CSCI 6612 - Visual Analytics
Final Report
Airbnb Recommender System
Fall 2019

Group HTTP 404

Soheil Latifi* Asal Jalilvand† Kewei Ma‡ Mir Erfan Gheibi§

1 December 2019

Abstract

This project aims to construct a recommender system for Airbnb listings. As the course name dictates, this project consists of two significant parts, machine learning, and visualization. On the machine learning side, core implementation is a K-means clustering algorithm to cluster listings. On the other part, some of the key features are maps, listing details, and some analysis to ease the choice among the options. Although establishing a machine learning system to predict some of the features in the context of Airbnb is not the goal of this project, we have reached some good results in terms of accuracy. Using the recommender system, the search space for the users will be narrowed down to a select set of options to choose from among them.

*Soheil.Latifi@dal.ca
†Asal.Jalilvand@dal.ca
‡Kewei.Ma@dal.ca
§egheibi@dal.ca

Contents

List of Tables

List of Figures

1 Introduction

With the growth of the sharing economy, some companies have been established to facilitate this concept. The sharing economy is an economic model that individuals are the providers and customers, instead of the traditional models that firms are providing. Uber for renting cars and Airbnb for sharing rooms are two well-known examples that help people and get commissions. In this project, we want to be a part of this revolutionary idea utilizing visual analytics.

Problem definition, Dataset, Visualization, Implementation, Applied Comments, and a brief conclusion at the end are the building blocks of this report on our group's project.

2 Problem Definition

Airbnb has 2 types of users; property owners and rental seekers. For the owners, Airbnb offers a smart price recommendation system that takes various factors into account, and suggests the best price for the owners to maximize their profit. However, for the rental seekers, it offers minimum insight to choose wisely from the options, which needs significant effort from the users' side to get relevant results. From Airbnb's point of view, maybe there is no need to construct such a system because the company gets its income from commission; therefore, there is no need to minimize price or maximize demand in one place. Also, the users can not get a pack of information such as travel time to certain areas as well as accommodations in the same place. We decided to tackle this problem by utilizing machine learning and visualization techniques to build a recommender system.

3 Dataset

In this section, the base dataset and the operations we did on it will be discussed. For the base dataset, we used "Melbourne Airbnb Open Data" from "Kaggle", an "Inside Airbnb" originated dataset. The dataset was compiled on the 7th. Dec. of 2018, and contains numerous features ranging from amenities to reviews of listings in Melbourne, Australia.

3.1 Pre-Processing

3.1.1 Cleaning the data

As shown in our preprocessing Jupyter Notebook, on the first step we visualized the data using "Pandas Profiling" package. This package gave us deep insight on the data. For example, the data distribution, number of missing data, most frequent values etc. After analyzing the data, it was clear that some of the features were either unrelated ("Host ID" etc.) to our task, or could have been used, but in cost of adding extra effort ("host about", "Listing summery", etc.)

3.1.2 Dealing with missing values

In this part, "number of bedrooms" and "number of bathrooms" were imputed using the most associated values method. For example, listings with 2 bedrooms on average have 1 bathroom so

for each row of our table with missing values on bathroom column, number of bedrooms was used to impute the missing value of bathroom feature.

Same method was used for imputing missing beds as well.

The Neighborhood feature suffered from severe data loss (76% missing values). It was necessary in the beginning for us to predict these values since we wanted to find average distance from a neighborhood to the other ones instead of using exact locations. Although, later instead of finding distances of listings to the other neighborhoods, we added average time spent using public transportation to travel to tourists site which seems to be more practical and bears more importance than the later option.

Logically, finding neighborhood is related to longitude and latitude of the listing. Therefore, these features were used to fill this column. First logistic regression was trained on the data, but the F1 measure was relatively low (58 percent), and so was the accuracy (65 percent). Thus, changing the model was taken into consideration. But when the data was visualized, it was noticed that there is not enough data across the map to achieve a higher accuracy.

KNN model looks promising since the intuition is to use the nearest neighbor to predict the target value using Euclidian distance. Therefore, it seemed to be a prominent candidate for us. This model got 78 percent accuracy and 65 percent f1 measure. Even though this model achieved the best result, we used Melbourne neighborhood polygons in Neighborhood.geojson file to impute the missing values because this way we were sure all the data is imputed without any error.

