|  |  |
| --- | --- |
|  |  |
| Madrid Airbnb Data |  |
|  |  |
|  |  |
|  | RAJAT CHAUHAN |

1. **Introduction**

Airbnb has transformed the way people travel by providing an alternative to traditional hotel accommodations. The platform offers a diverse range of properties, including apartments, houses, and villas, that can be rented out by hosts to travelers. Madrid, being a popular tourist destination, attracts a large number of visitors each year. The availability of affordable and comfortable accommodations on Airbnb has made it an attractive option for travelers visiting Madrid.

* **Problem Statement**:

The problem that this project aims to address is understanding the key factors that influence the rental prices of properties on the Madrid Airbnb platform. By analyzing these factors, we can gain insights into the Madrid Airbnb market, provide valuable information to both hosts and guests, and contribute to the broader understanding of the sharing economy and its impact on local housing markets.

* **Objective**

The objective of this project is to conduct a analysis on Madrid Airbnb data to identify the factors that influence the rental prices of properties. By analyzing the data, we aim to understand the key drivers that affect rental prices and develop a predictive model that can estimate the rental prices of properties in Madrid. This information can be useful for both hosts and guests, as it can help hosts optimize their pricing strategies and assist guests in making informed decisions about their accommodation choices. Furthermore, the study can provide insights into the broader housing market in Madrid and shed light on the potential impact of Airbnb on the local economy.

**II. Data Exploration**

**Data source and description**

The data for this project was collected from the Inside Airbnb website, which is an independent, non-commercial set of tools and data that provides information on the Airbnb platform in various cities around the world. The Madrid Airbnb data includes information on various aspects of the Airbnb listings in Madrid, such as the property type, neighborhood, room type, number of bedrooms, bathrooms, and amenities. The data also includes the rental price of each listing, along with other relevant information such as the minimum and maximum nights required for a stay, the availability of the property, and the number of reviews and overall rating of the listing.

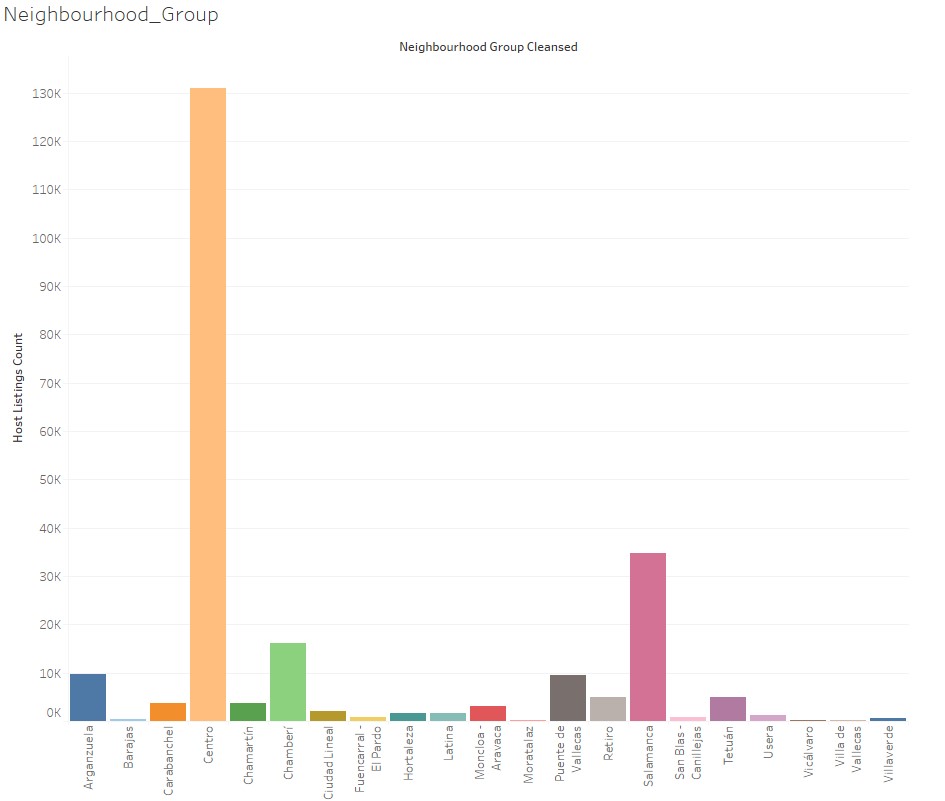
**Data cleaning and preprocessing**

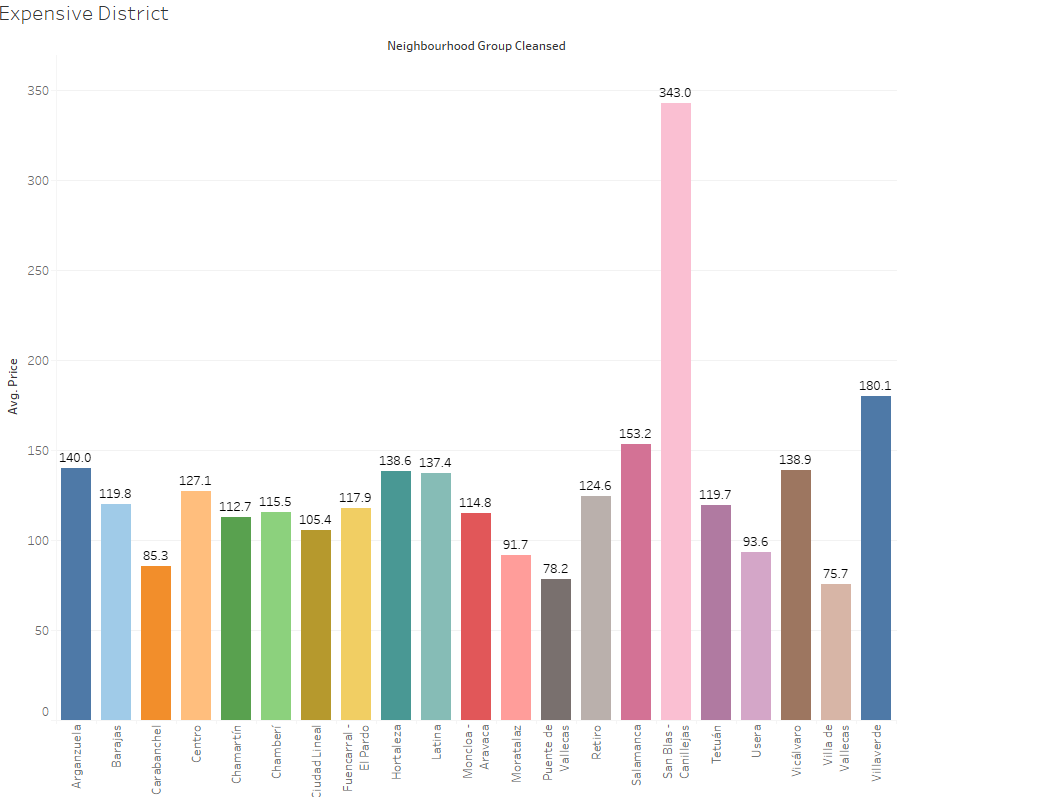
Data cleaning and preprocessing are crucial stages in data analysis that involve getting data ready for analysis by eliminating redundant or irrelevant data, handling missing data, converting categorical data into numerical form, and dealing with outliers. these procedures, it is ensured that the data used for analysis is precise, comprehensive, and prepared for modeling.

