Prediction of housing prices in Boston

Using Machine learning algorithms

Project Report
Rabia Danish
Student Number: 501210698
Supervisor's Name: Professor Dr. Ceni Babaoglu
Date of submission: 17th July 2023

Ryerson University

Table of Contents

Abstract

In today's ever challenging world of real estate, where housing prices are constantly on the rise, it has become increasingly important for individuals to make well-informed decisions when it comes to property investments. Our research uses available "Housing Prices Dataset" for Boston. By harnessing the power of this extensive dataset, our goal is to unravel the intricate factors that influence housing prices and provide individuals with the knowledge necessary to navigate the market with confidence and foresight. This will allow to determine future prices based on the available features which will enable people to make reliable decisions. Our dataset consists of various features related to the properties, including price, area, number of bedrooms, bathrooms, stories, and other factors that may influence housing prices. The objective of this project is to develop precise prediction models that can assist in estimating housing prices.

We aim in conducting exploratory data analysis to gain valuable insights into the dataset, identifying patterns, and understanding the relationships between the features and the target variable (price). Data preprocessing techniques such as handling missing values, feature scaling, feature selection, and encoding categorical variables are applied to ensure the data is in a suitable format for analysis.

In this study, we will employ several machine learning techniques, including linear regression, decision tree regression, and random forest regression, to develop predictive models for housing price estimation using cross Validation technique. These models will consider input features such as the number of bedrooms, bathrooms, stories, and the area of the house, aiming to capture the underlying linear relationship with the target variable, price. To assess the performance of these models, we will utilize commonly used evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) and R2 coefficient.

By comparing the performance of these models using these evaluation metrics, we will gain insights into their accuracy and reliability in predicting housing prices.

Additionally, correlation analysis is conducted to examine the relationships among the different attributes of houses using python. By determining the degree of correlation between variables, we can identify highly correlated attributes and assess their impact on housing prices. This analysis helps address the issue of multicollinearity, where highly correlated attributes can introduce redundancy and distort the accuracy of the prediction models. By identifying the most influential attributes, we can refine the models and focus on the features that have a significant impact on housing prices.

Overall, this data analytics capstone project aims to use the provided dataset to develop accurate prediction models for housing prices by considering different features which may impact those prices, therefore, enabling stakeholders to make informed decisions regarding real estate investments, pricing policies, and market analysis.

Data set:

The Project aims to use data to achieve the goal:

https://www.kaggle.com/datasets/yasserh/housing-prices-dataset

Introduction

In the ever-changing real estate market, accurate prediction of housing prices is essential for making informed decisions regarding property investments.

The literature review section delves into the significance of predicting house prices and its practical implications. As house prices continue to rise, accurate prediction models are essential for various stakeholders such as landowners, valuers, and policymakers. These models enable them to determine property valuations and appropriate sale prices, assisting potential buyers in making informed decisions. While physical conditions, architectural styles, and location are known to impact housing prices, the specific variables influencing prices can vary across different regions. Therefore, an effective prediction model must consider and accommodate the unique variables that drive housing prices in the specific area under consideration.

A number of scholars have conducted extensive research in predicting house prices. This literature review aims to explore existing research articles that focus on housing price prediction and how they can contribute to the research project based on the provided dataset. The articles mentioned provide insights into the application of regression models and machine learning techniques, such as Random Forest and multiple linear regression, for housing price prediction.

Literature Review

The study conducted by Adetunji et al., employed a regression model to analyze the Boston housing dataset with the goal of predicting house prices based on the dataset's features. The proposed approach involved developing a prediction model using the Random Forest algorithm. The UCI Machine Learning Repository's Boston housing dataset, consisting of 506 entries and 14 features [1], was used to evaluate the performance of the proposed model.

The study focused on exploring the effectiveness of the Random Forest machine learning technique for house price prediction. The Random Forest algorithm is a classifier that leverages multiple decision trees trained on different subsets of the dataset to enhance predictive accuracy. By aggregating the outcomes of these trees, the algorithm determines the final prediction based on the majority vote.

With the Random Forest algorithm at the core, the authors proceed to train the prediction model and meticulously assess its performance using well-established metrics like mean absolute error, mean squared error, and R-squared. A comparison between the predicted prices and the actual prices revealed that the model's predictions were within an acceptable range, with an error margin of ±5. This suggests that the model has the potential to provide reliable predictions for house prices.

Overall, this research highlights the successful application of the Random Forest machine learning technique in house price prediction. The findings demonstrate the model's ability to effectively analyze the Boston housing dataset and generate accurate predictions, making it a promising approach for real estate valuation and decision-making processes [1].
Similarly, the study by Truong et al., focused on the comparative analysis of different machine learning models for housing price prediction using the "Housing Price in Beijing" dataset. The study evaluated the performance of Random Forest, XGBoost, LightGBM, Hybrid Regression, and Stacked Generalization Regression models and analyzes their strengths and weaknesses.

The dataset comprises over 300,000 housing transactions with 26 variables as features for predicting the average price per square meter [2]. The evaluation metric used is the Root Mean Squared Logarithmic Error (RMSLE).

Results indicate that Random Forest performs well on the training set due to its suitability for datasets with Boolean features. However, it is prone to overfitting. XGBoost and LightGBM exhibit lower accuracy but do not suffer from overfitting. Hybrid Regression and Stacked Generalization Regression deliver promising results without extensive tuning. Hybrid Regression performs best on the training set, while Stacked Generalization Regression excels on the test set.

Stacked Generalization Regression's superior performance can be attributed to K-fold cross-validation and the coupling effect of multiple regressions. Hybrid Regression averages predictions, while Stacked Generalization Regression uses a second stacking level for more accurate predictions based on pre-estimated prices.

Overall the study found that different models had their pros and cons. Random Forest had the lowest error on the training set but was prone to overfitting and had high time complexity. XGBoost and LightGBM had decent accuracy and lower time complexity. Hybrid Regression performed better than the previous methods due to its generalization capability. Stacked Generalization Regression had the most complicated architecture but offered the highest accuracy. Time complexity should be considered, especially since both Hybrid Regression and Stacked Generalization Regression included Random Forest [2].

By utilizing multiple linear regression and Spearman correlation analysis, the study conducted by Zhang et al., aimed to predict housing prices using a multiple linear regression model and the Spearman correlation coefficient to identify influential factors. The analysis was conducted on the Boston housing price dataset. The results showed that the multiple linear regression model was able to effectively predict and analyze housing prices to some extent.

The Spearman correlation coefficient analysis revealed several significant factors influencing housing prices, including the proportion of lower-income groups in the region, the proportion of

property land area larger than 25,000 square feet, and the average number of rooms. These factors played a crucial role in shaping housing prices.

The model training process involved optimizing a gradient descent optimizer to calculate the model parameters and minimize the loss function. After training for 100 epochs, the model's prediction results were generally consistent with the real values in the comparison set, indicating its capability to capture the overall trend of housing prices. Subsequently, training for 500 epochs further improved the model's performance.

The validation results confirmed the accuracy and practicality of the empirical model constructed using the Boston housing price dataset. However, it was acknowledged that the prediction accuracy was still limited in certain cases. The study concluded that while the multiple linear regression model was effective to some extent, further research was needed to enhance its universality and explore advanced machine learning methodologies for housing price prediction.

Overall, the results demonstrated the capability of the multiple linear regression model to predict and analyze housing prices. The influential factors identified through the Spearman correlation coefficient analysis provided valuable insights into the housing market. Nonetheless, there is room for improvement to enhance the prediction accuracy and develop more robust algorithms using advanced machine learning techniques [3].

The study by Zhang et al., presented a methodology for predicting housing prices using a multiple linear regression model and the Spearman correlation coefficient. While the results showed some effectiveness in predicting housing prices, there are several limitation to this study and further research needs to be conducted. Firstly, the study relied solely on a multiple linear regression model, which limited to capture complex nonlinear relationships in the housing market. To enhance the predictive accuracy, the study could benefit from exploring more

advanced machine learning algorithms, such as decision trees, random forests, or neural networks. These algorithms have the potential to capture nonlinearities and interactions among variables, leading to improved predictions.

Additionally, the study primarily relied on the Spearman correlation coefficient to identify influential factors. While correlation analysis is valuable, it is important to consider other statistical techniques, such as feature selection algorithms or dimensionality reduction methods, to identify the most relevant and informative features for predicting housing prices. This would help refine the model and potentially improve its predictive performance.

Furthermore, the study could benefit from conducting a more comprehensive evaluation of the predictive model. This could involve using different evaluation metrics, such as mean absolute error or R-squared, to assess the model's performance and compare it with alternative approaches. It would provide a more robust assessment of the model's predictive capabilities.

In conclusion, while the study [3] presented a methodology for predicting housing prices, there are opportunities for improving the predictive model based on the collected dataset including:

1. Feature Selection: Identify the most significant factors influencing housing prices by calculating correlation coefficients or using other feature selection techniques. This will help in selecting the most relevant features for the model.
2. Nonlinear Relationships: Explore the possibility of nonlinear relationships between the independent variables and housing prices using scatter plots and correlation coefficients.
3. Advanced Regression Techniques: Instead of solely relying on multiple linear regression, consider using decision trees or random forests. These algorithms can capture nonlinear

relationships and interactions among variables, potentially improving predictive accuracy.

4. Model Evaluation: Evaluate the model using appropriate metrics such as mean absolute error (MAE) or root mean squared error (RMSE). Compare the performance of different models and select the one that provides the best balance between accuracy and generalizability.

Therefore, by considering the relevant approaches, we can enhance the predictive accuracy and interpretability of the model when analyzing the given housing price dataset.

The study conducted in Taiwan [4], focused on house price prediction and compared three predictive systems: boosting ensemble regression trees (BERT), support vector regression (SVR), and Gaussian process regression (GPR). These systems were optimized using Bayesian optimization (BO) to improve their performance. BERT, implemented with the least squares boosting algorithm, showed the lowest error rate median and low error variability, indicating stability and high accuracy. SVR had the highest error median, suggesting larger deviations in predicted and actual prices, while GPR exhibited significant variations in prediction accuracy. The results demonstrated the superior performance of BERT in terms of stability and accuracy. Overall, BERT was deemed a highly accurate and stable predictive system for house price forecasting, particularly suitable for small data samples. The study contributes valuable insights for real estate professionals, investors, and analysts in the field of house price prediction [4]. The application of machine learning algorithms for house price prediction in a small town in India emphasized the need for accurate models to aid customers in finding suitable houses [5]. The study incorporates various house attributes, such as the number of bedrooms, age, proximity to amenities, and nearby schools, into the prediction model. Fundamental machine learning

algorithms, including decision tree classification, decision tree regression, and multiple linear regression, are implemented using the Scikit-Learn tool. The research develops a model that predicts house availability and estimates prices, comparing decision tree regression and multiple linear regression. The findings show that multiple linear regression performs better for house price prediction. The study highlights the potential of machine learning algorithms for accurate house price prediction and suggests expanding the dataset and exploring advanced techniques for further improvement [5].

The study conducted by Ho et al., explored the use of machine learning algorithms, including support vector machine (SVM), random forest (RF), and gradient boosting machine (GBM), for predicting property prices in Hong Kong. Using a dataset spanning 18 years and over 40,000 housing transactions, the study compared the performance of these algorithms. The results show that RF and GBM outperform SVM in terms of predictive power, achieving lower mean squared error (MSE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). However, SVM remains valuable for producing accurate predictions within tight time constraints. The study also highlights the influential role of housing attributes such as floor area, property age, and proximity to the Central District on property prices. Overall, the research demonstrates the potential of advanced machine learning techniques in property price prediction, with RF and GBM proving to be effective for accurate estimations. The study suggests careful feature selection and acknowledges the interpretability challenges associated with machine learning models. Additionally, it emphasizes the importance of considering computation time, recommending SVM for quick forecasts and RF/GBM when high predictive accuracy is desired [6].

