Coursera - Applied Data Science

Capstone Project

Best Tourist Spots in Singapore

An exercise in Machine Learning using Clustering Algorithms

By – Ashwin Mohananath Syamala

Date: January 2020



# Introduction:

Singapore, an island city-state off southern Malaysia, is a global financial center with a tropical climate and multicultural population. Singapore is much more than the sum of its numerous attractions. It’s constantly evolving, reinventing, and reimagining itself, with people who are passionate about creating new possibilities.

It’s where foodies, explorers, collectors, action seekers, culture shapers, and socialisers meet―and new experiences are created every day.

There’s something for everyone―world-class theme parks, iconic waterfront attractions, the world’s first night safari, hidden gems to explore in charming neighbourhoods, lush greenery. Discover how a City in the Garden is rooted in history and heritage. Stroll along the colourful shophouses in Haji Lane. Or trek off the beaten path and walk atop trees at MacRitchie Reservoir. If you’re passionate about discovering places, here’s where nature lovers, history buffs, backpackers, city trippers, trekkers, discoverers, sightseers―meet.

# Business Problem:

Given the diverse range of activities, historical sites and places to visit, a traveller could easily be overwhelmed by the huge number of choices that are available and will make it very hard for them to organise an itinerary for their travel. Apart from the actual venues, often the number of places travellers could stay is also a key metric that is considered when making travel plans.

K-Means Clustering, a Machine Learning Algorithm, is implemented to analyse the various neighbourhoods in Singapore and group them into various clusters based on the categories of venues in those neighbourhoods and the number of available Airbnb rental properties that are close to the venues in those categories.

The objective of this specific use of K-means clustering algorithm is to identify places of interest from all the neighbourhoods of Singapore and group them into clusters and visualise the results in a map for better understanding.

The main venue categories that have been included in this analysis are purely tourist destinations such as monuments, historic places, adventure areas etc. Place such as restaurants and other categories have been excluded from the analysis.

# Target Audience:

Travel agents and tour organisers can make use of a model such as the one described above to create itineraries and tour packages for potential travellers.

# About the Data:

As the task involves identifying venues and their categories in each neighbourhood of Singapore, as well as the number of rental properties in those neighbourhoods, the following data have been obtained from the sources listed below.

**Source: Inside Airbnb**

**Link:** <http://insideairbnb.com/get-the-data.html>

## **Files Used:**

|  |  |
| --- | --- |
| **Name** | **Description** |
| neighbourhoods.geojason | GeoJSON file of neighbourhoods of the city. |
| Listings.csv | Summary information and metrics for listings in Singapore. |

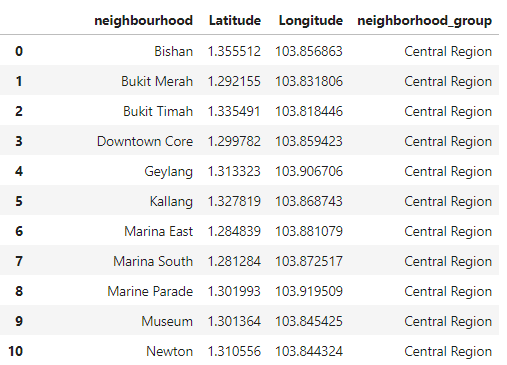
## **Details used from Source Files:**

**neighbourhoods.geojason**

|  |  |  |
| --- | --- | --- |
| **s.no** | **Column name** | **Description** |
| 1 | Neighbourhood | Name of the neighbourhoods. |
| 2 | Neighbourhood\_group | The Neighbourhood group the actual neighbourhood belongs to. |
| 3 | Coordinates | The Latitude and Longitude information that can be parsed from the neighbourhoods coordinates. |

The file is read using the library JSON in Python. With this library, the file can be read and parsed to extract the above information listed in the table. The file will contain all the Neighbourhoods, their groups and the coordinates of those neighbourhoods. By looping through all of them using a for loop, the results are stored in a Pandas Dataframe.

The screenshot containing a sample of the resulting Dataframe is shown below for reference.



