Data Mining Project Report

Praveen Doluweera, Nick Tagenhorst, Jeffery Kong
Data Mining
Georgia State University

Abstract - This data mining project attempts to use the K-Nearest Neighbor classifier to predict a star rating of a listing on AirBnB. This classifier is fitted with the dataset provided by AirBnB.

I. Introduction

Now in the 21st Century, traveling has been more accessible than ever across the globe. Individuals are able to now travel the world at affordable rates; this increase in travel has sprouted new businesses and industries never seen until recently. One such business: AirBnB, turned the hotel industry on its head by allowing normal individuals to rent out their own property to travellers. However, in some cases, the AirBnBs that an individual or group rents might not have a rating. Even moreso, potential hosts would like to know how their property would be rated by their customers. Because of these reasons, this project attempts to mine the ratings of AirBnB listings in certain cities to find what makes a listing on AirBnB's website receive a rating it does. This is done by using a dataset that contains features that are associated with a listing on AirBnB as well as the correlated rating. That dataset will be mined, reviewed, and fitted through a K-Nearest Neighbor classifier to create a classifier to predict how a listing will be reviewed and what star rating it will receive. However, we will utilize the same dataset coming from three different regions across the globe: Amsterdam in the Netherlands, Los Angeles in California, and Melbourne in Australia. This will provide classifiers, each with its own training based on the different regions.

II. Data Processing

As mentioned before, our dataset contains information about a listing on AirBnB's website. This data set is provided from insideairbnb.com. The features the dataset provides are: ID number, host name, description of the property, property location, how many the listing accommodates, number of bathrooms, types of amenities, as well as the average rating that listing currently has. The feature count for our dataset to 23 features.

However, the raw dataset is not applicable to be fitted in SKLearn's KNN algorithm. The data first needs to be cleaned and preprocessed so the KNN algorithm can work. When cleaning our data, we try to maintain only the information that would affect a review score on AirBnB. AirBnB does its review scoring in quite a unique way. Each listing is reviewed on six different metrics: Accuracy, Cleanliness, Check-In, Communication, Location, and Value. On the dataset, each is reviewed from a 1-10 range. The total review score is then summed together from a range of 1-100. AirBnB star rating is a visual representation of the total review score. The stars go from 0 stars to 5 stars, which is just the total review score divided by 5.

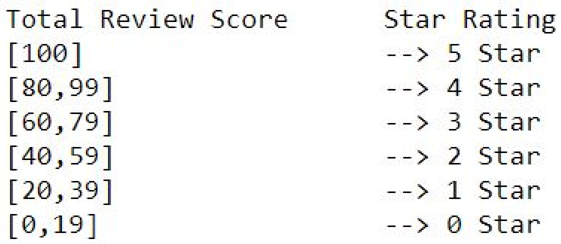
So, for our finalized dataset, we try to select features that affect the review score. In our finalized datasets the features that were selected were: host response time, response rate, acceptance rate, whether or not the host is a superhost, property and room types, price, amenities, and total review score.

After we have selected all of the features we wanted for our cleaned dataset, we began the process of label encoding all categorical data in the dataset. The categorical features in the dataset were: Property Type, Room Type, and Is Superhost.

Also, the K-Nearest Neighbor algorithm uses a distance metric to calculate the distances between observations, therefore it is heavily swayed by outliers and values with a large degree of ranges. Therefore the numerical features were standardized as well.

Lastly, we wanted to use the K-Nearest Neighbor classifier instead of the K-Nearest Neighbor regressor. Our current target feature, which is the total review score, was from range 1-100. Predicting this feature would be a regression problem. However, we can convert this into a classification problem by assigning each segment of the 1-100 range a class. In this case it would be a star rating. Listings with a certain total review score will be classified as follows.

