Technical University of Denmark

Introduction to Machine Learning and Data Mining, 02450

Project 3

Unsupervised Learning: Clustering and Density Estimation

Name	Student nr.
Luka Avbreht	**s191963**
Weston Jones	**s191380**
Michael Rupprecht	**s191759**

December 3, 2019

1 Introduction

Our data-set contains information about AirBnB property listings in New York City. We've included a brief introduction to our data-set and explanation of the attributes to help readers better understand the rest of the report.

Data Attribute Name	Explanation	Type
id	Listing unique identifier	Nominal
name	Name of listing	Nominal
host_id	Host unique identifier	Nominal
host_name	Name of host	Nominal
neighborhood_group	Listing borough (Manhattan, Brooklyn, etc.)	Nominal
neighborhood	Listing specific neighborhood	Nominal
latitude	Approximated latitude	Interval
longitude	Approximated longitude	Interval
room_type	Room type (Private or shared room)	Nominal
price	Price per night	Ratio
minimum_nights	Minimum number of nights in a stay	Discrete
number_of_reviews	Number of guests who've left reviews	Discrete
last_review	Date of the last review	Interval
reviews_per_month	Average number of reviews per month	Ratio
calculated_host_listings_count	Number of listings the host has	Discrete
availability_365	Number of says in the year listing available	Discrete

Note that we make frequent references to "boroughs" which refer to the five different subdivisions of New York City: The Bronx, Brooklyn, Manhattan, Queens, and Staten Island.

2 Collaboration

The first section work was split about 60% and 40% between Weston and Luca respectively.
The second section work was split about 20% and 80% between Weston and Luca respectively.
The third section work was split about 50% and 50% by Michael and Luca respectively.

3 Clustering

3.1 Question 1

We decided that a we would try to cluster our data based on neighborhood group or borough. This would give us a small number of predefined classes to work with and since neighborhood group is a directly related to latitude and longitude, we figured we'd easily be able to check our work.

First, our data was prepared by dropping unnecessary columns (ids, names, etc) as well as the last_review (the date format seemed a little hard to work with) and the neighborhood attribute (There were many unique neighborhoods that would have to be handled some way). The room_type attribute was 1-out-of-k encoded and the numerical attributes were normalized by subtracting the mean and dividing by the standard deviation.

The neighborhood group (our "Y" in this problem) was handled by assigning an integer to each borough. Bronx = 0, Brooklyn = 1, Manhattan = 2, Queens = 3, Staten Island = 4.

With the data prepared, we figured we'd start out with a "sanity check" by experimenting with various linkage functions on an X matrix composed of the latitude and longitude attributes. Below are the results of this initial experimentation. Note that the linkage functions are changed but distance is always calculated using euclidean distance and the maximum number of clusters is capped at 5 as there are 5 NYC boroughs.

