**User facing report**

A blue and white chart with black numbers

Description automatically generated

Cluster descriptions

* Cluster 1- The listings in this cluster are all average, no characteristics have any extreme values.
* Cluster 2- The listings in this cluster are characterized by their below average number of verified hosts.
* Cluster 3- The listings in this cluster are characterized by their below average number of beds.
* Cluster 4- The listings in this cluster are characterized by their above average number of minimum nights.
* Cluster 5- The listings in this cluster are characterized by their below average number of bedrooms.

Airbnb users should avoid: Cluster 2 for scams (unverified hosts), Cluster 3 for few beds (large groups), Cluster 4 for high minimum stays, and Cluster 5 for limited privacy (few bedrooms).

* **Final model:** I used the Boosted Tree model
* **Accuracy:** On average, the model’s predictions are within $67.63
* **Would you recommend moving forward with deploying the price suggestion feature, or do you think accuracy is too much of an issue?** In order to deploy a price suggestion feature for Airbnb listings the model should be more accurate as $67 is too much money to be off by.

A graph with text on it

Description automatically generated

* **Features associated with the price, and how?** The two most important features are the number of bedrooms and the number of people the listing can accommodate. More bedrooms increase price and the more a listing can accommodate increases price.
* **Features about the listing’s host associated with price, and how?** The most important feature about the host associated with the price is whether the host is verified or not. Verification increases price.
* **Next steps to improve:** An important attribute about a listing that would help predict price is the square footage. It may also help to know the last time the property was renovated or built if there have been no renovations.

Technical report

*Clustering* - steps and decisions

* **What data cleaning did you do before clustering?** I imputed any missing values with the mean, I logged columns that were skewed right, centered the data by subtracting the means, and scaled it by dividing the data by its standard deviation.
* **Are any features excluded from clustering? Why?** Before clustering, I removed all non-numerical data and removed columns that I thought didn’t have any valuable information. Categorical data types were excluded from clustering because clustering is based on measuring distances which cannot be done with categorical types. I removed columns like id, listing and picture URL, host id, and host URL because they have a very low correlation with price and do not add any valuable information that could help determine it.
* **What clustering algorithm did you use?** I used hierarchical clustering
* **How did you choose the number of clusters?** I used the plot\_mean\_by\_label\_table() function for different values of k and chose 5 clusters because each cluster had a defining attribute. 6 clusters were too many because each was not a distinct grouping.

*Price prediction* - steps and decisions

* **What data cleaning/feature engineering did you do before modeling?** I imputed missing data with the median of the column, logged columns like accommodates, bedrooms, bathrooms, beds, minimum night, and a few others due to their right skew, converted categorical columns into numerical using step\_dummy, and removed predictors with low variance.
* **Are any features excluded from modeling? Why?** I removed columns like id, listing and picture URL, host id, and host URL because they have a very low correlation with price and do not add any valuable information that could help determine it.
* **What was the model selection & tuning process?** The model I selected was the boosted trees model. I tuned hyperparameters mtry, trees, min\_n, tree\_depth, learn\_rate, and sample\_size via cross validation.
* **What models were considered, and why did you choose yours?** Other models I tried were a support vector machine, decision tree with cross validation, and linear regression. The SVM accuracy was 126.9, the decision tree was 122.4, and the linear regression model accuracy was 124.3. I chose to use the boosted trees model because it is the most accurate (67.6) and does a good job of combatting overfitting.
* **Model and algorithm:** Boosted decision tree/XGBoost
* **Hyperparameters and their values:** 2 was chosen for the mtry hyperparameter in the boosted trees model. This indicates the model is limiting the complexity to prevent overfitting. The optimal number of trees was determined to be 2000 which indicates that adding any additional trees would not increase the models accuracy enough to be worth the time and storage. Min\_n was tuned to 2 by cross validation indicating a very deep tree allowing the model to recognize complex patterns. 15 was chosen for tree\_depth indicating a deep and complex tree. The optimal value for learn\_rate was determined to be 0.1. This means that each tree will make small improvements each iteration instead of large improvements, which could cause the model to overshoot the best fit. Sample size was tuned to 1 via cross validation which means the full training dataset was used to train each tree in the model.

**Kaggle Submission:**

A screenshot of a computer

Description automatically generated