Since we used K-means clustering model we had to deal with categorical data, such as amenities and host verification, that must be converted to numeric features. But a major problem rises this way, K-means algorithm is not suitable for categorical features. So instead, label encoding was used for every categorical feature that its' instances could be ranked or compared with each other. For amenities, each amenity was set to be filtered by user's request. But for clustering, the count of each listing's amenities was used.

3.1.3 Filtering the data

After filters are chosen by the user, filter function applies them on the data and sends back the remaining listings IDs.

3.2 Clustering

A function was designed to do the clustering. In this function first the features are multiplied by the weights assigned to each feature. Needless to say, if the weight for a particular feature is equal to zero, that feature would have no effect on the clustering algorithm. On the other hand, the higher the weight, the more it effects the model. It is also important to mention that this system allows the user to find any kind of pattern she is interested in. Before weights are applied, all categorical and geographical features are dropped so the clustering remains robust and useful. After the data is clustered, the average value of 5 different features, namely average transportation time to touristic attractions, review score, number of amenities and price is calculated for each cluster and shown to the user to provide her a better point of view on the clusters.

3.3 Sentiment Analysis

Reviews were grouped by listings, then we used NLTK's vader to get sentiment analysis of each review. A new feature is added to our dataset showing the polarity average of all the reviews.

3.4 Adding Features

We were initially planing to find nearby places such as restaurants and supermarkets to each listing and give a point for accessibility of that listing to such places. However, when it came to data collection we realized this is a very cumbersome task and also there is no API providing such feature for free. Then we decided to collect data about Melbourne main neighborhoods and their top K restaurants, cafes. etc. Thus, all the data imputation for lost neighborhoods was done. But after the data imputation, we found another API that provided temporal distance between two places. In turn, we came up with a more relaxed, less cumbersome idea of calculating the distance of listings to the most important parts of the city, which for a traveler, is usually the tourist attractions.

The sharing economy is entangled with tourism. Therefore, in an airbnb recommender system, there should be a smart touch of tourism facilities. We decided to choose various exciting tourist attractions in Melbourne to give the user an extra insight into how where a specific listing resides utilizing its distance to the tourism sites.

For extracting the attractions, we used the PlanetWare website and its comprehensive article regarding the tourism sites , which led us to 12 fabulous locations. In the next step, we had to use these coordinations to calculate their distances from each listing.

The distance can be defined in several ways. We used the travel time of using public transportation systems. The travel time consists of the walking time to reach the correspondent stations, and the time between the station using the public transportation facility. To get this feature, we used HERE Technologies (former Nokia HERE Maps), which gives a rich API, and a limited amount of transactions for free. Using it, we were able to reach a precise set of distances to add in our dataset.

To give more degrees of freedom to the users, we decided to add other distance-based features, quite similar to the distances mentioned above. Melbourn can be divided into 30 suburb areas called neighborhoods. We also added each listing's distance to the rest of the regions. The latter features have less valuable information compared to the first set, due to the nature of neighborhoods and the geometry of them.

4 Visualization

Our main visualizations are as follows:

4.1 Melbourne Map

This map consists of two base layers:

1. Streets: Basic roadmap of Melbourne (fig. 1)

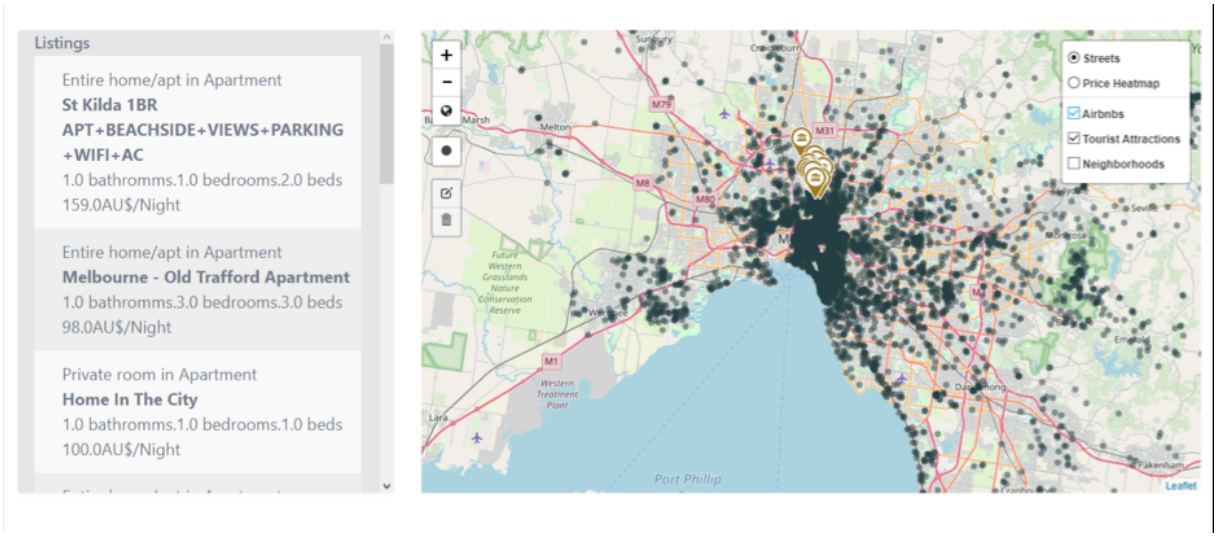

Figure 1: Melbourne map, "Streets" base layer, "Airbnb" and "Tourist Attractions" overlays selected

2. Price Heatmap: Heatmap of normalized price of Airbnbs across Melbourne (fig. 2)

And three overlays:

1. Airbnbs: Airbnb listings across Melbourne (fig. 1)
2. Tourist Attractions: Location of the 12 tourist attractions used in bonus functionality (Section 3.4) (fig. 3)
3. Neighbourhoods: Polygons of the main neighborhoods in Melbourne (fig. 4)

The street view and the Airbnbs evidently are for displaying the location and distribution of the Airbnbs in Melbourne. Since the map we used inherently has icons for displaying cafes, restaurants, bus stops etc. the user can also evaluate the places near a listing she's looking at. The visual icons of tourist attractions makes them stand out in the map which is more tangible for the user.

The price heatmap can help user identify the neighborhoods with prices that fall within her budget. For most users price is the most important factor for choosing an Airbnb, after location [1].

The neighborhood polygons help user see whether a listing's location falls within a particular neighborhood or not, or when they choose the price heatmap, they can get a grasp of the price range in each neighborhood.

There is also an input form for user to explore listings have her preferences such as price range, number of bedrooms, availability of WiFi etc. The heatmap and Airbnb markers on map change accordingly with each filtering (fig. 5). There is also a tool for filtering areas on map, for example in fig. 6, user can filter listings in downtown, near beach. Each selection, displays a popup which gives a brief description about distribution of listings inside and outside the circle.

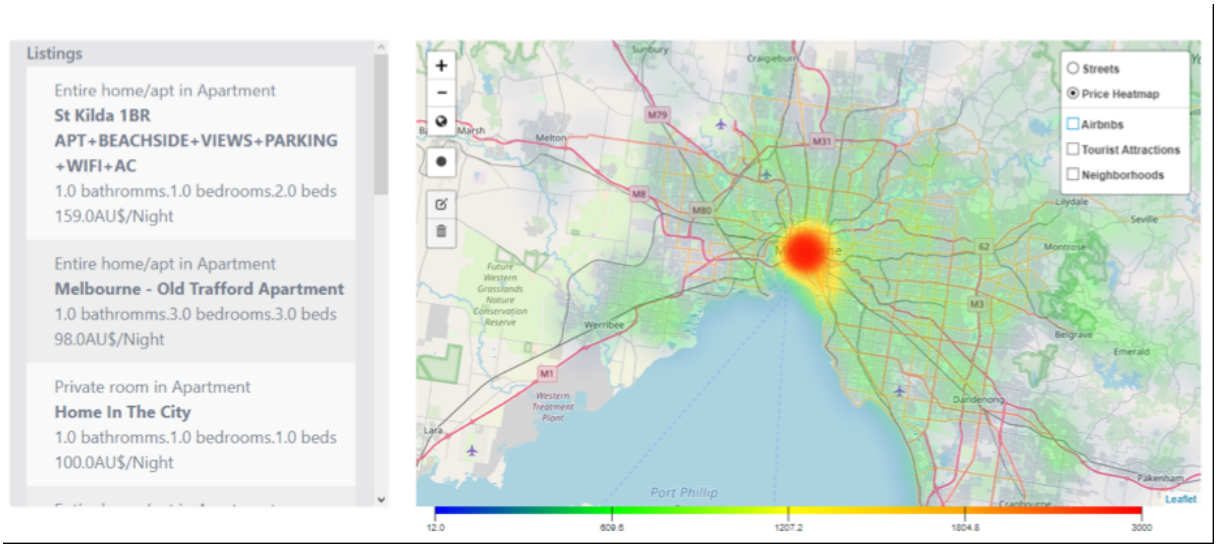

Figure 2: Price heatmap base layer selected

4.2 Reviews Wordcloud and Sentiment Analysis

Reviews of past guests helps users know what the service quality of the host might be like. According to studies [1], it is one of the most important factors for choosing a listing. It is often recommended to read through the reviews to see how hospital the host is, whether the house is truly the way it is described and such.