**Listings.csv**

|  |  |  |
| --- | --- | --- |
| **s.no** | **Column name** | **Description** |
| 1 | Neighbourhood | Name of the neighbourhoods. |
| 2 | Neighbourhood\_group | The Neighbourhood group the actual neighbourhood belongs to. |
| 3 | Latitude | Latitude of the Airbnb listing |
| 4 | Longitude | Longitude of the Airbnb listing |

The Pandas Dataframe is used again to read the listings.csv file to download the data from the file into a dataframe.

A screenshot of the resulting dataframe with all the columns (including the ones mentioned in the above table) have been shown below.



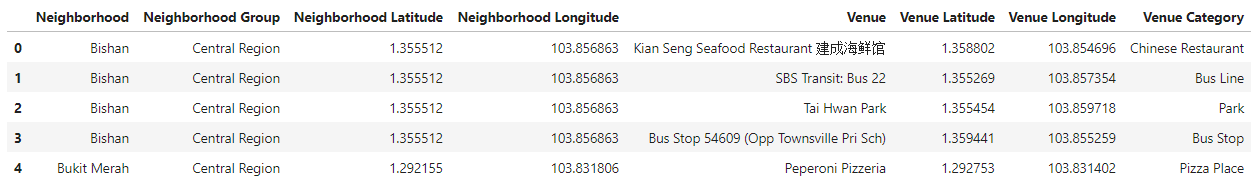
## **Foursquare Venue Data:**

In addition to the above files, calls will be made to Foursquare API using a URL to explore the different neighbourhoods using the geographical coordinates.

By using a custom defined function, all the neighbourhoods from the geojason file which has already been loaded into a dataframe are passed through the function which creates a URL to explore the venues and sent to the Foursquare API to get the venues surrounding those neighbourhoods in a json format.

The results from the Json are parsed and the venue names, latitude and longitude and the venue category are stored as a separate Pandas Dataframe.

A screenshot of the sample Dataframe has been displayed below for reference.



# Methodology:

The steps and procedures taken to derive the final clustering of Neighbourhoods is detailed below.

A brief outline of all the steps involved is listed as bullet points below.

* Getting Singapore Neighbourhood information
* Getting Venue information of the neighbourhoods using Foursquare API calls
* Getting Airbnb listing information for the Singapore Neighbourhoods.
* Calculating the distance of the Airbnb listings from the Venues.
* Categorising the distance as “close”, “moderately close” or “Far away” based on binning methods.
* Having a single dataframe that has Neighbourhood, Venue and Distance information.
* Getting the counts of listings close to each venue for each category of Distance.
* Categorising the counts of listings using binning methods as “low”, “average” or “high” no of listings and adding it to the dataframe.
* Running K-means algorithm to get the ideal number of clusters and visualising it in a map.

## Singapore Neighbourhood Information:

Using Pandas Dataframe, information about Singapore Neighbourhood is loaded into a dataframe named **neighbourhoods**. Screenshot of the dataframe is provided in the [Details used from Source Files section](#_Details_used_from).

## Venue Information using Foursquare API Calls:

Using the explore Endpoint of Foursquare API, calls are made to retrieve to venues with a predefined radius surrounding all the neighbourhoods in dataframe **neighbourhoods.** To get the information for all the neighbourhoods, a custom defined function is written to loop through all the neighbourhood and make API calls to each and every one of them and append the results to a dataframe called **all\_venues.**

**Information about the API call made:**

*LIMIT = 100 # limit of number of venues returned by Foursquare API*

*radius = 500 # define radius*

*# create URL*

*url = 'https://api.foursquare.com/v2/venues/explore?&client\_id={}&client\_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(*

*CLIENT\_ID,*

*CLIENT\_SECRET,*

*VERSION,*

*neighborhood\_latitude,*

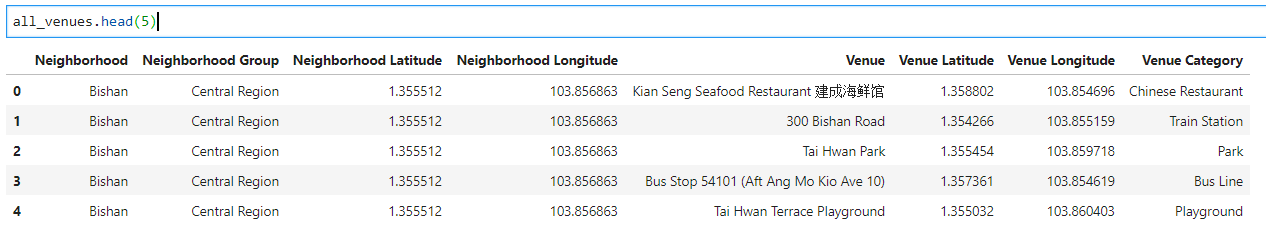
*neighborhood\_longitude,*

*radius,*

*LIMIT)*

*url*

Screenshot of the dataframe:



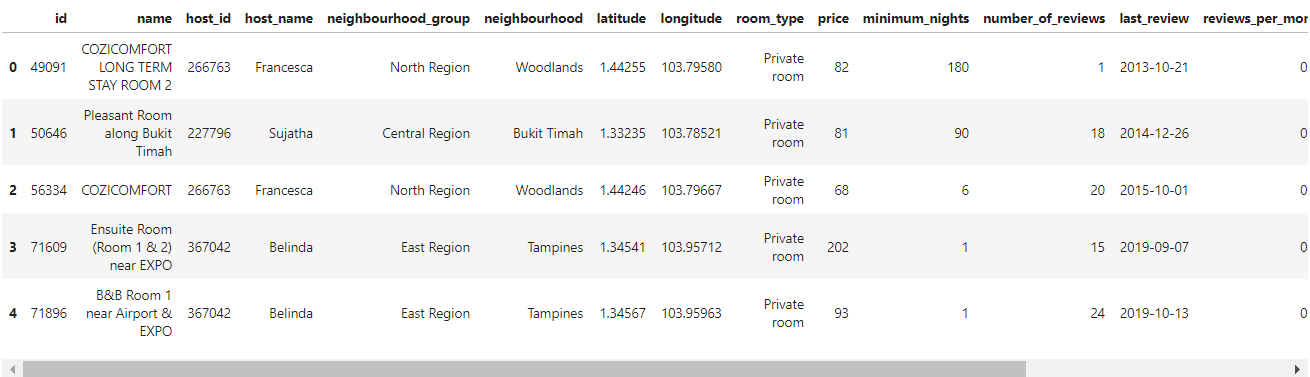
As can be seen, the coordinates of the venues are also obtained from the calls and its category.

Next, only the venues falling under the category *['Tunnel','Trail','Scenic Lookout','Water Park','Beach','River','History Museum','Theme Park','Theme Park Ride / Attraction','Art Museum','Aquarium','Pier', 'Boat or Ferry','Cruise','Art Gallery','Monument / Landmark','Waterfront','Harbor / Marina','Hot Spring','Historic Site']* from the available list of unique categories is used to create a separate dataframe called **req\_venues.**

This is because, as stated in the beginning of the project, the areas that are considered tourist attractions are used and places such as restaurants, gyms etc are excluded from the analysis.

## Airbnb Listings of Singapore

Using Pandas dataframe, Airbnb listings of Singapore which are available as a CSV file are loaded into a dataframe called **listings.**



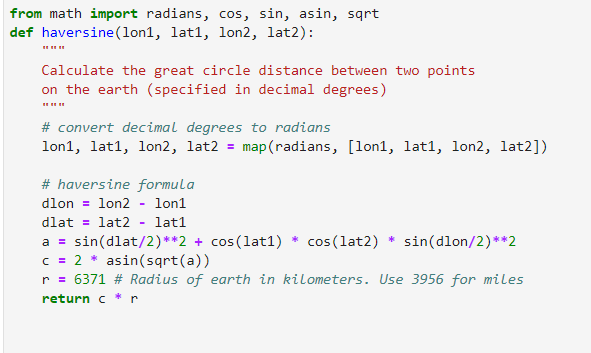
The screenshot shows above the dataframe and it has many columns of which we will be using the latitude, longitude information of the Airbnb listing for our analysis. We will also be using the count of listings in later stages described in subsequent sections.

