EXPLORING LONDON

IBM DATA SCIENCE PROFESSIONAL CERTIFICATE CAPSTONE PROJECT

KATE REES – SEPTEMBER 2020

Exploring London

Introduction

I live and work in London, UK: currently ranked as the 19th most expensive city in the world to live in. I am interested in exploring whether location data, such as that available with Foursquare, can predict which parts of the city are more expensive, and which are least expensive.

Well-documented in the UK press is the so-called 'Waitrose Effect[1]': house prices near a Waitrose store – a high-end supermarket chain in the UK – are higher. I am interested in exploring this idea further. I intend to use Foursquare's API to identify concentrations of different categories of amenities in each area, and combine it with house price data to look for correlations, and use data science methodology to explore whether we can use Foursquare's data to predict house prices in a given London ward.

The results of this project would be of interest to anyone wanting to invest in property in London. House buyers could use the results spot 'up and coming' areas before house prices rise, and property companies could use the results to decide where to invest in new building projects - and to estimate how much apartments might sell for in that area.

[1] https://www.propertywire.com/news/uk/research-reveals-waitrose-effect-uk-house-prices/

Data

Data Source 1: London House Price Data

The Mayor of London's office publishes statistics on house prices for each London Borough, and subdivided into Wards. It is available to download from https://data.london.gov.uk/download/average-house-prices/59be940c-ffb8-426d-a833-6146ea77de5c/land-registry-house-prices-ward.csv

This house price data contains values (in GBP) of both Mean and Median house prices in each ward. To decide which metric to use, I compared the distribution of both, and found that the median was was closer to a normal distribution and therefore less affected by outliers – e.g. a few very expensive properties.

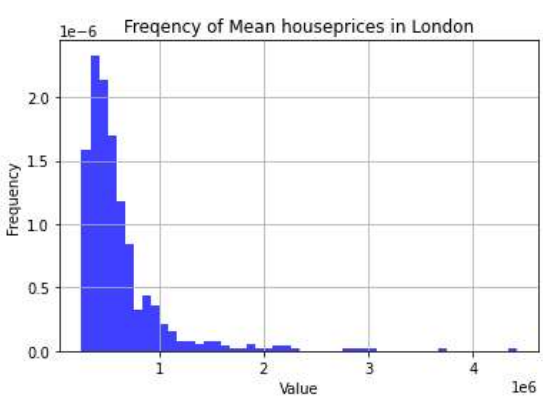
A sample of the data is shown below:

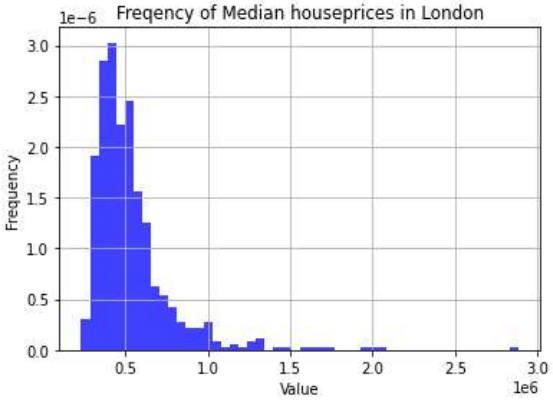
	Code	Ward_name	Borough	Year	Measure	Value
55440	E09000001	City of London	City of London	Year ending Dec 2017	Median	-
55441	E05000026	Abbey	Barking and Dagenham	Year ending Dec 2017	Median	231,000
55442	E05000027	Alibon	Barking and Dagenham	Year ending Dec 2017	Median	295,000
55443	E05000028	Becontree	Barking and Dagenham	Year ending Dec 2017	Median	300,000
55444	E05000029	Chadwell Heath	Barking and Dagenham	Year ending Dec 2017	Median	310,000

As well as Ward name and Borough, there is also a Code unique to each borough. This was useful as a key, as I found that Ward name had spelling differences (e.g. 'and' vs. '&') between the house price data and the geographic data.

Missing data:
Notice there is no value for house prices in the City of London – this is the central business / financial district, and falls under a different jurisdiction to the rest of Greater London. I decided to ignore the City of London for the purposes of this project.

