

The Battle of Neighborhoods

1. Introduction

Imagine Gordon Ramsay is looking to start a new fancy Italian restaurant in London. However, London is crowded to the brim with restaurants of different kinds and as the owner, he is naturally looking to ensure that the restaurant is profitable. If the restaurant is placed in an area which is already crowded with restaurants catering to the same customer segment, it will be battling the competition for guests. A better placement might be an area which is lacking restaurants (particularly Italian ones). However, besides low competition, Gordon also needs customers who can afford the food – and preferably many of them. A solution to this problem is to use data science to give recommendations of areas (in this case, we choose to limit our fidelity to borough-scale) where the ratio of competition to customer spending proclivity is low compared to other areas. The level of competition can be measured by the number of Italian restaurants in the borough. Customer spending proclivity can be estimated by calculating the median income per square kilometer, which ensures that both the individual income and population density is considered in a way which we can relate to ‘total amount of money that people who live nearby can spend’.

2. Data

One of the main datasets which will be used to solve the problem is Foursquare location data, which includes any venues – coffee shops, restaurants, shops, markets, hotels, gyms, parks and more – found within a certain specified radius of a set of coordinates [1]. A list of London boroughs and neighborhoods will be scraped from the Wikipedia page ‘List of areas of London’ [2]. Coordinates for each neighborhood will then be acquired through the Geocoder Python package [3]. To assess the suitability of each borough, Average Income of Taxpayers [4] and Land Area and Population Density [5] will be combined to calculate a measure of spending power per square km. These two datasets are provided by the Greater London Authority at the London Datastore.

3. Methodology

3.1 Data acquisition

3.1.1 Income of Taxpayers and Population Density

The Average Income of Taxpayers per Borough and Population Density per Borough datasets are provided as excel documents on the Longon Datastore website and could be downloaded directly into Pandas dataframes.

3.1.2 Number of restaurants

The Wikipedia page ‘List of areas of London’ contains a record of neighborhoods and their postal codes for each London borough. This data could easily be scraped using the BeautifulSoup package. Although the income per taxpayer and population density datasets only contain data at borough level, we require higher resolution when using the Foursquare API to find the number of Italian restaurants per borough. This is because each call to the API is limited to a maximum number of results in the response because we are using a free developer account. If we called the API only once per borough, we would risk only getting a small sample of the restaurants.

The next step was to get the coordinates for each neighborhood, which we could then use to call the Foursquare API. Coordinates were reverse geocoded from postal codes using Geocoder. This API is

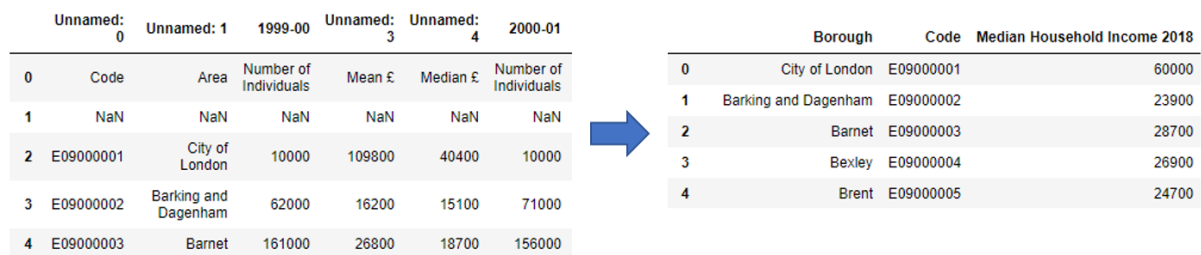
notoriously unstable and will often return only 'None'. This was circumvented by writing a function, *get_latlng*, which employed a while loop to call the API for the input address until a real return occurs. The resulting list was then cast into a Pandas dataframe containing the neighborhood name, borough, and coordinates.

3.2 Data cleaning

Data cleaning proved to require the bulk of effort in this project, as no advanced machine learning was necessary to analyze the cleaned data.

