THE BATTLE OF THE NEIGHBORHOODS

A PRESENTATION ON THE PROJECT UNDERTAKEN DURING SUMMER TRAINING 2020

TABLE OF CONTENTS

01

ABOUT THE TRAINING

A brief introduction about the course undertaken

02

PROBLEM AND SOLUTIONS

A dive into the problem statement and proposed solutions

03

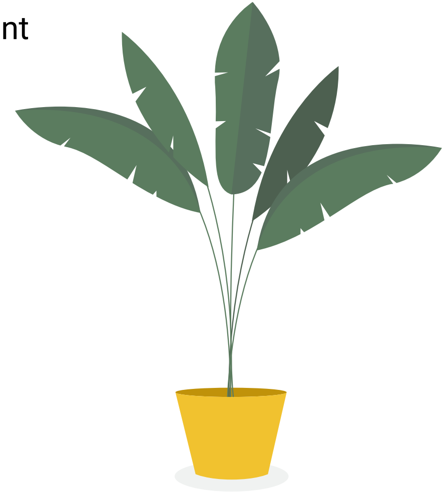
TECHNICAL SPECIFICATIONS

Explanation of the technical aspects and technologies employed in the solution

04

ANALYSIS AND DISCUSSION

Multifacet analysis of the project and future scope discussion.



ABOUT THE TRAINING

A BRIEF INTRODUCTION ABOUT THE TRAINING COURSE AND ASSOCIATED INSTITUTIONS

THE INSTITUTIONS

- The training course completed was offered by the multinational corporation IBM via the massive online learning platform Coursera.
- International business machine or IBM is a global technology company that provides hardware, software, and cloud based services to its clients. The company also provides certain cognitive computing services.
- The focus of IBM in the past couple of years has been to shift from an institution that is an infrastructure player to one that is more cloud and data-driven.
- The focus of the company has been on providing cloud-based, behind the scenes services and products to many large corporations. The most popular one is the IBM “Watson”.
- Powered by the latest innovations in machine learning, Watson is the open, multi-cloud platform that lets one automate the AI lifecycle. One can build powerful models from scratch or speed time-to-value with pre-built enterprise apps.
- Coursera is a world-wide online learning platform founded in 2012 by Stanford computer science professors Andrew Ng and Daphne Koller that offers massive open online courses also known as MOOCs, specializations, degrees, professional, and master track courses.
- Coursera works with universities and other organizations to offer online courses and degrees in a variety of subjects such as engineering, data science, machine learning, mathematics, business, computer science, digital marketing, humanities, medicine, and many others.

THE COURSE

- Machine learning is a subsection of the artificial intelligence domain that may be defined as the use of algorithms and computational statistics to learn from data without being explicitly programmed.
- The training and the associated project were completed under the guidelines and timelines associated with the course titled “Machine learning with python” as offered on the online platform Coursera.
- The course is offered by the multinational corporation IBM and focuses on the accusation of hands-on skills to practically apply complex machine learning concepts to real-world problems.
- The course material and projects are spread over a duration of six weeks and teach all the relevant skills that one needs to equip themselves with to gain industry-level insights and experience into the field of machine learning.
- The course dives into the basics of machine learning using an approachable, and well-known programming language, Python.
- The course focuses on two main components: First, it teaches the general purpose of machine learning and where it applies to the real world.
- Second, it gives a general overview of the purpose of machine learning topics such as supervised vs unsupervised learning, model evaluation, and machine learning algorithms.
- At the end of the course, a free for individual interpretation, the peer-graded project was assigned which was to be created using all of the tools and concepts previously taught.

WEEKLY TASK SUMMARY

WEEK 1

- Introduction to machine learning.
- Applications of machine learning.
- General overview of supervised and unsupervised learning.
- Final quiz (15% weightage).

WEEK 2

- Introduction to regression.
- Types of regression (linear, non-linear,etc).
- Evaluation metrics for accuracy determination.
- Final quiz (weightage 15%).

WEEK 3

- Introduction to classification.
- Classification algorithms such as KNN and more.
- Model evaluation metrics.
- Final quiz (15% weightage).

WEEK 4

- Introduction to clustering approaches.
- Types of clustering algorithms such as K means clustering etc.
- Final quiz (15% weightage).

WEEK 5

- Introduction to recommendation systems.
- Collaborative and content based filtering algorithms.
- Final quiz (15% weightage).

WEEK 6

- Free for interpretation capstone project.
- Preliminary information to construct project.
- Final submission (25% weightage).

PROBLEM
AND SOLUTION

- The problem statement
- Elaboration
- Target solution
- SWOT analysis of product

1 = Brampton
Richmond Hill
Pickering
Markham
Vaughan
Rouge River
401
407
400
404
Highland Creek
Scarborough
West Highland Creek
North York
Downsview Airport
Parkway
1
Humber River
427
409
East York
Don Valley
York
Etobicoke
0
8 km
Lester B. Pearson Intl. Airport
Mimico Creek
Old Toronto
Don River
Union Station
Greater Toronto Area
410
403
Etobicoke Creek
Gardiner Expressway
Billy Bishop Toronto City Airport
Durham
York
Peel
City of Toronto
Halton
Mississauga
Lake Ontario
Lake Ontario

THE PROBLEM

When faced by the possibility of a move, one of the main concerns people have is to find a neighborhood that meets their standards and is somewhat similar to the area they live in currently so as to have the smoothest transition possible.

