

'Date Night Planner'

Final IBM data science project

Introduction

The idea that I am looking to build out in Python is a tool that gives the user ideas on where best to have a date night in London, based on an entertainment venue, bar and restaurant. For the bar and restaurant, it will provide the price group of each one as well as the rating; allowing the user to make an informed decision on where to go for the evening.

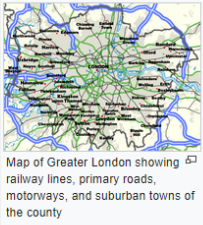
The target audience of this tool are those that are possibly less confident in dating and need some help planning their evening. It can also be used for people that are more confident in dating but are less well organised, again meeting their needs by planning the evening. Therefore it has a broad audience that could make use of it.

Data

I will be starting off the process using a Wikipedia page providing a table of London boroughs and their post codes. This is the most appropriate way to map out the various regions of London and also allow the user to pinpoint more effectively rough regions that they might be comfortable going on their data (London is huge so this is important).

en.wikipedia.org/wiki/List_of_areas_of_London

The northernmost settlement is **Crews Hill**, Enfield – westernmost is **Harefield**, Hillingdon – southernmost is **Old Coulsdon**, Croydon – easternmost as well as most outlying overall is **North Ockendon**, Havering.



Map of Greater London showing railway lines, primary roads, motorways, and suburban towns of the county

Map all coordinates in "Category:Areas of London" using: [OpenStreetMap](#)

Download coordinates as: [KML](#) · [GPX](#)

Location	London borough	Post town	Postcode district	Dial code	OS grid ref
Abbey Wood	Bexley, Greenwich ^[7]	LONDON	SE2	020	TQ465785
Acton	Ealing, Hammersmith and Fulham ^[8]	LONDON	W3, W4	020	TQ205805
Addington	Croydon ^[8]	CROYDON	CR0	020	TQ375645
Addiscombe	Croydon ^[8]	CROYDON	CR0	020	TQ345665
Albany Park	Bexley	BEXLEY, SIDCUP	DA5, DA14	020	TQ478728
Aldborough Hatch	Redbridge ^[9]	ILFORD	IG2	020	TQ455895
Aldgate	City ^[10]	LONDON	EC3	020	TQ334813
Aldwych	Westminster ^[10]	LONDON	WC2	020	TQ307810
Alperton	Brent ^[11]	WEMBLEY	HA0	020	TQ185835
Anerley	Bromley ^[11]	LONDON	SE20	020	TQ345695
Angel	Islington ^[8]	LONDON	EC1, N1	020	TQ345665
Anerfield	Bromley ^[11]	WESTERHAM	TN16	01959	TQ425585

Wikipedia page showing the London boroughs, locations and post codes in London.

Secondly, I'll be using the explore parameter of the Four Square API to pull back the entertainment venues, bars and restaurants for the user inputted area of interest.

```
url = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&v={}&ll={},{&radius={}&limit={}&query={}'
```

Finally, I'll use the venue ID parameter of the Four Square API to return the rating and price range of a particular set of bars or restaurants, in order for the user to decide which they'd most like to go to. The reason for doing this towards the end once the venues have been clustered is to reduce the number of calls through the API and not breach the 50 premium calls limit.

```
url = 'https://api.foursquare.com/v2/venues/{id}?client_id={}&client_secret={}&v={}'
```

Methodology

Once the data has been pulled from Wikipedia and the user has selected which region of London they would like to explore, I was able to start playing with the data. The main component of my project was the K mean clustering analysis I performed on both the bars query and the restaurant query. For each one I used a similar method:

1. Used one hot encoding to get the data into a uniform position. I used the Pandas dummies for this section on the bar/restaurant categories to show which bar/restaurant fell into which category.
2. Once I had this grouped, I went on to build the K clustered elbow method graph – to find the optimal K number of bars/restaurant clusters. It was important to adjust the range of K's to explore in the graph each time I ran it through to ensure there was a clear pattern showing the elbow point of the line graph.
3. With the clear elbow point showing the optimal number of K clusters, I put this into the K clusters algorithm, to assign each bar/restaurant with a K cluster number. This step was important because it allowed me to break out the venues into groups of similarity, with enough difference between the other clusters to make a decision by the user on which cluster to go to next significant.

Results

The final result of this project gave the user a randomised entertainment venue for them to go to on their date night. From there, the script would help them find a certain type of bar and a certain type of restaurant nearby to the entertainment venue. Finally, the bars and restaurants are re-run through the Four Square API to return their score and price range, allowing the user to have a solid list of venues to choose from and make an informed decision on visiting.

Discussion

This project was an excellent learning experience. It focussed heavily around API usage, with the K clustering section breaking down the data in a meaningful way. One of the key criticisms I have for my own project is the inability to find venue ratings and price range earlier on in the analysis. Had I done that I would have been able to give the user a wider range of venues to choose from in a specific area. However, that was not the case due to the Four Square API having a limit of 50 premium calls per day – this method would have exceeded it almost immediately.

Conclusion

To conclude, this Python script is designed with various types of audiences in mind. If the user of this script is somebody who is less confident on how to build a date night, or perhaps doesn't have the time to go through and research all restaurants and bars in the nearby area – this will answer those questions for them. There are certainly limitations, for example this is currently on based on London. Should it wish to be expanded to other areas, the entire basis of the first section of the analysis (Wikipedia pull, postcode selection) would have to be re-built to fit the location.