Price Recommender Web Application for New Properties on Airbnb

SIADS 699 Capstone Project Report

Team: GMT8-9
(Yangkang Chen, Wei Li Tan, Masato Ando, Dongyao Wang)

Table of Contents

Project Overview

How should I charge my Airbnb listing? Owners typically consider several factors including prices around the neighborhood, the configuration of the property, and the cost of owning and listing the unit. While some systems have provided solutions to a reasonable pricing recommendation, they are either less comprehensive in features input or did not mention the "competitiveness" at that price. This project aims to create a web application to allow users to figure out the optimal price point that is competitive with the market while not undervaluing the property for new property owners looking to list on Airbnb. Besides that, we aim to provide users with further helpful information about why they should charge at this range and what additional features they can include improving their pricing power.

Dataset

- Airbnb Listings data in Los Angeles
    - This project focuses on Los Angeles (LA) using data downloaded from Airbnb (http://insideairbnb.com/get-the-data/). LA is one of the most popular locations for property lister and that offers a large dataset and high usability for the final web application of this project The data set contains over 40,000 unique listings with detailed information about the configuration of the properties, amenities, location coordinates, description, price, and more. As our project focuses on the perspective of property hosts, particularly new hosts, only features available at the point of sign-up are utilized.
- "Amenity Universe" dataset
    - This dataset is a collection of common amenities in Airbnb listings collected by a previous project (Lewis, Data cleaning in Python: examples from cleaning Airbnb data, 2019)
- COCO128 dataset
    - This dataset contains 128 daily objects and is commonly used for training an object detection model in YOLO (https://github.com/ultralytics/yolov5) series of models.
- RealEstate dataset
    - This data set is provided by Redfin (https://www.redfin.com/news/data-center/), and house market data is aggregated by Zipcode. This data has been generated monthly since 2000, and up till January 2023 for our model.

Methodology

The project can be broken down into several broad categories (see flow chart):

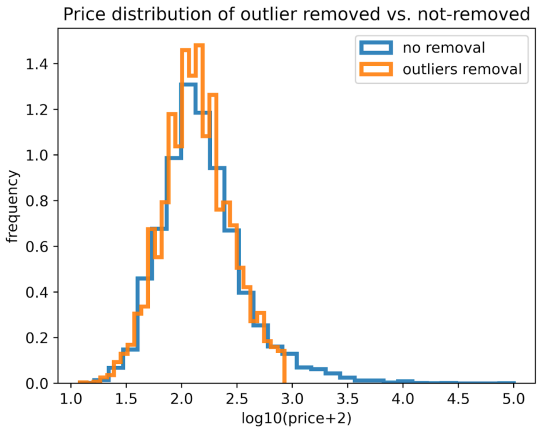
- Initial exploratory data analysis
- Feature engineering (four parts)
- Supervised machine learning model construction

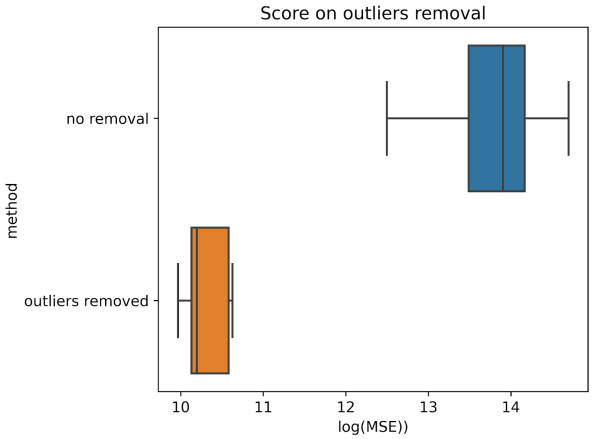
- Model integration into web application

Initial Exploratory Data Analysis (EDA)

In the initial exploratory phase, it is found that the price range of different units can vary greatly from the lows of below US$100 a night to US$99,999 a night. The high price does not always correspond with the property features, location, or amenities. Those price levels could just be a method for a host to rise to the top of search results when a user tries to sort the results based on the highest-to-lowest price. Those data points are considered outliers as they would skew the analysis and invalidate the analysis.

The percentile method is used where the top and bottom 2.5% of the data points are removed. This allows the model to predict a wide range of prices without being affected by the extremes, which are either irrelevant or exclusive, and a niche group of property hosts. This method also helps to preserve a larger set of data for subsequent analysis. This is one of the feedback from the mentor of the instruction team.





*RanksumsResult(statistic=2.61116, pvalue=0.00902)

Other than dependent variables of price, basic features are being explored too. They include the type of property type, neighborhoods, room type and count, the capacity of accommodation, bathrooms type and count, beds count, and minimum and maximum nights. However, the dependent variable remains the focus of the results and further exploration of other features will be discussed in the next section. While outliers are being identified during EDA, the actual removal of them is done after the feature engineering stage.

The data available for each property can be segmented into four parts below and these parts will be further discussed in the next section.

- Property configuration and amenities
- Location advantages
- Characteristics of top images
- Impression of the property description

Feature Engineering (Four Parts)

Based on the segmentation above, feature engineering is being performed for each category as different methods are used to extract its unique features. Then, each category is sealed into a class object forming a pipeline, with two main functions: “processing_Airbnb_data” (PAD) and “processing_new_data” (PND). PAD function takes in the raw data and extracts the designated features for the respective category for model training, while PND takes in the input from the user and converts them into features for model prediction. More details regarding the construction of the mode will be elaborated after this section on feature engineering.

