Coursera Capstone Project
Filling Barons Quay

Ashley Donson

11th May 2019

Introduction

Background

Northwich is a town in Cheshire, England with a population of approximately 75,000. Cheshire West and Chester Council (CWaCC) recently invested £80m to develop the Barons Quay shopping centre with brand new, glass-fronted retail units in order to inject life into the sleepy town centre, encouraging locals to shop closer to home and those further afield to travel to this new hub.

Problem

Unfortunately, as of May 2019, 40% of the retail space remains unused. Walking through the area it is easy to see the potential CWaCC saw in the centre, but with several units left bare and few prospective tenants, the development is not living up to its potential.

With reported loan interest payments of £40,000 per week, CWaCC should have great interest in finding businesses that would like to open a location in one of the lots; the sooner the available units are filled and the shopping centre is alive with business, the sooner the council see a positive return on their investment and the growth they want for the town.

Proposal

To see where Northwich can improve as a retail hub, one solution is to find what it is lacking in comparison to other towns in England. By analysing which businesses are commonplace in other towns but have no presence in Northwich, CWaCC may develop a better understanding of the direction in which they need to head in order to see improvement.

The goal

The end-goal of this exercise is a dataframe of businesses that CWaCC might approach, detailing the ubiquity of the business in other towns and how far away their nearest retail presence is from Barons Quay. The retailers of particular interest are those with an abundance of locations in other towns, but where a shopper in Northwich would have to travel a significant distance to visit their nearest location.

Data

Data sources

In order to achieve this goal, two types of data will need to be used:

1. Population data for English towns, in order to discover which towns are appropriate for comparison
2. Location data for the relevant towns to discover which retail businesses operate in their central regions

One Wikipedia article conveniently lists hundreds of English towns that can be compared to Northwich in an article named "List of towns in England". The town names are held in several tables, separated alphabetically. Each town links to its own Wikipedia article, wherein population statistics are usually held within the key facts sidebar and location datapoints are tagged in the page source with `<span class="geo">`

Figure 1: Population data in the page, left; Corresponding population data in the source, right

Iterating through the tables in the page, it is possible to extract population, latitude and longitude values for other towns in England and store them in a Pandas dataframe. With the location data, we can make API calls to Foursquare to record businesses within 3km of each coordinate presence in multiple areas. This will generate a new dataframe with company names.

After some manipulation, this dataframe will be reduced to only one record per business, detailing the number of occurrences of that business in the comparison towns. With this list of businesses entered back into Foursquare – this time with Barons Quay as the location centre – a final dataset will be produced of the distance to each business' closest location to the potential new spot in Northwich.

Data assumptions

For the population data, there are expected to be several degrees of inaccuracy in the results. The statistics on Wikipedia, while accompanied by citations, are prone to error and inconsistency from the open-source nature of the website; any data extracted is the result of the person writing the article and could be exaggerated, out of date, or subject to different criteria, e.g. the article for one town might have the population of the town proper while another has the population of the metropolitan area.

Fortunately, while the population data will determine the selection of towns to investigate, the statistics themselves will not be presented in the final result. We can accept that the population values are only a rough measure, as the information is only used to generate a sample of 100 large towns.

For the location data, Foursquare is the world leader in geolocating retail businesses, so it is unlikely that there is another data source available to provide more accurate data, short of collecting it first-hand for this report. Any errors in the first location dataset – retailers operating in towns in England – should be mitigated by the large sample of 100. Errors in the dataset of businesses' distances from Northwich will prove more difficult: a mislabelled datapoint here could lead us to believe that a retailer has no shop near Northwich where one already exists. As such, this data will be especially scrutinised, manually identifying and handling any problems thanks to my local knowledge.

Methodology

Extracting the list of towns

To extract the necessary town information from the "List of towns in England" Wikipedia article, I first used the `Requests` package in Python to covert the page HTML into a Python string that can be sliced. By exploiting the rigid table formatting within the page, particularly the `<tr>` tag denoting a new table row, I was able to iterate through the rows, slice the article link, slice the town name, then add these to a new row in dataframe `towns_list_df`.

It was important to exclude Northwich itself from the data, as the later location data would then be compromised, so the row matching the town name Northwich was not added to the dataframe.

