have been used to predict the value of houses with emphasis given to the features of location, areas, and number of bedrooms and bathrooms. This outlines various regression techniques to estimate house prices like XGBoost Regression, Random Forests Regression, Lasso Regression, Ridge Regression, Support Vector Regression (SVR), and Linear Regression. The capacities of each model of dealing with a particular problem, whether it is multicollinearity or a non-linear connection in data, are evaluated. The advantages of regularized models such as Ridge and Lasso for giving more accurate predictions are pointed out in this study while the ensemble approaches like Boost and Random forests improve the performance of the model by combing several predictions. The evaluation criteria used for comparing these models includes mean square. (Sandhya1, 2022)

3. Solution

3.1. Diagrammatic Representation

The flowchart presents a visual description of the steps included in the methodology. It shows the conversion of raw data into processed and encoded data, which then feeds into model training, evaluation, and prediction. Such a graphic representation gives clarity to the data flow and methods of developing an effective house price prediction model.

3.1.1. Flowchart For the Entire System

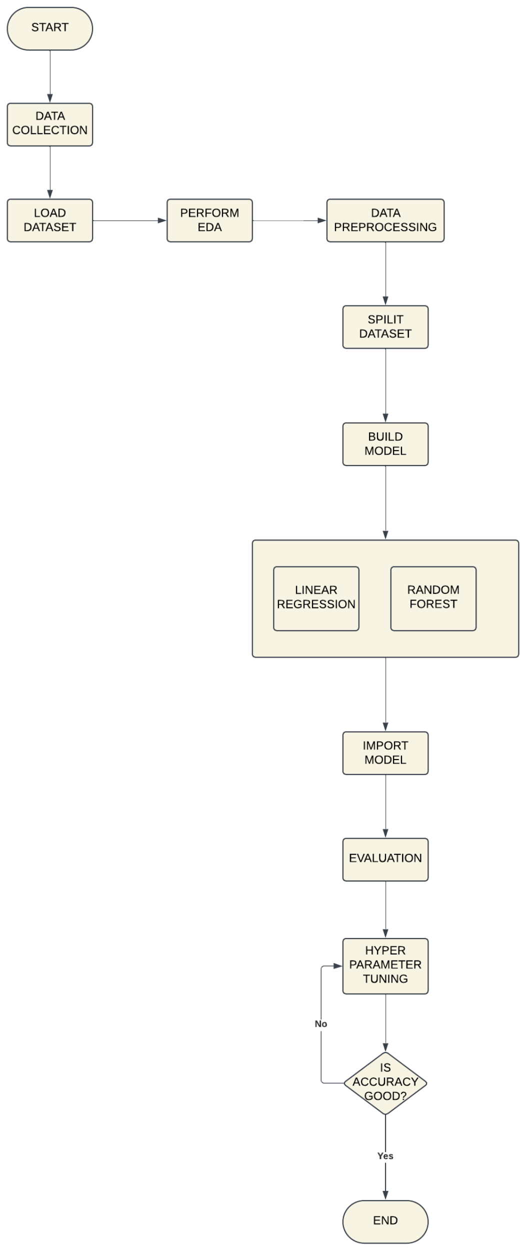


Figure 1:Flowchart of the system

3.1.2. Flowchart For Linear Regression

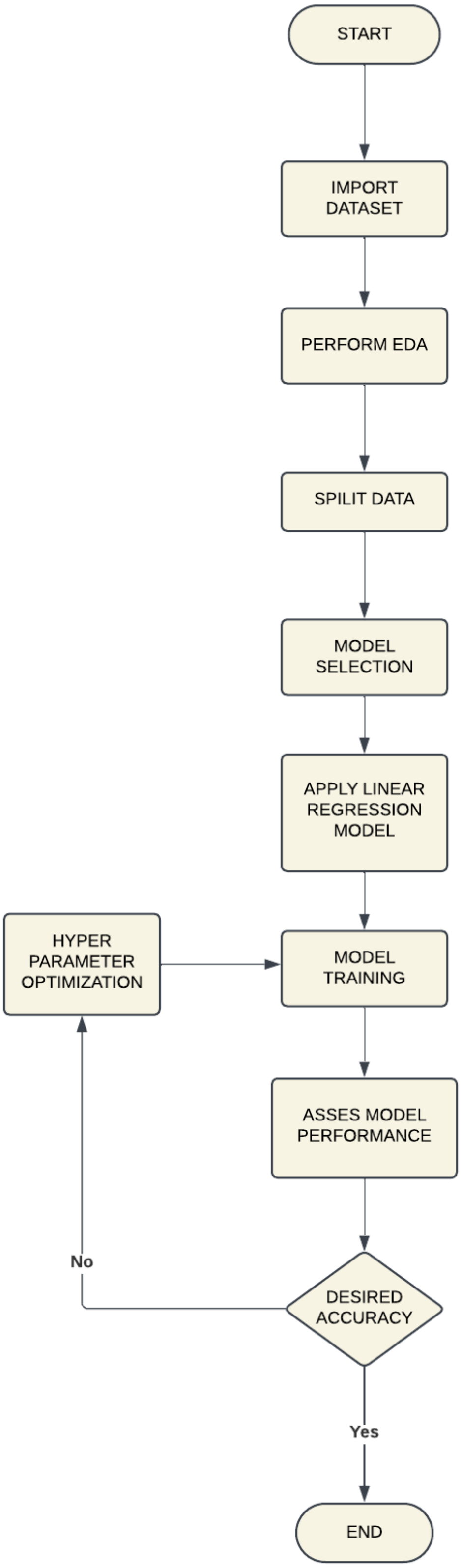


Figure 2:Flowchart of Linear regression

3.1.3. Flowchart For Random Forest

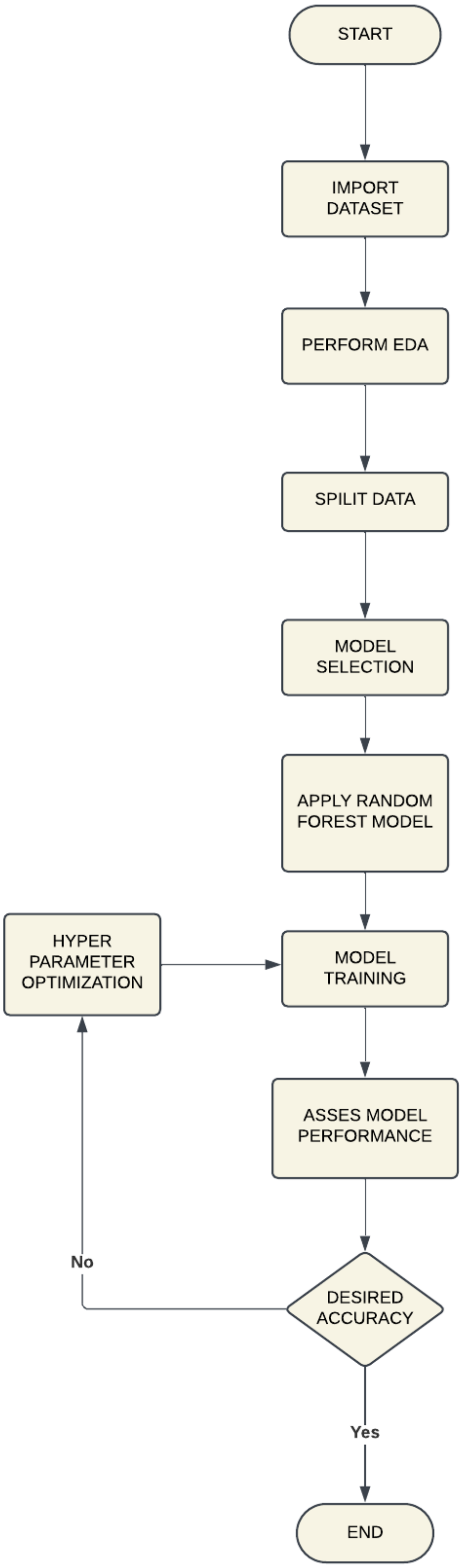


Figure 3:Flowchart for Random forest

3.2. Considered Algorithm

The following is a very brief summary of the considerations involved with the development of the model:

Linear regression: Linear regression establishes a linear relationship between the target variable (house price) and the input features, thus being a good option for an acceptable understanding of the impacts of features and known for computational efficiency with high interpretability, could in such a case prove to be an excellent baseline model for this task. However, its performance might cause it to fail when the target data distinguishes both interaction-type features and non-linearity features.

Why Linear Regression: Simple, easy to interpret; it works quite well when predicting continuous values.

Methodology:

The assumptions include linearity, independence of errors, homoscedasticity, and normal distribution of residuals. The parameters a and b are usually estimated using the least squares method, which minimizes the sum of the squared differences between the actual and predicted values. Model evaluation is made using R-square and other metrics which measure how much variance in the dependent variable is explained by the independent variables. (Qu, 2023)

Mathematical Foundations:

Linear regression can be expressed with a simple formula i.e.

$$Y = a + bX$$

Random Forest: Random Forest, being an ensemble learning method, constructs multiple decision trees and combines their predictions with mode or mean; thereby, Random Forest attempts to increase both the robustness and accuracy in prediction. Being able to handle any interaction of features and nonlinear relationships effectively. This makes Random Forest an extremely enticing option for catching complex patterns in housing data when coupled with the advantage of missing value handling and bagging, which curbs overfitting.

