Applied Data Science Capstone by IBM/Coursera Capstone Project

Simplify selecting a technology company to work at in Ireland

Roger Clarke

July 2021

Dublin City: *Image from Bing.com – licence public domain*

1. Identifying the Business Problem

1.1 Background

Ireland is a European hub to over 1000 foreign direct companies in technology, pharma, social media, and many others. FDI (Foreign Direct Investment) comprises around **20%** of all private sector employment with Dublin, Cork, Kildare and Limerick playing host to the top players such as Amazon, Apple, Facebook, Google, Microsoft and Salesforce being some of the largest employer. FDI companies employ. Many of the top employers are located either in city centres or in business parks on the outskirts.

Young top talent coming out of universities in many countries interested securing employment with these companies know the pay and conditions are good and that Ireland, in general, is a wonderful place to live. Transport

know that Ireland is a multicultural society, and that the Irish are infamous for being friendly and engaging. They want to ensure that they select their potential employer based on the location offering a wide variety of social venues within walking distance of the office (2km) which is about a 20-minute walk.

Dublin City has the largest part of the population and hosts the majority of the selected companies hence would be expected to have the most variety in venues. However, Limerick and Cork are also thriving cities and would be expected to have a decent variety, but the key differentiator here will the location of each company.

Some companies tend to find in business parks outside of the city centre which tends to limit venue availability within a 2km range. The distance of 2km is about twenty-minute walk.

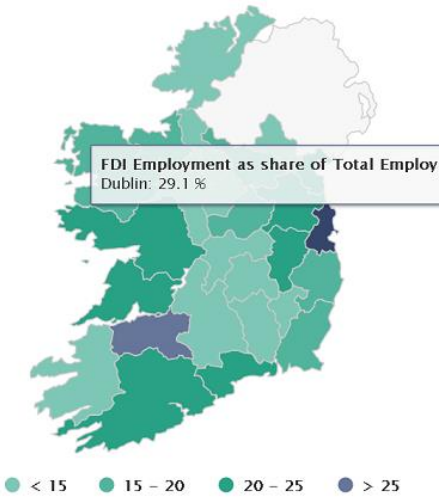


Image copyright of CSO Ireland

The objective of this this project is to assist would-be employees discover if living within a 20-minute walk of any company can provide them with all of amenities they would require for a reasonable quality of life from socialising to shopping.

2. Data Acquisition

2.1 Data Acquisition

2.1 Foreign Direct Employers

There are many foreign direct employers in Dublin, so many that there would be an excess of data points to deal with. Instead, a sample set will be used based on of those with the highest employment figures and some who are found outside of the city. I wanted to ensure the that the three main cities where companies location was represented. Some of the companies have multiple buildings in various parts of the city.

2.2 Foursquare

The Foursquare API was used supply all venues within 2km of each company selected for evaluation along with their category, latitude, and longitude. The choice of 2km is intentional as its distance you can quickly walk in 20-minutes.

3.1. Methodology

3.1.1 Company Data

The company's dataset required a company name, region (county name). street address, latitude, and longitude. Since the companies are in various parts of the country and postal districts only apply to part of Dublin, region was added to help slice the results by county name. This is especially useful in analysing the clustering results. The dataset was manually acquired as it easier than trying to automate and manage variances such as company building names. The exact business name and addresses were obtained by

searching Google Maps and copying the cited postal address. The Google Search was a manual process because, some cases, the search returns comparable results not related to the exact company in question. In some cases, the addresses were complete and needed the addition of a postal code improve the Google Map API search accuracy. Each company was added as a dictionary with the name, address, and postal code to a list. The list was then iterated and, using Google maps the latitude and longitude was derived using the business name, postal address and, where available, the postal code. The results were then appended to a dataframe.

	COMPANYNAME	COMPANYADDRESS	POSTALCODE	REGION	COMPANYLAT	COMPANYLNG
0	FaceBook	BLOCK J, FACEBOOK DUBLIN BALLSBRIDGE CAMPUS, S...	DUBLIN 4	DUBLIN SOUTH	53.329515	-6.225678
1	Google Grand Canal Quay	ONE, GRAND CANAL PLAZA, GRAND CANAL STREET UPPER	DUBLIN 4	DUBLIN CITY	53.340627	-6.239397
2	Google Building GRCQ1	1 Grand Canal Quay, Dublin, Ireland	DUBLIN 2	DUBLIN CITY	53.340740	-6.239371
3	Google	Google Building Gordon House, Barrow St, Dubli...	DUBLIN 4	DUBLIN CITY	53.340047	-6.235744
4	Dublin Google Data Center	Grange Castle Business Park South, Baldonnel R...	DUBLIN 22	DUBLIN WEST	53.313734	-6.449233
5	Google EMEA HQ	4 Barrow St Ringsend, Dublin 4, Ireland	DUBLIN 4	DUBLIN CITY	53.339758	-6.236944
6	Google Blackthorn	Blackthorn Road, Sandyford, Dublin, Ireland	DUBLIN 18	DUBLIN SOUTH	53.275835	-6.216937
7	Amazon	Burlington Plaza, 1 Burlington Rd	DUBLIN 4	DUBLIN CITY	53.332500	-6.246058
8	Twitter International	CUMBERLAND PLACE, FENIAN STREET	DUBLIN 2	DUBLIN CITY	53.341705	-6.249626
9	LinkedIn	GARDNER HOUSE, WILTON PLAZA, WILTON PLACE	DUBLIN 2	DUBLIN CITY	53.334907	-6.249499

3.1.2 Cleansing the data

The companies data required little cleansing however to be certain that Google Maps didn't fail to get the latitude and longitude of each address, a count of isna() values was done and no issues were found.