The feature engineering section is broken down into four parts pipeline, there is a flow chart in the append for a better understanding of the idea and process.

Property Configuration and Amenities

In this section, several different techniques are used. They include using regular expression (regex) to retrieve different amenities for each property from the data, count vectorization, and categorical encoding.

The property configurations are represented by the type of property, room type, bedrooms, beds, bathroom type and counts, and neighborhood. While most data like bedrooms and bathroom counts are numeric in nature, other data like type of property and bathroom are categorical.

As regressor models are the primary model used for our price range prediction, categorical data are converted into numerical data by encoding the exhaustive list numerically. (i.e. private bathrooms are 1 and shared bathrooms are 0). Each configuration characteristic represents a feature of the property.

Moving on to the amenities section, the series of amenities data provided is a string data type. Regex helps to extract substrings that represent different amenities based on a specific pattern. Each substring is extracted and put into a list. After going through the list of amenities in the list, it is found that there are many variations to amenities due to the ability to customize it. For example, some owners specify the brands of shampoo and the brand of bedsheets to signal a more premium listing. Those additional qualities could be a pull factor when comparing two very competitive listings but may not be generalizable across the different listings. Hence, we took the approach of constructing an exhaustive list of possible amenities offered on Airbnb by going through their website and searching for materials online (Lewis, Data cleaning in Python: examples from cleaning Airbnb data, 2019). The amenities universe list contains over 250 amenities that the property host can select at the point of listing.

The final amenities list for each property is only determined if the item in the list of amenities extracted from using regex is inside the amenities universe list. Finally, the count vectorization

method from the Sklearn library is applied to the final amenities list to record the presence of specific amenities for a listing. Each amenity represents a feature of the property.

Location Advantages

Location, location, location! This is usually the fundamental driver for the base price of the apartment. The factors that are focused on include average property prices of similar configurations around the area and proximity to public transportation and places of interest. Public transportation typically includes bus stops and the metro. Being close to any of these increases the level of convenience for a renter to move about within the city. Places of interest include zoos, supermarkets, shopping malls, ferries wheels, and any other attractions in the city. People usually travel to cities to visit special attractions and being close would shorten any time wasted on traveling to those places thereby allowing property lister to command a higher price.

Similarly, another important proxy for how premium a location is can be seen from the property prices around the neighborhood. Though this typically takes into consideration things like proximity to transportation and places of interest, it also accounts for other factors including the residents of the neighborhood, the cost of owning a property in the area, or even the level of safety in the district.

As such specific Python libraries and an external dataset containing those key information is being utilized for analysis and modeling.

The map information, extracting latitude, longitude, public transportation, cafes, and other store information from the input address and displaying the map, is shown by geopy (v2.3.0) and overpy (v0.6). Since these libraries cannot extract zip codes, the library uszipcode (v1.0.1) was used for zip code lookups.

For the land price information by zip code, we used a dataset provided by the site Redfin (https://www.redfin.com/news/data-center/). The data is downloaded in a form of a CSV file from the site and it includes a time series of housing information and information by state, Metro, Neighborhood, and ZipCode.

Characteristic of Top Images

First impressions count! Humans are generally visual animals and develop a liking or not by looking at the top images of the property. While the interior design theme of the property is highly subjected to individual preference, there are certain features that can be extracted from images like level of contracts, brightness, energy, and hues. A dim-looking apartment might not appeal to some audiences as it might give off a spine-chilling effect and the place has been inhabited for some time. Similarly, a bright and well-lit apartment might give off a clean, well-maintained, and welcoming impression.

This section uses several libraries to analyze the images and they include CV2 (v4.7.0), Skimage (v0.20.0), YOLOv8 (ultralytics v8.0.58) trained on the COCO128 dataset.

A convolutional neural network is applied and the last layer of the network, containing 10 neurons, is used to extract the outputs. Aesthetic feature extraction and object detection are being used too. The last step is to investigate how well the image fits the rule of the third, commonly used by professional photographers as a guideline for a “perfect” photo.

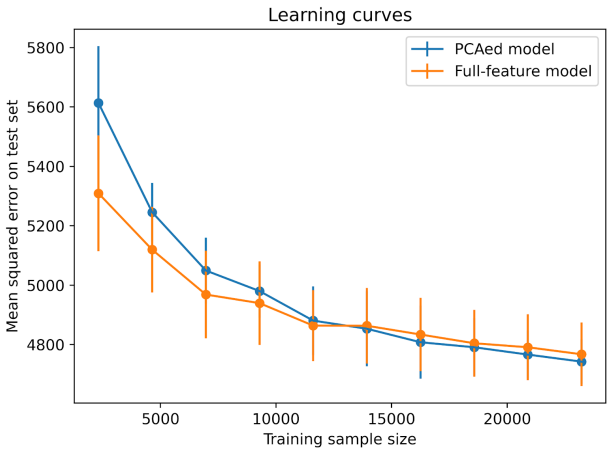
Impression of the Property Description

Similar to images, first impression matters, the tone of the description of the property also plays an important role in introducing the property and highlighting some of the vital attractive points of the property. Natural language processing (NLP) is being applied to the description of all valid listings in the dataset to understand whether the construct of a description has any contribution to the pricing ability of a host.

