SIADS 699 | Capstone Project Report
Team 14 - Sam Kobrin, Alex Chan, Sachiko Uchikoshi

Developing an Open-Source List Price Prediction Tool for Airbnb Hosts

With the rise of platforms like Airbnb and VRBO, hosting has become a popular side hustle to generate extra income. However, setting the right price for your property can be a daunting task. Each listing is unique, with many characteristics such as location, amenities, the number of bedrooms and bathrooms, and more. Additionally, demand fluctuates based on the day of the week, peak seasons, weather, and holidays. Although tools like AirDNA exist to help hosts set prices, they often require sharing personal data, raising privacy concerns. Our mission was to develop an open-source tool that allows Airbnb hosts to predict listing prices based on their property's characteristics without compromising privacy.

Methodology

To achieve our objectives, we designed a robust technical architecture. Our technology stack consists of a snowflake database, a machine learning model training script and a Streamlit dashboard to enable users to perform model inference through an easy to use interface.

ETL and Data Warehousing

Data Sources

Data collected for this project was obtained from Inside Airbnb, which scrapes data from public listings and stores it in CSV and GeoJSON formats. We chose to focus on seven major markets and use listings, reviews, and neighborhood data that was collected by Inside Airbnb on June 7, 2024. The markets we chose include Albany, Chicago, Los Angeles, New York City, San Francisco, Seattle, and Washington D.C.. We chose these markets because they had the largest number of Airbnb listings or, in the cases of Washington, D.C. and Albany NY, project authors resided in them.

Data	Description	Format	Size (Data Warehouse)
Listings	Listings data	CSV	111.1K rows
Reviews	Customer reviews for listings	CSV	4.1M rows

Data Warehouse Setup

The Snowflake data warehouse is configured with two schemas to streamline data management:

- **Operational Data Store (ODS):** Centralized repository for processed data.

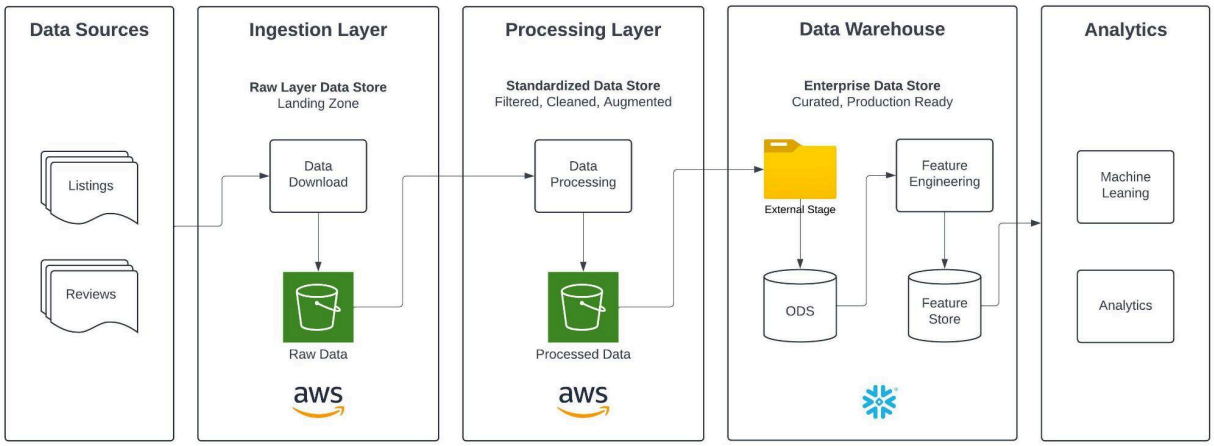
- **Feature Store:** Stores enriched data with features for machine learning and analytics.

Data Pipeline Architecture

Our data architecture combines AWS S3 and Snowflake to ensure a robust, scalable, and efficient data pipeline. Here's a detailed look at each stage of the process:

1. **Data Ingestion**: The raw CSV files are downloaded using *ingest_data.ipnyb* and the data is uploaded to an S3 bucket which serves as a data lake, providing scalable storage for the raw data.
2. **Data Processing**: Once the raw data is stored in S3, it undergoes processing using the *process_listings.ipynb* and *process_reviews.ipynb* notebooks. These notebooks are designed to handle the initial data cleaning and preprocessing tasks, such as handling missing values, normalizing data, and ensuring data quality.
3. **Data Storage and Management**: After data processing is completed, the data is loaded directly from S3 into Snowflake. The S3 bucket is connected to Snowflake as an external stage which allows us to manage and load data directly from S3 into Snowflake. This clean and processed data is stored in the Operational Data Store (ODS) schema.
4. **Feature Store**: Feature engineered data is loaded in the Feature Store schema. *sentiment_scores_pipeline.ipynb* is used to develop a pipeline that tokenizes text and generates sentiment scores, which is then ready for downstream analytical tasks.

The diagram below illustrates the architecture, depicting the flow from raw data ingestion to feature storage and its various components.



Data Preprocessing

Python scripts were crucial components of our data processing pipeline before loading the data into Snowflake. These scripts perform several key functions to ensure that the data is well-prepared and maintains its integrity:

1. **Loading:** Read compressed CSV files from the raw folder in AWS S3 into dataframe.
2. **Data Cleaning:**
    a. Add market column
    b. Rename columns
    c. Clean columns and convert data types, e.g., remove $ from price column and convert to numeric, remove % from host_response_rate and convert to float

    d. Check for duplicate rows
    e. Drop unnecessary columns
    f. Ensure date columns are formatted as date type
3. **Logging:** Incorporate logging to track the progress of data cleaning tasks for transparency and troubleshooting.
4. **Save to Parquet File:** Once the cleaning is completed, save the cleaned dataframe as a Parquet file. The benefits of a Parquet file is that they are columnar and support efficient compression. These file types allow for faster read times and are optimized for big data, and they maintain a consistent schema.
5. **Storage:** The Parquet file is loaded into a folder in S3 that stores processed data and then into the Operational Data Store schema in Snowflake.