3.2.1 Income of Taxpayers and Population Density

These datasets contained a plethora of data for a range of years as well as aggregated data for larger rural areas, Wales, Scotland, etc. For the project, only median household income, size in square kilometers and population per square kilometer data was needed for London boroughs. For the Income of Taxpayers dataset, that meant only the column for Median Household Income 2018 was kept and all other columns were filtered out. For the Population Density dataset, the area size in square kilometers and the population per square kilometer were kept and other columns filtered. Both dataset had the same structure of rows, with each area identified by a code. It was noticed that all London borough codes began with 'E09', so unwanted rows were filtered out using a Boolean operation on the first three characters in the Code column.



0	Code	Area	Number of Individuals	Mean £	Median £	Number of Individuals
1	NaN	NaN	NaN	NaN	NaN	NaN
2	E09000001	City of London	10000	109800	40400	10000
3	E09000002	Barking and Dagenham	62000	16200	15100	71000
4	E09000003	Barnet	161000	26800	18700	156000

	Borough	Code	Median Household Income 2018
0	City of London	E09000001	60000
1	Barking and Dagenham	E09000002	23900
2	Barnet	E09000003	28700
3	Bexley	E09000004	26900
4	Brent	E09000005	24700

Figure 1: Example of income data before (left) and after (right) filtering.

It was also noted that some multiple-word Borough names in the Income of Taxpayers dataset had hyphens, for example 'Richmond-upon-Thames'. This was not the case for the Population Density dataset. Hyphens were removed, allowing the two dataframes to be merged on the Borough column.

	Borough	Code	Square kilometres	Population per km^2	Median Household Income 2018
0	City of London	E09000001	2.90393	2645.03	60000
1	Barking and Dagenham	E09000002	36.1078	5892.71	23900
2	Barnet	E09000003	86.7483	4577.02	28700
3	Bexley	E09000004	60.5807	4126.71	26900
4	Brent	E09000005	43.2326	7791.78	24700

Figure 2: Cleaned and merged dataframe containing Income of Taxpayers and Population Density data.

Lastly, the latitude and longitude of the borough centers were called from Geocoder using the *get_latlng* function and added to the dataframe.

	Borough	Code	Square kilometres	Population per km ²	Median Household Income 2018	Latitude	Longitude
0	City of London	E09000001	2.90393	2645.03	60000	51.520500	-0.097430
1	Barking and Dagenham	E09000002	36.1078	5892.71	23900	51.543932	0.133157
2	Barnet	E09000003	86.7483	4577.02	28700	51.527095	-0.066826
3	Bexley	E09000004	60.5807	4126.71	26900	51.452078	0.069931
4	Brent	E09000005	43.2326	7791.78	24700	51.609783	-0.194672

Figure 3: Income of Taxpayers and Population Density data including borough coordinates.

3.2.2 Number of restaurants

It was noticed that some neighborhoods from the Wikipedia page were assigned to multiple boroughs, indicated by a comma.

	Neighborhood	Borough	PostalCode
0	Abbey Wood	Bexley, Greenwich [7]	SE2
1	Acton	Ealing, Hammersmith and Fulham[8]	W3, W4
2	Aldgate	City[10]	EC3
3	Aldwych	Westminster[10]	WC2
4	Anerley	Bromley[11]	SE20

Figure 4: Example of neighborhood assigned to more than one borough.

In this case, venues (restaurants) returned from Foursquare when calling the coordinates for Abbey wood would have to be assigned to either Bexley or Greenwich. To solve this issue, a function called *get_borough* was written. The function took the venue coordinates, the neighborhoods multiple assigned borough names, and a list of all London boroughs and their coordinates as inputs. Euclidian distance (i.e Pythagoras theorem) was used to approximate the distance from the venue to each of the borough centers using the coordinates. The borough with the minimum distance was then returned and the venue was assigned to this borough in the dataframe.