Since going through all the reviews might be time consuming and tiring for the user, we included two visualizations which summarize the information the user needs (fig. 7). The more frequent a word is in the reviews of a particular listing, the bolder and bigger it is in the wordcloud. Since the word frequency alone is not sufficient for understanding how positive or negative the reviews are, a gauge chart for the overall sentiment of the reviews is also provided. The colors of the words in wordcloud correspond to the positive/negative weight they have in the sentiment analysis. As an example, in (fig. 8), we can see two listings, one with generally positive reviews and the other one with negative ones. In the first one, we see words like "friendly" and "great" are bold and green which means that these words have been repeated a lot, and in a good sense. However the second one we see words like "terrible" and "desceptionate" bold and orange-ish which means past guests have not found this place pleasant enough.

4.3 Clustering and feature grouped bar chart

Our most important functionality is clustering the Airbnb listings. We thought that the most tangible way of showing the clustering result to the user is displaying each listing with a colored marker representing the cluster it belongs to. Therefore, there is no separate visualization for clustering and simply a color is assigned to listing markers per cluster (fig. 9). There are checkboxes with background colors which serve as both a legend for the clusters and also letting the user only view the clusters she is interested in (fig. 10).

It is not enough to simply show a group of clusters on the map. The user must know what features

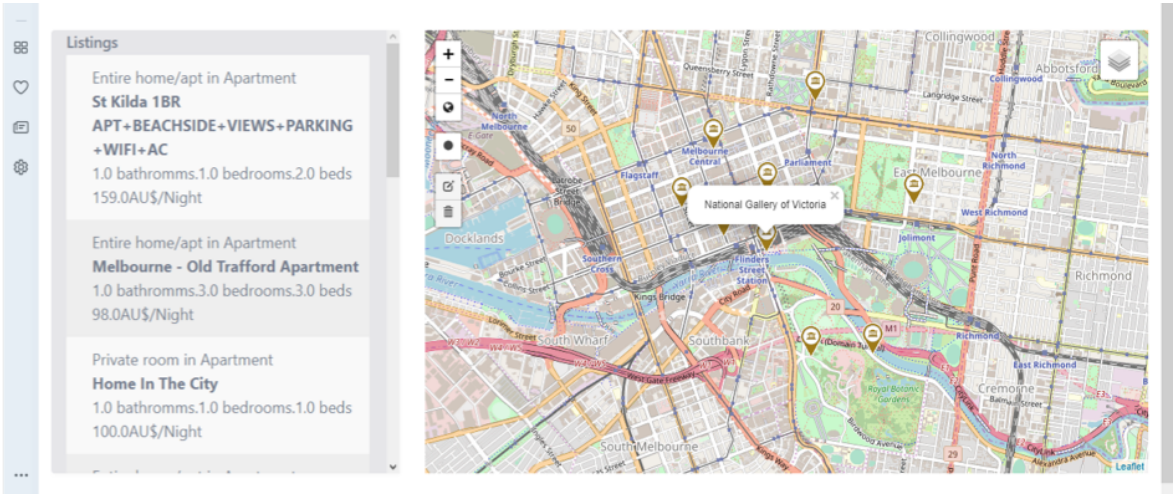

Figure 3: Tourist attractions overlay

the listings in each cluster have in common. We've used around 20 features in clustering, we chose five of them that we thought might be the most important and helpful for the user. User can see the average of values of these features in each cluster and conclude, for example, the average price in once cluster is low while the average distance to tourist spots is high so it is not an optimal cluster. Yet another cluster has a little higher price average but is much closer to touristic sights, so it might be a better choice for the user(fig. 11).

Another functionality related to clustering is providing the ability of biasing the model by user (fig. 12). Using sliders, user can give weights to features from 0 to 10 (Section 3.2).