Research Questions

How can feature selection techniques, including correlation analysis, be employed to identify the key factors influencing housing prices? In addition to multiple linear regression, how do advanced regression algorithms such as random forest and decision tree regression contribute to improving the predictive accuracy of the housing price model?

How can the performance of different models be evaluated and compared using appropriate metrics such as mean absolute error (MAE) , Mean Squared error (MSE), root mean squared error (RMSE) and R2 squared?

Discussion

Our study revolves around the application of machine learning algorithms for predicting house or property prices. Various algorithms, such as Random Forest, XGBoost, LightGBM, Hybrid Regression, Stacked Generalization Regression, Support Vector Regression (SVR), Gaussian Process Regression (GPR), and multiple linear regression, have been explored and compared in different contexts and datasets before. Our study aims to improve predictive accuracy and provide valuable insights into the factors influencing property prices.

Previous studies highlighted the need to consider several critical points. Although, machine learning algorithms show promise in predicting house prices, there are limitations and challenges. The choice of algorithm can impact predictive accuracy, with different algorithms performing better in specific contexts. Overfitting, generalizability, interpretability, and computation time are important factors to consider. Furthermore, the studies often focus on specific datasets or regions, which may limit the generalizability of the findings. There is a need for further research to explore advanced techniques, feature selection methods, ensemble methods, and larger or more diverse datasets.

None of the previous researches had considered the same housing attributes. Each article focuses on different datasets, regions, and machine learning algorithms. While there may be similarities in terms of the use of machine learning for house price prediction, the specific approaches, algorithms employed, and datasets used vary among the studies.

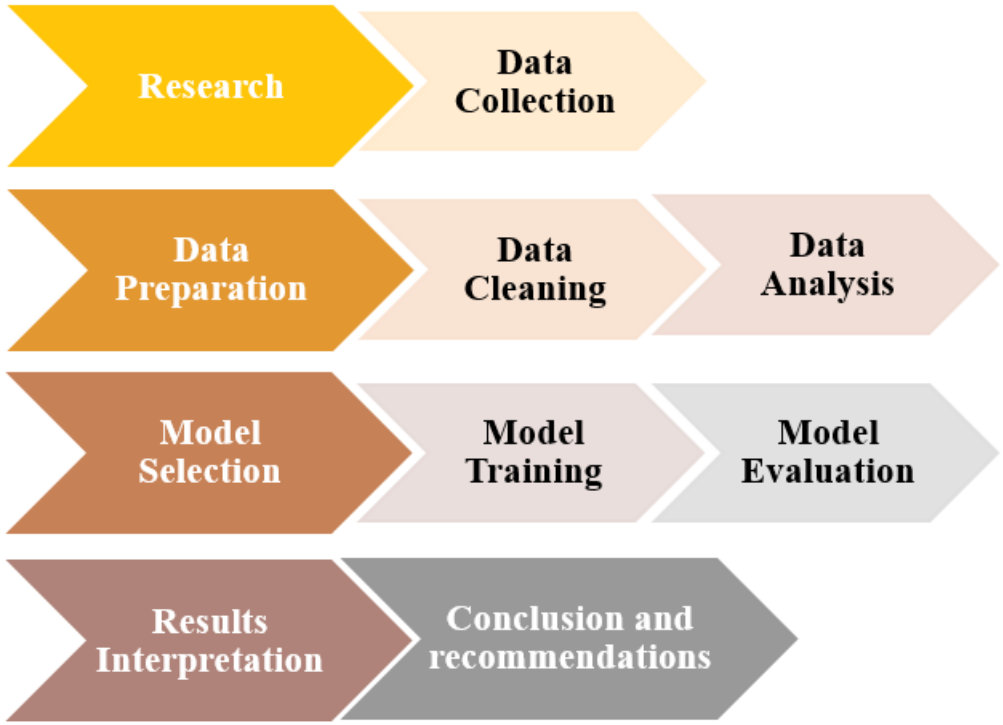
Previous studies also compared and evaluated the performance of different algorithms or techniques on various datasets. Some studies also examined the impact of housing attributes or factors on property prices. Our research provides a foundation for further exploration and improvement in predicting property prices.

Our study builds upon the limitations highlighted in the study by Zhang et al., by incorporating advanced techniques to enhance the predictive model for housing prices. The study suggests feature selection techniques, such as calculating correlation coefficients, to identify the most significant factors influencing housing prices. Additionally, our study uses scatter plots to explore the relationships between independent variables and housing prices, looking for non-linear patterns or trends. Correlation analysis was performed to quantify the strength of the linear relationship between each independent variable and housing prices. This analysis provided insights into the presence of linear or non-linear relationships in the data. Moreover, the project goes beyond multiple linear regression by considering advanced regression algorithms, including random forest and Decision tree regression. By incorporating these techniques, our research aims to improve the model's predictive accuracy. Furthermore, the project emphasizes the importance of evaluating the model using appropriate metrics such as mean absolute error (MAE) or root mean squared error (RMSE) and comparing the performance of different models. By selecting the model that provides the best balance between accuracy and generalizability, the project enhances the methodology used in [3]. Overall, our aim is to build upon existing research by

incorporating these suggestions and improving the predictive accuracy and interpretability of the model for housing prices using advanced techniques, including random forest, linear regression, Decision tree regression and comprehensive model evaluation.

Our study builds upon the models and offers stakeholders in the real estate market, such as professionals, investors, and analysts, more reliable tools for making informed decisions.

Methodology



Research: Conduct background research on housing pricing prediction, exploring relevant literature, and understanding the methodologies employed in similar projects.

Data Collection: Download the housing pricing dataset from Kaggle Housing Prices Dataset.

Data Preparation: The dataset is preprocessed by checking for missing values and duplicates. In this case, the data does not contain any missing values or duplicates. Additionally, feature scaling, feature selection and encoding of categorical variables are performed to ensure that the data is in a suitable format for further analysis.

Data Analysis: Perform exploratory data analysis (EDA) to gain insights into the dataset, identify patterns, and understand the relationships between features and the target variable. Conduct correlation analysis to examine the relationships among different attributes of houses, identify highly correlated features, and assess their impact on housing prices.

Model Selection: Machine learning techniques linear regression, decision tree regression, and Random Forest is used to develop predictive models for housing price estimation.

Model Training: The prepared dataset is used to train the models using three machine learning algorithms (Linear Regression, Decision Tree, and Random Forest). The 10-fold cross-validation technique is employed to capture the underlying linear relationship between the features and the target variable, ensuring robust model evaluation and performance estimation.

Model Evaluation: Evaluate the trained models using commonly used evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R2 coefficient to assess their performance and accuracy in predicting housing prices.

Results Interpretation: The results of the analysis are interpreted and conclusions are drawn based on the findings. This may involve visualizing the data and model outputs using scatter plots and charts.

Conclusion and recommendations: The key findings of the study are summarized, and recommendations for future research or actions are provided based on the results.

By following this methodology, data scientists and analysts can systematically collect, clean, analyze, and train models on the available dataset. This process helps in gaining a comprehensive understanding of the data, developing accurate predictive models, and evaluating their performance for making informed decisions.

Data Collection

The dataset utilized in this project was a sourced from Kaggle Inc [7]. The dataset contains information about different houses in Boston. The dataset consists of 545 samples and 12 feature variables.

Data Preparation

The steps ensure the data is in a suitable format for analysis and model training, improving the accuracy and reliability of the results.

Data Dictionary

The dataset can be described briefly as follows.

Table 1. Data Dictionary

Attributes	Explanation	Variable type
Price	Price of the Houses	Numeric/ Continuous
Area	Area of a House	Numeric/ Continuous
Bedrooms	Number of House Bedrooms	Categorical/nominal
Bathrooms	Number of Bathrooms	Categorical/nominal
Stories	Number of House Stories	Categorical/nominal
Mainroad	Weather connected to Main Road	Categorical/nominal
Guestroom	Weather the house has a guest room	Categorical/nominal
Basement	Weather the house has a basement	Categorical/nominal
Hotwaterheating	Weather the house has a hot water heater	Categorical/nominal
Airconditioning	Whether the house has an air conditioning system.	Categorical/nominal
Parking	The number of parking spaces available for the house.	Categorical/nominal

Prefarea	Whether the house is located in a preferred area or not.	Categorical/nominal
Furnishingstatus	The furnishing status of the house.	Categorical/nominal

```
RangeIndex: 545 entries, 0 to 544
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   price             545 non-null    int64
 1   area              545 non-null    int64
 2   bedrooms          545 non-null    int64
 3   bathrooms         545 non-null    int64
 4   stories           545 non-null    int64
 5   mainroad          545 non-null    object
 6   guestroom         545 non-null    object
 7   basement          545 non-null    object
 8   hotwaterheating   545 non-null    object
 9   airconditioning   545 non-null    object
 10  parking           545 non-null    int64
 11  prefarea          545 non-null    object
 12  furnishingstatus  545 non-null    object
dtypes: int64(6), object(7)
```

Fig 1. Data types in Python

Missing Values

After reading the “Hosuing.csv” dataset into a Pandas dataframe. We looked into datatypes, missing values and duplicate rows.

```
data.isnull().sum()
```

```
price             0
area              0
bedrooms          0
bathrooms         0
stories           0
mainroad          0
guestroom         0
basement          0
hotwaterheating   0
airconditioning   0
parking           0
prefarea          0
furnishingstatus  0
dtype: int64
```

```
data.duplicated().value_counts()
```

```
False    545
dtype: int64
```

Fig 2. Missing values and duplicates

Our analysis of the dataset revealed that there are no instances of missing values or duplicates. This indicates that the dataset is complete and unique. However, we can see that there are categorical variables in the dataset and we need to convert categorical variables into numerical representations before training the model. To handle categorical variables, we can use one-hot encoding or label encoding. One-hot encoding creates binary columns for each category, while label encoding assigns a unique numerical value to each category.

Outlier's detection for numerical attributes

Table 2. Outlier counts and percentages for the attributes

Attributes	# of Outliers	% of Outliers
Price	6 outliers	1.10%
Area	7 outliers	1.28%
Bedrooms	2 outliers	0.37%
Bathrooms	11 outliers	2.02%
Stories	0 outliers	0%

Parking	0 outliers	0%

The analysis aimed to determine the impact of outliers on housing price prediction. Outliers are observed in the price, area, bedrooms, and bathrooms columns. The count of outliers for each attribute is relatively low, ranging from 2 to 11 instances. No outliers are found in the stories and parking columns. The question at hand was whether removing outliers would enhance the accuracy of the linear regression model.

The results indicate that removing outliers could be beneficial for certain attributes, such as price, area, bedrooms, and bathrooms. Outliers have the potential to distort predictions and introduce bias into the model. By eliminating outliers, the model's accuracy may be improved. However, careful consideration is necessary when deciding to remove outliers, taking into account domain knowledge and specific requirements. Since the dataset had a relatively low count of outliers, their removal may not significantly impact the overall dataset. Further analysis is recommended to assess the validity of outliers and determine if they represent genuine extreme values or data errors. Documentation of the outlier identification criteria and an evaluation of the potential impact on the data's distribution and characteristics are essential aspects to consider.

Statistical Summary for Numerical attributes

Using the describe () function we generate a table that shows the summary statistics (count, mean, standard deviation, minimum, quartiles, and maximum) for numerical attribute of the dataset.

Table 3. Statistical Summary

	price	area	bedrooms	bathrooms	stories	parking
count	5.450000e+02	545.000000	545.000000	545.000000	545.000000	545.000000
mean	4.766729e+06	5150.541284	2.965138	1.286239	1.805505	0.693578
std	1.870440e+06	2170.141023	0.738064	0.502470	0.867492	0.861586
min	1.750000e+06	1650.000000	1.000000	1.000000	1.000000	0.000000
25%	3.430000e+06	3600.000000	2.000000	1.000000	1.000000	0.000000
50%	4.340000e+06	4600.000000	3.000000	1.000000	2.000000	0.000000
75%	5.740000e+06	6360.000000	3.000000	2.000000	2.000000	1.000000
max	1.330000e+07	16200.000000	6.000000	4.000000	4.000000	3.000000

During our analysis, we explored important aspects of the housing dataset to gain a better understanding of the data. Here are the key findings:

Price: The average house price is around 4.77 million, ranging from 1.75 million to 13.3 million. Most houses have prices below 5.74 million.