Here, we account for two types of features expressed in the host description: Named Entities Recognition (NER) and direct sentence embedding. In the NER section, we used a pre-trained model “en_core_web_lg” in Spacy (v3.5.1) (Learning, n.d.) package to query all FAC (Facility), LOC (Location), ORG (Organization) features in the description. These entities generally reflect how the host views the surrounding environment as a highlight in the neighborhood. Eye-catching entities will have a markup on the price of the listing. In the sentence embedding section, we used a MiniLM-based fine-tuned pre-trained model downloaded from hugging face (https://huggingface.co/flax-sentence-embeddings/all_datasets_v4_MiniLM-L6). The MiniLM is a distillation model of the larger “teacher” BERT-based model (Wang et al., 2020). It can be several times faster than the BERT-based models but still reach similar performance, therefore, it’s more capable of online data processing.

Supervised Learning Model Construction

After the extraction of all the relevant features and the creation of the respective pipelines for each segment of the dataset is completed, the construction of a synthesized model class called “My_Airbnb_Model” class is carried out. The model incorporates functions including four feature engineering processes, training/tuning process, and price prediction. Due to a large number of features available, dimensionality reduction and selection are also carried out in this



model class. The method for dimensionality reduction used is Principal Component Analysis (PCA). This helps to speed up the model training and prediction process during the construction of the mode. However, the final model uses the full set of features as there is a minimal lagging effect on the results generation process and based on the learning curve chart above, the results are similar. This helps to better match the actual feature with their respective influence. The final trained model is then saved into a pickle file which will then be integrated into the web application for the computation of a final price and the range of prices to give property owners discretion over how competitive they think the property should be.

The regressor model selected is Light Gradient-Boosting machine model (LightGBM). LightGBM is based on decision tree algorithms and is used for ranking, classification, and other machine-learning tasks (LightGBM, n.d.). One of the key advantages of using this model is the availability of quantile regression, which helps to show a range or reasonable distribution of prices based on the features entered by property owners. Besides, LightGBM is also supported by SHAP package usages and is way faster than XGBRegressor or RandomForestRegressor, making it possible to explicitly fine-tune the hyperparameters.

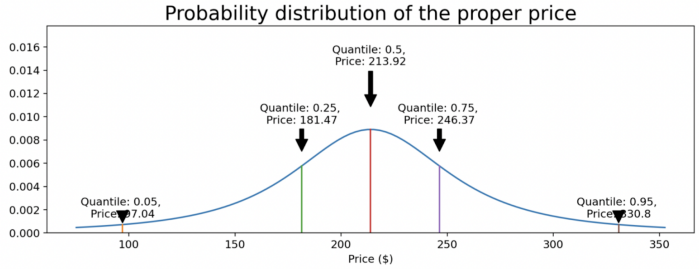
Even though the dataset provided by Airbnb offers good details about the features of a property listing, it does not encompass other important features necessary for ascertaining a precise price of a listing. Factors such as the age of the property, the interior design, and the condition of the property are not available in the form of data. Without those essential features, the price predicted by the model would have some tendency to under or overestimate the worth or price of the listing. Those missing factors generally open up room for subjectivity and are also reflected in the lower R-score of around 0.64 for the prediction.

This is where quantile regression comes into the model to add another layer of price range to present the uncertainty that can come with the trained prediction model. After adding that layer, the model is capable of suggesting the price, which is the 50th quantile, and also identifying the price at four other quantiles (5th, 25th, 75th, and 95th quantile).

The chart above shows the quantile model performance. Differently colored contours represent the five quantile models mentioned above. The black dashed lines represent y=x.

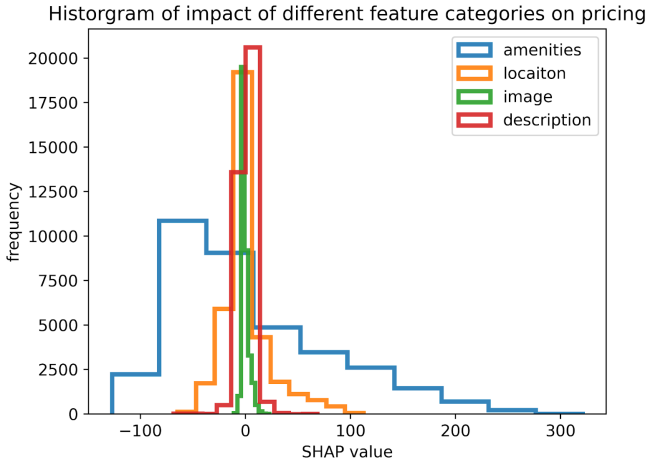
Suggested price: $ 213

	Quantiles	Price ($)
0	0.05	97.0399
1	0.25	181.4735
2	0.5	213.9194
3	0.75	246.3652
4	0.95	330.7988