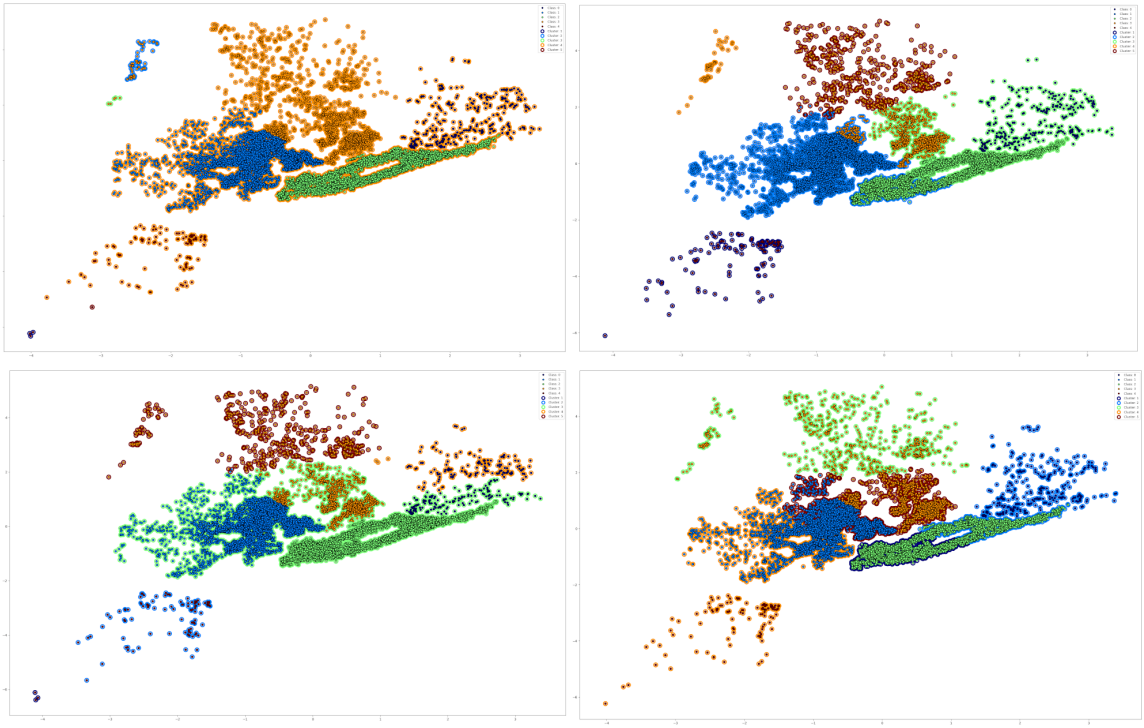

Figure 1: Effect of different linkage functions on clustering. Upper Left: Minimum Distance. Upper Right: Maximum Distance. Lower Left: Average Distance. Lower Right: Ward's Method

The NYC boroughs are organized into clusters based off of geographic separation by rivers and arbitrary divisions. Thus, the maximum distance algorithm, which only cares about maximum distance between groups and tends to favor compact clusters, works best. The minimum distance algorithm appears to treat all the central boroughs as one region and chains them together. Ward's algorithm and average distance perform a bit better, but mishandle the Staten Island properties in the south east.

Based off of this initial experiment, we could verify that our clustering code seemed to be working correctly and observe first hand the effects of different linkage functions on our data. Moving forward with a more interesting clustering problem, we decided we would try to cluster our listings by price. We standardized our numerical data the same as before and used 1-out-of-K encoding for the borough and room type attributes. We looked at a histogram showing the distribution of prices in our data and decided to divide the price attribute into six classes (0 = \$0-99 per night, 1 = \$100-199 per night, 2 = \$200-299 per night, 3 = \$300-399 per night, 4 = \$400-499 per night, and 5 = more than \$500 per night).

We ran our clustering using the maximum euclidean distance parameter based off of our previous work. The first few trials looked similar to the figure shown below - very messy and not much to be deduced.

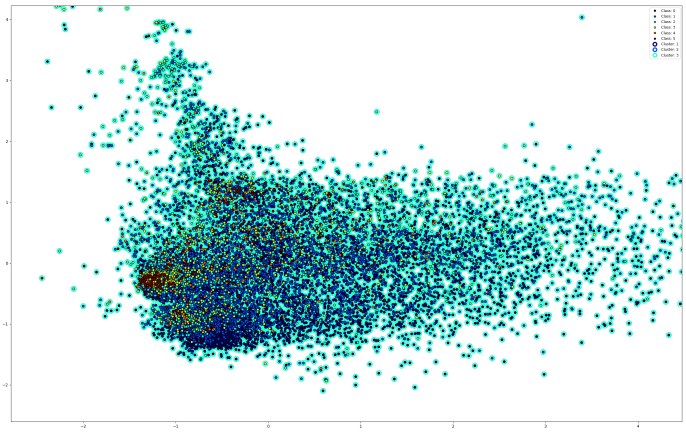

Figure 2: Clustering using all attributes projected into PCA space

Thinking about prices logically, we figured that location and room type would be the biggest determinants. We reran the clustering using latitude, longitude, and the k-encoded columns for room type as the only attributes.