3.2.1 Venues

The first step in using the Foursquare API was to create a free account and get a client id and secret. The second step was creating a Venues dataframe which would contain all of the venues within a 2km radius of each company's latitude and longitude. Using the API is quite simple and requires a client id, client secret, API version, latitude, longitude, and a radius to search against. The companies dataframe was iterated with each company's latitude and longitude passed to the API. The API returned a JSON of venues and the specific information needed was the name, category, latitude and longitude. These properties, the company name, latitude, and longitude where then added to the Venues dataframe.

	Unnamed: 0	COMPANYNAME	REGION	Latitude	Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	0	FaceBook	DUBLIN	53.329515	-6.225678	The Good Food Store	53.330478	-6.224752	Food & Drink Shop
1	1	FaceBook	DUBLIN	53.329515	-6.225678	The Bridge 1859	53.329319	-6.231768	Pub
2	2	FaceBook	DUBLIN	53.329515	-6.225678	Baan Thai Restaurant	53.328832	-6.230290	Thai Restaurant
3	3	FaceBook	DUBLIN	53.329515	-6.225678	Base Wood Fired Pizza	53.328869	-6.230202	Pizza Place
4	4	FaceBook	DUBLIN	53.329515	-6.225678	InterContinental Dublin	53.326608	-6.226079	Hotel

3.2.2 Cleansing the data

The results of querying Foursquare for all of these companies produced 136 unique venue categories ranging from Dining, drinking, shopping, transport (trains and bus stops) and ball games. 136 is too many to visualise so grouping them into subcategories helped reduce the clutter. I approach I used was to visually review all of the unique categories and group under 11 headings: Dining, Drinking, Leisure, Fitness, Shopping, Sport, Entertainment, Transport, Site Seeing, Services and Health and Beauty.

I steps I used add the categories to the venues a new VenueGrouping to the Venues Dataframe where:

- Get all of the unique categories df_venues['Venue Category'].unique()

```
df_venues['Venue Category'].unique()
```

```
array(['Food & Drink Shop', 'Pub', 'Thai Restaurant', 'Pizza Place',
       'Hotel', 'Soccer Stadium', 'Park', 'Wine Bar', 'Stadium', 'Spa',
       'Café', 'Nail Salon', 'Burger Joint', 'Hockey Field', 'Garden',
       'Gourmet Shop', 'Museum', 'Beach', 'Convention Center',
       'Indian Restaurant', 'Gym', 'Seafood Restaurant', 'Restaurant',
       'Gym / Fitness Center', 'Gastropub', 'Asian Restaurant',
       'Wine Shop', 'Japanese Restaurant', 'Concert Hall', 'Coffee Shop',
       'Pool', 'Plaza', 'Bakery', 'Theater', 'Bar', 'Hotel Bar',
       'Breakfast Spot', 'Outdoor Sculpture', 'Farmers Market', 'Bridge',
       'Steakhouse', 'Sushi Restaurant', 'Art Museum', 'Sculpture Garden',
       'Yoga Studio', 'Bookstore', 'Beer Bar', 'IT Services',
       'Golf Course', 'Paper / Office Supplies Store', 'Supermarket',
```

- Add a column to the venues dataframe df_venues['VenueGrouping'] = pd.Series()
- Created lists of venue groupings each populated with the relevant venue category
- Created a function to search to match the venue category against the Venue Grouping and add that to the Venues dataframe df_venues['VenueGrouping'] = df_venues['Venue Category'].apply(Clean_Grouping)
- I then removed any NaN values.