TABLE 1:



```
Total Review Score      Star Rating
[100]                   --> 5 Star
[80,99]                 --> 4 Star
[60,79]                 --> 3 Star
[40,59]                 --> 2 Star
[20,39]                 --> 1 Star
[0,19]                  --> 0 Star
```

The K-Nearest Neighbor classifier will compare testing data with its K-Nearest Neighbors' classification, and classify the testing data with the label with the most occurrences; the labels being the neighbor's star rating. K-Nearest Neighbor regressor was also a considered option. If we were to keep the original 1-100 range total review score, the K-Nearest Neighbor regressor would average out all the K-Nearest neighbors' total review score. But ultimately we decided on sticking with the classifier.

III. Correlation and Covariance Matrix

Data analysis was conducted on our cleaned dataset to gain better understanding about its features. The first analysis our group did was printing out the covariance and correlation matrix of each of the datasets from each region.

See Appendix for Correlation and Covariance Matrices.

Upon doing the covariance matrix of each dataset from each region, we notice some interesting trends.

First and foremost, each dataset from each region does not seem to drastically differ from one another. All three datasets contain the same patterns in their covariance and correlation matrix.

Second, there appears to be a minor positive correlation between the host being a super host with the total review score, as well as host response rate and total review score. This is suspected since the customer's interaction with the host is a key factor in calculating the review scores on AirBnB.

Some other interesting patterns occurred such as a relatively high correlation between the price of the airBnB and the number of beds available. This makes sense since a larger number of accommodations would mean a larger price as well.

IV. Principal Component Analysis

To visualize our dataset, Principal Component Analysis was conducted to dimensionally reduce our 23 features to its 2 principal components. We then used these two principal components to graph the dataset as long with its label, by color coding each observation by its label. Here are the 3 graphs of each dataset from each region:

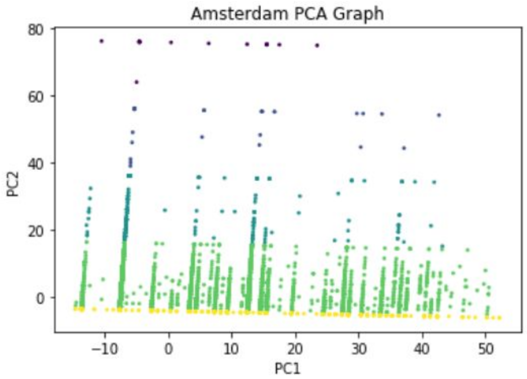


Fig. 1. Amsterdam PCA Graph

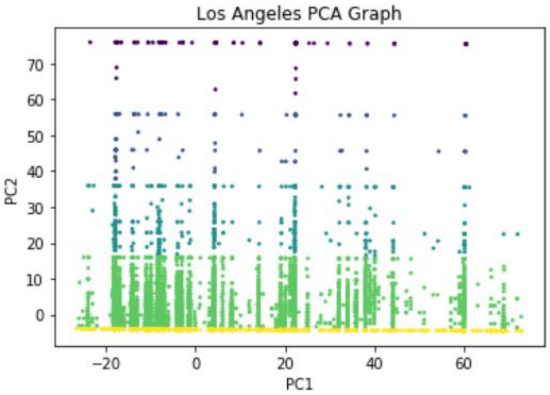


Fig. 2. Los Angeles PCA Graph

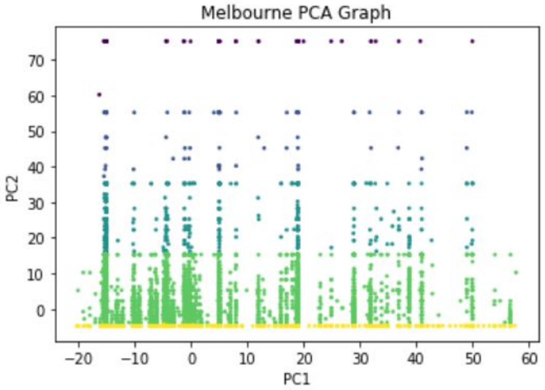


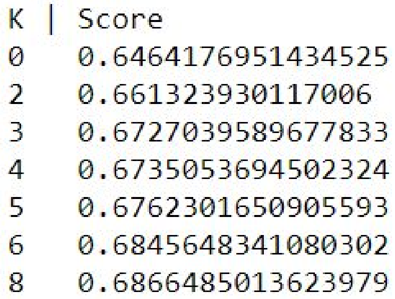
Fig. 3. Melbourne PCA Graph

When visualizing the datasets in figures 1-3, it seems that observations with a star rating of 4 (Light Green) dominate most of the datasets. Star ratings of 5 (Yellow) are mostly condensed at the bottom of the graph. Star ratings of 1 (Purple) is extremely scarce with few instances at the top of the Principal Component Analysis graph. However, the labels of all the observations are spreading out across the second principal component.