Why Random Forest: It can also handle rather complicated non-linear relations with less overfitting.

Methodology:

The methodology used in the project includes sampling techniques where randomly chosen subsets of training data are bootstrapped to train each tree and create diversity among the trees. At any point where a tree may split a node, a random subset from among the potential features is selected, in a bid to counter correlation among trees and to quell the effect of overfitting. For regression tasks, all predicted values, made by all trees, are to be averaged. (Breiman, 2001)

3.3. Pseudocode Illustrating the Solution Approach

The section on the pseudocode illustrating the solution approach encapsulates the methodology adopted in this project. This is intended to provide a better insight into the logical sequence and the major components in the workflow for solving the problem effectively.

3.3.1. Pseudocode For the Entire System

```
1. Load the dataset.
2. Perform Exploratory Data Analysis (EDA):
   a. Visualize data to understand patterns and trends.
   b. Analyze relationships between features and target variable.
3. Clean the dataset:
   a. Handle missing values.
   b. Remove duplicate or irrelevant data.
4. Perform feature engineering:
   a. Create new meaningful features.
   b. Normalize or scale features if necessary.
5. Split the dataset into training and testing sets.
6. Choose predictive models (e.g., Linear Regression, Random Forest).
7. Train each model using the training set.
8. Evaluate the performance of each model on the test set using evaluation metrics.
9. Compare model performances and select the best-performing model.
10. Deploy the final model for prediction.
11. Visualize insights, such as feature importance and prediction trends.
```

3.3.2. Pseudocode For the Linear Regression

```
1. Import necessary libraries (e.g., sklearn for Linear Regression).
2. Load the cleaned dataset.
3. Split the data into training and testing sets.
4. Initialize the Linear Regression model.
5. Train the model using the training set:
   a. Fit the model to the input features and target variable.
6. Evaluate the model on the test set:
   a. Predict the target variable for test data.
   b. Calculate evaluation metrics (e.g., Mean Squared Error, R-squared).
7. Analyze the results:
   a. Identify strengths and weaknesses of the model.
8. Visualize results, such as actual vs. predicted values.
```

3.3.3. Pseudocode For the Random Forest Regression

```
1. Import necessary libraries (e.g., sklearn for Random Forest).
2. Load the cleaned dataset.
3. Split the data into training and testing sets.
4. Initialize the Random Forest model:
   a. Specify parameters such as number of trees and maximum depth.
5. Train the model using the training set:
   a. Fit the model to the input features and target variable.
6. Evaluate the model on the test set:
   a. Predict the target variable for test data.
   b. Calculate evaluation metrics (e.g., Mean Squared Error, R-squared).
7. Re-evaluate the optimized model on the test set.
8. Analyze and visualize feature importance and results.
```

3.4. Tools and Development Process

The project made use of a variety of tools and libraries, most of which are typical of the workflow of the data science process, for efficient data management, analysis, and visualization. A handful of the most important tools used at various stages of this work are narrated below.

