PRICE PREDICTION AND RECOMMENDATION OF AIRBNB PROPERTY LISTINGS

Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

G.Nehemiah - 20B81A0548

G.Ishwarya - 20B81A0549

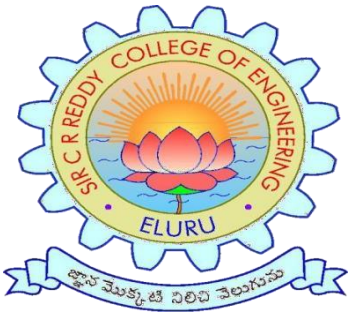
G.Pravalika - 20B81A0550

G.Ravali - 20B81A0551

G.Venkata Sai Monika - 20B81A0552

Under the Guidance of

V.SHARIFF



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SIR C R REDDY COLLEGE OF ENGINEERING

Approved by AICTE & Accredited by NBA

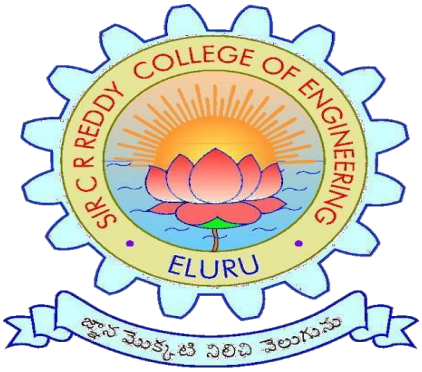
Affiliated to Jawaharlal Nehru Technological University,

KakinadaELURU-5340007

A.Y.2022-23

SIR C R REDDY COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the project report entitled "**PRICE PREDICTION AND RECOMMENDATIONOF AIRBNB PROPERTY LISTINGS**" being submitted by

G.Nehemiah	**- 20B81A0548**
G.Ishwarya	**- 20B81A0549**
G.Pravalika	**- 20B81A0550**
G.Ravali	**- 20B81A0551**
G.Venkata Sai Monika	**- 20B81A0552**

in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Sci-enceand Engineering to the Jawaharlal Nehru Technological University, Kakinada is a record of bonafide work carried out under my guidance and supervision.

Dr. M. Krishna M.Tech, Ph.D
Professor

Dr. A. YESUBABU M.Tech, Ph.D
Head of the Department

External Examiner

DECLARATION

I hereby declare that the Project entitled "**PRICE PREDICTION AND RECOMMENDATIONOF AIRBNB PROPERTY LISTINGS**" submitted for the B.Tech Degree is my original work and the Project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Place:ELURU

Date

PROJECT TEAM MEMBERS

G.Nehemiah - 20B81A0548

G.Ishwarya - 20B81A0549

G.Pravalika - 20B81A0550

G.Ravali - 20B81A0551

G.Venkata Sai Monika - 20B81A0552

ACKNOWLEDGEMENT

I express my sincere thanks to my principal **Dr. K. VENKATESWARA RAO**,Principal for providing the necessary infrastructure required for the project.

I would like to thank **Dr. A. YESU BABU**, Head of the Department of CSE, for providing the necessaryfacilities and his guidance in an efficient way for the completion of the project in the specified time.

I am grateful to **V.Shariff**, Assistant Professor, Department of CSE, Project guide for providing the necessary facilities and his guidance in the efficient completion of the project in a specified time.

I express my deep-felt gratitude to **Dr. N. Deepak**, Associate Professor, Department of CSE for his valuable guidance and unstinting encouragement enabled us to accomplish our project in time.

I am extremely grateful to my department staff members and teammates who helped me inthe successful completion of this project.

PROJECT TEAM MEMBERS

G.Nehemiah - 20B81A0548

G.Ishwarya - 20B81A0549

G.Pravalika - 20B81A0550

G.Ravali - 20B81A0551

G.Venkata Sai Monika - 20B81A0552

ABSTRACT

In today's rapidly evolving world of science, technology, and global connectivity, travel has reached unprecedented levels, with people exploring various destinations for business and personal needs. Finding suitable lodging away from home is a crucial aspect of this trip.

While hotels and motels have long been the go-to choose for travelers, the accommodation landscape has been transformed by Airbnb, initially known as Airbed and breakfast. This innovative company has become a favored alternative to traditional hotels, offering a uniqueexperience for travelers. Over time,

Airbnb has gained immense popularity, often surpassing conventional hotel options as the preferred choice for accommodation. The Airbnb business model is a two-sided marketplacethat serves both property owners and guests. Property owners offer their homes or rental properties on the platform, while guests book these properties for a specified period. Airbnb charges a service fee from both the guest and the property owner for each booking.

PRICE PREDICTION AND RECOMMENDATION OF AIRBNB PROPERTY LISTINGS

Prediction of the pricing using various regression methods like Linear Regression, Decision Tree Regression, Random Forest Regression between the input features like property, size, location, neighborhood and many more.

Clustering will be used for dimension reduction of the dataset and Cosine Similarity will be utilized to provide personalized recommendations for Airbnb listings based on user preferences. Additionally, we integrate hypothesis testing into our methodology, serving as a robust statistical tool for decision making processes, benefiting both property owners and customers in optimizing their listings and enhance the overall Airbnb experience.

Data sourcing: The data has been downloaded from Kaggle using the following link: https://www.kaggle.com/datasets/deeplearner09/airbnb-listings/data

It’s used for an Exploratory DataAnalysis study since we are taking data to provide analysis and recommendations.

Keywords: Clustering, Cosine Similarity, linear regression, Machine learning algorithms, Data visualization techniques.

TABLE OF CONTENT

LIST OF FIGURES

ABSTRACT

Airbnb (ABNB) is an online marketplace that connects people who want to rent out their homes with people looking for accommodations in specific locales. The company has come a long way since 2007, when its co-founders first came up with the idea to invite paying guests to sleep on an air mattress in their living room. According to Airbnb's latest data, it now has more than 7 million listings, covering some 100,000 cities and towns in 220-plus countries and regions worldwide.

The Airbnb business model is a two-sided marketplace that serves both property owners and guests. Property owners offer their homes or rental properties on the platform, while guests book these properties for specified period. The dataset primarily focusses on providing insights about the Airbnb model. Various data analysis models can be enforced to render more meaningful outcomes from Airbnb Listing Models. Implementing a machine learning method is a crucial step towards harnessing the potential of data driven decision making. By applying machine learning algorithms and data visualization techniques, the business strategy can be analyzed.

1.INTRODUCTION

The Airbnb business model is a two-sided marketplace that serves both property owners and guests. Property owners offer their homes or rental properties on the platform, while guests book these properties for a specified period. The dataset primarily focusses on providing insights about the Airbnb model. Various data analysis models can be enforced to render more meaningful outcomes from Airbnb Listing Models. Implementing a machine learning method is a crucial step towards harnessing the potential of data driven decision making. By applying machine learning algorithms and data visualization techniques, the business strategy can be analyzed.

Airbnb has gained immense popularity, often surpassing conventional hotel options as the preferred choice for accommodation. The Airbnb business model is a two-sided marketplace thatserves both property owners and guests. Property owners offer their homes or rental properties on the platform, while guests book these properties for a specified period. Airbnb charges a service fee from both the guest and the property owner for each booking. In this data set, each row represents a listing with details such as coordinates, neighborhood, host id, price per night,number of reviews, and so on. Purpose of using this dataset: Dataset primarily focusses on providing insights about the Airbnb model. Various data analysis models can be enforced to render more meaningful outcomes from Airbnb Listing Models.

This project revolves around predicting Airbnb listing prices, a critical task in the dynamic landscape of short-term property rentals. Airbnb, as a leading platform in the travel andhospitality industry, hosts a crowd of listings with diverse attributes.

The project's primary objective is to develop a robust machine learning model capable of accurately forecasting listing prices based on features like location, property type, and amenities.By doing so, the project aims to offer valuable insights to both hosts and potential guests, enabling hosts to optimize their offerings and aiding guests in making well-informed accommodation decisions

Additionally, visualizations will be employed to enhance data exploration and interpretation. Leveraging a comprehensive dataset obtained from Airbnb, the project will follow a systematicmethodology encompassing data preprocessing, exploratory data analysis, and modeldevelopment.

2.LITERATURE SURVEY

1.Airbnb Price Prediction Using Machine Learning and Sentiment Analysis - 2019

Link to reference papers: https://arxiv.org/pdf/1907.12665v1.pdf

- Pouya Rezazadeh Kalehbasti Stanford University pouyar@stanford.edu
- Liubov Nikolenko Stanford University liubov@stanford.edu

Research paper work: This paper aims to develop a reliable price prediction model using machine learning, deep learning, and natural language processing techniques to aid both the property owners and the customers with price evaluation given minimal available information about the property. Features of the rentals, owner characteristics, and the customer reviews will comprise the predictors, and a range of methods from linear regression to tree-based models, support-vector regression (SVR), K-means Clustering (KMC), and neural networks (NNs) will be used for creating the prediction model

Parts of the existing literature on property pricing focus on non-shared property purchase or rental price predictions. Previously, Yu and Wu [1] tried to implement a real estate price prediction using feature importance analysis along with linear regression, SVR, and Random Forest regression. They also attempted to classify the prices into 7 classes using Naive Bayes, Logistic Regression, SVC and Random Forest. They declared a best RMSE of 0.53 for their SVR model and a classification accuracy of 69% for their SVC model with PCA. In another paper, Ma et al. [2] have applied Linear Regression, Regression Tree, Random Forest Regression and Gradient Boosting Regression Trees to analyzing warehouse rental prices in Beijing. They concluded that the tree regression model was the best-performing model with an RMSE of 1.05 CNY/m2 -day

2. Predicting Airbnb Listing Price with Different models

- Haoqian Wang Physics, Nankai University, Tianjin, China - 2023

Reference paper link:

https://www.researchgate.net/publication/370698841

Research paper work:

Airbnb is a platform company that provides and directs connections between hosts and guests. People who have an open room or a vacant space can become a host on Airbnb and make it available to the world community. Airbnb offers hosts an easy way to turn otherwise wasted space into profitable space. Therefore, it is particularly necessary for hosts to forecast and analyze the price of the houses they own. Machine learning is the science of developing algorithms and statistical models. The regression model is a predictive modeling technique in machine learning. This technique is often used to discover causal relationships between variables, predictive analysis, and time series models. In this project, our goal is to predict Boston Airbnb listing prices through a variety of machine-learning methods. This paper chose four regression models, which are the random forest regression model, linear regression model, K-nearest neighbor regression model, and Gradient Boosting regression model. With one of the best regression models, this paper obtained R-squared values of 0.6593 in training and 0.7198 in testing on the Boston dataset.

