School of Science and Engineering

House price prediction

Asmaa Dalil

Supervised by:

Dr. Amin Amar

Contents

LIST OF FIGURES

Abstract:

In the scope of the Inferential Statistics class, we are required to work on a project to practice the material learned, Hence, this project explores house price prediction using regression models, focusing on factors like area, bedrooms, and more. The analysis showed that factors like house area and the number of bedrooms significantly affect house prices. Linear Regression and Random Forest models were built and tested for accuracy. The performance was measured using R-squared and Root Mean Squared Error, providing insights into the models' effectiveness and areas for improvement.

The project aims to deepen understanding of regression modeling and offer practical insights into the real estate market. It provides a foundation for further research and application of more complex models to improve prediction accuracy. The findings can assist buyers, sellers, and investors in making informed decisions in the housing market.

1. INTRODUCTION

In this project, we aim to predict house prices based on various factors. The dataset used in this project is particularly valuable for individuals who have recently completed basic regression techniques with a limited set of features and wish to explore regression analysis further. This study offers the opportunity to deepen understanding of regression modeling while gaining insights into the dynamics of housing markets, thereby enhancing decision-making processes in real estate investments and related fields.

1.1 Relationship between course ILOS and the project:

- *In milestone 1: Data Understanding and Cleanup*

Relation to ILO1: I used statistical fundamentals to comprehend the dataset by performing necessary data cleaning procedures. This involved examining the dataset for missing values and summarizing numeric columns, which are essential first steps in statistical analysis and inference.

- Milestone 2: Exploratory Data Analysis

Relation to ILO1: I implemented graphical and numerical explorations to understand the distribution and relationships among the variables. This allowed me to infer trends and characteristics beyond the raw data, such as the skewness in price and area distributions, which influence how these variables might be transformed or used in predictive models.

ILO2: Use the notion of parametric models, point estimation, and interval estimation of the parameters of those models (regression and time series models).

- Milestone 3: Regression Model Development and Evaluation

Relation to ILO2: I implemented a Linear Regression model and a Random Forest Regressor. I used these parametric models to perform point estimations, where I calculated coefficients for predictors like area, bedrooms, and bathrooms. I also used interval estimation by computing confidence intervals and significance tests (p-values) for these coefficients, helping validate the reliability of the model estimations.

ILO3: Demonstrate the plausibility of pre-specified ideas about the parameters of the model by examining the area of hypothesis testing.

- Milestone 3: Hypothesis Testing in Model Evaluation

Relation to ILO3: I tested hypotheses regarding the significance of model parameters. For instance, the F-statistic and associated p-value in the Linear Regression analysis helped me

reject the null hypothesis that all model coefficients are zero, confirming that the model has explanatory power concerning house prices.

ILO4: Tackle a number of inference problems using Bayesian and non-Bayesian methods and approximate inference. Demonstrate an understanding of the learning as inference.

- Milestone 3: Model Comparison and Hypothesis Testing

Relation to ILO4: I compared the performance of non-Bayesian models like Linear Regression and Random Forest, using statistical inference methods such as hypothesis testing to validate model assumptions and outputs. This comparison allowed me to understand how different models infer the relationship between features and house prices.

ILO5: Design: Identify a computer science project as a use case for data analytics, statistical inference, and prediction/forecasting and collect the required data and information. Implement: Formulate the statistical steps: data exploration, modeling, estimation, inference, and prediction/forecasting, with the help of python/R. Implement discussed algorithms (linear regression and random forest). Evaluate: Interpret results and I suggested way for improvements, to promote new ideas and innovative solutions.

- Milestone 1-3: Complete Project Execution

Relation to ILO5 Design: I identified the "Housing Price Prediction" project as a use case for applying data analytics and statistical inference. I collected and utilized data from a respected source (Kaggle), ensuring a robust dataset for analysis.

Relation to ILO5 Implement: I formulated and followed statistical steps across multiple milestones—data exploration, regression modeling using Linear Regression and Random Forest, estimation, and statistical inference. This was implemented using Python.

Relation to ILO5 Evaluate: I evaluated the models by interpreting the R-squared and RMSE values, which helped assess the effectiveness of the models. Based on these evaluations, I suggested that improvements could include exploring additional features, applying feature transformations to address skewness, or experimenting with more complex models to better capture nonlinear relationships.

2. Problem Statement

This project, titled "Housing Price Prediction," presents a straightforward yet challenging task of forecasting housing prices based on various factors such as house area, number of bedrooms, furnished status, proximity to main roads, etc. Despite its small dataset size, the project's complexity stems from the presence of strong multicollinearity among its variables.

2.1. Reasons behind this project:

This project on predicting housing prices is an excellent case for data analysis, prediction, and forecasting due to its real-world relevance and complexity. Housing markets impact the economy significantly, and insights into price dynamics can benefit buyers, sellers, and

investors alike. The dataset used includes a variety of variables such as area, number of bedrooms, and amenities, introducing real-world complexity that is ideal for applying advanced analytical techniques. Predictive modeling is particularly suitable here, given the numerous factors that influence housing prices, allowing for the use of both linear and non-linear models to understand different modeling approaches. Additionally, the data provides substantial opportunities for statistical inference, helping to draw broader conclusions about factors influencing prices. The skills and insights gained are not only applicable to other domains where predictive analytics are valuable but also enhance strategic thinking about the use of data-driven insights in practical scenarios.

3. PROJECT SPECIFICATIONS

The "Housing Price Prediction" project is designed to predict housing prices using advanced regression modeling techniques. It begins with a thorough understanding and cleanup of the dataset to ensure data quality, addressing any inconsistencies or missing values. The project aims to develop regression models that utilize both individual and multiple feature combinations, evaluating these models using metrics such as R-squared and Root Mean Squared Error (RMSE) to identify the most effective predictors.