Model Development

Literature Review and Documenting Domain Knowledge

We adhered to the traditional machine learning model building process, starting with stakeholder consultations and literature review. Our initial step involved discussions with an Airbnb host to pinpoint key features they believe influence listing prices. Her insights were complemented by an extensive literature review to identify features highlighted in previous analysis. For instance, Karkala identified the number of bedrooms, guest capacity, room type, and listing location as critical determinants of listing price (Karkala). Similarly, Mohamed Irfan's research on the Seattle market underscored the significance of listing type (entire home/private room/shared room) and amenities, noting that features like walkability, parking, and scenic views also positively correlate with higher list prices (Irfan).

Exploratory Data Analysis

Following the literature review, we conducted exploratory data analysis to refine our feature selection. We started by examining the correlation coefficients between price and 14 potential features, including host response rate, host acceptance rate, superhost status, neighborhood, host profile picture presence, identity verification, room type, guest capacity, number of bathrooms, number of beds, review scores, instant booking availability, and market. Our analysis revealed a strong correlation ($r > 0.2$) for room type, number of bedrooms, number of bathrooms, and guest capacity. Additionally, certain neighborhoods showed a strong price correlation, emphasizing the importance of location.

To ensure the robustness of our feature set, we employed lasso regression, a technique that helps in feature selection by penalizing the absolute size of coefficients. This method confirmed that room type, guest capacity, number of bathrooms, and number of bedrooms were the most influential features.

Sentiment Analysis of Reviews

To incorporate sentiment analysis into our dashboard, we collected reviews of Airbnb listings from inside Airbnb and stored them in a Snowflake database. The sentiment analysis component involved the following steps:

Text Preprocessing: Cleaned the review column for each city's review data by removing stopwords, punctuation, and performing tokenization.

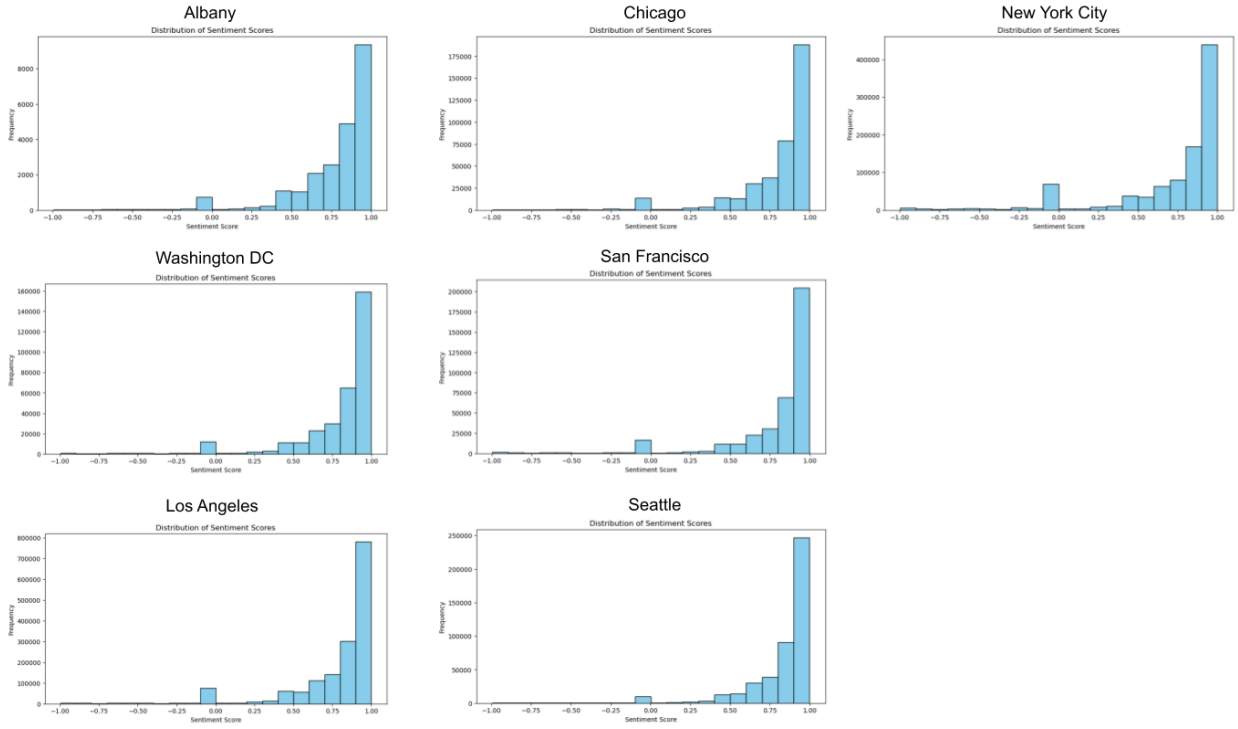
Sentiment Scoring: Using pre-trained sentiment analysis model VADER (Valence Aware Dictionary and sEntiment Reasoner), we assigned sentiment scores to each review.

Aggregating Sentiment Scores: Calculating average sentiment scores for each listing to use as a feature in the price prediction model.

This analysis was performed on the largest city markets in the United States including New York City, Chicago, Los Angeles, San Francisco, Seattle, and Washington DC, with the inclusion of Albany as a market of interest.

Sentiment Score Distribution

As seen in the collection of histograms below, every city had an overwhelmingly large percentage of positive reviews. The 3 largest bins for every city were sentiment scores between 0.75 and 1.00. This is an interesting observation because the number of reviews varied drastically between the cities. On the large end was Los Angeles with around 2,000,000 reviews and on the small end was Albany with 21,000 reviews. Despite the varying amounts of reviews, the distribution of sentiment scores was consistent.