3.4.1. Why this Particular Language?

Python is widely considered to be one of the most appropriate languages for data science, given its extensive support through open-source libraries for scientific computing and machine learning. Some libraries, namely Scikit-learn, TensorFlow, and Keras allow quick prototyping of the ML models. That further affirms the popularity of the language in terms of sufficient community support, regular updates, and in-depth, lucid documentation, all of which contribute to smoothness in model development with comparatively less learning curve for a variety of applications such as classification and reinforcement learning. With the community consistently growing larger, processes such as model development and deployment over a range of applications-from classification to reinforcement-are both smooth and efficient. (Yuan Ren, 2021)

3.4.2. Libraries and Frameworks Employed

The libraries and frameworks that were used in the development of this system included:

- **Scikit-learn:** Scikit-learn is a strong open-source library for Python that offers multiple tools for machine learning like classification, regression, clustering and dimensionality reduction.

- **NumPy:** NumPy is a key library for numerical computing in Python that can handle large multi-dimensional arrays and matrices. It offers a set of math functions to work with these arrays effectively, making it important for scientific computing.
- **Pandas:** Pandas is a library for data manipulation and analysis that provides structures like Data Frames and Series to help manage data. It makes tasks like data cleaning, transformation and analysis easier, which is why many data scientists prefer it.
- **Matplotlib:** Matplotlib is a complete library for making static, animated and interactive visualizations in Python. It offers an easy way to create plots and charts, helping users see data understandings clearly.
- **Seaborn:** Seaborn is built on Matplotlib toolbox and provides a high and attractive visual interface to statistical graphics. It simplifies how complex visualizations can be created by allowing coding that is less code-intensive while sparking more beautiful plots. (Dr. Priyanka Sisodia, 2022)
- **Jupyter Notebook**: It is pen-source web Java app that allows users to create and share documents containing live code, equations, visualizations, and narrative text 5; mostly popular within the data science field for enabling interactive coding and analysis workflow. (Dr. Priyanka Sisodia, 2022)

3.4.3. Development Platform

The project was developed on a local machine with the following specifications:

- Laptop: LAPTOP-2GEPNV74
- RAM: 8.00 GB (7.80 GB usable)
- System type: 64-bit operating system, x64-based processor
- Processor: 11th Gen Intel(R) Core (TM) i7-1165G7 @ 2.80GHz 2.80 GHz

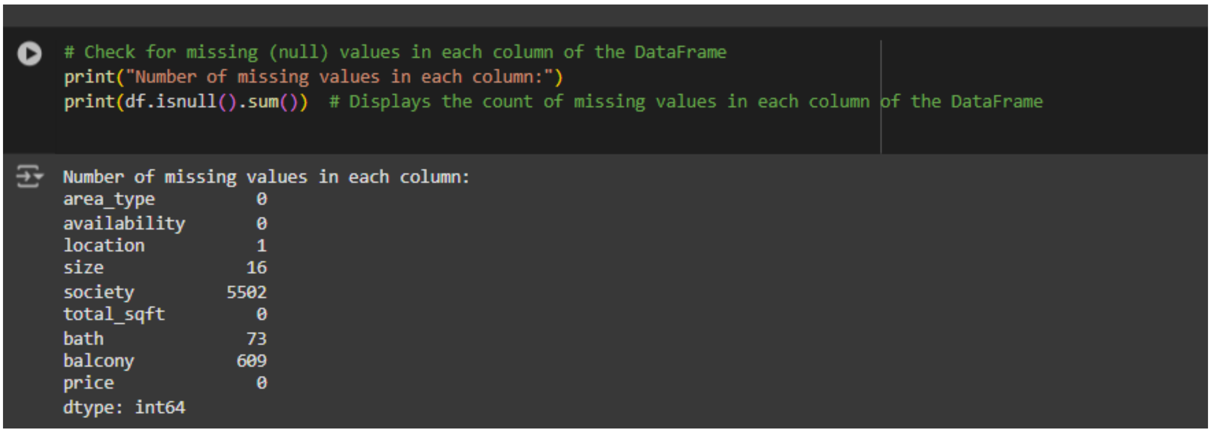
3.5. Data Preprocessing

With the preprocessing stage having almost come to completion and missing data, feature encoding, and scaling being handled successfully, the data has attained consistency and is ready for modelling.

1. **Handling Missing Values**

- **Identifying Missing Values:** The first approach involves identifying the column in dataset which contains missing values by generally using function as isnull (). sum () in pandas,

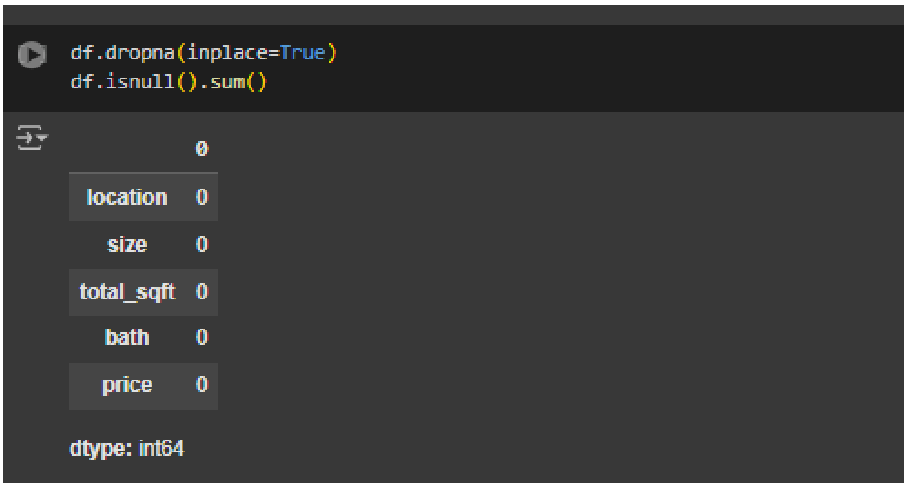
- **Imputation Techniques:** Usually, missing values can be fixed by performing various imputation methods such as replacing them by calculating mean, median or mode in their columns.
- **Dropping Columns:** In some cases, missing values are dropped if the data is not that necessary for the modelling.



```
# Check for missing (null) values in each column of the DataFrame
print("Number of missing values in each column:")
print(df.isnull().sum())  # Displays the count of missing values in each column of the DataFrame
```

```
Number of missing values in each column:
area_type          0
availability       0
location           1
size              16
society         5502
total_sqft         0
bath              73
balcony          609
price              0
dtype: int64
```