5 Implementation

5.1 Visualization

The main tools and libraries used in the project's visualization are as follows:

- D3: mainly used for manipulating HTML elements based on data, drawing gauge chart for sentiment analysis and drawing wordcloud (with a ready to use library, d3-cloud)
- Leaflet: interactive map
- Leaflet.heat: Leaflet heatmap plugin
- Leaflet-zoom-min: Leaflet plugin for zooming toolbox
- Leaflet.draw: Leaflet drawing plugin used for drawing circle to filter areas on map
- simplePagination: simple pagination for Airbnbs list
- Plotly: a graphing library used for plotting the grouped bar chart for clustering features
- Bootstrap: a CSS framework used for making the UI more responsive and user-friendly

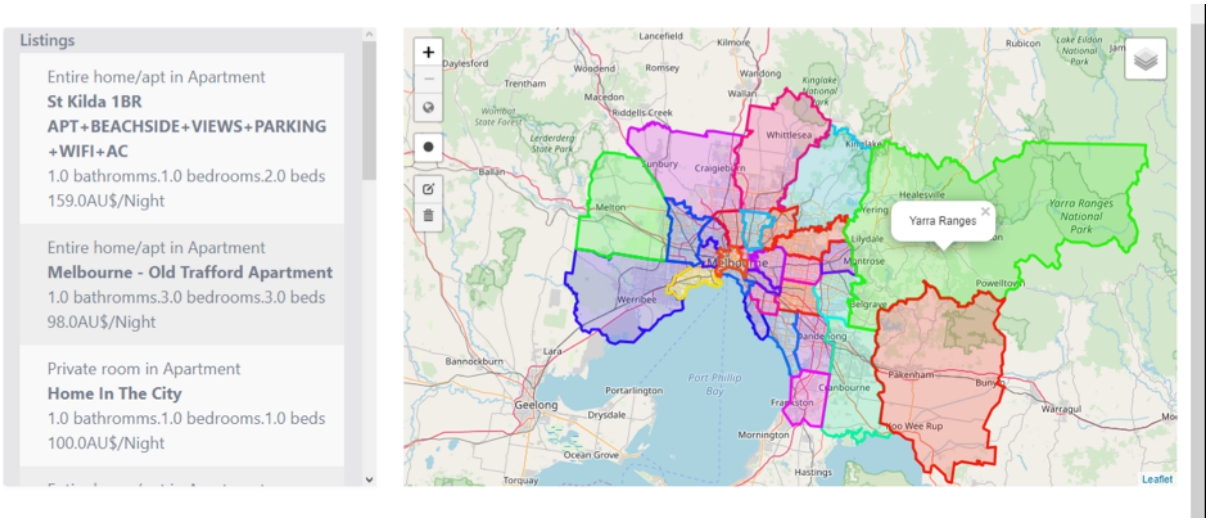

Figure 4: Neighborhood polygons

(a) Heatmap with Airbnb overlay

(b) Heatmap without Airbnb overlay

Figure 5: Custom filter on Airbnbs

5.2 Python Packages

Python packages used For the machine learning and system's backend.

- Pandas: used for loading and handling the data.
- Sklearn: Provided utilized models such as KNN logistic regression and K-means.
- NLTK (Vader lexicon): A simple package that was used in reviews sentiment analysis.
- Pandas–profiling: visualisation tool used for getting a better point of view on the data.
- Json: Used for converting backend results to json.
- Flask: micro web framework for connecting the visualizations and the clustering model

