IBM Data Science Capstone Project: Final Report

A District Recommender

Georgios Lekkas, June 3rd 2020

Table of Contents

1. Introduction / Business Problem

Any person moving to a new city for work or studies, faces the problem of having to decide where to stay. If that person has not travelled to the new city in the past and does not have the time or budget to do an on-site exploration, he or she must base their decision on the information they can find on the Web.

The problem is that online travelers' guides of new destinations are destined to tourists, while forums for expats or students have scant information and carry subjective opinions based on small samples.

The new "District Recommender" service comes to fill that gap and support decisions with objective data.

1.1 Interest

Users interested in the new "District Recommender" service may be workers who received and accepted job offers that require their relocating to a new city, or students who have accepted enrolment offers and are about to start their first year in a University. The problem is particularly felt in big cities that present a large number of options.

The majority of users will presumably be young, so they will appreciate a Recommender service that takes their personal preferences into account.

The service will also be of interest to business partners, because it can draw a lot of advertising revenue. Users will declare their intention to move to a new city and their preferences in lifestyle, supplied in terms of areas in cities they have visited and enjoyed in the past. This way it will be possible to show them advertisements related to the city and area, even without collecting and storing any other sensitive personal data.

1.2 Success

We will know the service is successful if it can produce neighborhood recommendations for new destinations that are compatible with a user's prior preferences.

The new system will be tested by evaluating its recommendations on a small number of cities.

2. Data

The choice of data sources used to solve the problem depends heavily on the purpose of the service. Our decision is to create a District Recommender service that supports mobility of young workers or students across many countries. The system was to be developed initially for three popular destinations:

- London, England;
- Glasgow, Scotland;
- Copenhagen, Denmark.

Some data sources were considered but not adopted, because to use it would restrict the scope of applicability. For example, happiness & wellbeing data and deprivation data were

easily available, but only for London boroughs. Data on house prices per square meter could be found for England and Wales, but not for Scotland.

The following sections present the data that was obtained and used by the District Recommender.

2.1 Districts

The areas / neighborhoods in each city are defined by *partially aggregated postcodes*; that is popular, coarse-grained subdivisions that define entire boroughs. We do not use the fully detailed postcode that can define specific streets or even single buildings in the UK or Denmark. So, for example, in the UK we use the postal district 'EC2' (Bishopsgate), not the full postcode 'EC2M 4NR', of which there are 1.7 million in the UK.

The easiest way to obtain postcode aggregate data is to scrape them from Wikipedia, for example from:

- https://en.wikipedia.org/wiki/List_of_postal_codes_in_Denmark
- https://en.wikipedia.org/wiki/G_postcode_area
- https://en.wikipedia.org/wiki/London_postal_district#List_of_London_postal_districts

The downloaded data is of the following form:

District_code	District_name	
1	G1	Merchant City
2	G2	Blythswood Hill, Anderston (part)
3	G3	Anderston, Finnieston, Garnethill, Park, Woodl...
4	G4	Calton (part), Cowcaddens (part), Drygate, Kel...

Table 1: List of districts

For each city, districts that are not residential but are used for mail distribution purposes were deleted from the datasets (aka "non-geographic postcodes").

The final data set consisted of 269 districts:

- London: 167
- Glasgow: 51
- Copenhagen: 51

2.2 Geolocation data

For each of the districts of the three cities, geolocation data was obtained by an online geolocation service.

Downloading data from a Web site was also considered, for example UK detailed postcode data available from the 'Office for National Statistics – Open Geography Portal', but was dismissed as too detailed, down to the level of individual street.

The online geocoding service provider used was ArcGIS by ESRI.

2.3 Venue data

For each district, the Foursquare API https://developer.foursquare.com was used to obtain a list of available venues in each location. For each district, we obtained a list with

- Each venue's name
- The venue geographical coordinates
- The venue category

The list of venues is used to define the profile of each district. This is done indirectly, through a further categorization of venues, as explained in the following paragraph.

