Capstone Project for the Applied Data Science course by IBM

Battle of the London Boroughs

By J Poulten

The Business Problem

- An estate agency has approached me. They wish to provide their customers with a service that analysis the current home address and identifies locations in the surrounding area that are similar.

The Tools

- Geocoder Library and ArcGIS API

Retrieve geographical data for specific postcodes and regions.

- Forsquare API

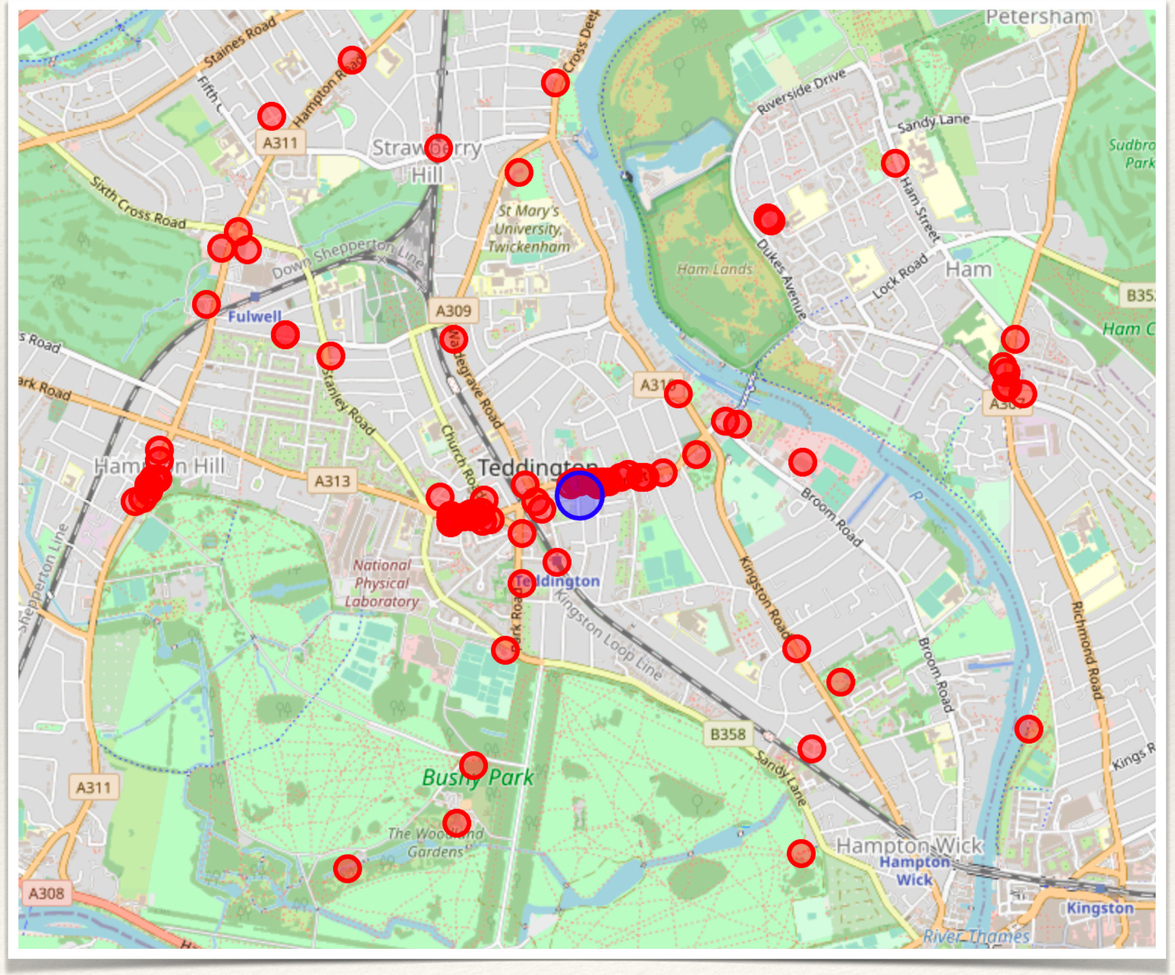
Return areas of interest and known venues with in a set radius of a specific point.

- Postcodes.io API

Return information relative to a given postcode.