V. Training K-Nearest Neighbor

To begin the process of K-Nearest Neighbor on all three datasets, we first want to find what number for ‘K’ best suits our datasets. Since all three datasets are similar as concluded from the data analysis, we ran the K-Nearest Neighbor algorithm and scored the Amsterdam dataset with multiple ‘K’ values from 1 to 30, and chose the ‘K’ value with the best score. When iterating through the ‘K’ values, 8 was found to be the best ‘K’ value with an accuracy of around 69%. Here is a table showing the iterative process of finding the best ‘K’ value:

TABLE 2:



```
K | Score
0   0.6464176951434525
2   0.661323930117006
3   0.6727039589677833
4   0.6735053694502324
5   0.6762301650905593
6   0.6845648341080302
8   0.6866485013623979
```

Now that the best “K” value, our group wanted to train the K-Nearest Neighbor classifier with the best training-testing split. To do so, K-Fold Cross Validation was performed on each of the 3 datasets. Doing so created 3 separate classifiers which were trained on data from 3 different regions of the world.

5-Fold Cross Validation was chosen as the right amount of folds to perform on our dataset. After researching which K-Fold Cross Validation is best, it was concluded that for the dataset of its size, having more than 5 folds is unnecessary and having less than 5 folds would cause the training of the K-Nearest Classifier to be skewed.

The best trained estimator will be chosen as the final estimator to be used for its respective region to predict a hypothetical list of AirBnB listings. After 5-Fold Cross Validation was conducted, accuracies increased from 69% up to

75%. The score of each region's classifier is as follows:

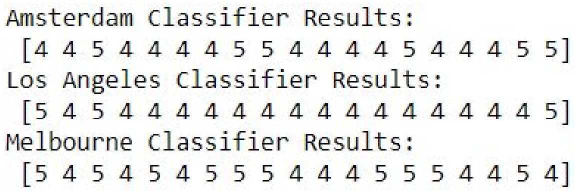
Amsterdam Classifier: 72.5% Accuracy
Los Angeles Classifier: 72.6% Accuracy
Melbourne Classifier: 74.9% Accuracy

See Appendix for full scoring.

VI. Predicting Hypothetical Listings and Results

Once we have our 3 K-Nearest Neighbor classifiers, we use each of them to give their predictions on a hypothetical list of 19 AirBnB listings to see what their predictions would be. This hypothetical listing attempts to be as diverse as possible; prices are varied from low to extremely high prices, different property types were included, and varied amounts of host response rate and time was added as well.

The hypothetical listing was then sent through to each classifier. Here each classifier's results for the 19 listings in the hypothetical list.



```
Amsterdam Classifier Results:
 [4 4 5 4 4 4 4 5 5 4 4 4 4 5 4 4 4 5 5]
Los Angeles Classifier Results:
 [5 4 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5]
Melbourne Classifier Results:
 [5 4 5 4 5 4 5 5 5 4 4 4 5 5 5 4 4 5 4]
```

Fig. 4. KNN Results

VII. Summary

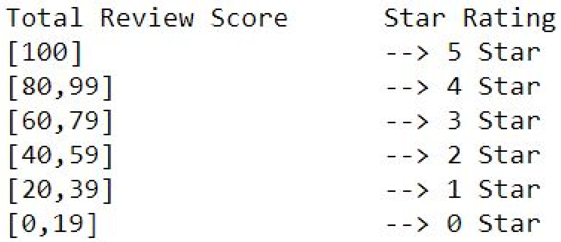
This project was very interesting to do, because it showed us what kinds of things we would be doing in the real world if we were to go into Big Data and what Data Mining is. Searching across different parts of the world to find the right data to use for our dataset was interesting and we could see how different actions or situations could affect the rating of the listed AirBnb.

