KE5106 DATA WAREHOUSING FOR BUSINESS ANALYTICS

RESTAURANT ANALYTICS

A web scraping project to discover the insights on Singapore's restaurants using MongoDB as the document-oriented database program

SUBMISSION DATE: 27 AUGUST 2018

TEAM: FAMOUS FIVE

SIDDHARTH PANDEY
PRANSHU RANJAN SINGH
EDUARD ANTHONY CHAI
NYON YAN ZHENG
TAN KOK KENG

MASTER OF TECHNOLOGY IN KNOWLEDGE ENGINEERING
KE30 (2018)

Problem Statement

There are countless food and restaurant recommendation sites and blogs on the web nowadays. When we need to look for a place to dine for a certain special occasion or casual meet up with friends, we tend to google for recommendations from various blogs and reviews site to make a decision.

Many a time, these decisions are heavily based on the reviews on the food and service, the location of the restaurant and price. Also, this process of searching and reading endless reviews from others who have been at the restaurant can be time-consuming if not, confusing as one review may be the complete opposite of another.

Business Goal

Our business goal is to scrape various websites for restaurants' data and user reviews to provide an integrated, dynamic and up-to-date overview about any restaurant to help a consumer to make a quick and informed decision on which restaurant to visit next.

Questions

1. What is the overall sentiment about a restaurant from the reviews?
2. Which are the more popular restaurants based on ratings and reviews?
3. Which location to go if there is a budget constraint?
4. What are the trending restaurants now that a consumer should consider trying out?
5. Which are the restaurants that people tend to favour together?

Data Collection

For the purpose of this project, data were collected from 2 sources, Hungrygowhere and Foursquare.

Hungrygowhere

Hungrygowhere was scraped using python scripts, Beautifulsoup library and the Selenium web driver. The data is extracted by parsing their HTML page. There were some pages that needed JavaScript for the information to be exposed. For such pages, we have used the Selenium to trigger the JavaScript function. We also require geolocation information of the restaurant which cannot be extracted from Hungrygowhere site. Hence, we have used the postal code of the restaurant and then retrieve the geolocation information from OneMap API. From Hungrygowhere, we have extracted 1,083 restaurants and 12,244 reviews.

Foursquare

Foursquare data was obtained through their developer's API. An account is required to make the API calls. A credit card verified free account allows up to 500 API calls a day. To efficiently scrape all the restaurant's data in Singapore, a premium account is required. However, with the cost and time constraints of this project, we have created multiple accounts to scrape

sufficient data quickly. From Foursquare, we have extracted 2,885 restaurants and 53,472 reviews.

Data Merging

The two data sources have different value formatting (i.e. the restaurant name is formatted differently even though they are the same) that make merging more challenging.

Our strategy to merge the data is by grouping restaurants by their postal codes and then score their name similarities. Similarities are scored using the cosine distance between two names. A few threshold values were tested and we decided to use 0.2 as our threshold. Score more than 0.2 means the two names are for the same restaurant.

Although our strategy works, there is a limitation that we are aware of. In our merging process, postal code is a compulsory field. Hence, data without postal code is discarded during the merging process.

Merging the data gives us 2,754 restaurants and 58,090 reviews in total.

Data Pre-processing

The merged data is further preprocessed for analytics purposes.

Restaurant's Subzone Geo-mapping

The first preprocessing we did is to find the restaurant's subzone. It is determined by using the restaurant's geolocation: latitude and longitude. From the subzone boundary shp file obtained from URA (Urban Redevelopment Authority), we mapped the restaurant's geolocation to the subzone in Singapore.

Review's Sentiment Analysis

Another preprocessing that we did is to calculate the sentiment score of a review, where -1 is the most negative and +1 is the most positive. We used the TextBlob python library to calculate the sentiment score. TextBlob provides us with a simple API to perform Natural Language Processing tasks like sentiment analysis. However, we are aware that it has its own limitation in terms of accuracy especially when we are dealing with Singapore's English language structure. We have done an assessment on the results and concluded that it is sufficient for our use case.