Figure 4:Checking for missing values



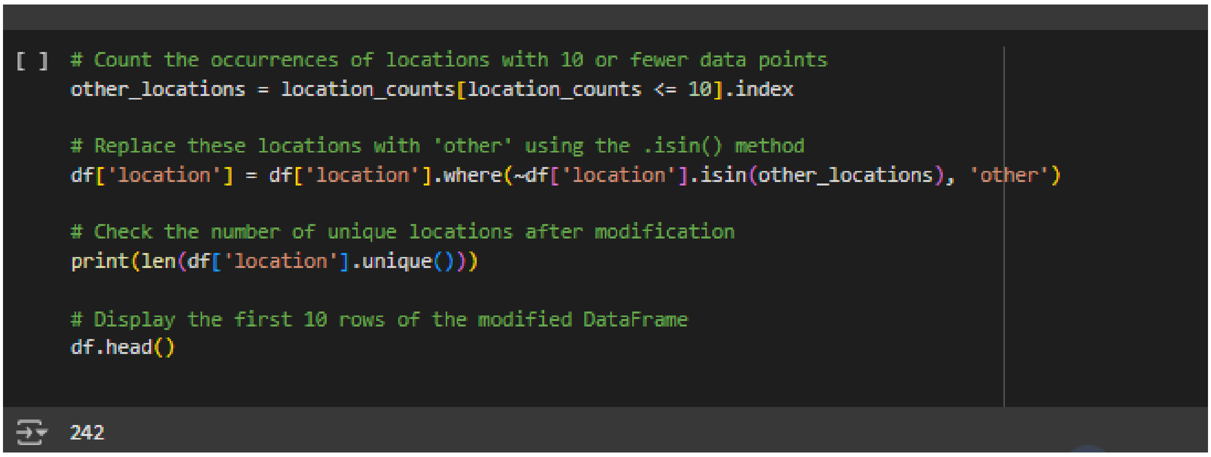
```
df.dropna(inplace=True)
df.isnull().sum()
```

```
              0
 location     0
     size     0
total_sqft    0
     bath     0
    price     0

dtype: int64
```

Figure 5:Dropping the unnecessary values

2. Dimension Reduction

- It is basically the process of reducing the number of features (dimensions) in a data set while keeping as much information as possible. It is particularly useful in enhancing model performance, reducing computational costs, and visualizing high-dimensional data.



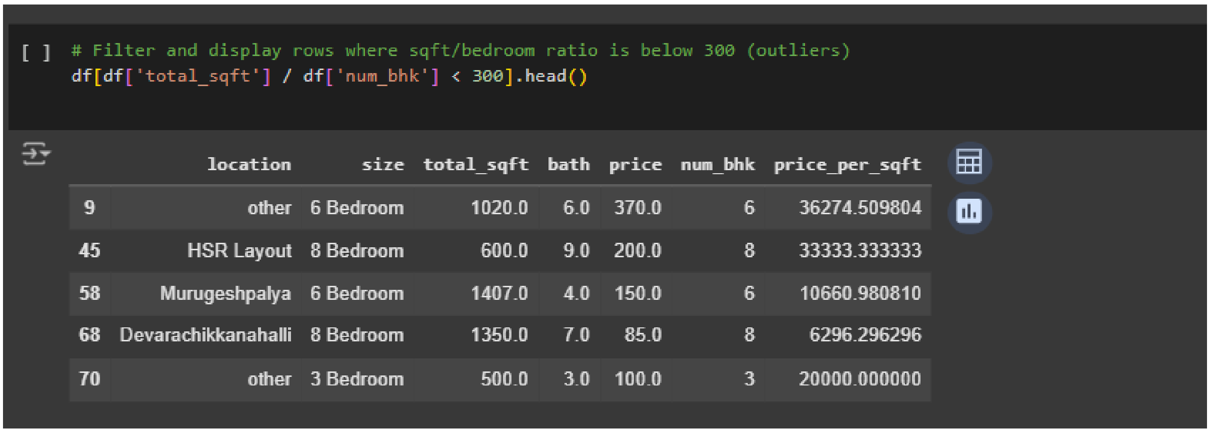
```
[ ] # Count the occurrences of locations with 10 or fewer data points
    other_locations = location_counts[location_counts <= 10].index

    # Replace these locations with 'other' using the .isin() method
    df['location'] = df['location'].where(~df['location'].isin(other_locations), 'other')

    # Check the number of unique locations after modification
    print(len(df['location'].unique()))

    # Display the first 10 rows of the modified DataFrame
    df.head()
```

```
242
```

Figure 6:Performing Dimesion Reductionality

3. Outlier Detection and Handling

- Outlier detection is the task of finding data points that greatly differ from the rest of the dataset. While these outliers might point to variability in measurements or experimental errors, they may also hint at new insights. The identification and moderate handling of outliers are crucial, since they can lead to misguided conclusions by distorting results and altering model fitting.