A struggle that we initially had was trying to figure out where we were going to get our data from. We did not want all of our data to be from one specific area only to find out later on that the predictions that we had made did not translate well across to different regions of the world, that is why we chose Amsterdam, Melbourne and L. A. as the locations to pull the data for our test set.

We ultimately feel that this project would help people who want to get into the AirBnb business find out if their listing would get good reviews or not and would potentially help current owners of AirBnb listings to get the ratings on their listings to a higher score.

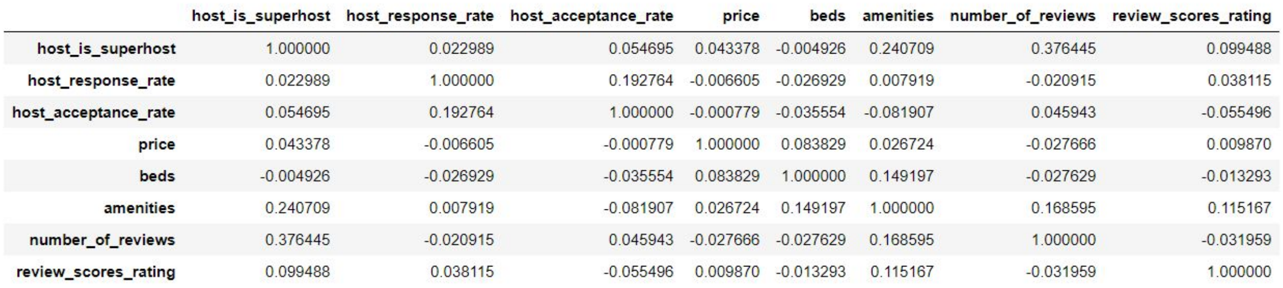
VIII. Appendix

Review Score Conversion Table



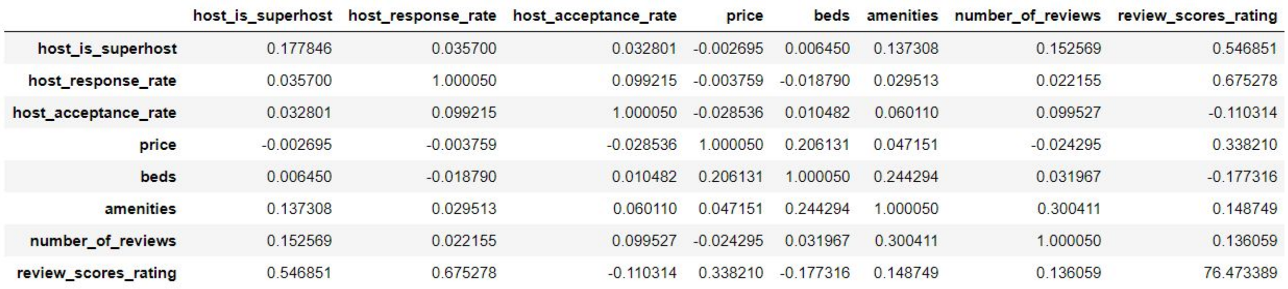
```
Total Review Score       Star Rating
[100]                    --> 5 Star
[80,99]                  --> 4 Star
[60,79]                  --> 3 Star
[40,59]                  --> 2 Star
[20,39]                  --> 1 Star
[0,19]                   --> 0 Star
```

