Content-based Recommender System

Utilizing User Data

(Applied Data Science Capstone)

by

Jay Mitchell

1/14/2020

Introduction

People often travel for work or pleasure. During that time, they may wish to find amenities similar to those that they normally utilize. Companies may also be looking for new patrons, and reaching out to the out-of-town crowd may increase sales.

A user may not desire to be reactive regarding venue choices. A user may desire that instead of inputting the desired venue, that a machine-coded program provide a venue that is desired without prompting.

Additionally, a company, such as Foursquare, which was utilized during this project, may desire to capitalize on its data. They may partner with local venues to drive business toward that venue. An example may be a coupon or similar incentive. The coupon may be presented by the user to the venue. The venue then may provide monetary reward to Foursquare. In addition, Mint data was utilized to generate a user profile. A company such as Mint may also benefit from such a scheme.

Data

Data was collected from two main sources: Mint and Foursquare. The Mint data was collected to create a user profile. This data may be collected from any such repository of user actions. Mint was selected as they categorize purchases, which simplifies the data cleaning. The data received was a text file. The text file was open and then read into a Pandas dataframe for further processing. Data processing included eliminating rows with “credit”, as the aim was to determine the user’s purchases. Also, many columns included unnecessary information and were eliminated. Simple dataframe manipulations suggested that the data comprised mainly object, and, thus, numerical analysis would not be useful. As a tool to determine the user profile a count was performed on the categories to determine those that were utilized significantly. Some categories were eliminated, such as “student loans” as a user would not likely wish to purchase more of those.

The Foursquare data was obtained using the categories derived from the Mint data. The closest five venues were pulled for each category. Foursquare was further utilized to get a user score for each venue. For those venues without a user score, one was generated at one standard deviation below the mean of those pulled in order to not lose much data and also to not over utilize Foursquare, which has data usage limits. The Foursquare and Mint data are then combined to create a recommendation engine.

Methodology

Functions include df.types and df.corr were utilized to gain familiarity with the data. See Figure 1.

```
[146]: print(df.dtypes)

Date                    object
Description             object
Original Description    object
Amount                 float64
Transaction Type        object
Category                object
Account Name            object
Labels                 float64
Notes                  float64
dtype: object
```

```
[147]: df.corr()
```

[147]:

	Amount	Labels	Notes
Amount	1.0	NaN	NaN
Labels	NaN	NaN	NaN
Notes	NaN	NaN	NaN

Figure 1. Initial analysis.

Based on this information, numerical analysis is unlikely to achieve results. A more categorical analysis would then be performed. To do so df.value_counts() was utilized to determine the count of the unique categories. This was visualized as a horizontal bar chart (see Figure 2). The bar chart aiding in eliminating certain categories from the data.

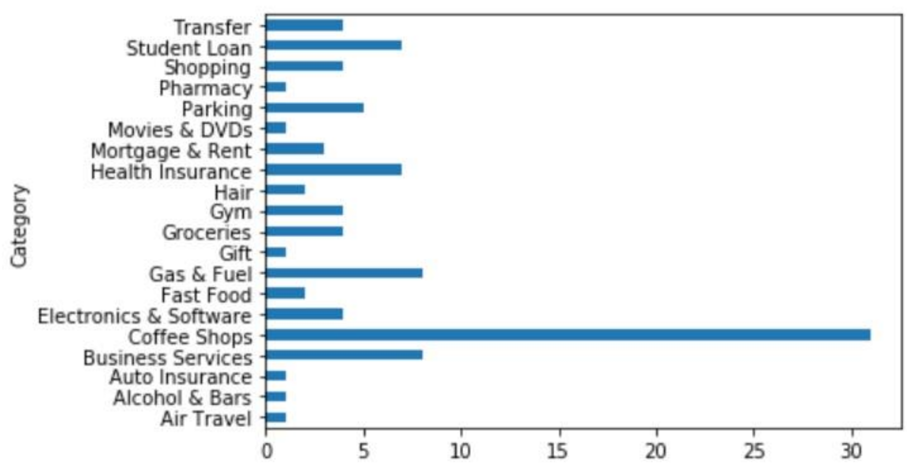

Figure 2. Categories.