2.4 Venue categories

A possible limitation in using venue categories returned by the Foursquare API, is that it may be too detailed to allow for meaningful comparisons. For our purposes, it seemed to be excessive to make very fine distinctions between the sub-types of restaurants or the types of outdoor recreational activities.

For example, how much difference is there between an Italian and a Spanish restaurant, or a Tibetan and an Indian restaurant? Should they be counted as two completely different traits when determining the 'profile' of a district?

We can derive the relation between similar venues from the Foursquare API. It classifies venues according to a total of 943 categories, organized hierarchically in multiple levels: https://developer.foursquare.com/docs/build-with-foursquare/categories/.

We used the Foursquare API to download its Category Hierarchy. Then on the basis of it, we defined a data structure called **Useful Categories**, i.e. a selection of a subset of categories, considered to be interesting to be used by our District Recommender service.

After some processing, each venue was augmented with a "High Level Category" element. The augmented venue data set looked like this:

District Longitude	Venue	…	Venue Category	HL Venue Category
EC1A	Pilpel	…	Falafel Restaurant	Food
EC1A	Virgin Active	…	Gym / Fitness Center	Outdoors & Recreation
EC1A	Postman's Park	…	Park	Outdoors & Recreation

Table 2: District Venue & Category data, augmented with aggregate categories

2.5 User preferences

The District Recommender service needs two kinds of user input:

- The **city of destination**, to which the user wants to relocate
- **Preferences** in terms of districts, in cities the user visited in the past, together with associated rating.

An example of preferences expressed by a user who has stayed in the city of London, would look as follows:

District code	District_name	City	Rating
E9	Hackney, Homerton	London	3.5
N1	Barnsbury, Canonbury, Islington	London	4.0
W11	Holland Park, Notting Hill	London	5.0

Table 3: User input: preferred London districts with their ratings

The user need only input the district code, the district name is added by the system and shown here for clarity.

This user input is used to calculate preferences in terms of district type and of the lifestyle they would like to enjoy.

The user also needs to supply a destination, for example:

- Destination city: **Copenhagen**

3. Methodology

The District Recommender problem we want to solve maps directly to a class of problems and solutions studied by Data Science, known as Recommender Systems. Generally speaking, there are two types of recommender systems:

a) **Content-based** recommenders, where the system tries to figure out the user's favourite aspects of an item, and then recommends items that present those aspects.

b) **Collaborative filtering** recommenders, where the system find other users that have similar preferences and opinions to the target user, and then recommends items that those others have liked.

With these recommender algorithms already available, the decision was to solve the problem by bulding a **content-based recommender** that recommends districts to our user. To do this, we must find out the user's favorite aspects from the districts and ratings given.

There are two kinds of data used:

a) A full dataset of **districts & venue categories**, that cover at least one city the user knows and the city where the user wants to move to. The dataset contains the number of venues in each category and each district.

b) A set of **user preferences,** in terms of districts of a known city and numerical ratings for each district.

We process this data to produce a recommendation in five steps:

1. **Transform** the dataset of districts and venue categories, using one hot encoding
2. **Extract** from the full dataset only the districts that correspond to the user's preferences
3. **Calculate a user profile vector** of weights assigned to Venue categories, by multiplying User ratings and Venue category numbers

4. Obtain a numeric **preference score for each unknown district**, by multiplying the user profile vector to the venue population data for each district.
5. **Present** the sorted top scoring alternatives, showing results also on a map.

The following sections will illustrate those five steps. Example data that is shown is partial and for reasons of clarity only. Results will be discussed in the concluding chapters.

3.1 Data Transformation

3.1.1 Category Aggregation

The Foursquare exploration API returns a detailed classification for each venue. We would like to pick specific categories in the category hierarchy, that can serve as **aggregation points** that represent all detailed categories below it.

The steps to implement this data transformation function were:

1) Define a list of high-level categories deemed useful for aggregation and comparison purposes
2) Transform the Foursquare Categories to a JSON Dictionary, and augment the dictionary with links that permit to navigate upwards in the category hierarchy
3) Define a search function that, given an input string, searches the Categories JSON dictionary to locate the corresponding category element
4) Define a lookup function that, starting from a specific node in the category tree, navigates upwards, until the node that will act as point of aggregation.

