Capstone Project - The Battle of the Neighborhoods

This report was written as part of final capstone project for IBM Data Science Professional Certification. It contains a summary of all the steps that are necessary for any Data Science project or research:

Table of Contents

Introduction:

Background

I have taken my first steps towards acquiring skills related to data science by doing the IBM Data Science Professional Certificate course on Coursera. The last module of this course is a capstone project. This project is about using data science toolset on a real-life problem and demonstrating the creation of value by applying the learned skills. I present here the summary of my project and the findings. The analysis was performed in Python. If you are interested in the details, then you can find a more detailed report and the Jupyter notebook at the end of the post.

Edmonton is the capital city of Canadian province of Alberta. Edmonton had a population of 932,546 in 2016, making it Alberta's second-largest city (after Calgary) and Canada's fifth-largest municipality. Edmonton is the major economic centre for northern and central Alberta and a major centre for the oil and gas industry.

Business Problem

The aim of this project is to provide evidence in support of choosing the best neighbourhood for opening of businesses in Edmonton, Alberta, Canada. With a density of 1,186 people per square kilometer (3,074/sq mi), Edmonton is on the lower end of spectrum in regards to population density. Additionally, Edmonton is a driving focussed city, which means that one has to look beyond population density to decide the best location for opening a commercial establishment.

For this purpose in addition to the neighbourhood population; this report will provide a client additional criteria of safety, real estate value and existing competition in a given neighbourhood to help in deciding the optimum location for opening a business. This report provides this information through effective, visually appealing and easy to understand data science tools such as data visualizations, maps, cluster analysis and regression. With this additional information, a client would be able to make a much more informed decision backed by facts.

Data

Following are the data sources listed under each of the criteria:

- Neighbourhood - This data is sourced from City of Edmonton’s Open Data Catalogue
- Population - This data is sourced from City of Edmonton’s Open Data Catalogue

- Safety - This data is sourced from City of Edmonton's Open Data Catalogue
- Assessed Value - This data is sourced from City of Edmonton's Open Data Catalogue
- Existing commercial establishments in a neighbourhood - This data will be fetched using Four Square API using the GET function to explore the neighbourhood venues and to apply machine learning algorithm to cluster the neighbourhoods and present the findings by plotting it on maps using Folium. The data is formated using one hot encoding with the categories of each venue. Then, the venues are grouped by neighborhoods computing the mean of each feature.

Methodology

Data mentioned above can be combined for detailed analysis. This analysis would include Correlations between different variables such as between Criminal Occurrences and Assessed Property Value. Visuals such as bar graphs and histograms will be created to visualize and understand the data better. Machine Learning algorithms will be used to cluster data to find solutions for the business problem stated above. Clustered data will be further plotted on Edmonton Map with markers for better understanding. Following are details of necessary steps to execute this methodology.

Data Acquisition and Wrangling

All the Data Sources required for this project are available online and data was easily acquired through the read-csv command. However data acquired was not in the format required for the project and needed to be transformed and reformatted. Several different processed and commands were used to complete this process, following is an example of steps taken to prepare the Criminal Occurrences Data from the City of Edmonton Neighbourhood Crime Occurrences table:

```
yeg_crime_import = pd.read_csv("https://dashboard.edmonton.ca/api/views/xthe-mnvi/rows.csv")
```

```
yeg_crime_grouped = yeg_crime_import.groupby(['Neighbourhood Description (Occurrence)']).sum()
```

Nulls were removed. There were many neighbourhoods that had fewer than 400 people, mostly industrial etc. initially the idea was to remove them, however they were retained as many of them had associated venues and it made for better analysis.

```
yeg_df = yeg_df2[yeg_df2.Population > 400]
yeg_df
```

]:

	Neighbourhood	Neighbourhood ID	Ward_x	Assessed Value	Latitude_x	Longitude_x	# Occurrences	Population	Area Sq Km
0	ABBOTTSFIELD	2010.0	Ward 7	317930	53.576324	-113.392299	976	1515	0.409629
1	ALBANY	3460.0	Ward 2	298350	53.637865	-113.547426	221	1498	0.843312
2	ALBERTA AVENUE	1010.0	Ward 2	271881	53.562133	-113.487590	5048	5101	1.680471
4	ALDERGROVE	4020.0	Ward 1	336942	53.517315	-113.646037	937	5016	1.490596
5	ALLARD	5458.0	Ward 10	374803	53.405780	-113.517552	263	6393	1.668744