Top 20 Most Common 3-grams

Trigram analysis was performed on reviews after stopword removal to identify common phrases. These phrases provide an insight into what airbnb customers liked or disliked about their stay. All the 3-grams are displayed in Appendix 1. For every city except for San Francisco, “great place stay” was the most common trigram. The most common trigram for San Francisco was “golden gate park”. The remaining 19 most common trigrams for each city contained positive words such as “enjoyed”, “recommend”, and “home”.

Some cities had a 3-gram in the top 20 most common that pertained to a specific attraction in that city. Examples include “empire state plaza” for Albany, “pike place market” for Seattle, “close subway

station" for New York City, and "golden gate park" for San Francisco. This tells us that Airbnb listings in these cities that are close to attractions are very popular and are in desired locations.

These 3-grams, along with other n-grams that could be analyzed within reviews, could be very useful for Airbnb hosts when it comes to choosing where to host and what amenities to include. Also, any negative reviews and negative n-grams can provide insight into what a host can improve for their listing(s).

Model Training

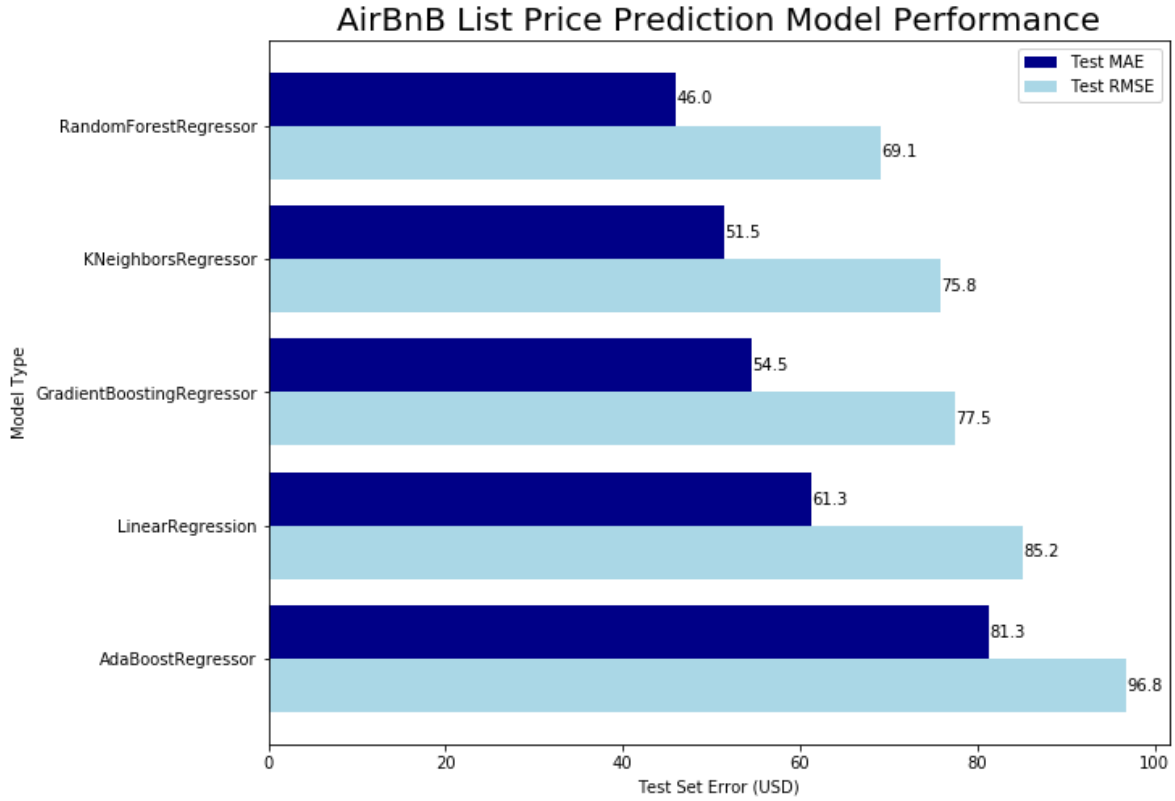
With these insights, we started our model training using the identified key features: number of bedrooms, number of bathrooms, guest capacity, and market, along with latitude and longitude coordinates to capture location information. By using latitude and longitude instead of one-hot-encoded neighborhood names, we aimed to avoid the curse of dimensionality—a scenario where the model becomes too complex and overfits due to an excessive number of features. This approach not only simplified our model but also retained the crucial aspect of location, ultimately enhancing the model's predictive power.

Model Evaluation

We defined success as predicting the price of an Airbnb across all 8 of our target markets with a mean absolute error (MAE) of less than $50. We chose MAE over root mean square error (RMSE) because MAE equally weights all errors, providing an easy to interpret average of error magnitudes. For AirBnB list price prediction it is important to treat all pricing errors equally since each dollar of revenue is equally valuable. For instance, an error of $10 should have the same impact on our evaluation as any other $10 error, regardless of the actual price range. RMSE, on the other hand, disproportionately penalizes larger errors, which might not be appropriate for our use case where consistency across all price points is crucial. Nevertheless, we also evaluated our trained models using RMSE as a robustness check to ensure the consistency of our results.