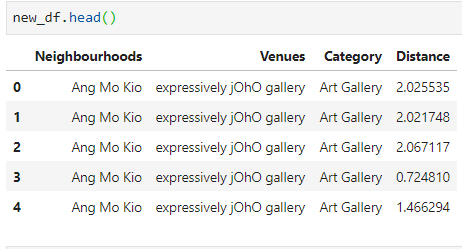
## Distance between Airbnb listing and Venue location:

Using the Haversine formula, distance between two geographical locations on Earth could be calculated. Information about the formula can be easily found in Wikipedia.

The formula is implemented in Python using a custom defined function.

The function is used to find the distance between the Airbnb **listings** dataframe coordinates and the Venues Coordinates in **req\_venues** dataframe and the results are stored in a new dataframe called **new\_df.**





**Note:** The above screenshot is correct and is not to be mistaken for duplicate information. There are multiple Airbnb listings in the neighbourhood of Ang Mo Kio. The implemented logic will take all the Airbnb listings in the neighbourhood and find the distance between the individual listing and the coordinates of the Venues in Ang Mo Kio. So suppose say there are 10 listings in Ang Mo Kio and 2 different venues in Ang Mo Kio, there would be 20 entries as the distance between the 10 listings and the 2 venues would be calculated and stored.

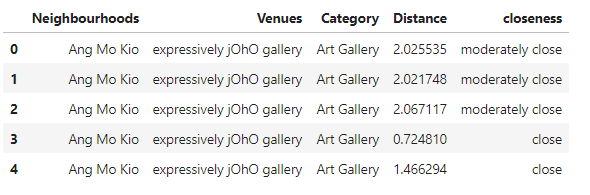
## Categorising Distance into Bins:

Using the **Cut** method of Pandas Dataframe, distance column of **new\_df** is categorised into 3 bins called “close”, “moderately close”, “far” based on the bin edges. The information is stored as a column in the dataframe.

Venues that are within 2 km are categorised as **close**.

Venues that are between 2 – 4km are categorised as **moderately close**.

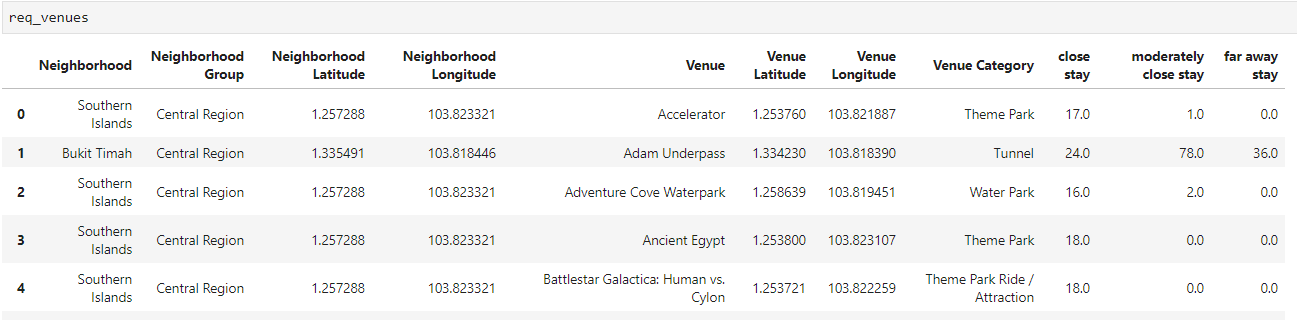
Venues that are between 4 – 7 km are categorised as **far**.



## Airbnb No of listings:

Counts of no of listings in each Distance category for every venues is found and is stored in the **req\_venues** dataframe in the columns “close stay”, “moderately close stay” and “far away stay”.

A sample screenshot is given below



## Categorising “Close Stay” into bins:

Similar to previous used binning methods, the **Qcut** method of pandas is used to create 3 categories of “low”, “average” or “high” for the no of listings that are in the “close stay category”.