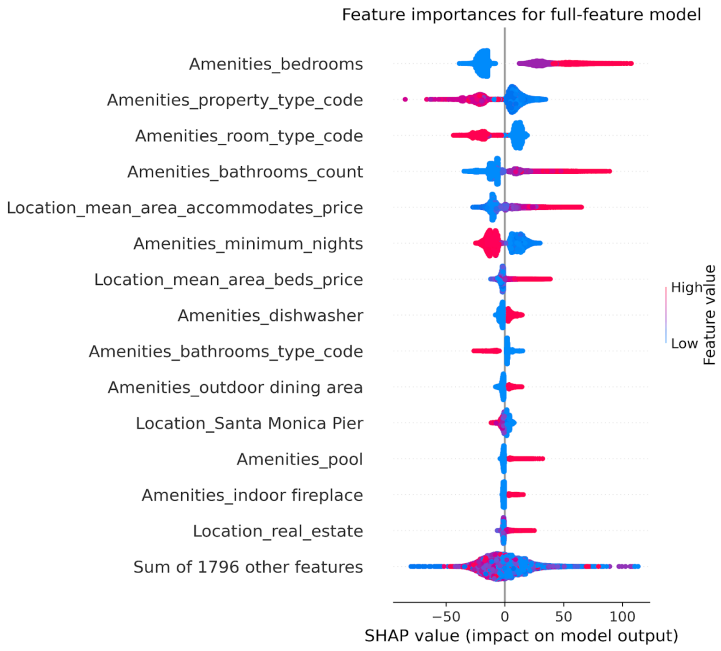
The model has also gone through hyperparameter tuning where the learning rate, max depth, and the number of leaves were put through Sklearn's Grid Search and Cross-Validation function to optimize the results. This step is performed on the model for both the PCA-ed and raw data for each quantile model.

The figure on the left is a snapshot of the part of the pricing report for the user. Those quantiles as shown in the figure on the left help users understand a reasonable upper or lower range based on the features entered.



SHapley Additive exPlanations (SHAP) package is used as a unified approach aiming to explain the output through the features of the LightGBM model. To explore the SHAP value by different categories, the mean value of all specific features in each category is used to construct the category-level importance index. We show that the amenities have the most significant impact on pricing among all four feature categories shown by the figure on the left, followed by location, description, and image. It shows the importance of house property and location as a hard standard and words and pictures as decorations.

The important features are also explicitly extracted from the final model to better understand which one has the largest influence on the final price of the property. The image on the right shows the top 15 features which have the largest influence on the prediction of the price of the property listing. The prefix on the like amenities and location simply reflects the category of the feature. Generally, the higher the number of red data points on the chart, the higher impact, while a positive SHAP value implies that a large number of the features has a positive influence on price and vice versa applies. For example, a higher number of bedrooms has a positive influence on the price, while a higher number of minimum nights has a negative impact on the price.



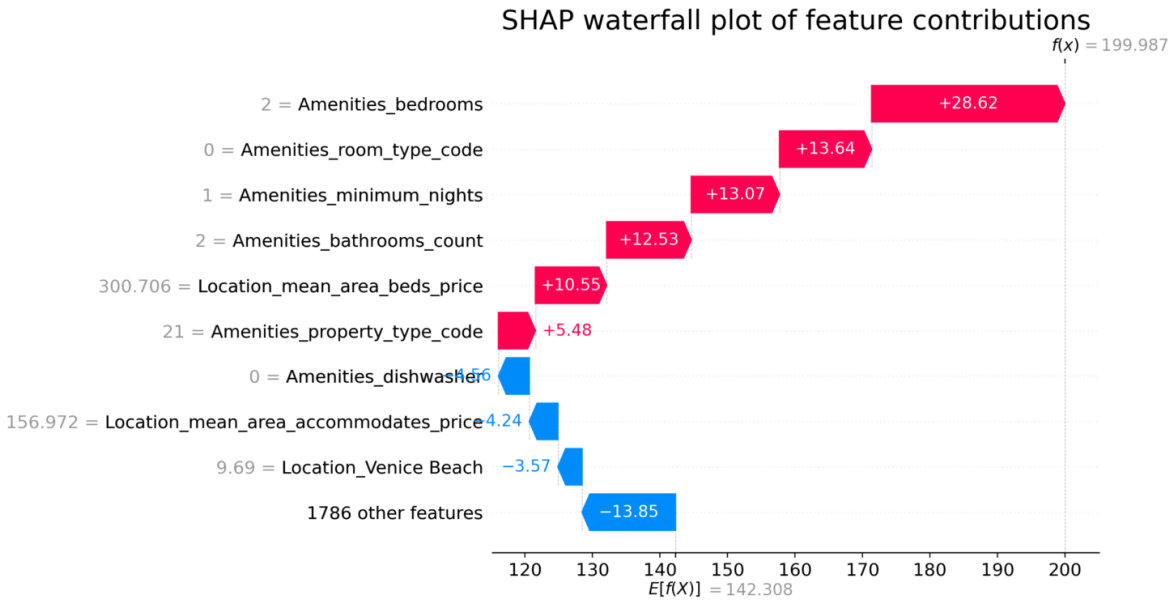
Web Application

Streamlit app is an open-source app framework for Machine Learning and Data Science applications. The application is hosted on Amazon Web Services (AWS).