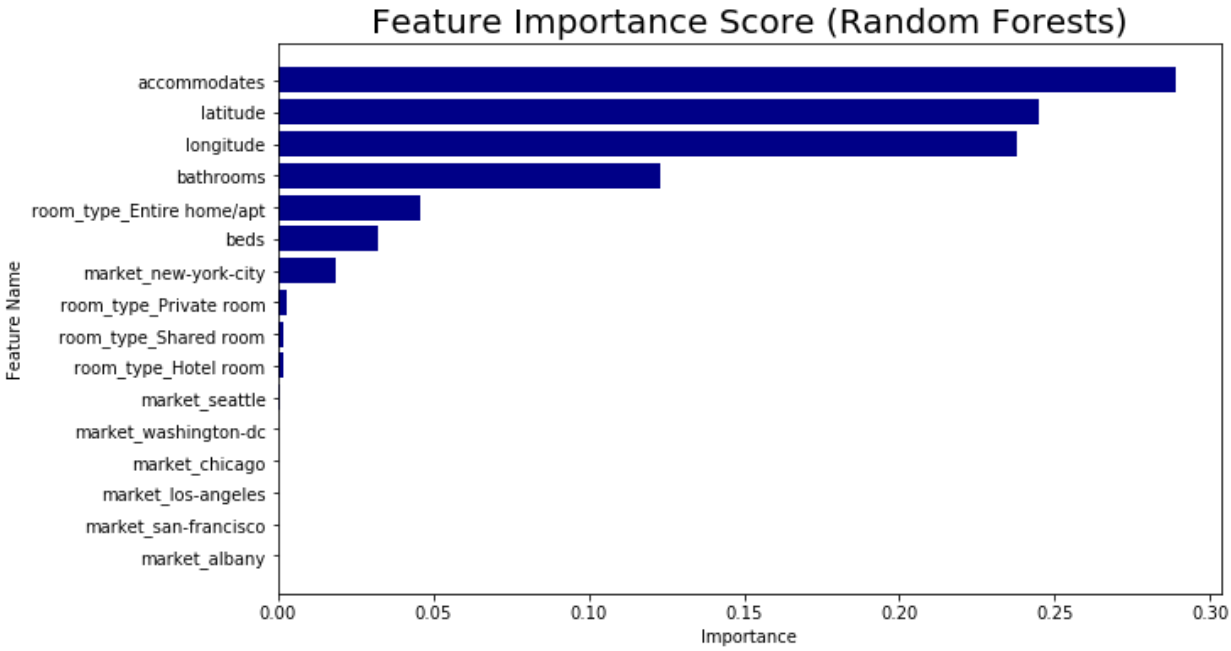
To maintain comprehensive and accessible records of our experiments, we implemented a custom experiment logging pipeline in JSON format, ensuring that our data was both human- and machine-readable. We started with linear regression as our baseline model before running experiments with more complex models such as K-Nearest Neighbors (KNN), Random Forest, XGBoost, and AdaBoost. Our experiments showed that the Random Forest model consistently performed best across all

our model runs and evaluation metrics.



Our best model in our initial round of model development, a Random Forest Regression, produced a mean absolute error of $46 USD on our the holdout (test) set with a root mean squared error of $69.1 USD. Because this meets our evaluation criteria or a mean absolute error of less than $50, we were fairly satisfied with this result. Furthermore, we chose to use a random forest model to productionalize as the training and inference processes for Random Forests are computationally efficient, allowing for quick model updates and real-time predictions in production environments. These characteristics make Random Forests a practical choice for deployment in our production list price prediction dashboard.

Our best model predominantly utilized the number of guests an Airbnb can accommodate to predict the listing price, with location being the next most significant factor. Interestingly, the number of bathrooms proved to be a more critical predictor than the number of bedrooms. This can be attributed to the multicollinearity between the number of beds and the accommodation capacity. Market information did not significantly influence the model's decisions, likely because location details are more accurately captured at a neighborhood level through latitude and longitude. Consequently, market features could be excluded from the model without significantly impacting its performance.



Improving The Model Using Hexagons

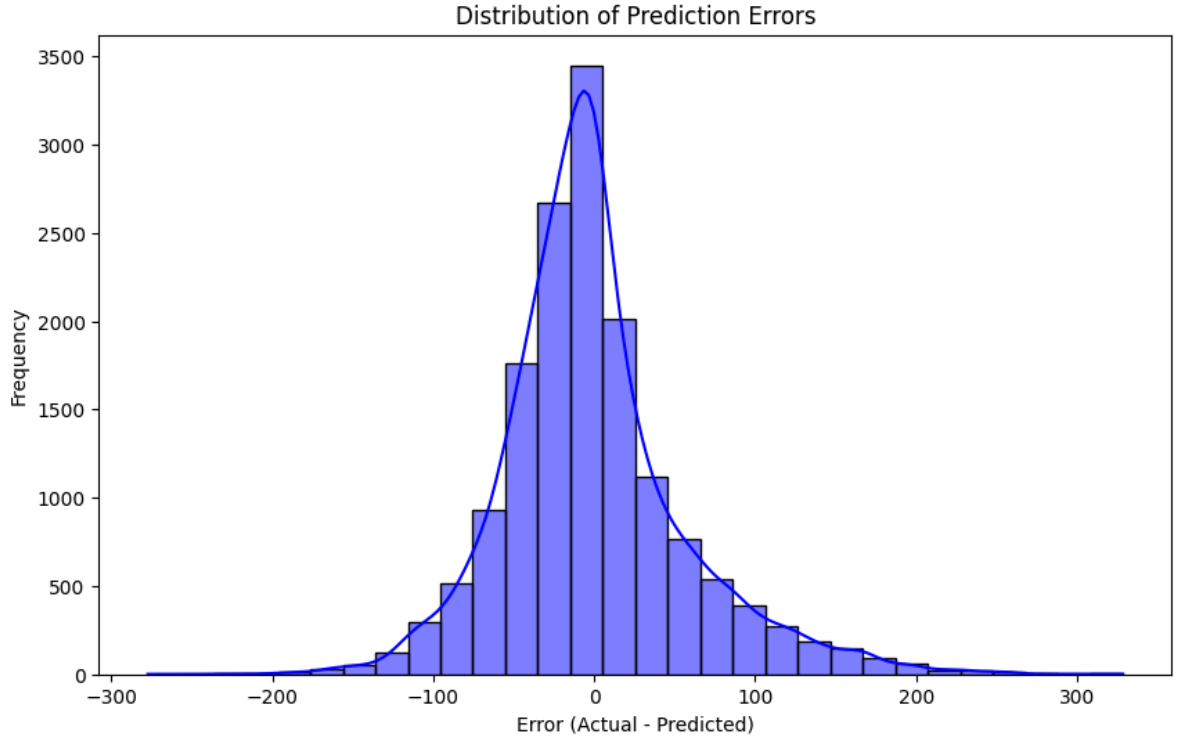
Building on top of the work in the initial training and evaluation of our model, we decided to explore the application of hexagons to each listing after drawing inspiration from the article "Exploring Location Data Using a Hexagon Grid" by Smith (2019). Uber's h3 library was used to assign each listing a hexagonal grid (h3_index), which allowed us to calculate aggregated features such as *accommodates_median, bathrooms_median, beds_median, and price_median*. Aggregating these features allowed the model to account for localized factors that influence price since listings in the same hexagon share the same aggregated features.

