MSAI 349 - Machine Learning Team: Rhett D’souza, Albert Guo, Lukas Justen, Keith Pallo

Airbnb Price Evaluator

Note: This document serves as the formal write up for our project - hence, all figures are shown following the write up in the appendix.

Goal and Approach

Our goal in this project was to gain insight into the world of Airbnb market dynamics. There are several different ways to accomplish this goal, but more specifically, we attempted to predict the price for any Airbnb given standard measures such as the location of the listing, and the features that any particular Airbnb offers. We divided this task into three different stages. First, in our basic analysis, we sought to recommend a price for an individual looking to put a listing on Airbnb for the first time. This means that features dependent on past historical performance cannot be included (this is mainly reviews and host attributes). In our second analysis, which we call intermediate analysis, we extended our use case to current Airbnb users, by adding in historical features. We hypothesized that reviews play a strong part of customer perception, and adding them to our model would boost accuracy significantly. This would also help determine if any given Airbnb is properly valued given historical performance - which could be utilized by current hosts to assist with future pricing. Finally, we performed an advanced analysis on further features using deep learning. This was an optional objective - and we have completed it as such.

Dataset, Preprocessing, and Evaluation

Our dataset was provided from three different Kaggle repositories - detailing the same features for Boston, Seattle, and New York City (all 5 boroughs included). Links to each dataset are included in Table 1. Preprocessing of the dataset was performed using Trifacta and Python. The preprocessing scripts are included in the data folder and the parse.py file on Github. In order to evaluate our models, we calculated 5-fold cross-validated R^2 scores (train and test), where our task was to predict the price value for an unseen Airbnb. The features used for each analysis are shown in Table 2. In total from each file there are 104 potential features from all data sources (NY has some additional that we removed). All modeling work is done in the model.py file on our Github (currently best parameters are in the file).

Basic Analysis

Once our data was preprocessed, we included the basic features from Table 2, and ran multiple different regression techniques (note, we did not include features with a * in the table, and choose a 3,000 listing subset of NYC data). We choose to run Linear, Lasso, Ridge, and Elastinet standard regressions to help with different dependencies that may be present within the data, along with a random forest and a gradient boosted regressor to represent ensemble methods. Our initial results are shown in Table 3. Looking at these values, we were surprised that our standard regression techniques were performing much better than our ensembles on test evaluation. We dove into the data and discovered two key insights. First, thinking about how we would personally search for an Airbnb, we realized that location was likely the most important feature. However, when we examined our basic attributes we realized that longitude and latitude were the only features representing location - and our model may have not been powerful enough to truly utilize this complex representation. We revisited the available features and added in neighborhood - which is a location based tag provided to users querying for an airbnb listing. Additionally, we looked further into the New York dataset, as removing the tuples from our initial approach had significantly impacted our scores, as shown in the right hand side of Table 3. When visualizing our data with Tableau (shown in Figure 1), we realized that the New York dataset was not fully representing the market diversity well, as the dataset was not fully shuffled. Shuffling our New York dataset (shown in Figure 2) and adding in neighborhood - our train and test accuracy increased substantially as shown in Table 4. This accuracy bump represented a core aspect of machine learning that we discussed in class around the Netflix recommendation prize - relevant features with a simple model are significantly more important than extraneous features with a very complex model.

From here, we tried tuning the hyperparameters of each model, largely being unsuccessful. We again turned to examining our overall approach. We noticed that when modifying our parameters, there were signs of overfitting. Additionally, we dove into the actual prices of Airbnb’s within each city and noticed that there were some listings that had an astronomical price difference (over $700 per night) - particularly in New York. Figure 3 represents the distributions. To alleviate these issues we put a variance limit on features (features under a variance limit would not be included) and a price limit on listings to be put into our training set was applied. This pushed our model scores even higher - and they can be found in Table 5.

Intermediate Analysis

Up until this point, we were quite happy with the results obtained in the basic analysis - and had high confidence that reviews would push us into 90% training accuracy territory. However, just adding in our features from the intermediate analysis in Table 2 and running the same models (with variance and price limit) only increased our accuracy very slightly. We attempted to tune the parameters of the models - but we still were not able to beat our best basic analysis score by much. The results are shown in Table 6. We scratched our heads and thought - maybe we do not have ample data from New York - given that is is a larger market. We increased our NYC number of tuples from 3,000 to 6,000, but the accuracy actually decreased across the board. This work demonstrated two key learnings - New York is a significantly more difficult market to predict than others even with a substantial amount of data, and reviews are not as important as we first hypothesized. We believe this is due to a bifurcation of human scoring. People either tend to leave amazing reviews, or reviews that “bash” a listing. This creates a feature that is not all that great to actually discern the price. Additionally, we attempted to train a model on all attributes in the entire dataset (104 attributes), but our accuracy dropped further, illustrating the concept of overfitting well. This is included in the exploratory_data_analysis.ipynb and hyperparameter tuning.ipynb jupyter notebook files.