Figure 2 also highlighted that there may be significant outliers. Such outliers may skew the recommendation too much. To determine the significance of any outlier, a box plot was generated. See Figure 3.

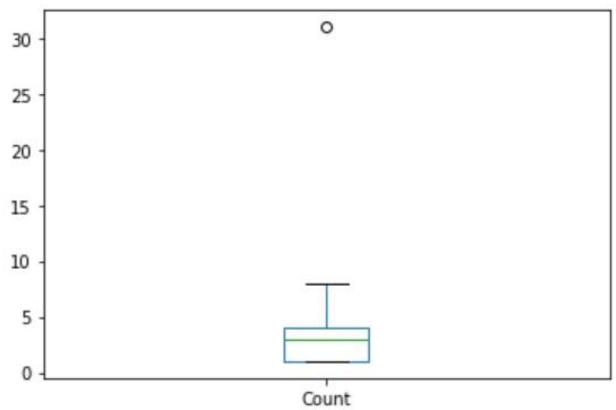

Figure 3. Initial box plot.

Figure 3 shows that one value is an extreme outlier. No other category was outside the top 75%. Thus, some reduction of weight was performed. Here, I determined to reduce the value of any outlier to 1.5X the next greatest value. It would still be weighted heavily, but not as extreme as depicted in Figure 3. The results are depicted in Figure 4.

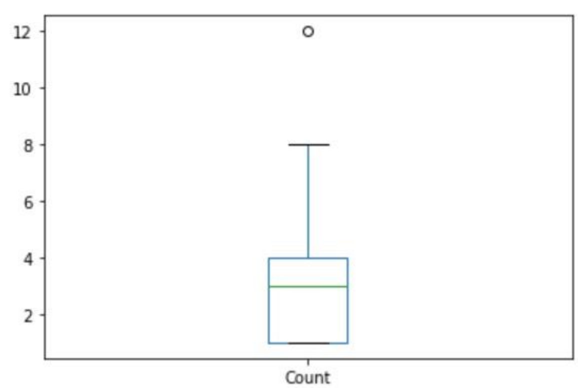

Figure 4. Final box plot.

The counts for each category was then converted into a weight. Here, I wanted weights of a positive values scaled to 10. Thus, I use simple feature scaling to get positive numbers between 0 and 10. I normalized to between 0 and 1 using Figure 5, the multiplied that score by 10). This was used instead of min-max scaling to avoid a zero value for the min. Also, this was used instead of Z-score as that would have given some negative values. The positive values will be utilized later as a weight. See Figure 6.

```
df_list['NormScore']=(df_list['Count'])/(df_list['Count'].max())
```

Figure 5. Simple Feature Scaling formula.

	Category	Count	NormScore
0	Alcohol & Bars	1	0.833333
1	Coffee Shops	12	10.000000
2	Electronics & Software	4	3.333333
3	Fast Food	2	1.666667
4	Gas & Fuel	8	6.666667
5	Gift	1	0.833333
6	Groceries	4	3.333333
7	Gym	4	3.333333
8	Hair	2	1.666667
9	Movies & DVDs	1	0.833333
10	Pharmacy	1	0.833333
11	Shopping	4	3.333333

Figure 6. Final Category Weights.

Foursquare data was then received based on the categories above. A Foursquare “search” was utilized with the search query set to the categories in Figure 6. The code utilizes Westminster as the location, but any may be utilized. The closest five (5) venues for each category were retrieved. In order process the data later, dummy rows were inserted in there are not 5 venues in the location. Figure 7 depicts the dataframe, though I encourage using the link to the GitHub code to read it.