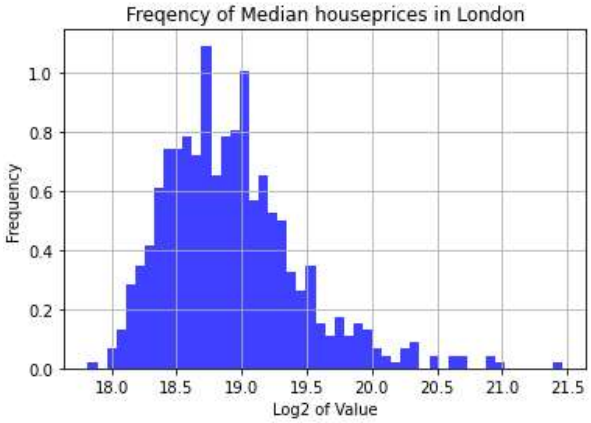
The house price data contained values for mean and median house prices for each time period, and as part of my analysis I had to decide which would be the best to use. To do this, I used histograms to visualise the distribution of each:





Both show a skewed distribution, and show that a small number of very high value properties are likely responsible for this. The median values were slightly better distributed, however, and I decided to use the median for the remainder of the analysis.

One way to account for this would be to use a logarithmic scale. I also histogrammed the $\log_2$ of the median values – this showed a distribution much closer to a normal distribution.



However, the log scale is not particularly intuitive to interpret. Instead, I decided to divide the median values into five equally sized 'bins' using the quantile function.

	Bin Range in GPB
Lowest	231000 - 369000
Low	369995 - 435250
Medium	436250 - 518750
High	520000 - 635000
Highest	637500 - 2882600

This has the benefit of being easy to interpret, and the 'highest' category accumulates the high value outliers that were causing the skew in our distribution plots. This scale was used to produce the choropleth map in the section below.

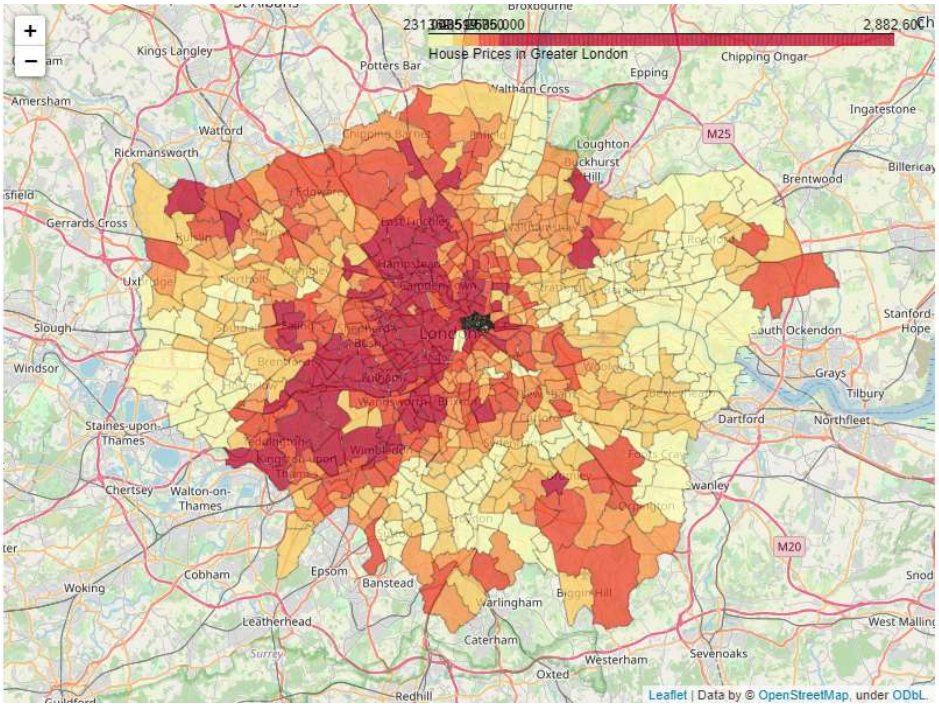
Data Source 2: London geographic data

The house price data does not contain details of the geographic position of the Wards and Boroughs, so we will need a separate data source for this.

