IBM Data Science on Coursera: Capstone
The Battle of Neighborhoods

1 INTRODUCTION

Finding places to visit when we are in a new place is something that could take too long, by indecisions or a lot of information to get in. Because of that, Big Data and Data Science can help us to make decision based on a data set of information about these locations. As an example, data can be collected from websites that allow users to publish their reviews and opinions about the experience found in this place. Then, it is possible, with Data Science to automate the ‘decision-making’ with powerful algorithms which take just few seconds to show possible places to go based in data collected. Furthermore, just with a simple initial data it turns possible to make decisions based in it. In this project, the user can simply input his Borough at Toronto, Canada, and get 4 clusters about places to go in. Coffee shop, parks, stores, hotels and so on. It is easy handled with just one input. It can be done multiples time as needed.

1.1 PROBLEM SITUATION

The algorithm has to compare different places in Toronto based in Data Collected on the internet using APIs and another database to help people choosing the best place to go. It includes parks, gym, restaurants, movie theatres etc. To do that, anyone who has not been in Toronto before could use it to provide the best decision based in a less time possible just entering the Borough that the person is in.

2 DATA

It will be used data based on internet resources such as Wiki, websites. Also, API settings will be used to return Data based in postal codes (names) and GEO Spatial location.

2.1 REQUIRED DATA

A list with postal codes in Toronto, a dataset that contains different types of venues located in Toronto with their respective postal code and GEO Location. Therefore, data should contain Postal Code, Borough, Neighbourhood, Latitude, Longitude. Based in the avenue, a list of places required will be returned by the API used in the algorithm.

2.2 DATA SOURCE

To apply these skills, a List of Postal Codes of Canada: Toronto used is available at Wikipedia. Foursquare API can provide information and Data about places inside avenue and next from the avenues and localization. Those data are returned as JSON and will be handled in Data Science with Python.

2.3 DATA WRANGLING

Using Python Data Science, it is possible to make decisions based in an algorithm that uses clusters and k-means to show the best result needed. For this process, all the learning skills gotten around the whole specialization will be used, therefore the principal is: Python Data Science Analysis and Machine Learning.