Due to a large amount of information in the Airbnb dataset and a large amount of information that can be mined, analyzing the Airbnb dataset has become more and more popular among scholars in recent years. Yu and Wu [1] previously tried to predict real estate values using feature significance analysis, linear regression, SVR, and random forest regression. They attempted to categorize prices into 7 groups while using Random Forest, SVC, Logistic Regression, and Naive Bayes. They reported their PCA SVC model's best RMSE of 0,53 and classification accuracy of 69% for the SVR model. Li et al. [2] introduced a Multi-Scale Affinity Propagation technique in a different publication and demonstrated how it significantly increases the accuracy of rational price predictions. Nicolau and Wang [3] analyze Airbnb listings using Quantile Regression Analysis and Normal Least Squares to explore the elements influencing prices in the sharing economy. Masiero et al. [4] used quantile regression to examine the relationship between tourist attractions, vacation properties, and hotel rates. Recently, Lewis [5] made a prediction based on machine learning and deep learning on a London property market and found that XGBoost offers the best accuracy (R2 = 0.7274), which is superior to other Kaggle competitions.

3. EXISTING SYSTEM

- Current approaches to, Airbnb has gained immense popularity among travelers seeking accommodations globally. Consequently, Airbnb get extensive datasets from its listings that contain rich features that have captured the attention of researchers. These datasets offer potentially valuable information that can be extracted to greatly assist individuals and governments in making more informed decisions.
- Current exisiting system used the large amount of data sets with various columns and data listings with the different machine learning models
- Existing system Research paper : Pricing a rental property on Airbnb is a challenging task for the owner as it determines the number of customers for the place. On the other hand, customers have to evaluate an offered price with minimal knowledge of an optimal value for the property. This paper aims to develop a reliable price prediction model using machine learning, deep learning, and natural language processing techniques to aid both the property owners and the customers with price evaluation given minimal available information about the property. Features of the rentals, owner characteristics, and the customer reviews will comprise the predictors, and a range of methods from linear regression to tree-based models, support-vector regression (SVR), K-means Clustering (KMC), and neural networks (NNs) will be used for creating the prediction model.

- EXISTING SYSTEM MODELS AND ITS DRAWBACKS :

Existing system used more amount of data set listings that can lead to many disadvantages and more time taking in the training the data

Using large datasets in machine learning can offer numerous advantages, but it also comes with its own set of challenges and disadvantages. Here are some disadvantages of using large datasets in machine learning:

Computational Complexity:Processing and analyzing large datasets require significant computational resources, including memory and processing power. Complex algorithms may struggle to scale efficiently, leading to longer training times and increased computational costs.

Increased Training Time:

Training machine learning models on large datasets can be time-consuming, especially for algorithms that iterate over the entire dataset multiple times (e.g., gradient descent-based methods). Longer training times can slow down experimentation and model development cycles.

Storage Requirements:

Storing large datasets requires substantial disk space, which can be expensive, especially for organizations with limited storage resources. Managing and maintaining large-scale storage systems adds complexity and operational overhead.

Overfitting:

With large datasets, there is a risk of overfitting, where the model learns to memorize the training data rather than generalize to unseen data. Complex models trained on large datasets may capture noise or irrelevant patterns, leading to reduced generalization performance on new data.

Data Quality Issues:

Large datasets often contain noise, outliers, and missing values, which can degrade the quality of training data. Cleaning and preprocessing large datasets to remove noise and address data quality issues can be challenging and time-consuming.

Complexity of Model Interpretation:

Models trained on large datasets tend to be more complex, making them difficult to interpret and explain. Understanding the underlying patterns and relationships in large-scale data requires advanced visualization and interpretation techniques.

Sampling Bias:

When working with large datasets, there is a risk of unintentional sampling bias, where certain subsets of the data are overrepresented or underrepresented. Biased sampling can lead to skewed analysis and inaccurate conclusions, impacting the reliability and generalizability of results.

Data Privacy and Security Concerns:

Large datasets often contain sensitive or confidential information, raising concerns about data privacy and security. Safeguarding large-scale datasets against unauthorized access, breaches, and misuse requires robust security measures and compliance with data protection regulations.

Data Access and Distribution Challenges:

Accessing and sharing large datasets can be challenging, especially when dealing with distributed or heterogeneous data sources. Ensuring data accessibility, consistency, and integrity across distributed systems requires effective data management and coordination efforts.

Some drawbacks of this existing system used models with large data set listings

1.Ridge Regression:

Computational Complexity: With large datasets, the computational cost of solving the ridge regression optimization problem increases significantly, potentially leading to longer training times.

Memory Requirements: Storing the large dataset and the associated computation matrices (especially the covariance matrix) may require substantial memory resources.

Limited Flexibility: Ridge regression may not capture complex nonlinear relationships present in large datasets as effectively as more flexible models.

2.K-means Clustering with Ridge Regression:

Scalability Issues: K-means clustering may struggle to handle large datasets efficiently due to its iterative nature, as it requires computing distances between data points and cluster centroids for each iteration.

Computational Complexity: Integrating ridge regression with k-means clustering increases computational complexity, especially when optimizing the ridge regression parameters for each cluster.

Interpretability Challenges: The combined approach may lead to less interpretable models, making it harder to extract actionable insights from large datasets.

3.Support Vector Regression (SVR):

Computational Intensity: SVR requires solving a quadratic optimization problem, which becomes increasingly computationally intensive with larger datasets, leading to longer training times and increased memory requirements.

Kernel Selection Challenges: Selecting appropriate kernel functions and tuning kernel parameters becomes more challenging with large datasets, potentially leading to suboptimal model performance.

Sensitivity to Noise: SVR may be sensitive to noise and outliers in large datasets, impacting the robustness and generalization ability of the model.

4.Neural Network:

Training Time: Training neural networks on large datasets can be time-consuming, as it requires processing a vast amount of data through multiple layers of neurons.

Computational Resources: Large neural networks may require significant computational resources, including memory and processing power, which can be expensive and difficult to scale.

Overfitting Risk: Neural networks are prone to overfitting, especially with large datasets, where complex models may memorize noise or irrelevant patterns rather than capturing underlying relationships.

5.Gradient Boosting Tree Ensemble:

Memory Consumption: Gradient boosting tree ensembles can consume large amounts of memory, especially with deep trees or a large number of trees in the ensemble, which may limit scalability.

Training Time: Training gradient boosting models on large datasets can be time-consuming, particularly if using a high number of trees or optimizing hyperparameters through cross-validation.

Sensitivity to Noise: Gradient boosting models may be sensitive to noise and outliers in large datasets, potentially leading to overfitting or suboptimal performance if not properly addressed.

4.PROPOSED SYSTEM

- Our proposed system integrates research aimed to develop an effective predictive model for Airbnb prices using limited features and attributes, such as owner information, property specifications and reviews from customers. The objective of price optimization was to assistAirbnb hosts in determining the optimal price for their listings when sharing their homes.
- Various ML techniques, like Clustering, Linear Regression, Decision Tree, Random Forest between the input features like property, size, location and neighborhood and many more.
- Creating a proposed system for Airbnb prediction and recommendation involves designing algorithms and workflows to enhance user experience, increase booking rates, and improvelisting quality.
- In this project we have used a dataset with some attributes like host_id, host_name, reviews, ratings, etc… Airbnb uses ML algorithms to rank search results based on factors like user preferences, listing quality, location, availability, and pricing.
- Extract relevant features such as location, amenities, pricing, availability, host performance metrics, and user preferences.
- Clustering will be used for dimension reduction of the dataset and Cosine Similarity will be utilized for recommendations of Airbnb listings. We use Hypothesis testing as well, which isused for both property owners and customers for decision making.
- Merits of our proposed system

Data Visualization:

1.Insight Generation: Data visualization techniques help in gaining insights into the distribution, patterns, and relationships within the dataset, even with limited samples. Visualizations such as histograms, scatter plots, and heatmaps provide valuable insights into the data's characteristics.
2.Identifying Outliers: Visual inspection of data can help identify outliers or anomalies, which may require further investigation or preprocessing steps to handle effectively.
3.Communicating Findings: Visualizations aid in communicating findings and results to stakeholders, facilitating decision-making and understanding of the dataset's properties and trends.

Data Imputation and Cleaning:

1.Data Completeness: Imputation techniques fill in missing values in the dataset, improving the completeness of the data and ensuring that all samples can be utilized for analysis or modeling, even with limited data.
2.Data Quality: Cleaning techniques help identify and correct errors, inconsistencies, or outliers in the data, ensuring data quality and reliability for subsequent analysis or modeling tasks.
3.Improved Model Performance: By addressing missing values and errors, imputation and cleaning techniques contribute to improved model performance and generalization ability, especially with limited datasets where every sample is valuable.

Clustering:

1.Pattern Discovery: Clustering algorithms help identify underlying patterns or structures within the data, even with limited samples, by grouping similar data points together.
2.Insight Generation: Clustering provides insights into the inherent structure of the data, aiding in exploratory data analysis and hypothesis generation, which is particularly valuable when working with small datasets.
3.Anomaly Detection: Clustering techniques can help identify outliers or anomalies in the data, which may represent interesting or unusual instances that require further investigation.

Linear Regression:

1.Interpretability: Linear regression models are simple and easy to interpret, making them suitable for understanding the relationship between independent and dependent variables, even with limited data.
2.Quick Training: Linear regression models can be trained efficiently, enabling rapid experimentation and iteration in model development, which is beneficial when working with limited datasets.
3.Baseline Model: Linear regression serves as a baseline model for predictive modeling tasks, providing a straightforward approach for estimating the relationship between variables in limited datasets.

Decision Tree Regression:

1.Nonlinear Relationships: Decision tree regression models can capture nonlinear relationships between features and the target variable, allowing for more flexible modeling of complex patterns in limited datasets.
2.Interpretability: Decision trees are inherently interpretable, making them suitable for understanding the predictive factors even with limited data. They represent decision-making processes through a series of if-else rules.
3.Feature Importance: Decision tree models can rank features based on their importance in predicting the target variable, providing insights into the most influential variables in the dataset despite its size.

Random Forest Regression:

1.Robustness to Noise: Random forest regression is robust to noise and overfitting, making it suitable for modeling with limited datasets where the risk of overfitting is high.
Ensemble Learning: By aggregating predictions from multiple decision trees, random forest regression reduces variance and improves predictive performance, even with a small number of samples.
Feature Selection: Random forest regression automatically selects the most informative features, helping to focus on the most relevant variables in limited datasets and mitigating the curse of dimensionality.