Although there exist web applications that help find suitable apartments, most of them pay little to no attention to the neighborhood the rentals are situated in.

Thus the need of the hour is to develop an application that helps users find similar neighborhoods by using machine learning techniques.

ELABORATION

- Consider the following scenario: you live on the west side of the city of Toronto. One day, a company from the east side of the city offers you a lucrative position. The pay is better, the benefits fantastic but there is one catch.
- To make sure that you make it to work on time each day and don't have to spend 3-4 hours of your valuable time traveling, you have to move. The problem is that you quite like the neighborhood you live in currently.
- The area is well connected, has plenty of grocery stores, and your personal favorite café is just an arm's reach away. So, what do you do? Do you give up the job opportunity and the better pay and career prospects that come with it or blindly risk moving to a neighborhood you know next to nothing about?
- It can be unanimously agreed that what a person most needs in such a situation, in addition to the courage to give up morning coffee, is more information about prospective neighborhoods.
- The most common solution is to contact a realtor and provide them with a list of your requirements. Although this is the most common practice, it is highly unreliable.
- The person who is contacted may not be an expert on the matter. Moreover, it is highly probable that they are not intimately acquainted with each neighborhood and are more concerned with the accommodation aspect rather than the surrounding factors.
- In such a situation, the only alternative that remains is for us to turn to technology and since machine learning and more importantly data science are fields that exclusively focus on gaining insights from raw data, we shall use them in this difficult time.
- Providing a solution to the above-posed problem is exactly what the final capstone of the summer training is focused on.

THE SOLUTION

Given the appropriate dataset, segment and cluster similar neighborhoods in the city of Toronto into groups or clusters of identical properties. Given the location coordinates of the center of each neighborhood, contact the Foursquare API to collect a list of popular venues from that neighborhood. Following this using machine learning and data science techniques to arrive at an appropriately grouped data set with cluster labels assigned to each area so that the data can be queried to obtain similar neighborhoods to a desired one.

Thus, the ultimate goal of the project is to create an appropriately labeled dataset which can suitably be queried. The user should be able to enter the coordinates of a preferred neighborhood and be provided in return with a list of all the other neighborhoods in the city with similar properties.

TECHNICAL SPECIFICATIONS

PLATFORMS USED

FOURSQUARE

This is the API service used to obtain the nearby venue data for each neighborhood given geographical coordinates.