Tables needed to be combined/consolidated by creating several inner joins to get all the variable data by for each neighbourhood. Below is an example of the joins:

```
In [13]: yeg_crime_value=pd.merge(yeg_price_df, yeg_crime_df, how ='inner', left_on = 'Neighbourhood', right_on = 'Neighbourhood Description (Occurrence)', validate ="1:m")
         yeg_crime_value
```

Out[13]:

	Neighbourhood	Neighbourhood ID	Ward	Assessed Value	Latitude	Longitude	# Occurrences
0	ABBOTTSFIELD	2010.0	Ward 7	317930	53.576324	-113.392299	976
1	ALBANY	3460.0	Ward 2	298350	53.637865	-113.547426	221
2	ALBERTA AVENUE	1010.0	Ward 2	271881	53.562133	-113.487590	5048
3	ALBERTA PARK INDUSTRIAL	4010.0	Ward 1	3121060	53.569709	-113.596230	260
4	ALDERGROVE	4020.0	Ward 1	336942	53.517315	-113.646037	937
...	...	...	...	...	...	...	...

Data Aggregation

Some of the tables had columns with a lot more granular information than needed, such columns were combined to create an aggregated table. Below is an example of such column aggregation:

```
sum_column = yeg_popu_import['Woman/girl'] + yeg_popu_import['Man/boy'] + yeg_popu_import['Identified as another gender']
yeg_popu_import['Population'] = sum_column
yeg_popu_import
```

5]:

	Ward	Neighbourhood Number	Neighbourhood Name	Woman/girl	Man/boy	Identified as another gender	Prefer Not to Answer	Population
0	Ward 1	4010	Alberta Park Industrial	0	0	0	0	0
1	Ward 1	4020	Aldergrove	2523	2479	14	288	5016
2	Ward 1	4011	Anthony Henday	0	0	0	0	0
3	Ward 1	4018	Anthony Henday Big Lake	0	0	0	0	0

Similar aggregations were done for rows in table different tables, for example assessed value was calculated by averaging assessed value of each property in a given neighbourhood. Similarly, number of Criminal Occurrences was calculated by aggregating number of occurrences by each neighbourhood.

```
yeg_price_df=yeg_price_clean.groupby(['Neighbourhood', 'Neighbourhood ID'], as_index = False).agg({'Ward':'first', 'Assessed Value':'mean',
       'Latitude':'first', 'Longitude':'first'})
yeg_price_df
```

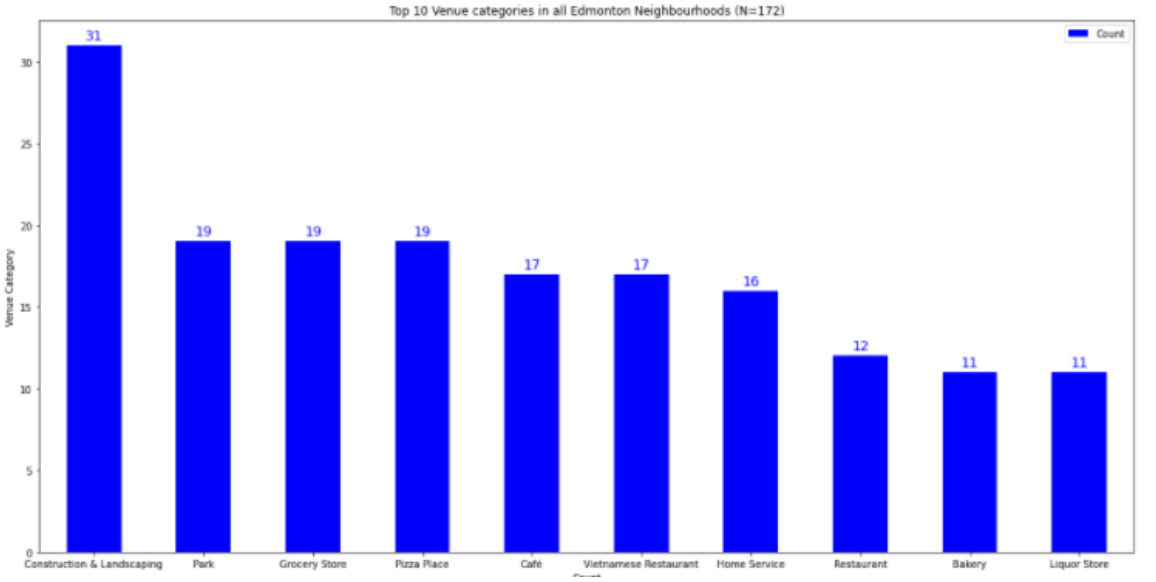
]:

	Neighbourhood	Neighbourhood ID	Ward	Assessed Value	Latitude	Longitude
0	ABBOTTSFIELD	2010.0	Ward 7	3.179302e+05	53.576324	-113.392299
1	ALBANY	3460.0	Ward 2	2.983504e+05	53.637865	-113.547426
2	ALBERTA AVENUE	1010.0	Ward 2	2.718814e+05	53.562133	-113.487590
3	ALBERTA PARK INDUSTRIAL	4010.0	Ward 1	3.121061e+06	53.569709	-113.596230
4	ALDERGROVE	4020.0	Ward 1	3.369428e+05	53.517315	-113.646037

Data Visualization

Several visualizations were created to understand the data better. For example, the visual below shows the top 10 venue categories in all Edmonton neighbourhoods. In total there were 172 venue categories.

In [40]:
```
# Plotting bar chart
area_graph = yeg_venue_top10[['Venue Category','Count']]

area_graph.set_index('Venue Category',inplace = True)

ax = area_graph.plot(kind='bar', figsize=(20, 10), rot=0, color='blue')

ax.set_ylabel('Venue Category')
ax.set_xlabel('Count')
ax.set_title(f'Top 10 Venue categories in all Edmonton Neighbourhoods (N={yeg_venue_top10.Count.sum()})')

for p in ax.patches:
    ax.annotate(np.round(p.get_height(),decimals=2),
                (p.get_x()+p.get_width()/2., p.get_height()),
                ha='center',
                va='center',
                xytext=(0, 10),
                textcoords='offset points',
                fontsize = 14,
                color='blue',
               )

plt.show()
```



Statistical Calculations

Several descriptive statistics were calculated using various pandas methods to understand the data better and also to know correlation between different variables. For example, Pearson Correlation Coefficient (Pearson's r) was calculated to measure the strength of linear relationship between two data sets.

Below is Pearson Correlation calculation between Population and # Venues showing the top 5 venues correlated to Population. This was repeated for all the other variables as well.

```
In [42]: # Calculating pearson correlations of all venue categories with population
         correlation = yeg_grouped_sum.corrwith(yeg_df["Population"], method='pearson')
         correlation.sort_values(ascending=False, inplace=True)
         print(f'Pearson correlation of top 5 venue categories with Population density: \n{correlation[:5]} \n')
         print(f'Pearson correlation of bottom 5 venue categories with Population density: \n{correlation[-5:]}')
```

```
Pearson correlation of top 5 venue categories with Population density:
Bistro                0.302514
Ethiopian Restaurant  0.302514
Theater               0.282381
Comic Shop            0.280608
Butcher               0.263498
dtype: float64

Pearson correlation of bottom 5 venue categories with Population density:
Clothing Store      -0.104686
Optical Shop        -0.107455
Salon / Barbershop  -0.107455
Scenic Lookout      -0.107455
Pizza Place         -0.114121
dtype: float64
```

Simple Correlation was calculated to understand the relationships between variables. All the variables had a positive correlation.

```
In [17]: #analyze correlations between columns
         column_1= yeg_df['Population']
         column_2= yeg_df['Assessed Value']
         #column_2= yeg_df['# Occurrences']
         #column_2= yeg_df['Area Sq Km']
         correlation = column_1.corr(column_2)
         print(correlation)
```

```
-0.2329551651531409
```

```
In [18]: #analyze correlations between columns
         column_1= yeg_df['Population']
         #column_2= yeg_df['Assessed Value']
         column_2= yeg_df['# Occurrences']
         #column_2= yeg_df['Area Sq Km']
         correlation = column_1.corr(column_2)
         print(correlation)
```

```
0.4743793223291898
```

```
In [19]: #analyze correlations between columns
         column_1= yeg_df['Population']
         #column_2= yeg_df['Assessed Value']
         #column_2= yeg_df['# Occurrences']
         column_2= yeg_df['Area Sq Km']
         correlation = column_1.corr(column_2)
         print(correlation)
```

```
-0.042456398631434016
```