5.REQUIREMENT ANALYSIS

Hardware requirements:

Processor	**:**	Intel Core
RAM	**:**	4GB

Software requirements:

Editor	**:**	Jupyter Notebook/ Vs code
Programming Languages	**:**	Python (3.11 version preferred)
Kerne	**:**	python3 (ipykernel)
Operating System	**:**	windows 11

6.IMPLEMENTATION

DATA SET:

This is the dataset that we have taken consist of fields like:

Id

Host_Name

Host_id

Neighbourhood

latitude

longitude

room_type

price

login Id

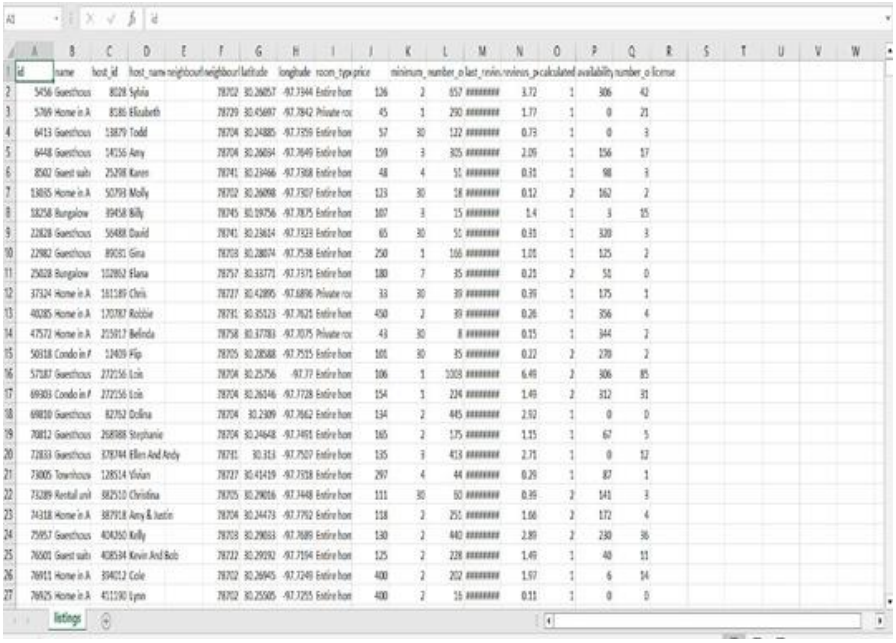
Host name

Minimum no.of reviews

Last review

Reviews per month

availability for 365 days



id	name	host_id	host_name	neighbourh	neighbourl	latitude	longitude	room_type	price	minimum_	number_o	last_revie	reviews_p	calculated	availability	number_o	license
5456	Guesthous	8028	Sylvia		78702	30.26057	-97.3944	Entire hom	126	2	657	########	3.72	1	306	42	
5769	Home in A	8186	Elizabeth		78729	30.45697	-97.7842	Private roo	45	1	290	########	1.77	1	0	21	
6413	Guesthous	13879	Todd		78704	30.24885	-97.7359	Entire hom	57	30	122	########	0.73	1	0	3	
6448	Guesthous	14156	Amy		78704	30.26054	-97.7649	Entire hom	159	3	305	########	2.09	1	156	17	
8502	Guest suits	25298	Karen		78741	30.23466	-97.7368	Entire hom	48	4	51	########	0.35	1	98	3	
13835	Home in A	50793	Molly		78702	30.26098	-97.7307	Entire hom	123	30	58	########	0.52	2	162	2	
18258	Bungalow	69458	Billy		78745	30.19756	-97.7875	Entire hom	107	3	15	########	1.4	1	3	15	
22828	Guesthous	56488	David		78741	30.23614	-97.7328	Entire hom	65	30	51	########	0.35	1	320	3	
22982	Guesthous	89081	Gina		78703	30.28074	-97.7538	Entire hom	250	3	166	########	1.01	1	125	2	
25628	Bungalow	102862	Elana		78757	30.33771	-97.7371	Entire hom	180	7	35	########	0.23	2	51	0	
37324	Home in A	161189	Chris		78727	30.42895	-97.6898	Private roo	33	30	39	########	0.39	1	175	1	
40285	Home in A	170787	Robbie		78731	30.35123	-97.7625	Entire hom	450	2	39	########	0.26	1	356	4	
47572	Home in A	215917	Belinda		78758	30.37783	-97.7075	Private roo	43	30	8	########	0.15	1	344	2	
56518	Condo in A	12409	Flip		78705	30.28588	-97.7515	Entire hom	101	30	35	########	0.22	2	270	2	
57187	Guesthous	272156	Lois		78704	30.25756	-97.77	Entire hom	106	1	1003	########	6.49	2	306	85	
69303	Condo in A	272156	Lois		78704	30.26546	-97.7728	Entire hom	154	1	224	########	1.49	2	312	31	
69810	Guesthous	82762	Dolina		78704	30.2309	-97.7662	Entire hom	134	2	445	########	2.92	1	0	0	
70812	Guesthous	268988	Stephanie		78704	30.24648	-97.7491	Entire hom	165	2	175	########	1.15	1	67	5	
72833	Guesthous	378744	Ellen And Andy		78731	30.313	-97.7507	Entire hom	135	3	413	########	2.71	1	0	12	
73005	Townhous	128514	Vivian		78727	30.41419	-97.7358	Entire hom	297	4	44	########	0.29	1	87	1	
73289	Rental unit	882510	Christina		78705	30.29016	-97.7448	Entire hom	111	30	80	########	0.59	2	141	3	
74318	Home in A	887918	Amy & Justin		78704	30.24473	-97.7702	Entire hom	118	2	251	########	1.66	2	172	4	
75957	Guesthous	404260	Kelly		78703	30.29063	-97.7689	Entire hom	130	2	442	########	2.89	2	230	36	
76001	Guest suits	408594	Kevin And Bob		78722	30.29092	-97.7194	Entire hom	125	2	228	########	1.49	1	40	11	
76911	Home in A	394012	Cole		78702	30.26945	-97.7249	Entire hom	400	2	202	########	1.97	1	6	14	
76925	Home in A	451190	Lynn		78702	30.25585	-97.7255	Entire hom	400	2	16	########	0.11	1	0	0	

6.1.1Data set before cleaning

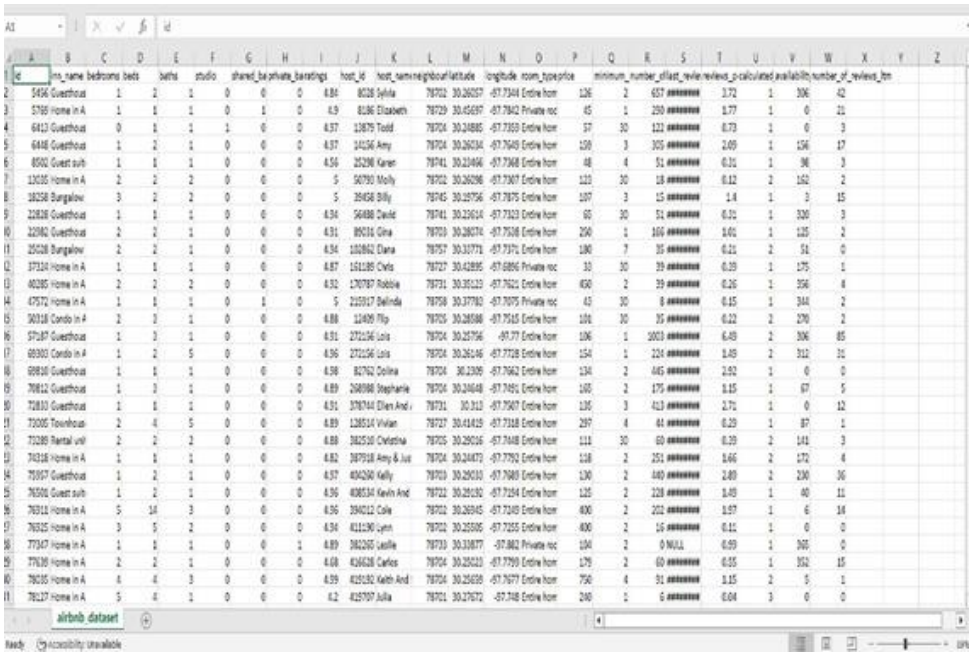
Cleaning the dataset helps ensure that the data is accurate, consistent, and reliable, leading to improved data quality. Removing errors, duplicates, and inconsistencies enhances the integrity of the dataset, making it more suitable for analysis and modeling tasks.

Clean datasets facilitate better interpretability of machine learning models. When the dataset is free from errors and inconsistencies, it is easier to understand the relationships between variables and interpret the model's predictions, making it more actionable for decision-making.

Cleaning the dataset lays the foundation for effective feature engineering. By identifying and addressing missing values, outliers, and irrelevant variables, cleaning helps create informative features that capture the underlying patterns and relationships in the data, improving model performance.

Cleaning the dataset ensures compliance with data quality standards and regulatory requirements. By adhering to data cleaning best practices, organizations can maintain data integrity, security, and privacy, reducing the risk of regulatory violations and data breaches.

We have cleaned the dataset inorder to clear the null values by Imputation. After cleaning the obtained dataset