* **Irrelevant data**: Incomplete or irrelevant data can interfere with analysis, resulting in inaccurate or biased findings. Making sure that only pertinent data is used for analysis by removing duplicates and useless data can increase the precision of models and insights drawn from the data.
* **Managing missing values**: Missing data values can interfere with analysis, producing inaccurate or incomplete findings. Identifying the missing values and choosing how to handle them, such as removing the rows or columns with missing data, requires handling missing data. Imputation methods can be used to fill in the missing values.
* **Transforming categorical variables**: Many machine learning algorithms need numerical data, but frequently, data may also include categorical variables, such as text or groups. This process converts categorical variables into numerical form. Dummy variables or methods like label encoding or one-hot encoding can be used to convert these variables into numerical representation.
* **Handle outliers**: Outliers are values that vary noticeably from the rest of the data's values and can skew results or impair the accuracy of models. Identifying outliers and choosing how to deal with them—either by getting rid of them, changing them, or using robust statistical techniques that are less impacted by outliers—require dealing with outliers.

**Data visualization and exploratory data analysis (EDA)**

1 Number of hosts listed in Centro is more in another neighborhood

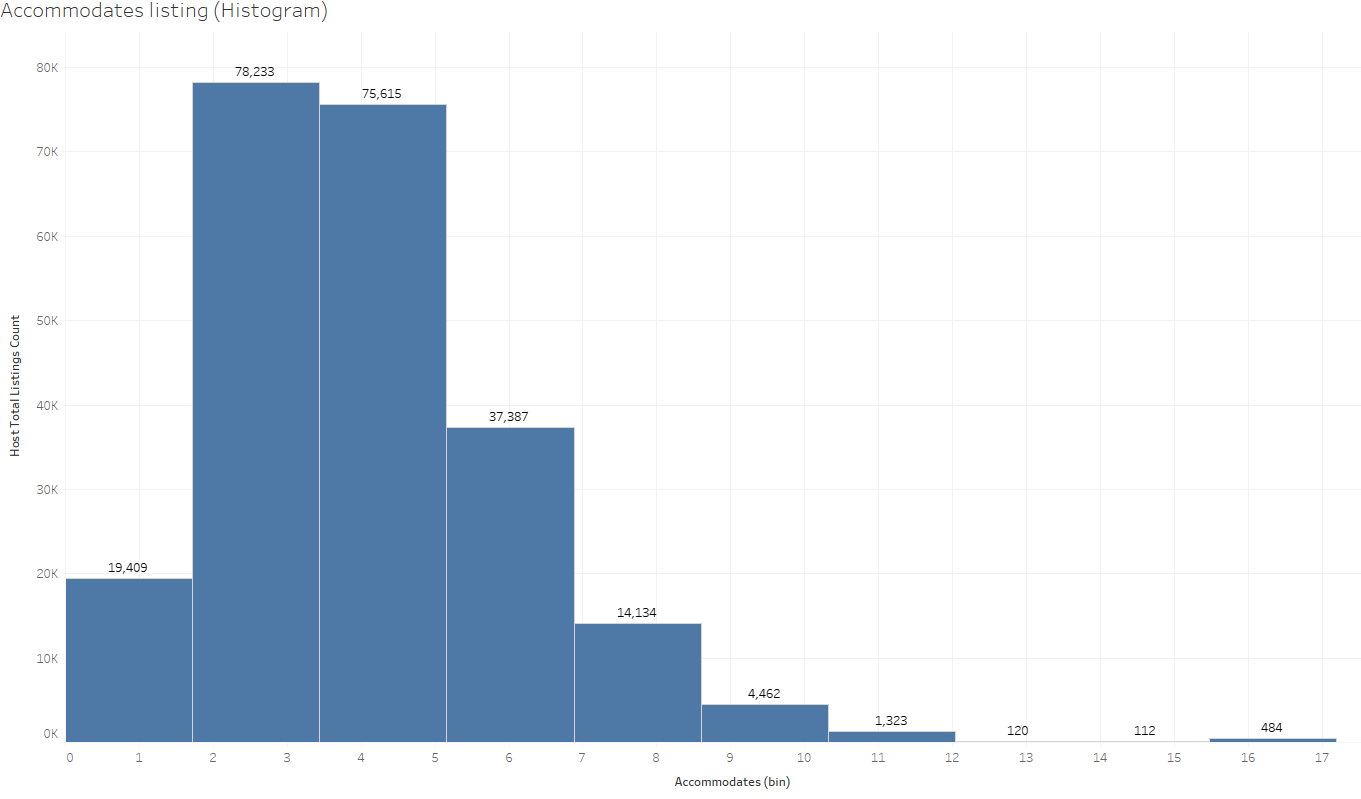




San Blas is Expensive District it is clear from graph

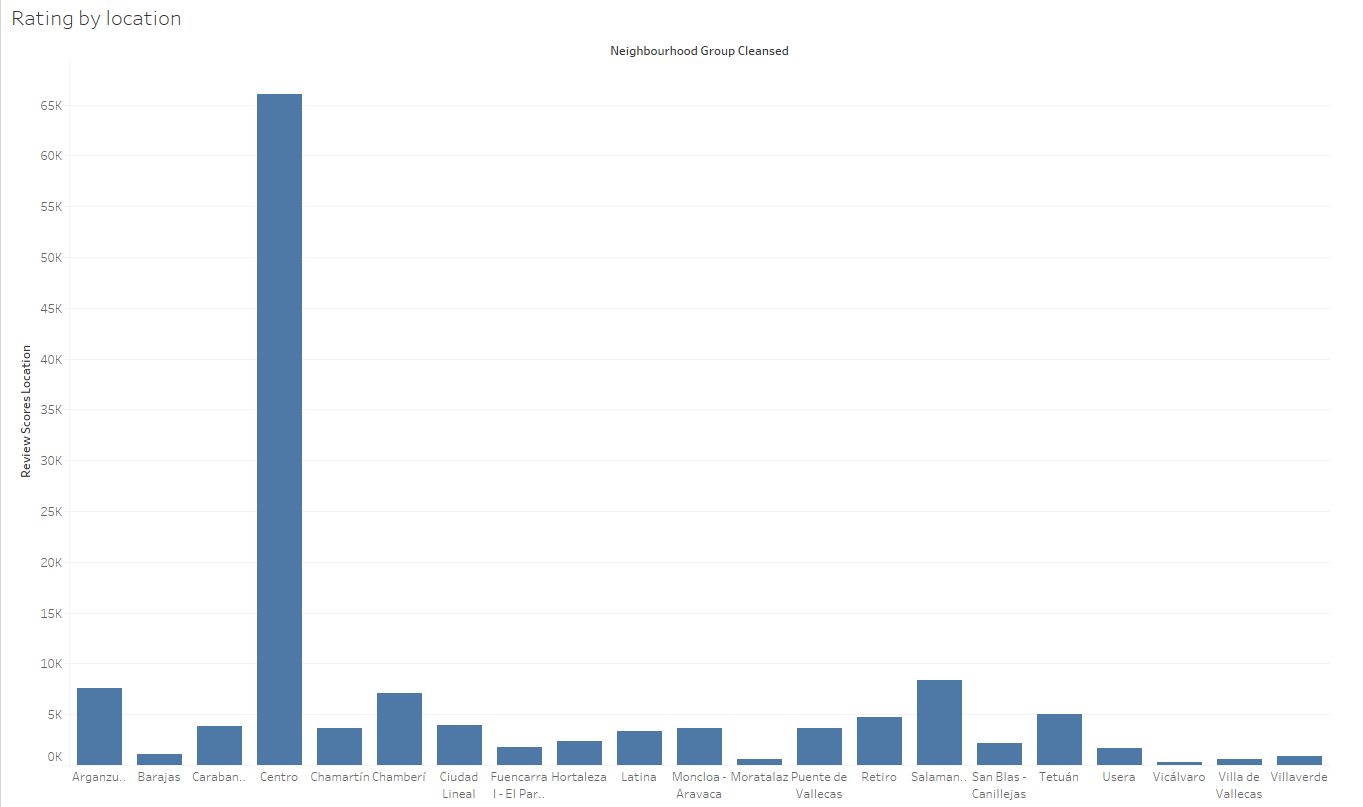


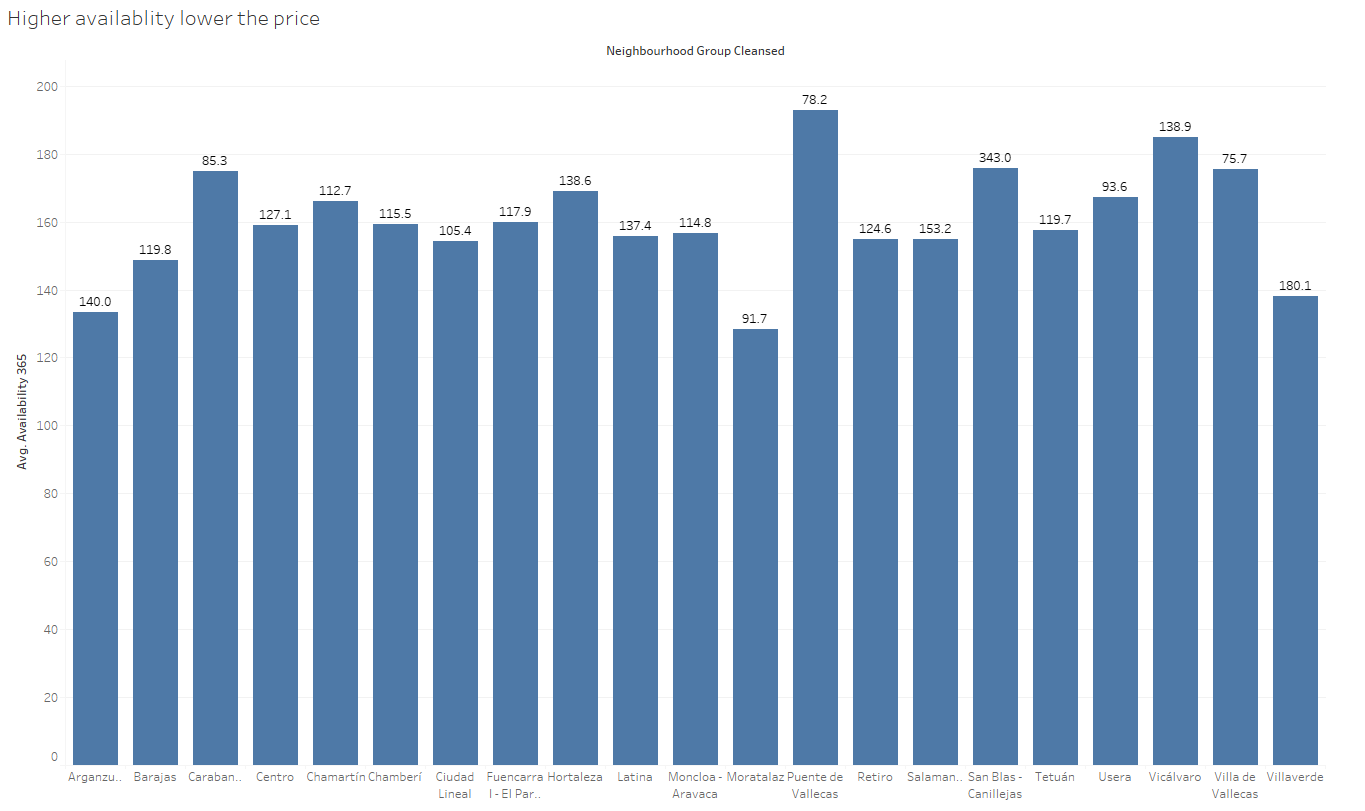
"Price rating Density Plot showing prices and their rating" describes a type of data visualization that displays the distribution of prices and their associated ratings.



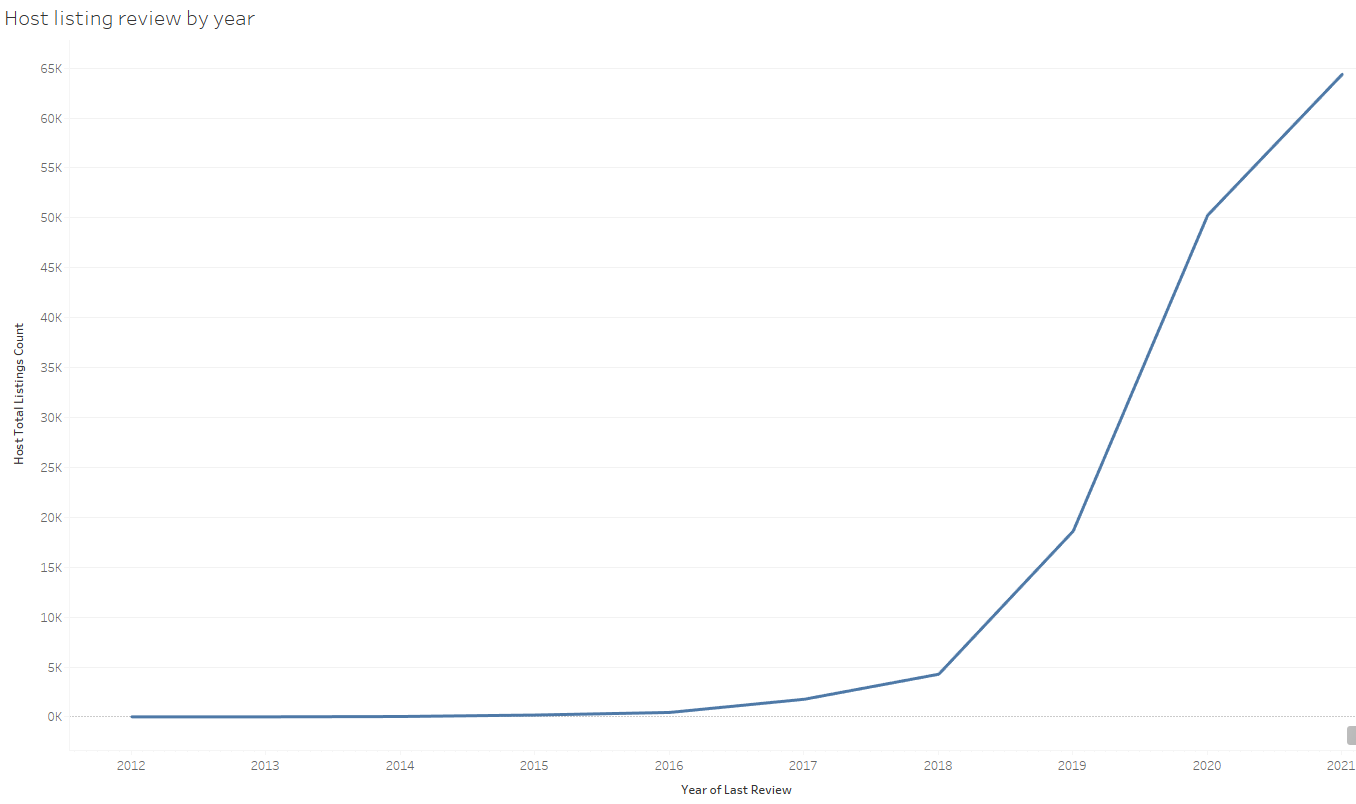
"accommodates" refers to the maximum number of guests that a particular listing can accommodate