This customization of the way the district profile is calculated and the districts are compared, can be used in two contexts:

a) the Recommended administrators can change the way the system works, so that it produces different results
b) the end User might customize the aggregation categories, so as to ignore or take into account categories according to personal preferences.

After aggregating the venue data for each district, and applying the mean(), it looked as follows:

District_code	Arts & Entertainment	Food	Nightlife Spot	…	Shop & Service	Travel & Transport
1000-1499	0.261364	0.443182	0.102273	…	0.102273	0.056818
1500-1799	0.000000	0.538462	0.076923	…	0.076923	0.153846
1800-1999	0.261364	0.443182	0.102273	…	0.102273	0.056818
2000	0.000000	0.411765	0.058824	…	0.294118	0.058824
2100	0.127273	0.381818	0.054545	…	0.218182	0.000000

Table 4: District Venue Category data, aggregated view

3.1.2 Data Scaling

The first attempts with the above dataset produced very inaccurate recommendations; they were substantially identical even when the user input changed. The cause was traced back to the aggregated table just shown, that calculates a numerical profile of each district in terms of located venues.

Looking at the table in the previous paragraph, it is evident that the Food category dominates others, with a weight counts for 50-60% of the overall district profile. Districts that are rich in food venues always came top of the recommendations list.

We needed a way to scale back the dominance of that category. Experimenting with the first candidate of **scaling methods**, min-max-scaling, gave promising results, but did not make the numbers sufficiently balanced. Almost all methods available in the scikit-learn documentation were tried out:

- min-max scaling (normalization): good but not enough
- z-scaling (standardization)
- Robust scaler
- PowerTransformer
- QuantileTransformer: the best so far.

Generally speaking, methods that perform standardization (producing a means of 0) did not do well. The idea of most of the Venue Categories contributing negative coefficients does not map well to human intuition, and the results in terms of proposed districts were not satisfactory.

The approach that produced far better results than any other, was the **QuantileTransformer (uniform output).**

The QuantileTransformer applies a non-linear transformation such that the probability density function of each feature is mapped to a uniform distribution, and is particularly robust in front of outliers.

The output of the data transformation is as follows:

District_code	Arts & Entertainment	Food	Nightlife Spot	…	Shop & Service	Travel & Transport
1000-1499	0.916667	0.494186	0.451550	…	0.362403	0.527132
1500-1799	0.000000	0.742248	0.284884	…	0.207364	0.860465
1800-1999	0.916667	0.494186	0.451550	…	0.362403	0.527132
2000	0.000000	0.368217	0.228682	…	0.841085	0.612403
2100	0.732558	0.304264	0.217054	…	0.740310	0.000000

Table 5: District Venue Category data, after scaling with the QuantileTransformer

The table contains no single category with numbers 5x-10x bigger than those of other categories, as previously.

3.2 Extraction of Districts Preferred by User

We extract the subset of the cities' encoded data that belongs to districts known to the user. The data corresponding to our example user input, is as follows:

District code	Arts & Entertainment	...	Food	Nightlife Spot	Outdoors & Recreation	Professional & Other Places	Shop & Service	Travel & Transport
E9	0.521318	...	0.612403	0.618217	0.866279	0.000000	0.476744	0.393411
N1	0.000000	...	0.689922	0.926357	0.542636	0.992248	0.220930	0.271318
W11	0.432171	...	0.701550	0.655039	0.449612	0.000000	0.759690	0.306202

Table 6: Venue Category data for districts preferred by user

3.3 Calculation of User Profile vector

The numerical User Profile represents user preferences on venue categories expressed as numbers. It is a vector with the weights of each venue category.