One more function was necessary to process the JSON files returned by Foursquare: *get_nearby_venues*. This function took a list of neighborhoods, their boroughs and coordinates as input. It then called the Foursquare API using *requests* on each neighborhood, using a search distance of 2 km. Foursquare then returns a list of venues within the radius of the coordinates supplied, including the venue coordinates and category. The *get_borough* function was then called to assign the venue to the correct borough, before appending the data to a dataframe.

The *get_nearby_venues* function was called using the neighborhood data scraped from Wikipedia. The resulting venues dataframe had a total of 28989 entries with 322 different categories. Because we were interested only in restaurants selling Italian cuisine, the dataframe was filtered for categories containing the strings 'italian' or 'pizza'. Any duplicate rows were removed. The result had a total of 545 pizza places and 477 Italian restaurants. Then, we grouped the venues by Borough and got the count – the number of restaurants for each borough. These values were then added to the borough dataframe containing the income and population data. Foursquare did not return any Italian restaurants for four of the boroughs – Barking and Dagenham, Havering, Hillingdon and Sutton. These entries were ignored in the further analysis.

Lastly, two performance parameters were added by calculating the number of restaurants per square kilometer and the income per square kilometer (population density * median household income*) for each borough. The header of the resulting dataframe is shown in Figure 5.

	Borough	Count	Square kilometres	Population per km ²	Median Household Income 2018	Latitude	Longitude	Venue Density	Income Density
0	Barnet	103.0	86.7483	4577.02	28700	51.627300	-0.253760	1.18734	1.31361e+08
1	Bexley	3.0	60.5807	4126.71	26900	51.452078	0.069931	0.0495207	1.11009e+08
2	Brent	60.0	43.2326	7791.78	24700	51.609783	-0.194672	1.38784	1.92457e+08
3	Bromley	11.0	150.135	2216.23	32000	51.601511	-0.066365	0.0732674	7.09193e+07
4	Camden	76.0	21.7893	11594.5	37300	51.591180	-0.165040	3.48795	4.32477e+08

Figure 5: Header of the final dataframe, ready for analysis. 'Count' indicates number of italian-related restaurants (categories 'pizza' and 'italian' summed)

4. Results

An exploratory data analysis was carried out by creating rankings for the two performance parameters. Boroughs were ranked on the number of restaurants per square kilometer (low is better) and cumulative income per square kilometer (high is better). The rankings are shown in Figure 6, where rank 1 is the best and 29 is the worst.

	Borough	Venue Rank	Income Rank		Borough	Venue Rank	Income Rank
0	Barnet	14.0	23.0	14	Hounslow	2.0	24.0
1	Bexley	4.0	27.0	15	Islington	25.0	1.0
2	Brent	16.0	14.0	16	Kensington and Chelsea	27.0	2.0
3	Bromley	5.0	29.0	17	Kingston upon Thames	1.0	20.0
4	Camden	28.0	6.0	18	Lambeth	17.0	7.0
5	City of London	29.0	19.0	19	Lewisham	20.0	12.0
6	Croydon	6.0	26.0	20	Merton	11.0	17.0
7	Ealing	8.0	16.0	21	Newham	18.0	13.0
8	Enfield	7.0	28.0	22	Redbridge	9.0	21.0
9	Greenwich	12.0	18.0	23	Richmond upon Thames	10.0	25.0
10	Hackney	19.0	5.0	24	Southwark	22.0	9.0
11	Hammersmith and Fulham	23.0	8.0	25	Tower Hamlets	26.0	3.0
12	Haringey	24.0	11.0	26	Waltham Forest	13.0	15.0
13	Harrow	3.0	22.0	27	Wandsworth	21.0	10.0
				28	Westminster	15.0	4.0

Figure 6: Boroughs ranked by venue density and income density.

The results were then plotted on a Folium map, using a colormap to display the rank of venue density as border and rank of income density as fill for the markers. The result is shown in Figure 7.