Rating by location



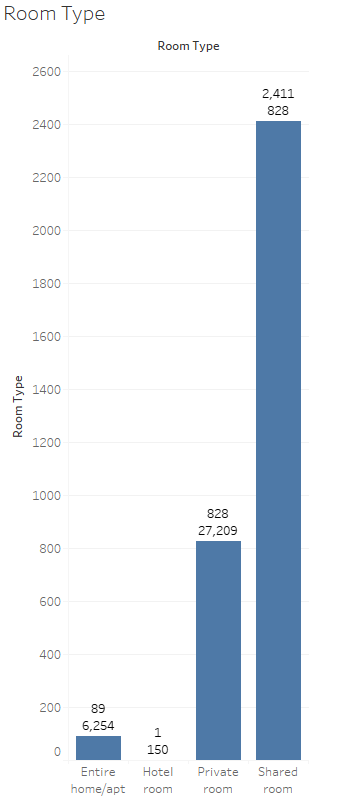


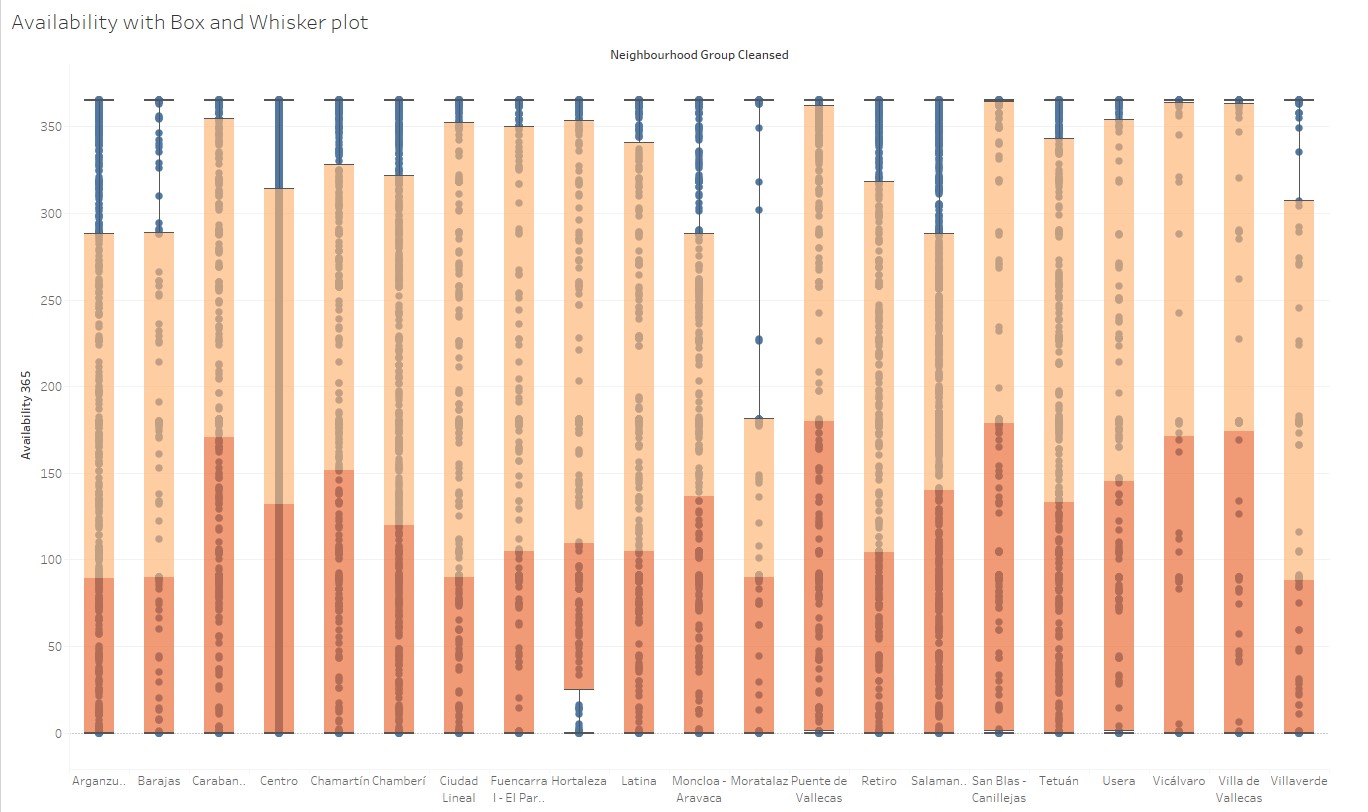
Here we can see the higher the availability lowers the price Puente de is highest availability