id	ms_name	bedrooms	beds	baths	studio	shared_be	private_ba	ratings	host_id	host_nam	neighbourl	latitude	longitude	room_type	price	minimum_	number_of	last_revie	reviews_p	calculated	availability	number_of_	reviews_ltm
5456	Guesthous	1	2	1	0	0	0	4.84	8038	Sylvia		78701	30.26057	-97.7344	Entire hom	126	2	657	########	3.72	1	306	42
5785	Home in A	1	1	1	0	1	0	4.9	8186	Elizabeth		78729	30.45697	-97.7842	Private roc	45	1	250	########	1.77	1	0	21
6413	Guesthous	0	1	1	1	0	0	4.57	13875	Todd		78704	30.24885	-97.7359	Entire hom	57	30	122	########	0.73	1	0	3
6448	Guesthous	1	2	1	0	0	0	4.87	14156	Amy		78704	30.26034	-97.7649	Entire hom	159	3	305	########	2.09	1	156	17
8502	Guest suit	1	1	1	0	0	0	4.56	25298	Karen		78741	30.23466	-97.7368	Entire hom	48	4	51	########	0.31	1	98	3
12035	Home in A	2	2	2	0	0	0	5	56793	Molly		78702	30.26096	-97.7307	Entire hom	123	30	18	########	0.12	2	162	2
18258	Bungalow	3	2	2	0	0	0	5	39458	Billy		78745	30.18756	-97.7875	Entire hom	107	3	15	########	1.4	1	3	15
22838	Guesthous	1	1	1	0	0	0	4.94	56488	David		78741	30.22614	-97.7323	Entire hom	60	30	51	########	0.31	1	330	3
22982	Guesthous	2	2	1	0	0	0	4.91	89016	Gina		78703	30.28074	-97.7538	Entire hom	250	1	166	########	1.01	1	135	2
25028	Bungalow	2	2	1	0	0	0	4.94	102862	Elana		78757	30.33771	-97.7371	Entire hom	180	7	35	########	0.21	2	51	0
37314	Home in A	1	1	1	0	0	0	4.87	161185	Chris		78727	30.42895	-97.6896	Private roc	30	30	39	########	0.39	1	175	1
40285	Home in A	2	2	2	0	0	0	4.92	170787	Robbie		78731	30.35123	-97.7621	Entire hom	450	2	39	########	0.26	1	356	8
47572	Home in A	1	1	1	0	1	0	5	215917	Belinda		78758	30.37782	-97.7075	Private roc	43	30	8	########	0.15	1	344	2
50318	Condo in A	2	3	1	0	0	0	4.88	12409	Tip		78705	30.28588	-97.7515	Entire hom	101	30	35	########	0.22	2	270	2
57187	Guesthous	1	3	1	0	0	0	4.91	272156	Lois		78704	30.25756	-97.77	Entire hom	106	1	1003	########	6.49	2	306	85
60303	Condo in A	1	2	5	0	0	0	4.96	272156	Lois		78704	30.26146	-97.7728	Entire hom	154	1	224	########	1.49	2	312	31
69810	Guesthous	1	1	1	0	0	0	4.98	82762	Dolina		78704	30.2305	-97.7662	Entire hom	134	2	445	########	2.92	1	0	0
70812	Guesthous	1	3	1	0	0	0	4.89	298988	Stephanie		78704	30.24648	-97.7491	Entire hom	165	2	175	########	1.15	1	67	5
72810	Guesthous	1	1	1	0	0	0	4.91	378744	Ellen And		78731	30.313	-97.7507	Entire hom	135	3	413	########	2.71	1	0	12
73005	Townhous	2	4	5	0	0	0	4.89	128514	Vivian		78727	30.41419	-97.7318	Entire hom	297	4	44	########	0.29	1	87	1
73289	Rental unit	2	2	2	0	0	0	4.88	382510	Christina		78705	30.29216	-97.7448	Entire hom	113	30	69	########	0.29	2	141	3
74318	Home in A	1	1	1	0	0	0	4.82	387918	Amy & Jus		78704	30.24473	-97.7792	Entire hom	118	2	251	########	1.66	2	172	4
75957	Guesthous	1	2	1	0	0	0	4.97	404260	Kelly		78703	30.29033	-97.7683	Entire hom	130	2	440	########	2.83	1	230	36
76506	Guest suit	1	2	1	0	0	0	4.96	408534	Kevin And		78722	30.28192	-97.7254	Entire hom	125	2	228	########	1.49	1	40	11
76911	Home in A	5	14	3	0	0	0	4.96	394012	Cole		78702	30.26945	-97.7249	Entire hom	400	2	202	########	1.57	1	6	14
76935	Home in A	3	5	2	0	0	0	4.94	411190	Lynn		78702	30.25595	-97.7255	Entire hom	400	2	16	########	0.11	1	0	0
77347	Home in A	1	1	1	0	0	1	4.89	381265	Lasilla		78733	30.33877	-97.862	Private roc	104	2	0	NULL	0.00	1	360	0
77638	Home in A	2	2	1	0	0	0	4.68	416628	Carlos		78704	30.23023	-97.7759	Entire hom	175	2	60	########	0.55	1	352	15
78035	Home in A	4	4	3	0	0	0	4.89	425192	Keith And		78704	30.25698	-97.7677	Entire hom	750	4	91	########	0.15	2	5	1
78127	Home in A	5	4	1	0	0	0	4.2	425797	Julia		78701	30.27672	-97.748	Entire hom	240	1	6	########	0.04	3	0	0

6.1.2 Dataset after cleaning

6.1. DATA PROCESSING

Identifying the data:

Data can be classified in either qualitative or quantitative data. further classified into, **Categorical-** can be Nominal, ordinal, binary. Categorical data is used to classify items or characteristics into groups based on specific attributes or qualities.

Binary: These columns have only two unique values, typically representing binary categories suchas0 and 1. The code identifies columns with two unique values and categorizes them as categorical binary. No operations can be done though the binary values are numerical like 1 and 0 so these are defined under Categorical data.

Discrete data: These are distinct or separate values. Discrete data can be counted. They are whole numbers or integers. The values cannot be divided into subdivisions into smaller pieces. Examples: Total students in a class, number of products.

Continuous data: These are numeric values that form a continuous range and can be measured. They are in the form of fractions or decimal. The values can be divided into subdivisions into smallerpieces. Examples: temperature readings, age

Based on the above definitions we divided the columns in the data set into following categories,

Categorical:

inn_name: This column contains the non-numeric (string) values represented as object datatype.

host_name: The numeric values here represent the unique id information of the litsing's host.

id: An integer value representing the id column in the dataset refers to numerical categorical-data.identifier.

host_id: An integer value representing the id column in the dataset refers to numericalcategoricaldata. identifier.

neighbourhood: The numeric values here represent the neighborhood values and are consideredtobethe categorical values.

room_type: Represents the type of room and is likely categorical but may contain textdescriptions.

last_review: contains the date values in y yyy-mm-dd format that are denoted as objects.

Binary:

studio: An integer datatype here reflects the binary values 0 and 1 which can be considered underthenumerical categorical value as False or True respectively.

shared_bath: Shows the binary values of 0 and 1 representing a numerical categorical value asFalseor True respectively.

private_bath: Shows the binary values of 0 and 1 representing a numerical categorical value.

Continuous:

ratings: Continuous numerical values representing ratings which are of float datatype.

latitude and longitude: Continuous numerical values representing the geographical coordinates which are of float datatype.

reviews_per_month: Continuous numerical values representing the average number of reviews per month

Dscrete or Numerical:

bedrooms: Having integer values denoting the numerical discrete values.beds: Having integer values representing the numerical discrete data. baths:Having integer values representing the numerical discrete data.

minimum_nights: Having integer values representing the numerical discrete

datanumber_of_reviews: Having integer values representing the numerical discrete values.

calculated_host_listings_count: Having integer values representing the numerical discrete data.

availability_365: Having integer values representing the numerical discrete data.

number_of_reviews_ltm: Integer values representing counts or quantities.

price: Integer datatype representing the price which is of numerical discrete data.

IMPUTATION PROCESS

Missing data is a common issue in real-world datasets and can arise for various reasons, such as data entry errors, intentional omission etc. To overcome this issue Imputation technique is used in data cleaning and preprocessing.

It involves filling in missing values with median values for the numerical columns and mode values for the categorical columns. Through imputation the data integrity is preserved which further helps in more robust analysis and modeling.

Here, imputation is performed which is based on replacing the null values in the dataset using median value.

Overview of the missing value data:

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

```
id                                  0
inn_name                            0
bedrooms                            0
beds                                0
baths                               0
studio                              0
shared_bath                         0
private_bath                        0
ratings                          4222
host_id                             0
host_name                           2
neighbourhood                       0
latitude                            0
longitude                           0
room_type                           0
price                               0
minimum_nights                      0
number_of_reviews                   0
last_review                      3103
reviews_per_month                3103
calculated_host_listings_count      0
availability_365                    0
number_of_reviews_ltm               0
dtype: int64
```

6.1.3 Overview of the missing value data

Handling the null and missing values:

1.The 4222 null values in the ‘ratings’ column are imputed with the median value of 4.89.

2.The 3103 null values in the 'reviews per month' column are imputed with the median valueof 0.99.

3.The null values in the ‘last review’ column are not imputed as we understand that some properties may not have received any reviews, either due to being relatively new to the platform or potentially being situated in remote locations or may the amenities be up to the mark etc. external reasons. Additionally, guests may have visited, but for various reasons,they chose not to leave a review. We believe that replacing the blank values with NULL accurately reflects these practical scenarios.

4.The two null values in the host name column are imputed with ‘Unknown’ as there are few

columns which are named as Unknown already, so we took that as reference and same waywe fillethe two nulls with Unknown.

Data Extraction:

A function utilizing regular expressions was implemented to extract numerical valuesassociated with each room type. The extracted data was stored in dictionaries for further processing.

Data Enrichment and Column Creation: The extracted data was utilized to create newcolumns in the Data Frame, 'bedrooms', 'beds', 'baths': Columns were initiated with default values and updated using extracted numerical data.

1.'studio', 'shared_bath', 'private_bath': Binary columns indicating presence or absence ofthese room types based on extracted indices.

2.'ratings': Extracted numeric ratings were added to a new column after converting '★'symbols to numeric values.

Data Refinement and Transformation:

The 'ratings' column was further refined by converting symbols to numeric values, while the 'last_review' column was converted to a datetime format for better analysis.

VISUALIZATION:

In our project, we leverage visualizations as powerful tools to enhance the clarity and interpretability of our findings. Through charts, graphs, and maps, we aim to simplify complex patterns and trends and draw the meaning full insights from the visualizations. Visual representations of geographical distribution, property characteristics, and price trends will not onlyenhance interpretability but also empower hosts and users to make informed decisions.

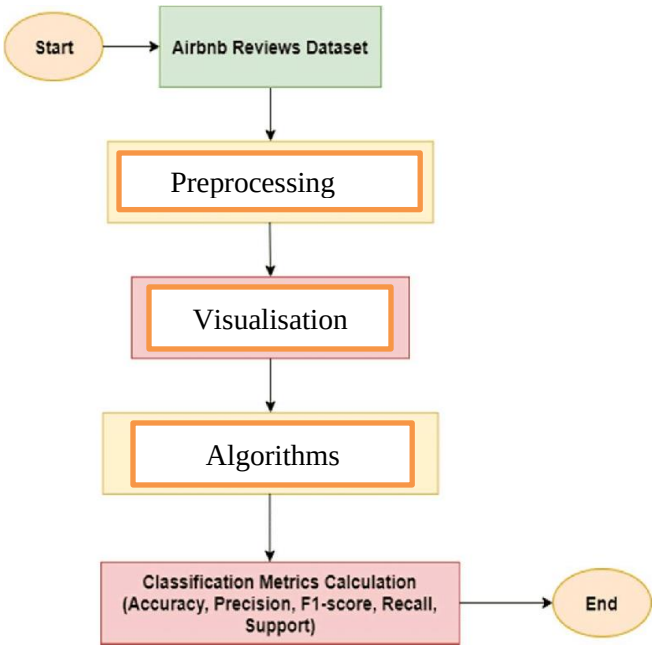
In our project we used Tableau, Python, Microsoft fabric in which Power BI is incorporated are used as visualization tools to represent the data visually.
Tableau is a versatile data visualization tool that allows users to create interactive and shareable dashboards. Its user-friendly interface makes it accessible for both technical and non-technical users, enabling the creation of compelling visualizations without extensive coding.

Python, with libraries like Matplotlib, Seaborn serves as a robust programming language for data visualization. Python's flexibility and extensive libraries make it a preferred choice for customizing visualizations and creating complex plots. Its integration with data analysis and machine learning tools further enhances its capabilities.

Microsoft Fabric, also known as Fluent UI, is a design system developed by Microsoft to create consistent and visually appealing user interfaces across different Microsoft applications. While not a standalone visualization tool, it plays a crucial role in maintaining a cohesive and polished design language within applications and contributes to a seamless user experience. Here Microsoft Power BI is incorporated as part of the Power Platform in fabric. It is a business analytics tool that facilitates interactive visualizations and business intelligence with an intuitive drag-and-drop interface. It seamlessly integrates with various data sources, making it convenient for users to transform data into insightful visuals, reports, and dashboards. We explored it as it is one of the emerging platforms which will help us to learn from the new applications.