Instead of using latitude and longitude coordinates as features, each listing's coordinates were converted to a h3_index using a resolution of 7 which is equivalent to a diameter of 0.87 miles. This method provided a consistent way to represent different geographic areas. In order to prevent data leakage, we split the data into training and test sets, and then calculated the aggregated hexagon features based on the *training* set. Afterwards, the features were attached to the training and test set. This ensured the model only used information available during training, leading to more generalizable results.

As done with our first phase of model evaluation, we used a Random Forest Regressor once again. Model tuning was performed via Randomized Search since it's known to work fast for large datasets and selects a random subset of hyperparameters.

Using the updated model, the Mean Absolute Error (MAE) was $41, which is an improvement of $5 when compared to the best model in our first phase of model development having an MAE of $46. The RMSE for this updated model is $58. Visualizing the distribution of errors shows that on average the model

overestimated the predicted values.



AirBnB Price Prediction Dashboard

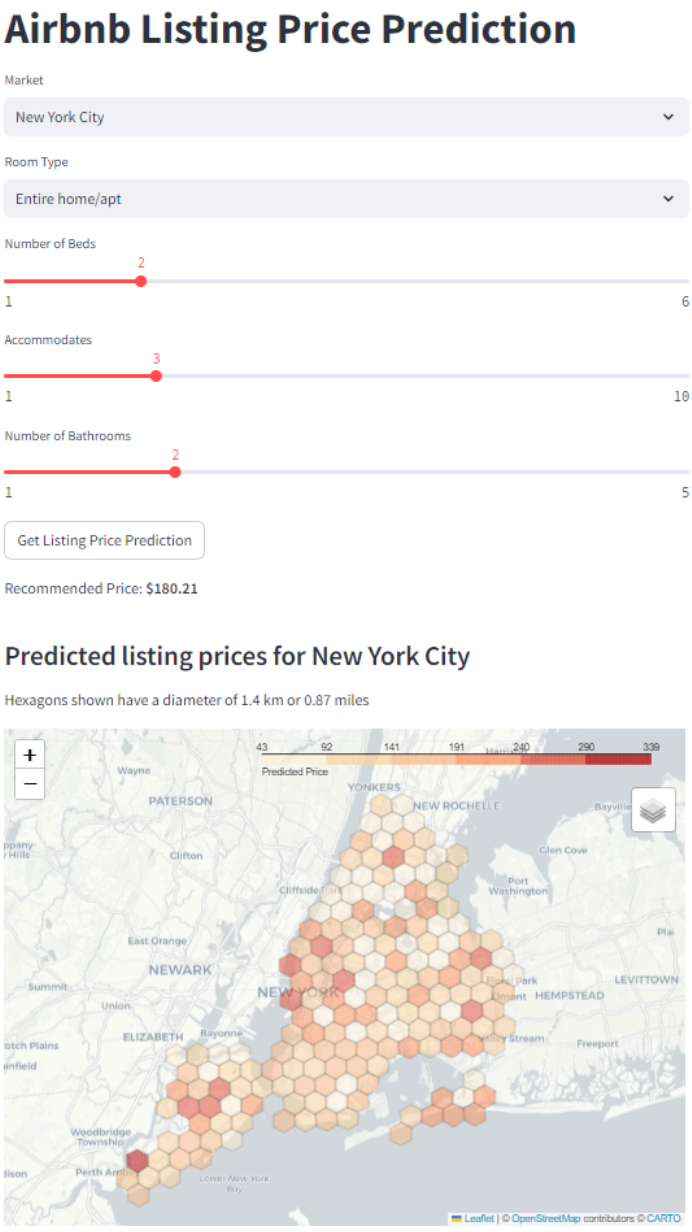
The development of our Airbnb Price Prediction Dashboard followed an agile, iterative approach, similar to our model development process. After each sprint, we conducted internal user testing, which served as our primary product evaluation strategy. By involving team members unfamiliar with the user interface, we ensured a seamless user experience throughout the dashboard.

We chose Streamlit as our development platform for its ability to rapidly prototype dashboards and provision of free public hosting on the Streamlit Community Cloud. This choice enabled us to iterate quickly and make the dashboard accessible to users without hosting costs.

Our dashboard includes the following pages:

- **Price Prediction:** This primary page offers users a recommended listing price for an Airbnb rental property. The prediction is based on factors such as market, room type, number of beds, accommodation capacity, and the number of bathrooms. We placed this feature on the first page to ensure easy access, anticipating it would be the most frequently used tool. Additionally, the map on this page includes a layer-switching option for the background, enhancing the user's ability to view and interpret geographic data.
- **Model Results:** The model results page provides transparency into our model development process. It offers insights into the models used for training and evaluation, helping users understand the accuracy and reliability of the predictions. This page was included to build user trust by offering a clear view of the methods behind the predictions.
- **Market Analysis:** The market analysis page provides valuable insights into the Airbnb market. Users can explore median prices for existing listings and examine the distribution of prices across

markets. This feature equips users with a better understanding of pricing in the overall market, helping them make informed pricing decisions for their own listings.