Methodologies include Exploratory Data Analysis (EDA) to understand variable distributions and relationships, data preprocessing to handle missing data and encode categorical variables and feature engineering to enhance model performance. A variety of regression models such as Linear Regression, Ridge Regression, and Random Forest are explored to select the best performer based on metrics and interpretability. Cross-validation is employed to ensure model robustness, while hyperparameter tuning is used to optimize model parameters. The final steps involve evaluating the models to select the most effective one, interpreting model coefficients or feature importances to understand the driving factors of housing prices, and visualizing the predictions against actual values to identify patterns. The project culminates in the deployment of the final model for real-time predictions, ensuring the process is scalable and maintainable. This structured approach not only leverages data science techniques for accurate forecasting but also provides interpretable models that offer valuable insights to stakeholders in the real estate sector.

4. LITERATURE REVIEW

The project sums up the systematic approach to employing statistical models for real estate valuation. Initially, the project identifies key objectives, focusing on understanding and cleansing the dataset, addressing the challenge of multicollinearity among variables—a common occurrence in housing data where features are often interrelated. This sets the stage for careful data preprocessing, which is crucial for the reliability of subsequent predictive modeling.

The project then conducts exploratory data analysis, revealing biased distributions in the housing market, more frequent occurrences of lower-priced, smaller houses. This shows the real-world dynamics where a range of property sizes and values exist. Sampling methods, such as random and stratified techniques, provide a representative cross-section of the data, an important step for model accuracy and generalizability.

The core analysis utilizes Linear Regression and Random Forest models to estimate house prices, supported in correlation analysis and feature selection. Model performance is quantitatively evaluated using root mean squared error and R-squared metrics.

4.1 Challenges in Predictive Modeling for House Prices

Multicollinearity: A significant challenge in predictive modeling, especially with housing data, is the high correlation among independent variables, which can distort the true effect of each predictor on the outcome variable (house prices).Also, ensuring that the dataset is comprehensive and accurately reflects the market conditions is crucial. Any inconsistencies, missing values, or errors in the data can lead to inaccurate predictions and models that perform poorly when generalized to other datasets.

Not to fofrget that choosing the right model complexity is hard. Overly complex models may fit the training data well but generalize poorly (overfitting), while overly simple models may fail to capture important nuances (underfitting). Many relationships in real estate, such as those between price and location or size, may not be linear. Capturing these non-linear relationships without overcomplicating the model poses a technical challenge.

5. Data presentation

I collected my data from the Kaggle platform. Here is a link to the source:

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

The dataset has 13 features and 545 samples. grid_3x3

All variables come from the same source:

price: Quantitative - The sale price of the house. It's the target variable for

prediction, with values ranging from 1,750,000 to 13,300,000.

area: Quantitative - The size of the property in square feet. This numerical variable ranges from 1,650 to 16,200 square feet.

bedrooms: Quantitative - The count of bedrooms in the house. This is a numerical variable representing the number of bedrooms, which varies across the dataset.

bathrooms: Quantitative - The count of bathrooms in the house. Another numerical variable that indicates the number of bathrooms.

stories: Quantitative - The number of levels or floors in the house, which is a numerical representation of the story count.

mainroad: Qualitative (Categorical) - Indicates if the house has access to a main road ('yes' or 'no').

guestroom: Qualitative (Categorical) - Indicates whether the house has a guest room ('yes' or 'no').

basement: Qualitative (Categorical) - Represents whether the house has a basement ('yes' or 'no').

hotwaterheating: Qualitative (Categorical) - Denotes if the house is equipped with hot water heating ('yes' or 'no').

airconditioning: Qualitative (Categorical) - Shows whether the house has air conditioning ('yes' or 'no').

parking: Quantitative - The number of parking spots available, which is a numerical variable.

prefarea: Qualitative (Categorical) - Specifies if the house is in a preferred area ('yes' or 'no').

furnishingstatus: Qualitative (Categorical) - The furnishing status of the house, which could be 'furnished,' 'semi-furnished,' or 'unfurnished'.

5.1 Anonymization data protection procedures

In the scope of this project, It is not necessary to examine the anonymization and data protection procedures as the dataset consists only of property-related information with no personal identifiers linked to individual homeowners or residents.

5.2 Reliability, consistency and validity of sources

The Kaggle dataset is appropriate for educational use and initial data analysis, as commonly utilized within the data science community for these purposes. Although datasets on Kaggle are community-contributed and lack formal external validation, they are typically considered reliable for examining and academic projects.

For applications that require rigorous validation, such as in-depth research or professional use, additional verification against established data sources is recommended to ensure accuracy and reliability.

Community reviews and comments: A community of data scientists often reviews and comments on datasets on Kaggle. Take notice of any comments, opinions, or conversations made about the dataset by the community as these may provide light on its dependability.

5.3 Data exploration :

Tables:

```
dt=pd.read_csv("Housing.csv")
dt.head()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished

```
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
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
```

Figure1: Data exploration

After loading the data, we examined the dataset for missing values, summarize the numeric columns, and explore the data types of each column to understand the metadata better by using the following functions:

Missing values: The dataset contains 545 entries and does not have any missing values across all columns.

```
dtype: object
#missing_values
dt.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
```

```
dt.shape
```

```
(545, 13)
```

Figure2: Missing values

5.4 The response and the Explanatory variables

When predicting house prices

- The response variable (the variable you are trying to explain or predict) is the price of the house. This is the target variable.
- Explanatory variables (variables used to explain or predict the response variable) can include any other features in the dataset that might influence the price of a house. Potential explanatory variables could include: area, bedrooms, stories, furnishingstatus.

5.5 Numeric Features and Categorical Features

Numeric Features:

price: The house prices range from 1,750,000 to 13,300,000 with a mean of approximately 4,766,729.

area: The area of the houses ranges from 1,650 to 16,200 square feet, with a mean area of 5,150.54 square feet.

bedrooms, bathrooms, stories, and parking are numerical features with their respective counts and distributions.

Categorical Features:

mainroad, guestroom, basement, hotwaterheating, airconditioning,

prefarea, and furnishingstatus are object types indicating categorical data.