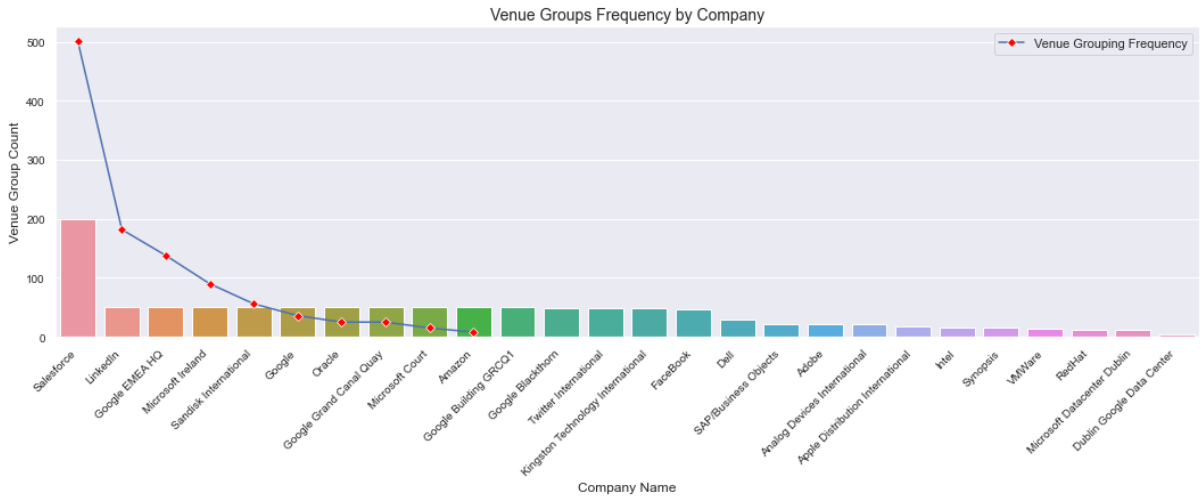
	Unnamed: 0	COMPANYNAME	REGION	Latitude	Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	VenueGrouping
0	0	FaceBook	DUBLIN	53.329515	-6.225678	The Good Food Store	53.330478	-6.224752	Food & Drink Shop	Shopping
1	1	FaceBook	DUBLIN	53.329515	-6.225678	The Bridge 1859	53.329319	-6.231768	Pub	Drinking
2	2	FaceBook	DUBLIN	53.329515	-6.225678	Baan Thai Restaurant	53.328832	-6.230290	Thai Restaurant	Dining
3	3	FaceBook	DUBLIN	53.329515	-6.225678	Base Wood Fired Pizza	53.328869	-6.230202	Pizza Place	Dining
4	4	FaceBook	DUBLIN	53.329515	-6.225678	InterContinental Dublin	53.326608	-6.226079	Hotel	Dining

3.2.3 Analysing the Results

The first step was to understand how the venue grouping frequency. The below result is interested because I would have expected Dining to the followed by Drinking, not Shopping. Might be a result of societal changes in Ireland.

VenueGrouping	COMPANYNAME
Dining	500
Shopping	182
Drinking	138
leisure	90
fitness	56
entertaintment	36
SiteSeeing	25
transport	25
Sport	15
Services	8

A more interesting view is the frequency of venue groups by company name. The below Seaborn barplot was created by creating a panda's count series for VenueGrouping and CompanyName, converting those dataframes and doing a bar and line plot on the same chart.



What does Salesforce have 500 venue groupings compared to LinkedIn or any of the other city centre-based companies?

A side-by-side comparison of the two companies reveals some interesting frequency insights but still does not explain the disparity.

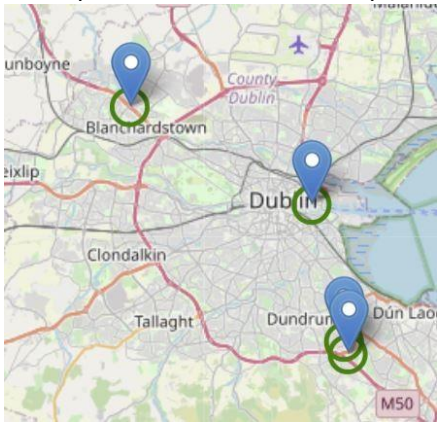
Salesforce

VenueGrouping	COMPANYNAME
Dining	99
Shopping	42
Drinking	24
leisure	14
fitness	11
entertaintment	5
SiteSeeing	3
Services	2

LinkedIn

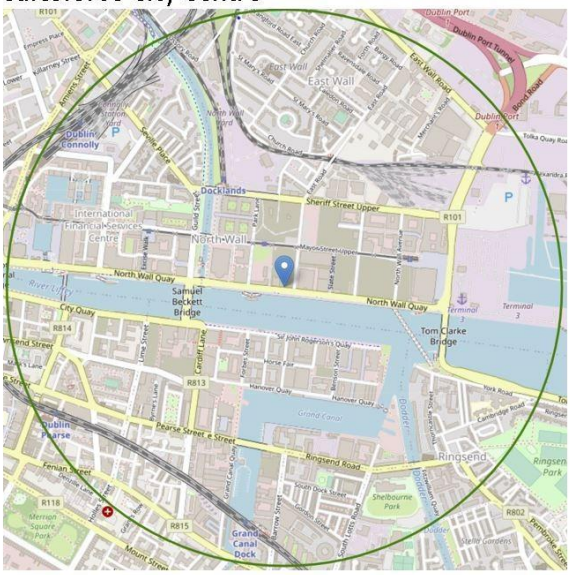
VenueGrouping	COMPANYNAME
Dining	28
Drinking	6
leisure	6
Shopping	4
SiteSeeing	2
entertaintment	2
fitness	2

I decided that perhaps the location of Salesforces offices proximity to urban facilities might be driving the numbers. I used Folium maps to create Markers only for Salesforce and we an see four buildings in the same country with three outside the city centre.



Drilling down further, we see that Blanchardstown, the pin at the top and in the right image, is located beside a huge shopping centre where was the pin in the centre of the top image, on the left below, is located in docklands.