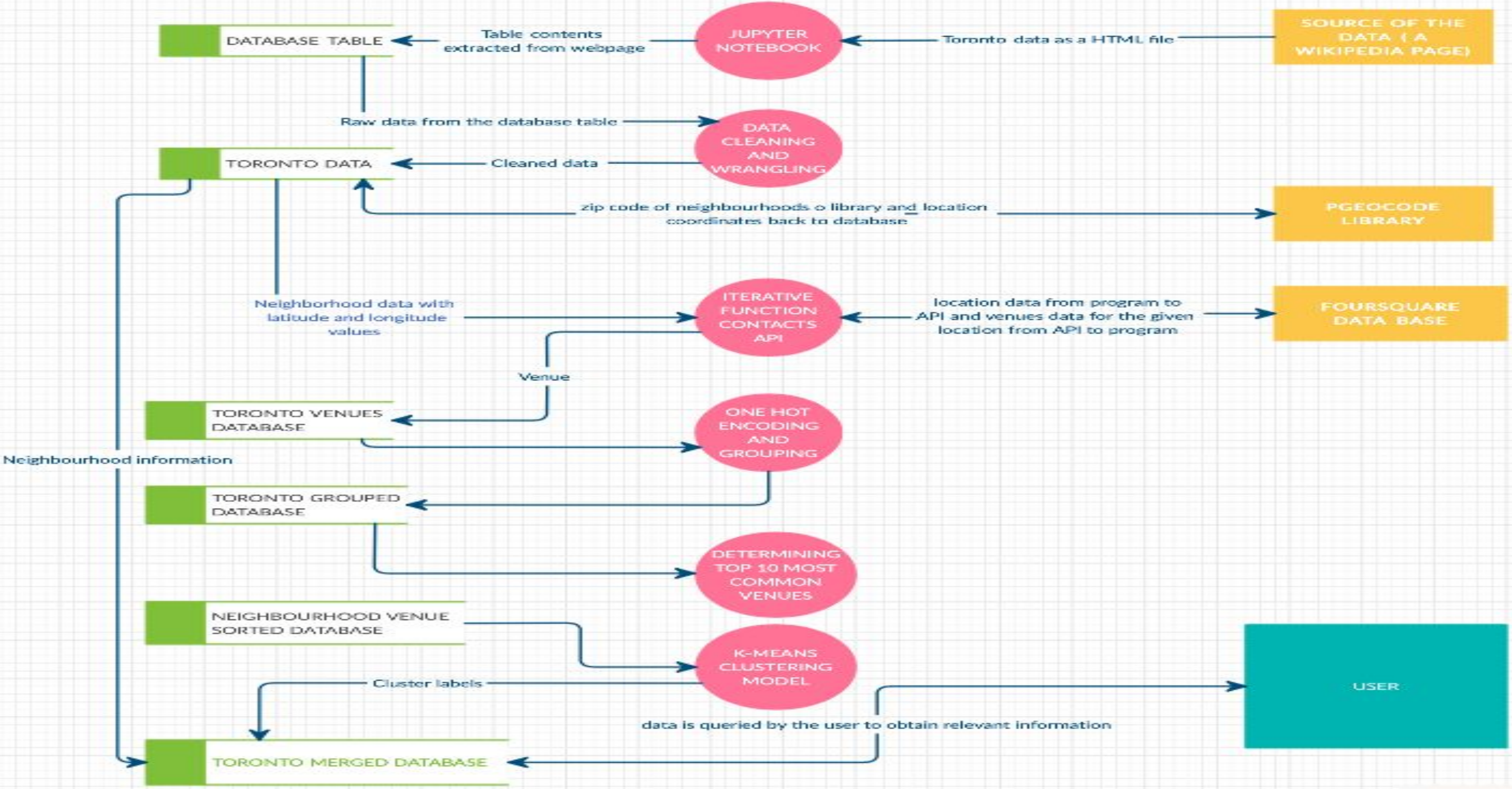
WATSON STUDIO

An online integrated environment provided via IBM cloud that helps to centrally store all the data relevant to ones data science projects.

JUPYTER LABS

A web based integrated environment to created jupyter notebooks which are interactive documents used primarily for data analytics and research projects

DATA FLOW DIAGRAM



DATABASE TABLE

Table contents extracted from webpage

JUPYTER NOTEBOOK

Toronto data as a HTML file

SOURCE OF THE DATA (A WIKIPEDIA PAGE)

Raw data from the database table

DATA CLEANING AND WRANGLING

TORONTO DATA

Cleaned data

zip code of neighbourhoods o library and location coordinates back to database

PGEOCODE LIBRARY

Neighborhood data with latitude and longitude values

ITERATIVE FUNCTION CONTACTS API

location data from program to API and venues data for the given location from API to program