The generated dendrogram shows that the groups are much closer together and that making a max-clusters cut anywhere around 4-6 seems ideal to minimize distance extremes. We went with 6 clusters, because we also have six classes and we're trying to build a model that fits our price classifiers.

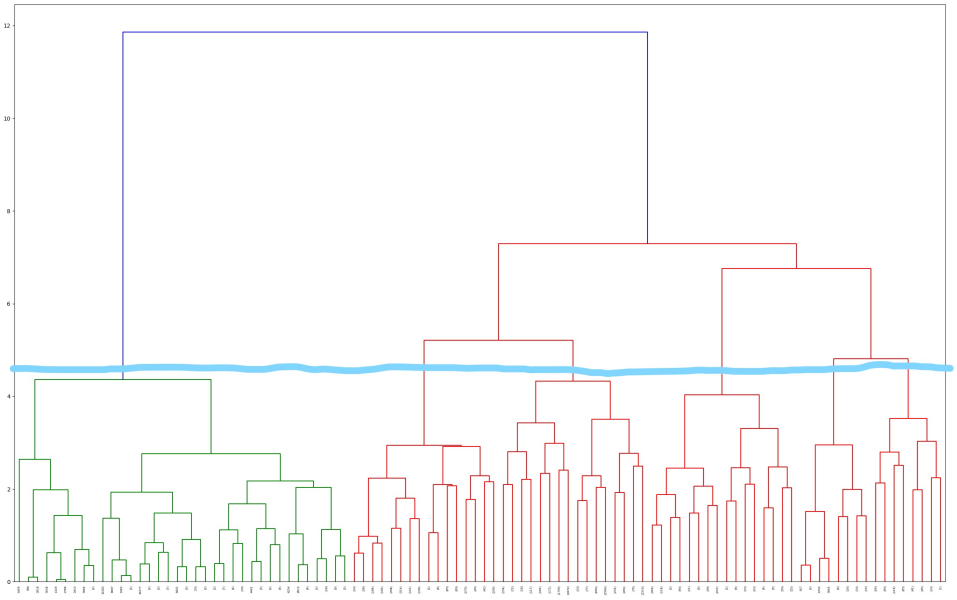

Figure 3: Dendrogram using more selective attributes

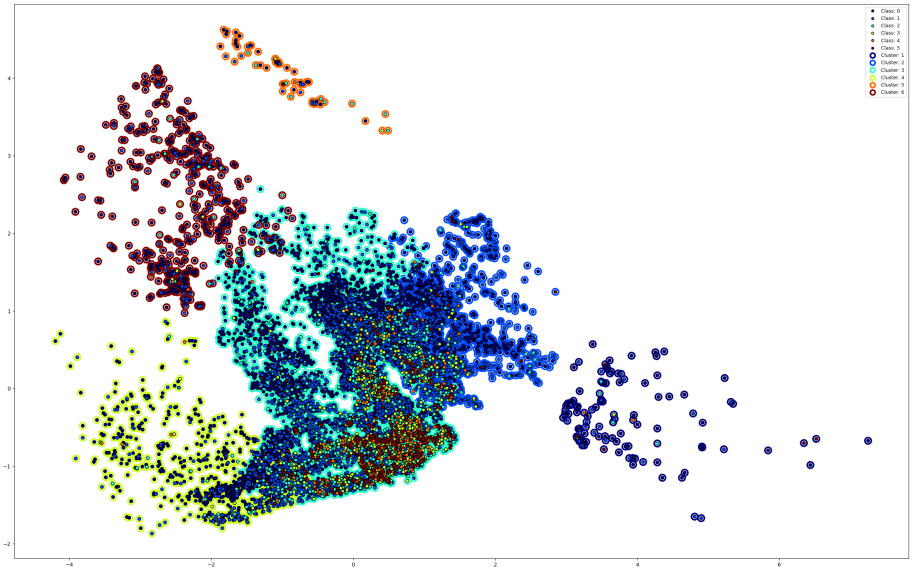

Figure 4: Clustering using more selective attributes projected into PCA space

The clustering appears much more coherent than before. Individual groups can be made out and appear to somewhat match up with our actual results. The model struggles in the more concentrated regions.