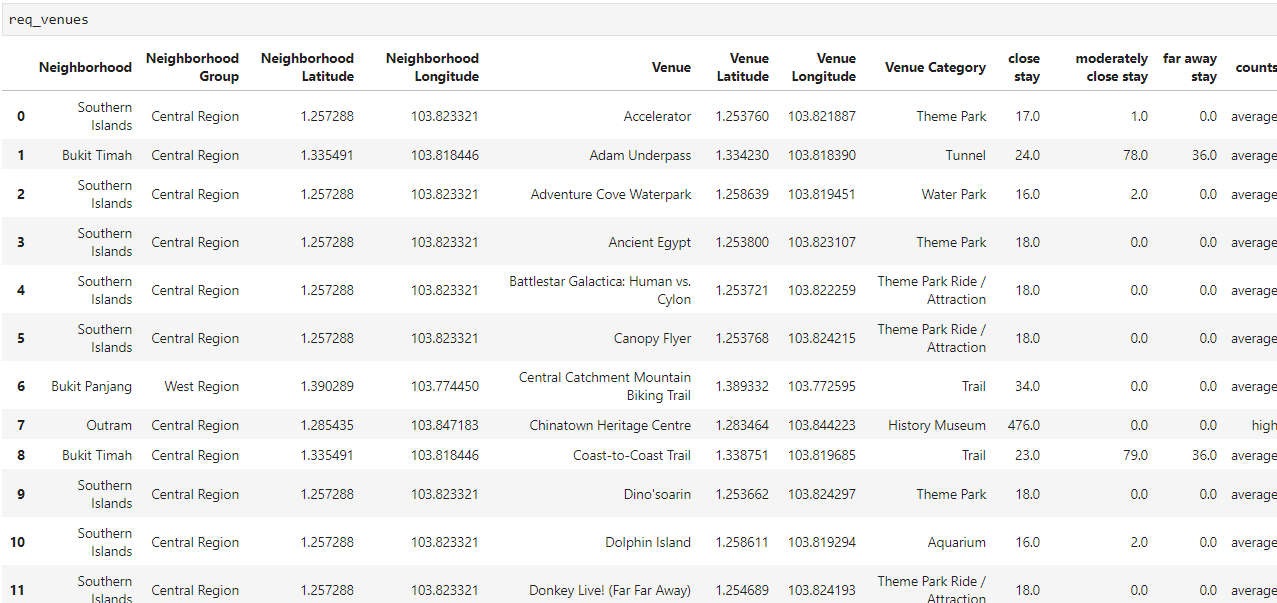
If count is between 0 -15 , its categorised as “low”

If count is between 16 -50 , its categorised as “average”

If count is between 51 -1000 , its categorised as “high”

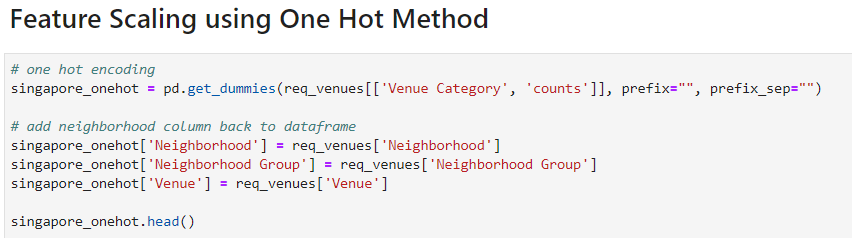
This is done only for the “close stay category”, as we are assuming that travellers would primarily be interested only in the listings that are closest to the venues.

A sample screenshot is shown below



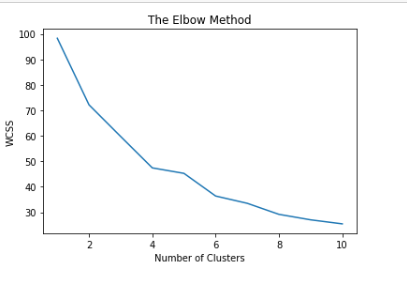
## Feature Scaling: One Hot Encoding Method

Now that we have the **req\_venues** dataframe that has all the information about the neighbourhood , neighbourhood group, venue and its coordinates,venue categories, the number of Airbnb listings etc. we can implement Feature Scaling for the **Venue** **categories** and **counts** column to change it to zeros and ones before clustering it using K-means algorithm.

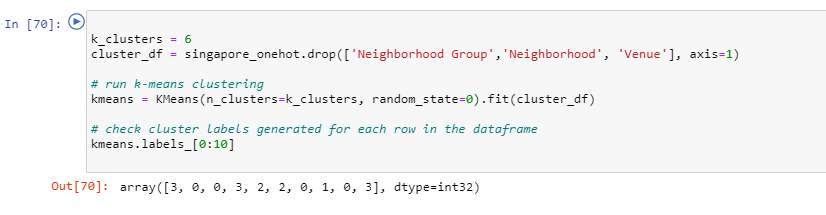


## K-Means Clustering

Using the elbow method, the best value for number of clusters (k) is found.