FOURSQUARE DATA BASE

Venue

TORONTO VENUES DATABASE

ONE HOT ENCODING AND GROUPING

Neighbourhood information

TORONTO GROUPED DATABASE

DETERMINING TOP 10 MOST COMMON VENUES

NEIGHBOURHOOD VENUE SORTED DATABASE

K-MEANS CLUSTERING MODEL

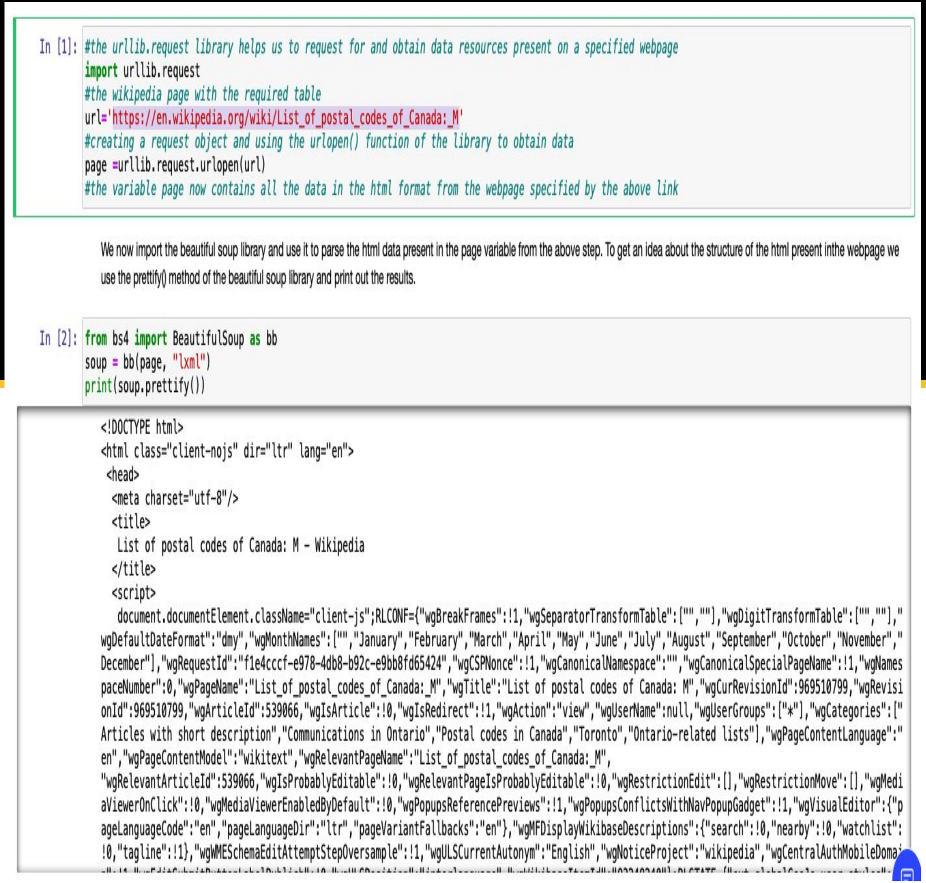
Cluster labels

USER

data is queried by the user to obtain relevant information

TORONTO MERGED DATABASE

CODE AND SNAPSHOTS

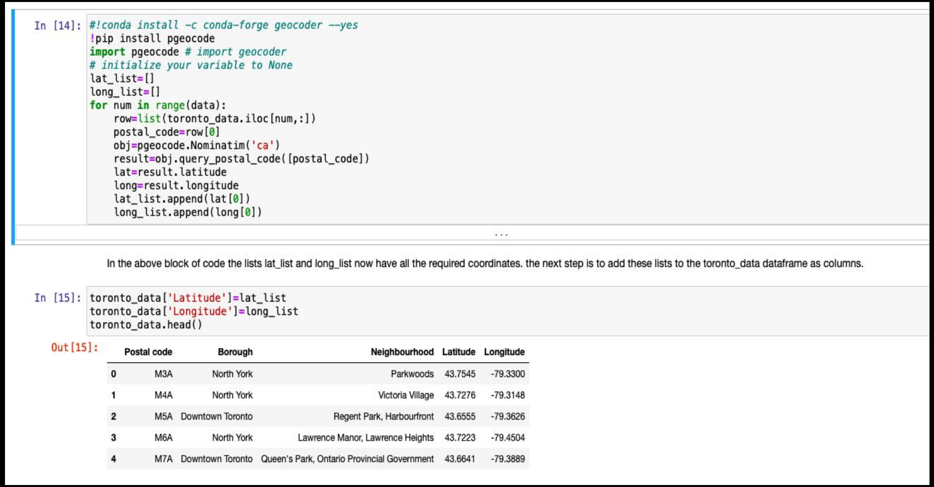


```
In [1]: #the urllib.request library helps us to request for and obtain data resources present on a specified webpage
        import urllib.request
        #the wikipedia page with the required table
        url='https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M'
        #creating a request object and using the urlopen() function of the library to obtain data
        page =urllib.request.urlopen(url)
        #the variable page now contains all the data in the html format from the webpage specified by the above link
```

We now import the beautiful soup library and use it to parse the html data present in the page variable from the above step. To get an idea about the structure of the html present inthe webpage we use the prettify() method of the beautiful soup library and print out the results.

```
In [2]: from bs4 import BeautifulSoup as bb
        soup = bb(page, "lxml")
        print(soup.prettify())
```

```
<!DOCTYPE html>
<html class="client-nojs" dir="ltr" lang="en">
 <head>
  <meta charset="utf-8"/>
  <title>
   List of postal codes of Canada: M - Wikipedia
  </title>
  <script>
   document.documentElement.className="client-js";RLCONF={"wgBreakFrames":!1,"wgSeparatorTransformTable":["",""],"wgDigitTransformTable":["",""],"
wgDefaultDateFormat":"dmy","wgMonthNames":["","January","February","March","April","May","June","July","August","September","October","November","
December"],"wgRequestId":"f1e4cccf-e978-4db8-b92c-e9bb8fd65424","wgCSPNonce":!1,"wgCanonicalNamespace":"","wgCanonicalSpecialPageName":!1,"wgNames
paceNumber":0,"wgPageName":"List_of_postal_codes_of_Canada:_M","wgTitle":"List of postal codes of Canada: M","wgCurRevisionId":969510799,"wgRevisi
onId":969510799,"wgArticleId":539066,"wgIsArticle":!0,"wgIsRedirect":!1,"wgAction":"view","wgUserName":null,"wgUserGroups":["*"],"wgCategories":["
Articles with short description","Communications in Ontario","Postal codes in Canada","Toronto","Ontario-related lists"],"wgPageContentLanguage":"
en","wgPageContentModel":"wikitext","wgRelevantPageName":"List_of_postal_codes_of_Canada:_M",
"wgRelevantArticleId":539066,"wgIsProbablyEditable":!0,"wgRelevantPageIsProbablyEditable":!0,"wgRestrictionEdit":[],"wgRestrictionMove":[],"wgMedi
aViewerOnClick":!0,"wgMediaViewerEnabledByDefault":!0,"wgPopupsReferencePreviews":!1,"wgPopupsConflictsWithNavPopupGadget":!1,"wgVisualEditor":{"p
ageLanguageCode":"en","pageLanguageDir":"ltr","pageVariantFallbacks":"en"},"wgMFDisplayWikibaseDescriptions":{"search":!0,"nearby":!0,"watchlist":
!0,"tagline":!1},"wgWMESchemaEditAttemptStepOversample":!1,"wgULSCurrentAutonym":"English","wgNoticeProject":"wikipedia","wgCentralAuthMobileDomai
```