Area: The average house size is approximately 5150 square units, ranging from 1650 to 16200 square units. The majority of houses have an area below 6360 square units.

Bedrooms: On average, houses have about 3 bedrooms, ranging from 1 to 6. Most houses have 3 or fewer bedrooms.

Bathrooms: Houses typically have around 1.3 bathrooms, ranging from 1 to 4. The majority of houses have 2 or fewer bathrooms.

Stories: Most houses have 2 or fewer stories, with an average of 1.81 stories and a range of 1 to 4.

Parking: On average, houses have approximately 0.69 parking spaces, ranging from 0 to 3. The majority of houses have 1 or fewer parking spaces.

These findings provide valuable insights into the dataset, giving us a clearer picture of the distribution and characteristics of the variables. This knowledge can be used to further analyze and develop models for predicting housing prices based on these features.

Exploratory Data Analysis (EDA)

Python was chosen as the programming language for this project due to its versatility and the availability of numerous packages for basic statistical analysis and building complex models. The following packages were loaded for data cleaning, preparation, building, and plotting the dataset: Numpy, Pandas, Sklearn, Scipy, Seaborn, dataprep, and Mathplotlib. We downloaded the dataprep library for the summary of the exploratory data analysis.

Dataset Statistics

Number of Variables	13
Number of Rows	545
Missing Cells	0
Missing Cells (%)	0.0%
Duplicate Rows	0
Duplicate Rows (%)	0.0%
Total Size in Memory	251.7 KB
Average Row Size in Memory	473.0 B
Variable Types	Numerical: 2 Categorical: 11

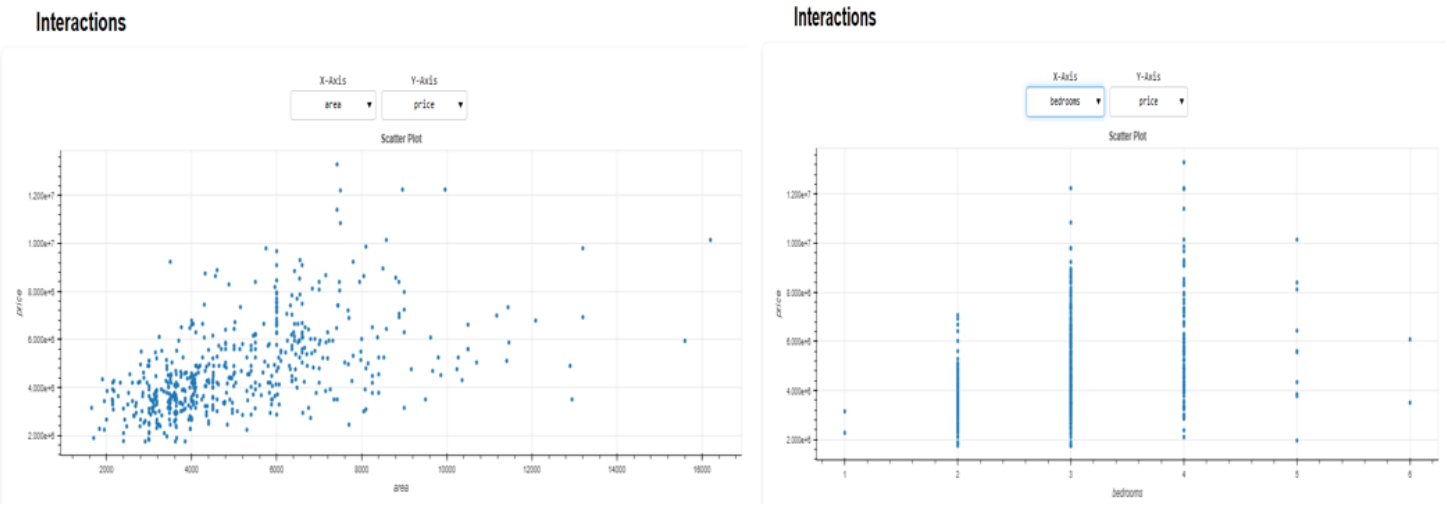


Fig 3. Exploratory data analysis

The Analysis of Main Factors Affecting Housing Price

Housing price is affected by multiple factors and features of a specific house. According to the previous research, some analysts have proposed several variables that significantly influence the overall housing price. House factors can be divided into several types. The most influential type is residential factors, including residence, usability, and number of rooms. When people consider purchasing a house for living purposes, the factors above are the main determinants for the living quality. Buyers with family members would typically attach more importance to the essential feature of the house, like the living area and number of rooms, which have a significant impact on the overall living quality and experience in the house. Besides, the intangible features, like the view of residence and usability, also have a rather considerable influence on the housing price, through affecting buyers' experience on the house and willingness to pay.

On the other hand, floor factors, like the number of stories, have also impacted the housing price significantly. Typically, household prefers the house with the number of the stories most suitable for their daily convenience. A family with children and elders tends to prefer a house with

multifloor construction, which offers different family members separate living areas with appropriate privacy while living together.

Univariate Analysis

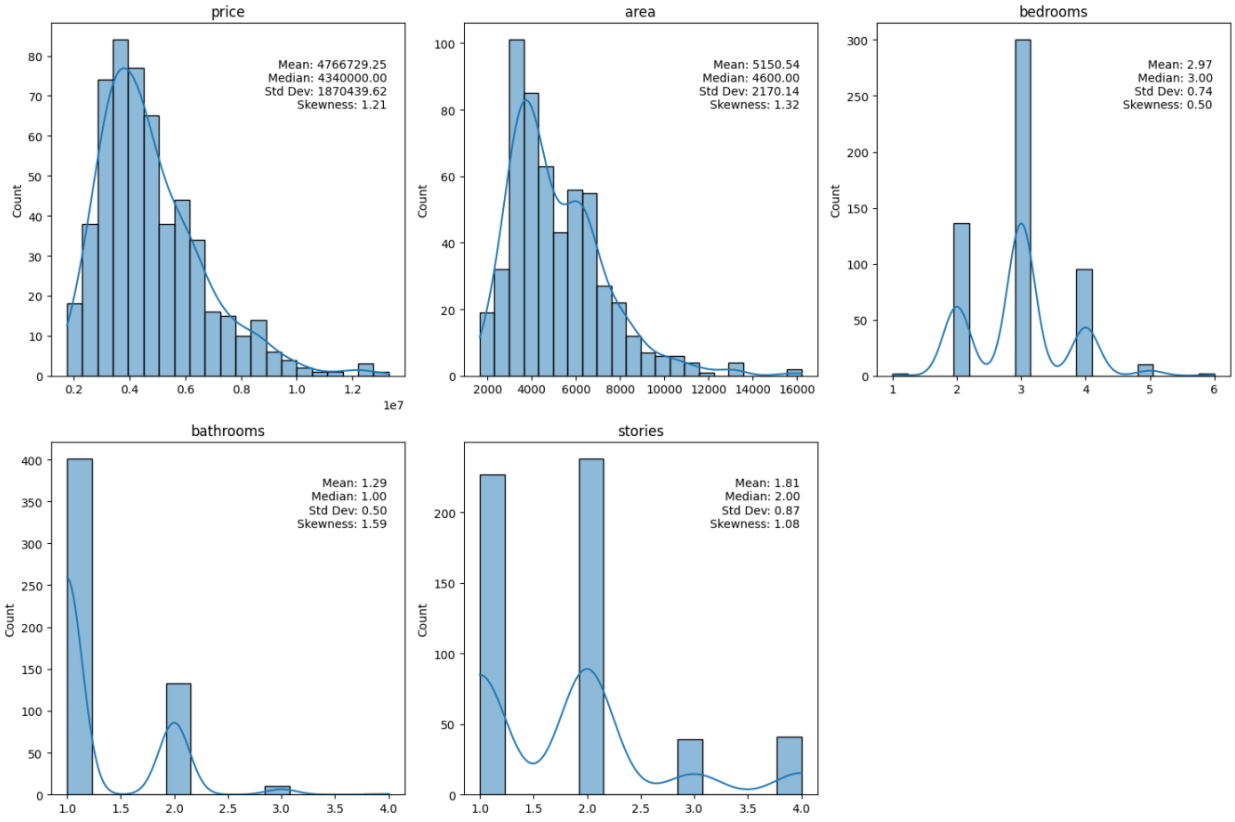


Fig 4. Univariate Analysis

We have used the seaborn library to plot histograms with kernel density estimation. It also calculates the skewness of each numerical attribute using the skew() function from pandas, and displays the skewness value on the plot. The skewness value indicates the degree of asymmetry in the distribution of the attribute. Positive skewness indicates a right-skewed distribution, negative skewness indicates a left-skewed distribution, and a skewness close to zero indicates a roughly symmetric distribution.

Bivariate Analysis

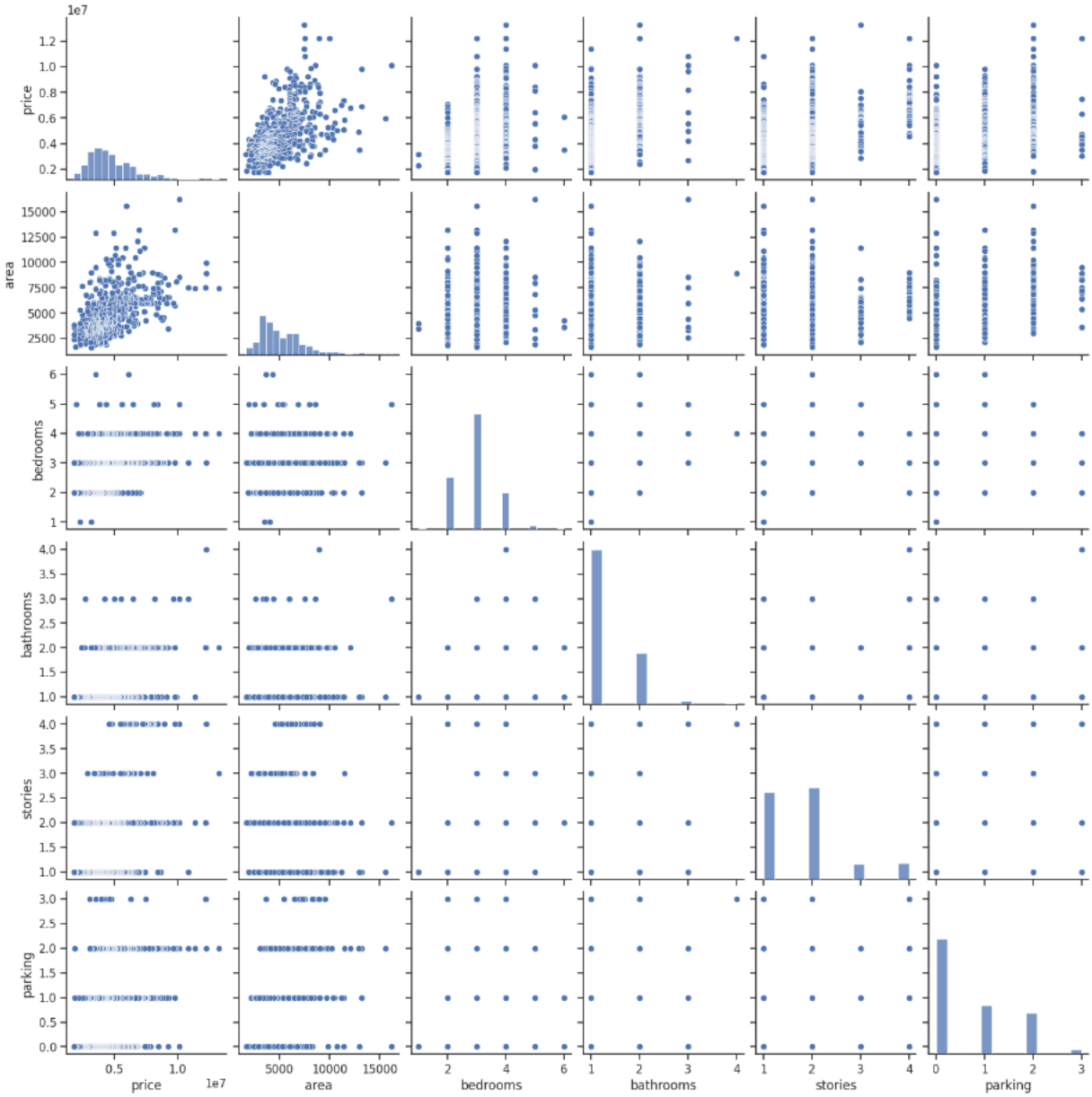


Fig 5. Bivariate Analysis

Correlation Analysis

The correlation analysis reveals the following key relationships in the dataset:

Price is positively correlated with area (0.54), bedrooms (0.37), bathrooms (0.52), stories (0.42), and parking (0.38). This suggests that larger houses with more bedrooms, bathrooms, stories, and parking spaces tend to have higher prices.