Data Acquisition and cleaning

- Select a borough of London you wish to inspect
- Retrieve the Lat Lon of the borough using the ArcGIS API
- Use the Forsquare AIP to return interesting venues in the surrounding area
- Use the Postcode API to retrieve spatial information and plot locations on a map



Group data ready for cluster analysis

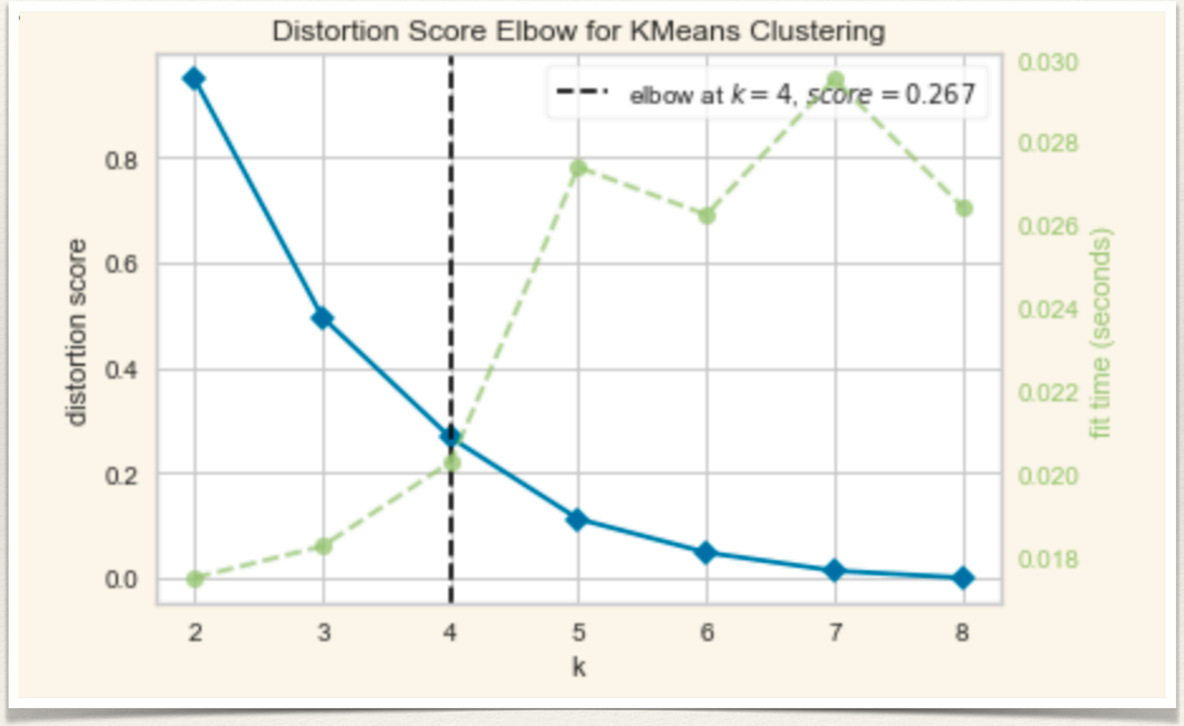
Some of the categories returned by the Forsquare API are too specific. It is therefore useful to organise these into more generalised groupings before running the KNN analysis.

```
ted_locations_df = teddington_places_df.copy()

# Convert postcodes to all upper case
ted_locations_df['Postcode'] = ted_locations_df['Postcode'].str.upper()

# beginning with some very simple lambda functions we can reduce our initial 151 catagories down to 101
ted_locations_df.Category = ted_locations_df.Category.apply(lambda x: 'Restaurant' if 'Restaurant' in x else x)
ted_locations_df.Category = ted_locations_df.Category.apply(lambda x: 'Pub' if 'Pub' in x else x)
ted_locations_df.Category = ted_locations_df.Category.apply(lambda x: 'Café' if 'Café' in x else x)
ted_locations_df.Category = ted_locations_df.Category.apply(lambda x: 'Café' if 'Cafe' in x else x)
ted_locations_df.Category = ted_locations_df.Category.apply(lambda x: 'Café' if 'Coffee' in x else x)
ted_locations_df.Category = ted_locations_df.Category.apply(lambda x: 'Bar' if 'Bar' in x else x)
ted_locations_df.Category = ted_locations_df.Category.apply(lambda x: 'Public Transport' if 'Station' in x else x)
ted_locations_df.Category = ted_locations_df.Category.apply(lambda x: 'Public Transport' if 'Bus' in x else x)
ted_locations_df.Category = ted_locations_df.Category.apply(lambda x: 'Gym' if 'Gym' in x else x)

 # Inspecting the remaining catagory list we can see other locations we can class as "restrauents"
other_restrauants = ['Pizza Place','Gastropub']
for R in other_restrauants:
  ted_locations_df.Category = ted_locations_df.Category.apply(lambda x: 'Restaurant' if R in x else x)

# Other catagories we can class as "shops"
other_shops = [ 'Bookstore','Garden Center','Pharmacy','Optical Shop',
  'Grocery Store','Newsagent','Supermarket','Bakery','Convenience Store','Wine Shop']
for S in other_shops:
  ted_locations_df.Category = ted_locations_df.Category.apply(lambda x: 'Shops' if S in x else x)

# and others we can count as sports facilities
other_sports = [ 'Park','Gym','Pool','Athletics & Sports','Golf Course']
for Sp in other_sports:
  ted_locations_df.Category = ted_locations_df.Category.apply(lambda x: 'SportsFacilities' if Sp in x else x)

print(f"we have now reduced our initial {len(teddington_places_df.Category.value_counts())} feature set down to {len(ted_locations_df.Category.unique())}")
```