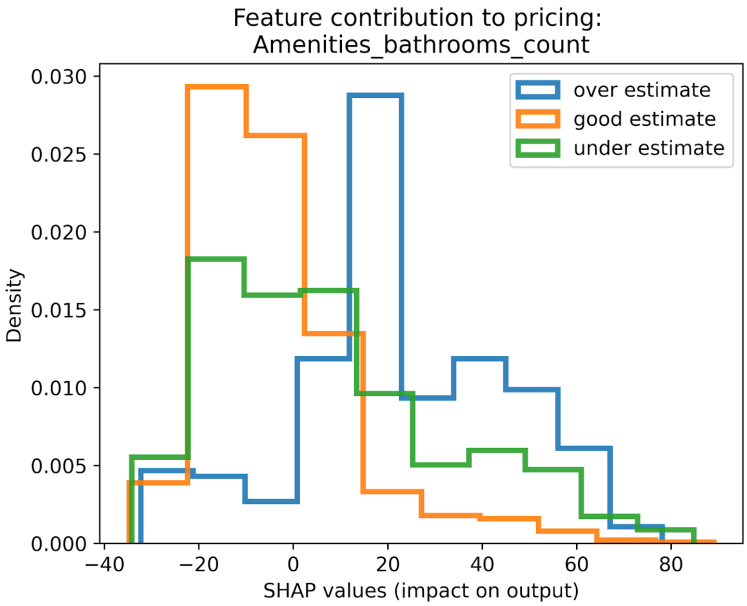
a. Create an EC2 instance: Start by creating an EC2 instance in our AWS account.
b. Connect our instance: Connect to your instance using SSH.
c. Install necessary software: Install Python and other necessary dependencies for our Streamlit app.
d. Built our app using Streamlit: Streamlit is a Python library that allows us to easily create web applications for this project.

The application aims to take users through a similar process of listing as Airbnb. There is a series of screenshots of the application for a clearer understanding of the interface. The user will start by identifying the configuration of the property, the amenities provided, top images, and finally writing a description for the apartment. All these inputs from the customer will be used for the prediction model.

Finally, the project aims to present several items to the user including the price range of the potential listing through the different quantiles predicted by the model and the factors that contribute to the price of the listing. The waterfall chart below is a snapshot of features' contribution by SHAP on the web application.



Failure Analysis



*SHAP values for bathrooms count in the three categories.

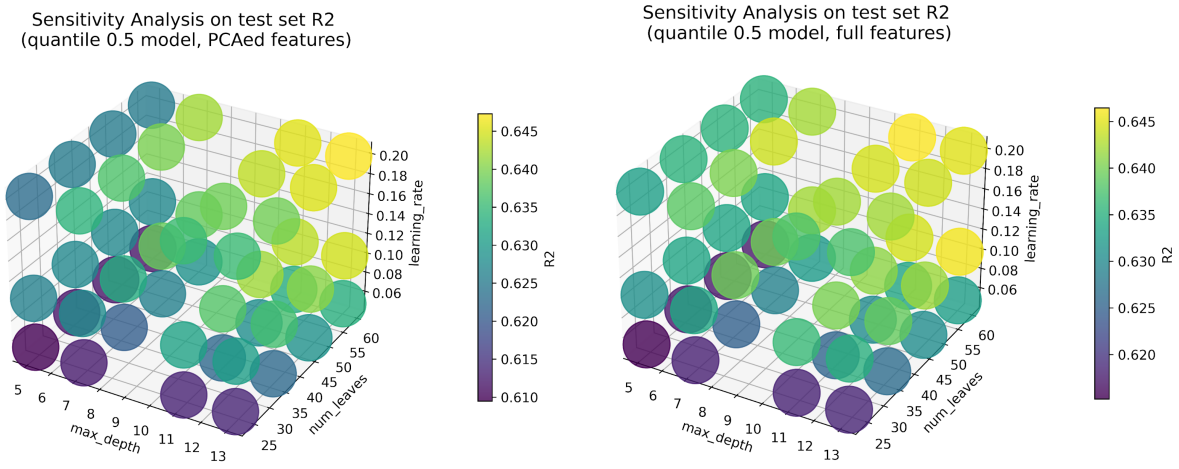
Failure analysis is being performed and the criteria are defined by records with predicted prices more than $100 higher than the true price are regarded as overestimates, and those less than $100 are considered underestimates. Records sitting in the middle are regarded as good estimates. By plotting the distribution of SHAP values (contribution to the output) for each high-impact feature, we can understand how the features play roles in these failures. In figure x, the model takes a high bathroom count too much weight so that it significantly contributed to high pricing in those overestimated samples, while, conversely, over-penalize those listings with low bathroom counts. It is possible to adjust hyperparameters in the future so that a single feature will not have an extensively high impact on the final output, which could reduce the sensitivity to a certain feature.

Another observation is that the quantile models are underestimating high price values, though they are generally covered in the 5th - 95th percentile range. This error could be introduced during the outlier detection phase, the removal of outliers results in a narrower-than-expected range of prices. This led to the failure to adequately capture the profile of high-price listings. Another reason could be that the price distribution follows a skewed normal distribution, with the main body located at around $50 to $400. High-price listings are rare and could lack the chances to be learned by the model. Although underestimates happened at the high-price point, the web application could suggest this bias to any user with high-value properties to reasonably increase their pricing since the ground true price is covered in the 5th - 95th percentile range.