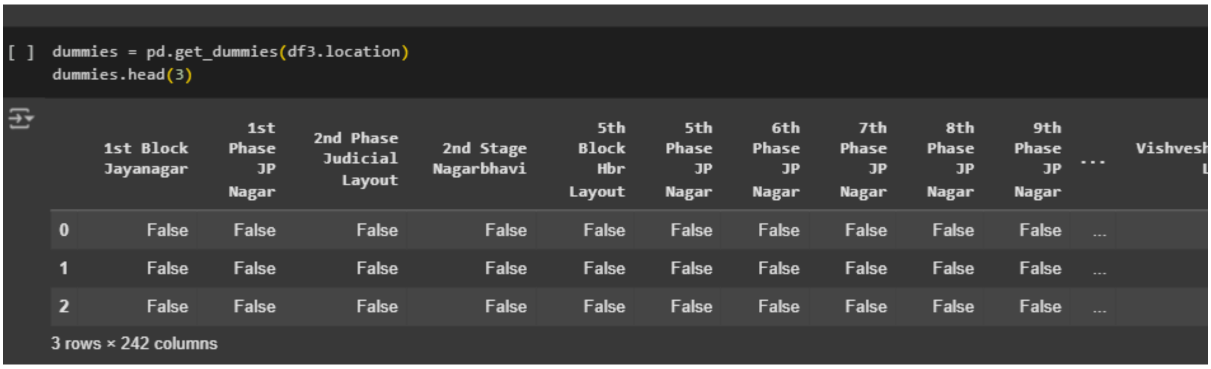
```
[ ]  # Filter and display rows where sqft/bedroom ratio is below 300 (outliers)
     df[df['total_sqft'] / df['num_bhk'] < 300].head()
```

	location	size	total_sqft	bath	price	num_bhk	price_per_sqft
9	other	6 Bedroom	1020.0	6.0	370.0	6	36274.509804
45	HSR Layout	8 Bedroom	600.0	9.0	200.0	8	33333.333333
58	Murugeshpalya	6 Bedroom	1407.0	4.0	150.0	6	10660.980810
68	Devarachikkanahalli	8 Bedroom	1350.0	7.0	85.0	8	6296.296296
70	other	3 Bedroom	500.0	3.0	100.0	3	20000.000000

Figure 7:Outliear Deetection and Handling

4. Performing One Hot Encoding

- In one-hot encoding, categorical variables are converted into a unique numeric form that machine learning algorithms can use. In the case of a feature called "location" within a dataset, one-hot encoding generates binary columns for each unique location, with a 1 denoting that a specific given row contains that location, and 0 indicating its absence.



```
[ ]  dummies = pd.get_dummies(df3.location)
     dummies.head(3)
```

	1st Block Jayanagar	1st Phase JP Nagar	2nd Phase Judicial Layout	2nd Stage Nagarbhavi	5th Block Hbr Layout	5th Phase JP Nagar	6th Phase JP Nagar	7th Phase JP Nagar	8th Phase JP Nagar	9th Phase JP Nagar	...	Vishvesh
0	False	False	False	False	False	False	False	False	False	False	...	
1	False	False	False	False	False	False	False	False	False	False	...	
2	False	False	False	False	False	False	False	False	False	False	...	

3 rows × 242 columns

Figure 8:Perfoming One Hot Encoding

3.6. Selection of Evaluation Metrics

Following a set of extensive analysis, the predictions are made and evaluated on the basis of use of the following set of important metrics.

- **Mean Absolute Error (MAE):** It accounts for only the absolute size of how an estimated with errors deviates from Es.
  Formula:
  $$MAE = \frac{1}{n} |y_i - \hat{y_i}|$$

- **Mean Squared Error (MSE):** The mean value of squared differences from the actual prediction values.
  Formula:
  $$MSE = \frac{1}{n} \left(y_i - \hat{y_i}\right)^2$$

- **Root Mean Squared Error (RMSE):** One way of giving an error measure that keeps the magnitude of error in the same unit as the output because it consists of the squared root of MSE.
  Formula:
  $$RMSE = \sqrt{MSE}$$

- **R-squared (R^2):** An indicator of how much proportion of variance expressed in the dependent variable can be completed from the selected independent variables.

3.7. Justification for Using Metrics

The metrics have been used for the following reason(s):

- **R-squared:** R-squared tells what proportion of variance in the dependent variable is caused by changes in independent variables in a regression model.
- **MAE (Mean Absolute Error):** MAE calculates the average of absolute differences between the predicted and actual values, which provides a simple understanding of prediction accuracy

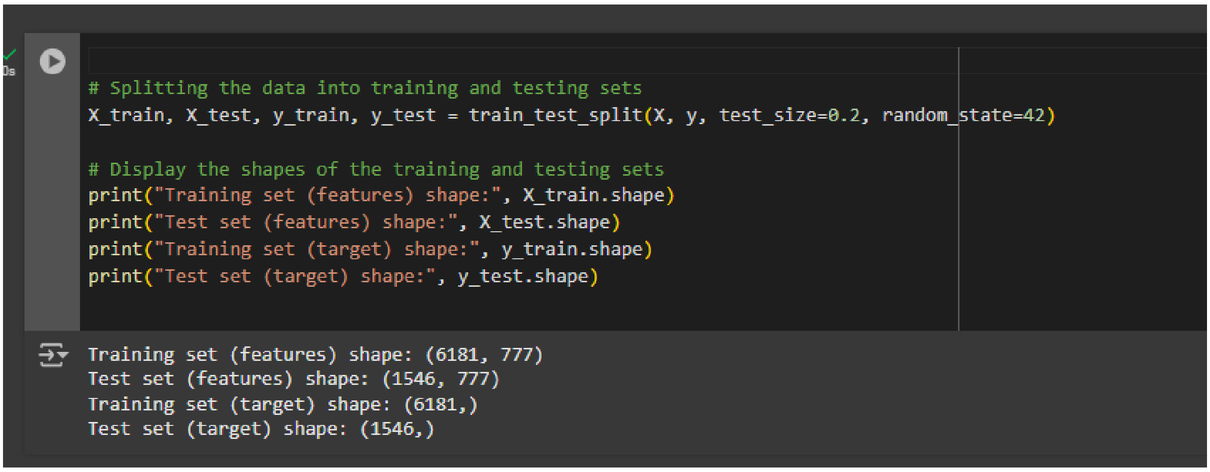
- **MSE (Mean Squared Error):** MSE is the average of the squared differences between the predicted and actual values, which puts more weight on larger errors because of squaring
- **RMSE:** RMSE is calculated as MSE square root; it is the standard deviation of the prediction errors.