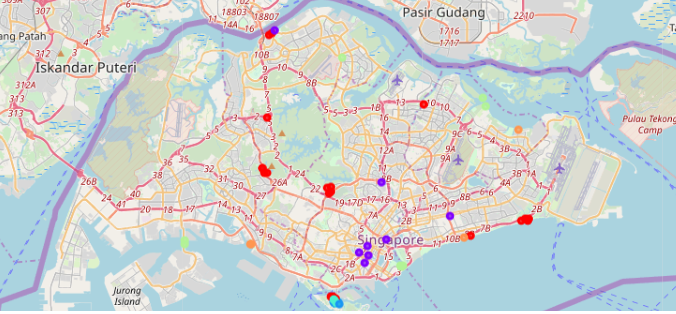


Then running the algorithm again using a value of 6 for K, the clusters are obtained and the cluster labels are added to the dataframe **req\_venues**



# Results:

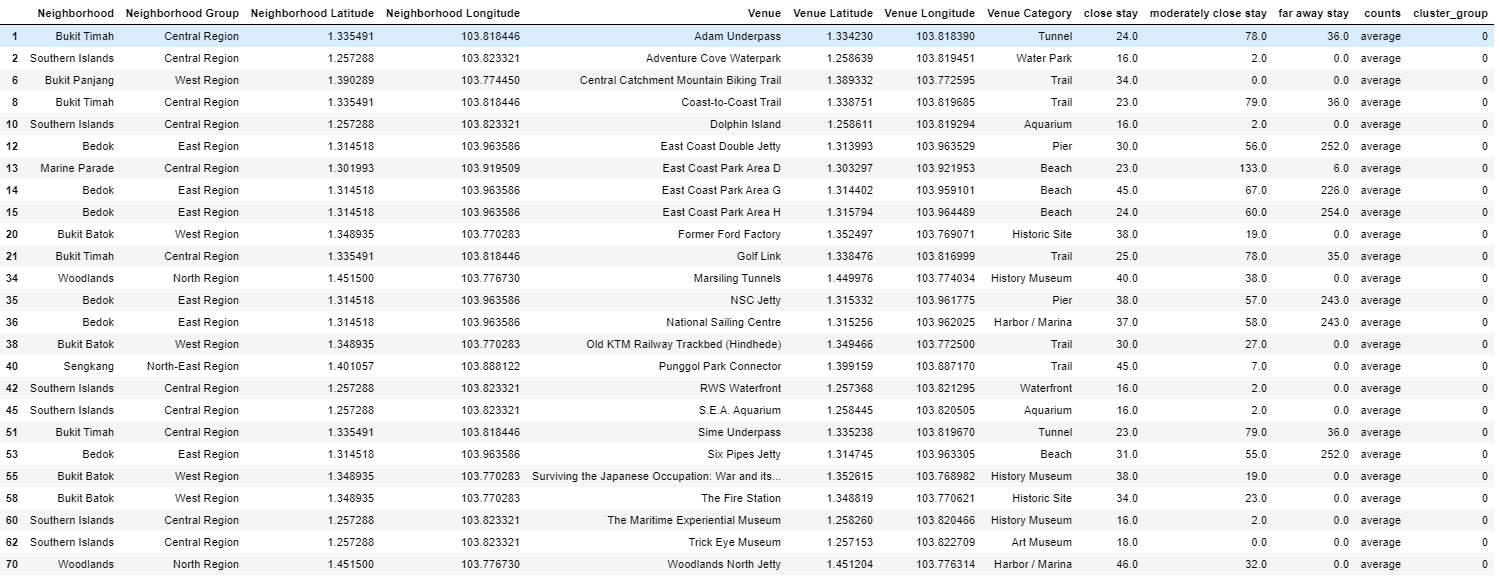
The clusters are then visualised on a map using Folium library. The clusters are displayed in different colors to distinguish them.



