IBM Data Science Professional Certificate: Capstone Project

Tourism Recommendation Program

Alexander Derek Veale

21 April 2021

# Introduction

## 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

## 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.

## 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.

# Data Collection and Cleaning

## 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.

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.

## 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 their city and calculating the mean.

# Methodology

## Exploratory Data Analysis

## 3.2 Data Modeling

# Results

# Discussion and Recommendations

# Conclusion

# Appendix