Conclusion

In conclusion, we developed an open-source predictive model and dashboard that allows Airbnb hosts to predict listing prices based on their property's characteristics without sharing personal data. The tool is user-friendly, reasonably accurate, and accessible to everyone. Future improvements could include incorporating real-time data updates, improving the performance of the model, and expanding the tool to cover more geographical markets.

Our tool has the potential to significantly streamline the pricing decision process for Airbnb hosts, enabling them to manage their businesses more efficiently while maintaining data privacy. However, we

are mindful of the ethical implications of pricing every location in eight major U.S. cities, especially in the context of short-term rentals, which have historically contributed to making neighborhoods unaffordable for local residents, as seen in places like New York City where the New York City comptroller's office found that roughly 9.2% of the rent increases imposed by landlords from 2009 to 2016 were attributable to the effect of Airbnb alone ("Why Cities Are Cracking Down on Short-Term Rentals"). Despite these concerns, we believe that the additional revenue generated by Airbnbs will ultimately outweigh the costs to longtime residents. By carefully considering the ethical implications of our work, we aim to create a balanced market information solution that benefits both local AirBnB hosts and the communities they live in.

Bibliography

1. Karkala, Deepak. "About the Project: Airbnb Price Modeling." Deepak Karkala, https://www.deepakkarkala.com/docs/articles/machine_learning/airbnb_price_modeling/about/index.html. Accessed 28 July 2024.

2. Mohamed Irfan, S. H. "Airbnb Data Science Project." Mohamed Irfan S. H., https://mohamedirfansh.github.io/Airbnb-Data-Science-Project/. Accessed 28 July 2024.

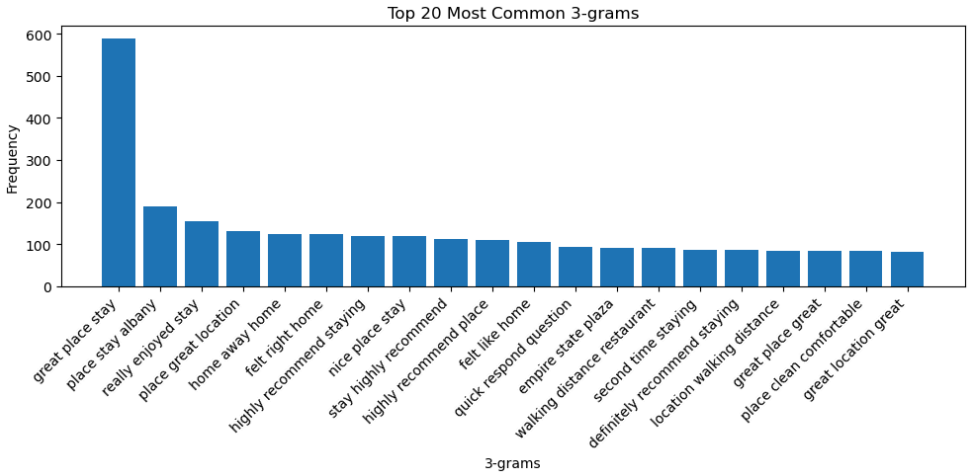
3. Gartrell, Nate. "Why Cities Are Cracking Down on Short-Term Rentals." Bloomberg, 9 July 2024, https://www.bloomberg.com/news/articles/2024-07-09/airbnb-abnb-vrbo-expe-why-cities-are-cracking-down-on-short-term-rentals. Accessed 28 July 2024.

4. Smith, John. "Exploring Location Data Using a Hexagon Grid." Towards Data Science, https://towardsdatascience.com/exploring-location-data-using-a-hexagon-grid-3509b68b04a2. Accessed August 3, 2024.

