IBM Data Science Professional Certificate: Capstone Project Final Report

Tourism Recommendation Program

Alexander Derek Veale

21 April 2021

1. Introduction

1.1 Background

International travel and tourism are becoming increasingly popular and ever more affordable. Apart from the major impact that covid-19 has had on the sector, the sector is expected to grow year on year. Tourism companies such as 'On the Go Tours' and 'Topdeck' spend lots of time, effort and money trying to attract customers to sell them travel deals and packages.

Different people travel for different reasons and enjoy different types of holidays. Some people enjoy relaxing on a beach, others enjoy learning about history and/or culture and some even travel just for the nightlife. According to my research the following main holiday types can be discerned:

- Wildlife and nature holidays
- Beach holidays
- Sport and/or adventure holidays
- Cultural and/or historical holidays
- City and/or nightlife holidays

1.2 Problem Statement

This report sets out to create a program that can recommend travel destinations to a travel company's new or existing customers based on other cities that they have really enjoyed visiting. I will make use of KMeans clustering to cluster the top 100 travel destinations of 2019 into 5 clusters, each representing 1 of the 5 holidays types identified earlier. I will explore the cities and retrieve venue data within a 7.5km radius. I will then use this data to perform the clustering of the cities. A new or existing customer will then be able to tell the sales representative who is using the program a city that they really enjoyed traveling to and the program will take that information and make recommendations for future travel destinations.

1.3 Interest

Companies and/or investors would be interested in this application, because it would allow them to make better holiday recommendations to their clients, which would improve their customers' experience and increase the chances of the customer returning to their company the next time they are thinking about traveling. It would also allow them to target their advertising for particular destinations to particular clients, especially when combined with a database of clients' pervious holidays and their respective ratings.

2. Data Collection and Cleaning

2.1 Data Source

In order to create the desired application, I will need a list of the top 100 travel destinations. I have decided to use 2019 data, as that will provide a fairer/less biased view, since covid-19 would have had a major impact on the 2020 data and would not reflect the reality of holiday travel, but rather it would only reflect emergency travel.

'Euromonitor' has a downloadable pdf containing a list of the top 100 travel destinations of 2019. I downloaded the pdf and used an online format converter to convert the file from a pdf document into html file, which I then downloaded too. I then loaded the file in python and created a BeautifulSoup object of the html file and extracted the city, country and rank into a data frame.

Next, I needed to retrieve the coordinates of each of the cities. I used the geocoder library to achieve this and added the latitude and longitude for each city to my data frame. However, since the geocoder library would not function properly, I downloaded my data frame as a csv file, then loaded it into the 'Skill Network Labs' environment where I was able to use the 'geopy' library to retrieve the coordinate data, I then saved the new data in a new csv file and imported that back into my original notebook. (This code can be found in Appendix A)

Finally, I used the Foursquare API to explore the top 100 travel destinations. I used the API's explore function to retrieve venues as well as their categories within 7.5km, limiting my results to 250 venues per city. To do this, I constructed an appropriate url and used the 'requests' and 'json' libraries to retrieve the desired data.

2.2 Data Cleaning

Some of the country names that I retrieved from the html file from Euromonitor were either shorthand or incomplete and I replaced these values with their correct values. For some reason when scraping from my BeautifulSoup object it failed to pull in the data for Berlin, Sydney and Moscow, so I had to create a data frame for those 3 cities manually and merge it with my final data frame of the top 100 cities along with their country and rank.

I removed a lot of venues from my data frame of venues, for a number of reasons. The categories were either a) of little or no concern to an average tourist or b) too vague/ambiguous and would be found in all destinations and/or c) were categories that would have a high frequency and would push down the proportional frequency of more relevant venue categories to tourists.

I also merged multiple similar categories into a single category for both sports venues and nightlife venues. For sports venues, I did this, because a city might for example only have 1 soccer stadium and having that as its own category would mean that's its relative frequency of occurrence out of some 503 unique categories would be almost zero, so I combined all sporting venues into a single category to try and increase their relative frequency of occurrence.

Conversely, for the different nightlife venues there were multiple similar categories with relatively high frequencies of occurrence that were all competing with the other categories, for instance in Dublin, Ireland both 'Bars' and 'Pubs' were in the top 8 most frequently occurring categories of venues and it would make sense to only have a single nightlife category for the other categories to 'compete' with.