Salesforce City Centre



Salesforce Outside the City Centre

4. Modelling

I used the supervised learning algorithm k-means to cluster the venues arounds the companies. The first step was to create a one-hot encoding which calculates how many times each venue category occurs around each company location.

```
# one hot encoding
company_onehot = pd.get_dummies(df_venues[['VenueGrouping']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
company_onehot['COMPANYNAME'] = df_venues['COMPANYNAME']

# move neighborhood column to the first column
fixed_columns = [company_onehot.columns[-1]] + list(company_onehot.columns[:-1])
companyvenues_onehot = company_onehot[fixed_columns]

companyvenues_onehot.head()
```

The resulting dataframe looks like this

	COMPANYNAME	Dining	Drinking	Services	Shopping	SiteSeeing	Sport	entertaintment	fitness	leisure	transport
0	FaceBook	0	0	0	1	0	0	0	0	0	0
1	FaceBook	0	1	0	0	0	0	0	0	0	0
2	FaceBook	1	0	0	0	0	0	0	0	0	0
3	FaceBook	1	0	0	0	0	0	0	0	0	0
4	FaceBook	1	0	0	0	0	0	0	0	0	0

Before running clustering, I grouped the onehot results with the venue dataframe.

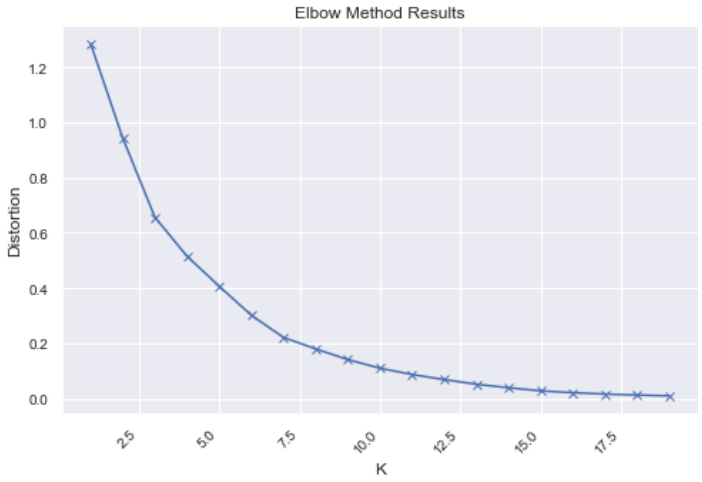
	COMPANYNAME	Dining	Drinking	Services	Shopping	SiteSeeing	Sport	entertaintment	fitness	leisure	transport
0	Adobe	0.500000	0.045455	0.00	0.181818	0.000000	0.045455	0.000000	0.000000	0.000000	0.227273
1	Amazon	0.500000	0.180000	0.00	0.080000	0.040000	0.000000	0.040000	0.040000	0.120000	0.000000
2	Analog Devices International	0.450000	0.150000	0.05	0.300000	0.000000	0.000000	0.050000	0.000000	0.000000	0.000000
3	Apple Distribution International	0.333333	0.000000	0.00	0.166667	0.055556	0.111111	0.055556	0.111111	0.111111	0.055556
4	Dell	0.310345	0.103448	0.00	0.275862	0.000000	0.034483	0.000000	0.068966	0.068966	0.137931

The second step was to categorise the venue frequencies in terms of ordinals. Taking the top 10 venues, I created a sorted merged dataframe of company names and the venue position by ordinal with the following result:

	COMPANYNAME	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
0	Adobe	Dining	transport	Shopping	Drinking	Sport	Services	SiteSeeing	entertaintment	fitness	leisure
1	Amazon	Dining	Drinking	leisure	Shopping	SiteSeeing	entertaintment	fitness	Services	Sport	transport
2	Analog Devices International	Dining	Shopping	Drinking	Services	entertaintment	SiteSeeing	Sport	fitness	leisure	transport
3	Apple Distribution International	Dining	Shopping	Sport	fitness	leisure	SiteSeeing	entertaintment	transport	Drinking	Services
4	Dell	Dining	Shopping	transport	Drinking	fitness	leisure	Sport	Services	SiteSeeing	entertaintment

The last step was to run k-means clustering. I needed to figure out how many clusters were needed or the K value. I used the Elbow method to decide the optimum K value by looking for sharp turns in the curve. I experimented a range of values from 3 to 6 with 5 being the maximum. The experimentation required running the cluster algorithm with different k values and then evaluating the shape of the cluster number. Using 6 results in 0 hence 5 was the optimum figure.

```
shape = dublin_merged.loc[dublin_merged['Cluster Labels'] == cls, dublin_merged.columns[[0] + list(range(1, dublin_merged.shape[1]))]].shape[0]
```



The resulting dataframe looks like this.