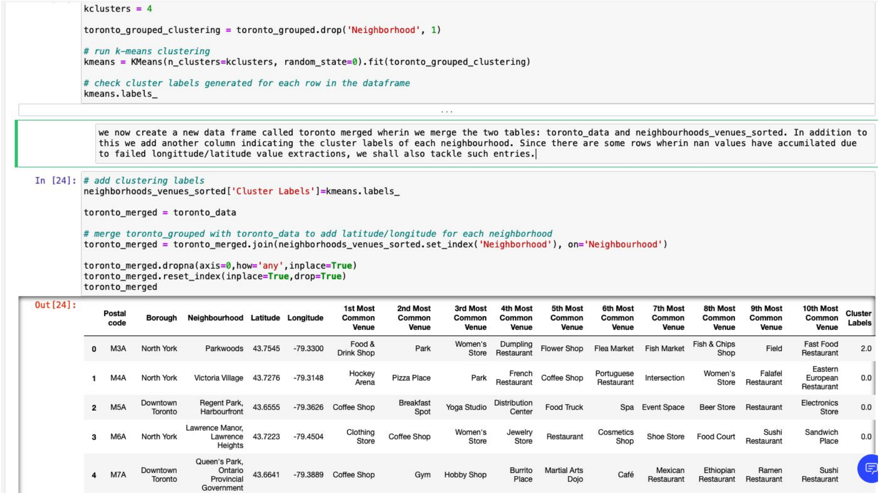
```
In [14]: #!conda install -c conda-forge geocoder --yes
         !pip install pgeocode
         import pgeocode # import geocoder
         # initialize your variable to None
         lat_list=[]
         long_list=[]
         for num in range(data):
             row=list(toronto_data.iloc[num,:])
             postal_code=row[0]
             obj=pgeocode.Nominatim('ca')
             result=obj.query_postal_code([postal_code])
             lat=result.latitude
             long=result.longitude
             lat_list.append(lat[0])
             long_list.append(long[0])
```

In the above block of code the lists lat_list and long_list now have all the required coordinates. the next step is to add these lists to the toronto_data dataframe as columns.

```
In [15]: toronto_data['Latitude']=lat_list
         toronto_data['Longitude']=long_list
         toronto_data.head()
```

Out[15]:

	Postal code	Borough	Neighbourhood	Latitude	Longitude
0	M3A	North York	Parkwoods	43.7545	-79.3300
1	M4A	North York	Victoria Village	43.7276	-79.3148
2	M5A	Downtown Toronto	Regent Park, Harbourfront	43.6555	-79.3626
3	M6A	North York	Lawrence Manor, Lawrence Heights	43.7223	-79.4504
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.6641	-79.3889



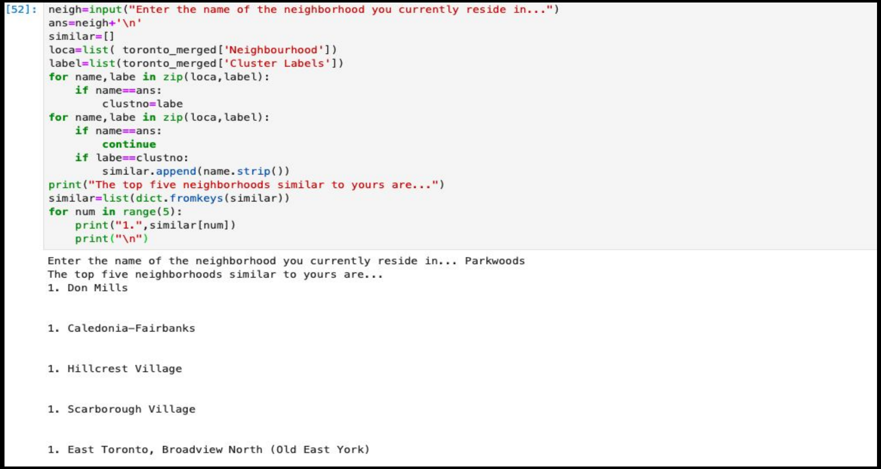
```
kclusters = 4

toronto_grouped_clustering = toronto_grouped.drop('Neighborhood', 1)

# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(toronto_grouped_clustering)

# check cluster labels generated for each row in the dataframe
kmeans.labels_
```

we now create a new data frame called toronto merged wherin we merge the two tables: toronto_data and neighbourhoods_venues_sorted. In addition to this we add another column indicating the cluster labels of each neighbourhood. Since there are some rows wherin nan values have accumilated due to failed longittude/latitude value extractions, we shall also tackle such entries.

```
In [24]: # add clustering labels
         neighborhoods_venues_sorted['Cluster Labels']=kmeans.labels_

         toronto_merged = toronto_data

         # merge toronto_grouped with toronto_data to add latitude/longitude for each neighborhood
         toronto_merged = toronto_merged.join(neighborhoods_venues_sorted.set_index('Neighborhood'), on='Neighbourhood')

         toronto_merged.dropna(axis=0,how='any',inplace=True)
         toronto_merged.reset_index(inplace=True,drop=True)
         toronto_merged
```