4. Result

The present analysis discusses building a prediction model that will try to estimate the price of properties using a set of features present in the dataset. The input is being processed-the input data represents attribute fields like area type, availability, location, size, society, built-up area, number of bathrooms, and balconies. This preprocessed information is then cleaned and provided in an encoded format. The model is trained on a subset of the data so it can learn how these features correlate with the target variable.

4.1. Training and Running Model

The process of training and running the model consists of several key steps. First, the dataset is divided using the train_test_split function into a training control set and a testing set so that one part of the data set will serve as a training set and the other part to validate the model. The Regression model is then instantiated and trained on the training set, teaching the model how to relate the features to the target variable.



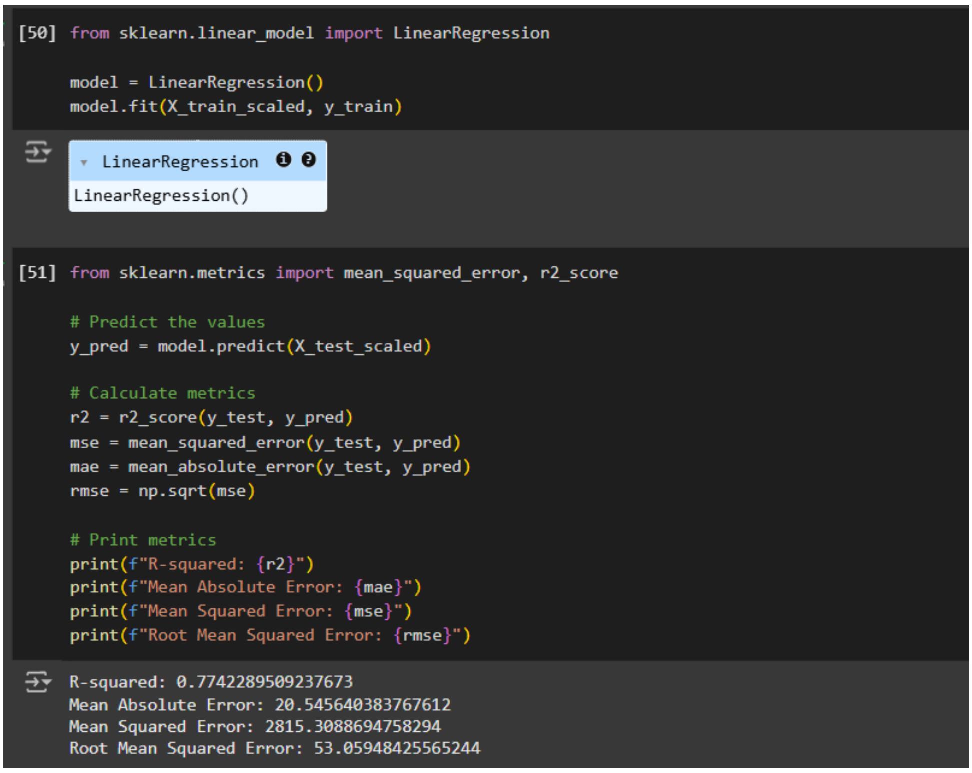
```
# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Display the shapes of the training and testing sets
print("Training set (features) shape:", X_train.shape)
print("Test set (features) shape:", X_test.shape)
print("Training set (target) shape:", y_train.shape)
print("Test set (target) shape:", y_test.shape)
```

```
Training set (features) shape: (6181, 777)
Test set (features) shape: (1546, 777)
Training set (target) shape: (6181,)
Test set (target) shape: (1546,)
```

Figure 9:Training The Model

After we complete splitting the data, our next step is to work with the model one at a time, and check which model can give us good score:

- Linear Regression



```
[50] from sklearn.linear_model import LinearRegression

     model = LinearRegression()
     model.fit(X_train_scaled, y_train)
```

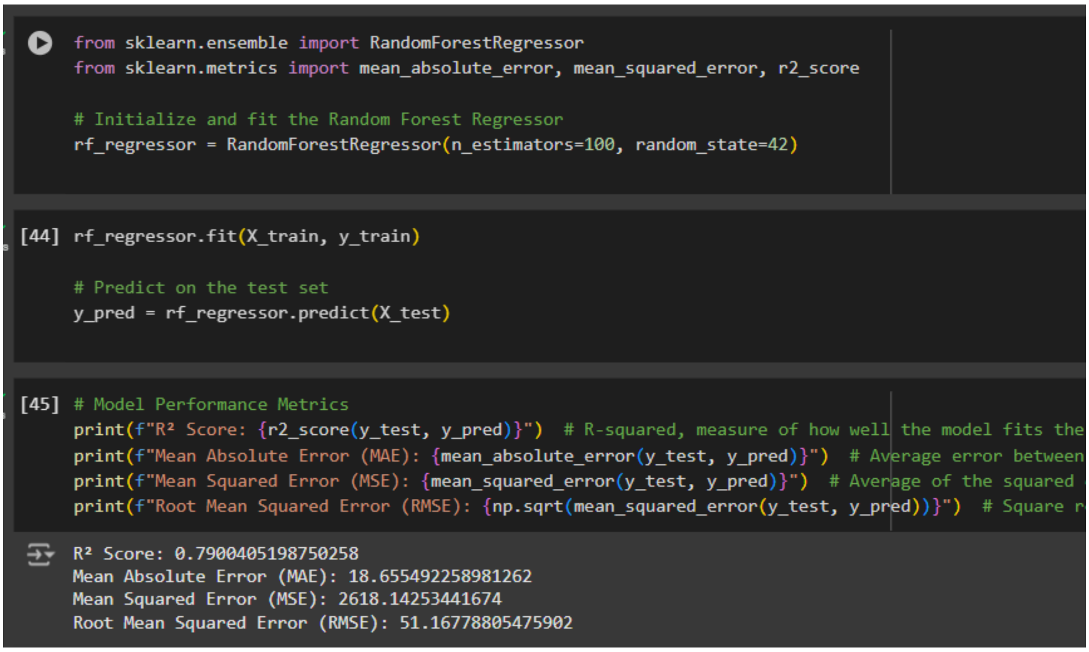
```
LinearRegression
LinearRegression()
```

```
[51] from sklearn.metrics import mean_squared_error, r2_score

     # Predict the values
     y_pred = model.predict(X_test_scaled)

     # Calculate metrics
     r2 = r2_score(y_test, y_pred)
     mse = mean_squared_error(y_test, y_pred)
     mae = mean_absolute_error(y_test, y_pred)
     rmse = np.sqrt(mse)

     # Print metrics
     print(f"R-squared: {r2}")
     print(f"Mean Absolute Error: {mae}")
     print(f"Mean Squared Error: {mse}")
     print(f"Root Mean Squared Error: {rmse}")
```

```
R-squared: 0.7742289509237673
Mean Absolute Error: 20.545640383767612
Mean Squared Error: 2815.3088694758294
Root Mean Squared Error: 53.05948425565244
```