2.3 Feature Selection

I couldn't perform clustering on a data frame of venues and their respective categories, I needed to transform this data into a data frame containing the proportion of occurrence of each category in each city. I achieved this by using the pandas 'get_dummies' function on the category column of the venues data frame and then grouping the results by the city column and calculating the mean.

3. Methodology

3.1 Exploratory Data Analysis

Once I had created a data frame of all venues along with their location, category and corresponding cities and I had removed and/or amalgamated all of the venues that were in categories that I did not want or wanted grouped together then I began my analysis.

To begin with, I looked at the head of my data frame as well as some basic statistics about the number of venues in each city by using the 'describe' method and got the following results:

In [156]:
```
city_venues.head()
```

Out[156]:

	city	latitude	longitude	venue	v_lat	v_long	category
0	Hong Kong	22.279328	114.162813	Hong Kong Park Aviary (香港公園鸚鳥園)	22.277140	114.161399	Zoo
1	Hong Kong	22.279328	114.162813	Hong Kong Park (香港公園)	22.277700	114.161854	Park
8	Hong Kong	22.279328	114.162813	Pure Fitness	22.278475	114.161363	Gym / Fitness Center
9	Hong Kong	22.279328	114.162813	Mott 32 (卅二公館)	22.280286	114.159080	Dim Sum Restaurant
12	Hong Kong	22.279328	114.162813	Pacific Place (太古廣場)	22.277696	114.165048	Shopping Mall

Exploratory Data Analysis

First, I explore the statistics of the number of venues in each city, followed by the total number of unique categories in the dataframe

In [157]:
```
city_venues.groupby('city')['venue'].count().describe()
```

Out[157]:
```
count    100.000000
mean      54.700000
std       16.921984
min        1.000000
25%       49.000000
50%       59.000000
75%       65.000000
max       78.000000
Name: venue, dtype: float64
```

As can be seen above, there is a count of 100 – corresponding to the top 100 travel destinations in the original data frame and; on average each city had around 55 venues of interest within the 7.5km radius, with the city with the smallest number of venues of interest only being 1 venue and the city with the most venues of interest having 78 venues. I also found that the number of unique categories left in the data frame after I performed data cleaning was 325 different venue categories.

I then created a one-hot encoded data frame of the venue categories with 325 columns, each representing 1 of the unique venue categories as well as an extra column for each row to contain the city which the row represents. I then grouped this new data frame by city and calculated the mean, thereby creating a frequency table of the proportion of venues each category represented in each city. I then determined the top 8 most frequent venues in each city, below is a small snippet of the result:

```
~~~~Agra~~~~
                          venue  frequency
0                Indian Restaurant      0.300
1                    Historic Site      0.167
2  Multicuisine Indian Restaurant      0.100
3                           Market      0.067
4                      Pizza Place      0.067
5                     Train Station      0.033
6              Fried Chicken Joint      0.033
7               Italian Restaurant      0.033

~~~~Amsterdam~~~~
                venue  frequency
0           Nightlife      0.137
1         Pizza Place      0.078
2               Plaza      0.078
3       Deli / Bodega      0.059
4    Organic Grocery       0.039
```

And I then created a data frame with the 10 most frequent venues in each city, a snippet of which can be seen below:

	city	1	2	3	4	5	6	7	8	9	10
0	Abu Dhabi	Middle Eastern Restaurant	Park	Beach	Pizza Place	Turkish Restaurant	Juice Bar	Lounge	Sports Venue	Spa	Shopping Mall
1	Agra	Indian Restaurant	Historic Site	Multicuisine Indian Restaurant	Market	Pizza Place	Garden	Clothing Store	South Indian Restaurant	Fried Chicken Joint	Indian Sweet Shop
2	Amsterdam	Nightlife	Plaza	Pizza Place	Deli / Bodega	Art Museum	French Restaurant	Park	Canal	Organic Grocery	Lebanese Restaurant
3	Antalya	Gym / Fitness Center	Nightlife	Park	Scenic Lookout	Museum	Theater	Mosque	Mediterranean Restaurant	Clothing Store	Department Store
4	Athens	Nightlife	Historic Site	Meze Restaurant	History Museum	Boutique	Gourmet Shop	Souvlaki Shop	Pedestrian Plaza	Park	Falafel Restaurant
5	Auckland	Nightlife	Park	Japanese Restaurant	Italian Restaurant	Plaza	Sushi Restaurant	Asian Restaurant	Vietnamese Restaurant	Gourmet Shop	Mountain
6	Bangalore	Indian Restaurant	Nightlife	Brewery	Lounge	Boutique	Shopping Mall	Park	Italian Restaurant	Asian Restaurant	Sushi Restaurant