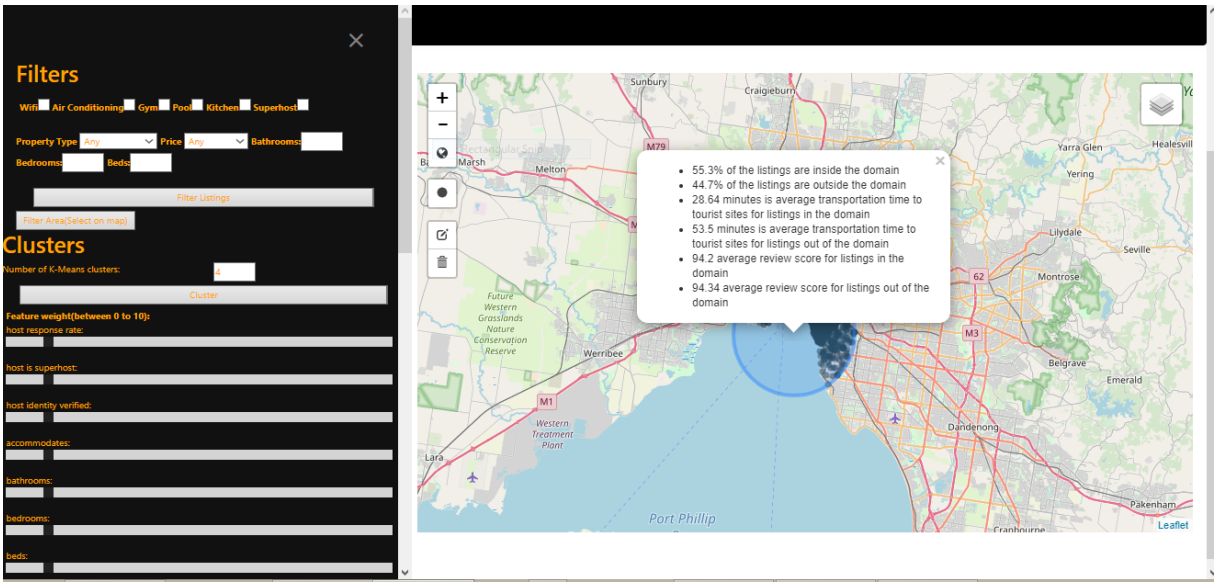

Figure 6: Filtering area on map with statistical information about listings inside and outside the selection

6 Applied Comments

Our group, fortunately, got three sessions of constructive comments. The sessions were on the proposal day, the initial presentation for the professor and TAs, and the final demonstration of the project. In this section, we will go through the list of comments and the solutions we used to compensate for the shortages in an abridged manner.

- The proposal day:
  - Use another distance metric instead of a physical waking distance because a rapid public transportation service is a game-changer: We used public transportation travel time in our project.
- Early presentation:
  - Enlarge the map, decrease the blindness: This comment consumed a considerable amount of person-hour workload. We used the Bootstrap library to bring the outline of our UI/UX to a level of acceptable responsiveness to display size. Also, we used an enlarged version of our map. Furthermore, we packed the sentiment analysis and word cloud in the tooltip pop up of each listing to push the limits of depicting a vast amount of information on a regular-sized display to some extent. Additionally, the sparse bar charts revolutionized into a grouped bar chart.
- Final demonstration:
  - Using the same coloring scale used in sentiment analysis gauge chart, instead of a random coloring scheme. The color of each word reflects its positivity/negativity level in the sentiment analysis.