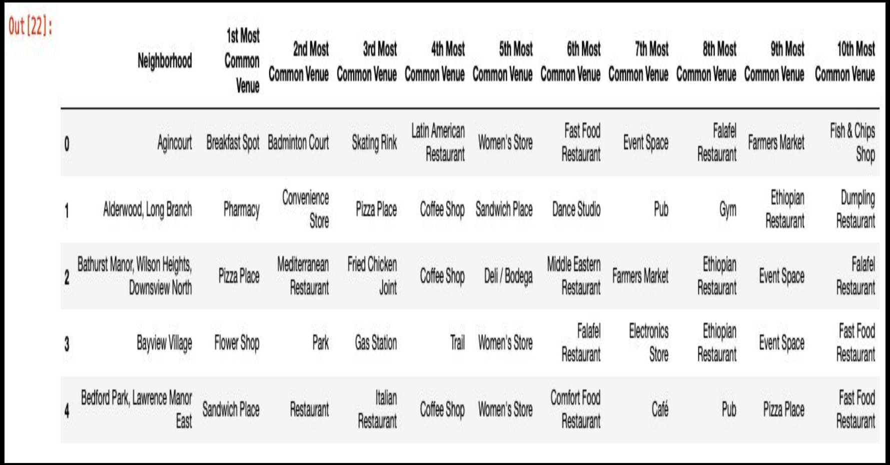
Out[24]:

	Postal code	Borough	Neighbourhood	Latitude	Longitude	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	10th Most Common Venue	Cluster Labels
0	M3A	North York	Parkwoods	43.7545	-79.3300	Food & Drink Shop	Park	Women's Store	Dumpling Restaurant	Flower Shop	Flea Market	Fish Market	Fish & Chips Shop	Field	Fast Food Restaurant	2.0
1	M4A	North York	Victoria Village	43.7276	-79.3148	Hockey Arena	Pizza Place	Park	French Restaurant	Coffee Shop	Portuguese Restaurant	Intersection	Women's Store	Falafel Restaurant	Eastern European Restaurant	0.0
2	M5A	Downtown Toronto	Regent Park, Harbourfront	43.6555	-79.3626	Coffee Shop	Breakfast Spot	Yoga Studio	Distribution Center	Food Truck	Spa	Event Space	Beer Store	Restaurant	Electronics Store	0.0
3	M6A	North York	Lawrence Manor, Lawrence Heights	43.7223	-79.4504	Clothing Store	Coffee Shop	Women's Store	Jewelry Store	Restaurant	Cosmetics Shop	Shoe Store	Food Court	Sushi Restaurant	Sandwich Place	0.0
4	M7A	Downtown Toronto	Queen's Park, Ontario Provincial Government	43.6641	-79.3889	Coffee Shop	Gym	Hobby Shop	Burrito Place	Martial Arts Dojo	Café	Mexican Restaurant	Ethiopian Restaurant	Ramen Restaurant	Sushi Restaurant	



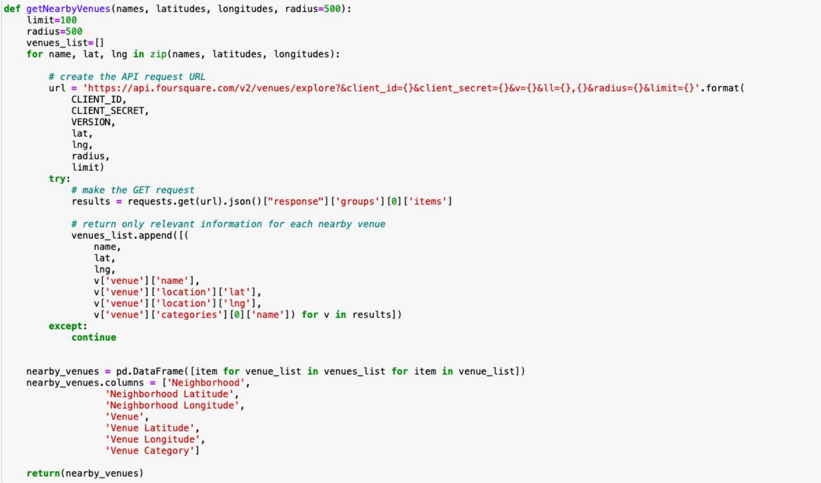
```
[52]: neigh=input("Enter the name of the neighborhood you currently reside in...")
      ans=neigh+'\n'
      similar=[]
      loca=list( toronto_merged['Neighbourhood'])
      label=list(toronto_merged['Cluster Labels'])
      for name,labe in zip(loca,label):
          if name==ans:
              clustno=labe
      for name,labe in zip(loca,label):
          if name==ans:
              continue
          if labe==clustno:
              similar.append(name.strip())
      print("The top five neighborhoods similar to yours are...")
      similar=list(dict.fromkeys(similar))
      for num in range(5):
          print("1.",similar[num])
          print("\n")
```

```
Enter the name of the neighborhood you currently reside in... Parkwoods
The top five neighborhoods similar to yours are...
1. Don Mills


1. Caledonia-Fairbanks


1. Hillcrest Village


1. Scarborough Village


1. East Toronto, Broadview North (Old East York)
```