Host listing rating over the year increasing year by year

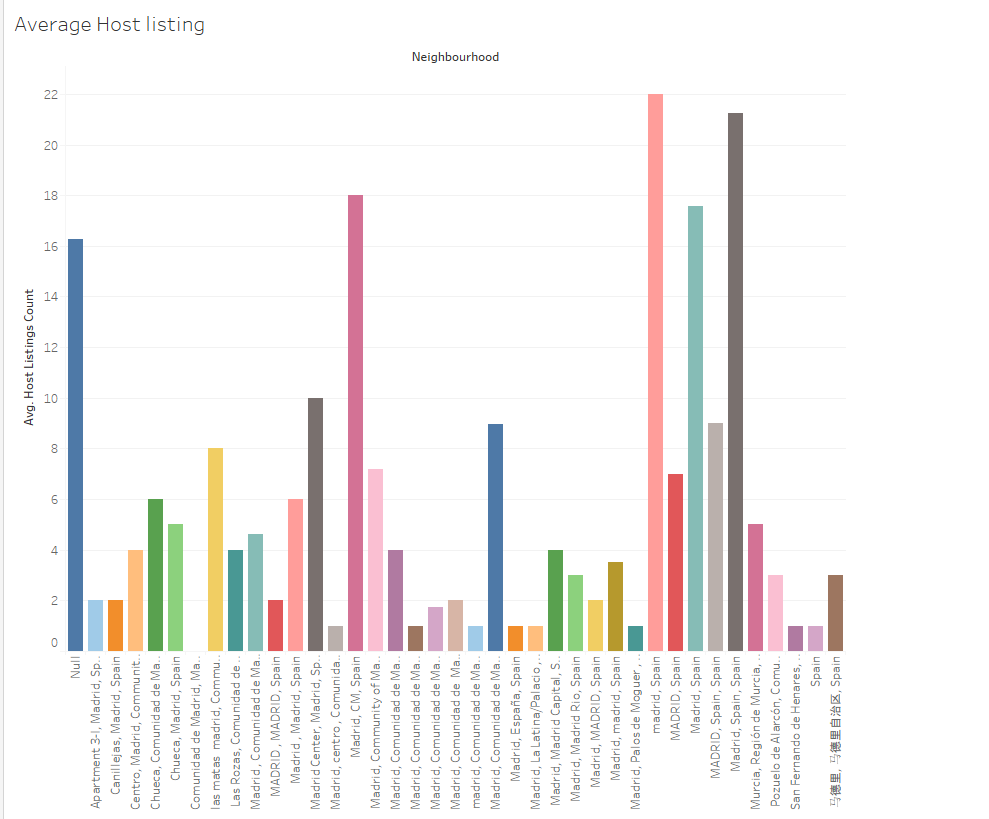
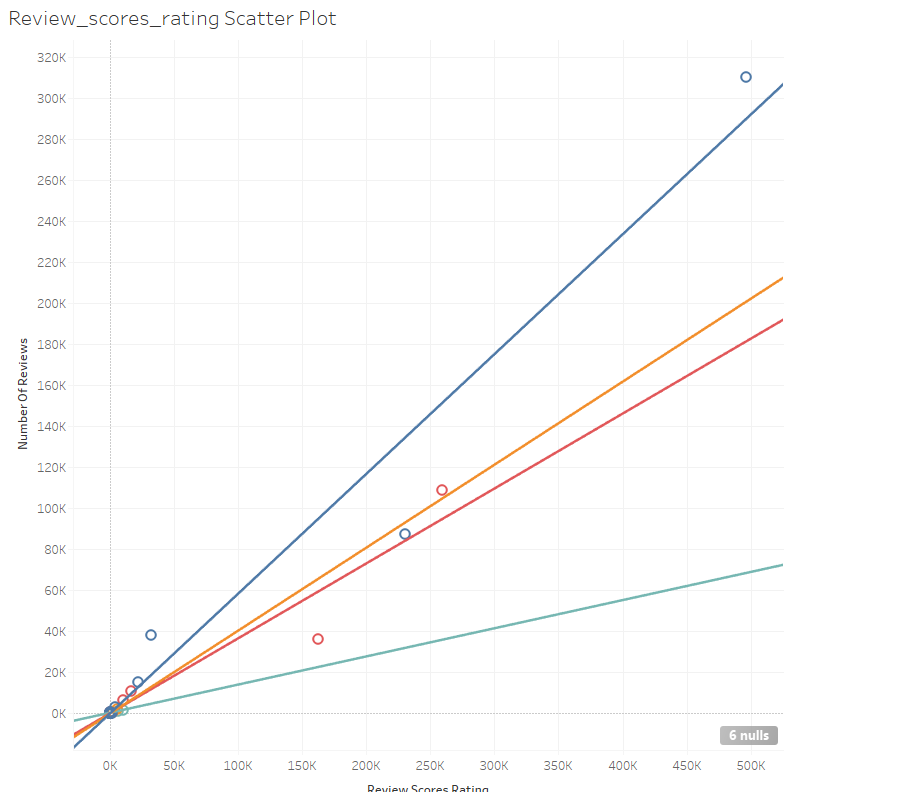
4 Room type with

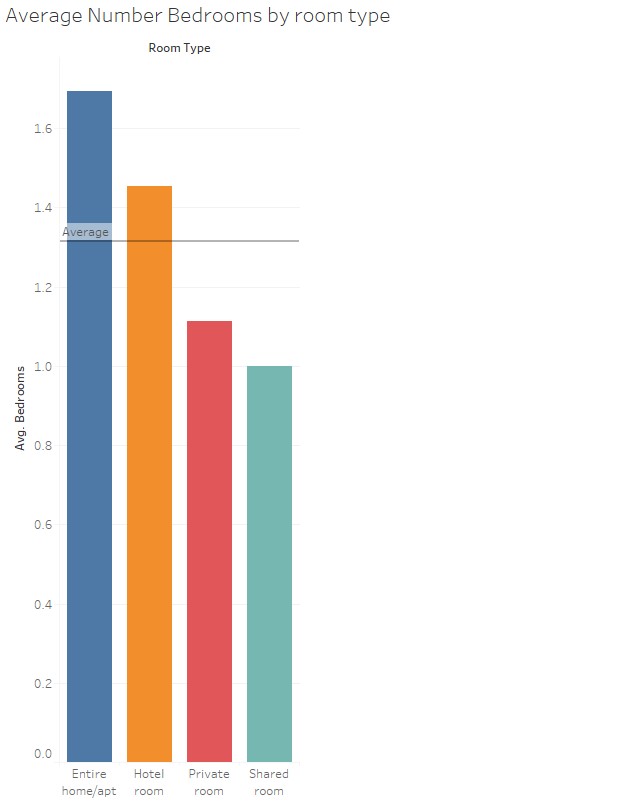
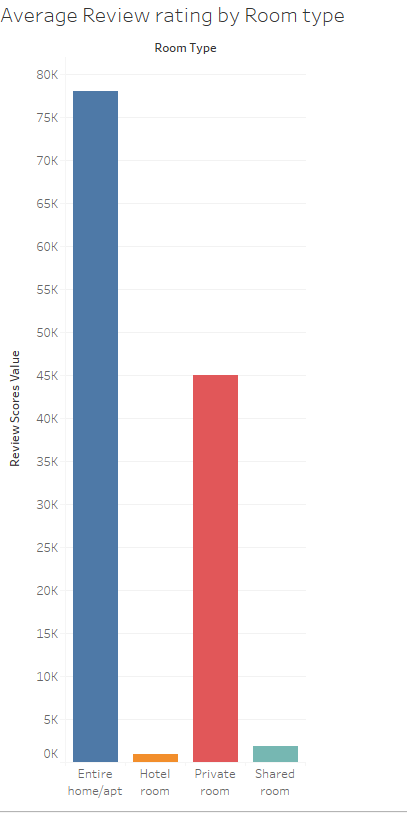