3.2 Question 2

Initially, we tried fitting a GMM to all the attributes, but got errors saying that the algorithm could not converge. As before, we limited the attributes to the latitude, longitude, and k-encoded room type columns in order to get more sensible, workable models.

Using cross validation to generate a plot of K versus AIC, BIC, and scores it appears that more components in the mixture are better. Even after the K is increased to 20, the score never seems to improve. In this case, we determined it best to use the "elbow" method to pick 4 as the optimal value of K.

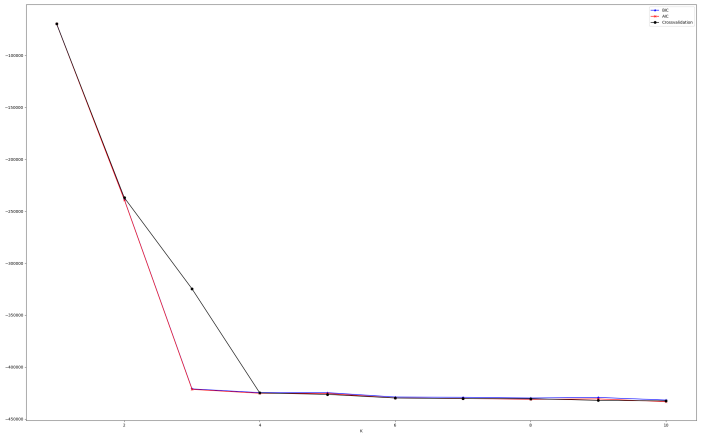

Figure 5: K versus model effectiveness

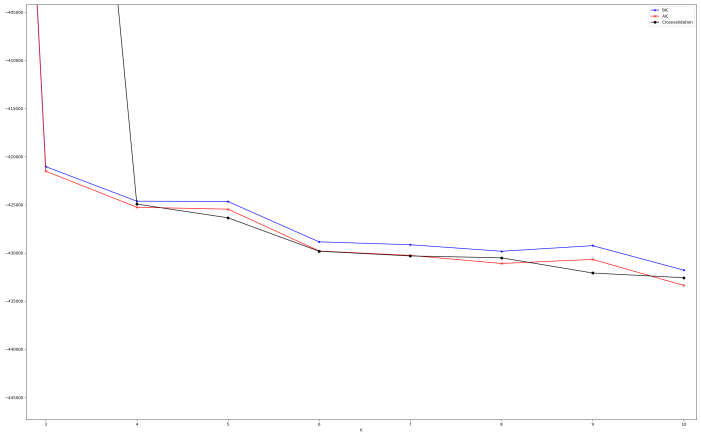

Figure 6: K versus model effectiveness (zoomed in)