3.2 Data Modeling

I used KMeans clustering to model my data. I chose to use 5 clusters to group my data, because of my research that said that holidays could be broadly grouped into 5 different types. I imported the model and trained it on the frequency table that I created earlier. I did not train my model using the data frame only containing the 10 most frequent venue categories in each city, because then the model could be missing out on valuable data, and because KMeans is a typically fast algorithm, I could afford to use all 325 unique categories in my model.

I then added the cluster labels fit by the model to a data frame containing the city name, country, rank (in top 100 travel destinations), latitude, longitude and the top 10 most frequent venues in the city. A portion of the result can be seen below:

rank	city	country	latitude	longitude	cluster label	1	2	3	4	5	6	7	8
1	Hong Kong	Hong Kong, China	22.279328	114.162813	2	Nightlife	Japanese Restaurant	Thai Restaurant	Italian Restaurant	Yoga Studio	Park	Gym / Fitness Center	Scenic Lookout
2	Bangkok	Thailand	13.754424	100.493040	0	Nightlife	Palace	Thai Restaurant	Noodle House	Seafood Restaurant	Shopping Mall	Park	Asian Restaurant
3	London	United Kingdom	51.507322	-0.127647	2	Nightlife	Department Store	Park	Garden	Theater	Art Gallery	Art Museum	Pedestrian Plaza
4	Macau	Macau, China	22.189945	113.538045	2	Portuguese Restaurant	Nightlife	Cantonese Restaurant	Chinese Restaurant	Italian Restaurant	Historic Site	Lounge	French Restaurant
5	Singapore	Singapore	1.357107	103.819499	0	Park	Italian Restaurant	Zoo Exhibit	Trail	Indian Restaurant	Nature Preserve	Scenic Lookout	Japanese Restaurant
6	Paris	France	48.856697	2.351462	2	Plaza	French Restaurant	Fountain	Art Museum	Garden	Nightlife	Italian Restaurant	Creperie
7	Dubai	United Arab Emirates	25.265347	55.292491	0	Nightlife	Middle Eastern	Beach	French Restaurant	Historic Site	Park	Sports Venue	Shopping Mall

I then visualized the resulting clusters on a world map using the folium library. The different coloured markers on the map represent the different clusters. The map can be seen below:

4. Results

4.1 Examining the Different Clusters

Then I inspected the similarities between the types of venue categories in each city found in each cluster, using the following results:

Results

Here I explore the similarities in the types of venue categories that are found in cities within the same clusters

In [170]:
```
#Cities in cluster 0
cities[cities['cluster label']==0]
```

city	country	latitude	longitude	cluster label	1	2	3	4	5	6	7	8	9	
Bangkok	Thailand	13.754424	100.493040	0	Nightlife	Palace	Thai Restaurant	Noodle House	Seafood Restaurant	Shopping Mall	Park	Asian Restaurant	Japanese Restaurant	
Singapore	Singapore	1.357107	103.819499	0	Park	Italian Restaurant	Zoo Exhibit	Trail	Indian Restaurant	Nature Preserve	Scenic Lookout	Japanese Restaurant	Thai Restaurant	
Dubai	United Arab Emirates	25.265347	55.292491	0	Nightlife	Middle Eastern Restaurant	Beach	French Restaurant	Historic Site	Park	Sports Venue	Shopping Mall	Seafood Restaurant	R
New York City	USA	40.712728	-74.006015	0	Park	Gourmet Shop	Thai Restaurant	Italian Restaurant	Scenic Lookout	Pier	Nightlife	Speakeasy	Garden	R
Istanbul	Turkey	41.009633	28.965165	0	Mosque	Historic Site	Art Gallery	History Museum	Nightlife	Sports Venue	Plaza	Shrine	Turkish Restaurant	Ar
Shenzhen	China	22.555454	114.054330	0	Nightlife	Shopping Mall	Chinese Restaurant	Park	Lounge	Hong Kong Restaurant	Electronics Store	BBQ Joint	New American Restaurant	R
Phuket	Thailand	7.936602	98.352929	0	Thai	Sports	Dim Sum	Department	Seafood	Noodle	Shrine	Italian	Som Tum	