✓ 0.7s Python

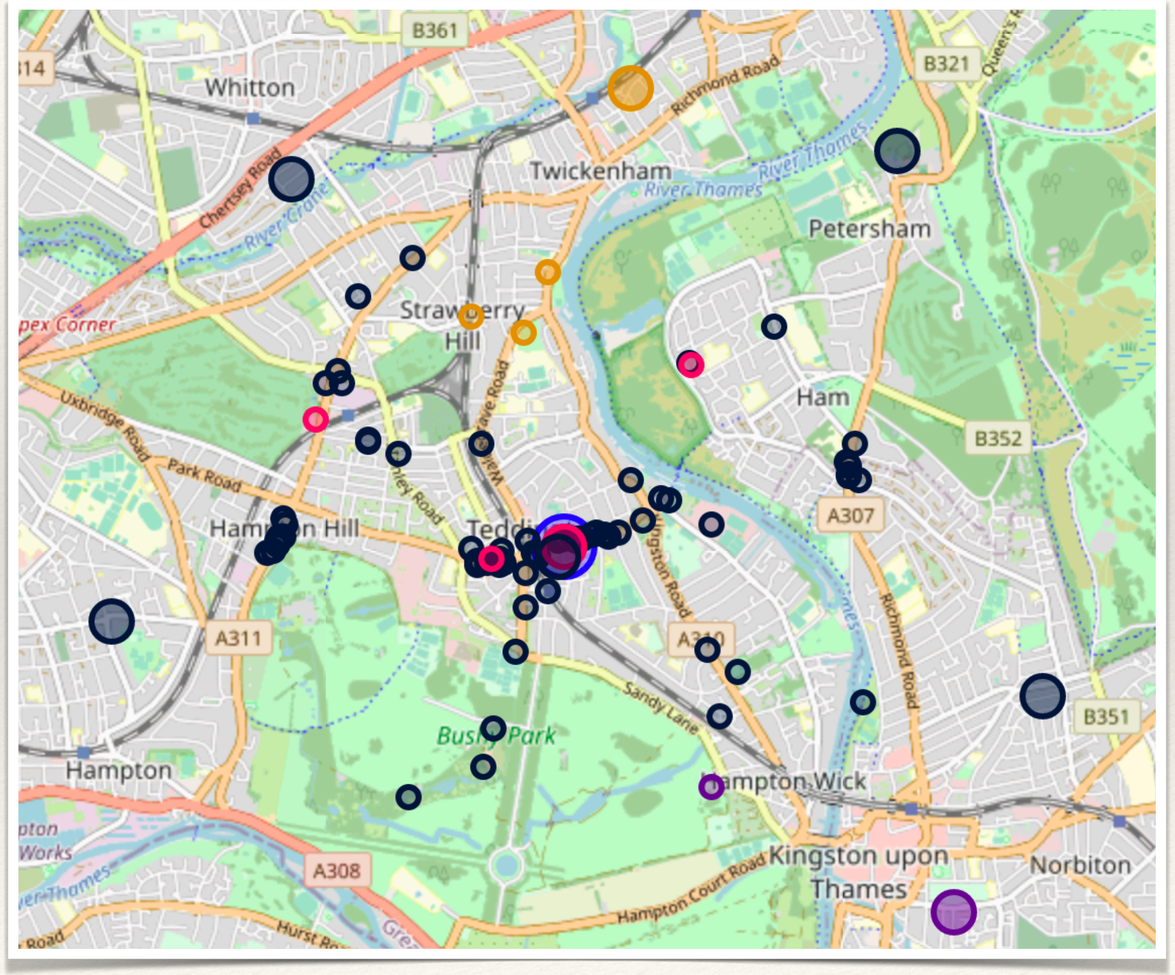
Perform KNN cluster analysis

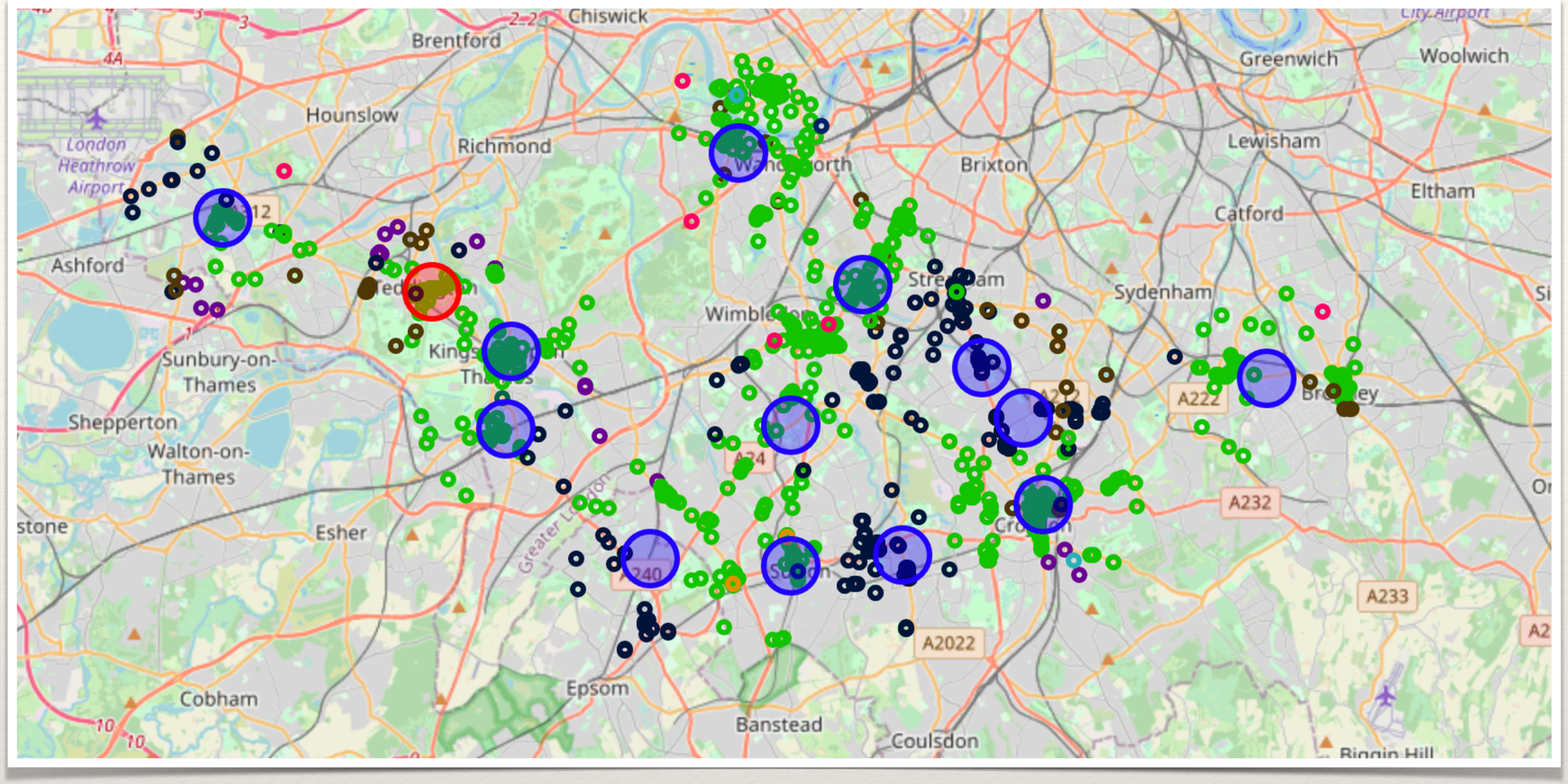
Once the data has been grouped appropriately and one-hot-encoding has been applied.We can perform our KNN cluster analysis and identify the appropriate number of clusters.



Visualise the results

- Once the data has been clustered, it is important to visualise the results.
- This ensures that there has been no obvious errors in our analysis.
- Here we can clearly see regions and ares of Teddington that are similar





South London Boroughs

Extend the analysis

Apply the same analysis to a wider collection of data

Results and conclusions

Using KNN analysis we are able to identify regions of south London, as grouped by postcode outcodes, that share similar characteristics.

We see that Teddington (red solid circle) is most similar to regions denoted by the black circles.