I found a geoJSON file with boundary data for London wards on the Financial Times' visual and data journalism Github repository at https://github.com/ft-interactive.

	ward	gss_code_ward	gss_code_borough	borough	geometry
0	Chessington South	E05000405	E09000021	Kingston upon Thames	POLYGON ((-0.33068 51.32901, -0.33059 51.32909...
1	Tolworth and Hook Rise	E05000414	E09000021	Kingston upon Thames	POLYGON ((-0.30846 51.37586, -0.30834 51.37606...
2	Berrylands	E05000401	E09000021	Kingston upon Thames	POLYGON ((-0.30385 51.39249, -0.30375 51.39252...
3	Alexandra	E05000400	E09000021	Kingston upon Thames	POLYGON ((-0.26990 51.38845, -0.26975 51.38838...
4	Beverley	E05000402	E09000021	Kingston upon Thames	POLYGON ((-0.24662 51.39921, -0.24672 51.39921...

This will allow us to define the boundaries of each London ward. The Choropleth Map below shows the London Wards coloured by their house price value bin. The black region is the City of London, for which we have no data.



I used the Geopandas library to process the GeoJSON, as whilst it contained the geometry of the wards – the boundaries used to create the choropleth map above – to work with the Foursquare API we need a point to represent a rough centre of each ward.

Geopandas has a centroid function, and I used this to calculate the approximate centre of each ward, and then use Python string functions to extract the latitude and longitude as separate values.

	ward	gss_code_ward	gss_code_borough	borough	geometry	Center_point	long	lat
0	Chessington South	E05000405	E09000021	Kingston upon Thames	POLYGON ((-0.33068 51.32901, -0.33059 51.32909...	POINT (-0.31203 51.34797)	-0.312027	51.347966
1	Tolworth and Hook Rise	E05000414	E09000021	Kingston upon Thames	POLYGON ((-0.30846 51.37586, -0.30834 51.37606...	POINT (-0.28990 51.37427)	-0.289905	51.374269
2	Berrylands	E05000401	E09000021	Kingston upon Thames	POLYGON ((-0.30385 51.39249, -0.30375 51.39252...	POINT (-0.28946 51.39265)	-0.289457	51.392646
3	Alexandra	E05000400	E09000021	Kingston upon Thames	POLYGON ((-0.26990 51.38845, -0.26975 51.38838...	POINT (-0.27527 51.38389)	-0.275273	51.383887
4	Beverley	E05000402	E09000021	Kingston upon Thames	POLYGON ((-0.24662 51.39921, -0.24672 51.39921...	POINT (-0.25894 51.40274)	-0.258938	51.402740

Data Source 3: Foursquare Location Data

Foursquare is best known for it's 'City Guide' social media application that enables users to discover places of interest near their location. It also provides an API, which allows developers access to a rich source of location data.

I used the 'venues' endpoint to query each ward's centroid point for the first 100 venues within a 500 metre radius.

An example of the data obtained for each Ward is shown below:

	Ward_name	Ward Latitude	Ward Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Abbey	51.539711	0.077935	Nando's	51.539780	0.082297	Portuguese Restaurant
1	Abbey	51.539711	0.077935	Cristina's	51.536523	0.076672	Steakhouse
2	Abbey	51.539711	0.077935	The Gym London Barking	51.536193	0.078601	Gym
3	Abbey	51.539711	0.077935	Subway	51.538000	0.081319	Sandwich Place
4	Abbey	51.539711	0.077935	Costa Coffee	51.539272	0.081341	Coffee Shop

Machine learning algorithms cannot process categorical data such as that returned by Foursquare's API. Therefore we use one-hot encoding to pivot the data and transform the categories into columns, with a '1' if that category is present in a ward, and '0' if it is not present.

The venues are then grouped by ward, with each venue category which (here we are using the 'Code' column as a key to represent the ward).

	Code	ATM	Accessories Store	Adult Boutique	Advertising Agency	Afghan Restaurant	African Restaurant	Airport	Airport Lounge	Airport Service	...	Windmill	Wine Bar	Wine Shop
0	E05000026	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
1	E05000027	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
2	E05000028	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
3	E05000029	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
4	E05000030	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0