In [171]:
```
#Cities in cluster 1
cities[cities['cluster label']==1]
```

Out[171]:

	rank	city	country	latitude	longitude	cluster label	1	2	3	4	5	6	7	8	9	10
79	80	Guilin	China	25.238317	110.175225	1	Clothing Store	Hindu Temple	Garden Center	Gastropub	General Entertainment	German Restaurant	Gift Shop	Gluten-free Restaurant	Gourmet Shop	Government Building

In [172]:
```
#Cities in cluster 2
cities[cities['cluster label']==2]
```

Out[172]:

	rank	city	country	latitude	longitude	cluster label	1	2	3	4	5	6	7
0	1	Hong Kong	Hong Kong, China	22.279328	114.162813	2	Nightlife	Japanese Restaurant	Thai Restaurant	Italian Restaurant	Yoga Studio	Park	Gym / Fitness Center
2	3	London	United Kingdom	51.507322	-0.127647	2	Nightlife	Department Store	Park	Garden	Theater	Art Gallery	Art Museum
3	4	Macau	Macau, China	22.189945	113.538045	2	Portuguese Restaurant	Nightlife	Cantonese Restaurant	Chinese Restaurant	Italian Restaurant	Historic Site	Lounge
5	6	Paris	France	48.856697	2.351462	2	Plaza	French Restaurant	Fountain	Art Museum	Garden	Nightlife	Italian Restaurant
8	9	Kuala Lumpur	Malaysia	3.151696	101.694237	2	Nightlife	Malay Restaurant	Shopping Mall	Italian Restaurant	Japanese Restaurant	Boutique	Jewelry Store
11	12	Antalya	Turkey	36.900964	30.695485	2	Gym / Fitness Center	Nightlife	Park	Scenic Lookout	Museum	Theater	Mosque
15	16	Rome	Italy	41.893320	12.482932	2	Plaza	Church	Park	Italian	Historic	Art	Nightlife

In [173]:
```
#Cities in cluster 3
cities[cities['cluster label']==3]
```

Out[173]:

	rank	city	country	latitude	longitude	cluster label	1	2	3	4	5	6	7	8	9	10
96	97	Rhodes	Greece	36.17263	27.919418	3	Grilled Meat Restaurant	Greek Restaurant	Cruise Ship	Dam	Gastropub	General Entertainment	German Restaurant	Gift Shop	Gluten-free Restaurant	Gourmet Shop

4]:

```
#Cities in cluster 4
cities[cities['cluster label']==4]
```

4]:

city	country	latitude	longitude	cluster label	1	2	3	4	5	6	7	8	9	
Delhi	India	28.651718	77.221939	4	Indian Restaurant	Lounge	Monument / Landmark	Nightlife	Market	South Indian Restaurant	Mediterranean Restaurant	Sculpture Garden	History Museum	Res
Mumbai	India	19.075990	72.877393	4	Indian Restaurant	Lounge	Nightlife	Deli / Bodega	Seafood Restaurant	Vegetarian / Vegan Restaurant	Italian Restaurant	Gym / Fitness Center	Spa	(Enterta
Agra	India	27.175255	78.009816	4	Indian Restaurant	Historic Site	Multicuisine Indian Restaurant	Market	Pizza Place	Garden	Clothing Store	South Indian Restaurant	Fried Chicken Joint	Indiar
Chennai	India	13.083694	80.270186	4	Indian Restaurant	Italian Restaurant	Nightlife	Juice Bar	Men's Store	Chinese Restaurant	Middle Eastern Restaurant	Department Store	Jewelry Store	Vege Res
Jaipur	India	26.915458	75.818982	4	Indian Restaurant	Historic Site	Nightlife	Italian Restaurant	Shopping Mall	Sports Venue	Hostel	Pizza Place	Art Museum	

Looking at the results above combined with the map visualization, you can discern that cluster 0 is made up lots of islands and cities located along coastlines. Nightlife, parks, beach, piers/waterfront, scenic outlooks and seafood restaurants are a common theme throughout this cluster's most frequent venue categories. One might be tempted to say that cluster zero represents the 'beach' holiday type.