The result was `towns_list_df` listing the name and Wikipedia article link for 965 English towns.

```
towns_list_df.head()
```

Out[4]:

	Town	Link
0	Abingdon-on-Thames	/wiki/Abingdon-on-Thames
1	Accrington	/wiki/Accrington
2	Acle	/wiki/Acle
3	Acton	/wiki/Acton,_London
4	Adlington	/wiki/Adlington,_Lancashire

Figure 2: A sample of the resultant dataframe

Finding town populations and coordinates

Following the method of extracting the town details above, to extract the population I also used `Requests` to set each page's HTML into a string variable, this time using the links extracted previously.

For each town in the dataframe, I queried the HTML to find a population value held in the sidebar as shown in *Figure 1*. There were three common formats used across the 965 articles that allowed me to pull the statistic successfully for most of the towns with only three searches through the string. I sliced the population string from the greater text, removed any commas separating the thousands, and cast the value to an integer for numerical handling later on. This became the variable `Population`.

Similarly, the latitude and longitude values for each town were held in the sidebar, but this time in a much more standardised format. Searching for the `<span class="geo">` tag I was able to slice the values from the page and assign them to variables `Latitude` and `Longitude`. `Population`, `Latitude` and `Longitude` were then added to the dataframe `towns_df`.

Where the population could not be found with one of the search methods used, `Population` was set to the value "`ERROR`" and the town name was printed for review.

```
No population value found for Ashby Woulds
No population value found for Barking
No population value found for Bexley
No population value found for Blackwater and Hawley
No population value found for Braunstone Town
No population value found for Broadstairs and St Peter's
No population value found for Bude-Stratton
No population value found for Chingford
No population value found for Corringham
No population value found for Dartford
No population value found for Dovercourt
No population value found for Finchley
No population value found for Great Torrington
No population value found for Harlow
No population value found for Harworth and Bircotes
No population value found for Heathfield
No population value found for Hendon
No population value found for Ipswich
No population value found for Liskeard
No population value found for Lytchett Minster &amp;amp; Upton
No population value found for Newlyn
No population value found for Northleach with Eastington
No population value found for Ollerton and Boughton
No population value found for St Mawes
No population value found for Selsey
No population value found for South Kirkby and Moorthorpe
No population value found for Southend-on-Sea
No population value found for St Mary Cray
No population value found for Swanscombe and Greenhithe
No population value found for Watford
No population value found for West Tilbury
No population value found for Willesden
```

Figure 3: The towns with no population data found

There were relatively few towns with a missing population value – 32 out of 965 – so it was safe not to consider these towns for the rest of the exercise; rows with a `Population` value of "`ERROR`" were therefore stripped, leaving 933 towns with a known value.

In order to only focus on large towns – towns worth comparing to Northwich – those with a `Population` value in the largest 100 were extracted and set to a new dataframe `top100`. This had a population range of 62,500 (Sittingbourne) to 225,700 (Northampton).

Determining popular retailers

With latitude and longitude data for 100 towns, I was able to query the Foursquare API to return relevant local venue data for each town. Plugging in the coordinates, I requested the details of as many venues as possible within 3km of each town that were tagged "shop & service" or "food", as these would be appropriate tenant types for Barons Quay.

The API calls each returned data in the JSON format, which I sliced to extract venue name and type to the variables `Name` and `Type`. For each venue result, a row was added to the new dataframe `retailers_df`, alongside the `Town` value for that API query; this dataframe consisted of 7,653 retail locations for the 100 towns.

In order to group the retailers into operating businesses, it was necessary to manipulate the `Name` values. The first priority was to convert all lowercase characters to uppercase, remove any instances of the `Town` matched within the `Name` value, then strip any trailing/leading spaces.

That next problem was retailers that brand their stores in slightly different ways, e.g. Tesco Extra, Tesco Express and Tesco Metro. For the purposes of this exercise, these should be treated as the same retailer, so I iterated though the `Name` values, checked whether each was a match for one of the affected retailers, then replaced it with the standardised name if a match was made.

With this clean list of retailers, the data could now be grouped. A new dataframe, `retailers_grouped`, was generated with the columns `Retailer` and `Count`, wherein `Retailer` is the business name and `Count` is the number of locations for that business in the 100 towns. After dropping the "`MARKET`" row – a result of stripping the town names from the venue name – the dataframe held 2,302 rows.