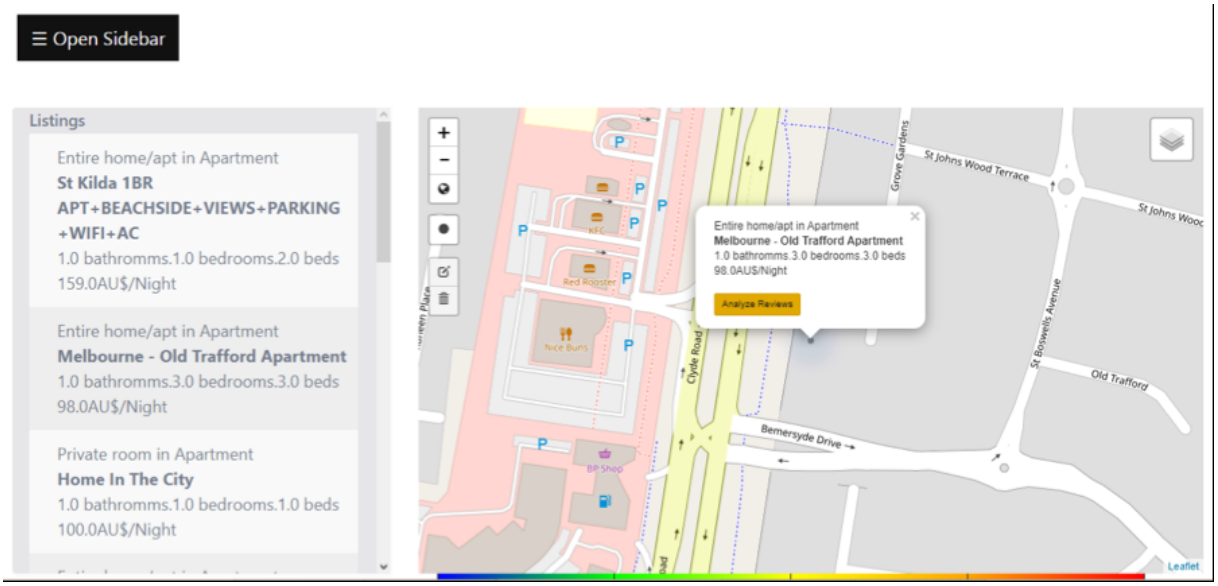

Figure 7: Review analysis option when selecting a listing from the list

(a) Positive (b) Negative

Figure 8: Two listings with positive and negative reviews

– Statistical information of listings inside and outside the circle when filtering an area on map: We added information about the density, average transportation time and average review score of the listings inside and outside the selection.

7 Conclusions, and Future Work

We managed to deliver minimum, expected and bonus functionalities in our project proposal. The user can find patterns in Airbnb listings by playing around with the clustering model and its feature weights. Also the need for reading through all the reviews is reduced to a great extend thanks to the wordcloud and gauge chart visualizations which are simple and easy to understand.

There are several things that we could have done if we had more time and financial resources to access licensed APIs.

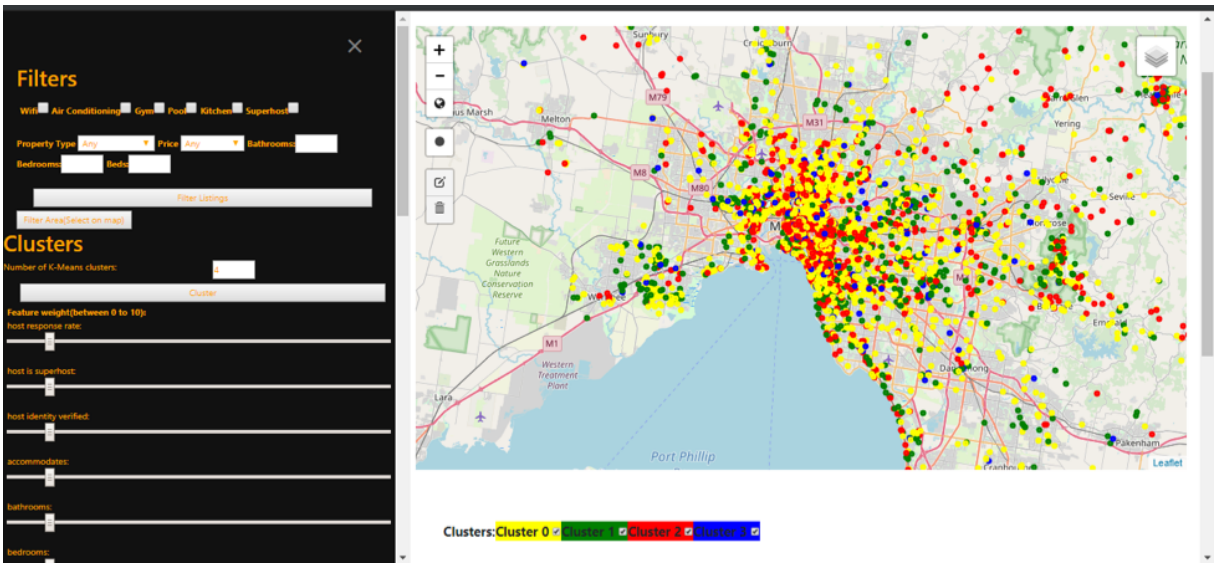

Figure 9: Clustering result on map

- Having had access to Google API and enough time for data collection and its following computations, we could have implemented our original idea in the proposal where we wanted to include the accessibility of restaurants, supermarkets etc. in the clustering.
- The pricing of the Airbnb listings depends on season and demand. Including temporal data to find periodic trends is another interesting functionality we could add.
- If we had access to Airbnb real-time data through an open-access API, we could have delivered a real application useful for travelers all around the world.