A look at the various cluster formations based on k-Means algorithm is given below.

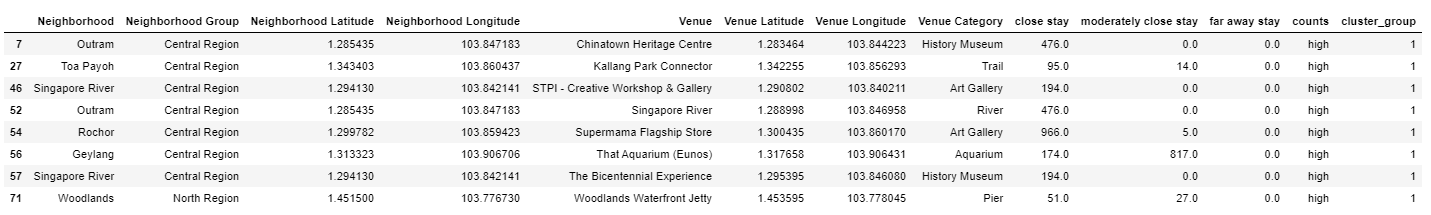
## **Cluster 0:**

This cluster has a **good mix of most of the venue** falling under different categories and also with an **average** number of Airbnb listings(within 2 km) that are close to the venues.



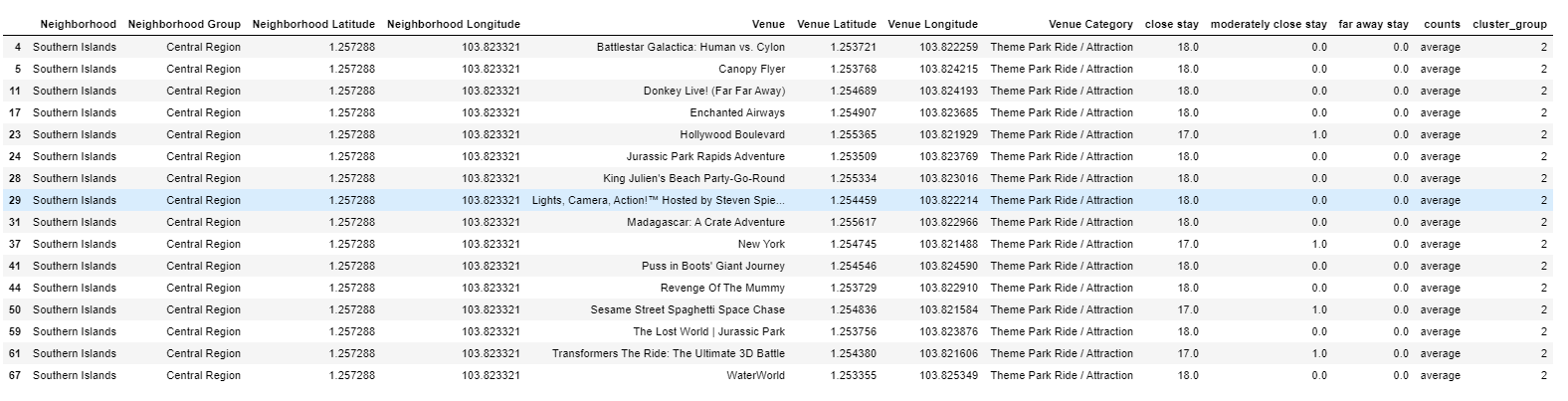
## **Cluster 1:**

This cluster has **venues** belonging to a **select few categories** with a **high** number of Airbnb listings that are within 2 kms to the venues.