MongoDB Schema

The merged and pre-processed data is then stored in MongoDB server. The MongoDB server is hosted in Google Cloud Platform. We spawned a Virtual Machine instance and set up the MongoDB server. With this server, we ensure that our team are working on the same dataset and ensure consistency of our analysis. The data is divided into 2 collections. The schema design for each collection is given below.

Restaurants Collection

The "restaurants" collection contained the basic information about each restaurant such as its location, average price as well as the subzone information that was mapped from the URA's boundary shp file.

```
{
    "_id": STRING,
    "address": STRING,
    "category": STRING,
    "latitude": FLOAT,
    "longitude": FLOAT,
    "name": STRING,
    "postal_code": STRING,
    "rating": INT,
    "source": STRING,
    "avg_price": FLOAT,
    "subzone": STRING
}
```

Key	Description
_id	Document id
address	Restaurant's address
category	Restaurant's category
latitude	Restaurant's latitude
longitude	Restaurant's longitude
name	Restaurant's name
postal_code	Restaurant's postal code
rating	Restaurant's rating
source	Data source (hungrygowhere or foursquare)
avg_price	Restaurant's average price based on reviews
subzone	Restaurant's location subzone based on Singapore map

Reviews Collection

The "reviews" collection contained reviews from all the restaurants. The information stored are such as the review text, the user rating of the restaurant, date of review and the sentiment score which was looked-up during data pre-processing stage.

```
{
    "_id": STRING,
    "body": STRING,
    "date": TIMESTAMP,
    "rating": INT,
    "restaurant_id": STRING,
    "source": STRING,
    "title": STRING,
    "user": STRING,
    "sentiment": FLOAT
}
```

Key	Description
_id	Document id
body	Review's content
date	Review's submission date
rating	User rating of the restaurant
restaurant_id	Restaurant's document id
source	Data source (hungrygowhere or foursquare)
title	Review's title
user	Username of reviewer
sentiment	Sentiment score of the review

Schema Design Considerations

We have some considerations when designing our MongoDB schema such as the design needs to adopt the schema design best practices, good queries execution performance and scalability. Hence, we have decided to have 2 main collections: restaurants and reviews. We have separated the reviews into its own collection to avoid a growing list in the "restaurants" collection. We have included the restaurant id in the reviews document so we can easily refer back the reviews to the "restaurants" collection. With this schema design, we

can exploit MongoDB features such as map-reduce and aggregation pipeline to quickly perform operations on the data. Through this, we can minimize the aggregation operations in our analysis.

Scope

The geographical scope of our project is in Singapore. Food courts and hawker centres, although present in our data, they are excluded from the analysis.

Data Analysis

Data analysis is done on both R (using the mongolite library) and Python (using the pymongo library). We have performed 4 different types of analysis: sentiment analysis of the reviews, geographical distribution of amount spent at restaurants, trending restaurants based on the number of reviews and association analysis of restaurants often liked together.

Sentiment Analysis

From the reviews sentiment score obtained during the pre-processing stage, we performed sentiment analysis to check the high frequency positive and negative bi-gram phrases from the reviews of all restaurants. From the bi-gram phrases, we also check the frequency on the restaurant level to see which restaurants have the highest count of these words.

Figure 1 shows the distribution of position and negative sentiments from all the reviews in our database. We can see that generally, the sentiment is positive where approximately 35,000 reviews bore a positive sentiment and 22,500 of them bore a negative sentiment. This means these customers who wrote reviews are generally satisfied with the restaurants in Singapore.

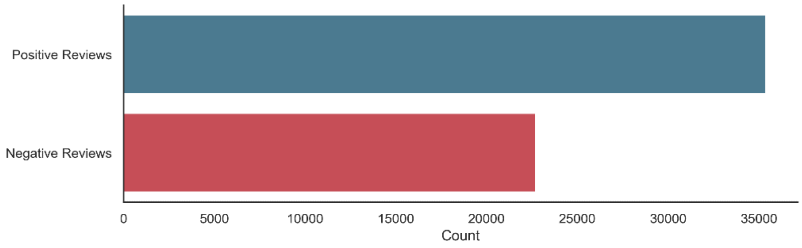

Figure 1 Distribution of Positive and Negative Sentiments

Next, we also check the most commonly written phrases in the reviews. Figure 2 and Figure 3 shows the most common positive and negative bi-gram phrases appearing in the reviews. From the bi-grams, we can see that customers of restaurants are generally concerned about the quality of the food and its service.