Availability with Box and Whisker plot

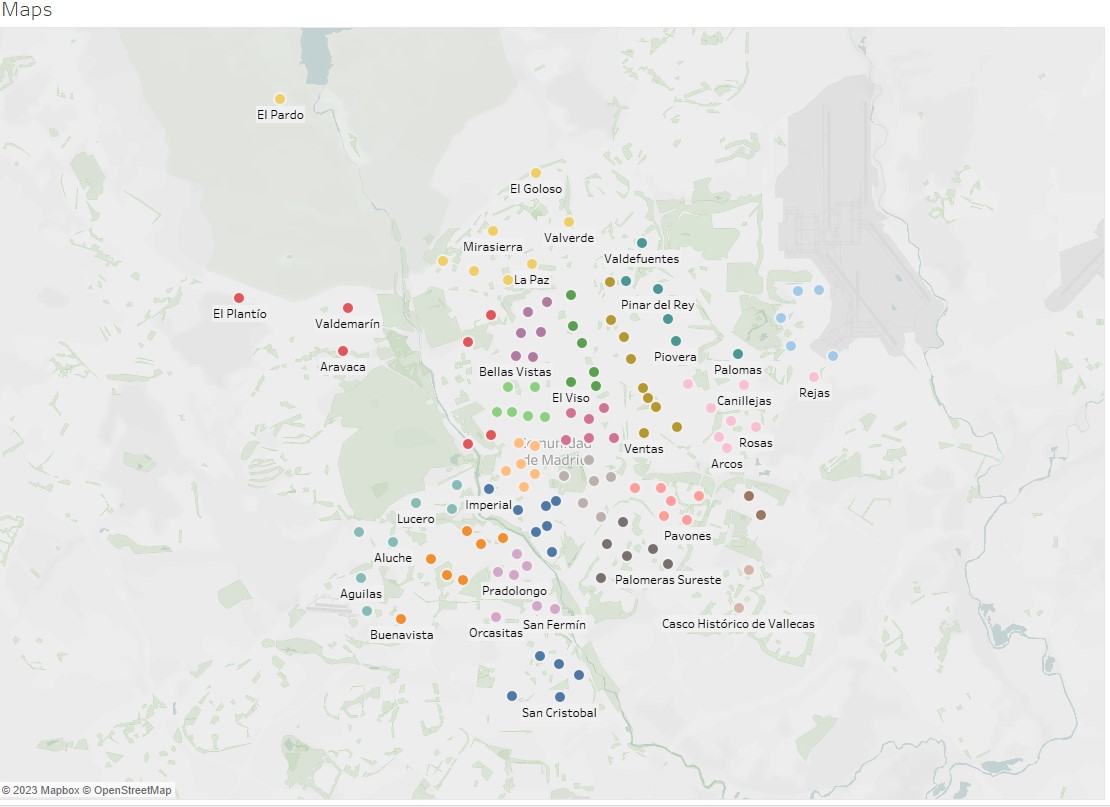
Review\_scores\_rating Scatter Plot

Average Host listing

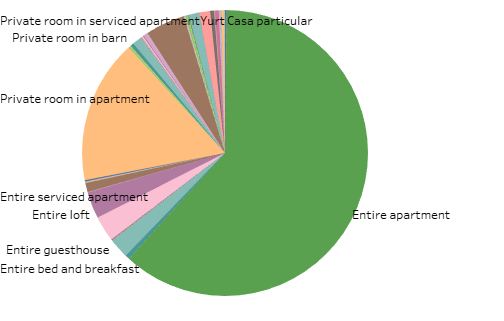


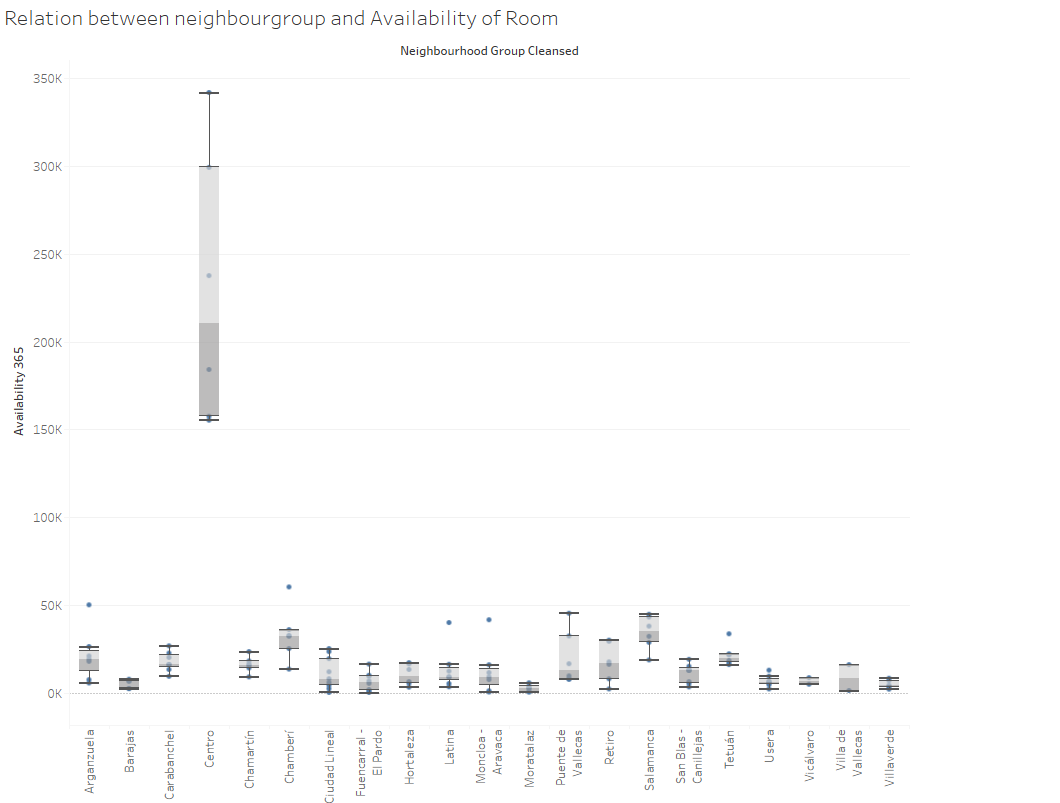
Average Number Bedrooms by room type Average Review rating by Room type 

Maps of Neighbourgood\_group



Pie Chart by property type





**Key findings from data exploration**

Key findings from EDA Some of the key findings from EDA include:

* The majority of listings in Madrid are entire apartments or private rooms, with very few shared rooms or hotel rooms. This suggests that visitors to Madrid prefer to have their own private space rather than sharing with others.
* The most expensive listings tend to be located in the city center, with lower prices in the suburbs. This finding suggests that location is an important factor in determining the price of a listing.
* Hosts with multiple listings are more likely to be running a business, and are unlikely to be living in the property. This finding suggests that some hosts may be using platforms like Airbnb as a way to generate income by renting out multiple properties.
* The higher the availability of a property, the lower its price. This finding suggests that properties that are in high demand are likely to be priced higher, while those that are less in demand may be priced lower.
* Private rooms have more ratings than other types of listings. This finding suggests that visitors to Madrid prefer to stay in private rooms, and that these rooms may be more popular or more commonly available than other types of listings.