UML DAIGRAMS

1.State flow diagram



6.1.4 State flow diagram

The process of implementing data pre-processing, visualization, model training, and evaluation of classification metrics involves breaking down the process into distinct states and transitions. Here's how you could represent this process in a Stateflow diagram:

Start State (Start): This is the initial state where the process begins.

Data Pre-Processing State (Preprocess): This state represents the data pre-processing step. Here, you perform tasks such as cleaning, transforming, and organizing the raw data.

Visualization State (Visualize): After pre-processing the data, you move to the visualization state where you create visualizations to explore the data and gain insights.

Model Training State (Train): In this state, you train machine learning models using algorithms like Linear Regression, Decision Tree Regression, Random Forest Regression, and Clustering.

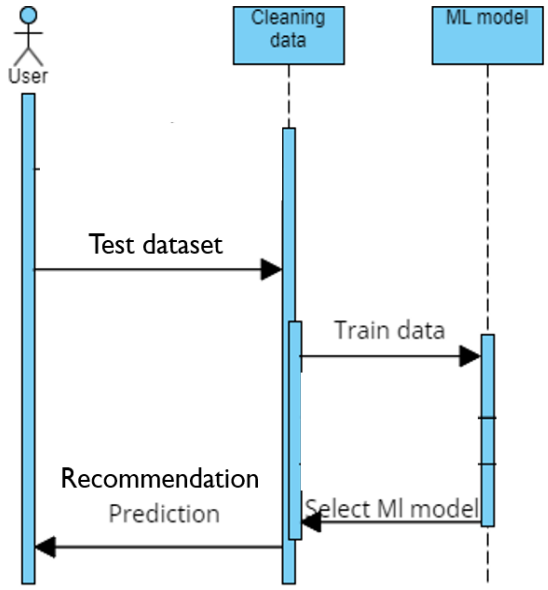
Evaluation State (Evaluate): After training the models, you transition to the evaluation state where you calculate classification metrics such as Accuracy, Precision, F1-Score, etc

End State (End): This is the final state where the process ends.

Here's how these states could be represented in a Stateflow diagram:

[Start] –> [Preprocess] –> [Visualize] –> [Train] –> [Evaluate] –> [End]

2.Sequence diagram



6.1.5 Sequence diagram

In this sequence diagram:

The User initiates each step of the process.

The Data Pre-Processing Module, Visualization Module, Model Training Module, and Evaluation Module are depicted as separate participants/modules responsible for carrying out their respective tasks

Each module performs its tasks as requested by the user.Communication between the user and each module is shown with arrows indicating the flow of control.

The sequence of actions is shown chronologically, with each step following the completion of the previous one.

Participant: User

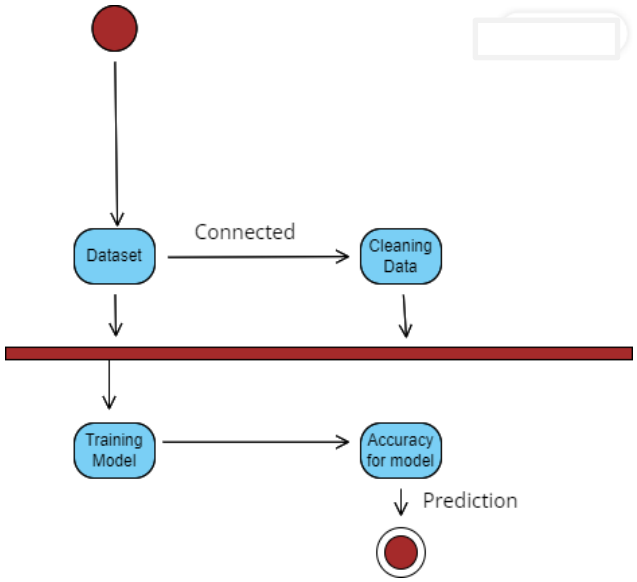
Participant: Data Pre-Processing Module

Participant: Visualization Module

Participant: Model Training Module

Participant: Evaluation Module

3.Activity diagram



6.1.6 Activity diagram

In this activity diagram:

The Start node represents the beginning of the process.

Each rectangular box represents an action or task to be performed.

The arrows indicate the flow of control between activities.

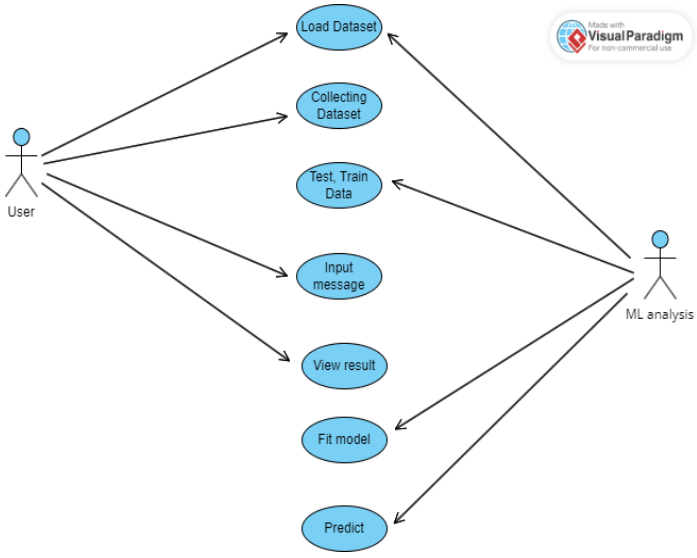
The diamond-shaped decision nodes represent points where the flow of control can take different paths based on conditions.

The End node represents the conclusion of the process.

Process:

Start -> Data Pre-Processing -> Visualization -> Model Training -> Evaluation -> End

4.Use case diagram



6.1.7 Use case diagram

Actor: User

Use Cases:

• Data Pre-Processing

The user initiates the process of preparing raw data for analysis by performing tasks such as cleaning, transforming, and organizing the data.

• Visualization

The user generates visual representations of the pre-processed data to explore patterns, relationships, and distributions within the dataset.

• Model Training

The user trains machine learning models using algorithms such as Linear Regression, Decision

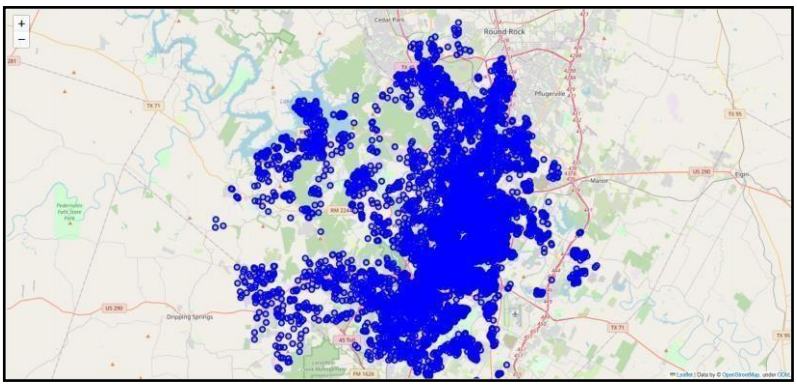
Tree Regression, Random Forest Regression, and Clustering on the pre-processed data to learn patterns and make predictions.

• Evaluation

The user evaluates the performance of the trained models by calculating classification metrics such as Accuracy, Precision, F1-Score, etc., to assess their effectiveness in making predictions.

SOME OF THE VISUALIZATIONS WE IMPLEMENTED ARE:

Geographical distribution of Airbnb Listings- Geographical distribution of Airbnb listings have been shown through the folium maps in Python. Folium is used to create a geographical map with markers representing the locations of Airbnb listings. Key features include markers where each marker on the map represents the location of an Airbnb listing. This visualization provides an overview of the geographical distribution of listings. We canzoom in andzoom out to explore different geographical areas.



6.2.1.Geographical distribution of Airbnb listings

Relationship between Ratings and Price features- The scatter plot shows the relationshipbetween ratings and price. This helps in understanding how customers perceive therelationship between the price and rating of Airbnb listings. We can see that higher ratingsare concentrated between price range from 0 to 2500.

6.2.2.Relationship between Ratings and Price features

6.3. MACHINE LEARNING MODELS:

Linear Regression

Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable we want to predict is called the dependent variable. The variable we are using to predict the other variable's value is called the independent variable.

This form of analysis estimates the coefficients of the linear equation, involving one or more independent variables that best predict the value of the dependent variable. Linear regression fits a straight line or surface that minimizes the discrepancies between predicted and actual output values. There are simple linear regression calculators that use a “least squares” method to discoverthe best-fit line for a set of paired data. We then estimate the value of X (dependent variable) fromY(independent variable).

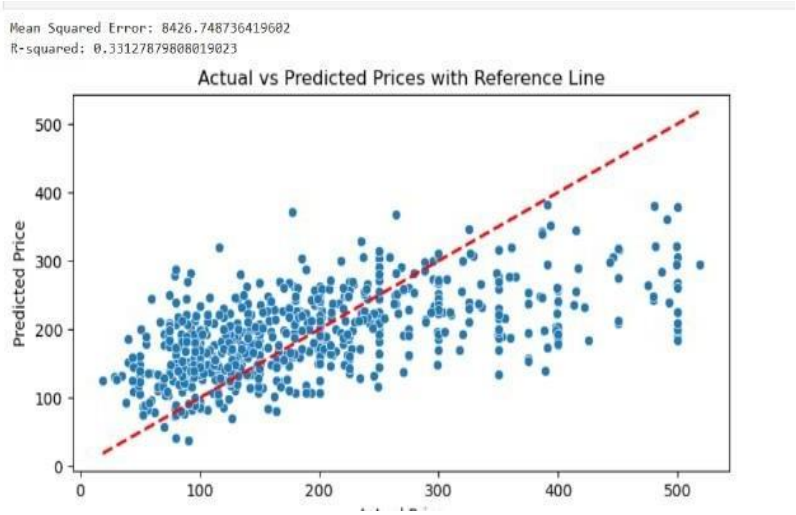
Implementation of Linear Regression for Price Prediction Model -

Through Python code we have imported necessary libraries and loaded the Airbnb dataset. It then calculates the Pearson correlation coefficients between the features and the target variable, 'price.' The top three most positively and negatively correlated features are identified. As negatively correlated coefficients tending towards zero don’t have significant impact on target variable, we have taken top 3 positively correlated features which are closer to 1. Subsequently, a linear regression model is created using the three most positively correlated features. The model is trainedon the training data, and predictions are made on the test set. The linear regression model assumesa linear relationship between the selected features and the target variable, 'price’.