To obtain the weights we need to multiply the user ratings vector by the matrix containing the districts and venue categories preferred by the user

User profile = (User Ratings) x (Venue Category Data of user-preferred districts)

The result of the dot product for our example is as follows:

Venue High Level Category	Weight
Arts & Entertainment	3.985465
College & University	0.000000
Food	8.410853
Nightlife Spot	9.144380
Outdoors & Recreation	7.450581
Professional & Other Places	3.968992
Shop & Service	6.350775
Travel & Transport	3.993217

Table 7: User Profile vector

The new adjustments bring Nightlife and Outdoors & Recreation to weigh nearly as much as or more than Food. That seems fair and in line with the London areas chosen by the user.

3.4 Calculation of Preference Score for each district

To arrive at a single number of 'likeability' for each district, we'll calculate a **weighted average**: the weights corresponding to venue categories of each district will be multiplied by the weight of the user's preferences for each category. The formula is

District score = (Venue Category Data) x (User profile vector)

To avoid calculating preference values for the entire world, we use the destination city to filter out districts of other cities.

3.5 Sorting and presentation of results

As a final act, the table of district scores is sorted and the highest scoring resuts are presented to the user: they are the recommended solutions.

District_code	District_name	Score
2200	Copenhagen N	0.610662
2400	Copenhagen NV	0.546174
1500-1799	Copenhagen V	0.451454
2000	Frederiksberg	0.443338
2100	Copenhagen Ø	0.432967

Table 8: Districts recommended by the system

The results are also shown on a city map, with clickable markers that reveal the district name and current preference score.

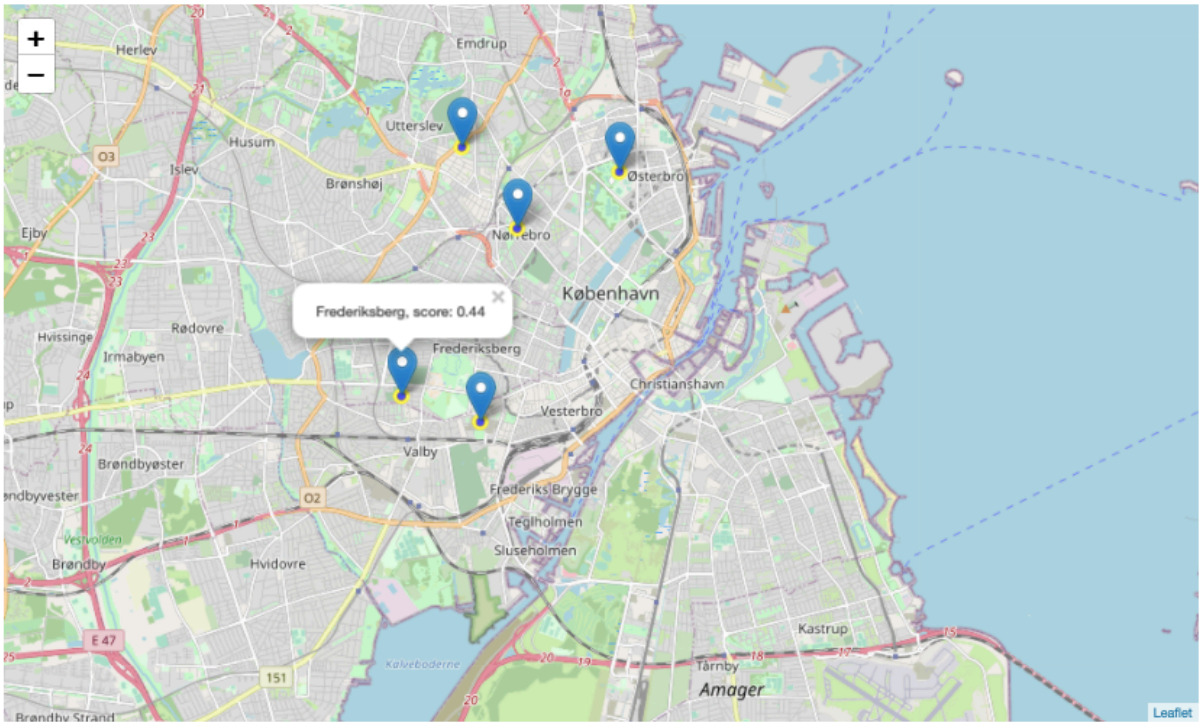

Figure 1: Recommended districts, shown on city map

4. Results

4.1 Main case

The initial test case had the user declare Copenhagen, Denmark as preferred destination city. Lifestyle preferences were declared as three London neighborhoods with associated high ratings: Hackney, Islington and Notting Hill.