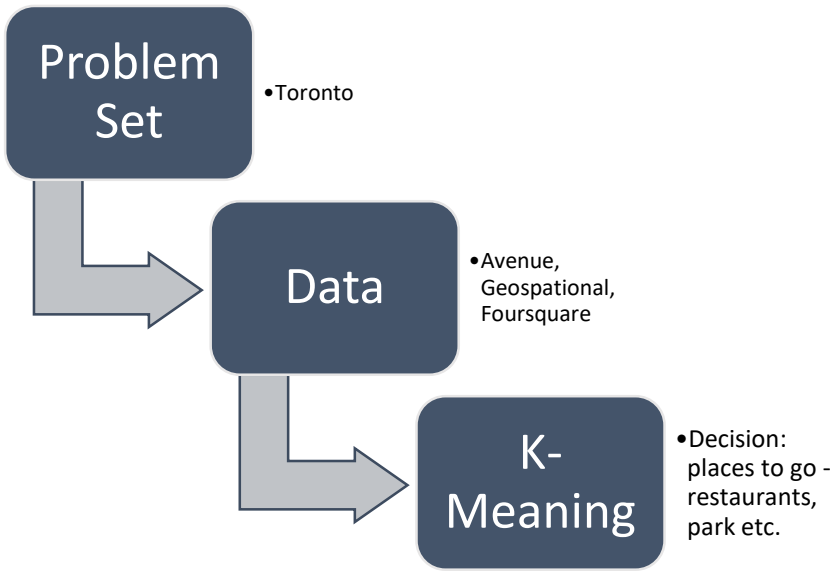
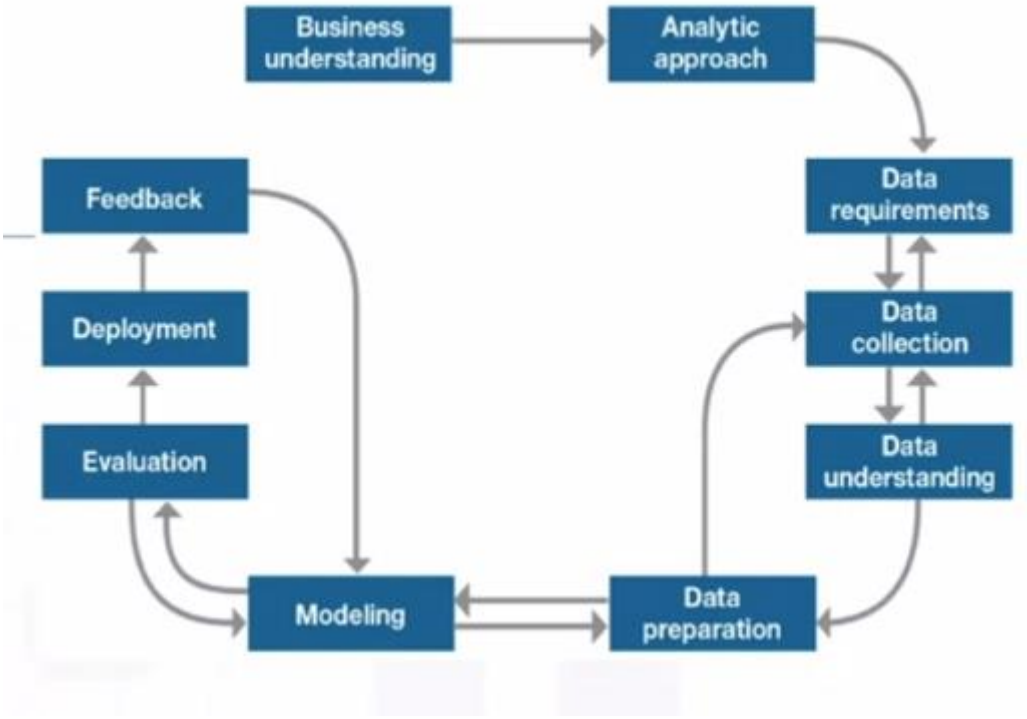


Figure 1: Algorithm idea in a nutshell.

3 METHODOLOGY

Data Science methodology consist in ten steps as learned in the IBM specialization. First thing is the Business Problem. Here, we would like to get a result based in an input. Which places to show for the user? To solve it, data is necessary. Then, we have Data Understanding: you learn more about your data the more you study it. After, the Deployment. Refined model must be redeployed. Next, Modelling to Evaluating that is the importance of asking if the question has been answered. Finally, Requirements and Collection which tell us the importance of identifying the correct sources of data for the project. The next figure shows step by step in a nutshell.



.

For this project, a Jupyter Notebook will be used. It has been chosen that IBM Watson is the best platform to attend the required skills using Python 3.x. The problem to solve is about places to go around Toronto, Canada. By just typing a Borough, the algorithm will make a K-Meaning Decision to show four clusters. These clusters contain places to go around the Borough. It is possible through the data collected on the internet using Python libraries as Pandas, Numpy etc and API such as Foursquare.

The steps which the algorithm works is:

a. Get Toronto Coordinates;
b. get Toronto Borough and coordinates;
c. use Foursquare API to return data about the Borough selected;
d. do a K-Meaning to show 4 clusters based in places.

Finally, the project has been developed to show 4 clusters to the user containing the places in the Borough selected.