	COMPANYNAME	REGION	Latitude	Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	VENUEDISTANCE	VenueGrouping	Cluster Labels	1st Most Common Venue	2nd Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th M Com Ve
627	Adobe	DUBLIN	53.290558	-6.419989	The Streat Cafe	53.282587	-6.423354	Café	0.91	Dining	0	Dining	transport	Shopping	Drinking	Sport	Services	SiteSe
628	Adobe	DUBLIN	53.290558	-6.419989	Eddie Rocket's	53.282400	-6.422955	Diner	0.93	Dining	0	Dining	transport	Shopping	Drinking	Sport	Services	SiteSe
629	Adobe	DUBLIN	53.290558	-6.419989	Lidl	53.285452	-6.418304	Supermarket	0.58	Shopping	0	Dining	transport	Shopping	Drinking	Sport	Services	SiteSe
630	Adobe	DUBLIN	53.290558	-6.419989	Anvil Restaurant	53.280977	-6.443511	Restaurant	1.89	Dining	0	Dining	transport	Shopping	Drinking	Sport	Services	SiteSe
631	Adobe	DUBLIN	53.290558	-6.419989	Dunnes Stores	53.283249	-6.422496	Supermarket	0.83	Shopping	0	Dining	transport	Shopping	Drinking	Sport	Services	SiteSe

To make this more intelligible, grouped the company name by the ordinal position. The results are interesting but difficult to translate into how they group around a company if that company's location offers sufficient variety.

2nd Most Common Venue	COMPANYNAME	REGION	Latitude	Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	VENUEDISTANCE	VenueGrouping	Cluster Labels	1st Most Common Venue	3rd Most Common Venue	4th Most Common Venue	5th Most Common Venue	6th Most Common Venue	7th Most Common Venue	8th Most Common Venue	9th Most Common Venue	10th Most Common Venue
Dining	Kingston Technology International	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48	48
	RedHat	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
Drinking	Amazon	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
	Dublin Google Data Center	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	FaceBook	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47	47
	Google	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
	Google Building GRCQ1	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
	Google EMEA HQ	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
	Google Grand Canal Quay	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
	LinkedIn	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
	Oracle	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
	Twitter International	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49
Services	Microsoft Datacenter Dublin	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
Shopping	Analog Devices International	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
	Apple Distribution International	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
	Dell	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29	29
	Google Blackthorn	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49
	Intel	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
	Microsoft Court	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
	Microsoft Ireland	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
	SAP/Business Objects	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22
	Salesforce	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200	200
	Sandisk International	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
	Synopsis	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
	VMWare	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
transport	Adobe	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22

4.1 Examining the Clusters

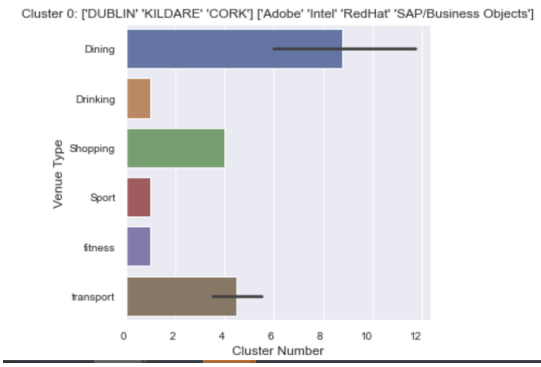
Rather than work off tabulated data, I decided to use an approach that visualised the venue cluster count around each of the company's and the counties related to each of the 6 clusters. The below approach was repeated for each cluster.

```
c0 = dublin_merged.loc[dublin_merged['Cluster Labels'] == 0, dublin_merged.columns[[0] + list(range(1, dublin_merged.shape[1]))]]
c0.head(10)
```

	COMPANYNAME	REGION	Latitude	Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	VENUEDISTANCE	VenueGrouping	Cluster Labels	1st Most Common Venue	2nd Most Common Venue
627	Adobe	DUBLIN	53.290558	-6.419989	The Streat Cafe	53.282587	-6.423354	Café	0.91	Dining	0	Dining	transport
628	Adobe	DUBLIN	53.290558	-6.419989	Eddie Rocket's	53.282400	-6.422955	Diner	0.93	Dining	0	Dining	transport
629	Adobe	DUBLIN	53.290558	-6.419989	Lidl	53.285452	-6.418304	Supermarket	0.58	Shopping	0	Dining	transport
630	Adobe	DUBLIN	53.290558	-6.419989	Anvil Restaurant	53.280977	-6.443511	Restaurant	1.89	Dining	0	Dining	transport
631	Adobe	DUBLIN	53.290558	-6.419989	Dunnes Stores	53.283249	-6.422496	Supermarket	0.83	Shopping	0	Dining	transport
632	Adobe	DUBLIN	53.290558	-6.419989	McDonald's	53.283518	-6.422547	Fast Food Restaurant	0.80	Dining	0	Dining	transport
633	Adobe	DUBLIN	53.290558	-6.419989	McGettigans Cookhouse & Bar	53.301135	-6.418931	Gastropub	1.18	Dining	0	Dining	transport
634	Adobe	DUBLIN	53.290558	-6.419989	Costa Coffee	53.283152	-6.422802	Coffee Shop	0.84	Dining	0	Dining	transport
635	Adobe	DUBLIN	53.290558	-6.419989	Café Togo	53.290555	-6.420824	Café	0.06	Dining	0	Dining	transport
636	Adobe	DUBLIN	53.290558	-6.419989	Citywest Hotel	53.285933	-6.446203	Hotel	1.82	Dining	0	Dining	transport

```
tmp = c0.groupby(['VenueGrouping', 'COMPANYNAME', 'REGION']).count()
tmp = tmp.reset_index()

barplot = sn.barplot(x="Cluster Labels",y="VenueGrouping",data=tmp).set_title("Cluster 0: %s %s" % (c0.REGION.unique(), c0.COMPANYNAME.un
fig = plt.gcf()
plt.xticks(horizontalalignment='right', fontweight='light', fontsize='small')
plt.yticks(fontweight='light', fontsize='small',)
fig.set_size_inches(5, 5)
plt.xlabel('Cluster Number')
plt.ylabel("Venue Type")
plt.show()
```