Figure 2 shows that when customers gave a positive feedback, the food ("good food", "food good", "great food") and service ("good service", "service good", "great service") are often mentioned. From Figure 3, we can see that when negative reviews are concerned, phrases about service ("bad service", "service bad", "slow service", "poor service", "service slow", "horrible service") appeared in 6 out of the top 10 negative phrases, more often than phrases about food. This means that an unsatisfactory service could affect a customer's experience

in a restaurant much more than unsatisfactory food quality. Figure 4 and Figure 5 show the positive and negative phrases in a word cloud.

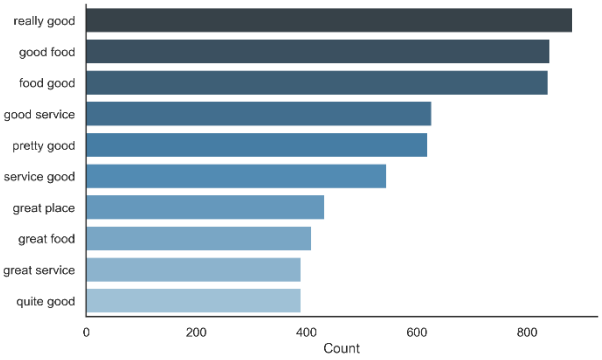

Figure 2 Frequency of positive bi-gram phrases

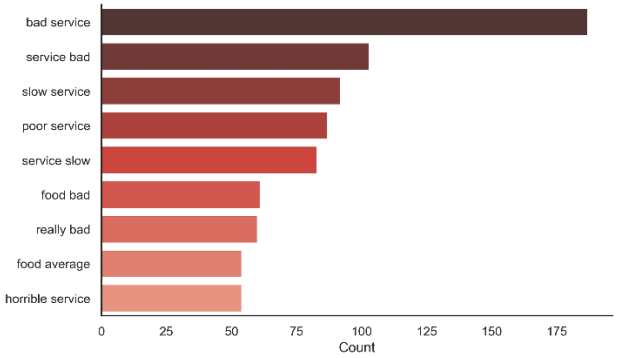

Figure 3 Frequency of negative bi-gram phrases

Figure 4 Word cloud for positive bi-gram phrases

Figure 5 World cloud for negative bi-gram phrases

From the list of frequently appearing positive and negative phrases, we did a count of the phases on the reviews at the restaurant level to see which restaurants are often associated with these positive and negative phrases in their reviews. Table 1 and Table 2 give a summary of the word count.

Table 1 Top restaurants associated with common positive phrases

Phrase	Restaurant Name	Count
fine dining	Forlino	19
friendly staff	Chicken Up (Tanjong Pagar)	13
good food	Chicken Up (Tanjong Pagar)	19
good service	Chicken Up (Tanjong Pagar)	15
great food	Chicken Up (Tanjong Pagar)	11
great place	Chicken Up (Tanjong Pagar)	12
great place	Hood Bar & Music	11
great service	Chicken Up (Tanjong Pagar)	23
high tea	The Rose Veranda	14
medium rare	Bedrock Bar Grill	11
really good	Chicken Up (Tanjong Pagar)	23

Table 2 Top restaurants associated with common negative phases

Phrase	Restaurant Name	Count
bad service	Banafee Village Restaurant	3
bad service	Ambush	3
food bad	The Boiler Seafood Bar & Beer (Howard)	1
horrible service	Brotzeit German Bier Bar & Restaurant (Vivocity)	2
horrible service	El Patio Mexican Restaurant & Wine Bar	2
poor service	LeVeL33	2
poor service	Restoran Todak	2
really bad	Prego	2
service slow	Alaturka Restaurant	1
slow service	Watami Japanese Casual Restaurant	3
slow service	Arnolds Fried Chicken	2

From Table 1, we can see that Chicken Up at Tanjong Pagar has received very positive reviews about its food and service. In summary, the sentiment analysis could provide a general sentiment from customers who have been at the restaurant from various website, reducing the need to go from website to website to read the reviews about the restaurants.