To focus on only popular business and avoid making over 2,000 API calls, rows with a `Count` value of less than 10 were then stripped, leaving the dataframe with 97 popular retailers.

```
retailers_grouped.head(10)
```

Out[18]:

	Retailer	Count
0	TESCO	403
1	CO-OP FOOD	293
2	SAINSBURY'S	209
3	BOOTS	150
4	MARKS & SPENCER	135
5	WHSMITH	129
6	ARGOS	119
7	LIDL	116
8	ASDA	110
9	ALDI	101

Figure 4: The most popular retailers

Finding business proximity to Northwich

With a list of 97 potential tenants and their approximate popularity, all that was needed was their closest distance to Northwich where CWaCC might propose that the businesses open up a new location.

For each `Retailer` in the dataframe, a final request was made to Foursquare. Using the coordinates of Barons Quay as the latitude and longitude parameters, I conducted a search with no maximum distance bound and extracted the distance in metres held within the resulting JSON output; this result was assigned to the variable `Closest` and committed to the dataframe.

When viewing the head of the dataframe, I immediately noticed a problem. The closest location for the retailer Boots was showing as over 15km away, when in reality I know there to be a branch of Boots in Northwich less than 200m away. This proved there were some inaccuracies with the location data. Perhaps that particular location within Foursquare's database followed a different naming convention or was missing altogether.

For this exercise, the goal was to build recommendations rather than strict facts, so the solution was to manually investigate the rows with a `Closest` value over 10,000 and cut out any that I knew had locations in Northwich, to avoid tainting the final pool. Seven of the 97 retailers were cut as a result.

While removing these rows with an existing Northwich location, I also cut all rows with a `Closest` value less than `1000.0`; if these retailers already have a presence so close to Barons Quay, they would not be a priority business to approach. 20 of the remaining 90 retailers were cut as a result.

For visualisation and modelling, I then normalised the data to a value between `0` and `1.0`. The normalised `Count` column became `Popularity`, and the normalised `Closest` column became `Distance`. These new values were then plotted to a scatter diagram to observe the results.

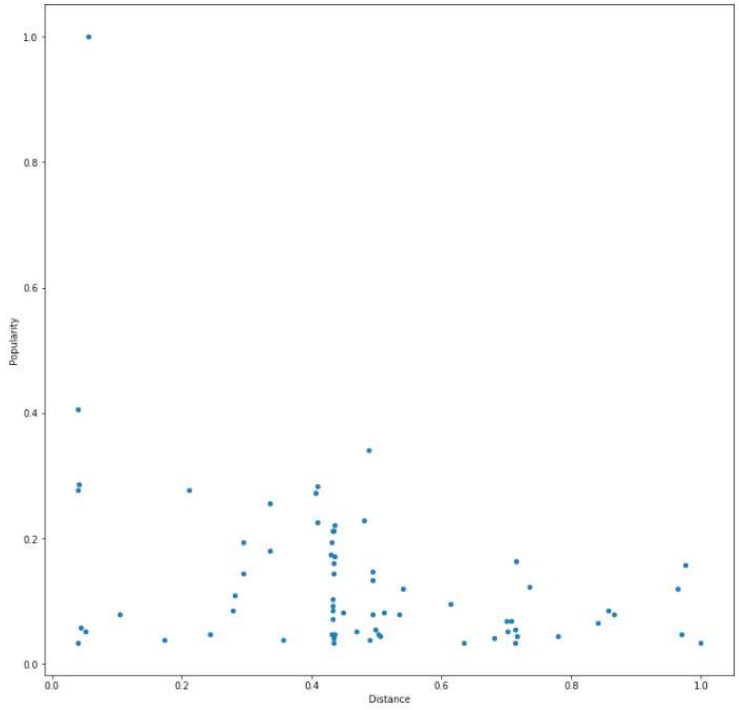

Figure 5: The scatter plot with normalised values from 0 to 1

Clustering using K Means

With 70 datapoints spread out with different levels of popularity and distance, it would be beneficial to cut the dataframe down further and retrieve a smaller subset to deliver as the final result. One way to do this is by clustering the points.

The hypothetical perfect result would be business that is highly popular in other towns with its nearest retail presence extremely far away from Barons Quay. By leveraging K Means clustering with starting centroids set to the four extremes of the above plot, I was able to assign a cluster label to each point and pick out those most similar to the ideal `[1.0, 1.0]` retailer.