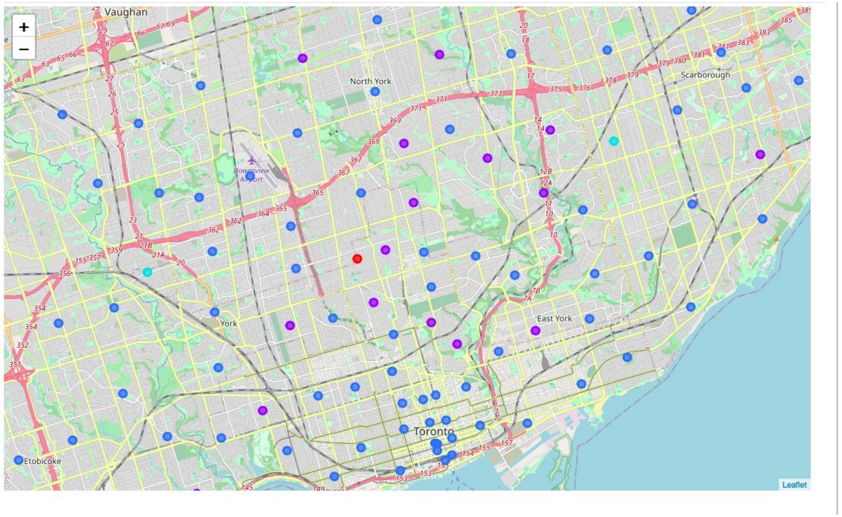
Out[22]:

	Neighborhood	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	10th Most Common Venue
0	Agincourt	Breakfast Spot	Badminton Court	Skating Rink	Latin American Restaurant	Women's Store	Fast Food Restaurant	Event Space	Falafel Restaurant	Farmers Market	Fish & Chips Shop
1	Alderwood, Long Branch	Pharmacy	Convenience Store	Pizza Place	Coffee Shop	Sandwich Place	Dance Studio	Pub	Gym	Ethiopian Restaurant	Dumpling Restaurant
2	Bathurst Manor, Wilson Heights, Downsview North	Pizza Place	Mediterranean Restaurant	Fried Chicken Joint	Coffee Shop	Deli / Bodega	Middle Eastern Restaurant	Farmers Market	Ethiopian Restaurant	Event Space	Falafel Restaurant
3	Bayview Village	Flower Shop	Park	Gas Station	Trail	Women's Store	Falafel Restaurant	Electronics Store	Ethiopian Restaurant	Event Space	Fast Food Restaurant
4	Bedford Park, Lawrence Manor East	Sandwich Place	Restaurant	Italian Restaurant	Coffee Shop	Women's Store	Comfort Food Restaurant	Café	Pub	Pizza Place	Fast Food Restaurant



```
def getNearbyVenues(names, latitudes, longitudes, radius=500):
    limit=100
    radius=500
    venues_list=[]
    for name, lat, lng in zip(names, latitudes, longitudes):

        # create the API request URL
        url = 'https://api.foursquare.com/v2/venues/explore?&client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}'.format(
            CLIENT_ID,
            CLIENT_SECRET,
            VERSION,
            lat,
            lng,
            radius,
            limit)
        try:
            # make the GET request
            results = requests.get(url).json()["response"]['groups'][0]['items']

            # return only relevant information for each nearby venue
            venues_list.append([(
                name,
                lat,
                lng,
                v['venue']['name'],
                v['venue']['location']['lat'],
                v['venue']['location']['lng'],
                v['venue']['categories'][0]['name']) for v in results])
        except:
            continue


    nearby_venues = pd.DataFrame([item for venue_list in venues_list for item in venue_list])
    nearby_venues.columns = ['Neighborhood',
                  'Neighborhood Latitude',
                  'Neighborhood Longitude',
                  'Venue',
                  'Venue Latitude',
                  'Venue Longitude',
                  'Venue Category']

    return(nearby_venues)
```

[31]:



PROJECT DEMO

ANALYSIS

AND

DISCUSSION

SWOT ANALYSIS

STRENGTHS

A strong and novel idea that would help ease the troubles of movers.

WEAKNESSES

In case the user do not like their current neighborhood, no utility to search for alternative venues exists

OPPORTUNITY

Further enhancements can be made in the GUI of the product. It can also be integrated into existing rental portals to enhance user experience

THREATS

The project is completely data dependent and thus any inconsistencies in the information database can cause the system to give unreliable results