Cluster 1 only consists of 1 city - Guilin China – and it's most frequent venues seem to be random, but upon a simple google search, one can find that it is a very small city, but it is particularly beautiful, nestled amongst forested hills and mountains. This cluster could represent the 'nature/wildlife' holiday type.

Cluster 2 is made up of mostly capital cities with lots of nightlife, museums, historic sights and entertainment venues. Cluster 2 most probably represents the 'city and/or nightlife' and/or the 'cultural/historical' holiday type.

Cluster 3, like cluster1 only contains 1 city - Rhodes, Greece – which doesn't seem to have any very distinguishable venue categories in it's most frequent categories, which is probably why it has been grouped into a cluster by itself.

In cluster 4, all of the cities are in India and all of the most frequently occurring venue category in every city in the cluster is 'Indian restaurant, followed by lots of other types of restaurants, this cluster was most probably formed due to this relationship and might not be as useful to us.

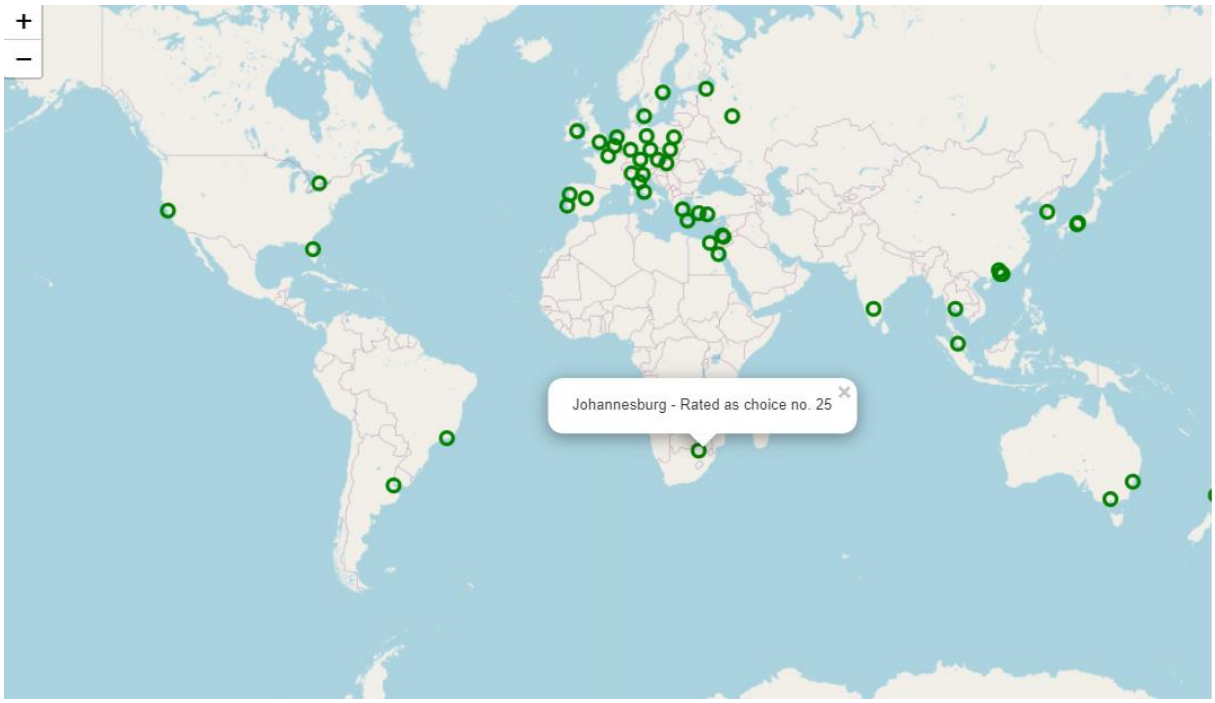
4.2 The Results of a Client Using the Program

Any new/existing client can enter the name of a city that they have really enjoyed travelling to, along with its country. The program with then predict which cluster this city falls into, using the frequency of venue categories found in the city and the model that has been built. Cities that are in the same cluster are then recommended to the user, ranked according to their place in the top 100 cities ranking, the results are presented in a sorted data frame, as well as visually on a map.