Feature importance is derived from the correlation analysis, where the three most positively correlated features with the target variable, 'price,' are identified. However, in the context of linearregression, the emphasis is on identifying the features that contribute most to the prediction model.In this case, the features selected for the linear regression model are deemed important as they arebelieved to have a strong linear relationship with the target variable.

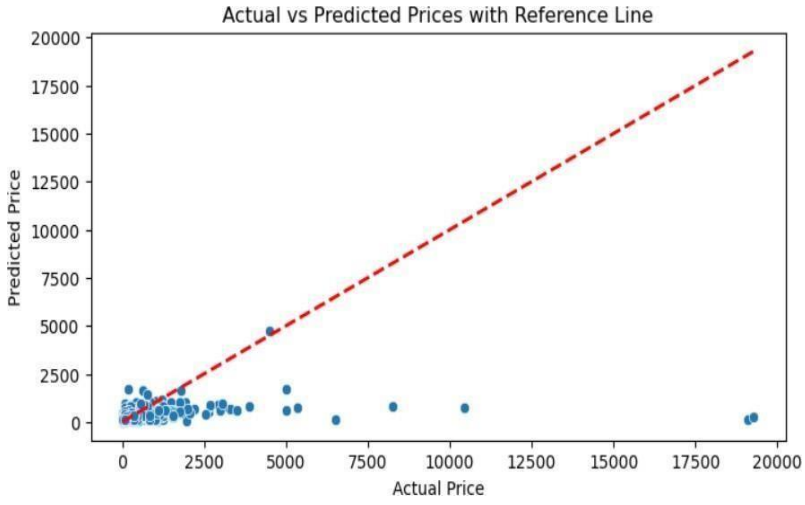
Calculated the **Mean Squared Error (MSE)**, a measure of the average squared difference betweenthe predicted and actual values on the test data. Lower MSE values indicate better modelperformance. Additionally, the **R-squared value**, which measures the proportion of the variance in the target variable, is explained by the model. A higher R-squared value signifies a better fit. The provided output

shows the MSE and R-squared values for the linear regression model, providing insights into its predictive accuracy and overall goodness of fit.



6.3.1.Accuracy of Linear Regression

Scatter plot is generated for comparing the actual prices in the test data against the predicted prices from the linear regression model. The reference line (red dashed line) represents a perfect prediction scenario where actual and predicted values are equal. The scatter plot allows for a visual assessment of how well the linear regression model aligns with the actual prices, providinginsights into the model's performance and potential areas for improvement.



6.3.2.Scatterplot of linear regression

Advantages of Linear Regression

1.Linear Regression is simple to implement and easier to interpret the output coefficients.

2.When you know the relationship between the independent and dependent variable have a linear-relationship, this algorithm is the best to use because of its less complexity compared to other algorithms.

3.LinearRegression is susceptible to overfitting,but it can be avoided using some dimensionality reduction techniques, regularization (L1 and L2) techniques and cross- validation.

4.Linear regression gives a quantitative degree of the quality and direction of the relationship between factors.

CODE:

Linear Regression:

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error
import seaborn as sns
import matplotlib.pyplot as plt

# Read the dataset
df = pd.read_csv('airbnb_dataset.csv')

# Drop any non-numeric columns or columns that cannot be converted to float
df = df.select_dtypes(include=['float64', 'int64'])

# Identify and handle outliers using the interquartile range (IQR) method
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
df = df[~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis=1)]

# Separate features and target variable
```

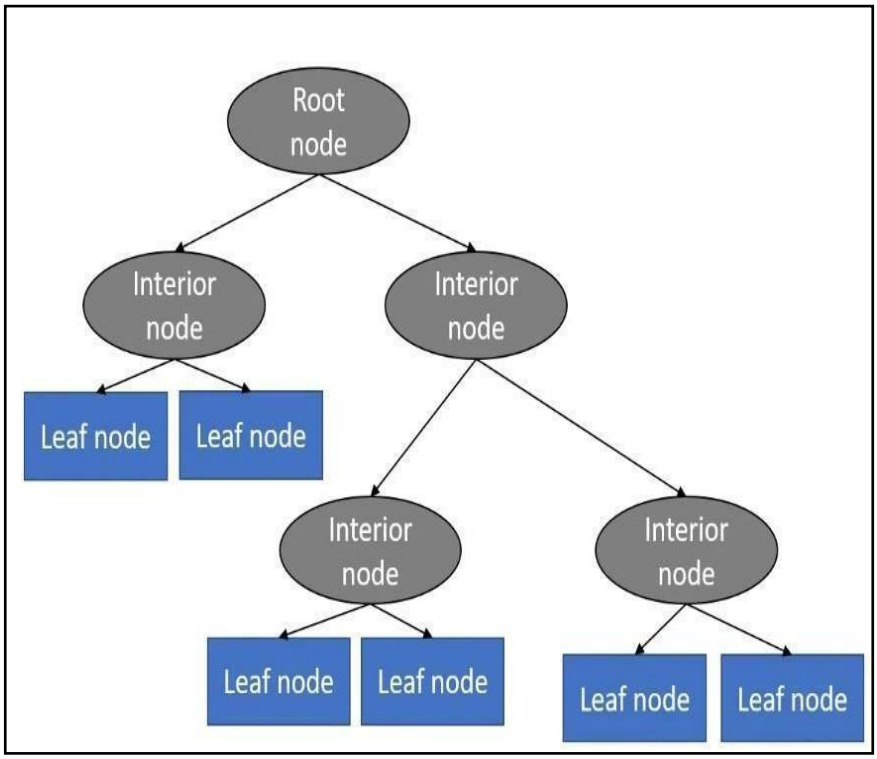
```
X = df.drop(['price'], axis=1)
Y = df['price']

# Normalize the features using StandardScaler
scaler = StandardScaler()
X_normalized = scaler.fit_transform(X)

# Split the normalized dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_normalized, Y, test_size=0.2, ran-
dom_state=42)

# Create and fit the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate Mean Squared Error
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")

# Print the R-squared value
r_squared = model.score(X_test, y_test)
print(f"R-squared: {r_squared}")

# Plot actual vs predicted prices
plt.figure(figsize=(8, 4))
sns.scatterplot(x=y_test, y=y_pred)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], linestyle='--', color='red',
linewidth=2)
plt.xlabel('Actual Price')
plt.ylabel('Predicted Price')
plt.title('Actual vs Predicted Prices with Reference Line')
plt.show()
```

DECISION TREE REGRESSION

Decision Tree is one of the most used, practical approaches for supervised learning. It can be usedto solve both Regression and Classification tasks with the latter being put more into practical application.

It is a tree-structured classifier with three types of nodes. The Root Node is the initial node which represents the entire sample and may get split further into further nodes. The Interior Nodes represent the features of a data set, and the branches represent the decision rules. Finally, the LeafNodes represent the outcome. This algorithm is very useful for solving decision-related problems.

With a particular data point, it is run completely through the entire tree by answering True/False questions till it reaches the leaf node. The final prediction is the average value of the dependent variable in that leaf node. Through multiple iterations, the Tree can predict a proper value for the data point.



6.3.3.Decision tree Regreesion

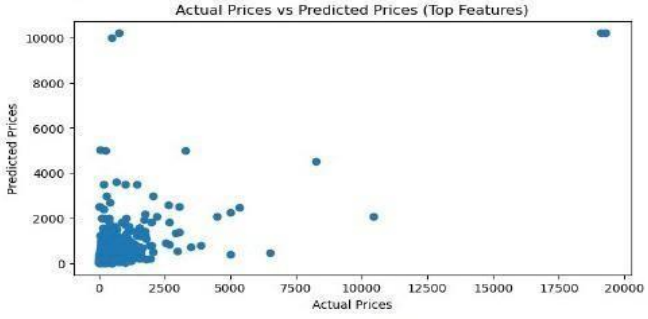
Implementation of Decision Tree Regression for Price Prediction Model

The Python code implements Decision Tree Regression to predict Airbnb prices based onvarious features. The Decision Tree Regressor is trained on the dataset, and the model captures non- linear relationships by recursively partitioning the feature space. This approach is well- suited forscenarios where the relationship between features and the target variable is intricate andinvolves complex interactions. Decision trees are advantageous for their interpretability, as the resulting tree structure provides insights into the decision-making process, making them valuablefor understanding feature importance and relationships within the data.

For **feature importance** the code extracts feature importance directly from the trained Decision Tree Regressor. The importance of each feature is shown and sorted in descending order. The topfive features are selected based on their importance, providing a valuable understanding of whichfeatures contribute most significantly to the prediction model. This information is crucial for feature selection and helps streamline the model to focus on the most influential variables.

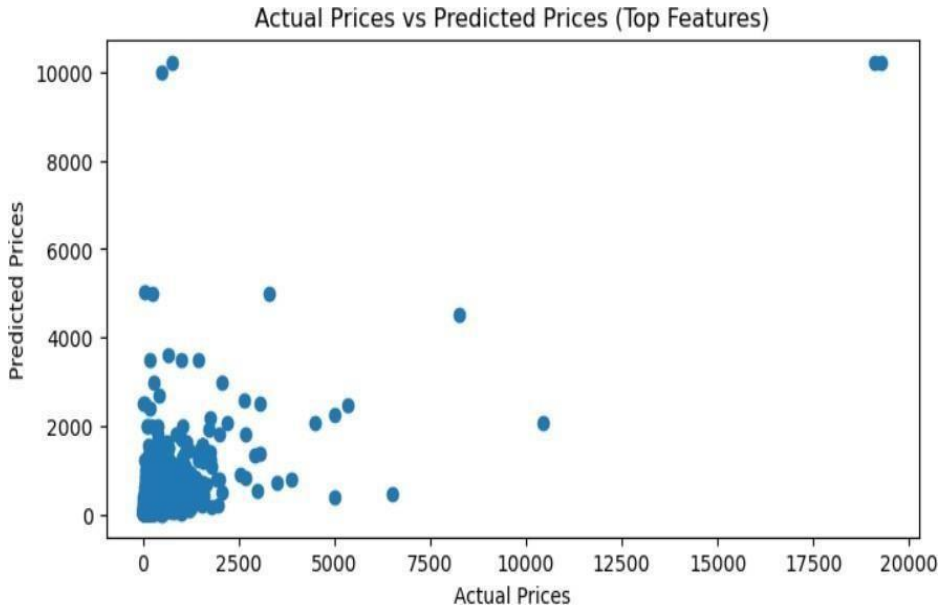
After training the Decision Tree Regressor with the top features, the Mean Squared Error (MSE) is calculated, which quantifies the average squared difference between predicted and actual valueson the test set. Additionally, the R-squared value is computed, and output value are shown below-