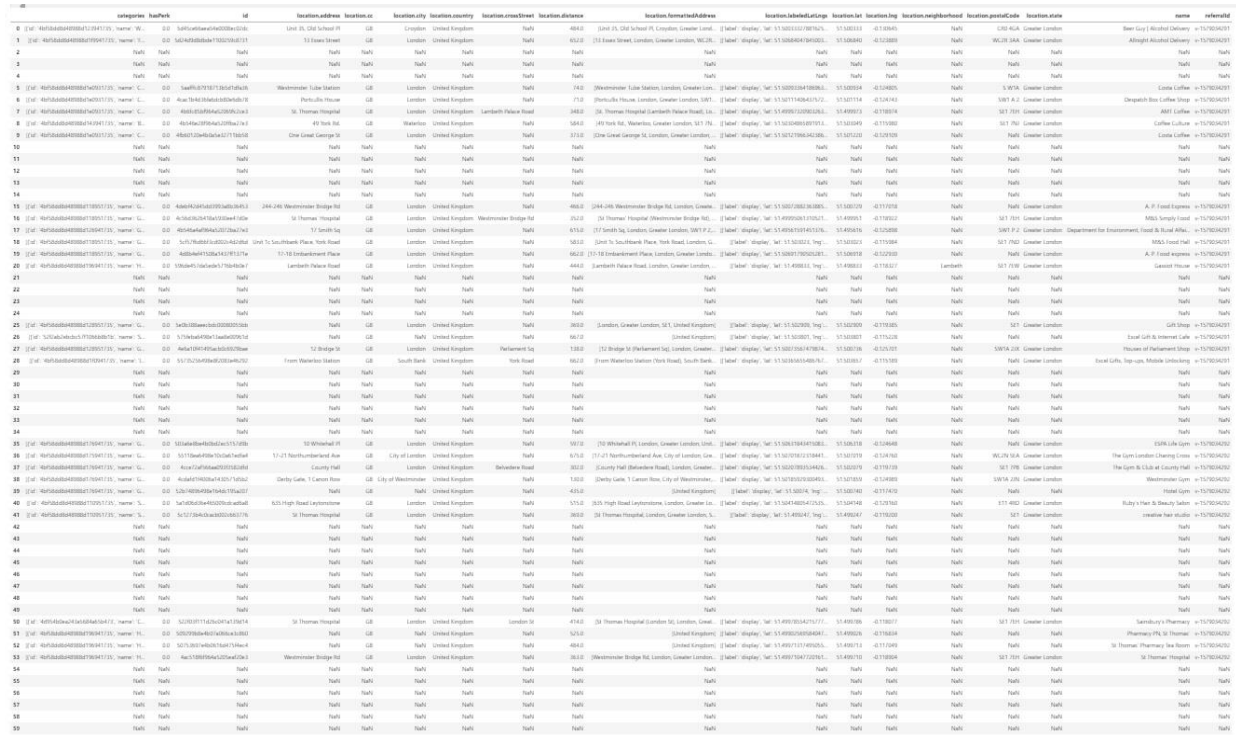

Figure 7. Venues.

Once the venues were obtained, the categories from the Mint data were added using df.concat. Once this was done, the dummy rows were then deleted. The dummy rows may the concatenation of the dataframes much easier.

Next, each venue included a venue id in Figure 7 above. These venue ids were utilized along with the Foursquare “venues” option to return the values for each venue. Some venues, however, did not have a value and were initially recorded as ‘NaN’. In order to provide some, but not a lot of weight to unrated venues, the mean and standard deviation of the rated venues were determined and the unrated venues were given a rating of (mean – standard deviation). This puts them on the low end, but not as outliers. As these ratings are already on scale to ten (10), they were not further manipulated. See Figure 8.

0	6.240819
1	6.240819
2	7.900000
3	6.240819
4	6.600000
5	6.200000
6	6.240819
7	6.240819
8	6.800000
9	6.240819
10	6.240819
11	6.240819
12	6.240819
13	6.240819
14	6.240819
15	6.240819
16	6.240819
17	7.400000
18	6.100000
19	7.600000
20	6.240819
21	6.240819
22	6.240819
23	6.240819
24	6.240819
25	6.240819
26	6.240819
27	6.240819

Figure 8. Venue Ratings.