The recommendations produced by the algorithm were credible. The three London neighborhoods preferred by the user, Hackney, Islington and Notting Hill, can roughly be associated to the "trendy / hipster" category.

Looking at Copenhagen, the District Recommender suggested the areas of Copenhagen North, North-West, West, East and of Federiksberg. These are the central / semi-central boroughs of the city with many venues for Food, Art, Nightlife, Shops and Recreational activities. The districts of **Nørrebro, Vesterbro, Østerbro** and the municipality of **Frederiksberg** are well-known international tourist destinations.

4.2 Alternative case

What is the response of the algorithm under different inputs? We changed the user's inputs to be the areas of Wimbledon and Twickenham, two green suburbs of London of the affluent type.

District_code	District_name	City	rating
SW19	Merton, Wimbledon	London	5.0
TW	Twickenham	London	3.0

Table 9: Districts preferred by the user for alternative case

The recommendation of the system mostly changed, with Copenhagen North and North-West coming again first and second, with their scores 10% lower than in the main test case.

In places 3 to 5, come three inhabited areas that belong to the greater Copenhagen area, quite far from the capital and featuring large green spaces.

District_code	District_name	Score
2200	Copenhagen N	0.562705
2400	Copenhagen NV	0.520415
2990	Nivå	0.459110
2640	Hedehusene	0.459110
2980	Kokkedal	0.459110

Table 10: Districts recommended by the system in alternative case

The distance from the centre can be seen better if we show the recommendations on a map. These seem indeed to be as far from the city centre, as Twickenham is distant from Central London.