Covariance Matrix Amsterdam



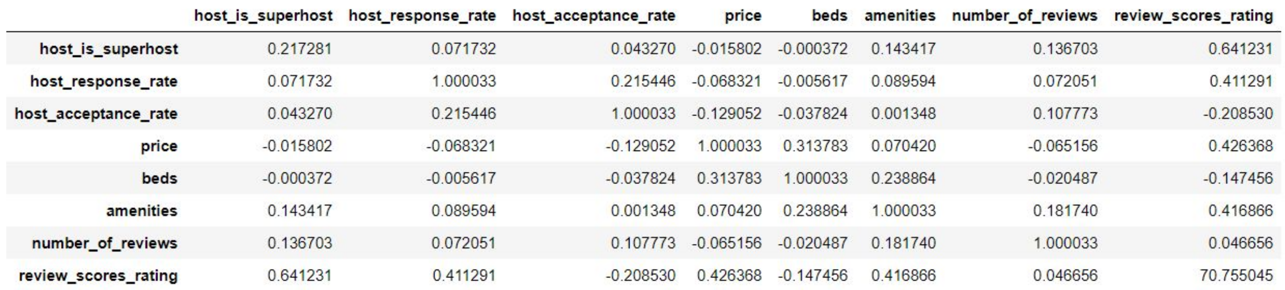
	host_is_superhost	host_response_rate	host_acceptance_rate	price	beds	amenities	number_of_reviews	review_scores_rating
host_is_superhost	1.000000	0.022989	0.054695	0.043378	-0.004926	0.240709	0.376445	0.099488
host_response_rate	0.022989	1.000000	0.192764	-0.006605	-0.026929	0.007919	-0.020915	0.038115
host_acceptance_rate	0.054695	0.192764	1.000000	-0.000779	-0.035554	-0.081907	0.045943	-0.055496
price	0.043378	-0.006605	-0.000779	1.000000	0.083829	0.026724	-0.027666	0.009870
beds	-0.004926	-0.026929	-0.035554	0.083829	1.000000	0.149197	-0.027629	-0.013293
amenities	0.240709	0.007919	-0.081907	0.026724	0.149197	1.000000	0.168595	0.115167
number_of_reviews	0.376445	-0.020915	0.045943	-0.027666	-0.027629	0.168595	1.000000	-0.031959
review_scores_rating	0.099488	0.038115	-0.055496	0.009870	-0.013293	0.115167	-0.031959	1.000000

Correlation Matrix Melbourne:



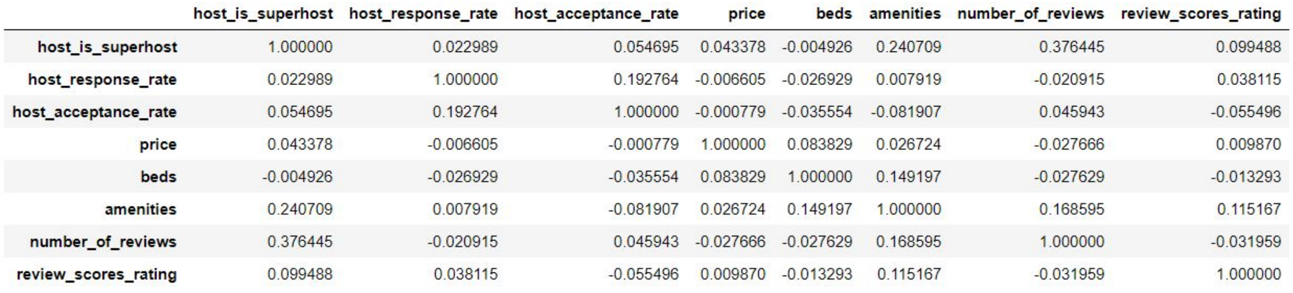
	host_is_superhost	host_response_rate	host_acceptance_rate	price	beds	amenities	number_of_reviews	review_scores_rating
host_is_superhost	0.177846	0.035700	0.032801	-0.002695	0.006450	0.137308	0.152569	0.546851
host_response_rate	0.035700	1.000050	0.099215	-0.003759	-0.018790	0.029513	0.022155	0.675278
host_acceptance_rate	0.032801	0.099215	1.000050	-0.028536	0.010482	0.060110	0.099527	-0.110314
price	-0.002695	-0.003759	-0.028536	1.000050	0.206131	0.047151	-0.024295	0.338210
beds	0.006450	-0.018790	0.010482	0.206131	1.000050	0.244294	0.031967	-0.177316
amenities	0.137308	0.029513	0.060110	0.047151	0.244294	1.000050	0.300411	0.148749
number_of_reviews	0.152569	0.022155	0.099527	-0.024295	0.031967	0.300411	1.000050	0.136059
review_scores_rating	0.546851	0.675278	-0.110314	0.338210	-0.177316	0.148749	0.136059	76.473389