Area shows a positive correlation with bedrooms (0.15) and bathrooms (0.19), indicating that larger houses tend to have a slightly higher number of bedrooms and bathrooms.

Bedrooms exhibit a positive correlation with bathrooms (0.37) and stories (0.41), suggesting that houses with more bedrooms tend to have more bathrooms and stories.

Bathrooms show a positive correlation with stories (0.33), indicating that houses with more bathrooms tend to have more stories.

The correlation between stories and parking is weak (0.046), suggesting a minimal relationship between the number of stories and parking spaces.

These correlation coefficients provide valuable insights into the relationships between variables and can help understand the factors influencing housing prices. It indicates that house size, number of bedrooms and bathrooms, and availability of parking spaces are important factors to consider when predicting housing prices.

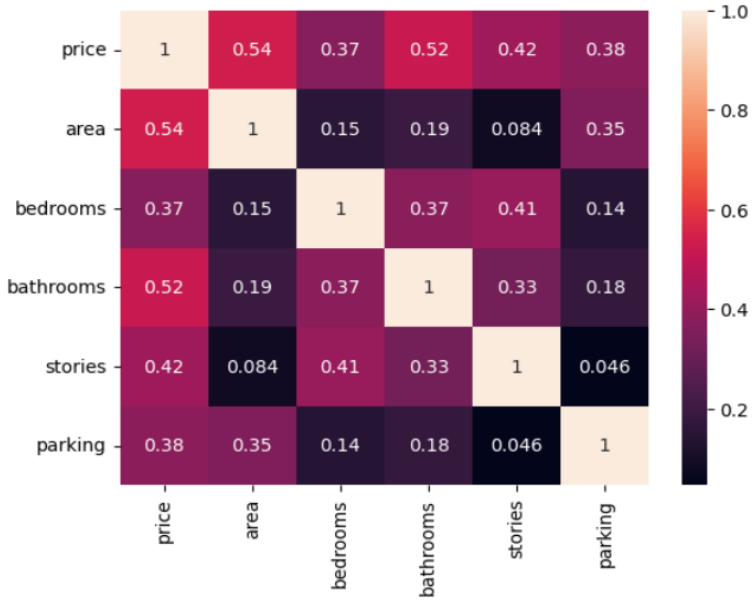


Fig 6. Correlation Analysis

Data Visualization

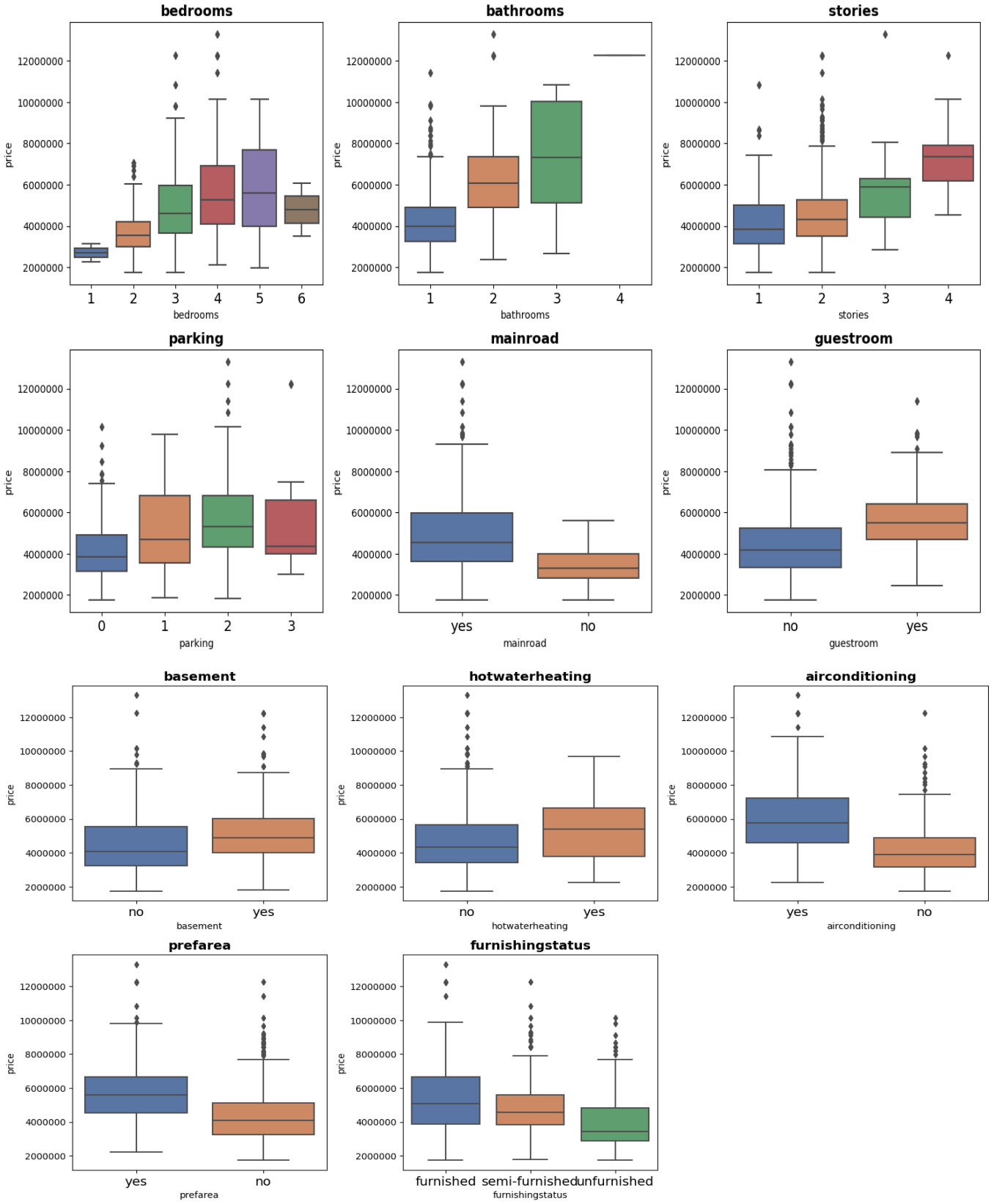


Fig 7. Box plot

The box plot shows a visual representation of the central tendencies, range, median, outliers and variabilities of the attributes. Understanding these characteristics helps us in analyzing the relationship between the attributes and the target variable (Price).

Further analysis was done of Average Price Variation for above and below Average Houses by Categorical Variables and the results obtained provide insights into the impact of different attributes on housing prices. We can observe that certain attributes have a substantial effect on the average price of houses. For instance, the number of bedrooms, bathrooms, stories, and the furnishing status seem to significantly influence housing prices. Houses with more bedrooms, bathrooms, and stories tend to have higher average prices, indicating that larger and more spacious properties command higher values.

Additionally, the furnishing status appears to play a crucial role in determining housing prices. Furnished houses have higher average prices compared to semi-furnished or unfurnished ones. This suggests that the level of furnishing and amenities provided in a house can contribute to its value.

It is important to note that other attributes such as the presence of a basement, hot water heating, air conditioning, parking availability, and preferred area also contribute to price variations, although their impact may not be as prominent as the aforementioned attributes.

By understanding the influence of these attributes on housing prices, individuals can make more informed decisions when buying or selling properties. Real estate professionals can also utilize this information to better assess property values and advise clients accordingly.