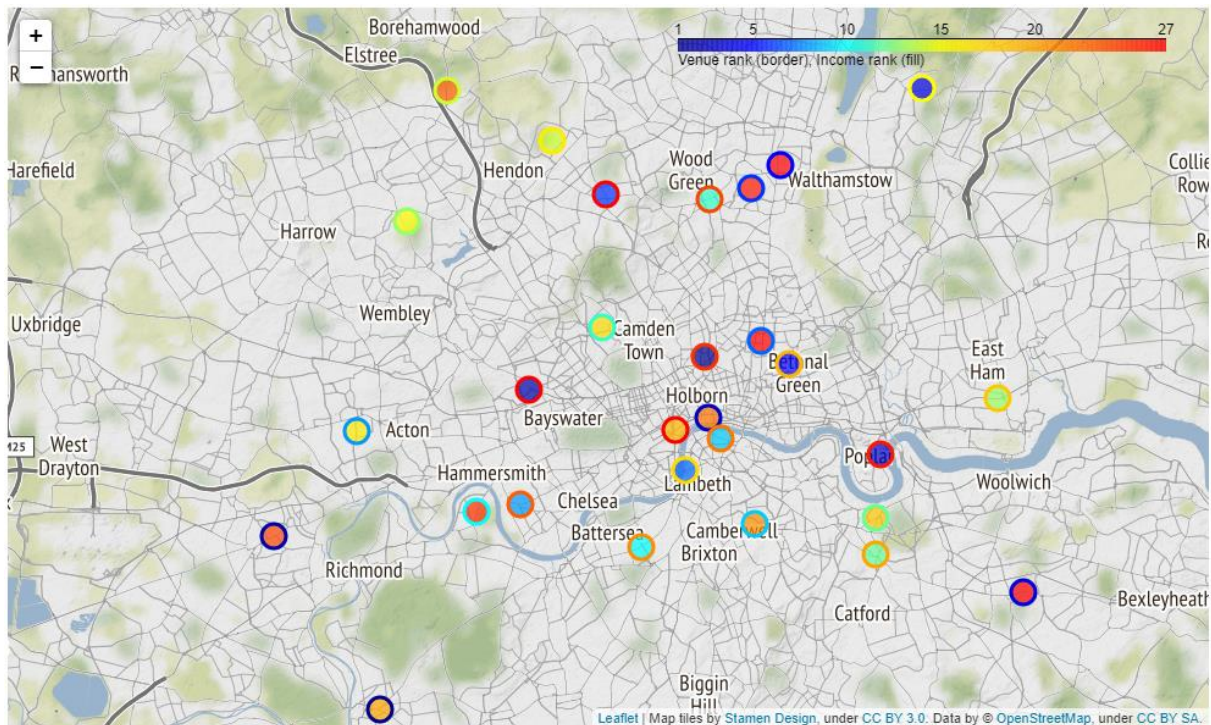


Figure 7: Heat map showing attractiveness of boroughs by their rank in venue density (border) and income density (fill).

To find the optimal borough for Gordon Ramsay's new restaurant, we want to find boroughs with high rank of both venue density and income density – i.e markers with both border and fill on the blue side of the spectrum. Based on the interactive Folium map and the table, some promising boroughs are Ealing, Hackney, Lambeth, Waltham Forest and Westminster.

However, by merely looking at the rankings we do not consider how 'close' the difference between one rank and another may be. A better measure may be to plot the relationship between venue density and income density. Such a plot is presented in Figure 8. A linear regression trendline was added in black.