```
bedrooms: 0.1336467279405704
beds: 0.03379596341822028
baths: 0.04629899329715774
studio: 0.02833688447484972 5
shared_bath: 0.0009416461915709787
private_bath: 0.008983545697331502
host_id: 0.03309803436127578 6
neighbourhood: 0.07462784129306 59
latitude: 0.19680866018127496
longitude: 0.1154175718406153
minimum_nights: 0.06729104498614194
number_of_reviews: 0.035514004435046057
reviews_per_month: 0.07216732555090061
calculated_host_listings_count: 0.04700621280873449 5
availability_365: 0.0907681262471562 1
number_of_reviews_ltm: 0.01530041727608808 1
top_features: ['latitude', 'bedrooms', 'longitude', 'availability_365', 'neighbourhood']

Performance Metrics for Decision Tree Regression Model:
MSE: 281190.70836768777
R-squared: 0.3593879156287394
```



6.3.4.Accuracy of Decision tree

Scatter plot is generated for comparing the actual prices in the test set to the predicted prices using the top features. The scatter plot visually represents how well the model predictions align with the actual prices. Ideally, the points on the plot should cluster closely to a diagonal line, indicating accurate predictions.



6.3.5.Scatterplot of Decision tree

To summarize, the code effectively implements Decision Tree Regression, explores featureimportance, evaluates model performance using MSE and R-squared, and visualizes predictions through a scatter plot, offering a comprehensive analysis of the predictive capabilities of the model.

Advantages of Decision Tree Regression

1.Compared to other algorithms, decision trees require less effort for data preparation duringpre-processing.

2.A decision tree does not require normalization of data.

3.A decision tree does not require scaling of data as well.

4.Missing values in the data also do not affect the process of building a decision tree to any considerable extent.A Decision tree model is very intuitive and easy to explain to technical teams as well asstakeholders.

CODE:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

airbnb = pd.read_csv('airbnb_dataset.csv')
airbnb = airbnb.dropna(subset=['price'])

features = ['bedrooms', 'beds', 'baths', 'studio', 'shared_bath', 'private_bath', 'host_id',
        'neighbourhood', 'latitude', 'longitude', 'minimum_nights', 'number_of_reviews',
        'reviews_per_month', 'calculated_host_listings_count', 'availability_365', 'number_of_re-
views_ltm']

target = 'price'

X_train, X_test, y_train, y_test = train_test_split(airbnb[features], airbnb[target], test_size=0.2,
random_state=42)

# Fit the decision tree regressor
regressor = DecisionTreeRegressor(random_state=42)
regressor.fit(X_train, y_train)

# Get feature importance directly from the model
feature_importance = regressor.feature_importances_
```

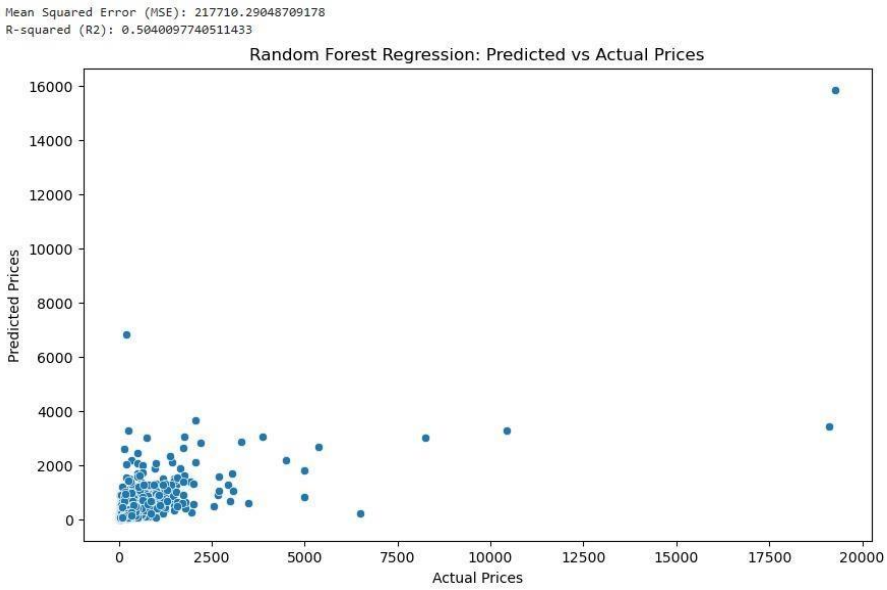
```
# Create a list to store feature names and their importance values
feature_importance_list = []

for feature, importance in zip(features, feature_importance):
    feature_importance_list.append((feature, importance))
    print(f"{feature}: {importance}")

# Sort feature importance in descending order
feature_importance_list = sorted(feature_importance_list, key=lambda x: x[1], reverse=True)

# Select the top 5 features
top_features = [feature for feature, _ in feature_importance_list[:5]]

print ("top_features:", top_features )

# Use only the top features for predictions
X_train_top = X_train[top_features]
X_test_top = X_test[top_features]

# Fit a new decision tree regressor using only the top features
regressor_top = DecisionTreeRegressor(random_state=42)
regressor_top.fit(X_train_top, y_train)

# Predictions on the test set using the top features
y_pred_top = regressor_top.predict(X_test_top)

# Calculate Mean Squared Error (MSE) and R-squared for the top features
mse_top = mean_squared_error(y_test, y_pred_top)
r2_top = r2_score(y_test, y_pred_top)

# Display performance metrics for the top features
print("\nPerformance Metrics for Decision Tree Regression Model:")
print(f"MSE: {mse_top}")
print(f"R-squared: {r2_top}")

# Plot the predicted vs actual prices for the top features
plt.figure(figsize=(8, 4))
plt.scatter(y_test, y_pred_top)
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.title('Actual Prices vs Predicted Prices (Top Features)')
plt.show()
```

RANDOM FOREST REGRESSION

The Random Forest Regressor is an ensemble learning method that enhances predictive accuracy and stability by constructing numerous decision trees during training. Each tree is trained on a random subset of the dataset, making decisions based on features. The aggregation of predictions from these individual trees results in the final output, typically the mean prediction for regression tasks. This ensemble approach not only leverages the strength of individual trees but alsominimizes overfitting, providing a robust and effective tool for regression analysis.

Feature selection:

The data set consists of various features related to accommodation listings. Amon all the features we have implemented the inbuilt function feature importances function which is inbuilt in the sklearn. ensemble in the random forest itself. There by using it we calculated the importance scoresfor the features and are mentioned below. We later took top most 9 features for building the most and gave the best predictions out of it. This feature selection process aids in identifying and prioritizing the key variables that influence pricing decisions in the context of Airbnb listings.



6.4.1.Accuracy of Random forest

Algorithm:

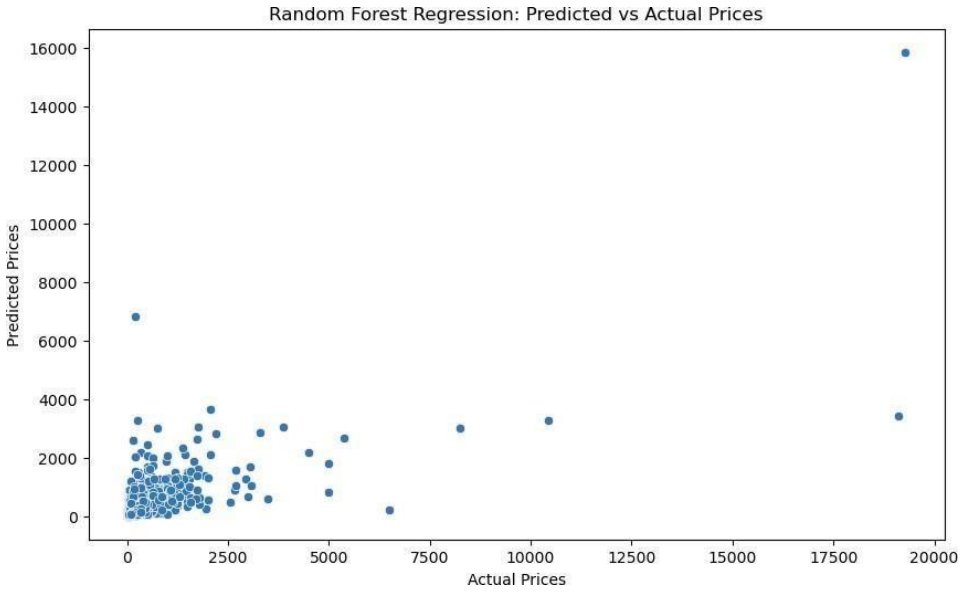
A subset of relevant features is selected based on the top features from the feature selected. Here after trying with different of features, we found top nine features gives the best score. The relevant features for predicting the 'price' target variable are selected and encoded, with categorical variables like 'room type' and 'neighborhood' transformed using Label Encoder as they are categorical columns. The model is then fitted to the selected features and target variable. The dataset is split into training and testing sets, with 80% used for training the model and 20% for evaluation. Subsequently, a Random Forest Regressor model is instantiated with 100estimators forrobust predictions and then trained on the training set. Afterward, predictions are made on the testset, and the model's performance is evaluated using metrics such as Mean Squared Error (MSE) and R-squared (R2). Finally, the predicted prices are visually compared to the actual prices througha scatter plot, providing insights into the model's accuracy in capturing the price variations in the Airbnb listings.

In conclusion the Importance score is a quantitative measure indicating the contribution of each feature to the model's ability to make accurate predictions. Higher importance scores suggest that the feature is more influential in determining the target variable.

Model Implementation:

The Mean Squared Error (MSE) of 217710.29 indicates the average squared difference between the predicted and actual prices. A lower MSE is preferable usually.
The R-squared (R2) score of 0.50 suggests that approximately 50.40% of the variability in the target variable ('price') can be explained by the model. R-squared values range from 0 to 1, where1 indicates a perfect fit, so 0.50 indicates a good level of predictive power when compared to all other machine learning models which we implemented here.
These metrics provide insights into the performance of the Random Forest Regression model.

Visualization:



6.4.2.prediction vs actual price

The scatter plot visually represents the model's predictions against the actual prices. Each point on the plot corresponds to a data point in the test set. The x-axis represents the actual prices, while the y-axis represents the predicted prices by the Random Forest Regression model. The points are scattered around the diagonal line, indicating the disparity between the predicted and actual values.A more accurate model would have points closely aligned along the diagonal. Here if we observe most points are aligned in the direction of diagonal, which states a good model.

Advantages of Random Forest regression:

High Predictive Accuracy: Random Forest tends to provide high predictive accuracy byaggregating multiple decision trees, reducing the risk of overfitting

Handles Non-Linearity: It can effectively model complex non-linear relationships in the data, making it suitable for a wide range of regression tasks.

Feature Importance: The algorithm provides a feature importance score, helping identify the most influential features in making predictions.

Robust to Outliers: Random Forest is robust to outliers and noise in the data due to its ensemble nature,which averages out individual errors.