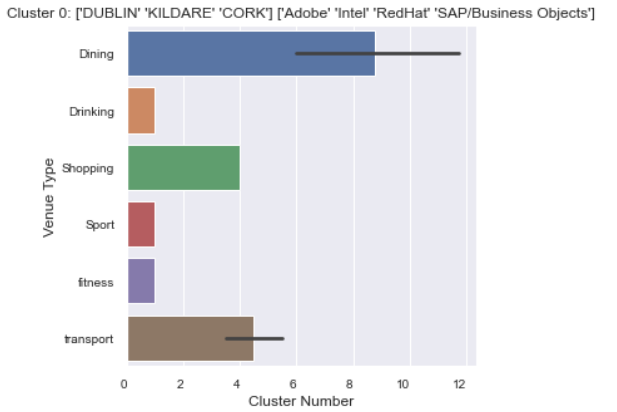


Dining is constantly in the highest frequency bucket but there are some interesting deviations in categories. Clusters 1 and 2 stand out for not having any form of transport which does not make sense given the company locations.

Microsoft in Cluster 1 has tram line less than 500m from both of its south county buildings which are splitting d istance from each other, and Salesforce is right beside the tram station.

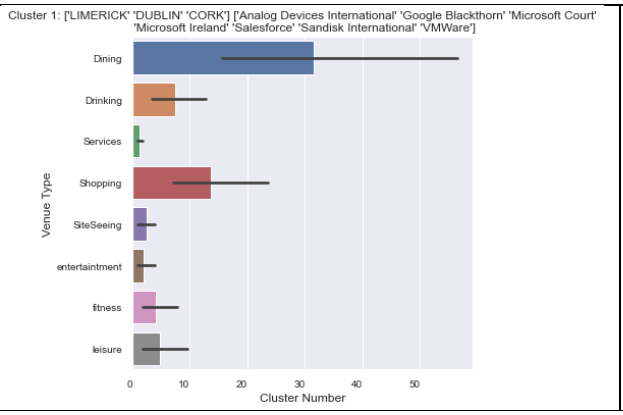
Cluster 2 has companies in the heart of the city with a tram line nearby and a lot of bus services. Foursquare's t ransport data for these locations may be incorrect which could easily mislead the reader assuming that individ ual is dependent on public transport.

Cluster 0



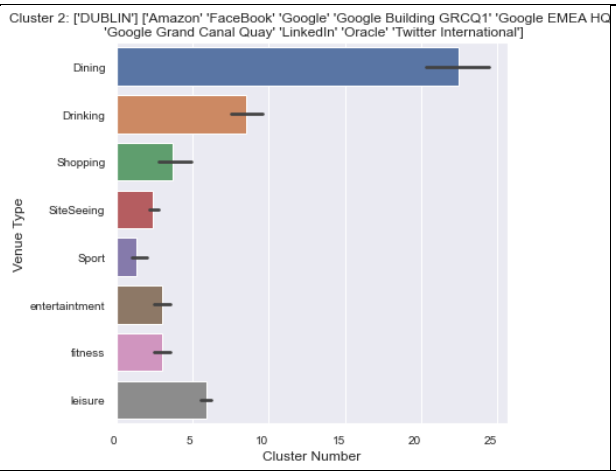
Dining is the dominant venue category for three c ompanies in Kildare and Cork.

Cluster 1



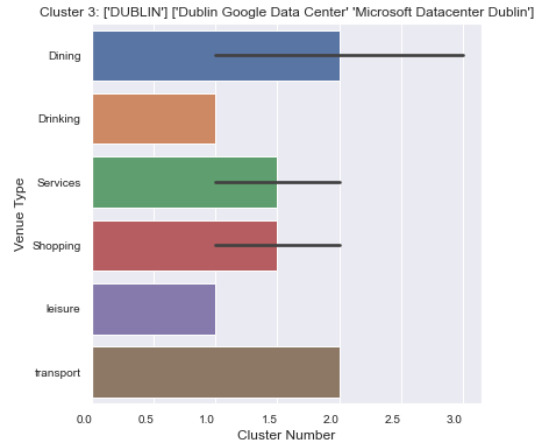
Again, Dining is the dominant venue category for 5 different companies in Dublin and Cork. Cork is shared with cluster 0. Microsoft appears twice du e to the buildings in the same location. No transp ort options.