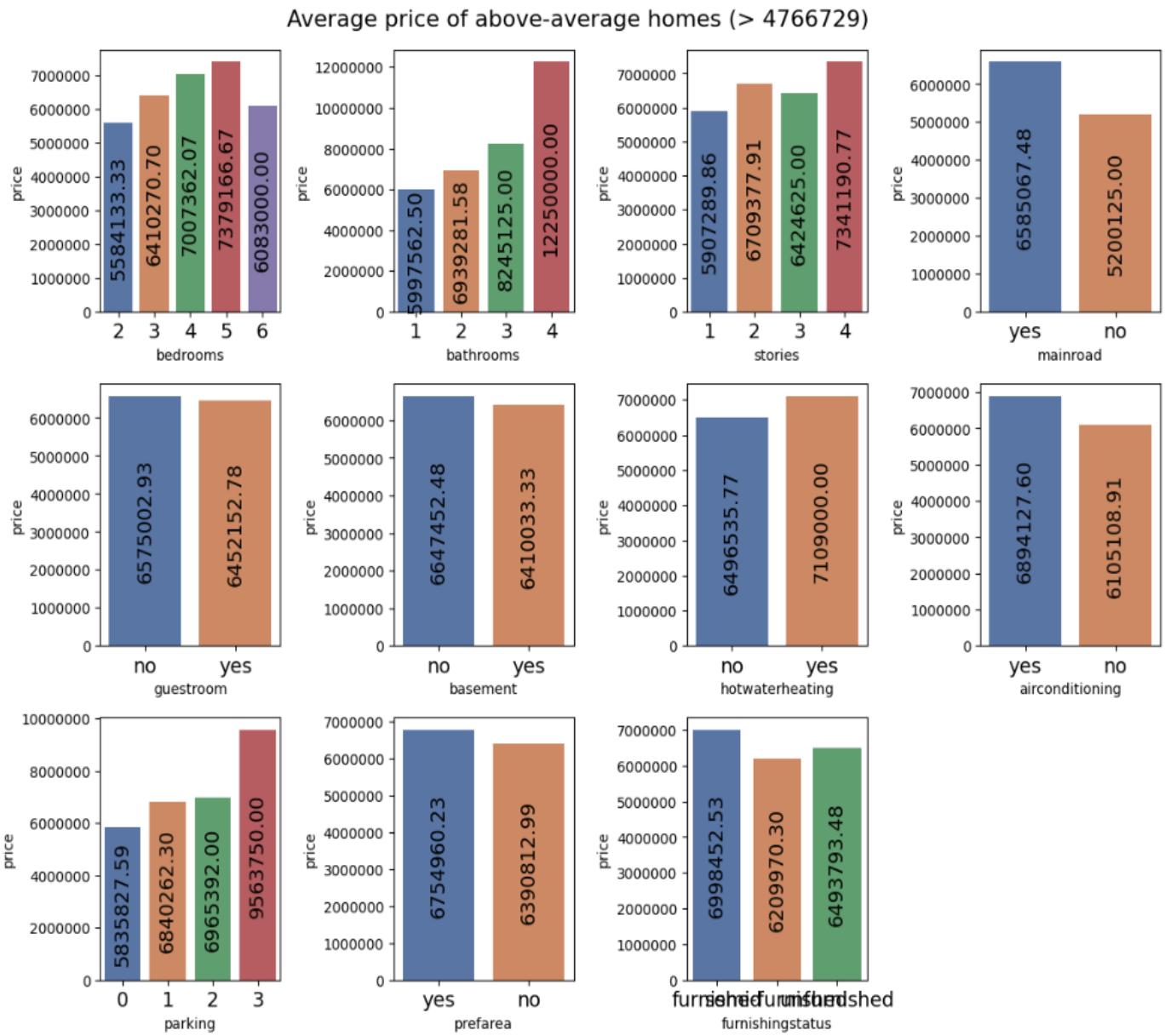


Fig 8. Average price of above-average houses

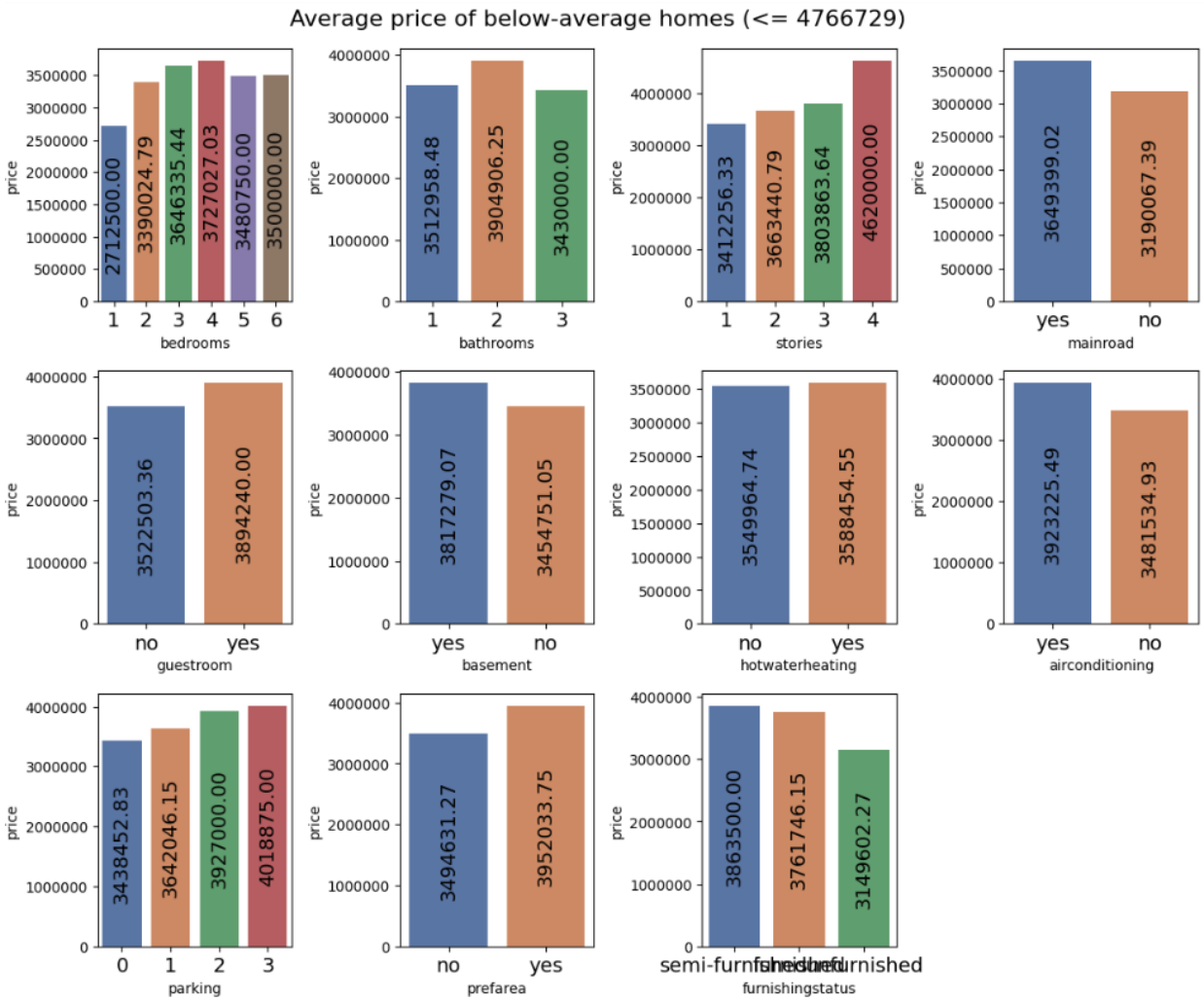


Fig 9. Average price of below-average houses

Data Normalization

Data normalization, also known as feature scaling, is an important technique used to standardize the range of values in numerical features of a dataset. In this project, we utilized the StandardScaler function from the scikit-learn library in Python. The StandardScaler method transforms the data by subtracting the mean from each data point and dividing it by the standard deviation. This process ensures that the transformed data has a mean of 0 and a standard deviation of 1. By bringing the features to a similar scale, StandardScaler enables better comparability and improves the performance of machine learning algorithms, particularly those sensitive to the data's scale.

Table 4. Feature Scaling

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	1.046726	1.403419	1.421812	1.378217	yes	no	no	no	yes	1.517692	yes	furnished
1	12250000	1.757010	1.403419	5.405809	2.532024	yes	no	no	no	yes	2.679409	no	furnished
2	12250000	2.218232	0.047278	1.421812	0.224410	yes	no	yes	no	no	1.517692	yes	semi-furnished
3	12215000	1.083624	1.403419	1.421812	0.224410	yes	no	yes	no	yes	2.679409	yes	furnished
4	11410000	1.046726	1.403419	-0.570187	0.224410	yes	yes	yes	no	yes	1.517692	no	furnished

Encoding Categorical Variables

When encoding categorical attributes, the choice of encoding method depends on the nature of the categorical variables and the specific requirements of your problem. Here are a few common encoding methods [7]:

One-Hot Encoding: This method creates binary columns for each category of a categorical variable. It is suitable when there is no inherent order or hierarchy among the categories. We can use pd.get_dummies() in pandas to perform one-hot encoding.

Label Encoding: This method assigns a unique numeric label to each category of a categorical variable. It is suitable when there is an ordinal relationship or some kind of inherent order among the categories. We can use LabelEncoder from the sklearn.preprocessing module to perform label encoding.

Ordinal Encoding: This method assigns integer values to the categories based on their order or rank. It is suitable when there is a meaningful order or ranking among the categories. We can manually define a mapping of categories to integers or use the OrdinalEncoder from the sklearn.preprocessing module.

Binary Encoding: This method represents each category with binary digits, resulting in fewer columns compared to one-hot encoding. It is suitable when dealing with high-cardinality

categorical variables. We can use the category_encoders.BinaryEncoder from the category_encoders library to perform binary encoding.

Hashing Encoding: This method applies a hashing function to the categories and assigns them to a fixed number of bins. It is suitable when dealing with high-cardinality categorical variables and limited memory. We can use the category_encoders.HashingEncoder from the category_encoders library to perform hashing encoding.

Once we have dealt with outliers, we can proceed with one-hot encoding for categorical variables. This step is important for machine learning algorithms like linear regression, as they typically require numerical input. The provided code uses the LabelEncoder from sklearn.preprocessing to encode categorical variables in the data DataFrame. LabelEncoder is used because it is a simple and straightforward encoding technique for categorical variables. It assigns unique numeric labels to each category, making it suitable for algorithms that require numeric input.

Table 5. Encoded categorical variable

First 5 rows of the dataset:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	1.046726	1.403419	1.421812	1.378217	1	0	0	0	1	1.517692	1	0
1	12250000	1.757010	1.403419	5.405809	2.532024	1	0	0	0	1	2.679409	0	0
2	12250000	2.218232	0.047278	1.421812	0.224410	1	0	1	0	0	1.517692	1	1
3	12215000	1.083624	1.403419	1.421812	0.224410	1	0	1	0	1	2.679409	1	0
4	11410000	1.046726	1.403419	-0.570187	0.224410	1	1	1	0	1	1.517692	0	0

Feature Selection

Feature selection techniques aim to identify the most relevant and informative features from a given dataset. They help to reduce dimensionality, improve model performance, and mitigate the risk of overfitting. Here are some commonly used feature selection techniques [8]:

Filter Methods: These methods assess the relevance of features independently of any specific machine learning algorithm. They typically rely on statistical measures such as correlation, chi-square test, or mutual information to rank the features. Examples include SelectKBest and Pearson's correlation coefficient. Filter methods are computationally efficient but may overlook feature dependencies.

Wrapper Methods: These methods evaluate feature subsets by training and testing a specific machine learning algorithm. They consider the performance of the model as a criterion for selecting features. Examples include Recursive Feature Elimination (RFE) and Forward/Backward Stepwise Selection. Wrapper methods can be computationally expensive but provide more accurate feature subsets.

Embedded Methods: These methods incorporate feature selection within the learning algorithm itself. They optimize feature selection as part of the model training process. Examples include Lasso (L1 regularization), Ridge (L2 regularization), and Decision Tree-based methods like Random Forest and Gradient Boosting. Embedded methods provide a balance between filter and wrapper methods, as they consider feature relevance and model performance simultaneously.

Based on the insights gained from EDA, we can further engineer or select relevant features that might have a significant impact on the housing price. This step helps in improving the performance of the model by including meaningful and informative features.

The provided code performs feature selection using the SelectKBest method from sklearn.feature_selection. This method selects the top k features based on their score calculated using the f_regression score function.

First, the code separates the features (X) and the target variable (y) from the DataFrame. Then, it

creates an instance of the SelectKBest class and specifies the score function as f_regression and the desired number of features to select as k=9.

The fit_transform method is called on the selector object, which fits the selector to the data and transforms the data to contain only the selected features.

The indices of the selected features are obtained using the get_support method with the indices=True parameter.

Finally, the names of the selected features are extracted from the DataFrame using the selected indices.

The code aims to select the most important features from the dataset based on their relationship with the target variable. The selected_features variable will contain the names of the top 9 features that are deemed most relevant for the prediction task.

In the given code, SelectKBest is used as a filter-based feature selection technique. It evaluates the relationship between each feature and the target variable using the f_regression score function. It ranks the features based on their relevance in predicting the target variable. The advantage of SelectKBest is that it is computationally efficient and provides a straightforward way to select a fixed number (k) of top features based on their relationship with the target variable. It does not require training a specific model and can be used as a preprocessing step before applying any machine learning algorithm.

```
Selected Features:
  Selected Features
0              area
1          bedrooms
2         bathrooms
3           stories
4          mainroad
5   airconditioning
6           parking
7          prefarea
8  furnishingstatus
```

Fig 10. Selected Features

Machine learning Algorithms

Various methods can be used to predict the house price. In our work, we propose three models: Linear Regression, Decision Tree, and Random Forest Regression to see which one gives better performance.

Linear Regression

Linear regression is a supervised machine learning method that is used by the Train Using AutoML tool and finds a linear equation that best describes the correlation of the explanatory variables with the dependent variable. This is achieved by fitting a line to the data using least squares. The line tries to minimize the sum of the squares of the residuals. The residual is the distance between the line and the actual value of the explanatory variable. Finding the line of best fit is an iterative process.

The following is an example of a resulting linear regression equation:

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots$$

In the example above, y is the dependent variable, and x_1, x_2, and so on, are the explanatory variables. The coefficients (b_1, b_2, and so on) explain the correlation of the explanatory variables with the dependent variable. The sign of the coefficients (+/-) designates whether the variable is positively or negatively correlated. b_0 is the intercept that indicates the value of the dependent variable assuming all explanatory variables are 0 [9].