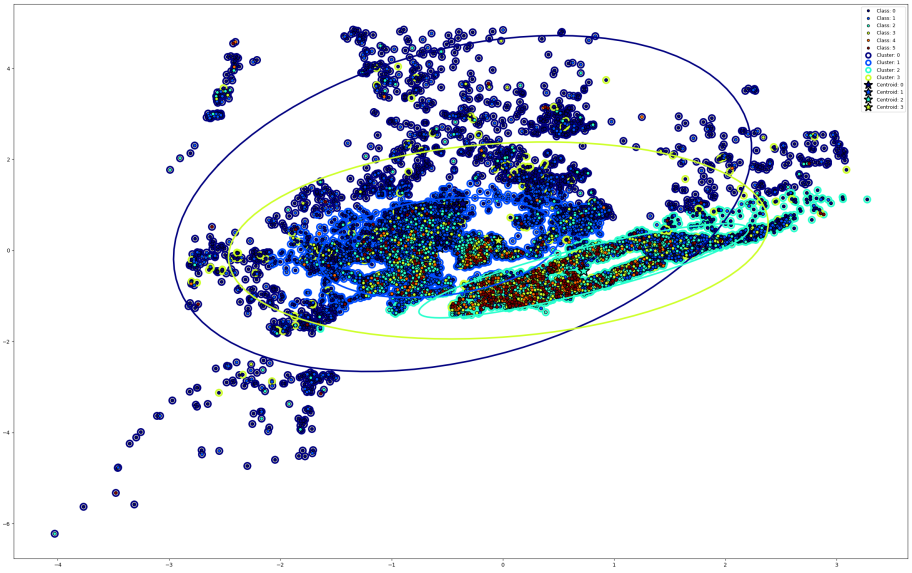

Figure 7: GMM with K=4 on the Latitude, Longitude, and room type attributes. Plotted in (latitude, longitude) space

Each of the four components and their corresponding centeroids seem to correspond to the areas of varying levels of density. Cluster 0 is the most spread out and catches the outliers. Cluster 1 captures both cluster 2 and 3. Cluster 2 encapsulates the stretched lighter blue area and Cluster 3 encapsulates the most dense dark blue area in the center.

3.3 Question 3

We reran the hierarchical clustering setup from before with different values of K alongside the GMM. We then compared to results of the two using the Rand, Jacard, and NMI metrics for cluster validity. The results below show the results of the comparison. Both seem to perform similarly across values of K, though the GMM performs best at K=4 as we predicted earlier.