Figure 10: Linear Regression Model

- Random Forest



```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Initialize and fit the Random Forest Regressor
rf_regressor = RandomForestRegressor(n_estimators=100, random_state=42)
```

```
[44] rf_regressor.fit(X_train, y_train)

     # Predict on the test set
     y_pred = rf_regressor.predict(X_test)
```

```
[45] # Model Performance Metrics
     print(f"R² Score: {r2_score(y_test, y_pred)}")  # R-squared, measure of how well the model fits the
     print(f"Mean Absolute Error (MAE): {mean_absolute_error(y_test, y_pred)}")  # Average error between
     print(f"Mean Squared Error (MSE): {mean_squared_error(y_test, y_pred)}")  # Average of the squared
     print(f"Root Mean Squared Error (RMSE): {np.sqrt(mean_squared_error(y_test, y_pred))}")  # Square r
```

```
R² Score: 0.7900405198750258
Mean Absolute Error (MAE): 18.655492258981262
Mean Squared Error (MSE): 2618.14253441674
Root Mean Squared Error (RMSE): 51.16778805475902
```

Figure 11:Random Forest Model

4.2. Model Prediction and Output

Model prediction utilizes historical data to project future, based on regression and classification techniques. This entails data collection and preparation, model selection, and validation. It aims to fit a model to patterns that can make accurate predictions for decision-making purposes. With accurate handles on model predictions, operational efficiency and strategic planning in various spheres will greatly benefit.

- **Scatterplot of Predicted vs Actual Price**

  The scatterplot of actual and predicted prices gives a visual representation of the relationship between the two variables. Each point in the plot indicates a predicted price against the actual price, enabling a rough check on the accuracy of the model.

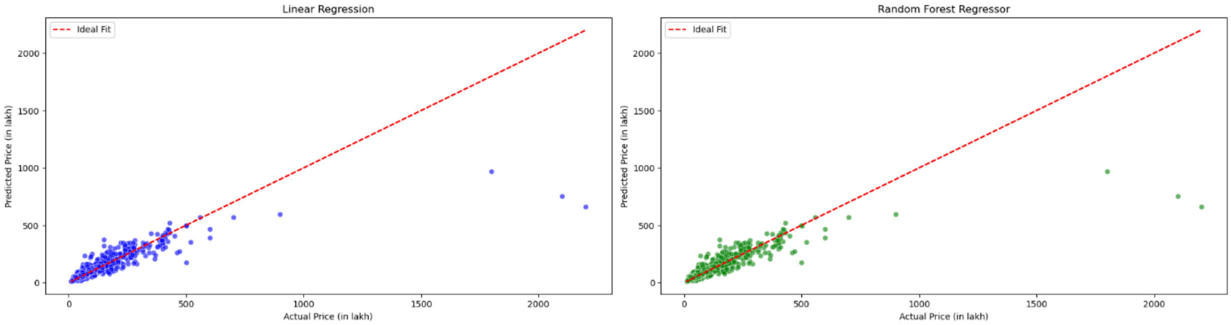


Figure 12:Scatter plot

- **Residual Plot**

  A residual plot has residuals on the vertical axis and the independent variable on the horizontal axis. Ideally, residuals have a random scatter about zero, indicating a good fit of the model. Patterns in residuals indicate further potential concerns, such as non-linearity or outliers, which hint at using a different modelling approach.