CODE:

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler, LabelEncoder
import pandas as pd

# Assuming 'airbnb' is the DataFrame containing the dataset
features = airbnb[['bedrooms', 'beds', 'baths', 'studio', 'minimum_nights', 'availability_365', 'reviews_per_month', 'private_bath', 'ratings', 'calculated_host_listings_count','room_type','neighbourhood','id','longitude','number_of_reviews_ltm','host_id','shared_bath']]
target = airbnb['price']

# Encode categorical variables using .loc method to avoid SettingWithCopyWarning
le = LabelEncoder()
features.loc[:, 'room_type'] = le.fit_transform(features['room_type'])
features.loc[:, 'neighbourhood'] = le.fit_transform(features['neighbourhood'])

# Assuming 'model' is a RandomForestRegressor
model = RandomForestRegressor(n_estimators=100, random_state=42)
```

```
# Fit the model to your data
model.fit(features, target)

# Access the feature importances
feature_importances = model.feature_importances_

# Map feature names to importance values
feature_names = features.columns
feature_importance_dict = dict(zip(feature_names, feature_importances))

# Sort features by importance
sorted_features = sorted(feature_importance_dict.items(), key=lambda x: x[1], reverse=True)

# Display feature importances
for feature, importance in sorted_features:
print(f"{feature}: {importance}")


from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
airbnb['neighbourhood'] = le.fit_transform(airbnb['neighbourhood'])
airbnb['room_type'] = le.fit_transform(airbnb['room_type'])
features_selected = airbnb[['longitude','id', 'bedrooms','calculated_host_listings_count','availabil-
ity_365','host_id','neighbourhood','minimum_nights','reviews_per_month']]
```

```
target = airbnb['price']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(features_selected, target, test_size=0.2, random_state=42)

#Random Forest Regressor model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# predictions on the test set
predict = model.predict(X_test)

#Estimate the model
mse = mean_squared_error(y_test, predict)
r2_square_score = r2_score(y_test, predict)

#Calculations
print(f'Mean Squared Error (MSE): {mse}')
print(f'R-squared (R2): {r2_square_score}')

# Plotting predicted vs actual prices
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_test, y=predict)
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.title('Random Forest Regression: Predicted vs Actual Prices')
plt.show()
```

6.4.RECOMMENDATIONS

CLUSTERING

K-means clustering is a widely utilized unsupervised machine learning algorithm designed to partition datasets into distinct, non-overlapping subgroups or clusters. The primary objective of this algorithm is to aggregate similar data points, categorizing them into clusters based on specific features or characteristics.

K-means clustering on Airbnb listing data to identify distinct clusters based on various features. The feature importance is then calculated by an importance score which is already present as in-built function kmeans cluster centers in the library sklearn cluster import kmeans there by uitilizing that we calculated the importance scores. Based on the scores we just took the top 7 features as it gives the best solution. We tried with different count as well. The top 7 featuresare taken for clustering. Our analysis encompassed varying the number of clusters (2, 3, 10, 15,18) tocomprehensively understand the datadistribution. To assess the efficiency of different cluster sizes, we employed two key metrics: the Silhouette Score, which measures how similaran object is to its own cluster compared to other clusters, and the Davies-Bouldin Index, whichevaluatesthe compactness and separation of clusters. The evaluation indicated that a cluster sizeof 15yieldedthe most optimal distribution, despite some overlapping. This cluster configurationdemonstrated the most effective distribution among the alternatives considered.

The feature importance is then calculated for each cluster, revealing the key attributes that contribute to the differentiation between clusters. The top features are sorted based on their importance scores. Additionally, a recommendation function is implemented using cosine similarity, suggesting listings like a given input listing.

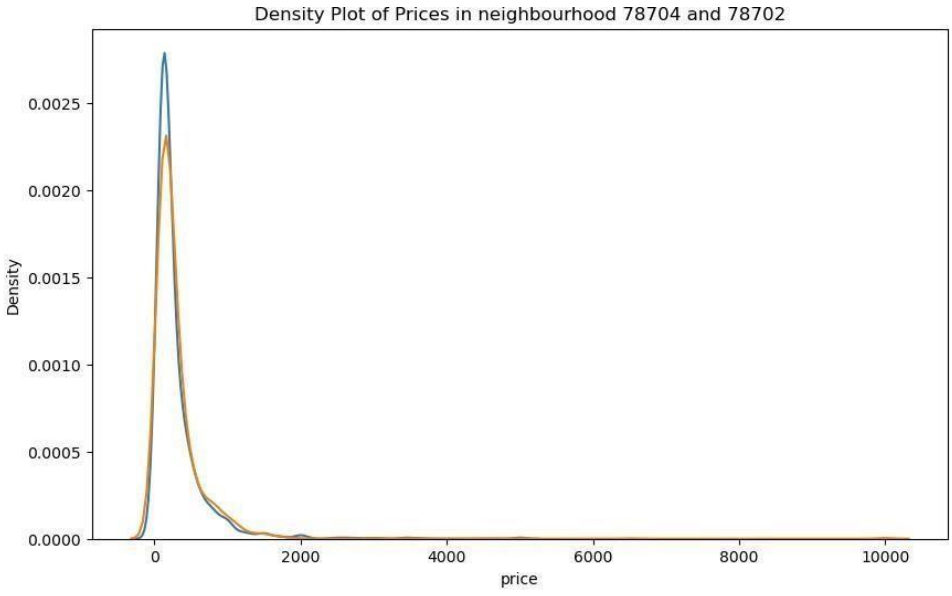
In practical terms, the clustering results can be used to categorize Airbnb listings into groups with shared characteristics. For example, clusters might represent listings with similar pricing patterns, review frequencies, and other relevant features. Let say one propertyid is given then it recommends the top5 property id with similar characteristics. The recommendation function allows for personalized suggestions by identifying listings with similarities to a chosen property. This information can be leveraged for targeted marketing, pricing strategies, or providing tailored recommendations to users based on their preferences. The clustering results can be used to categorize Airbnb listings into groups with shared characteristic

7.HYPOTHESIS TESTING

Suppose an owner is considering purchasing a property with the intention of listing it on Airbnb. After some initial analysis it becomes evident that the top two neighborhoods are the most promising options. However, a crucial question emerges: Does investing in a property in either ofthese neighborhoods yield equivalent returns? To address this inquiry, a hypothesis testing is undertaken

The null hypothesis shows that there is no significant difference in the mean price betweenthe top two neighborhoods. The findings reveal that null hypothesis is true.

With this information in hand, the prospective property buyer can now broaden their focusbeyond property price per night alone. Factors such as the overall locality, unit pricing andother relevant considerations can be incorporated into the decision-making process.




```
Test Statistic: -0.8239680449613584
P-value: 0.4100159287062213
Fail to reject the null hypothesis: No significant difference in mean prices.
```

7.1.Hypothesis Testing

8.RESULTS AND SOLUTION EVALUATION

	Linear Regression	Decision Tree Regression	Random Forest
MSE	8426.7487	281190.70	217710.29
R-Squared	0.3312	0.35	0.50

By comparing the Mean squared error (MSE) and R squared for all the machine learningalgorithms implemented, we can say that **Random Forest Regression** provides the lowest MSE value and highest R squared value. Highlighting its effectiveness in explaining and capturing the variability in the target variable.

Among the models evaluated—Linear Regression, Lasso Regression, Gradient Boosting Regressor, Decision Tree Regression, and Random Forest—Random Forest stands out as the mosteffective. It achieved the MSE of 271,710.29 and the highest R-Squared of 0.50, indicating superior predictive performance and a better ability to explain variance in the target variable. Theseresults suggest that Random Forest is the preferred model for predicting accommodation prices inthis dataset, offering a balance of accuracy and explanatory power.

The remarkable efficacy of the Random Forest model in predicting accommodation prices is highly advantageous for various stakeholders in the real estate and hospitality industry. This predictive capability is crucial for property owners, hosts, and investors to make informed decisions regarding pricing strategies, optimizing rental income, and maximizing occupancy rates. Additionally, the insights gained from feature importance analysis contribute to a deeper understanding of the key factors influencing pricing, enabling data-driven decision-making, and enhancing overall business strategies in the dynamic real estate landscape.

Here in all the algorithms, we performed models by taking multiple count of features, checked the performance with different count of features, tried different functions and methods to calculate the feature importance scores while coding in the report we have mentioned the best of all those.

9.CONCLUSION

1.To conclude we can say that non-linear regression models perform better than the linearregression algorithms by comparing the MSE and R squared values for the given dataset.

2.This analysis provides valuable suggestions into the factors influencing property prices aiding property owners and customers in strategic decision-making. The application of machine learning techniques demonstrated robust predictive capabilities for propertyprices.

3.The Insights obtained from this project contribute to the growing body of knowledge in thefield of machine learning for real-world applications and provide a foundation for further refinement and optimization of pricing prediction models in the dynamic context of real estate.

In conclusion, the price prediction and recommendation model for Airbnb property listings using machine learning can greatly enhance the decision-making process for both hosts and guests. By leveraging historical data and various features such as location, amenities, and property characteristics, the model can accurately predict listing prices and offer personalized recommendations. This empowers hosts to optimize their pricing strategy and maximize revenue, while helping guests find accommodations that best fit their preferences and budget. However, it's crucial to continuously update the model with new data and refine its algorithms to ensure its accuracy and relevance over time. Overall, the implementation of such a model can significantly improve the efficiency and effectiveness of the Airbnb marketplace for all stakeholders involved.

Based on the analysis conducted, it's clear that predicting prices for Airbnb property listings involves various factors such as location, amenities, seasonality, and market demand. To enhance accuracy, machine learning algorithms like regression or neural networks can be employed. However, it's essential to continually update models due to changing market dynamics. Recommendations for Airbnb property listing include optimizing pricing based on demand fluctuations, improving property amenities to attract guests, and leveraging data-driven insights to make informed decisions. Additionally, fostering positive guest experiences through excellent service can lead to higher occupancy rates and better reviews, ultimately increasing profitability.

FUTURE SCOPE : The future scope for price prediction and recommendations of Airbnb property listings using machine learning is promising and expansive. Here are some potential avenues for further development:

1.Enhanced Feature Engineering: Continuously refining and expanding the set of features used in the model can improve its predictive accuracy. This could include incorporating more granular location data, sentiment analysis of guest reviews, and real-time external factors such as local events or economic trends.

2.Dynamic Pricing Strategies: Implementing dynamic pricing algorithms that adjust listing prices in real-time based on demand fluctuations, seasonal trends, and competitor pricing can optimize revenue for hosts and improve booking rates.

User Feedback Loop: Implementing a feedback loop where user interactions and booking outcomes are fed back into the model can continuously improve its performance and adapt to changing market dynamics and user preferences.

6.Expansion to New Markets: Scaling the model to cover additional geographic markets and property types beyond traditional Airbnb listings, such as vacation rentals, long-term rentals, or boutique hotels, can broaden its applicability and impact.

Overall, the future scope for price prediction and recommendations of Airbnb property listings using machine learning is vast, with ample opportunities for innovation and improvement to create a more efficient, transparent, and user-centric marketplace experience.