```
numerical_features = dt.select_dtypes(include=['int64', 'float64']).columns.tolist()
categorical_features = dt.select_dtypes(include=['object']).columns.tolist()

# Count the number of categorical and numerical features
num_categorical_features = len(categorical_features)
num_numerical_features = len(numerical_features)

print(f"Number of categorical features: {num_categorical_features}")
print(f"Number of numerical features: {num_numerical_features}")
```

```
Number of categorical features: 7
Number of numerical features: 6
```

```
6 numerical features: price, area, bedrooms, bathrooms, stories, parking.
7 categorical features: mainroad, guestroom, basement, hotwaterheating, airconditioning, prefarea, furnishingstatus.
```

Figure3: Number of Features

This dataset is quite clean with no missing values, allowing for proceeding directly with further analysis or modeling without needing imputation steps.

5.6 Graphical exploration

For a graphical exploration, we can visualize some of the numeric data distributions and relationships between house price and other features.

Handling categorical variables

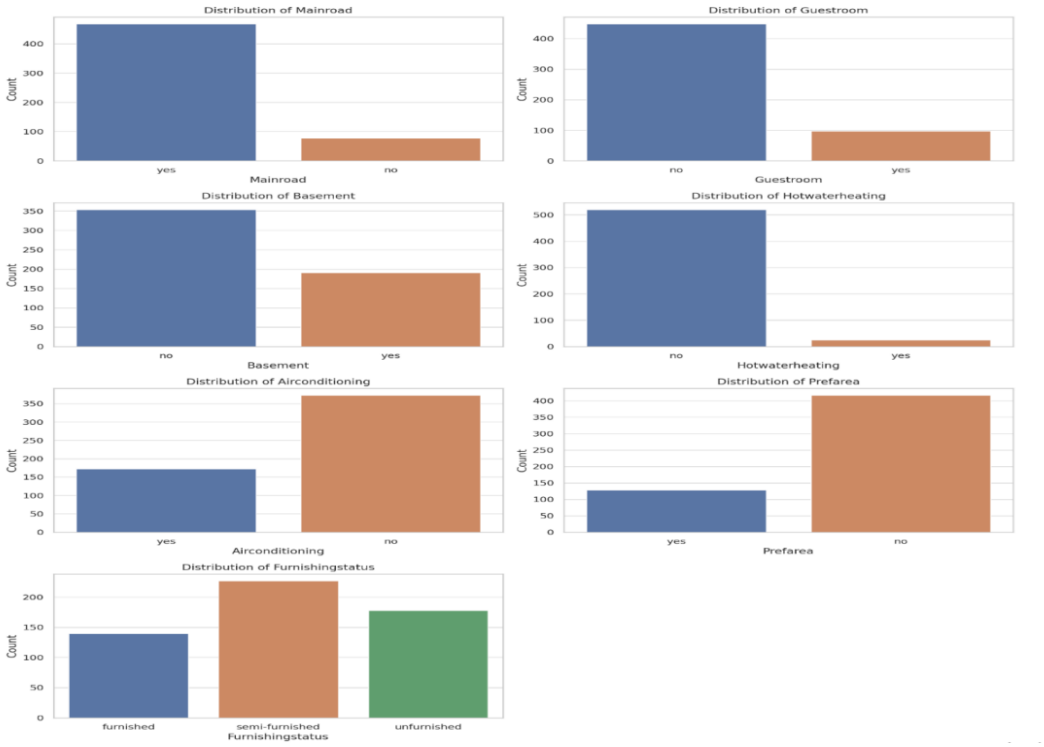


Figure4: Categorical values

Mainroad: A significant majority of houses have access to a main road. Guestroom: Most houses do not have a guestroom.
Basement: A greater number of houses do not have a basement compared to those that do.
Hotwaterheating: Few houses are equipped with hot water heating. Airconditioning: There are more houses without air conditioning than with.
Prefarea: A majority of houses are not in the preferred area.
Furnishingstatus: There is a relatively even distribution among furnished, semi-furnished, and unfurnished houses, with semi-furnished being slightly more common

These plots are valuable for understanding the distribution of categorical features and can provide insights when building predictive models or performing further analysis

Handling numerical variables :

Distribution of Price: The distribution is right-skewed, showing that most of the houses are on the lower end of the price spectrum, with fewer houses at higher prices. *Distribution of Area*: This also appears right-skewed, indicating that a majority of the houses have a smaller area, with fewer properties having a larger area.

Distribution of Bedrooms: Most houses have between 2 and 4 bedrooms, with 3 bedrooms being the most common.
Distribution of Bathrooms: It seems that 1 and 2 bathrooms are most common, with very few houses having more than 2 bathrooms.
Distribution of Stories: The majority of houses have 1 or 2 stories.
Distribution of Parking: Many houses do not have parking space, and the rest mostly have space for 1 or 2 cars.

These distributions give an overview of the numeric features in the dataset, which can be useful for understanding the range and common values for each feature.

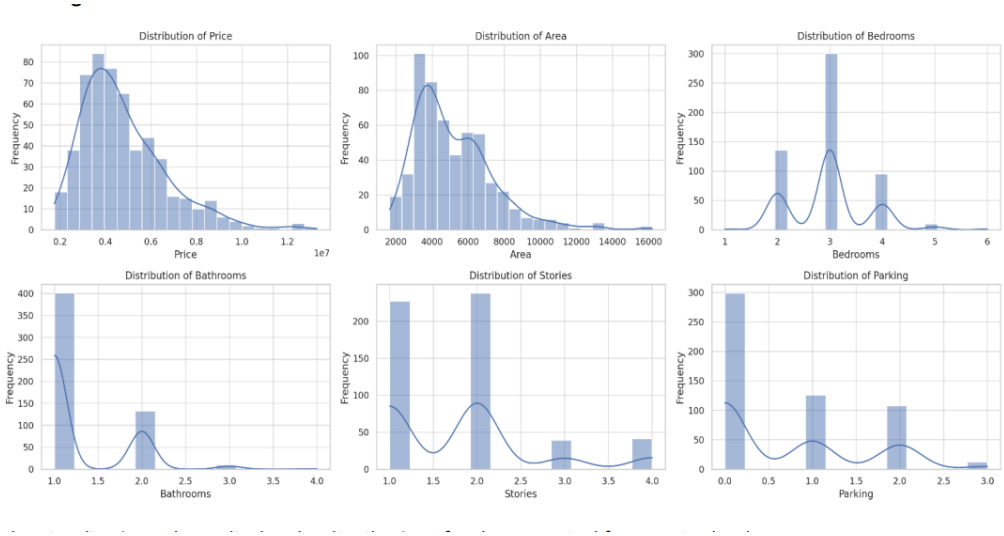


Figure5: Numerical values

6 Model :