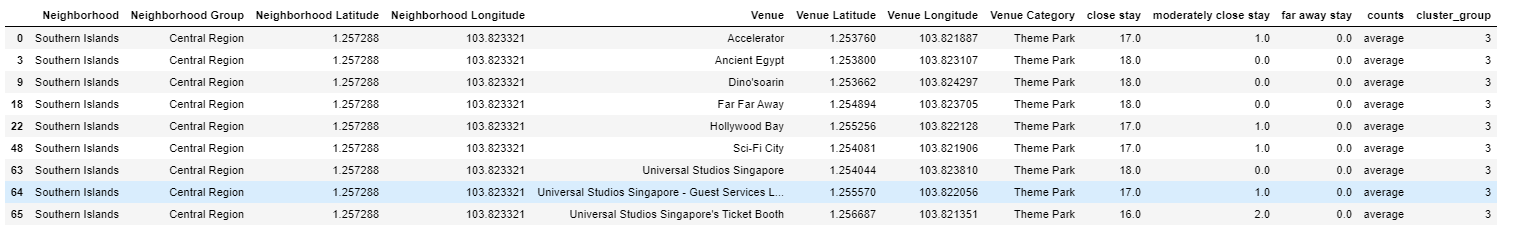
## **Cluster 2:**

This cluster has only venues belonging to the **Theme Park Ride/Attraction** category and with an **average** number of Airbnb listings that are within 2 kms to the venues.



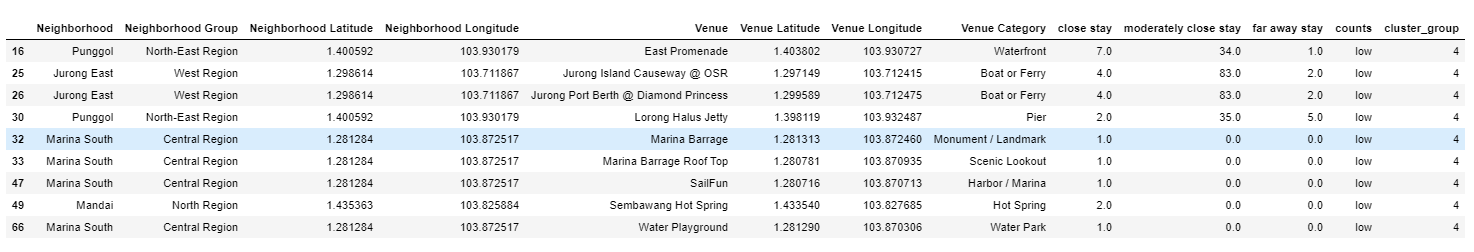
## **Cluster 3:**

This cluster has only venues belonging to the **Theme Park Ride** category and with an **average** number of Airbnb listings that are within 2 kms to the venues.



## **Cluster 4:**

This cluster represents venues of the categories that are near or related to **water bodies** that have **low** no of Airbnb listings that are within 2 kms of the venues.



## **Cluster 5:**

This cluster has **venues** belonging to a **Beach** with a **high** number of Airbnb listings that are within 2 kms to the venues.



**Note:** The data could have been cleaned so that **Theme park** and **Theme park Ride/Attraction** are put under one category before the clustering algorithm was run. But I couldn’t do this as I didn’t have time.

# Discussion:

With the results obtained from the clustering algorithm, it can be seen that each cluster has some unique property such as some might be best suitable for a particular type of Venue category whereas some other clusters might be better just because it has a high number of Airbnb rental listings that are close to the venues in that cluster.

Hence Travel agents can create specific itineraries which might suggest the best places to visit for travellers based on their needs and also based on the number of Airbnb listings.

As an example, if a traveller is looking for places that has the best trail, it can be suggested to go for the Neighbourhoods in Cluster 0 which has relatively more Trail venues and an average number of Airbnb listings.

For travellers, who prefer beach venues, Cluster 5 can be suggested as it has an high no of Airbnb listings close to those venues.

# Conclusion and Further Enhancement:

The objective that was set in this project has been met with the data about the Neighbourhoods of Singapore and the Airbnb dataset. This exercise involved meticulous work in getting and cleaning and combining the data from 3 categories (neighbourhoods’ information, venue information and Airbnb data) and also transforming the data in some instances into categorical variables before feeding it in the clustering algorithm. Finally, the results have also been visualised for better readability thus covering all the aspects of Machine learning.

It is to be noted however, that this model could be refined with better use of the Airbnb data. For example, the Airbnb dataset has information relating to the price of the listings, type of property listed, review of the property etc. These data could be cleaned and used in the model to suggest options that would take into consideration all of these new parameters.

As a different approach, even Density based clustering approaches can be used to cluster the Airbnb data.

Thus, we can see that travel agents would really benefit from clustering models and the use case of these algorithms can be applied to different scenarios under different Industry needs.