Covariance Matrix Los Angeles



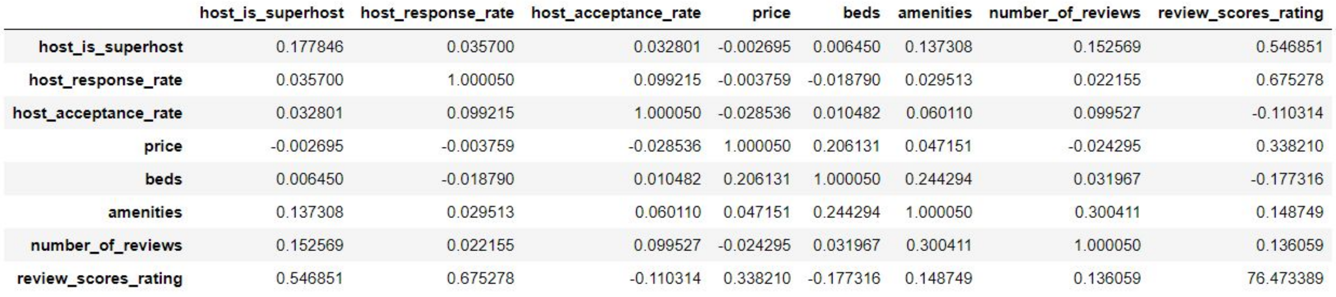
	host_is_superhost	host_response_rate	host_acceptance_rate	price	beds	amenities	number_of_reviews	review_scores_rating
host_is_superhost	0.217281	0.071732	0.043270	-0.015802	-0.000372	0.143417	0.136703	0.641231
host_response_rate	0.071732	1.000033	0.215446	-0.068321	-0.005617	0.089594	0.072051	0.411291
host_acceptance_rate	0.043270	0.215446	1.000033	-0.129052	-0.037824	0.001348	0.107773	-0.208530
price	-0.015802	-0.068321	-0.129052	1.000033	0.313783	0.070420	-0.065156	0.426368
beds	-0.000372	-0.005617	-0.037824	0.313783	1.000033	0.238864	-0.020487	-0.147456
amenities	0.143417	0.089594	0.001348	0.070420	0.238864	1.000033	0.181740	0.416866
number_of_reviews	0.136703	0.072051	0.107773	-0.065156	-0.020487	0.181740	1.000033	0.046656
review_scores_rating	0.641231	0.411291	-0.208530	0.426368	-0.147456	0.416866	0.046656	70.755045

Correlation Matrix Amsterdam



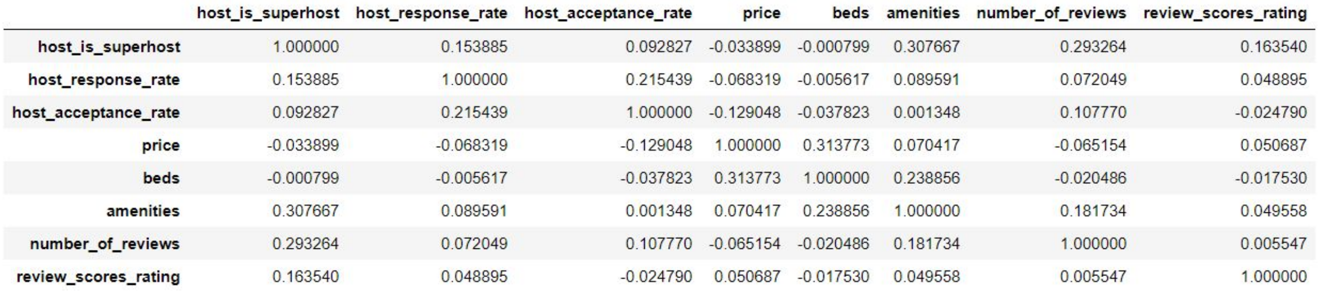
	host_is_superhost	host_response_rate	host_acceptance_rate	price	beds	amenities	number_of_reviews	review_scores_rating
host_is_superhost	1.000000	0.022989	0.054695	0.043378	-0.004926	0.240709	0.376445	0.099488
host_response_rate	0.022989	1.000000	0.192764	-0.006605	-0.026929	0.007919	-0.020915	0.038115
host_acceptance_rate	0.054695	0.192764	1.000000	-0.000779	-0.035554	-0.081907	0.045943	-0.055496
price	0.043378	-0.006605	-0.000779	1.000000	0.083829	0.026724	-0.027666	0.009870
beds	-0.004926	-0.026929	-0.035554	0.083829	1.000000	0.149197	-0.027629	-0.013293
amenities	0.240709	0.007919	-0.081907	0.026724	0.149197	1.000000	0.168595	0.115167
number_of_reviews	0.376445	-0.020915	0.045943	-0.027666	-0.027629	0.168595	1.000000	-0.031959
review_scores_rating	0.099488	0.038115	-0.055496	0.009870	-0.013293	0.115167	-0.031959	1.000000