This error can be reduced by further adjusting the outlier detection parameters (for example, allowing more high pricing values) to strike a balance between data integrity and data quality. Also, adjusting sample weights to lift over the importance of high-price listings may also help.

Sensitivity Analysis

Sensitivity analysis is conducted to test the variation of the results by adjusting the hyperparameters. Three hyperparameters with their respective values are learning rate (0.05, 0.1, 0.2), max depth (5, 7, 11, 13), and number of leaves (24, 36, 48, 60).



Charts above show sensitivity analysis with three hyperparameters, on both PCAed model (left) and the full-feature model (right). The color delineates the R2 score on the test set. Results show that the model is not sensitive to hyperparameters and maintained high scores even under poorly performed parameters.

Findings and Conclusions

There are several key takeaways and conclusions from working on this project and they can be broken down into two parts: the technical process of the project and the results from the Airbnb data.

First of all, the data available may not be able to provide the full set of information required to make a precise prediction. There will still be times when external knowledge about the industry is required to rationalize the score of the prediction. The R2 score for the prediction is around 0.64 due to certain missing information suggesting that some level of discretion is still involved when property owners list their property on Airbnb. The final model aims to suggest a price and a reasonable range of prices the host can select to use as a reference when exercising their discretion.

Key results that can be drawn from the machine learning model are the top features that generally contribute to the determination of the prediction of price. Examples of the top features include the number of bedrooms, number of bathrooms, presence of a pool, and location of the

real estate. As more of those features are included, the property owner is expected to be able to command a higher price.

Among all the features, the features of the property itself (like property type, bathroom/bedroom count), amenities, and locations situated as the top influential features, while description and pictures are considered less significant to pricing. This is within expectations, as the “hardware” and section decide the most, and images and descriptions are just decoration on those. The feature importance result also provides insights into the valuable efforts the host could put into if they want to increase their pricing capability.
Being nice and kind in description/introduction, mentioning significant sites surrounding your house, or taking fascinating pictures certainly work, but are not likely to change the order of magnitudes that have already been decided by the truth of your listings.

For the modeling part, the price prediction turned out to be more complex than we had expected. These sophisticated feature engineering and modeling pipelines only lead to around 0.64 R2, which possibly indicates the subjectivity in Airbnb pricing. Like any other supply and demand market, Airbnb pricing is also highly dynamic and involves multiple latent factors that lead to the final presented price.

Some limitations are being identified during this project. This project primarily focuses on predicting static pricing for each listing. While this is a simplification of the problem, real situations involve seasonal dynamics of pricing that change with significant events, festivals, and tourism. Future research could investigate the temporal dynamics by including time series prediction and evaluation. Luckily, these data are already available online.

Broader Impacts

As the world recovers from the Covid-19 pandemic, travel demand picked up steadily as reported by popular media outlets. This raises the demand for accommodations as well as business opportunities for property owners who have the intention to list their places to capture this trend. This price recommender can assist new and existing property listers to understand how they can price competitively and things that can be done to improve their pricing ability.

Ethical considerations

- Data privacy: The data provided by Airbnb contain a website link to the actual listing on the company’s website. Only public information chosen to be released by the property owners about the list is available for analysis. Limited personal information is being analyzed or needs to be revealed and no information is being stored during the recommendation process.
- Location bias: The website application declares that the recommender is only meant to suggest a price range in the Los Angeles (LA) area. The model may not be accurate for other cities in the United States or the world. The location section of the application contains a function to inform the user if an address outside of LA is being entered

Other Resources

- Web application:
  http://18.205.39.151:8502/my_app
- GitHub:
  https://github.com/foye501/Capstone_GMT89
- Report:
  https://docs.google.com/document/d/161fEv0t4Ops9SG5NPMAXZgnTigGPNvgPrR8gCyeM7x0/
- Video explanation:
  https://www.youtube.com/playlist?list=PL-Ih8lEqwhvFGjcMphHh4x3e4OoW4smun
- Poster:
  https://docs.google.com/presentation/d/1yLdUa_lTITbjeXRz0Wyespa-H65qcISo/
- Medium blog:
  https://medium.com/@chenyk_80392/beyond-a-single-price-pricing-range-improvement-suggestions-for-new-airbnb-hosts-a33ffa718fbb

Statement of Work

Wei Li Tan: Amenities analysis, final report consolidation
Yangkang Chen: Image analysis, model consolidation
Masato Ando: Location analysis, github
Dongyao Wang: Description NLP analysis, Web application

References

How Much is an Image Worth? Airbnb Property Demand Analytics. (2018, Sep). Retrieved from https://higherbookings.com/wp-content/uploads/2018/09/How-Much-is-an-Image-Worth-Airbnb-Property-Demand-Analytics-Leveraging-A-Scalable-Image-Classification-Algorithm.pdf

Learning, D. i. (n.d.). 15. Natural Language Processing: Pretraining. Retrieved from https://d2l.ai/chapter_natural-language-processing-pretraining/index.html

Lewis, L. (2019, May 16). Data cleaning in Python: examples from cleaning Airbnb data. Retrieved from https://towardsdatascience.com/predicting-airbnb-prices-with-deep-learning-part-1-how-to-clean-up-airbnb-data-a5d58e299f6c

Lewis, L. (2019, May 22). Predicting Airbnb prices with machine learning and deep learning. Retrieved from https://towardsdatascience.com/predicting-airbnb-prices-with-machine-learning-and-deep-learning-f46d44afb8a6