Results

Now that there is a rating from the venue, each venue has a category, and there is a weight for each category, a final WeightScore (Venue Rating x NormScore) is determined for each venue. The dataframe is sorted in descending order by WeightScore and the top 10 venues are selected. Further dataframe manipulation is performed to ensure that the name of the venue and its latitude and longitude are in the dataframe for plotting. The dataframe is depicted in Figure 9. Again, dataframe viewing is best done via GitHub.

	name	categories	address	cc	city	country	crossStreet	distance	formattedAddress	labeledLatLngs	lat	lng	neighborhood	postalCode	state	id	WeightScore	Category
2	Costa Coffee	[{'id': '4bf58dd8d48988d1e0931735', 'name': 'C...	Westminster Tube Station	GB	London	United Kingdom	NaN	74.0	[Westminster Tube Station, London, Greater Lon...	[{'label': 'display', 'lat': 51.50093364186963...	51.500934	-0.124805	NaN	S W1A	Greater London	NaN	79.000000	Coffee Shops
4	AMT Coffee	[{'id': '4bf58dd8d48988d1e0931735', 'name': 'C...	St. Thomas Hospital	GB	London	United Kingdom	Lambeth Palace Road	348.0	[St. Thomas Hospital (Lambeth Palace Road), Lo...	[{'label': 'display', 'lat': 51.49997320903263...	51.499973	-0.118974	NaN	SE1 7EH	Greater London	NaN	66.000000	Coffee Shops
3	Despatch Box Coffee Shop	[{'id': '4bf58dd8d48988d1e0931735', 'name': 'C...	Portcullis House	GB	London	United Kingdom	NaN	71.0	[Portcullis House, London, Greater London, SW1...	[{'label': 'display', 'lat': 51.50111406437572...	51.501114	-0.124743	NaN	SW1 A 2	Greater London	NaN	62.408193	Coffee Shops
6	Costa Coffee	[{'id': '4bf58dd8d48988d1e0931735', 'name': 'C...	One Great George St	GB	London	United Kingdom	NaN	373.0	[One Great George St, London, Greater London, ...	[{'label': 'display', 'lat': 51.50121966342386...	51.501220	-0.129109	NaN	NaN	Greater London	NaN	62.408193	Coffee Shops
5	Coffee Culture	[{'id': '4bf58dd8d48988d143941735', 'name': 'B...	49 York Rd.	GB	Waterloo	United Kingdom	NaN	584.0	[49 York Rd., Waterloo, Greater London, SE1 7N...	[{'label': 'display', 'lat': 51.50304865891913...	51.503049	-0.115980	NaN	SE1 7NJ	Greater London	NaN	62.000000	Coffee Shops
12	Gassiot House	[{'id': '4bf58dd8d48988d196941735', 'name': 'H...	Lambeth Palace Road	GB	London	United Kingdom	NaN	444.0	[Lambeth Palace Road, London, Greater London, ...	[{'label': 'display', 'lat': 51.498833, 'lng':...	51.498833	-0.118327	Lambeth	SE1 7EW	Greater London	NaN	41.605462	Gas & Fuel
19	The Gym & Club at County Hall	[{'id': '4bf58dd8d48988d176941735', 'name': 'G...	County Hall	GB	London	United Kingdom	Belvedere Road	302.0	[County Hall (Belvedere Road), London, Greater...	[{'label': 'display', 'lat': 51.50207893534426...	51.502079	-0.119739	NaN	SE1 7PB	Greater London	NaN	25.333333	Gym
17	ESPA Life Gym	[{'id': '4bf58dd8d48988d176941735', 'name': 'G...	10 Whitehall Pl	GB	London	United Kingdom	NaN	597.0	[10 Whitehall Pl, London, Greater London, Unit...	[{'label': 'display', 'lat': 51.50631843415083...	51.506318	-0.124648	NaN	NaN	Greater London	NaN	24.666667	Gym
21	Hotel Gym	[{'id': '4bf58dd8d48988d176941735', 'name': 'G...	NaN	GB	NaN	United Kingdom	NaN	435.0	[United Kingdom]	[{'label': 'display', 'lat': 51.50074, 'lng': ...	51.500740	-0.117470	NaN	NaN	NaN	NaN	20.802731	Gym
20	Westminster Gym	[{'id': '4bf58dd8d48988d176941735', 'name': 'G...	Derby Gate, 1 Canon Row	GB	City of Westminster	United Kingdom	NaN	130.0	[Derby Gate, 1 Canon Row, City of Westminster,...	[{'label': 'display', 'lat': 51.50185929300493...	51.501859	-0.124989	NaN	SW1A 2JN	Greater London	NaN	20.802731	Gym

Figure 9. Mapped Venues.