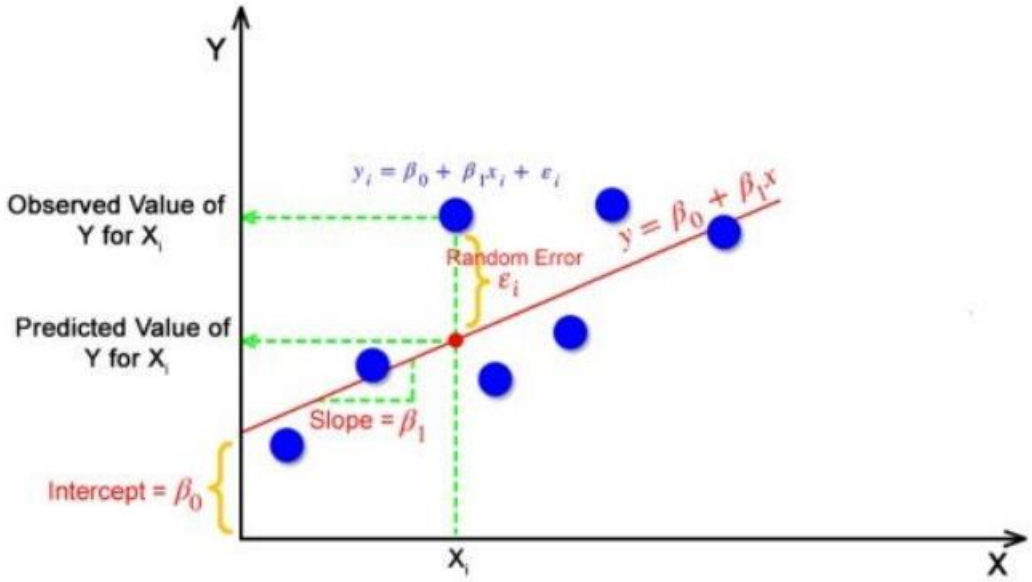


Fig 11. Linear regression Mode

The importance of linear regression lies in its simplicity, interpretability, and broad applicability. Linear regression provides interpretable coefficients that quantify the relationship between the independent variables and the dependent variable. These coefficients represent the change in the dependent variable for a unit change in the corresponding independent variable, holding other variables constant. This interpretability allows for clear understanding and insights into the impact of each variable on the outcome.

Decision Tree

Decision trees is a type of supervised machine learning algorithm that is used by the Train Using AutoML tool and classifies or regresses the data using true or false answers to certain questions. The resulting structure, when visualized, is in the form of a tree with different types of nodes—root, internal, and leaf. The root node is the starting place for the decision tree, which then branches to internal nodes and leaf nodes. The leaf nodes are the final classification categories or real values. Decision trees are easy to understand and are explainable.

To construct a decision tree, start by specifying a feature that will become the root node. Typically, no single feature can perfectly predict the final classes; this is called impurity. Methods such as Gini, entropy, and information gain are used to measure this impurity and identify how well a feature classifies the given data. The feature with the least impurity is selected as the node at any level. To calculate Gini impurity for a feature with numerical values, first sort the data in ascending order and calculate the averages of the adjoining values. Then, calculate the Gini impurity at each selected average value by arranging the data points based on whether the feature values are less than or greater than the selected value and whether that selection correctly classifies the data. The Gini impurity is then calculated using the equation below, where K is the number of classification categories and p is the proportion of instances of those categories.

$$Gini\ Impurity = 1 - \sum_{i=1}^{K} p_i^2$$

The weighted average of the Gini impurities for the leaves at each value is calculated. The value with the least impurity is selected for that feature. The process is repeated for different features

to select the feature and value that will become the node. This process is iterated at every node at each depth level until all the data is classified. Once the tree is constructed, to make a prediction for a data point, go down the tree using the conditions at each node to arrive at the final value or classification. When using decision trees for regression, the sum of squared residuals or variance is used to measure the impurity instead of Gini [10].

Decision trees provide a transparent and easy-to-understand representation of the decision-making process. The tree structure consists of a series of if-else conditions based on the values of the input features, leading to a final decision or prediction. This interpretability allows stakeholders to comprehend and trust the decision-making process, making it particularly useful in domains where explainability is crucial.

Decision trees can be applied to both classification and regression tasks, making them versatile algorithms. They can handle a wide range of problems, including customer segmentation, fraud detection, medical diagnosis, and financial forecasting.

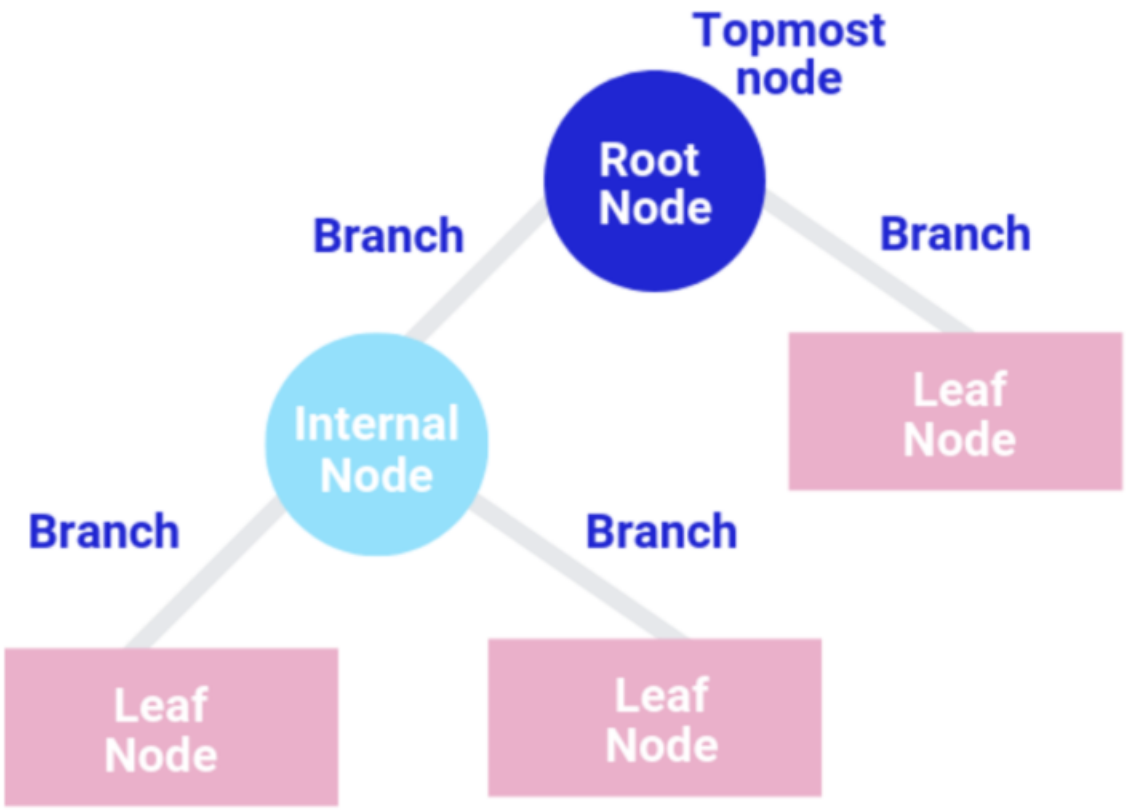


Fig 12. Structure of simple decision tree

Random Forest

Random Forest is a powerful ensemble learning algorithm that combines the predictions of multiple decision trees to make accurate predictions. It is widely used for classification and regression tasks. Here's an overview of how the Random Forest algorithm works:

Ensemble of Decision Trees: Random Forest consists of a collection of decision trees, where each tree is trained on a different subset of the training data. This ensemble approach helps to reduce the risk of overfitting and provides more robust predictions.

Random Subsampling: The Random Forest algorithm randomly selects a subset of the training data (with replacement) for each tree. This process is known as bootstrapping. By creating

random subsets, each tree gets exposed to different variations of the data, leading to diverse and independent trees.

Random Feature Selection: In addition to using random subsets of the data, Random Forest also performs random feature selection for each split in the decision tree. At each node, only a subset of features is considered for splitting, which introduces further randomness and reduces the correlation between trees. This process helps to capture different aspects of the data and avoid over-reliance on a single feature.

Building Decision Trees: Each decision tree in the Random Forest is grown using a recursive process called recursive partitioning. The tree is built by repeatedly splitting the data based on the selected features and their thresholds, aiming to minimize impurity or maximize information gain. The process continues until a stopping criterion is met, such as reaching a maximum depth or achieving a minimum number of samples at a leaf node.

Aggregating Predictions: Once all the decision trees are built, the Random Forest algorithm combines their predictions to make the final prediction. For classification tasks, the majority voting of the individual tree predictions determines the final class label. For regression tasks, the predictions of all trees are averaged to obtain the final continuous output.

Advantages of Random Forest: Random Forest offers several advantages. It provides robustness against overfitting, handles high-dimensional data, and is less sensitive to outliers and missing values. It can capture complex relationships, handle both categorical and numerical features, and estimate feature importance. Random Forest also allows parallelization, making it efficient for large datasets.

Overall, Random Forest is a versatile and powerful algorithm that leverages the strength of multiple decision trees to deliver accurate predictions. Its ability to reduce overfitting, handle diverse data, and provide valuable insights makes it a popular choice for various machine learning tasks.

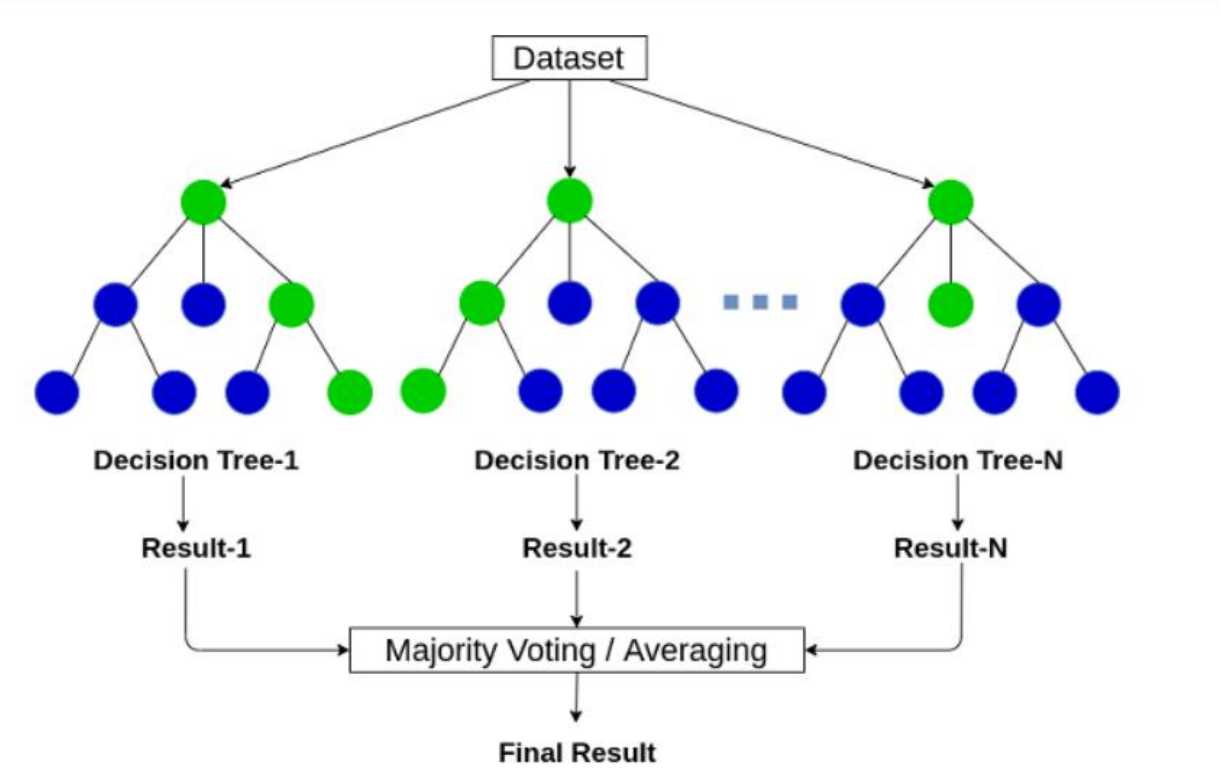


Fig 13. Structure of a simple Random forest Model

Model Training and Model Selection

The prepared dataset is used to train the models using three machine learning algorithms (Linear Regression, Decision Tree, and Random Forest). The cross-validation technique is employed to capture the underlying linear relationship between the features and the target variable, ensuring robust model evaluation and performance estimation.