LightGBM. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/LightGBM

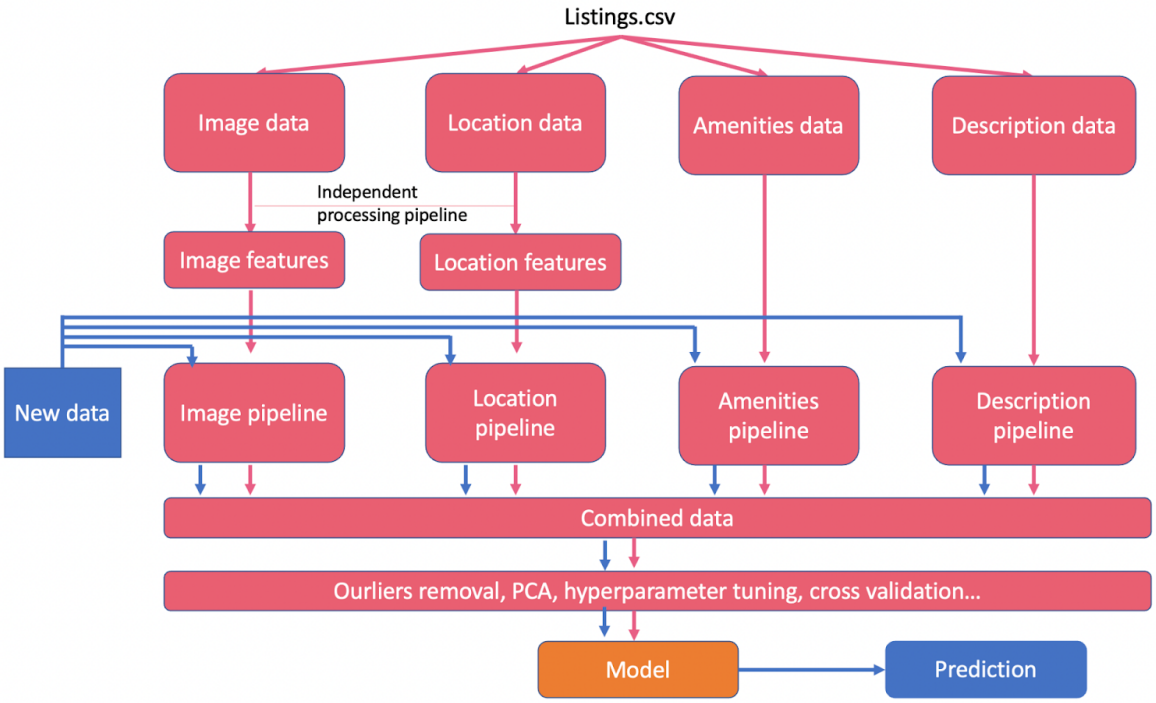
Spiegelhalter, D. (2017). https://doi-org.proxy.lib.umich.edu/10.1146/annurev-statistics-010814-020148. Annual Review of Statistics and Its Application, Vol. 4:31-60.

Wang, W. (2020, February 25). *[2002.10957] MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers*. arXiv. Retrieved April 11, 2023, from https://arxiv.org/abs/2002.10957