**Data quality issues**

During the EDA process, several data quality issues were identified and addressed. These included missing values in some variables and outliers in some numerical variables. These issues were dealt with using appropriate techniques such as outlier removal to ensure that the analysis was not affected by these issues.

**III. Methodology**

**Model selection and justification**

Once the data was prepared, multiple regression models were selected for this analysis. The models chosen were Linear Regression, Decision Tree Regression, Random Forest Regression, GBT Regression, and Generalized Linear Regression. These models were chosen because they are commonly used for predicting continuous variables and have been shown to perform well

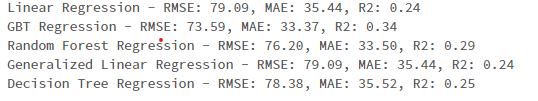
The goal of the analysis is to compare the performance of different regression models on the dataset. The models that are used are Linear Regression, Random Forest Regression, GBT Regression, Generalized Linear Regression, and Decision Tree Regression.

The first step is to split the dataset into training and testing sets using the randomSplit method. The data is split into 80% for training and 20% for testing, and a seed of 42 is used for reproducibility.

Then, the regression models are initialized, and each model is trained using the training data. The models are trained using the fit method, and the featureassembler.transform method is used to transform the training data into a format suitable for the models.

Once the models are trained, predictions are made on the test data using each model, and the performance of each model is evaluated using the RMSE, MAE, and R2 metrics. The RegressionEvaluator method is used to compute these metrics.

In summary, this performs regression analysis on a dataset, compares the performance of different regression models, and evaluates the models using RMSE, MAE, and R2 metrics.



The results of the evaluation show that the GBT Regression model performs the best in terms of RMSE, MAE, and R2 metrics. Its RMSE score is 73.59, MAE score is 33.37, and R2 score is 0.34. The Random Forest Regression and Decision Tree Regression models also perform relatively well, with RMSE scores of 76.20 and 78.38 respectively. The Linear Regression and Generalized Linear Regression models perform the worst, with RMSE and MAE scores of 79.09 and R2 scores of 0.24.

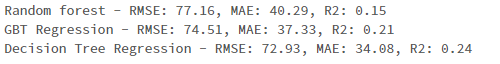
**Model selection and evaluation metrics**

Compares the performance of different regression models, including Random Forest Regression, Gradient Boosted Tree Regression, and Decision Tree Regression.

First, the code defines the input features for the analysis. Then, it creates a pipeline to normalize or standardize the data and apply PCA to reduce the dimensionality of the data. The pipeline includes a feature assembler to assemble the input features into a single vector, and a PCA transformer to transform the assembled features into a lower-dimensional space.

Next, the code splits the data into training and testing sets and initializes the regression models. It trains each model on the training data and makes predictions on the testing data using each model.

Finally, the code evaluates the performance of each model using RMSE, MAE, and R2 metrics, and defines a metrics function that computes and prints the RMSE, MAE, and R2 for a given model and predictions.



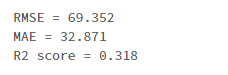
Based on the results, it seems that the **Decision Tree Regression** model performs the best, as it has the lowest RMSE and MAE and the highest R2 score among the three models.

**Hyperparameter tuning and optimization**

Now we used to train and evaluate a Decision Tree Regression model using cross-validation with different hyperparameters to find the best model for the given data.

First, a pipeline is created with a feature assembler and the Decision Tree Regression model. Then, a parameter grid is defined for grid search to find the optimal hyperparameters for the Decision Tree Regression model.

Three evaluators are defined for RMSE, MAE, and R2 metrics. Finally, the cross-validator is instantiated with the pipeline, parameter grid, and RMSE evaluator.

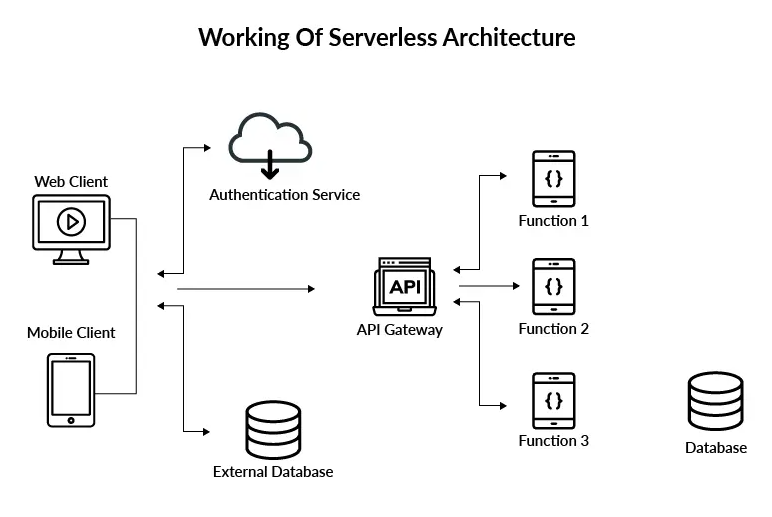


**The output summarizes the performance of the model**

* The RMSE (Root Mean Squared Error) score of the model is 69.352. This means that the average difference between the predicted and actual prices is around $69.
* The MAE (Mean Absolute Error) score of the model is 32.871. This means that the average absolute difference between the predicted and actual prices is around $33.
* The R2 score of the model is 0.318. This means that around 31.8% of the variability in the dependent variable (price) can be explained by the independent variables used in the model.

IV**. Architecture**

Three different architectures for deploying a house price prediction model using Airbnb data: serverless, containerized, and traditional virtual machine (VM) based architectures. Each has its own benefits and drawbacks



1. **Serverless Architecture** (e.g., using AWS Lambda, Amazon SageMaker, and API Gateway):

Pros:

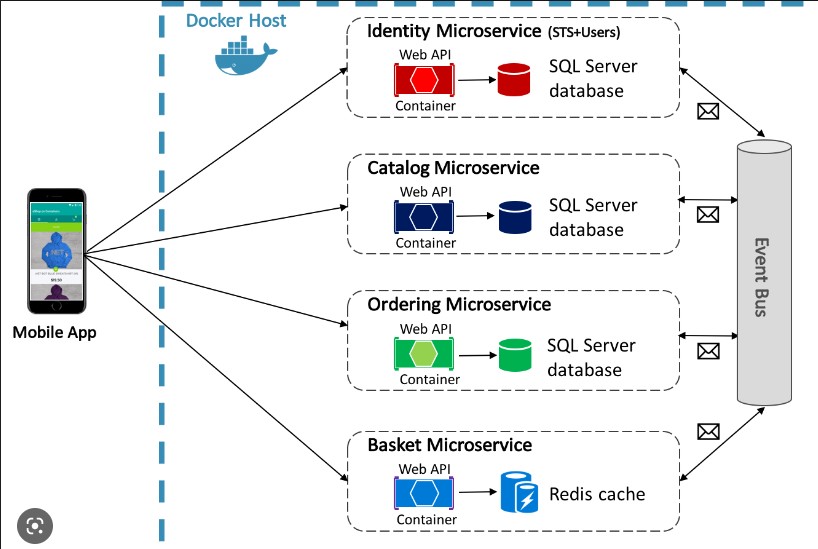
* Automatically scales based on demand, making it cost-effective.
* No need to manage the underlying infrastructure, reducing maintenance overhead.
* Pay-as-you-go pricing model, only pay for the compute resources you actually use.