Geographical Distribution of Amount Spent

The average amount spent figure for every restaurant is based on the average of customer 's submission to the website. From the restaurant's average amount spent, we analyzed the average amount spent by subzones in Singapore to see if the location of restaurants affects the average amount spent at restaurants. Using MongoDB's aggregation framework, we obtained the average amount spent by subzones in Singapore from the "restaurants" collection and plotted the results on the Singapore subzone boundary shp file.

```
restaurants$aggregate('[{"$group":{
                              "_id": "$subzone",
                              "avgPrice": {"$avg":"$avg_price"}}}]')
```

From the geographical plot in Figure 6**Error! Reference source not found.**, we found that people generally spend more when they are dining at restaurants located in the central area in Singapore and the amount gradually decreases as the location is further away from the central area. The white areas show no restaurant data present in those subzones. We have also plotted a similar map in a html widget format, presented in file "widget-avg_amount_map.html" and in Figure 7.

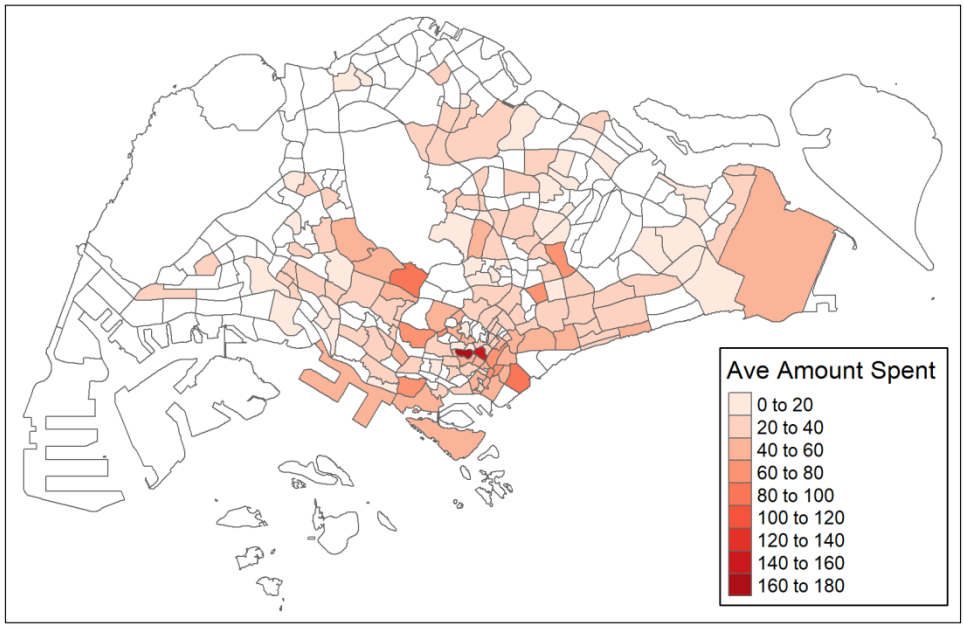

Figure 6 Average restaurant amount spent by subzones in Singapore

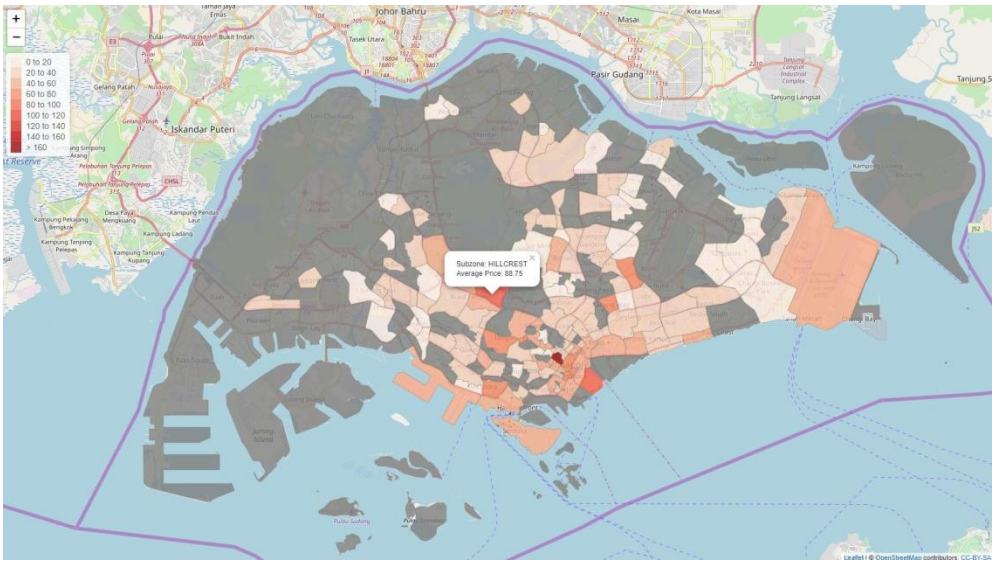

Figure 7 html Widget showing the distribution of average spent

We have extracted the subzones by the price range in **Error! Reference source not found.**. We can see that generally these places are in the Central Business District, Marina Bay and Orchard Road. As a summary, if consumers have a limited budget for a meal, they can consider visiting restaurants that are further away from the central of Singapore.

Table 3 Subzones with price range above $40

Average Price Range	Subzones
Above $100	Institution Hill, Fort Canning
Between $80 to $100	Marina South, Hillcrest
Between $60 to $80	City Hall, Tanglin, Raffles Place, Kallang Way, Clifford Pier, Tai Seng, Telok Blangah Drive
Between $40 to $60	Mountbatten, Chinatown, Tanjong Rhu, Anson, Marymount, Cecil, Marina Centre, Robertson Quay, Bayfront, Central, Sentosa, Bugis, Changi Airport, Somerset