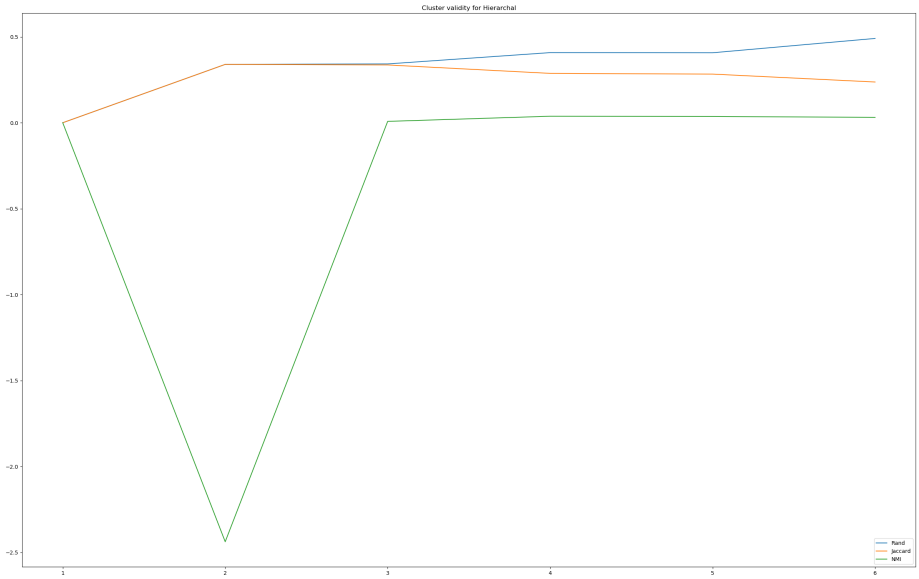

Figure 8: Effectiveness of Hierarchical Clustering with various values of K

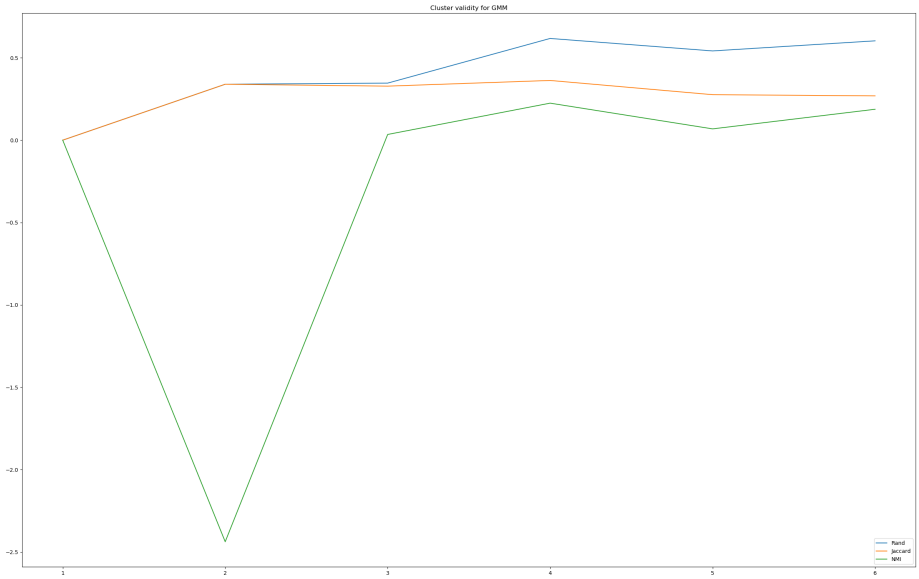

Figure 9: Effectiveness of GMM with various values of K

4 Outlier detection/Anomaly detection:

In this part of the exercise, we tried to detect outliers in listings that are listed as entire apartments. We used one-of-K encoding for neighbourhood parameters in our data, and standardised other data to have a mean of 0 and standard deviation of 1.

4.1 Question 1

Figures 10, 11, 12, and 13 represent the ranking of our data with different metrics. Since representing listing in our data set is not really representative, we did some other hopefully more representative measures.

First we checked how many of the first 200 listings were from each of the districts. We did so because the listings in Manhattan are to be expected to be the most extreme. The results of this can be seen in table 1. The other thing we checked was the price of the first 10 outliers. The results of that can be seen in table 2.

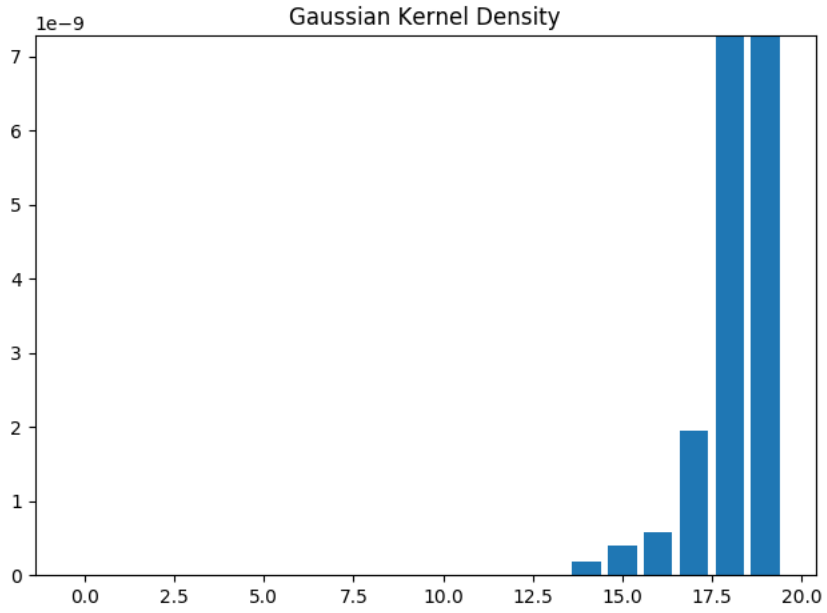

Figure 10: Gaussian Kernel density Outlier score

Gaussian Kernel Density	54	20	58	9	59
KNN Density	17	45	94	27	17
KNN Relative Density	0	62	93	45	0
5'th Neighbourhood	0	109	78	13	0

Table 1: District distribution within first 200 Outliers

4.2 Question 2

Looking at the KNN density outlier score chart, it looks like many of our clusters have relatively high densities. However, our average relative density outlier score chart, based on looks, appears to