In this part of the report, we will focus on the model used in the project.

Predicting house prices (Y) from a dataset like the Housing.csv involves using various features (explanatory variables) such as area, number of bedrooms, bathrooms, stories, etc. Here are the two statistical models that I am using for such predictions: Linear Regression and Random Forest Regressor.

6.1 Linear Regression Model:

6.1.1 Definition:

Linear regression is one of the most basic and commonly used statistical models for predicting a continuous outcome variable based on one or more predictor variables. The model assumes a linear relationship between the independent variables (explanatory variables) and the dependent variable (Y - house price in this context). Underlying Hypothesis: - H0 (Null Hypothesis): There is no relationship between the explanatory variables and the house price. (All the regression coefficients are equal to zero, except the

intercept.) - H1 (Alternative Hypothesis): At least one of the explanatory variables has a linear relationship with the house price. (At least one regression coefficient is not equal to zero.)

6.1.2 Implementation:

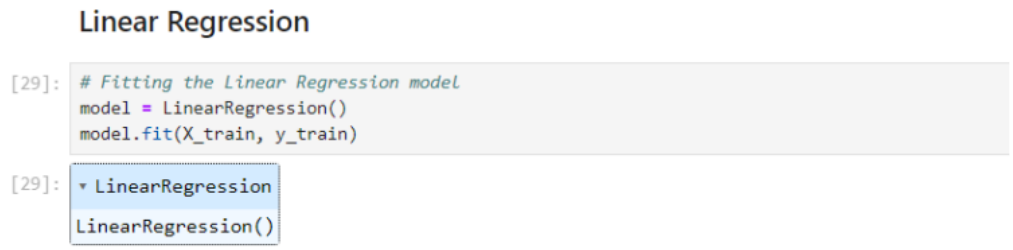


Figure 6: Linear regression

6.1.3 Prediction and results:

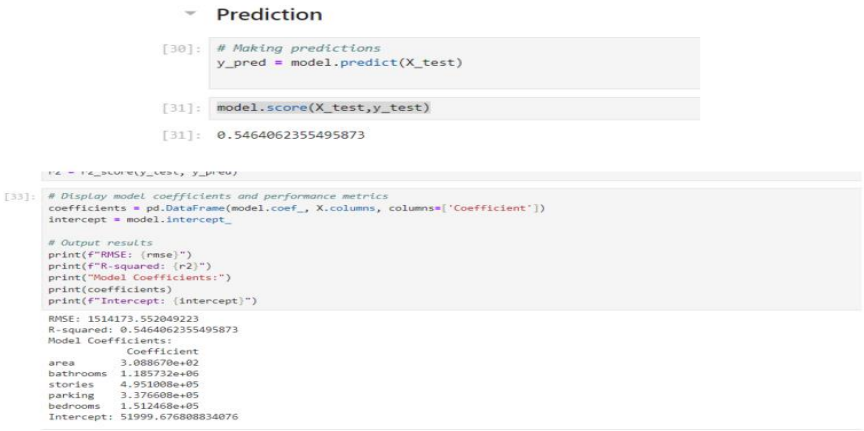


Figure 7: linear regression results

6.1.4 Evaluation:

We evaluate the model performance using r square and RMSE .

RMSE (Root Mean Squared Error): 1,514,173. This value indicates the average deviation of the predicted prices from the actual prices in the dataset. Given the price range, this RMSE suggests a moderate level of prediction error. R-squared: 0.546. This value indicates that approximately 54.6% of the variance in the house prices can be explained by the model. It's a measure of how well the observed outcomes are replicated by the model.

6.2 Random Forest Regressor model :

6.2.1 Definition:

Random Forest is an ensemble learning method based on decision tree algorithms. It operates by constructing a multitude of decision trees at training time and outputting the mean

prediction of the individual trees for regression tasks. Random Forest can handle non-linear relationships between features and the target variable and is robust to overfitting.

Model Concept:

The model builds multiple decision trees during training.

Each tree makes a prediction of the house price.

The final prediction is typically the average of all the tree predictions.

6.2.2 Implementation:

Figure 8: Random Forest Regressor

6.2.3 Evaluation :

The model has a moderate R-squared value and the RMSE is relatively high, which suggests that there is room for improvement.

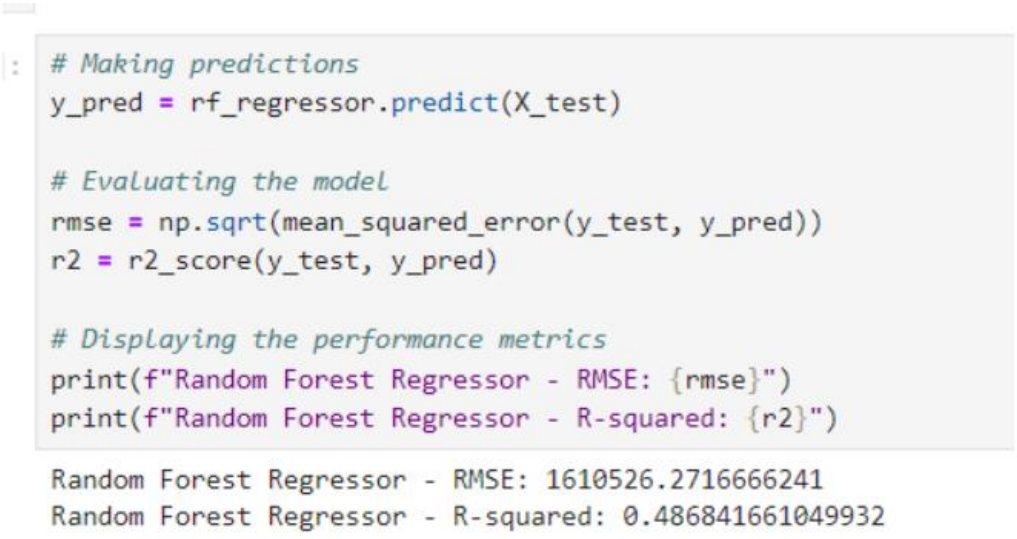


Figure 9: Random Forest results

6.3 Software:

The software I used in this project for predictive modeling and analysis is Python.

6.3.1 Software definition:

Python is a popular choice for data analysis and machine learning due to its extensive libraries and frameworks. Here are some key reasons why Python is used for this project:

Python is the primary programming language used in the notebooks, well-suited for data manipulation, statistical modeling. It can handle everything from data cleaning and visualization to advanced machine learning.

Rich Libraries: Python boasts powerful libraries such as Pandas for data manipulation, NumPy for numerical data, Scikit-learn for machine learning, and Matplotlib and Seaborn for data visualization. These tools greatly facilitate the process of data analysis and model building.

Ease of Learning and Use: Python's syntax is clear and intuitive, making it accessible to newcomers and convenient for experts. This ease of use speeds up the development process and enhances productivity.

6.3.2 Libraries Used:

Pandas: A library for data manipulation and analysis, providing data structures and operations for manipulating numerical tables and time series.

NumPy: Utilized for large array and matrix processing, with an assortment of mathematical functions to operate on these arrays.

Matplotlib and Seaborn: These libraries are used for creating static, interactive, and animated visualizations in Python. Matplotlib provides the basic framework, while Seaborn extends Matplotlib and makes generating complex visualizations easier.

Scikit-learn: This machine learning library is used for modeling. It includes tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities.

6.3.3 Techniques and Methodologies:

This project is providing a comprehensive look at the application of statistical and machine learning methodologies to predict housing prices, leveraging Python's extensive ecosystem for data analysis and machine learning.

We will look at some methodologies like sampling, feature selection, hyperparameter testing, and bootstrapping I implemented in this project.

6.3.3.1 Feature Selection :

- Correlation Analysis: We first compute the correlation matrix to identify how each feature relates to the price.

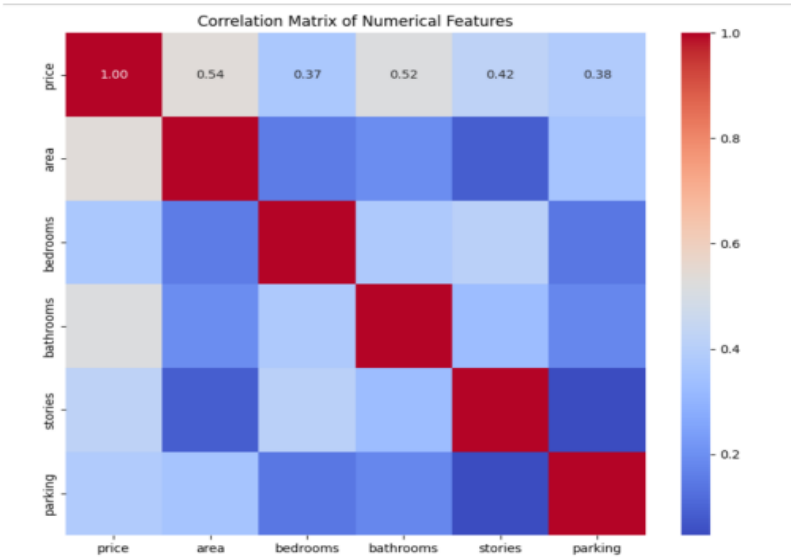


Figure10 : Correlation Matrix Of Numerical Features

- Selecting Features: we select features that have a correlation coefficient greater than 0.3 with the price.
- Selected Features: Based on the correlation analysis, we include features that are more directly associated with the house price.
  The positive values indicate that as these features increase, the price tends to increase as well.

Based on the correlation analysis, area, bedrooms, bathrooms, stories, and parking are selected as features for the Linear Regression model, as they show significant correlation with the house price.

6.3.3.2 Sampling :

We will perdorm 3 types of sampling which are : simple sampling , Stratified Sampling , and Cluster Sampling . Further , we are going to compare between them.

- Simple sampling : In random sampling, every member of the population has an equal chance of being selected. Illustration.

simple samling