Trending Restaurants

We have used the number of reviews as a proxy to find trending restaurants. This is based on a rationale that if a restaurant is trending, more people would be talking and commenting about it. From the "reviews" collection, we obtained the number of reviews in a 90-days period using MongoDB's map-reduce command. A sample map-reduce operation that was performed through R's mongolite library to count the number of reviews for each restaurant is shown below.

```
reviews$mapreduce(
                map = "function(){emit(this.restaurant_id, 1); }",
                reduce = "function(key, values){return values.length; }",
                query = '{"date": {"$gt": 1526256000, "$lte": 1534032000}}' )
```

From the data, we have identified the top 6 restaurants that were trending in the past 90 days between 14 May to 12 Aug 2018 where they have received more than 6 reviews during

the 90 days period. Consumers will be able to use this information to visit trending restaurants. The top 6 trending restaurants are (1) Indochili (Zion Road) (2) Trattoria Nonna Lina (3) Ronin (4) 23 Jumpin (5) Princess Terrace (6) Escape Restaurant Lounge.

We further extend the analysis to include the 90 days before 14 May 2018, which is from 13 Feb to 13 May 2018 and we found that there are 3 restaurants with more than 5 reviews for both period. This is shown in Table 4.

Table 4 Trending restaurants in both 90 days period

Restaurant Name	**Reviews Count** (13 Feb – 13 May)	**Reviews Count** (14 May – 12 Aug)
DePizza	106	5
Cast Iron	6	5
Imperial Treasure Super Peking Duck Restaurant @ Paragon	5	5

From the table above, we also found that DePizza received over a hundred reviews in the 90 days from 13 Feb to 13 May 2018. The unusually high number of reviews could be due to the restaurant running a promotional event. Looking deeper into DePizza's reviews, we found that this restaurant generally received good reviews and the average rating is 4.85 out of 5. This could be one of the restaurants to go to when one feels like eating pizza.