4 RESULTS AND DISCUSSION

The algorithm using K-Meaning shows that it is possible to decide between data just with few inputs section. First, postal data was collected from Wikipedia and handled to merge GEO location in the data frame. The result is shown here:

	Postal code	Borough	Neighborhood	Latitude	Longitude
0	M3A	North York	Parkwoods	43.753259	-79.329656
1	M4A	North York	Victoria Village	43.725882	-79.315572
2	M5A	Downtown Toronto	Regent Park , Harbourfront	43.654260	-79.360636
3	M6A	North York	Lawrence Manor , Lawrence Heights	43.718518	-79.464763

Then the clustering process is started. As we get Toronto coordinates, a folium map is rendered. After, a input message is shown: *Please, type your Borough at Toronto. Example: Downton Toronto.* This is the most important step. User must input a correct Borough that turns Foursquare API step. The example used here is Downtown Toronto (as shown in the example message).

```
# Global Variable to Borough
PLACE_BOROUGH = input(print('Please, type your Borough at Toronto. Example: Downton Toronto'))
```

```
Please, type your Borough at Toronto. Example: Downton Toronto
    Downtown Toronto
```

After that, the Borough GEO location will be generated as shown in the next figure:

Exploring Toronto Selected Borough

```
neighbourhood_latitude = toronto_data['Latitude'].mean() #toronto_data.loc[0, 'Latitude'] # neighborhood l
atitude value [0]
neighbourhood_longitude = toronto_data['Longitude'].mean() # toronto_data.loc[0, 'Longitude'] # neighborho
od longitude value [0]

#neighbourhood_name = toronto_data.loc[0, 'Neighborhood'] # neighborhood name
borough = toronto_data.loc[0, 'Borough']

# Print all those informations
print('Latitude and longitude mean values of the Borough {} are {}, {}.'.format(borough,
                                                                                 neighbourhood_latitude,
                                                                                 neighbourhood_longitude))
```

```
Latitude and longitude mean values of the Borough Downtown Toronto are 43.65459717894736, -79.383971568421
05.
```

Decisively, K-mean returns the clusters. In the first one, it is possible to see almost restaurants place to go in.

```
c_toronto_merged.loc[c_toronto_merged['Cluster Labels'] == 0, c_toronto_merged.columns[[1] + list(range(5,
 c_toronto_merged.shape[1]))]]
```

	Borough	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Mos Commo Venue
2	Downtown Toronto	0	Clothing Store	Coffee Shop	Café	Middle Eastern Restaurant	Bubble Tea Shop	Cosmetics Shop	Japanese Restaurant	Lingerie Store	Electroni Store
3	Downtown Toronto	0	Coffee Shop	Café	Italian Restaurant	Cocktail Bar	Restaurant	Gastropub	Beer Bar	Bakery	Clothing Store
7	Downtown Toronto	0	Coffee Shop	Restaurant	Café	Gym	Bakery	Thai Restaurant	Bookstore	Hotel	Lounge
11	Downtown Toronto	0	Café	Restaurant	Bar	Italian Restaurant	Japanese Restaurant	Bookstore	Bakery	Yoga Studio	Beer Bar
12	Downtown Toronto	0	Café	Vietnamese Restaurant	Coffee Shop	Dumpling Restaurant	Vegetarian / Vegan Restaurant	Mexican Restaurant	Bar	Grocery Store	Arts & C Store
16	Downtown Toronto	0	Coffee Shop	Pizza Place	Café	Restaurant	Pub	Italian Restaurant	Pharmacy	Bakery	Conveni Store
18	Downtown Toronto	0	Coffee Shop	Japanese Restaurant	Gay Bar	Restaurant	Sushi Restaurant	Gastropub	Pub	Burger Joint	Hotel

The second one shows just a Park that is a good location to. And so on in those others clusters.

```
c_toronto_merged.loc[c_toronto_merged['Cluster Labels'] == 1, c_toronto_merged.columns[[1] + list(range(5,
 c_toronto_merged.shape[1]))]]
```

	Borough	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue
14	Downtown Toronto	1	Park	Trail	Playground	Dance Studio	Dumpling Restaurant	Donut Shop	Doner Restaurant	Dog Run	Distribution Center

5 CONCLUSION

In conclusion, the algorithm has worked as expected. Four clusters were generated. Using the Data Science IBM methodology learned through specialization coursers ensured a great tool and skills (python and jupyter notebooks) to make possible the final Capstone Project. The battle of the neighbourhoods is done. With a just simple input user can decide in places to go or visit.