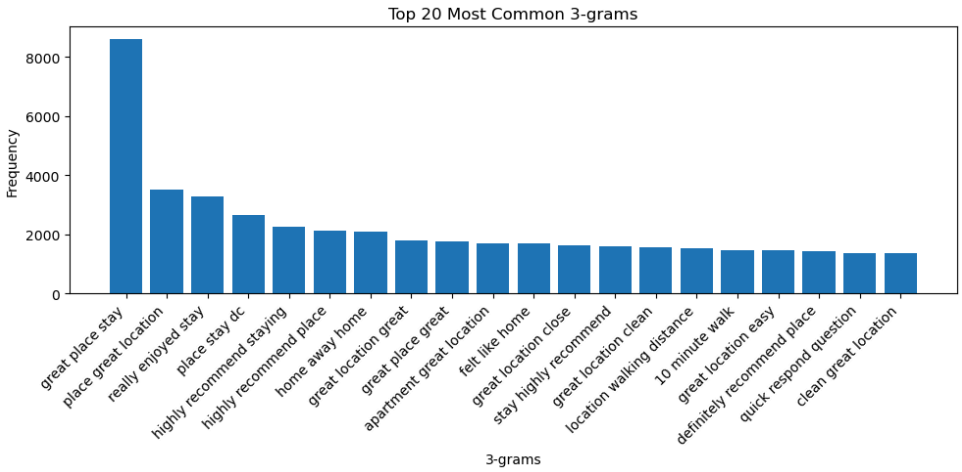
Appendix

1. 3-grams

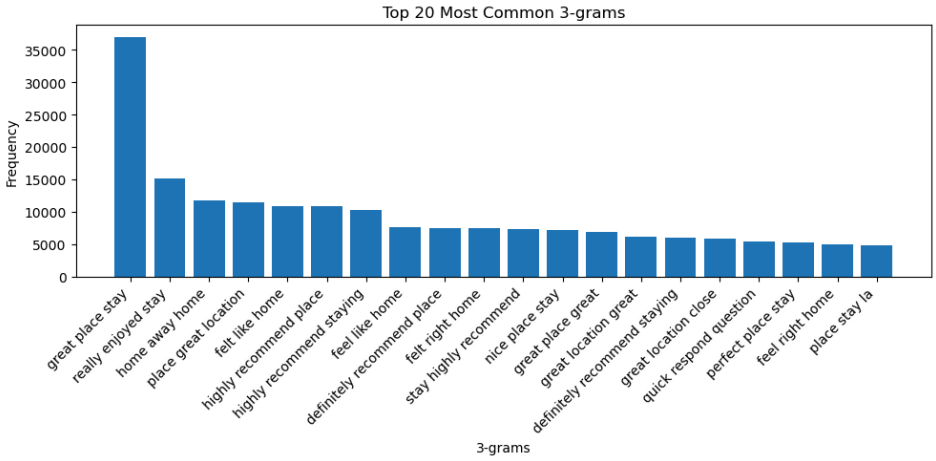
1a. Albany



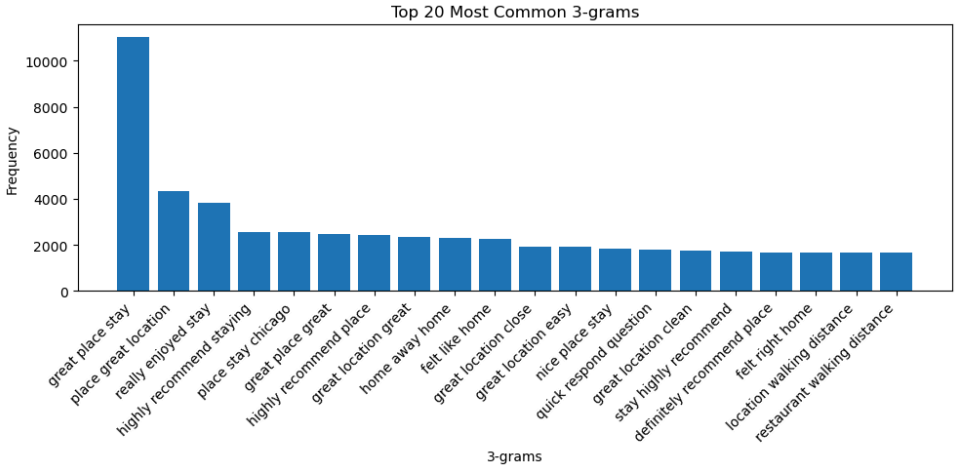
1b. Washington DC



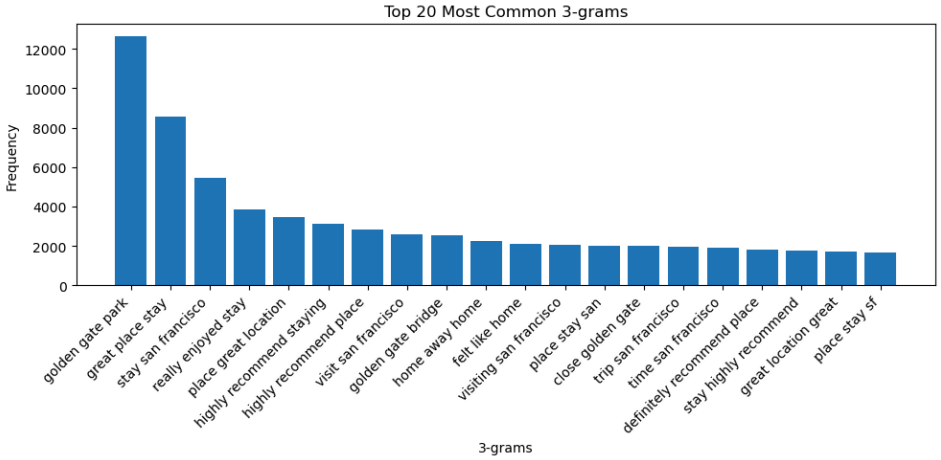
1c. Los Angeles



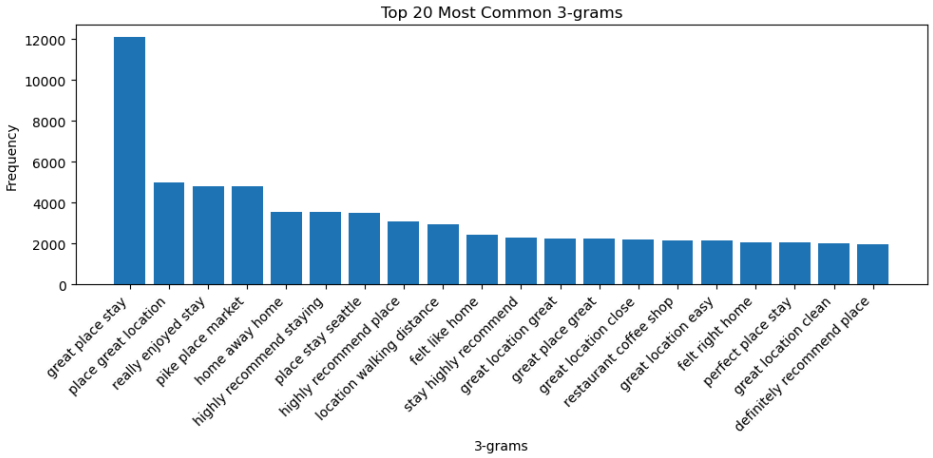