Methodology

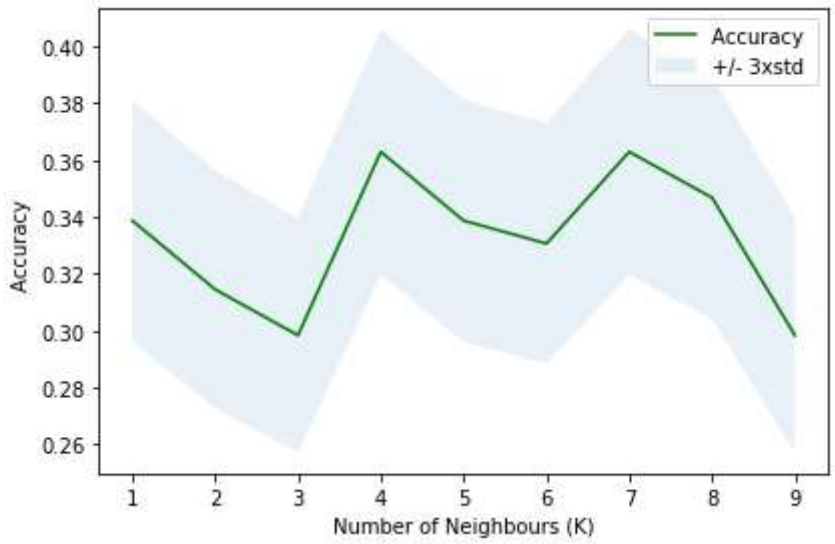
To use machine learning methods on our data, the first stage was to split our data into train and test sets. To do this I used SciKit Learn's train_test_split function, with a test size of 20% of our data.

I then used the train set to explore three different methods of machine learning, to see which might be suitable for my experiment.

- K-means clustering
- Decision Tree
- Support Vector Machine.

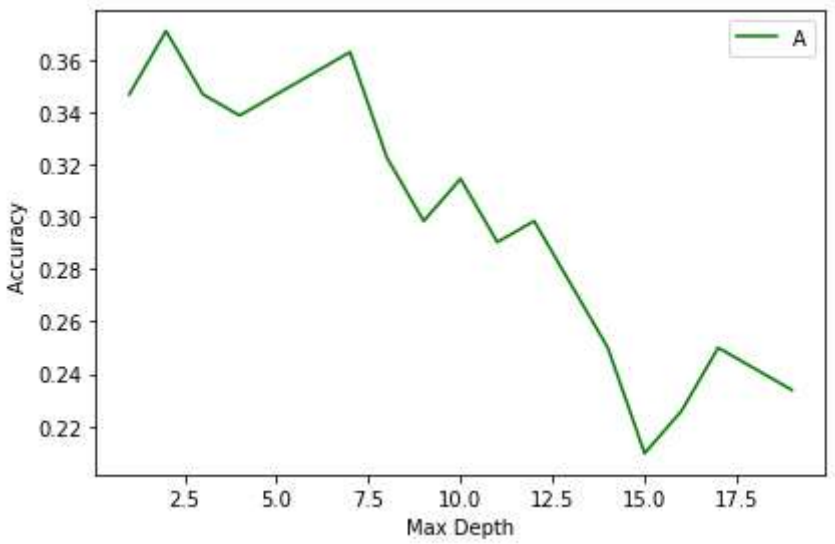
K-means Clustering

I tested different values of K, and as the following chart shows, a K of 4 was narrowly the best, although no value of K gave an accuracy above around 0.36. This accuracy is low, which may indicate that clustering will not give us optimum results.



Decision Tree

I suspected that there were too many discrete categorical values to make decision tree a credible methodology. The chart below shows different maximum tree depths, and as you can see, it peaks at around 0.36, but at a low max depth which given the number of categories seems unlikely to give good results. Higher max depths actually *decreased* accuracy, so we will not use Decision Tree as a classifier.