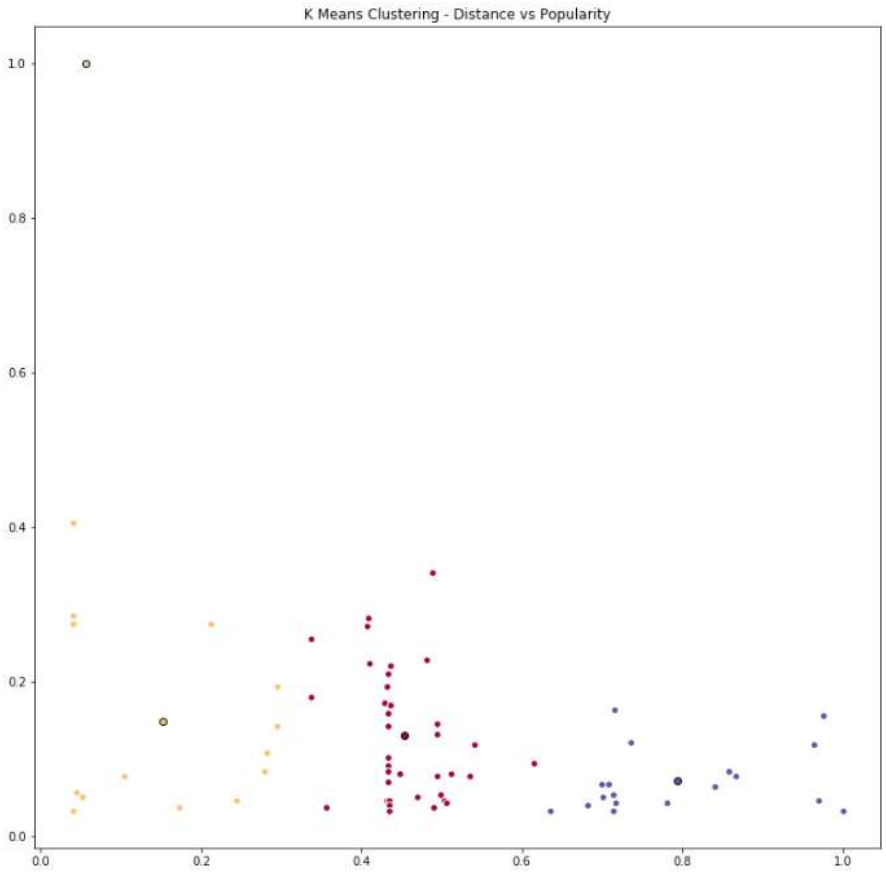

Figure 6: Visualisation of the K Means clustering outcome

The central group in *Figure 6* displays the cluster assigned to the starting centroid `[1.0, 1.0]` and is therefore the group with the greatest overall balance of `Distance` and `Popularity`. By adding these labels into the dataframe and extracting rows with a `Classification` value of `0` I was able to generate what became the final dataframe, `Prospective_Retailers`.

To display the final dataframe in some significant order, I created the new column `Score` that was a summation of the two normalised values plotted previously, a number somewhere between `0` and `2.0`. The dataframe was then sorted with this value in descending order to give the best `Score` first.

Results

Overview

The final dataframe has 37 prospective retailers with a `Count` value between `10` and `100`, a `Closest` value between `11774.0` and `21502.0`, and a `Score` between `0.39399` and `0.82974`.

The list

Retailer	Count	Closest	Score
LLOYDSPHARMACY	100	17096	0.82974
ONE STOP STORES LTD	28	21502	0.709888
HOME BARGAINS	67	16815	0.709084
TK MAXX	83	14298	0.691779
WILKO	80	14235	0.67974
TOPMAN	35	18929	0.660267
VISION EXPRESS	65	15248	0.657488
PRIMARK	62	15171	0.645049
VODAFONE	62	15156	0.64462
F&F CLOTHING	43	17284	0.640572
POUNDLAND	66	14324	0.634501
DUNELM	39	17292	0.627149
O2 SHOP	57	15105	0.626098
NATIONAL TYRES AND AUTOCARE	23	18744	0.614026
DEBENHAMS	50	15242	0.606122
CEX	51	15030	0.603478
RIVER ISLAND	47	15180	0.594111
COHENS CHEMIST	24	17887	0.592954
CARPHONE WAREHOUSE	75	11774	0.592363
TOPSHOP	42	15177	0.576961
HOBBYCRAFT	23	17277	0.572113
MOTHERCARE	16	17467	0.553650
GRAINGER GAMES	14	17625	0.551339
HOUSE OF FRASER	13	17698	0.550011
EUROCHANGE	30	15156	0.535405
MAJESTIC WINE	24	15686	0.530070
NISA	11	17129	0.526929
HMV	27	15150	0.524995
JOHNSON CLEANERS	15	16429	0.520581
CARD FACTORY	25	15146	0.518055
GAME	53	11776	0.517335
GEEK SQUAD	21	15162	0.504860
LUSH	14	15225	0.482769
VIRGIN MEDIA STORE	14	15112	0.479541
THREE	12	15194	0.475058
PAPERCHASE	10	15191	0.468146
MCCOLL'S	11	12476	0.393990