Cluster 2



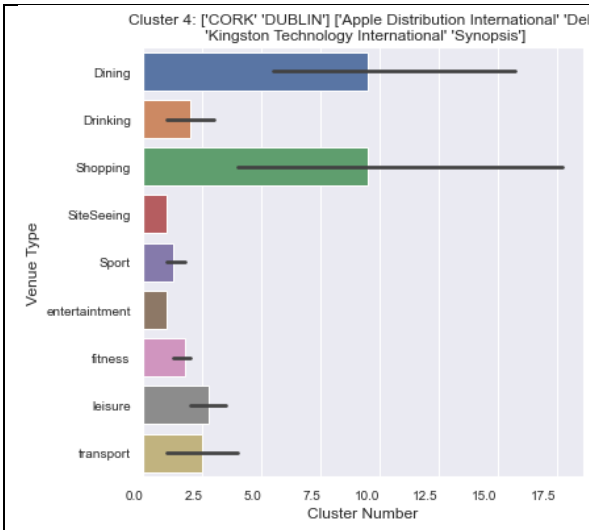
Cluster 2 is Dublin centric covering again 5 compa nies with Dining dominating. Looking carefully at t he companies you'll see that Google's four buildin gs, named differently, are all in the same location. Again, no transport options

Cluster 3



Cluster 3 is having two data centres with a more bal anced frequency between venue categories but not the full range. Data centres tend to be found well ou tside dense urban areas.

Cluster 4



Cluster 4 has the full range of venue categories acr oss two counties with shopping and dining domina ting. A similar frequency pattern to cluster 0

5.Visualisation

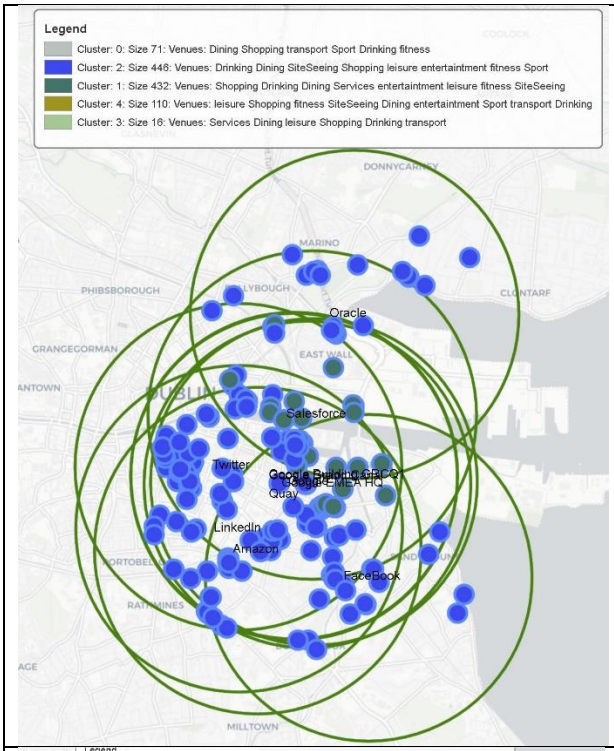
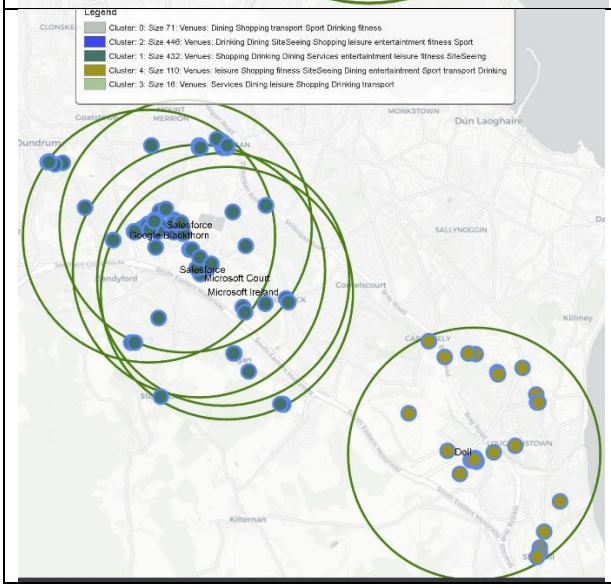
The cluster charts offer some insight into the venue category frequencies by country and company but what they don't show is how both the clusters and venue categories sit within or outside of a 2km radius of each company. I used Folium maps creating text markers for each company (save having to click on the pin for the company name) and surrounded each company with circle of 2k metres. The clusters were added as coloured markers along with legend using the same colours which shows the cluster and associated venue groupings.

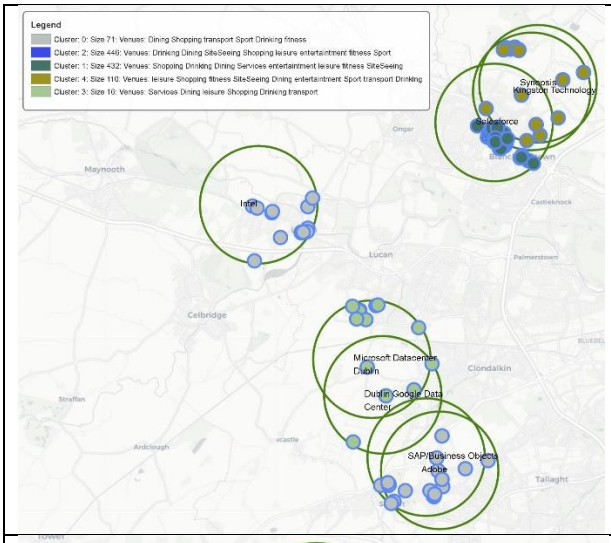
The choice of companies distributed in different counties is intention but it also makes displaying their results space consuming on the blog. So I apologies in advance for all of the scrolling you're about to do. That said, visualising the results by region is far more effective than showing datasets with large volumes of rows. I'll break out the visualisation into county and area. As you work through the visuals, a number of patterns begin to emerge:

- Where a company is located in a business park, the range of venues diminishes significantly when compared to an urban area.