Advanced Analysis

As previously mentioned in our update - the advanced analysis was a potential objective. To make quick progress, we trained a simple neural network on just the listing description, as shown in the text_analysis.py file. Then, using TF-IDF and the TextBlob module we derived the sentiment from each description, and added it as a feature to our best performing previous model. This boosted our results very slightly for the ensemble methods but caused the other regressors to slightly decrease, which we believe that is due to overfitting. The results are included in Table 7.

We also attempted to use the created TF-IDF vector to train a deep neural network to regress the value a listing may receive as an overall rating score. Although the mean-squared error (MSE) of the network during the training phase dropped substantially over the epochs, indicating the trained state of the network, the testing MSE remained high. This indicates that there is no specific nature of a superficial description that results in better (or worse) ratings, or there are no specific keywords of a description that are good indicators of a particular rating.

Suggestions for Future Work

Our conclusions to our work are summarized at a high level on the main landing webpage. In terms of directing future work, there are many things that we would like to do. First, we would very much like to see how our model generalizes to other cities (with some training data). Also - we think trying to extract information from the images included could potentially be very powerful.

Group Member Work

Throughout the duration of the project, all group members put substantial work into the direction and execution of the work completed. However, segmenting the work as requested, Albert focused on exploratory feature analysis, Keith was responsible for model and feature selection, and Rhett and Lukas built and tuned the models. We feel strongly that each member was essential and had great value add.

Appendix:

Table 1: Data Sources

Data Source	Link	Number of Tuples
Boston	https://www.kaggle.com/airbnb/boston	3586
Seattle	https://www.kaggle.com/airbnb/seattle	3818
New York	https://www.kaggle.com/peterzhou/airbnb-open-data-in-nyc	44317 (3,000 or 6,000 examples used)

Table 2: Features utilized at analysis stages

Row Number	Basic	Intermediate	Advanced
1	latitude	host_response_time	description
2	longitude	review_scores_rating	picture_url
3	property_type	host_identity_verified	host_picture_url
4	room_type	number_of_reviews	
5	accommodates	host_is_superhost	
6	bathrooms	host_verifications	
7	bedrooms	reviews_per_month	
8	beds	host_total_listings_count	
9	bed_type	host_acceptance_rate	
10	amenities	host_response_rate	
11	guests_included		
12	minimum_nights		
13	cancellation_policy		
13	neighborhood_cleansed*		

Table 3: Initial Basic Results (No Neighborhood, NYC not shuffled)

	Total Dataset		Without New York	
Model	Train - R2	Test - R2	Train - R2	Test - R2
Random Forest	0.8851	0.1629	0.926	0.4349
Gradient Boosted Regressor	0.4623	0.0057	0.4129	0.3974
Lasso Regression	0.2414	0.3241	0.355	0.366
Ridge Regression	0.2823	0.3172	0.3894	0.3886
Elastinet Regression	0.251	0.2895	0.3423	0.3519
Linear Regression	0.2824	0.3166	0.3894	0.3884

Table 4: Second Round Basic Results (Neighborhood Added, NYC shuffled)

	Total Dataset		Without New York	
Model	Train - R2	Test - R2	Train - R2	Test - R2
Random Forest	0.9385	0.5853	0.981	0.8408
Gradient Boosted Regressor	0.4165	0.402	0.4176	0.4
Lasso Regression	0.3536	0.3749	0.4924	0.4908
Ridge Regression	0.4223	0.3982	0.5378	0.5269
Elastinet Regression	0.3436	0.3723	0.493	0.491
Linear Regression	0.4228	0.3877	0.5378	0.5269

Table 5: Third Round Basic Results (Neighborhood Added, NYC shuffled, Price limit and variance threshold applied)

	Total Dataset		Without New York	
Model	Train - R2	Test - R2	Train - R2	Test - R2
Random Forest	0.9655	0.7562	0.9821	0.8676
Gradient Boosted Regressor	0.5503	0.5449	0.6039	0.5985
Lasso Regression	0.4874	0.4855	0.5539	0.5532
Ridge Regression	0.5555	0.5481	0.6027	0.5952
Elastinet Regression	0.4878	0.4857	0.5361	0.5343