The trending restaurant analysis allows us to find out which are the trending restaurants based on the reviews and hence recommend consumers to check out those trending restaurants.

Association Analysis

Using the "arules" library in R, we have implemented the APRIORI algorithm to mine the association rules to find out which restaurants are often liked to together by a customer. We have assumed that if a user gave a 5-star rating to a restaurant, they must have liked the restaurant.

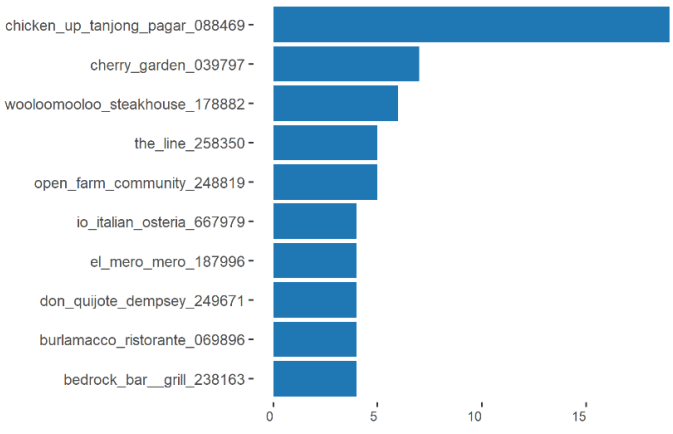

Figure 8 Restaurants with the highest number of 5-star ratings

From the data, the restaurants that each user has rated 5 stars are converted into a transactional format and analyzed using the apriori algorithm. Figure 8 shows the top 10 restaurants with the highest number of unique 5-star ratings. Same user who rated the same restaurant twice will not be counted in this analysis.

We can see that Chicken Up (Tanjong Pagar) has received the highest number of the 5-star rating among all other restaurants in this database.

Using the transactional data, a total of 53 rules are mined from 143 unique reviewers visit patterns to 221 restaurants. Figure 9 shows a snapshot of some of the 53 rules that were obtained.

	LHS	RHS	support	confidence	lift
	All	All		All	
[13]	{lewin_terrace_179290}	{chicken_up_tanjong_pagar_088469}	0.014	0.667	5.053
[14]	{chicken_up_tanjong_pagar_088469}	{lewin_terrace_179290}	0.014	0.105	5.053
[7]	{tsukada_nojo_chinatown_point_059413}	{chicken_up_tanjong_pagar_088469}	0.014	1.000	7.579
[8]	{chicken_up_tanjong_pagar_088469}	{tsukada_nojo_chinatown_point_059413}	0.014	0.105	7.579
[39]	{open_farm_community_248819}	{wooloomooloo_steakhouse_178882}	0.014	0.400	9.600
[40]	{wooloomooloo_steakhouse_178882}	{open_farm_community_248819}	0.014	0.333	9.600
[3]	{lantern_049326}	{cherry_garden_039797}	0.014	0.667	13.714
[4]	{cherry_garden_039797}	{lantern_049326}	0.014	0.286	13.714
[37]	{ristorante_da_valentino_287994}	{wooloomooloo_steakhouse_178882}	0.014	0.667	16.000

Figure 9 Snapshot of some of the rules from the apriori algorithm

We have further plotted the rules into a network graph and found that restaurants are generally clustered into 8 clusters. Snapshots of 2 of the clusters are shown in Figure 10 and Figure 11 where the darker red circle means a higher lift value for the rule. An interactive widget for all the network graphs is given in html file "widget-rules.html".