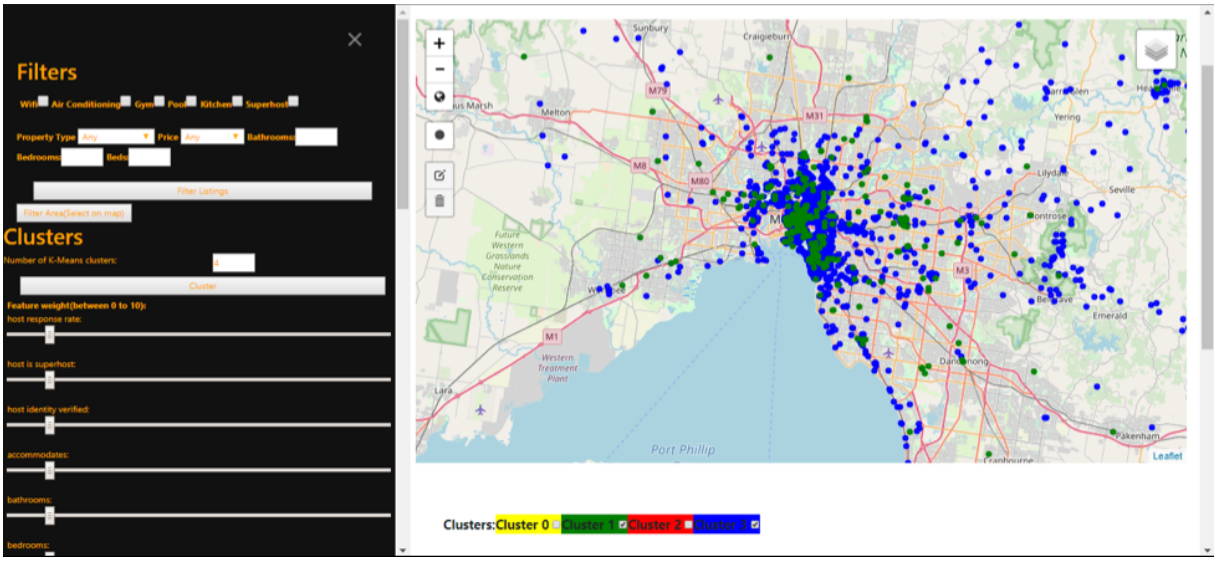

Figure 10: Clustering result on map, selecting only two clusters to display

Figure 11: Grouped bar chart for five of the features used in clustering

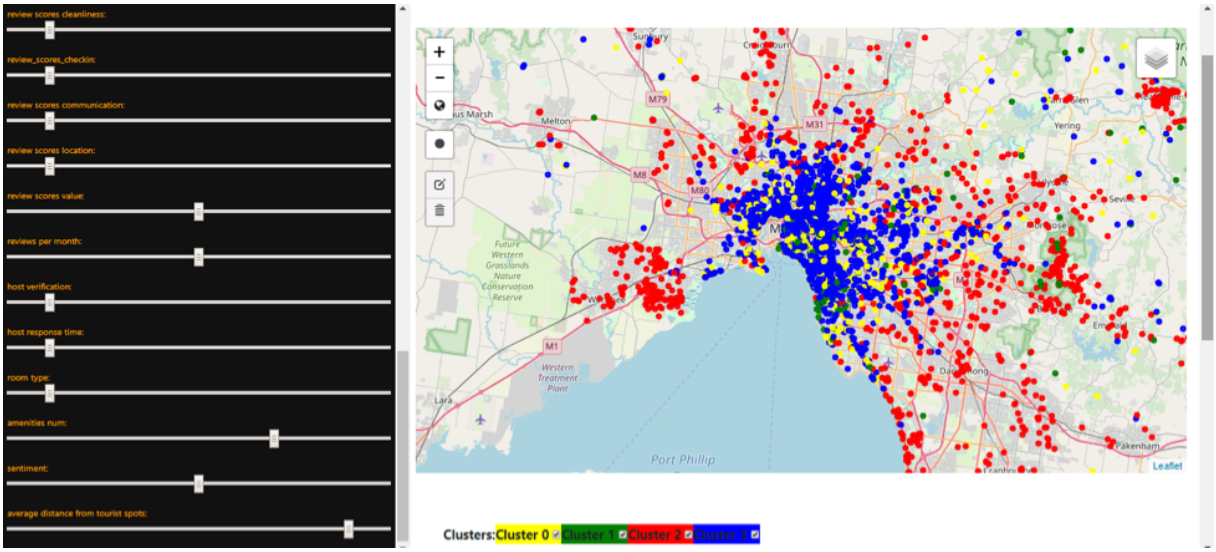

Figure 12: Sliders for biasing the clustering model

References

[1] Daniel Guttentag. Progress on airbnb: a literature review. *Journal of Hospitality and Tourism Technology*, 2019.