- Outside of the city centres, venues don't generally congregate around the companies. Instead they spread out to the outer boundaries of the radius limit indicating a longer walking times.

Let's look at some of the companies, how the venues visually cluster around them, the frequency and category range. You can view the clustering map here.

	Dublin City Most of the companies in the city centre or, close to it are in cluster 2 with the largest number of venues. Of note is Salesforce close to East Wall which has the lower number of venues. This is simply down to location. The combination of companies and venues gives the candidate the most choice.
	Dublin South There are five companies, four located within a 20-minute walk of each other with Dell being the outlier. Dell is in a rapidly developing area with active plans to build 6k+ homes and tram line already running through the centre. In time the cluster and venue frequency will increase. A candidate's best choice here would be the companies in the upper part of the map.

	Dublin West. A large area to cover. Salesforce is in a dense urban/business area beside a huge shopping centre offering an extensive range of venues with no train lines but a lot of busses. SAP/Adobe are found close to a tram line right into the city centre. For a candidate, SAP/Adobe and Salesforce would be good choices.
	Cork Its interesting to see that VMWare, which further from the city centre than Apple, has more venues. A difficult choice for the candidate work for huge company like Apple close to the city centre or for small companies further away and with less venue options. Still, a great county to live in.
	Limerick Only one company located well outside the city centre with no tram lines, yet the venue variety is impressive but many of them are at least a 15-minute walk.

6. Discussion

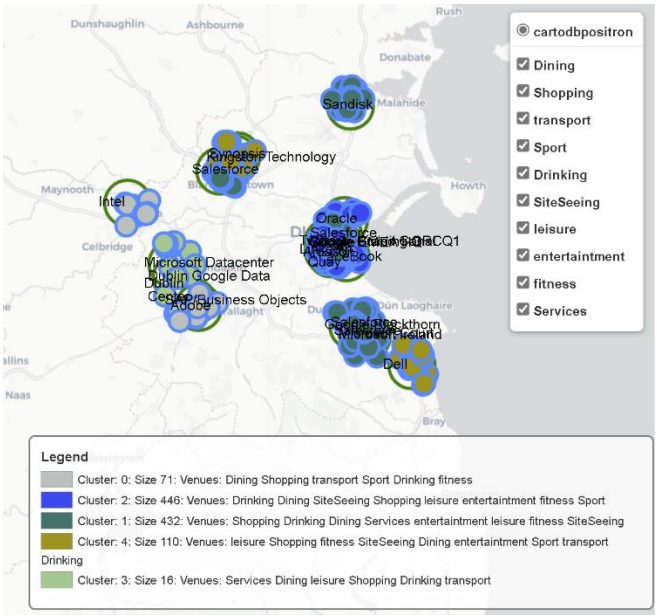
The cluster analysis reveals that we eat, shop, and then drink which is interested in a county famous for alcohol consumption. I do wonder if Foursquare dataset is more U.S centric in Ireland than they realise or is it indicative of change in Irish attitudes. I'll leave that to the reader to decide.

Microsoft & Google in Cluster 1 have a tram line less than 500m from both of their south county buildings. It does not make sense that Foursquare has no transport data. Cluster 2 has companies in the heart of the city with a tram line nearby and a lot of bus services. Foursquare's transport data for these locations may be incorrect which could easily mislead the reader assuming that individual is dependent on public transport.

Urban density and companies locating in business parks tend to decide both the range of venues and their frequency. Where a company is in a business park, the range of venues diminishes significantly when compared to an urban area. Outside of the city centres, venues don't congregate around the companies. Instead, they spread out to the outer boundaries of the radius limit showing a longer walking times.

In terms of choosing a company based on venue range and frequency, Salesforce offers the most choice based on its locations. With Microsoft and Google, the candidate needs to be careful in choosing the building location. The data centres are in an area with poor venues choices, but the head offices offer the opposite.

A bonus feature in the Folium map is a venue category menu where you can switch on an/off venue categories to discover which are nearest or furthest from the 2km radius ring on the map.



7. Conclusion

This project has achieved its objective, that being, to simplify discovering the range venues that are within walking distance of one of the selected companies. Whilst limited to 30 companies, the clustering and visualisation certainty makes discovery a lot easier. Adding more companies would be possible but using large volumes of markers on Folium maps has tendency to affect the performance of Windows.

A bonus to the project would have been to include the availability of rental accommodation, its cost and types which would create a full picture of living, working, and socialising within that 2km radius. Sadly, the property sites with said information prohibit the use of their data for this purpose. An alternative considered was the average purchase price of properties but, the data shared by the same property sites with the central statistics office is missing cost information for 70% of all properties.

The code can be found on GitHub.