We have used K fold cross validation technique. Cross-validation is a technique used to assess the performance of a model on unseen data by splitting the available data into multiple subsets or

folds. The cv parameter specifies the number of folds to create. During cross-validation, the model is trained and evaluated multiple times, with each fold serving as the test set once while the remaining folds are used as the training set. This helps in obtaining a more reliable estimate of the model's performance by reducing the impact of the specific data split.

In the code we have used, CV=10 means that the data will be divided into 10 equal-sized folds. Using cv=10 in cross-validation provides a more robust estimate of model performance compared to using a single train-test split. It helps to reduce the impact of variability in the training and testing data splits and provides a more comprehensive evaluation of the model's performance across different subsets of the data.

By increasing the number of folds (e.g., from cv=5 to cv=10), you can obtain a more precise estimate of the model's performance. However, this also increases the computational cost since the model needs to be trained and evaluated 10 times instead of 5. It is a trade-off between computational complexity and the accuracy of the performance estimation.

The evaluation metrics (MAE, MSE, RMSE) are calculated using the negative scoring approach. By taking the negative of the scores, lower values indicate better performance for these metrics. The average of the negative scores is then taken to compute the actual values of the evaluation metrics (MAE, MSE, RMSE). These metrics provide insights into the model's accuracy and performance.

The code also calculates the R-squared (R2) coefficient using the r2_score function. R2 represents the proportion of variance in the target variable explained by the model. A higher R2 value (closer to 1) indicates a better fit of the model to the data.

Additionally, cross_val_predict is used to obtain the predicted values for each fold during cross-validation. These predicted values, along with the actual values, are then used to create a scatter plot with the fitted line using plt.scatter and plt.plot functions. This allows visualizing the

relationship between the actual and predicted values and assessing the performance of the model visually.

We then compared MAE, MSE, RMSE and R2 values for all 3 machine learning algorithms first on all features and then on the selected features and the model that gives lower value of MAE, MSE and RMSE and higher value of R2 is selected.

Table 6. Evaluation metrics

Metric	Definition	Benchmark
Mean Absolute error (MAE)	This calculates mean of the absolute variances between the projected and the observed values.	Lower is better
Mean Squared error (MSE)	This measures the average of the squared deviations between predicted values and actual values.	Lower is better
Root mean squared error (RMSE)	Measures the square root of the MSE and provides a more interpretable metric in the same units as the original data.	Lower is better
R squared (R2)	Indicates percentage of the variance in the predicted values that can be attributed to the model. Values closer to 100% indicate better fit of the model to the data.	Higher is better

Result Evaluation

Linear Regression

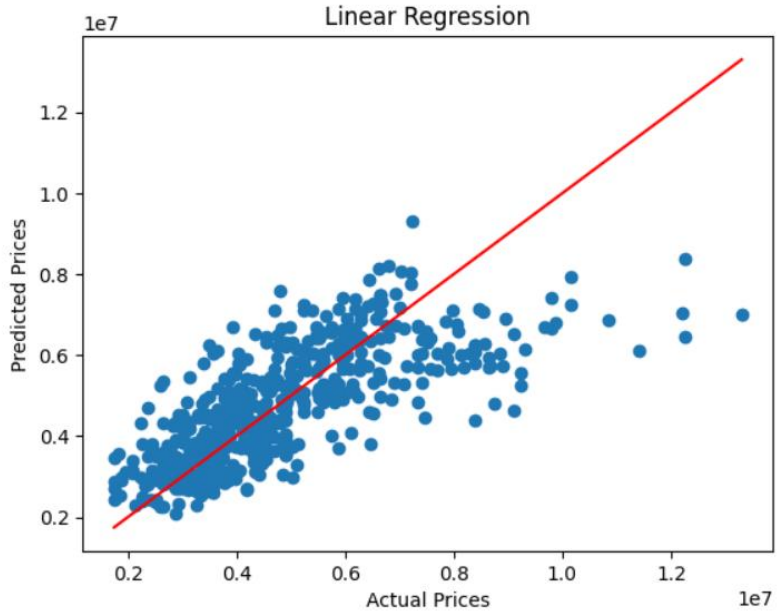
The linear regression model on all features, lower values of MAE, MSE, and RMSE indicate better performance. The MAE of 923,752.22 indicates that, on average, the model's predictions deviate from the actual values by approximately 923,752.22. The RMSE of 1,125,761.78 indicates the average prediction error of the model is around 1,125,761.78. The R-squared

coefficient of 0.54 indicates that approximately 54% of the variability in the target variable can be explained by the linear regression model.

The linear regression model on the selected features shows slightly worse performance compared to the linear regression model built using all features. The MAE of 942,081.97 indicates that, on average, the model's predictions deviate from the actual values by approximately 942,081.97. The RMSE of 1,147,761.34 indicates the average prediction error of the model is around 1,147,761.34. The R-squared coefficient of 0.52 suggests that approximately 52% of the variability in the target variable can be explained by the linear regression model on the selected features.

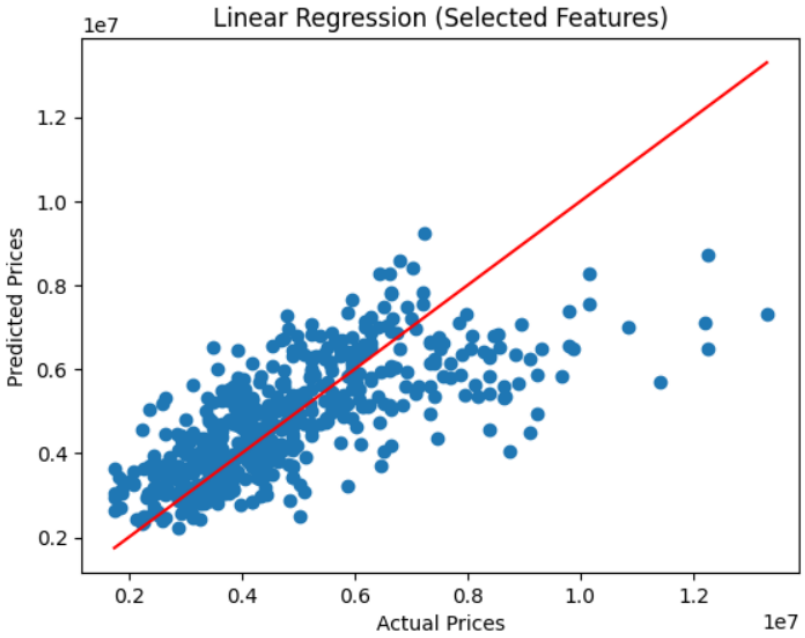
Table 7. Linear Regression Results: All Features vs. Selected Features

Machine learning Algorithms	On All Features		On Selected Features	
Linear Regression	Mean Absolute error (MAE)	923752.22	Mean Absolute error (MAE)	942081.97
	Mean Squared error(MSE)	1611590237884.42	Mean Squared error(MSE)	1681628847671.12
	Root Mean Squared Error (RMSE)	1125761.78	Root Mean Squared Error (RMSE)	1147761.34
	R-squared coefficient (R2)	0.54	R-squared coefficient (R2)	0.52




```
Mean Absolute Error (MAE): 923752.22
Mean Squared Error (MSE): 1611590237884.42
Root Mean Squared Error (RMSE): 1125761.78
R-squared coefficient (R2): 0.54
```

Fig 14. Linear regression model based on all features




```
Mean Absolute Error (MAE): 942081.97
Mean Squared Error (MSE): 1681628847671.12
Root Mean Squared Error (RMSE): 1147761.34
R-squared coefficient (R2): 0.52
```

Fig 15. Linear regression model based on selected features

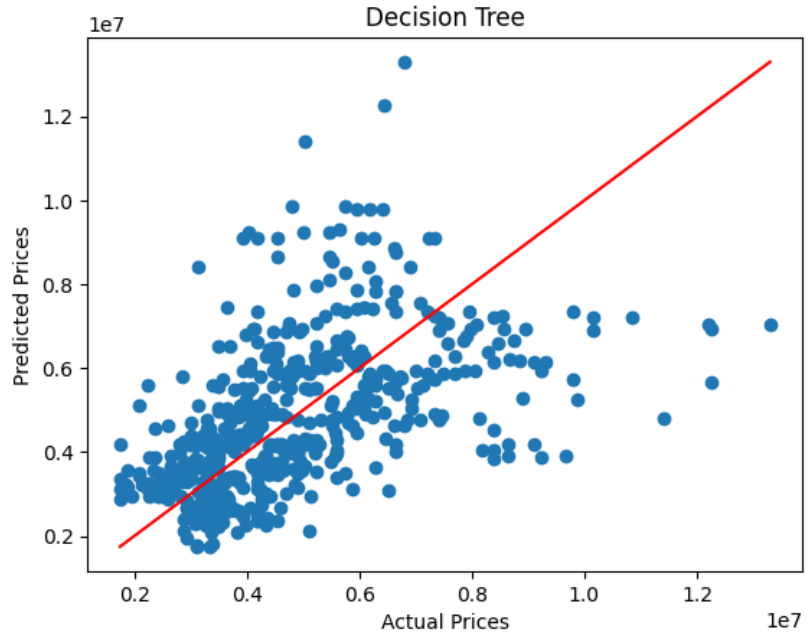
Decision Tree

The decision tree regression model based on all features has higher MAE, MSE, and RMSE values compared to linear regression indicate that the decision tree model's predictions have higher errors. The R-squared coefficient of 0.09 suggests that only approximately 9% of the variability in the target variable is explained by the decision tree model.

The decision tree model on the selected features shows similar performance as before, with high errors and low R-squared coefficient. The MAE and RMSE indicate larger errors compared to the linear regression models. The R-squared coefficient of 0.07 suggests that only approximately 7% of the variability in the target variable is explained by the decision tree model on the selected features.

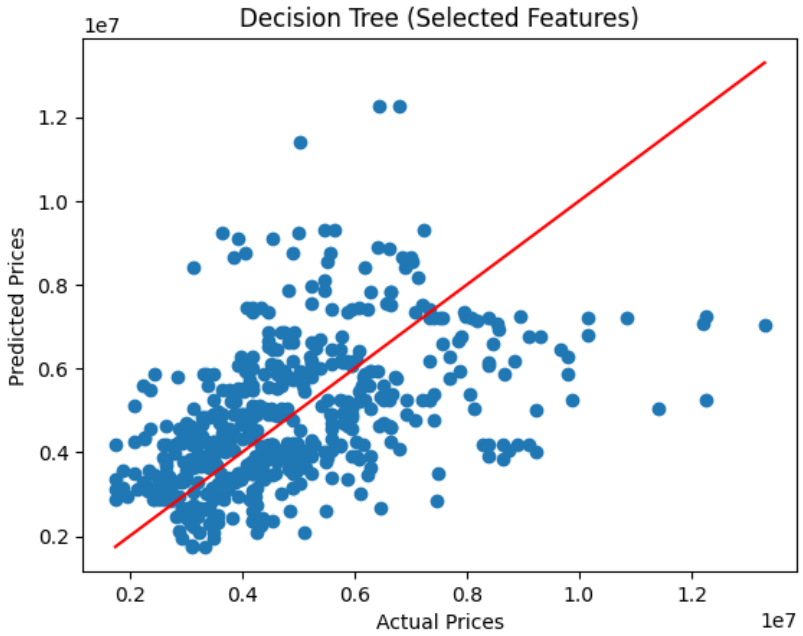
Table 8. Decision Tree Results: All Features vs. Selected Features

Machine learning Algorithms	On All Features		On Selected Features	
Decision Tree	Mean Absolute error (MAE)	1330459.8	Mean Absolute error (MAE)	1384848.4
	Mean Squared error(MSE)	3032178147171.84	Mean Squared error(MSE)	3316564971710.5
	Root Mean Squared Error (RMSE)	1632908.48	Root Mean Squared Error (RMSE)	1720871.29
	R-squared coefficient (R2)	0.09	R-squared coefficient (R2)	0.07




```
Mean Absolute Error (MAE): 1330459.8
Mean Squared Error (MSE): 3032178147171.84
Root Mean Squared Error (RMSE): 1632908.48
R-squared coefficient (R2): 0.09
```