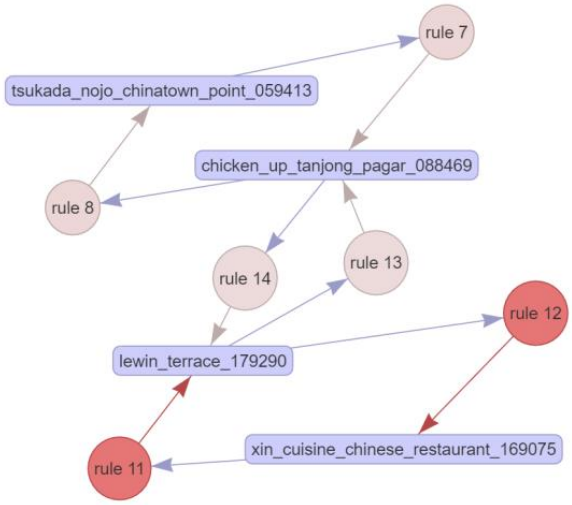

Figure 10 Cluster of restaurants

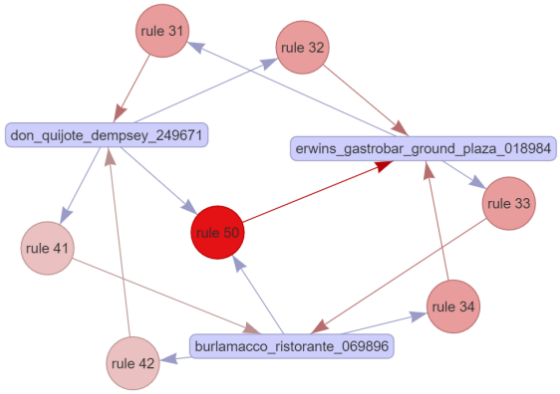

Figure 11 Cluster of restaurants

The list of restaurants in each of the cluster is also extracted in table format shown in Table 5. The rules obtained from this analysis could be used as the basis of a rules-based recommender system to recommend new restaurants to a consumer based on the restaurants that they have been and liked before. For example, if someone enjoyed his or her experience at The Song of India (cluster 7), then we can definitely recommend him or her to also try the Zaffron Kitchen at East Coast Road.

Table 5 Clusters of restaurants based on association analysis

Cluster 1 1. Tsukada Nojo (Chinatown Point) 2. Chicken Up (Tanjong Pagar) 3. Lewin Terrace 4. Xin Cuisine Chinese Restaurant	**Cluster 2** 1. Wooloomooloo Steakhouse 2. Open Farm Community 3. Gattopardo Ristorante di Mare 4. Ristorante Da Valentino
Cluster 3 1. Ito Kacho Yakiniku 2. Nassim Hill Bakery Bistro Bar 3. Etna Italian Restaurant (Duxton Road)	**Cluster 4** 1. Alba 1836 2. Punjab Grill 3. The Boiler Seafood Bar & Beer (Howard)
Cluster 5 1. Don Quijote (Dempsey) 2. Erwins Gastrobar (Ground Plaza) 3. Burlamacco Ristorante	**Cluster 6** 1. Lantern 2. Prego 3. Cherry Garden
Cluster 7 1. The Song of India 2. Zaffron Kitchen (East Coast Road)	**Cluster 8** 1. AXIS Bar & Lounge 2. Salt Grill & Sky Bar

Further Developments

1. Currently, we have only scrape restaurant data from 2 websites. We can extend this to include data from more websites such as TripAdvisor, Yelp and review blogs such as ladyironchef, Daniel's Food Diary, Sethlui.com and many others. We can consolidate the data from all these websites and analyze the restaurants from all these sources.
2. We can build a recommender system based on the consumer's preference on location and budget, incorporating the association rules to recommend restaurants that are trending and those with positive sentiments.
3. The geographical map that we have plotted currently only contained the average amount spent across Singapore. We could extend this geographical map to include the results from the associate mining as well as incorporate the information on trends and sentiments on the map so that all information is presented in one view to the consumer.

Summary

With big data and advanced machine learning tools nowadays, we often see big organizations conducting analysis to understand its customers better. On the other hand, with the ability to scrape the web for data, we as consumers are now able to harness the insights from various websites and blogs to our advantage to visit only the best restaurants out there. This indirectly also provide an incentive for the restaurants to buck up if they are falling short of the quality and standards. Our analysis and the further developments allow consumers to make a well-informed choice based on the reviews of other customers.

As a summary of the assignment, the team has gain invaluable experience in web scraping and an opportunity to deviate away from the traditional database through the MongoDB document database.