Support Vector Machine

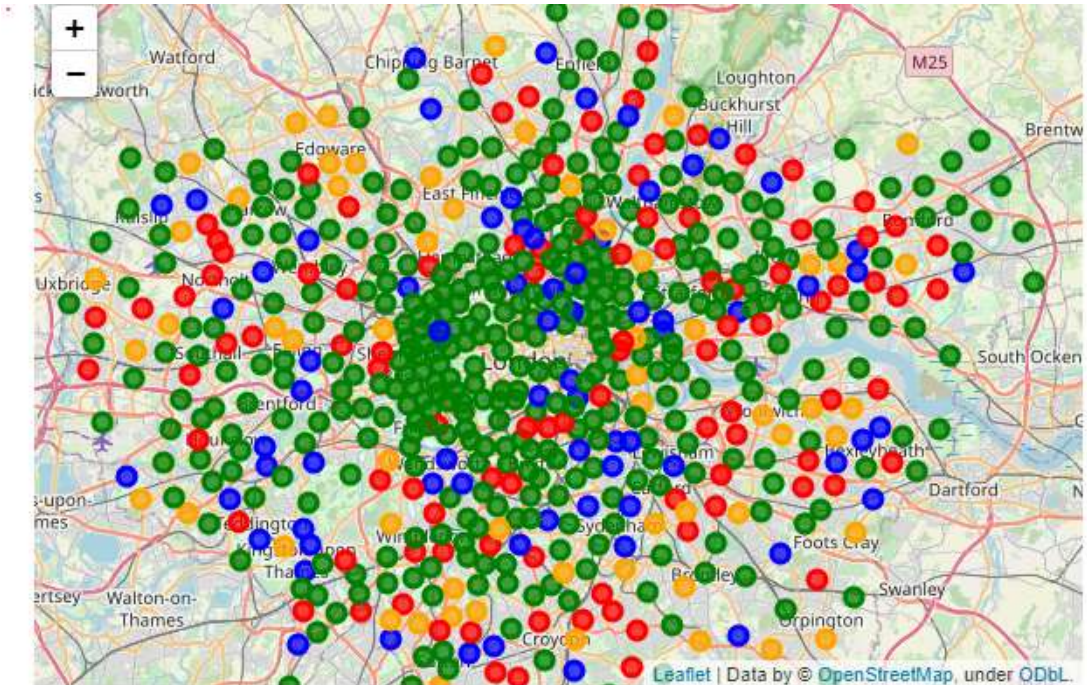
For completeness, we also trialled the use of Support vector Machine (SVM) in Scikit Learn. This also did not produce good results for this set of data, with a Jaccard index of 0.15, and F1 score of 0.04.

Results

Using K-means clustering

Despite the low accuracy predicted in our train/test split, we decided to proceed with using K-means clustering with a K of 4. This produced four clusters which you can see represented on the map below. In the following section, we will examine each of these clusters to identify any trends.

For each cluster, we will histogram the property values to ascertain whether the cluster skews towards low or high value properties, and we will use word clouds to visualise the most popular venues within each cluster.

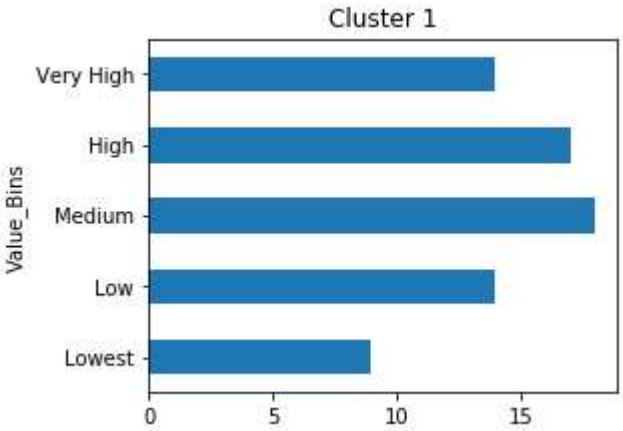


Key:

Cluster 1
Cluster 2
Cluster 3
Cluster 4

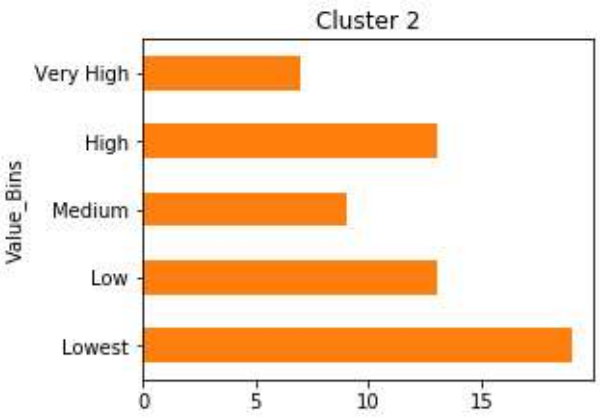
Cluster 1

Cluster 1 is characterised by Pubs and Restaurants, and Exhibition spaces. You can see from the map that Cluster 1 (the blue dots) are spread, and not concentrated in any more area. The property value histogram, however, is fairly close to a normal distribution, with only a slight skew towards higher value properties. Therefore, membership of this category is not a good indicator of property value in the area.