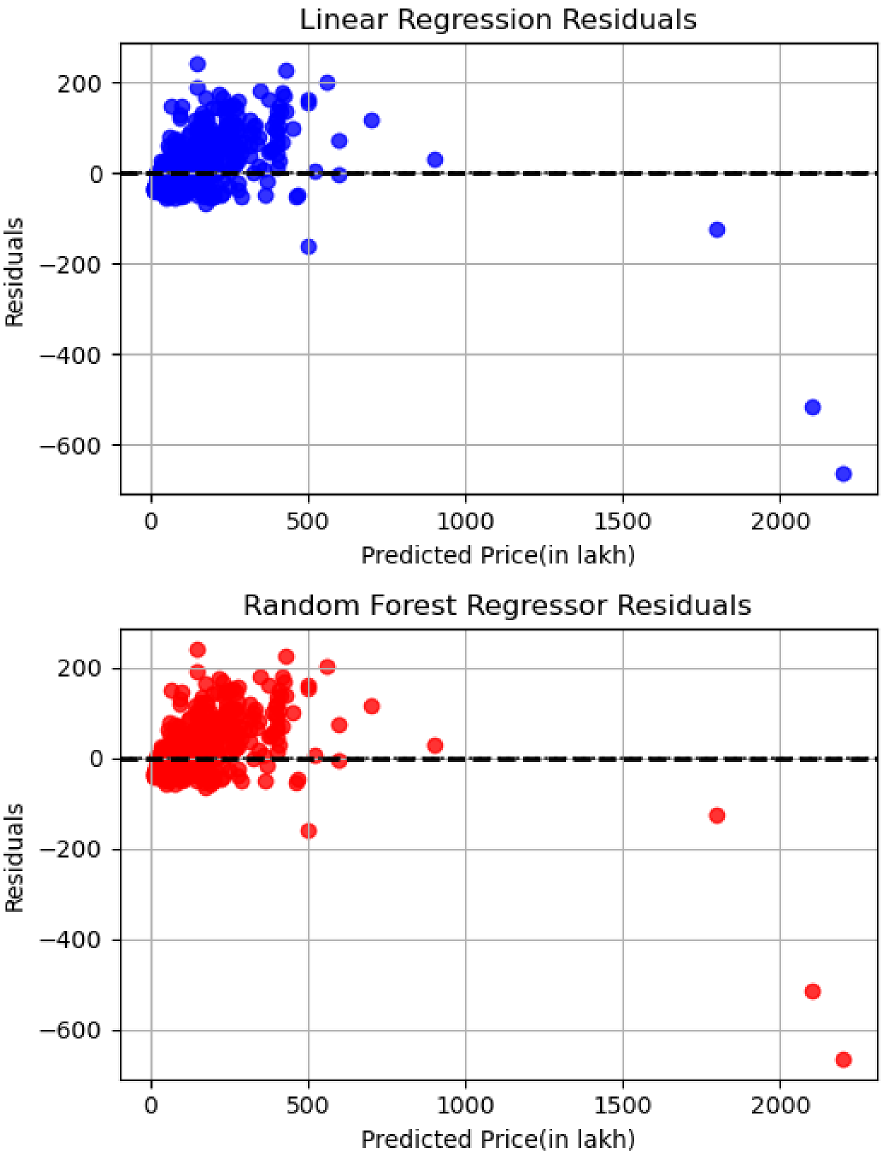


Figure 13:Residuals Plot

4.3. Comparison Of Model Performance

Upon evaluating the results of models, it was observed that the Linear Regression model demonstrated the highest accuracy among the options tested. Additionally, further refinement and testing are necessary to enhance the model's reliability and ensure it generalizes well to unseen data. While Linear Regression shows promise, these improvements are essential for achieving optimal performance.

MODEL	R-squared	Mean Absolute Error	Mean Squared Error	Root Mean Squared Error
Linear Regression	0.83	19.27	2332.78	48.29
Random Forest Regressor	0.69	20.96	4439.12	66.62

Table 2: Comparsion table

5. Conclusion

This project will allow for making a good demonstration of the applicability of machine learning techniques for predicting house prices in Bengaluru on tenanted property features using a well-designed pre-processing pipeline. All models are evaluated based on advantages and disadvantages for finding the one that suits best for house pricing methods according to their accuracy. The changing nature of the housing sector and intricate factors that influence it hint at the necessity of using data-driven solutions. The current study is about developing predictive models that are accurate and reliable enough to provide buyers, sellers, or investors with the clear insight needed into a rapidly developing housing market.

5.1. General Analysis of Work

In reviewing the analysis work conducted, we see that they were able to implement several models of regression, among them the Linear Regression, and the Random Forest Regressor. The preprocessing pipeline had enhanced the data quality and consistency to such a high degree as to lend a great deal of improvement to such predictions as a whole. Visualizations in the form of scatter plots and residual plots further opened up a clear window into how models were performing and the areas for further improvement for these models.

5.2. Model Effectiveness.

Model effectiveness as measured by R-squared, MAE, MSE, and RMSE is evaluated by traditional models. Linear Regression demonstrated comparatively better accuracy among the ones implemented with respect to each compared method very likely due to its strength in capturing complex relationships in the data. However, trade-offs exist within the context of competing models, based on model selection for a particular use case and availability of data.

The project aims to solve practical problems by providing viable solutions toward the various challenges faced by the institutions in the real estate market. These models for prediction can serve as tools in which stakeholders are enabled to make informed decisions concerning property evaluations and articulate the housing price trend situations. Such insights become quite core in dynamic markets like Bengaluru, whereby timely and accurate information can make a major difference.

5.3. Future Work

Based on this research study, further recommendations may be made to include more features in the analytical process. The addition of features such as infrastructure development, neighborhood crime rates, and economic indicators is recommended to gain optimal predictive accuracy. Such advanced methods as ensemble methods or deep learning models can also give the system a major boost in performance. Making the application either a web or mobile application would further give it an edge.

Bibliography

Amey Thakur1, M. S., 2021. BANGALORE HOUSE PRICE PREDICTION. *International Research Journal of Engineering and Technology (IRJET),* 08(09), p. 4.

Anon., n.d.

Breiman, L., 2001. *RANDOM FORESTS,* Berkeley: Statistics Department University of California.

Dr. Priyanka Sisodia, D. B. S., 2022. An Implementation on Python for Data Science. *International Journal Of Creative Research Thoughts,* 10(3), pp. a90-a907.

Kumari Sandhya1, S. S., 2022. House Price Prediction Using Machine Learning. *International Journal for Research in Applied Science & Engineering Technology (IJRASET),* 10(V), p. 6.

Madhuri, C. R. a. A. G. a. P. M. V., 2019. 2019 International Conference on Smart Structures and Systems (ICSSS). *House Price Prediction Using Regression Techniques: A Comparative Study,* pp. 1-5.

Manasa, J. a. G. R. a. N. N. S., 2020. Machine Learning based Predicting House Prices using Regression Techniques. *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA),* p. 630.

Meghna Chandel, 2022. A Study on Machine Learning and Python's Framework. *International Journal of Computer Sciences and Engineering,* 10(5), pp. 58-64.

Qu, K., 2023. *Research on linear regression algorithm.* China, Shandong Xiehe University, 250100, JiNan, Shandong, China.

Sandhya1, K., 2022. House Price Prediction Using Machine Learning. *International Journal for Research in Applied Science & Engineering Technology (IJRASET),* 10(V), p. 6.

Shinde, P. P. a. S. S., 2018. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). *A Review of Machine Learning and Deep Learning Applications,* pp. 1-6.

Thakur, A. a. S. M., 2021. Bangalore House Price Prediction. *Department of Computer Engineering, University of Mumbai, Mumbai, MH, India,* 8(9), pp. 193-196.

Yuan Ren, S. D. U. S. C., 2021. Python Machine Learning: Machine Learning and Deep Learning With Python,Scikit-Learn, and TensorFlow 2, Third Edition. *International Journal of Knowledge-Based Organization,* 11(1), p. 770.