The Folium mapping tool was then used to depict the venues in Figure 9. To determine the current location, a large red dot was plotted. The blue dots then represent each of the venues. The radius of the blue dots plotted was based on the WeightScore, such that the relative size is proportional to the WeightScore. Each dot may be interacted with to show the venue name and the category to which it belongs. A sample map is depicted in Figure 10, with one venue interacted with. Note that the map figure does not fit into my browser.

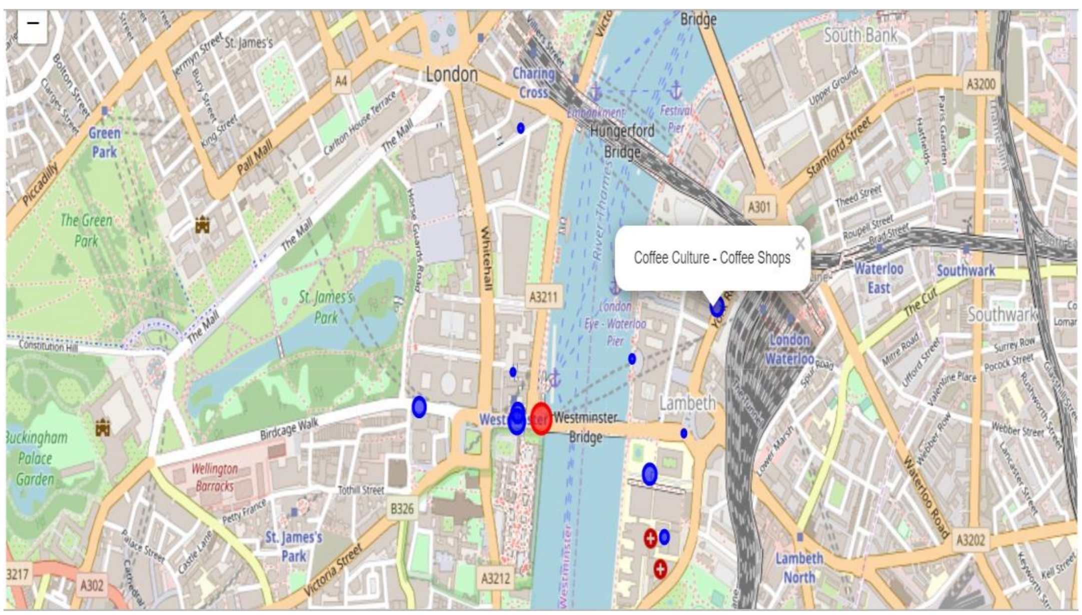

Figure 10. Local Points of Interest.

Discussion

The above recommendation engine did produce local venues of interest based on my interests. Further implementation may focus on improving data, processing, and profit.

Data

The above recommendation engine did produce local venues of interest based on my interests. Further implementations my utilize other data sources based on the particular user. Not everyone utilizes Mint and Foursquare. Also, the data from Mint may be updated from time to time to ensure relevancy. More generalized categories may be used as well. With further data, a k means clustering algorithm may be used to determine which categories are similar.

Processing

The final recommendation was very much skewed toward those categories with many counts. Counts may not be the best way to determine a user profile. Other methods may reduce the impact of the outlier. Binning may be utilized to get items that are 'high', 'medium', and 'low', or some other scheme. Each bin may then be associated with a weight. This may result than more than just coffee shops. Though, I not complaining.

Profit

This implementation also needs more profit. The popup label may also include a hyperlink to a coupon or other incentive. Such an incentive should include a way for the data sources (and the app developer) to make a cut of the earnings from the directed customer.

Conclusion

This recommendation engine profiled a user based on historical data. That data was then used to determine nearby venues, such as when traveling, in a different part of town, or just looking for hidden gems. Previous users had rated those venues. Those ratings, along with the user profile, determined the venues that were displayed to the user. The user then could interact with a Folium map to determine their present location and the locations and weight score of venues of interest.

Code: https://github.com/jmitchell4390/Coursera_Capstone/blob/master/Capstone_Project.ipynb

Map: https://nbviewer.jupyter.org/github/jmitchell4390/Coursera_Capstone/blob/master/Capstone_Project.ipynb

Text file: https://github.com/jmitchell4390/Coursera_Capstone/blob/master/transactions.txt