Cluster 2

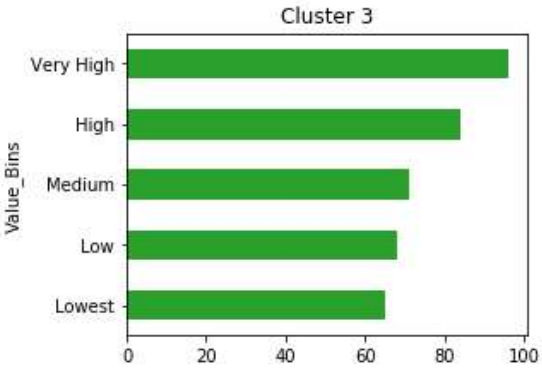
Cluster 2, histogram shows a clear trend towards lower value properties. On the map Cluster 2 (orange dots) is absent entirely from the City centre, but is present in the 'outer ring' of Greater London. Notably, 'Factory' is prominent on the word cloud, suggesting that Cluster 2 might be indicative of Industrial areas, which might go some way to explaining the presence of less desirable, hence lower value properties.



Cluster 3

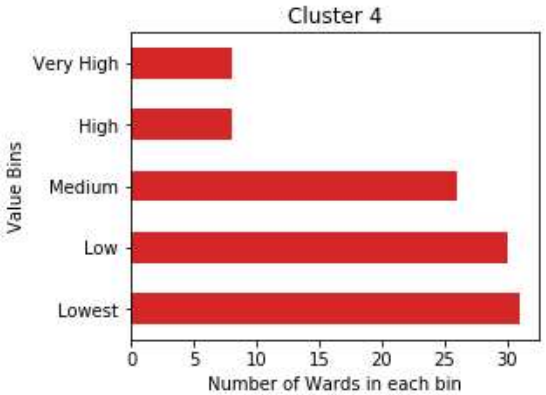
Cluster 3 on the map (green dots) is most densely clustered in the city centre, but has presence throughout the suburbs. It is characterised by restaurants, coffee shops and sandwich bars – possibly indicating urban centres, and in the suburbs likely reflects suburban 'town centres' – a characteristic of London's geography where towns have grown together into a conurbation.

The histogram has a slight trend towards higher value properties but overall there is not a strong correlation between property value and Cluster 3.



Cluster 4 11

Cluster 4 (red dots) is geographically spread throughout the suburbs, but is absent from central London. It is characterised by grocery and convenience stores, pizza and take-away restaurants, and similar to Cluster 2, also features factories. This cluster has a fairly strong weighting towards lower value properties.



Discussion

Despite the train-test set showing disappointing accuracy scores for K means clustering, the clusters did show some trends for house value prediction. Cluster 2 and Cluster 4 both showed a fairly distinct weighting towards low value properties – likely due to the proximity of industrial areas given that factories were a common feature in both clusters.

This could present an interesting opportunity for future analysis – could we use Foursquare's location data to look for factories or other industrial locations – ignoring all other venue types - to see if we can identify how close properties can be to industrial areas before property value is negatively affected.

Whilst out of scope for this project, it would be interesting to score different venue types on how much of a positive or negative impact they have on house prices in a given area.

Some venue types appeared to have little bearing on property value – for instance 'restaurant'. It is possible that this category is too broad – there is obviously a real-world nuance between a Michelin-starred venue in a prestigious area, and a family-run restaurant in a suburb. There may be other end points in Foursquare's API that we could use to approximate this nuance, such as venue rating or price tier (£ to ££££) that could be used to improve the result set.

Finally, for this experiment, I used the first 100 venues for each ward, with a 500m radius around the ward centroid. It would be interesting to see whether changing the parameters, e.g. increasing the number of venues, changes the character of each ward to a significant degree, and therefore how it changes the accuracy of the clustering.

Conclusion

Without further refining of the methodology, such as including premium end-points from Foursquare's API such as price tier – a measure of how expensive a venue is – merely selecting on venue types was overall a poor indicator of house price value.

Using the parameters given, it was *possible* to predict the presence of low value properties given venue data alone. A common feature of the clusters skewing as low value was the presence of factories, indicating a proximity to industrial areas.