Cons:

* Limited customization of the underlying infrastructure.
* May not be the best fit for applications with consistently high resource requirements.
* Vendor lock-in, as serverless offerings can be provider-specific.

Why it could be a good fit: A serverless architecture is ideal for this use case because it offers excellent scalability, flexibility, and cost-effectiveness. It allows you to focus on developing the property price prediction model without worrying about the underlying infrastructure. Moreover, it's perfect for applications with variable workloads and usage patterns, as the infrastructure automatically scales based on demand.

**Containerized Architecture (e.g., using Kubernetes or Docker Swarm):**

****

Pros:

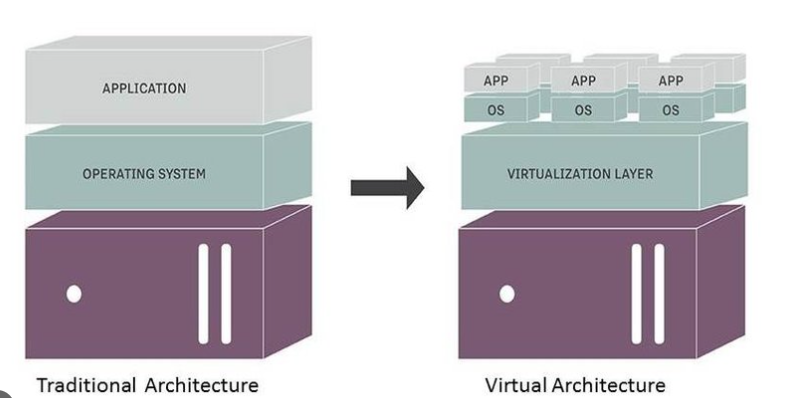
* Provides a consistent and reproducible environment for developing, testing, and deploying applications.
* Eases the deployment and scaling of applications across different platforms and environments.
* Can be deployed on-premises or in the cloud, providing flexibility in choosing the infrastructure.
* Supports rolling updates and zero-downtime deployments.

Cons:

* Can be complex to set up and manage, especially for large-scale deployments.
* Might have higher operational costs compared to serverless architectures.
* Requires more maintenance and management of the underlying infrastructure.

Why it could be a good fit: A containerized architecture is suitable for organizations that want more control over their infrastructure or require a consistent environment across development, testing, and production stages. It's also a good fit for teams already familiar with containerization technologies like Docker and Kubernetes.

**Traditional Virtual Machine (VM) based Architecture:**



Pros:

* Complete control over the underlying infrastructure, including hardware and operating system configurations.
* Can be deployed on-premises or in the cloud.
* Easier to set up compared to containerization for small-scale deployments.

Cons:

* Higher operational and maintenance costs compared to serverless and containerized architectures.
* Requires manual scaling and resource management.
* Less efficient in terms of resource utilization, as each VM runs a separate operating system.

Why it could be a good fit: A traditional VM-based architecture can be a suitable choice for small-scale deployments or organizations with existing VM infrastructure. It's also a good fit for teams who are comfortable with VM management and do not require the additional benefits provided by containerization or serverless architectures.

**Comparison:**

* In summary, serverless architecture is the best fit for deploying a house price prediction model using Airbnb data due to its scalability, cost-effectiveness, and reduced operational overhead. Containerized and traditional VM-based architectures can be considered based on specific organizational requirements and existing infrastructure.
* Containerized architecture can be a good alternative for organizations that want more control over their environment or already have experience with container technologies. It offers a consistent and reproducible environment across all stages of the application lifecycle.
* Traditional VM-based architecture can be suitable for small-scale deployments or organizations with existing VM infrastructure. However, it comes with higher operational and maintenance costs compared to serverless and containerized architectures.

**Discussion of limitations and potential areas for improvement**

Incorporating additional features such as crime rates, proximity to public transport, and nearby amenities such as restaurants and parks could improve the model's accuracy. The present model only considers a single moment in time's worth of data. To integrate the trends and patterns of the data over time, a time-series analysis could be carried out.

**Future work and potential extensions of the project**

The current model used a limited set of features for the prediction of house prices. Incorporating additional features such as crime rates, proximity to public transport, and nearby amenities such as restaurants and parks could improve the model's accuracy.

Deep learning models: In regression problems, deep learning models like neural networks have shown encouraging results. The accuracy of the predictions could be increased using a deep learning algorithm.

**V. Conclusion**

The exploratory data analysis performed on the Madrid Airbnb listings data revealed key findings related to the type of listings, their location, and availability. Multiple regression models including Linear Regression, Decision Tree Regression, Random Forest Regression, GBT Regression, and Generalized Linear Regression were selected and evaluated using RMSE, MAE, and R2 metrics to compare their performance on the dataset. The GBT Regression model was found to be the best-performing model based on the evaluation metrics.

methodology used to compare the performance of different regression models, including Random Forest Regression, Gradient Boosted Tree Regression, and Decision Tree Regression. The Decision Tree Regression model was found to be the best-performing model based on the evaluation metrics.

Finally, the text includes information on hyperparameter tuning and optimization, where a Decision Tree Regression model was trained and evaluated using cross-validation with different hyperparameters to find the best model for the given data. The output of the model evaluation includes RMSE, MAE, and R2 scores, which show the performance of the model in predicting the prices of Madrid Airbnb listings. The RMSE score of the model is 69.352, indicating that the average difference between the predicted and actual prices is around $69. The MAE score of the model is 32.871, meaning that the average absolute difference between the predicted and actual prices is around $33. The R2 score of the model is 0.318, indicating that the independent variables used in the model can explain around 31.8% of the variability in the dependent variable (price).

**VI. References and Literature Review**

Zhang, W., & Skitmore, M. (2018). Predicting housing prices with machine learning techniques. Journal of Property Investment & Finance

Huang, Y., Yang, L., Deng, Y., & Kang, J. (2020). A comprehensive review of machine learning for housing price prediction. International Journal of Automation and Computing

Kim, J. H., Cho, S. H., & Lee, H. (2018). House price prediction using machine learning: A review. International Journal of Advanced Science and Technology

Jiang, Y., Li, H., Zhang, X., & Wang, J. (2020). House price prediction using machine learning: A systematic literature review.

Kusiak, A. (2018). Predicting housing prices with machine learning using scoringcard features. Applied Soft Computing,