Correlation Matrix Melbourne



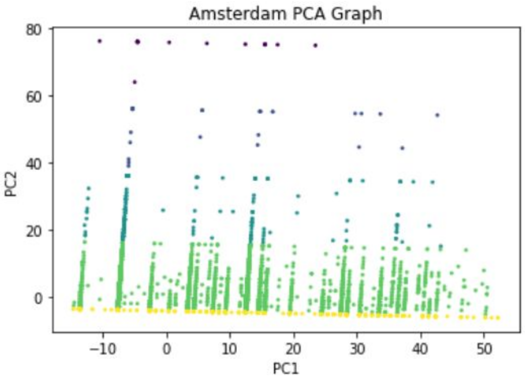
	host_is_superhost	host_response_rate	host_acceptance_rate	price	beds	amenities	number_of_reviews	review_scores_rating
host_is_superhost	0.177846	0.035700	0.032801	-0.002695	0.006450	0.137308	0.152569	0.546851
host_response_rate	0.035700	1.000050	0.099215	-0.003759	-0.018790	0.029513	0.022155	0.675278
host_acceptance_rate	0.032801	0.099215	1.000050	-0.028536	0.010482	0.060110	0.099527	-0.110314
price	-0.002695	-0.003759	-0.028536	1.000050	0.206131	0.047151	-0.024295	0.338210
beds	0.006450	-0.018790	0.010482	0.206131	1.000050	0.244294	0.031967	-0.177316
amenities	0.137308	0.029513	0.060110	0.047151	0.244294	1.000050	0.300411	0.148749
number_of_reviews	0.152569	0.022155	0.099527	-0.024295	0.031967	0.300411	1.000050	0.136059
review_scores_rating	0.546851	0.675278	-0.110314	0.338210	-0.177316	0.148749	0.136059	76.473389