```
# Identify the size of the dataset
dataset_size = len(dt)

# Choose a sample size - as a rule of thumb, let's take a sample size of around 30% of the dataset
sample_size = int(0.3 * dataset_size)

# Perform Simple Random Sampling
sample = dt.sample(n=sample_size, random_state=1)  # Using a fixed random state for reproducibility
```

```
print(sample)
```

```
       price   area  bedrooms  bathrooms  stories mainroad guestroom basement  \
62   7070000   6240         4          2        2      yes        no       no
247  4550000   8400         4          1        4      yes        no       no
142  5600000  10500         4          2        2      yes        no       no
107  6125000   6420         3          1        3      yes        no      yes
483  2940000   6615         3          1        2      yes        no       no
..       ...    ...       ...        ...      ...      ...       ...      ...
407  3465000   2145         3          1        3      yes        no       no
450  3150000   3450         3          1        2      yes        no      yes
542  1750000   3620         2          1        1      yes        no       no
408  3430000   4000         2          1        1      yes        no       no
80   6629000   6000         3          1        2      yes        no       no
```

Figure11: simple sampling

- Stratified sample: The population is divided into strata, and random samples are taken from each stratum.

```
# Stratified sampling based on the number of bedrooms
stratified_sample = dt.groupby('bedrooms', group_keys=False).apply(lambda x: x.sample(frac=0.3, random_state=1))

print(stratified_sample)
```

```
       price   area  bedrooms  bathrooms  stories mainroad guestroom basement  \
445  3150000   3450         1          1        1      yes        no       no
338  3885000   3780         2          1        2      yes       yes      yes
311  4123000   6060         2          1        1      yes        no      yes
459  3115000   3500         2          1        1      yes        no       no
146  5600000  10500         2          1        1      yes        no       no
..       ...    ...       ...        ...      ...      ...       ...      ...
173  5250000   5300         4          2        1      yes        no       no
34   8120000   6840         5          1        2      yes       yes      yes
536  1960000   3420         5          1        2       no        no       no
271  4340000   1905         5          1        2       no        no      yes
112  6083000   4300         6          2        2      yes        no       no
```

Figure12**:** Stratified sample

This stratified sample maintains the proportion of each subgroup (in this case, the number of bedrooms) relative to the original dataset. This stratified sample ensures that the proportion of houses with different numbers of bathrooms is similar to that in the full dataset.

In summary, the stratified sample provides a representative overview of the housing dataset across different features, which can be useful for analyzing the housing market, understanding property values, and building predictive models for house prices.

- Cluster Sampling:

In Cluster Sampling, The population is divided into clusters, and a random selection of clusters is chosen. All individuals within the chosen clusters are included in the sample A good candidate for clustering might be the area feature. We'll divide the area feature into clusters more dynamically, focusing on practical ranges that might reflect different types of properties (e.g., small houses, medium houses, large estates).

Count Records in Each Cluster: Before sampling, we'll count the records to ensure each cluster has a meaningful number of observations.

Sample Clusters: Randomly select a few clusters to sample from, ensuring a diverse representation.

Retrieve Sample: Collect all records from the selected clusters to form our cluster sample. T

his approach will provide a sample that is representative across different property sizes, which is a key variable in housing data.

Let's implement this strategy. The adjusted approach for cluster sampling based on the area feature of the housing dataset involved the following steps:

1- Defining Clusters Based on Quantiles of Area: We divided the area feature into practical clusters using quantiles to ensure a meaningful division of property sizes.
2- Sampling Clusters: We randomly selected clusters 2, 1, and 4 for our sample.
3- Retrieving the Sample: The sample consists of all records from the selected clusters.

- Comparison of clustering methods

When discussing sampling approaches, it typically refers to methods used for selecting a subset of data from a larger dataset for analysis.

1. Simple Random Sampling: - This method is the most basic form of sampling. It's unbiased and simple to implement but doesn't guarantee that the sample will be representative of subgroups within the population.

2. Stratified Sampling: - Stratified sampling ensures that characteristics of interest are represented proportionally in the sample. This can yield more accurate and reliable results than simple random sampling, especially when there are significant differences between strata. However, it requires more detailed information about the population upfront and can be more complex to implement.

3. Cluster Sampling: - Cluster sampling is often used when it is impractical or costly to conduct simple random or stratified sampling. It can be less precise than stratified sampling because not all clusters will be representative of the population. However, it can reduce costs and is more manageable when the population is large and spread over a wide geographical area.

6.3.3.3 Hypothesis testing :

We use the mean of the initial sample as our hypothesized population mean (μ_0) and perform a one-sample t-test with our simple random sample to see if its mean significantly differs from μ_0.

Power Calculation: We calculate the power of our test based on the observed effect size, sample size, and significance level.

This tells us the probability of correctly rejecting the null hypothesis if it is false. Required Sample Size: We estimate the sample size needed to detect a specified effect size (a shift in mean of 2 units) with a desired power (90%) and a Type I error rate of 5%. mu_0 is not a fixed value in the code snippet because it depends on the specific values of the price column in the Housing. In this case where seed =42

```
# Generate an initial sample and a simple random sample from the 'price' column
np.random.seed(42)  # For reproducibility
initial_sample = np.random.choice(price_data, size=100, replace=False)
simple_random_sample = np.random.choice(price_data, size=100, replace=False)

# Compute the mean of the initial sample
mu_0 = np.mean(initial_sample)
```

Figure13: mean computation

Result illustration :

```
# Display required sample size
print(f"Required sample size: {required_sample_size}")
print(f"Required mu_0:{mu_0}")
```