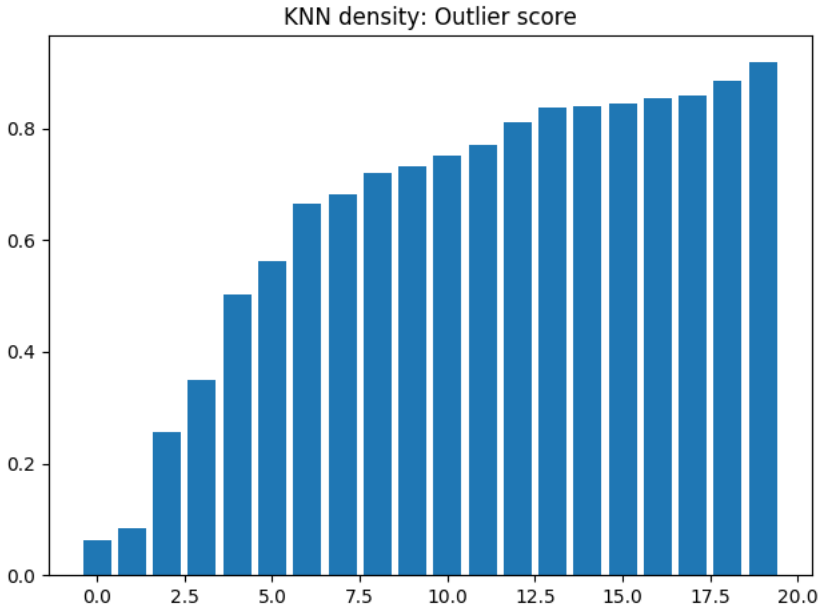

Figure 11: KNN density Outlier score

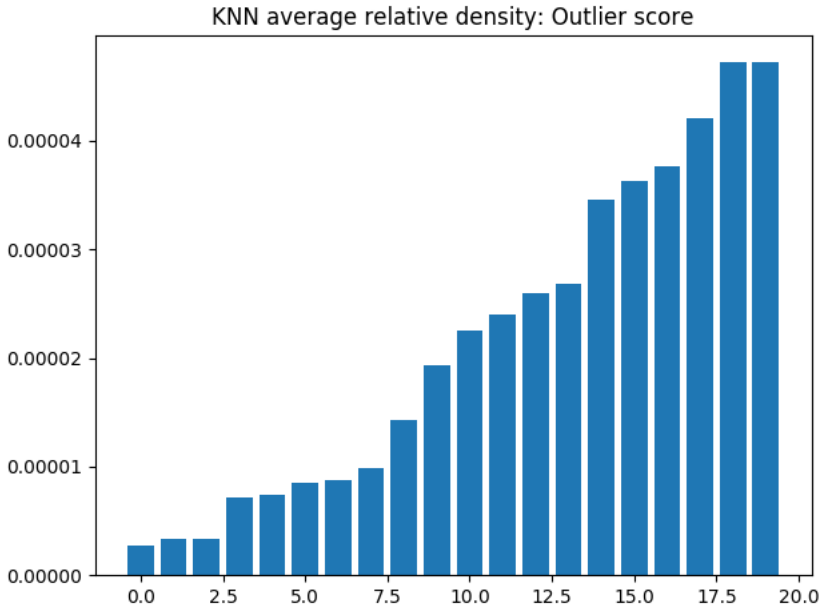

Figure 12: average relative density Outlier score

have some "groups" of densities, where various clusters have very similar densities, where some are quite low, perhaps suggesting a relatively high number of outliers.