Correlation Matrix Los Angeles

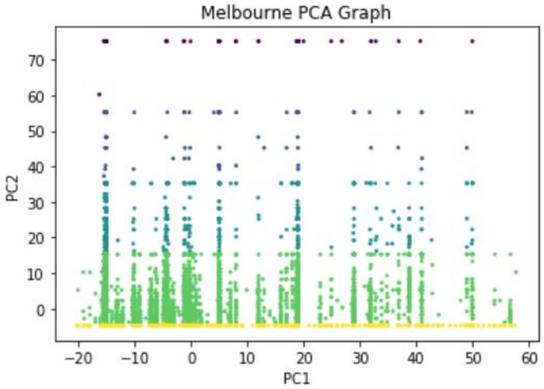


	host_is_superhost	host_response_rate	host_acceptance_rate	price	beds	amenities	number_of_reviews	review_scores_rating
host_is_superhost	1.000000	0.153885	0.092827	-0.033899	-0.000799	0.307667	0.293264	0.163540
host_response_rate	0.153885	1.000000	0.215439	-0.068319	-0.005617	0.089591	0.072049	0.048895
host_acceptance_rate	0.092827	0.215439	1.000000	-0.129048	-0.037823	0.001348	0.107770	-0.024790
price	-0.033899	-0.068319	-0.129048	1.000000	0.313773	0.070417	-0.065154	0.050687
beds	-0.000799	-0.005617	-0.037823	0.313773	1.000000	0.238856	-0.020486	-0.017530
amenities	0.307667	0.089591	0.001348	0.070417	0.238856	1.000000	0.181734	0.049558
number_of_reviews	0.293264	0.072049	0.107770	-0.065154	-0.020486	0.181734	1.000000	0.005547
review_scores_rating	0.163540	0.048895	-0.024790	0.050687	-0.017530	0.049558	0.005547	1.000000

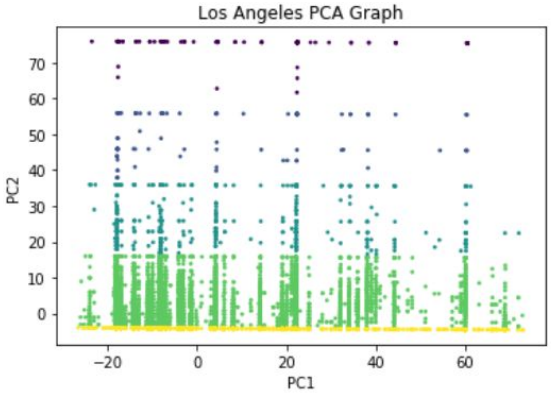
PCA Graph Amsterdam



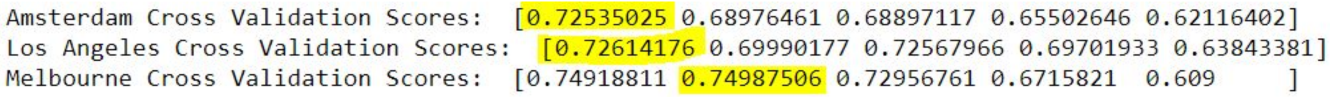
PCA Graph Melbourne



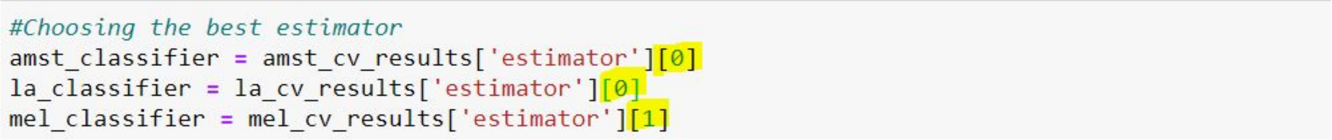
PCA Graph Los Angeles



K-Fold Cross Validation

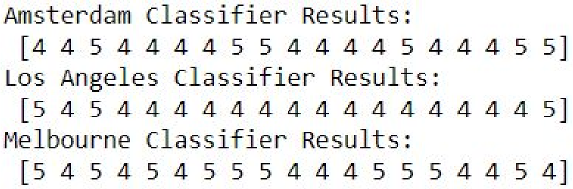


```
Amsterdam Cross Validation Scores:  [0.72535025 0.68976461 0.68897117 0.65502646 0.62116402]
Los Angeles Cross Validation Scores:  [0.72614176 0.69990177 0.72567966 0.69701933 0.63843381]
Melbourne Cross Validation Scores:  [0.74918811 0.74987506 0.72956761 0.6715821  0.609     ]
```



```
#Choosing the best estimator
amst_classifier = amst_cv_results['estimator'][0]
la_classifier = la_cv_results['estimator'][0]
mel_classifier = mel_cv_results['estimator'][1]
```

KNN Results



```
Amsterdam Classifier Results:
 [4 4 5 4 4 4 4 5 5 4 4 4 4 5 4 4 4 5 5]
Los Angeles Classifier Results:
 [5 4 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5]
Melbourne Classifier Results:
 [5 4 5 4 5 4 5 5 5 4 4 4 5 5 5 4 4 5 4]
```