Discussion

Scope for improvement

With knowledge of the Barons Quay site and the types of retailers listed in the final dataframe, it is clear that some of the final suggestions would be less appropriate than others for a tenancy in the shopping centre.

The retail units are mostly large, with glass fronts for extensive displays. As such, the second suggestion ONE STOP STORES LTD – a chain of small convenience stores – is not likely to be a good fit and not a retailer to approach regarding a tenancy. The units are also mostly in a pedestrianised area, so NATIONAL TYRES AND AUTOCARE – an automotive retailer that tends to place itself in industrial estates – would also be unsuitable.

Now the model has been created, it is simple to fine-tune the details to alleviate this problem of inappropriate retailers. By discussing the stakeholders' needs to find out whether they have preferred types of tenant, the Foursquare categories used in the exploratory analysis can be changed from the general "shop & service" and "food" to something more specific like "Bookstore", "Clothing Store" and "Hobby Shop", for example.

Another potential issue is that, in plotting and modelling the data, the `Distance` and `Popularity` were not weighted before normalising, so a 0.2 difference in the `Distance` score is treated as equally valuable to a 0.2 difference in the `Popularity` score. While this is not necessarily a problem in itself, the reality is likely that the stakeholders would value one over the other. If, for example, CWaCC were more concerned with a potential retailer's popularity, this value could be squared before normalisation to give higher values higher weighting. Again, this is fine-tuning that could be solved alongside the interested parties.

One final problem with the model is the result of using K Means with four predetermined centroids. Figure 6 shows that it was a good way of finding which retailers were both popular and distant, but it becomes clear that this method was not the ideal for maximising both values, as these were not the only datapoints clustered into that category. For example, there are some retailers in the central category that have similar `Popularity` values to those in the far-right category, but these in the latter in fact have a much higher `Distance` value and could perhaps be more valuable targets.

If this problem needed to be solved, one might reattempt to cluster using K Means, but instead using only the two extremes `[1.0, 1.0]` and `[0, 0]` to group them as "of interest" and "not of interest". Alternatively, by forgoing machine learning altogether, one could attempt to score each retailer – as done to reorder the final dataframe – and choose only the required number of retailers from the highest scores in that list, effectively choosing *N* number of points with the minimum distance from the ideal `[1.0, 1.0]`.

Successes

Despite its faults, the results produced by this exercise did highlight several businesses whose absence from a town as large as Northwich is conspicuous. Popular retailers such as TK Maxx, Primark and Dunelm could meet local demands if they had a tenancy at Barons Quay and they typically have large premises that could fill the spacious lots available.

Even without refining the model, it is possible to make recommendations to CWaCC regarding which businesses they should approach to take up residence at the site and there are good variety of business types to choose from, should they wish to avoid filling the centre with competing businesses.

With general goods stores Home Bargains and Poundland, clothing shops such as Topman, Primark and River Island, and hobby shops Hobbycraft and Grainger Games, Barons Quay could become a retail hub for locals and those further afield alike.

Conclusion

Through this exercise, I identified a substantial list of businesses that are popular in England who do not have a retail presence close to Northwich. Cheshire West and Chester Council would now be able approach the desired business in turn in an attempt to fill the 40% of units that are vacant and receive a better return on their investment.

The final list ought to be sufficient to decide which retailers would be a good fit, with the popularity and distance values ascertained providing a strong opening argument to the business for opening up a location at Barons Quay.

However, if CWaCC would prefer to further refine the final list, the model is open to easy amendments, allowing them to target different retailer types or prioritise one of the statistics over the other as required.