```
T-Statistic: -0.4930067943849101, P-Value: 0.623099551205845
Power of the test: 0.06389817384674232
Required sample size: 16075983977640.37
Required mu_0:4898285.0
```

Figure14: sample size results

6.3.3.4 Boostrappping

Bootstraping is a powerful statistical technique used to estimate the uncertainty of a statistic or a model parameter by using random sampling with replacement from the data. It is widely applicable across various statistical methods and is especially useful when the underlying distribution of the data is unknown or the sample size is small.

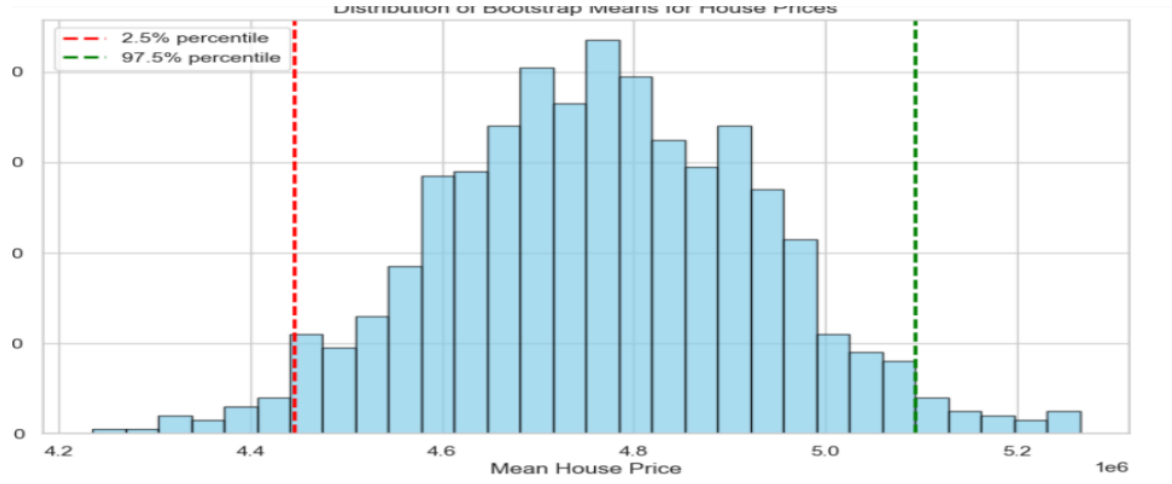
Implementation :

```python
# Sample size for each bootstrap sample
sample_size = 120

# Initialize an empty list to store bootstrap sample means for the 'price' column
bootstrap_means = []

# Number of bootstraps
num_bootstraps = 1000

# Perform bootstrapping for the 'price' column
for _ in range(num_bootstraps):
    # Randomly sample with replacement from the 'price' column
    bootstrap_sample = dt['price'].sample(n=sample_size, replace=True)

    # Calculate the mean price of the bootstrap sample
    bootstrap_mean = bootstrap_sample.mean()

    # Append the bootstrap mean to the list
    bootstrap_means.append(bootstrap_mean)

# Calculate the 95% confidence interval for the bootstrapped means
confidence_interval = np.percentile(bootstrap_means, [2.5, 97.5])

# Print the confidence interval
print("95% Confidence Interval of the House Prices:", confidence_interval)
```

```
95% Confidence Interval of the House Prices: [4444845.3        5093251.04166667]
```

Figure15: Bootstraping

Plotting



Figue16: Distribution of boostrap mean

Here's the plot showing the distribution of bootstrap means for house prices, with the 95% confidence interval highlighted. The dashed red line represents the 2.5th percentile, and the dashed green line represents the 97.5th percentile of the bootstrap means.

This visualization helps illustrate where the bulk of the sampled means lie and visually confirms the range of the confidence interval calculated from the bootstrap samples.

7. Result and interpretations

7.1 Linear Regression Model:

7.1.1 Results:

OLS Regression Results

Dep. Variable:	price	R-squared:	0.562
Model:	OLS	Adj. R-squared:	0.557
Method:	Least Squares	F-statistic:	110.4
Date:	Wed, 03 Apr 2024	Prob (F-statistic):	7.71e-75
Time:	07:38:56	Log-Likelihood:	-6707.6
No. Observations:	436	AIC:	1.343e+04
Df Residuals:	430	BIC:	1.345e+04
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	5.2e+04	2.56e+05	0.203	0.839	-4.52e+05	5.56e+05
area	308.8670	27.383	11.280	0.000	255.047	362.687
bathrooms	1.186e+06	1.32e+05	8.955	0.000	9.25e+05	1.45e+06
stories	4.951e+05	7.28e+04	6.797	0.000	3.52e+05	6.38e+05
parking	3.377e+05	7.05e+04	4.790	0.000	1.99e+05	4.76e+05
bedrooms	1.512e+05	8.66e+04	1.747	0.081	-1.89e+04	3.21e+05

Omnibus:	60.824	Durbin-Watson:	1.913
Prob(Omnibus):	0.000	Jarque-Bera (JB):	135.419
Skew:	0.740	Prob(JB):	3.93e-30
Kurtosis:	5.295	Cond. No.	2.64e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.64e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

Figure 17: Linear regression summary

7.1.1 Interpretation:

F-statistic: 110.4 Prob (F-statistic) or p-value for the F-test: 7.17e-75 The F-statistic is a measure of the overall significance of the regression model. The p value is zero for most intercepts Like the p-value for area is 0.000, indicating strong evidence against the null hypothesis, hence area is statistically significant. The p-value of this F-statistic, tests the null hypothesis that all the regression coefficients are equal to zero, meaning that the model has no explanatory power. A small p-value (typically ≤ 0.05) rejects the null hypothesis and indicates that your model provides a better fit than the intercept-only model.

7.2 Random Forest Regressor model

7.2.1 Results :