Linear Regression	0.5555	0.5481	0.6027	0.5952

Table 6: Intermediate Results (Same as third round basic, historical attributes added)

	Total Dataset		Without New York	
Model	Train - R2	Test - R2	Train - R2	Test - R2
Random Forest	0.9655	0.7562	0.9842	0.8875
Gradient Boosted Regressor	0.5503	0.5449	0.6161	0.6094
Lasso Regression	0.4874	0.4855	0.5191	0.5169
Ridge Regression	0.5555	0.5481	0.5791	0.5647
Elastinet Regression	0.4878	0.4857	0.5486	0.5443
Linear Regression	0.5555	0.5481	0.5791	0.5696

Table 7: Advanced Analysis Results

	Total Dataset		**Without New York**	
Model	Train	Test	Train	Test
Random Forest	0.9678	0.7672	0.9841	0.8874
Gradient Boosted Regressor	0.9999	0.77	0.9999	0.915
Lasso Regression	0.4653	0.4645	0.5447	0.5433
Ridge Regression	0.5488	0.5431	0.5986	0.5778
Elastinet Regression	0.5157	0.5115	0.5774	0.5691
Linear Regression	0.5488	0.5431	0.5986	0.5778

Figure 1: Unshuffled New York Dataset

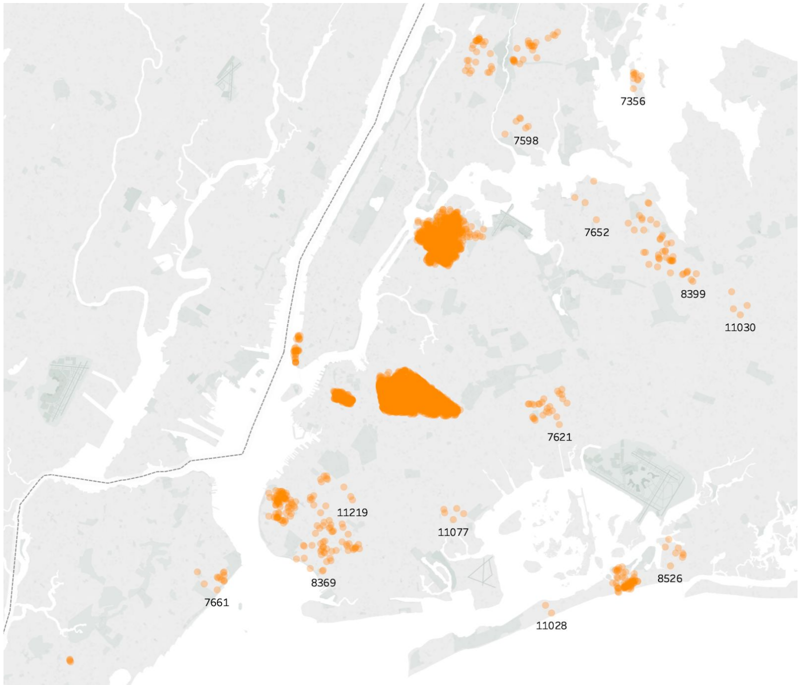


Figure 2: Shuffled New York Dataset

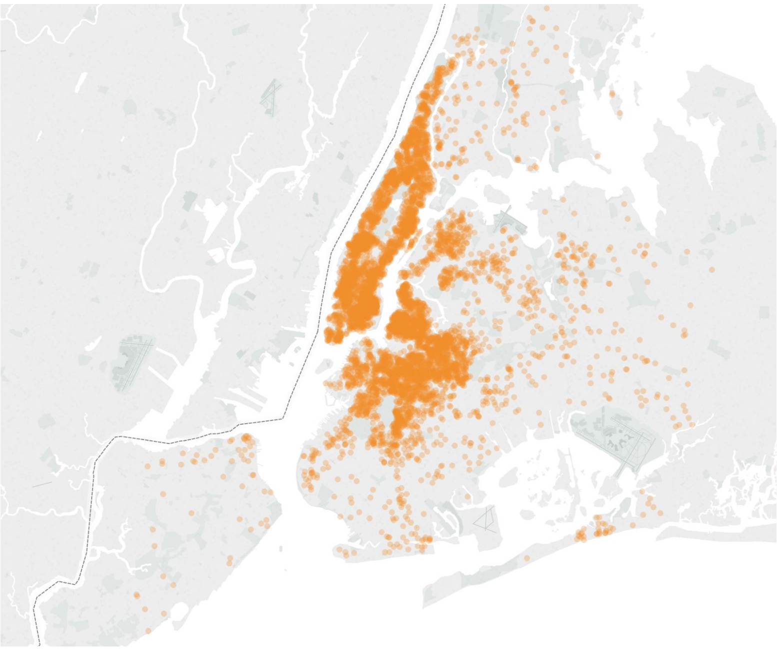
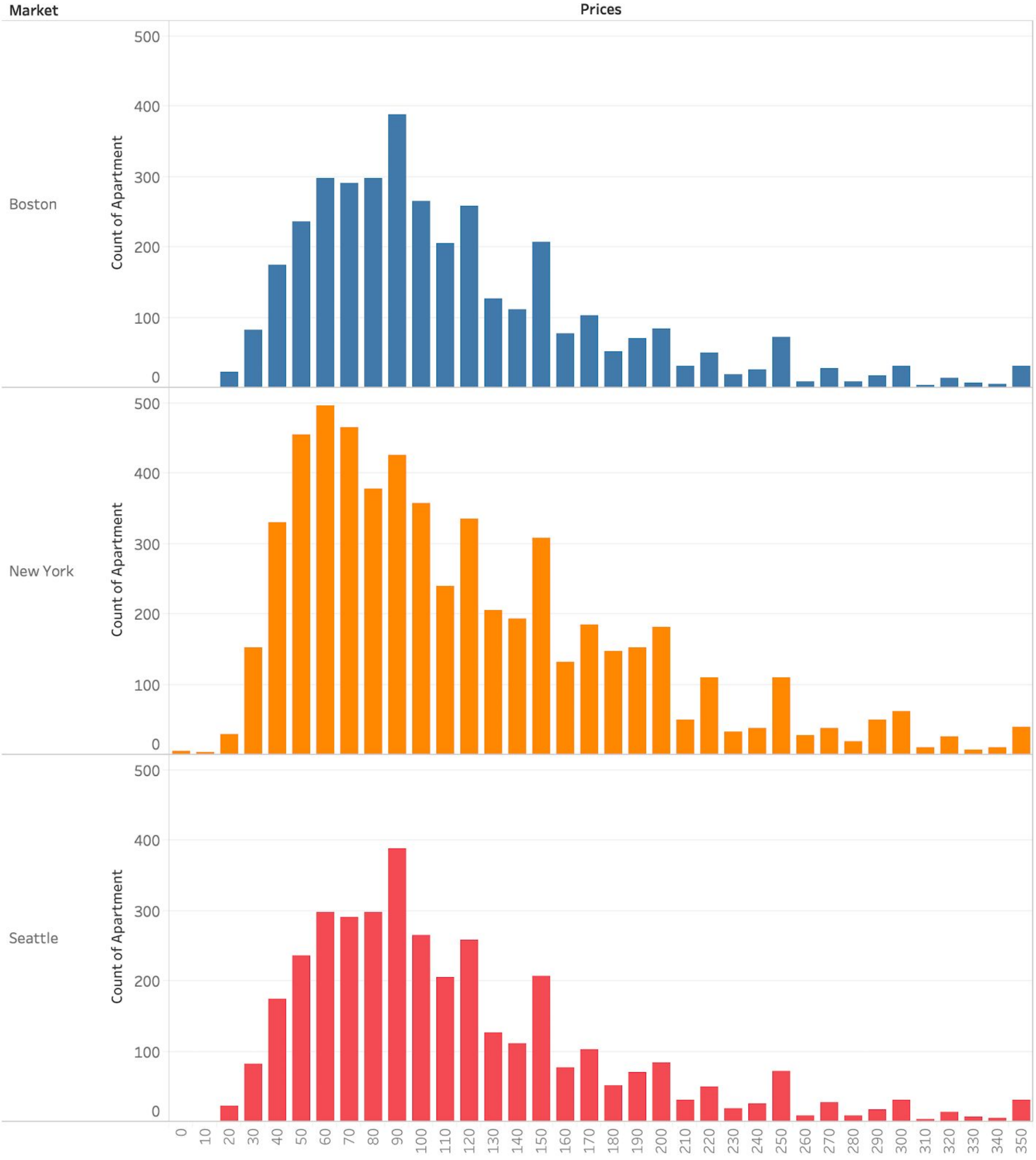


Figure 3: Airbnb Listing Price Distribution by City



Note: Large number of Airbnb’s in +350 per night. Many are over $1,000 in this range.