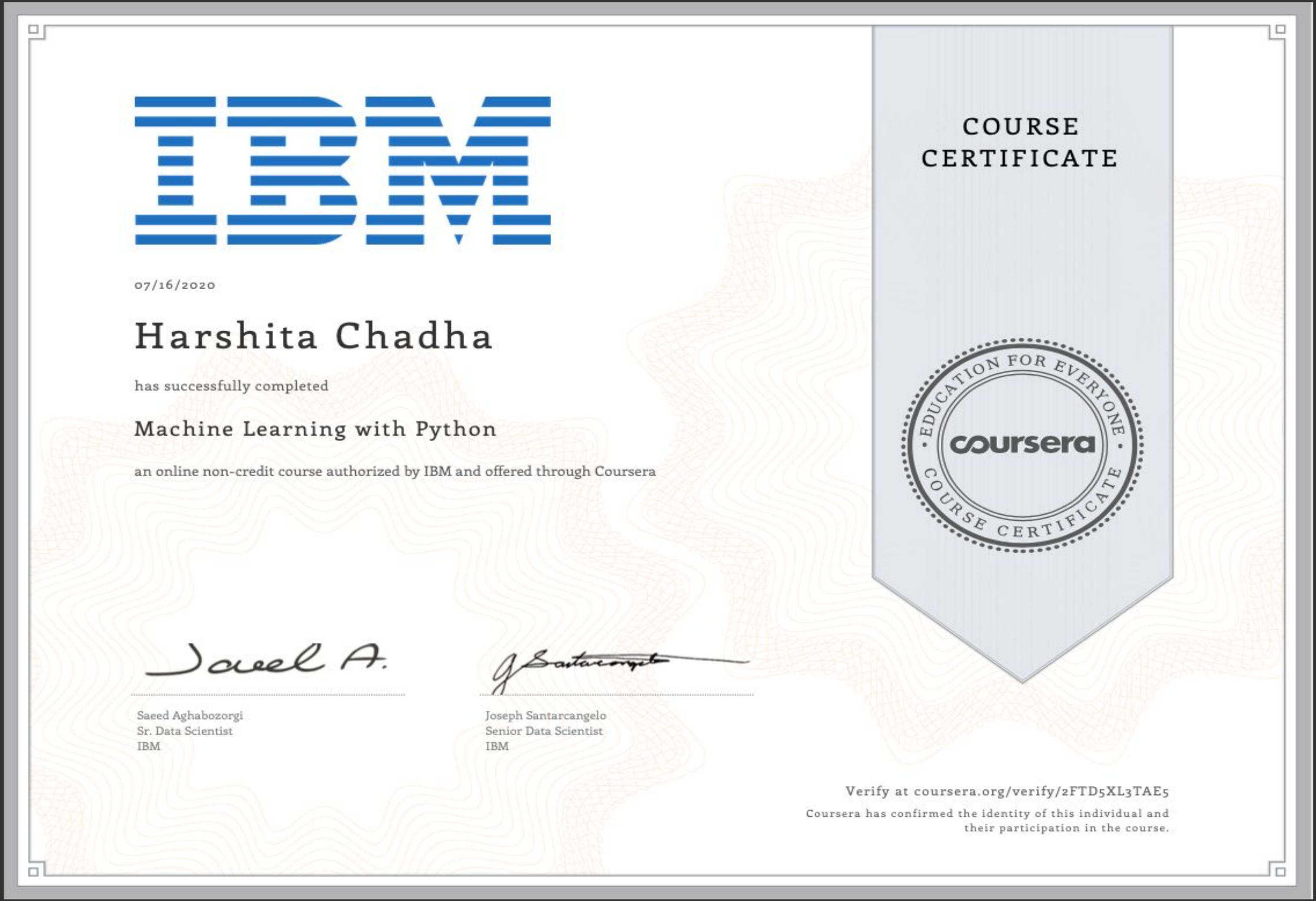
RESULTS

- The project helped gain another new skill which is to interact with the foursquare API and manage and manipulate the results returned by it to draw meaningful conclusions.
- At the end of the development cycle, a highly informative database was created which was flexible enough to be updated to present real-time, up to date information and could be easily queried to extract useful information.
- Thus, as a result of the training, the individual was able to master industry-relevant skills that have a high value in the tech market presently.
- What is more, is that practical hands-on experience in solving real-life problems using data science was also gained which helped build candidate portfolio and further career aspirations.

Future scope & CONCLUSIONS

The project was designed to solve the problem of people who faced the prospect of a house move.

- Using the machine learning utility designed, they can just enter the name of the neighborhood that they currently live in and a list of top 5 most similar neighborhoods shall be output by the program.
- Although quite sophisticated in itself, the project has great potential to be further enhanced and elevated to a state-of-the-art level.
- The final database that was created during the project development can be enhanced to include venue data and cluster labels for other locations of the world to make it more inclusive.
- Furthermore, some self-dependent recommendation system principles may also be applied to the data set to create a most popular neighborhood recommendation mechanism, that could consider the user’s profile and the places they have been to and the rating that they gave those places to make final recommendations.
- What is more, is that a web or mobile application with an interactive Graphical user interface may be created for the program which may make the utility created more easily accessible by the general public.
- The recommendation for improvement made above can be considered as the future scope of the project as they focus on enhancing its capabilities further to make it more inclusive and capable.



IBM

07/16/2020

Harshita Chadha

has successfully completed

Machine Learning with Python

an online non-credit course authorized by IBM and offered through Coursera

Saeed Aghabozorgi
Sr. Data Scientist
IBM

Joseph Santarcangelo
Senior Data Scientist
IBM

COURSE CERTIFICATE

EDUCATION FOR EVERYONE
coursera
COURSE CERTIFICATE

Verify at coursera.org/verify/2FTD5XL3TAE5

Coursera has confirmed the identity of this individual and their participation in the course.

THANK YOU

Do you have any questions?
Please feel free to visit the link below and explore my GitHub repository with all the code and related data objects.

https://github.com/hersheychadha/Coursera_capstone.git