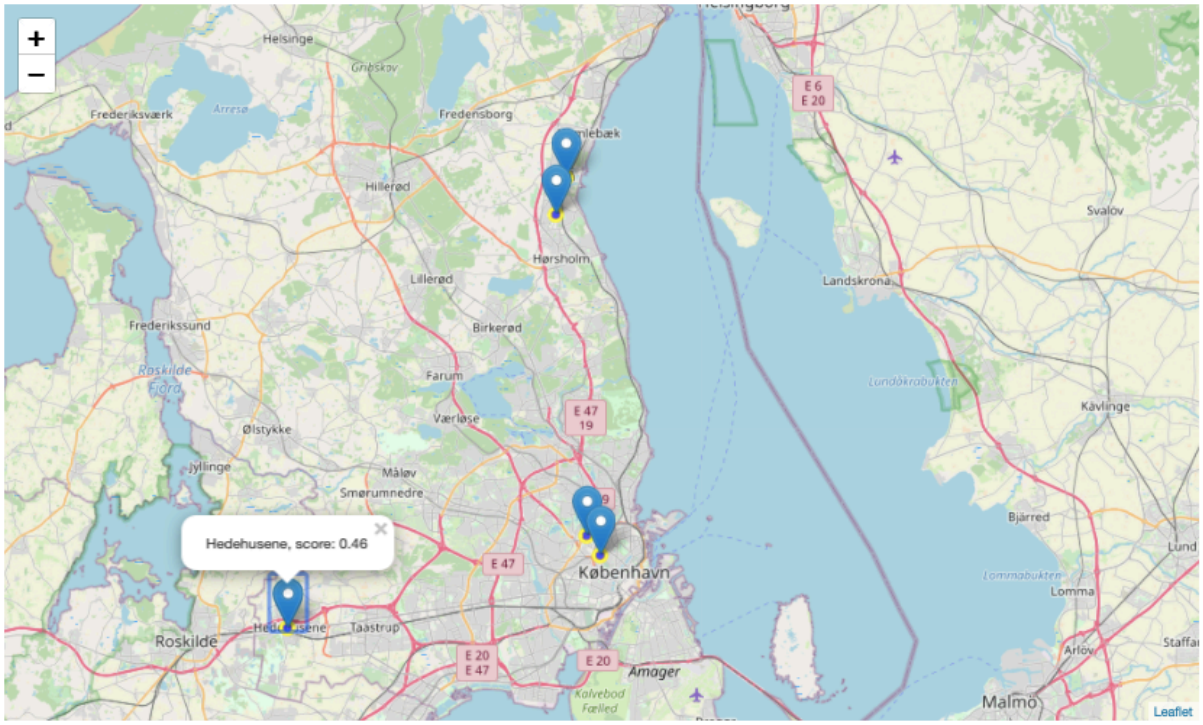

Figure 2: Recommended districts for alternative test case, shown on city map

How do these results fare? They seem good enough, but the same 2 districts featuring in the two top recommendations is a sign that the system can be improved.

Part of this coincidence in recommendations may be unavoidable, due to the nature of the two Copenhagen districts: they are large and are of a double nature: they comprise areas bustling with shops and ethnic restaurants, and at the same time, contain parks and other recreation spaces typical of distant suburbs.

6. Discussion

The Recommender produces results that seem acceptable and could be used as support to human decision-making. This section discusses various ways in which the system could be improved.

A key issue in the quality of the results produced by the Recommender is the **representation** of the district profiles.

How should we best represent the profile of each district? How much importance should we give to the existence of one vs. dozens of venues of the same type? Is it enough to mark "at least one venue exists"? How many venues of a type are sufficient, i.e. what is the marginal advantage for a user in having more than 30 coffee shops in the same area? Should we differentiate districts more, based on the number of similar venues? Experiments with different scaling methods produced drastically different outcomes. The solution used by the Recommender could be improved by further research.

This search of better representations could perhaps be supported by **clustering** algorithms and data **visualization**, that would make more evident the quality of each alternative representation in a more explicit manner.

Apart from scaling and number representations, another aspect related to ratings is the **impact of high and low ratings** on the final result. We are used to thinking that a rating of one star out of five is low, and this is a negative judgement on the item being rated. The feeling from running the Recommender with the current setup, is that high and low ratings were perceived as good and 'a bit less good' inputs. There was no real penalty generated by a district profile that had received an unfavorable review, only a somewhat lower advantage.

Another aspect that could improve results in the future would be the **addition of other data sources**, for example additional data on **property prices**. There is one aspect of the profile of each district that is not well represented in the venue profile obtained by Foursquare, and that is the value of land and consequently on the prices of nearby activities. An "Italian Restaurant", for example, can be a cheap takeaway or an expensive venue.

Prices are an additional indicator of the type of neighborhood. Unfortunately, price data were not easy to find outside England. In a future attempt, a proxy could be used to represent the cost of living, for example rental prices for each area could be obtained from a Real Estate web site.

Another improvement to the system would be to fine-tune the quality of recommendations based on a **different set of venue categories**. It was decided not to use the 943 venue categories of Foursquare for comparisons, and to use the 10 top-level aggregates instead. Perhaps better results can be obtained if another set is used, that sits at an intermediate level between the two. The complexity of the task demands this search of the solution space to be conducted in an automated way.

Improvement could also come from using a different set of districts, adopting other **criteria of geographical decomposition**. Smaller districts would be more characteristic and less generic, leading to improved accuracy in recommendations.

7. Conclusion

The Content-Based Recommender discussed in this report follows a well-known basic approach that can produce results based only on a user's preferences. The algorithm can be customised because the user can select which categories to use to compare city districts, picking arbitrarily venue categories located anywhere in the category hierarchy.

The system produces basic recommendations that are credible and in sync with reality, although not yet ready for production use.

To improve the algorithm's accuracy, more data sources should be integrated, preferably adding prices.Also more alternatives should be studied relatively to the optimal representation of the district profiles.

Future should take into consideration the advances of research on the topic of Recommenders published in journals and discussed at international conferences.

List of Tables

List of Figures