Appendix

Flow chart of the feature engineering pipeline

Overall Workflow



Amenity Pipeline

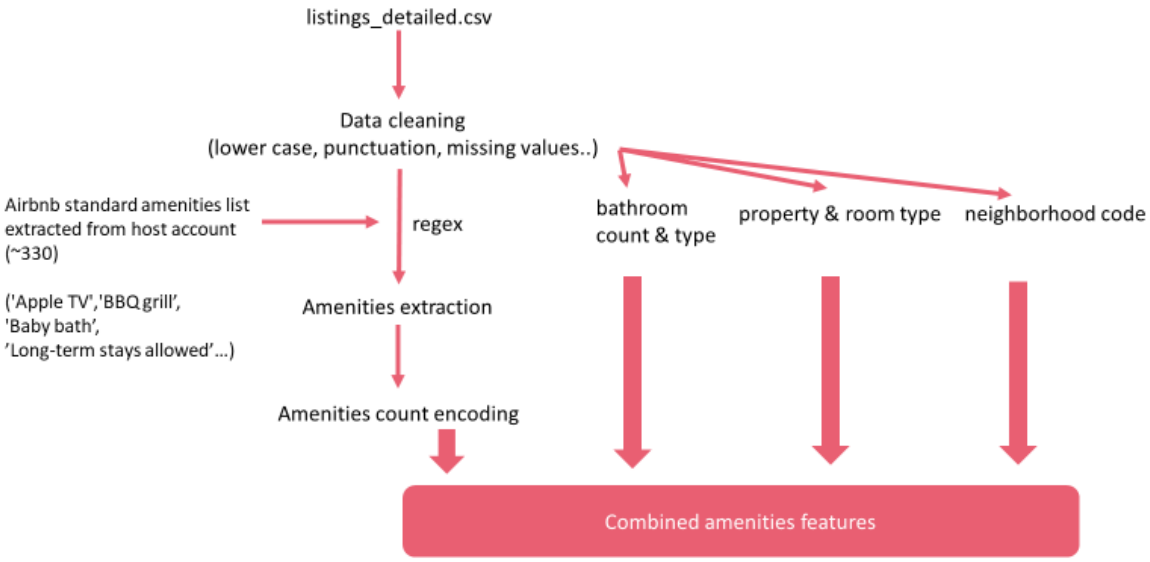
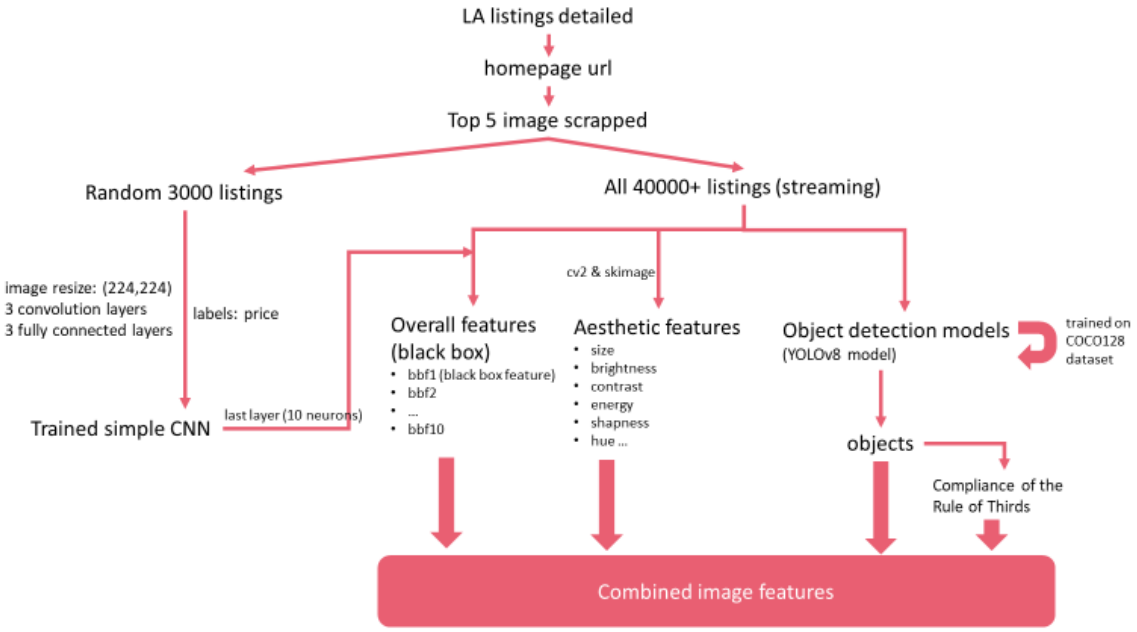
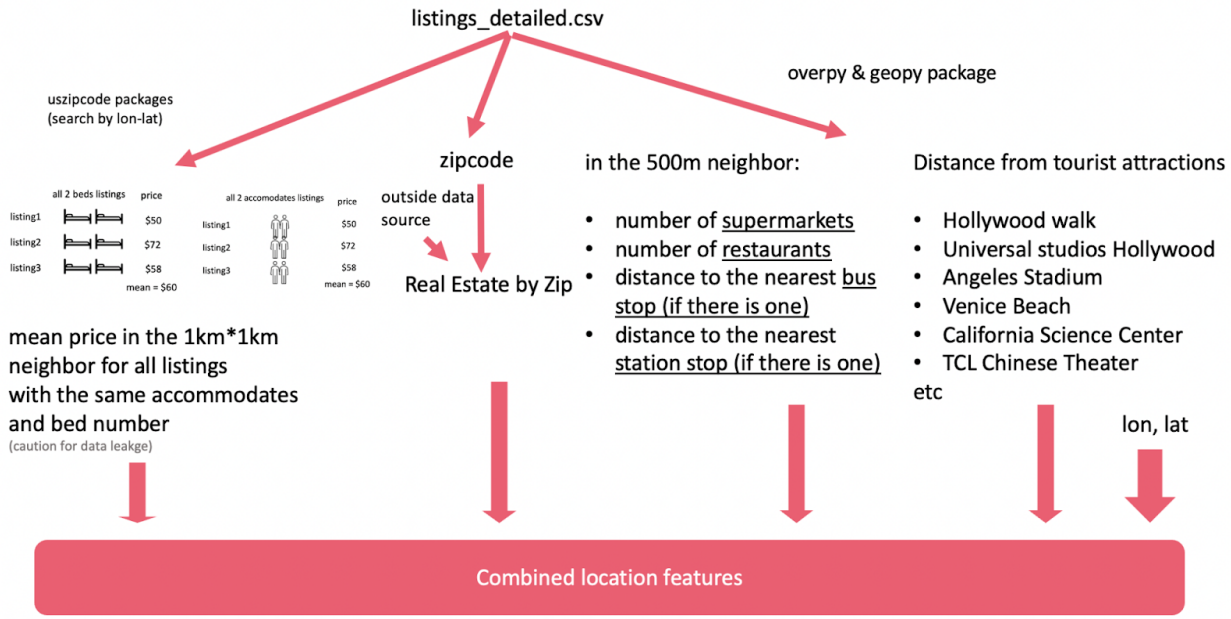


Image Pipeline



Location Pipeline



Natural Language Processing Pipeline