```
# Making predictions
y_pred = rf_regressor.predict(X_test)

# Evaluating the model
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)

# Displaying the performance metrics
print(f"Random Forest Regressor - RMSE: {rmse}")
print(f"Random Forest Regressor - R-squared: {r2}")
```

```
Random Forest Regressor - RMSE: 1610526.2716666241
Random Forest Regressor - R-squared: 0.486841661049932
```

Figure 18: Random Forest summary

7.2.2 Interpretation:

Root Mean Squared Error (RMSE): The RMSE for the Random Forest model is 1,610,526.27, which quantifies the average deviation between the predicted house prices and the actual values in the dataset. The relatively high RMSE suggests that the model's predictions are, on average, about 1.61 million units away from the actual sale prices. This indicates that the model's predictive accuracy could be improved.

R-squared (R^2) Value: The model has an R^2 of 0.4868, which means that approximately 48.68% of the variability in the house prices can be explained by the model's predictions. An R^2 below 0.5 shows that the model has not captured a substantial portion of the variance in the housing prices, which is also an indication that there is significant room for improvement.

8. Learning strategies:

8.1 Statistical Modeling:

Model Selection: The project likely involves choosing between different statistical models (e.g., Linear Regression and Random Forest) to find the one that best captures the complexities of the housing market data. This involves understanding the assumptions and limitations of each model.

Hypothesis Testing: Statistical tests are used to infer the significance of the models' outputs, such as determining if the model coefficients are significantly different from zero, which supports the model's explanatory power on house prices.

8.2 Machine Learning:

Cross-validation: Employed to assess the generalizability of the statistical models beyond the sample data. This helps in understanding how the model will perform in real-world settings.

Performance Metrics: Metrics such as RMSE and R-squared are used to quantify the model's accuracy and explainability. Learning to interpret these metrics helps in evaluating and comparing different models.

8.3 How to improve the performance:

The performance metrics suggest that while the Random Forest model has provided some insights into the data, it does not offer a highly accurate prediction of house prices in its current form. To enhance the model's performance, I could explore various strategies such as:

Adding more relevant features that could have a significant impact on house prices.

Tuning the hyperparameters of the Random Forest model more extensively, perhaps by using grid search or random search to find a better combination.

Handling any potential outliers or skewed distributions in the dataset that could be affecting model performance.

Using a more sophisticated ensemble method or trying different algorithms that may capture the complex patterns in the data better.

Regularization Techniques: To prevent overfitting, especially in complex models like Random Forest, regularization methods such as Lasso, Ridge, or Elastic Net might be applied. These techniques penalize the loss function by adding a complexity term that depends on the model coefficients.

8.4 Insights:

The iterative nature of model building and evaluation, involving continuously learning from model results, refining the model based on insights, and adapting to new data, is an essential learning strategy. It ensures that the predictive models remain relevant and accurate as new data becomes available or as market conditions change.

In conclusion, the learning strategies derived from this project not only enhance the technical proficiency in handling and modeling complex datasets but also improve strategic thinking about the application of these models in real-world scenarios. By incorporating advanced techniques in feature engineering, model tuning, and optimization, the project not only aims to achieve high accuracy but also ensures that the models are robust, interpretable, and easily adaptable to new challenges.

9. Conclusion :

The project successfully applied statistical and machine learning techniques to predict house prices, showcasing a substantial grasp of both theoretical and practical aspects of regression modeling. By using Linear Regression and Random Forest models, the project not only adhered to the academic goals of applying learned concepts but also provided practical insights into real estate pricing dynamics. Key achievements include:

9.1 Future Directions for Improving House Price Predictive Models

Incorporation of More Diverse Data Sources: To enhance model accuracy, additional data sources such as economic indicators, demographic statistics, and even social media trends could be incorporated to capture broader market influences.

Advanced Machine Learning Techniques: Employing more complex machine learning algorithms such as neural networks or ensemble methods that combine multiple models might improve prediction accuracy and better handle non-linear relationships.

Real-time Data Integration: Developing systems that can integrate and analyze real-time data will allow models to be more responsive to market changes, improving predictive accuracy.

Cross-Market Models: Developing models that can generalize across different markets by identifying universal factors influencing house prices can extend the applicability of predictive tools.

Ethical and Transparent Modeling: Ensuring that models do not inadvertently perpetuate biases and are transparent in their decision-making process is crucial, particularly as these models might influence significant economic decisions.

Continuous Model Evaluation and Updating: Regularly updating models to incorporate new data and techniques, and continuously evaluating model performance against current market conditions will ensure their ongoing relevance and accuracy.

10. References:

DataCamp. (n.d.). Random forests classifier Python. Retrieved May 2, 2024, from https://www.datacamp.com/tutorial/random-forests-classifier-python

GeeksforGeeks. (n.d.). Linear regression Python implementation. Retrieved May 2, 2024, from https://www.geeksforgeeks.org/linear-regression-python-implementation/

Holbrook, R. (n.d.). Feature engineering for house prices. Retrieved May 2, 2024, from https://www.kaggle.com/code/ryanholbrook/feature-engineering-for-house-prices

Python Software Foundation. (n.d.). Python 3.8.2 documentation. Retrieved May 2, 2024, from https://docs.python.org/3/library/index.html

Shejwal, Y. (n.d.). House price prediction system using ML algorithms. Medium. Retrieved May 2, 2024, from https://medium.com/@yash.shejwal20/house-price-prediction-system-using-ml-algorithms-e7ef83ded562

W3Schools. (n.d.). Python ML - Linear Regression. Retrieved May 2, 2024, from https://www.w3schools.com/python/python_ml_linear_regression.asp