Say for example, there is a client who has recently visited Perth, Australia and had a great time and they are interested in travelling again soon. The following recommendation would be made by the model:

rank	city	country	latitude	longitude	cluster label	1	2	3	4	5	6	7	8	
1	Hong Kong	Hong Kong, China	22.279328	114.162813	2	Nightlife	Japanese Restaurant	Thai Restaurant	Italian Restaurant	Yoga Studio	Park	Gym / Fitness Center	Scenic Lookout	Rest
3	London	United Kingdom	51.507322	-0.127647	2	Nightlife	Department Store	Park	Garden	Theater	Art Gallery	Art Museum	Pedestrian Plaza	
4	Macau	Macau, China	22.189945	113.538045	2	Portuguese Restaurant	Nightlife	Cantonese Restaurant	Chinese Restaurant	Italian Restaurant	Historic Site	Lounge	French Restaurant	L
6	Paris	France	48.856697	2.351462	2	Plaza	French Restaurant	Fountain	Art Museum	Garden	Nightlife	Italian Restaurant	Creperie	Histo
9	Kuala Lumpur	Malaysia	3.151696	101.694237	2	Nightlife	Malay Restaurant	Shopping Mall	Italian Restaurant	Japanese Restaurant	Boutique	Jewelry Store	Monument / Landmark	
12	Antalya	Turkey	36.900964	30.695485	2	Gym / Fitness Center	Nightlife	Park	Scenic Lookout	Museum	Theater	Mosque	Mediterranean Restaurant	C
16	Rome	Italy	41.893320	12.482932	2	Plaza	Church	Park	Italian	Historic	Art	Nightlife	Monument /	Fr

And visually with popups containing the city name and ranking in the cluster, rather than it's overall rank in the top 100 cities, as follows:



5. Discussion and Recommendations

The clustering had somewhat of the desired result, however, better results could definitely be achieved. The problem was that the data set that was used is biased. The problem with the dataset used (Top 100 Travel destinations) is that the rankings in this dataset were determined by the number of flights arriving in each city.

This means that smaller city that don't have big, international airports won't make it onto this list. For example, The Kruger National Park is one of South Africa's most popular tourist destinations, but there is no airport there, people first have to fly to Johannesburg and then commute 5-6 hours by car/bus to get there and therefore would never make it onto the lists.

This means that our dataset would not be completely representative of all the holiday types, particularly the 'nature/wildlife' and the 'sport/adventure' types, since these places are usually remote. It is unlikely that you will find nature reserves of ski resorts in/near large cities with international airports and thus, our dataset lacks the destinations that would have the distinguishable venue categories that would be able create these clusters.

Therefore, in order to improve the results of the model, one would need to increase the size of the initial dataset and make sure to include more remote popular destinations, especially ones that might fall into the aforementioned holiday types. The model lacks data from African and South American cities in particular too, which should be rectified.

Another problem that could be solved is the restaurant dilemma. The problem with restaurants is that there are a lot of them and therefore their proportional frequency compared to other venue categories is much higher and thus, their effect on the model is much greater. Due to globalization, there are lots of restaurants of all kinds and flavors all over the world. We would obviously want to include restaurants that fall under local/regional/traditional cuisine, but exclude other cuisine.

For example, in the 4th cluster 60% of the top 10 most frequently occurring venues fall under different restaurant categories. Since all of the countries in this cluster are in India, we would want to keep the number 1 venue category in each city as 'Indian Restaurant', but 'Italian Restaurant' and 'Pizza Place' feature heavily too and we would probably want to exclude these. It becomes hard, however, because we can't just remove all venues in the 'Italian Restaurant' category, because if they were in an Italian city, we would want them to feature. Maybe the solution is to leave out restaurants from the model all together?

The recommended results could be further refined/ranked to meet the clients' desired venue category characteristics, rather than just using the order of their ranking in the top 100 travel destinations dataset.