1d. Chicago



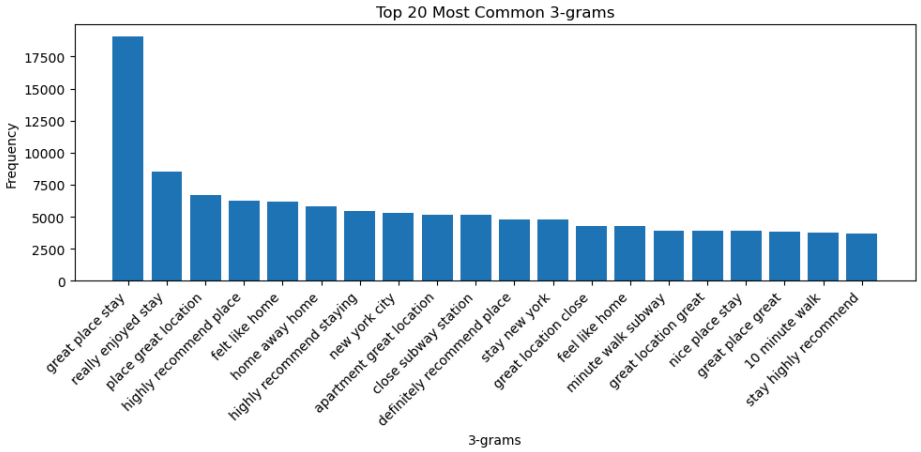
1e. San Francisco



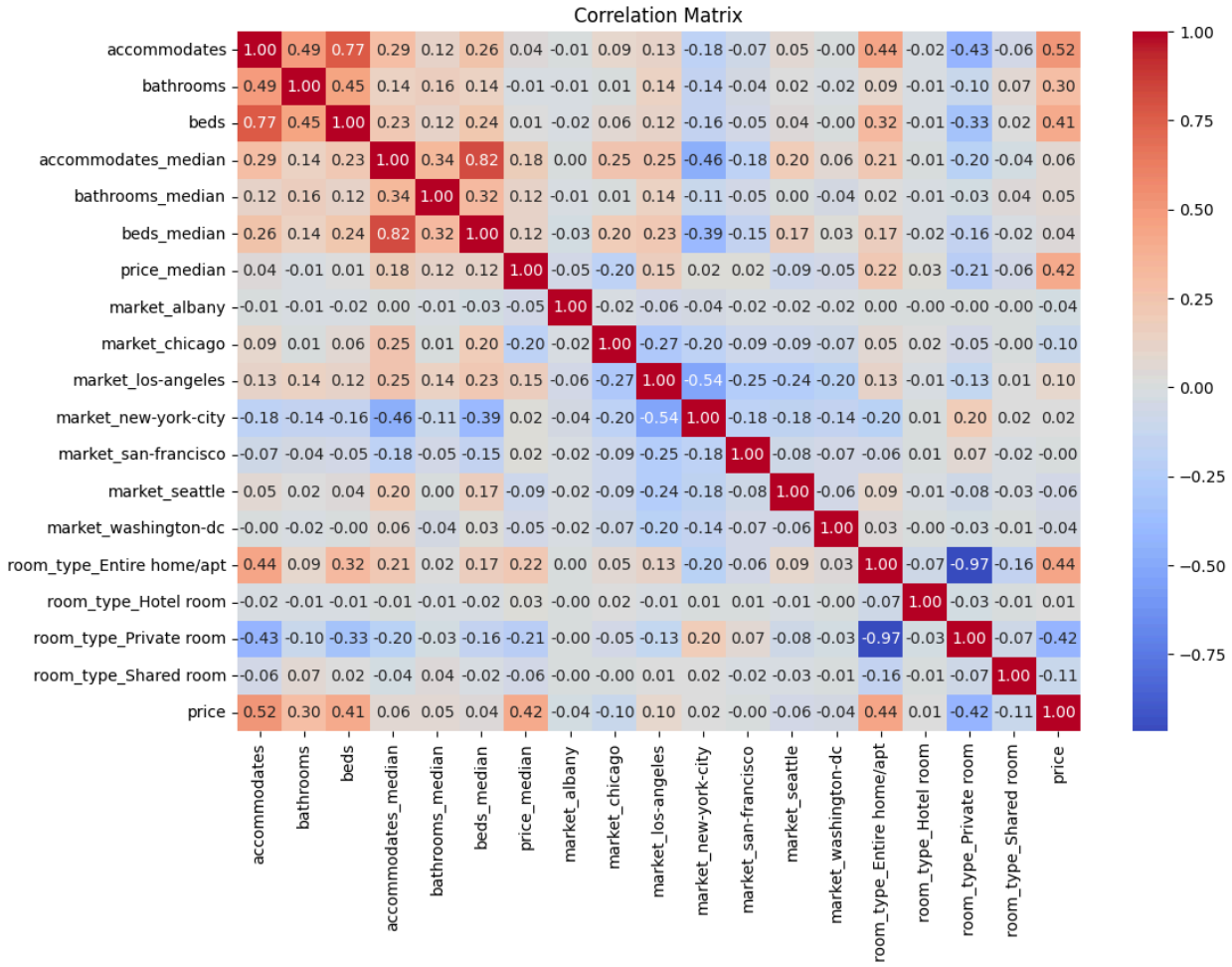
1f. Seattle



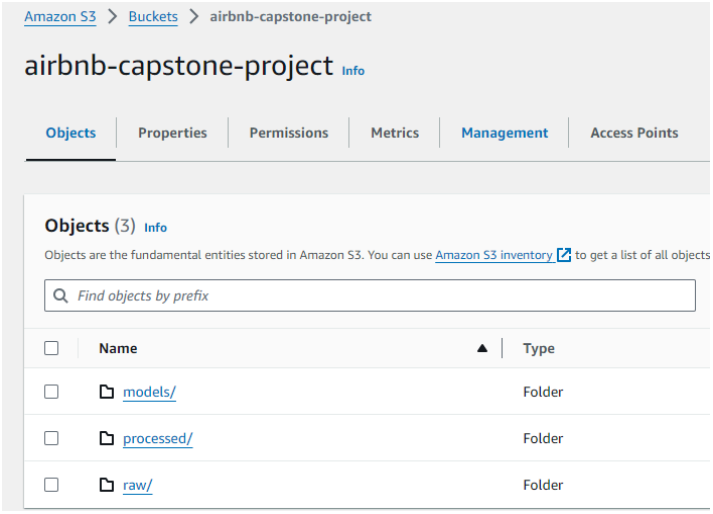
1g. New York City



2. H3 Model Correlation Matrix



3. AWS Setup



4. Snowflake Setup