Fig 16. Decision Tree model based on all features




```
Mean Absolute Error (MAE): 1384848.4
Mean Squared Error (MSE): 3316564971710.5
Root Mean Squared Error (RMSE): 1720871.29
R-squared coefficient (R2): 0.07
```

Fig 17. Decision Tree model based on selected features

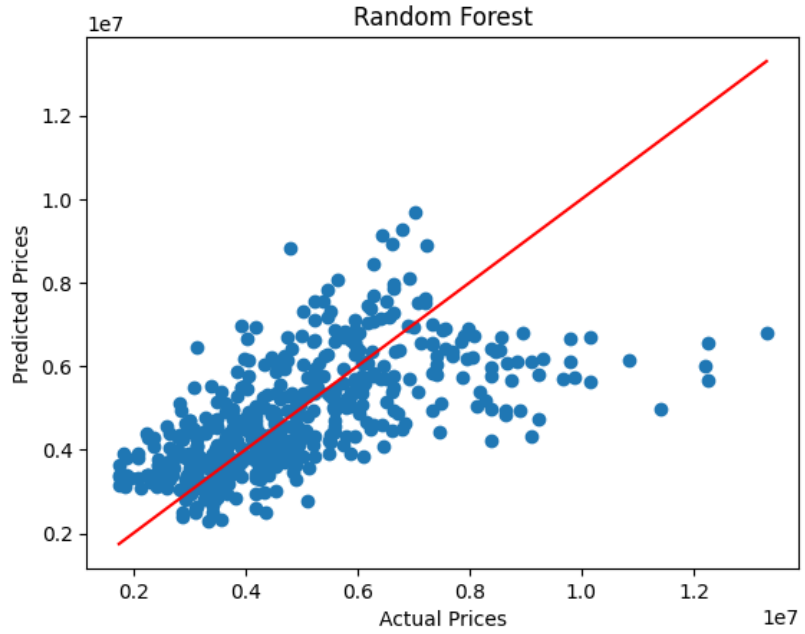
Random Forest

The Random Forest model on all features MAE and RMSE values are lower compared to the decision tree, indicating that the random forest model performs better in terms of prediction accuracy as compare to decision tree. The R-squared coefficient of 0.4 suggests that approximately 40% of the variability in the target variable can be explained by the random forest model.

The random forest model on the selected features performs similarly to the previous results obtained on all features. The R-squared coefficient of 0.35 suggests that approximately 35% of the variability in the target variable can be explained by the random forest model on the selected features.

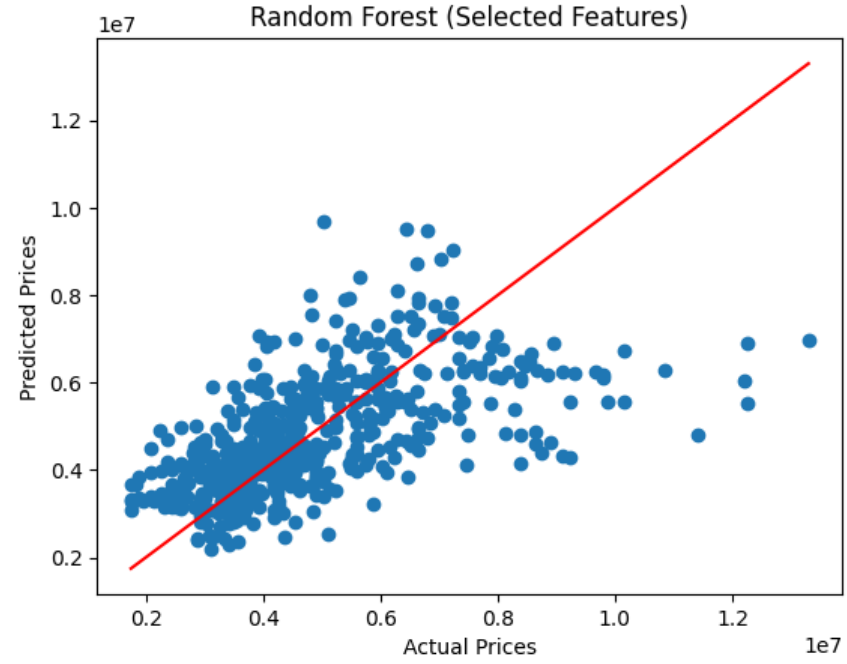
Table 9. Random Forest Results: All Features vs. Selected Features

Machine learning Algorithms	On All Features		On Selected Features	
Random Forest	Mean Absolute error (MAE)	1049105.27	Mean Absolute error (MAE)	1091003.72
	Mean Squared error(MSE)	2110493045137.91	Mean Squared error(MSE)	2274692051097.68
	Root Mean Squared Error (RMSE)	1278042.42	Root Mean Squared Error (RMSE)	1342702.08
	R-squared coefficient (R2)	0.4	R-squared coefficient (R2)	0.35




```
Mean Absolute Error (MAE): 1049105.27
Mean Squared Error (MSE): 2110493045137.91
Root Mean Squared Error (RMSE): 1278042.42
R-squared coefficient (R2): 0.4
```

Fig 18. Random Forest regression model based on all features




```
Mean Absolute Error (MAE): 1091003.72
Mean Squared Error (MSE): 2274692051097.68
Root Mean Squared Error (RMSE): 1342702.08
R-squared coefficient (R2): 0.35
```

Fig 19. Random Forest regression model based on selected features

Table 10. Overall Results of the 3 models

Machine learning Algorithms	On All Features		On Selected Features	
Linear Regression	Mean Absolute error (MAE)	923752.22	Mean Absolute error (MAE)	942081.97
	Mean Squared error(MSE)	1611590237884.42	Mean Squared error(MSE)	1681628847671.12
	Root Mean Squared Error (RMSE)	1125761.78	Root Mean Squared Error (RMSE)	1147761.34
	R-squared coefficient (R2)	0.54	R-squared coefficient (R2)	0.52
Decision Tree	Mean Absolute error (MAE)	1330459.8	Mean Absolute error (MAE)	1384848.4
	Mean Squared error(MSE)	3032178147171.84	Mean Squared error(MSE)	3316564971710.5
	Root Mean Squared Error (RMSE)	1632908.48	Root Mean Squared Error (RMSE)	1720871.29
	R-squared coefficient (R2)	0.09	R-squared coefficient (R2)	0.07
Random Forest	Mean Absolute error (MAE)	1049105.27	Mean Absolute error (MAE)	1091003.72
	Mean Squared error(MSE)	2110493045137.91	Mean Squared error(MSE)	2274692051097.68
	Root Mean Squared Error (RMSE)	1278042.42	Root Mean Squared Error (RMSE)	1342702.08
	R-squared coefficient (R2)	0.4	R-squared coefficient (R2)	0.35

From the overall results obtained, the evaluation metrics for the selected features show higher values of MAE, MSE, and RMSE compared to the evaluation metrics for all features. This indicates that the model trained on all features performs better than the model trained on the selected features.

Based on the results, the Linear Regression model trained on all features exhibits the lowest prediction errors and deviations, making it the better choice for predicting housing prices among the three models considered. It provides a relatively accurate estimation of housing prices and recommended for this task.

Note: considering the magnitude of the price variable in millions, it is evident that the high values of MAE, MSE, and RMSE obtained are mainly due to the difference in scales between the features and the target variable. In this case, we applied feature scaling but the result obtained are almost the same with our without feature scaling so it is not necessary to go for feature scaling because the high errors are expected due to the magnitude of the target variable.

Scaling the features can be more beneficial when the features themselves have different scales and need to be standardized for better model performance and interpretability. However, in this scenario, the high errors obtained can be attributed to the inherent magnitude of the target variable, and for our research feature scaling may not provide significant improvements in model performance.

Limitation of the work

Here are some project limitations and can be addressed in future work.

First off, our study uses a small dataset (545 rows) that limit how broadly the models may be used. To get over this limitation, getting a larger dataset with more samples would give more varied training examples and perhaps boost the model's accuracy and reliability.

Second, even though three different machine learning models were evaluated in this research, it would be worthwhile to look into more advance models or variations of these models. It might be beneficial to take other models or even ensemble techniques into account to enhance the performance even further.

Finally, to enhance the performance of these models, additional hyperparameter adjustment may be necessary.

Conclusion

In this data analytics capstone project, we utilized the "Housing Prices Dataset" for Boston to develop prediction models for estimating housing prices. By conducting exploratory data analysis and employing preprocessing techniques, we gained insights into the dataset and prepared the data for analysis.

We explored three machine learning techniques: linear regression, decision tree regression, and random forest regression, to build predictive models. These models took into account key features like the number of bedrooms, bathrooms, stories, and the area of the house, aiming to capture the underlying linear relationship with the price. We evaluated the performance of these models using commonly used metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the R2 coefficient.

Among the models considered, the Linear Regression model trained on all features stood out as the best model for housing pricing prediction. It showcased the lowest prediction errors and deviations, offering a relatively accurate estimation of housing prices. This model serves as a valuable tool for predicting housing prices and guiding individuals in making informed decisions in the real estate market. The limitations identified in this work provide valuable insights for future research. Future studies should consider larger datasets and explore alternative models for improved predictive performance.

References

1. Adetunji, A. B., Akande, O. N., Ajala, F. A., Oyewo, O., Akande, Y. F., & Oluwadara, G. (2022). House price prediction using Random Forest Machine Learning Technique. *Procedia Computer Science*, *199*, 806–813. https://doi.org/10.1016/j.procs.2022.01.100

2. Truong, Q., Nguyen, M., Dang, H., & Mei, B. (2020). Housing price prediction via Improved Machine Learning Techniques. *Procedia Computer Science*, *174*, 433–442. https://doi.org/10.1016/j.procs.2020.06.111

3. Zhang, Q. (2021). Housing price prediction based on multiple linear regression. *Scientific Programming*, *2021*, 1–9. https://doi.org/10.1155/2021/7678931

4. Lahmiri, S., Bekiros, S., & Avdoulas, C. (2023). A comparative assessment of machine learning methods for predicting housing prices using Bayesian optimization. *Decision Analytics Journal*, *6*, 100166. https://doi.org/10.1016/j.dajour.2023.100166

5. Thamarai, M., & Malarvizhi, S. P. (2020). House price prediction modeling using machine learning. *International Journal of Information Engineering and Electronic Business*, *12*(2), 15–20. https://doi.org/10.5815/ijieeb.2020.02.03

6. Ho, W. K. O., Tang, B.-S., & Wong, S. W. (2020). Predicting property prices with machine learning algorithms. *Journal of Property Research*, *38*(1), 48–70. https://doi.org/10.1080/09599916.2020.1832558

7. Saxena, S. (2023, July 14). *Here's all you need to know about encoding categorical data (with python code)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/08/types-of-categorical-data-encoding/

8. Gupta, A. (2023, April 26). *Feature selection techniques in Machine Learning (updated 2023)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/

9. *How linear regression algorithm works*. How Linear regression algorithm works-ArcGIS Pro | Documentation. (n.d.). https://pro.arcgis.com/en/pro-app/latest/tool-reference/geoai/how-linear-regression-works.htm

10. *How decision tree classification and regression algorithm works*. How Decision tree classification and regression algorithm works-ArcGIS Pro | Documentation. (n.d.). https://pro.arcgis.com/en/pro-app/latest/tool-reference/geoai/how-decision-tree-classification-and-regression-works.htm

11. Mali, K. (2023, May 2). *Everything you need to know about linear regression!*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/

12. Decision trees - the atlan data wiki. (n.d.). https://wiki.atlan.com/decision-trees/

13. Dataset https://www.kaggle.com/datasets/yasserh/housing-prices-dataset

GitHub Link

https://github.com/rabiadanish/CIND-820-Project