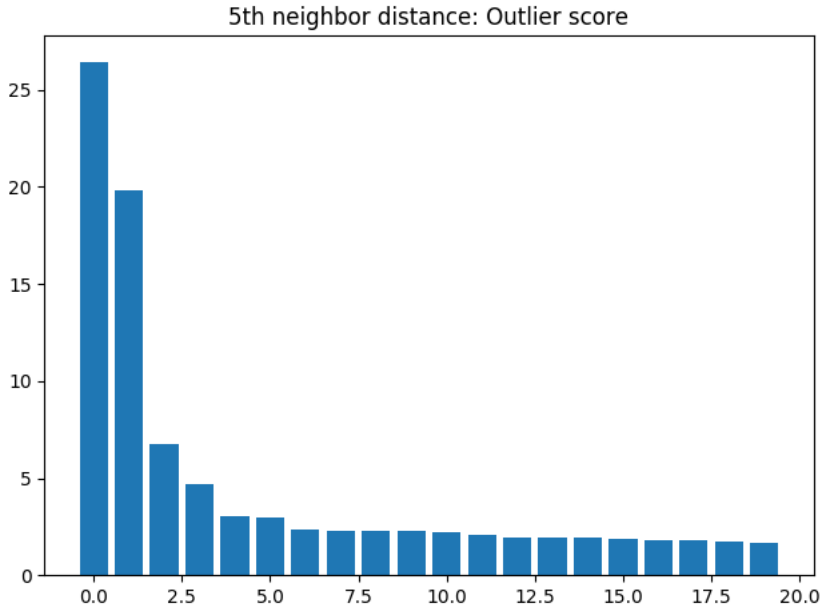

Figure 13: 5'th Neighbourhood distance Outlier score

1	195	195	200	50
2	85	300	39	40
3	120	50	95	125
4	90	120	350	150
5	350	85	130	199
6	300	90	40	89
7	50	95	85	178
8	196	45	59	109
9	675	675	480	100
10	90	196	150	115

Table 2: First ten elements of each Outlier measure

5 Association Mining

As requested we also run the Apriori algorithm on our data, in order to find the connections and associations between our data. To do so, we again had to make some data modifications. We used one of K encoding for room type and neighbourhood values. For other, continuous data we binarized it using the provided binarize2 function. Doing so we split each continuous value into two columns, one for the values that are above average, and the other for those below it.

5.1 Question 1

```
{availability 0th-50th percentile} -> {host listing count 0th-50th percentile}  (supp: 0.406, conf: 0.811)
{price 50th-100th percentile} -> {host listing count 0th-50th percentile}  (supp: 0.355, conf: 0.711)
{rev per month 0th-50th percentile} -> {host listing count 0th-50th percentile}  (supp: 0.360, conf: 0.720)
{review number 0th-50th percentile} -> {host listing count 0th-50th percentile}  (supp: 0.355, conf: 0.705)
{rev per month 0th-50th percentile} -> {review number 0th-50th percentile}  (supp: 0.409, conf: 0.818)
```

```
{review number 0th-50th percentile} -> {rev per month 0th-50th percentile} (supp: 0.409, conf: 0.814)
{rev per month 50th-100th percentile} -> {review number 50th-100th percentile} (supp: 0.406, conf: 0.813)
{review number 50th-100th percentile} -> {rev per month 50th-100th percentile} (supp: 0.406, conf: 0.817)
```

We chose to include the rules that had a confidence of over .7, as this included roughly the top quarter of our results.

5.2 Question 2

Our highest-confidence rules were simply between two very correlated attributes: total number of reviews and number of reviews per month. These were bidirectional and had very similar confidence numbers. These highest-confidence rules were exclusively between the lower half and lower half as well as between the upper half and upper half. This is likely just an artifact of the way we organized our data.

Our other rules with high confidence numbers were unidirectional, relating availability, number of reviews, and reviews per month in the lower half to the host listing in the lower half. The latter two relations likely stem from availability, as greater availability would suggest more reviews per month (and thus more reviews). It suggests that some hosts with more properties are more likely to make them available for longer (perhaps because they don't live in all their properties at once), while the fact that it is unidirectional means there exist hosts that nonetheless choose to take their properties off the market for longer. The confidence for the reverse direction was .618 – markedly lower, and well in line with the majority of our rules.

Lastly, we saw a rule that related price in the upper half to host listing count in the lower half. I suspect this arises out of chance, as I otherwise don't know how to interpret this. They seem to be unrelated.

I would also like to note that many of our confidence numbers ended up being approximately double our support numbers. I don't know what to make of this.