6. Conclusion

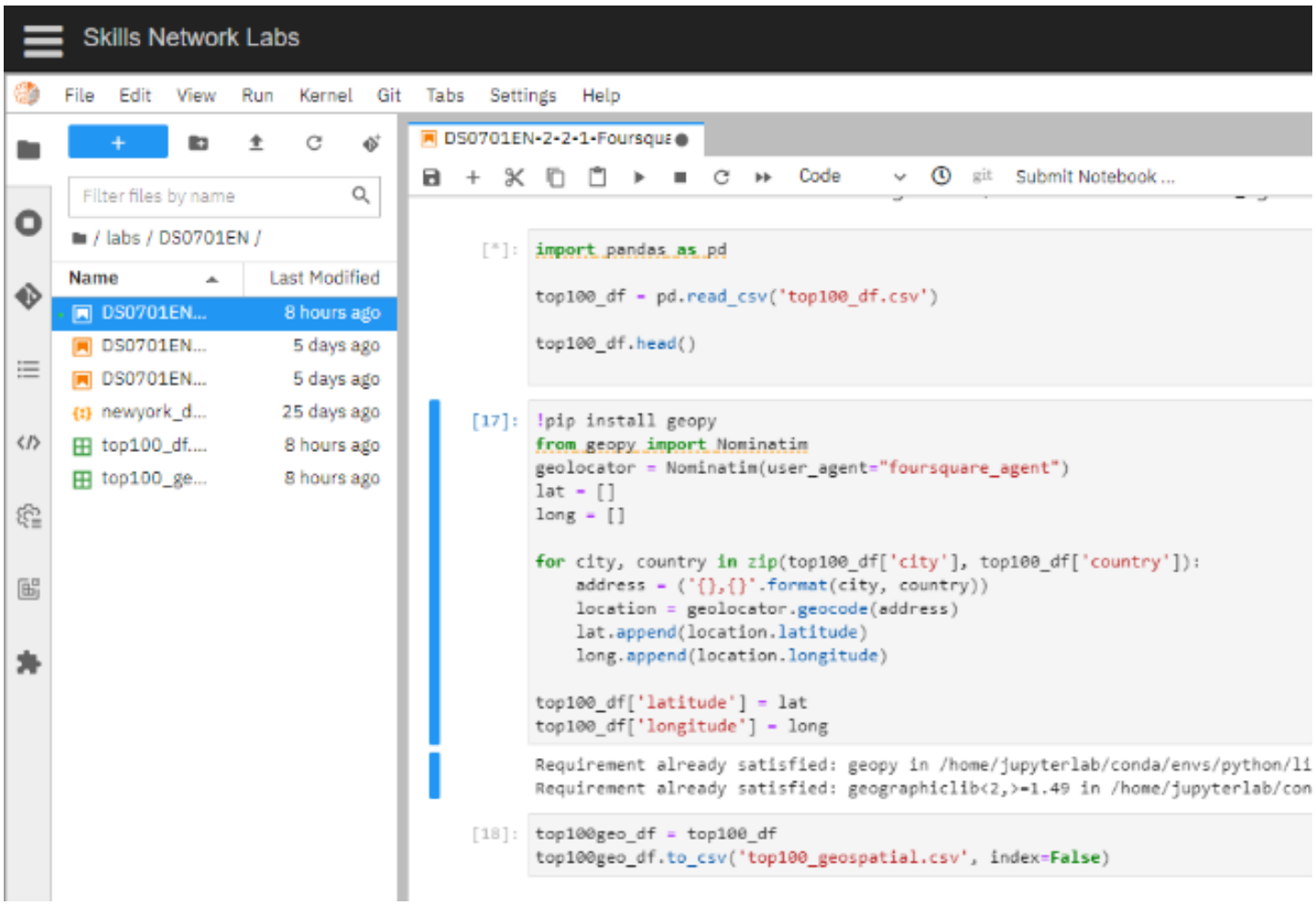
I was able to successfully create a program that clustered the top 100 most popular travel destinations in the world into 5 clusters based on venue category data retrieved from the Foursquare API.

I was also able to make recommendations about future prospective travel destinations based off unseen cities by predicting the city's cluster using the KMeans model.

A link to my Jupyter Notebook on my Github repository can be found in Appendix B.

Appendix A

Below is the code that I used in the Skills Network Labs environment to get the city coordinate data using the geopy library.



Appendix B

Below is a link to my Capstone Project Notebook on Github.

https://github.com/ADV87/Coursera_Capstone/blob/22306ffb3c39f73107cc0414f716ac427210f011/Capstone%20Project.ipynb