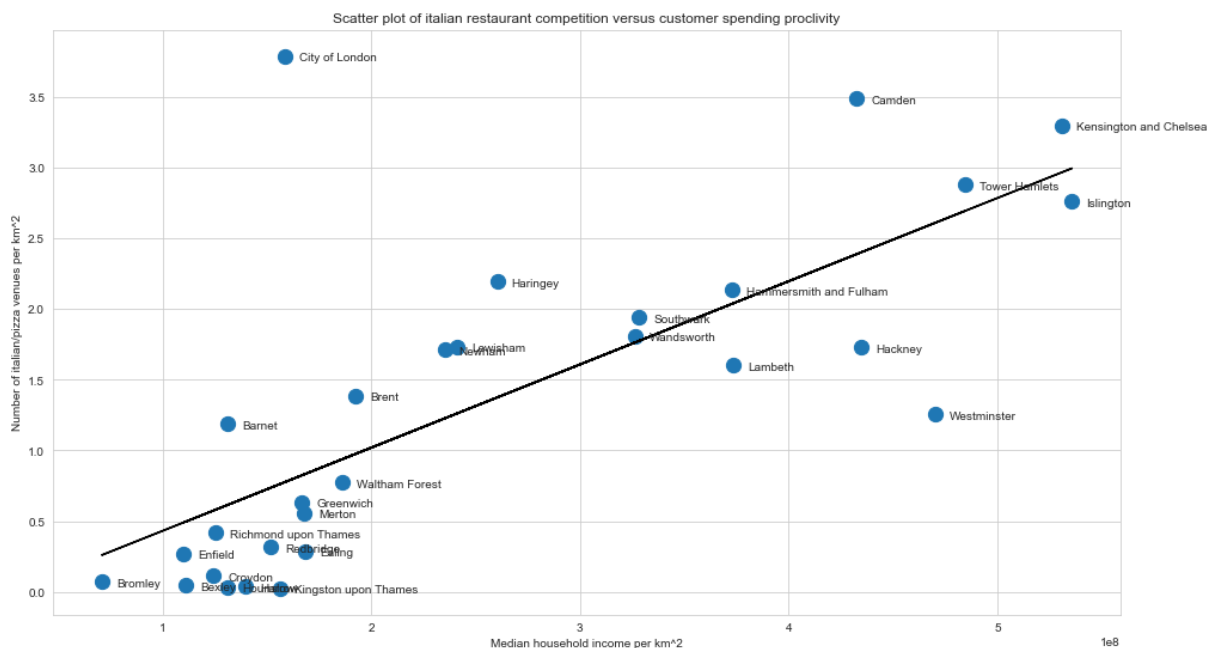


Figure 8: Scatter plot of borough income density and venue density. Linear regression trendline indicated in black.

5. Discussion

Boroughs which are clear outliers on the right side of the line indicate a high income density but low restaurant density compared to other boroughs, which is exactly what we are looking for. Moreover, we are interested in areas with high income to spend on Gordons high end restaurant. It is clear that Westminster is our best choice. Lambeth, Hackney and Islington are also good choices. However, the previously mentioned Ealing and Waltham Forest are positioned in the cluster to the bottom left, which indicates a relatively low income density – likely not ideal for a high end restaurant. These boroughs may be well suited to a smaller family owned restaurant with a different target audience.

This analysis was based on three main parameters; Number of restaurants, income and population density. It could be improved by including more parameters which can have an impact on the suitability of a particular area for starting a restaurant. For example, the cost of property, availability (e.g closeness to parking, public transport) or even the density of other restaurant types which could compete for customers. Lastly, the fidelity could be increased to the neighborhood level instead of borough – but publicly available data on this level seems difficult to come by, as a fair amount of web searching did not give any results for us.

6. Conclusion

In this project, data from the UK government, Wikipedia, Geocoder and Foursquare was used to analyze the ideal London borough for Gordon Ramsay to start a new high end Italian restaurant. Parameters used were number of competing Italian restaurants per square kilometer and (potential customer) cumulative income per square kilometer. The data indicated Westminster to be the best suited borough, with alternatives being Lambeth, Hackney and Islington. Further analysis should be conducted on the neighborhood level and including a greater range of parameters, for example cost of property and number of competing non-italian restaurants.

References

- [1] <https://developer.foursquare.com/>
- [2] https://en.wikipedia.org/wiki/List_of_areas_of_London
- [3] <https://geocoder.readthedocs.io/index.html>
- [4] <https://data.london.gov.uk/dataset/average-income-tax-payers-borough>
- [5] <https://data.london.gov.uk/dataset/land-area-and-population-density-ward-and-borough>