Search Radius to search for venues in each neighbourhood was calculated using the following formula

$$\textbf{Search Radius (m)} = \sqrt{\frac{Area}{\pi}} \times 500$$

This changed the search radius in the FourSquare API request according to the size of each neighbourhood. Limitation of using is this formula is that neighbourhoods aren't circular and using this formula is a gross oversimplification. Below is the Python code

```
# Adding a dynamicSearch Radius column into dataframe, and re-ordering columns
# The new Search Radius will be used in the Foursquare API search query
yeg_df["Search Radius"] = yeg_df["Area Sq Km"].apply(lambda x: round(sqrt(x/pi)*500))
yeg_df
```

Machine Learning

Feature Extraction

Feature extraction involves reducing the number of resources required to describe a large set of data. It tarts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. One Hot Encoding is used for feature extraction in terms of categories. This turn each feature into binary, 1 meaning that the respective category is found in the venue and 0 means the opposite. Then, all the venues are grouped by the neighborhoods, computing at the same time the mean. This gives a venue for each row and each column will contain the frequency of occurrence of that particular category.

```
# one hot encoding
yeg_onehot = pd.get_dummies(yeg_venues_deduplicated[['Venue Category']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
yeg_onehot.insert(loc = 0, column = 'Neighbourhood', value = yeg_venues_deduplicated['Neighbourhood'])

# summing one-hot values
yeg_grouped_sum = yeg_onehot.groupby('Neighbourhood').sum().reset_index()
yeg_grouped_sum
#yeg_onehot
```

	Neighbourhood	ATM	Accessories Store	American Restaurant	Antique Shop	Arcade	Art Gallery	Arts & Crafts Store	Asian Restaurant	Athletics & Sports	...	Turkish Restaurant	Vegetarian / Vegan Restaurant	Videc Game Store
0	ABBOTTSFIELD	0	0	0	0	0	0	0	0	0	...	0	0	0
1	ALBERTA AVENUE	0	0	0	0	0	0	0	0	0	...	0	0	0
2	ALBERTA PARK INDUSTRIAL	0	0	0	0	0	0	0	0	0	...	0	0	0
3	ALDERGROVE	0	0	0	0	0	0	0	0	0	...	0	0	0

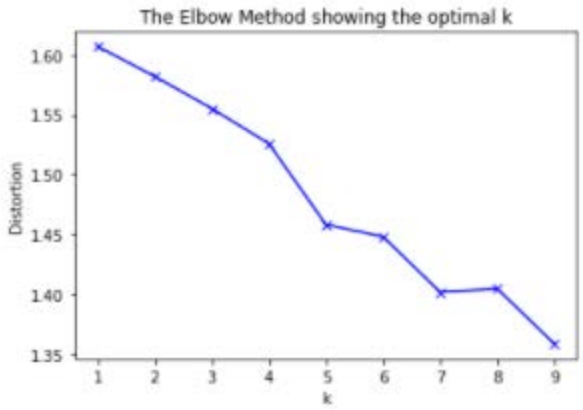
Unsupervised Learning

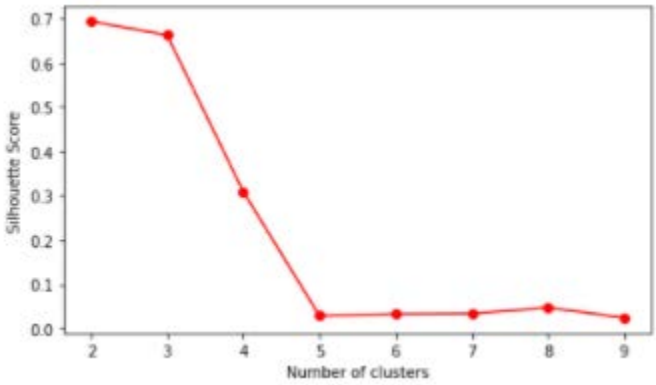
Unsupervised learning will help us in identifying undetected patterns within the neighbourhoods based on the number of venues.

K-Means:

K-Means is a clustering algorithm. This algorithm search clusters within the data and the main objective function is to minimize the data dispersion i.e. to minimize the sum of distances between the data points and the respective centroids for each cluster. Thus, each cluster represents a set of data with a similar pattern.

To determine the optimum number of clusters based on the Edmonton neighbourhoods' dataset both the elbow and silhouette methods were used.





Based on both the methods, number of clusters was determined to 5.

Following is the formula to calculate cluster labels

```
: # set number of clusters
  kclusters = 5
  yeg_grouped_clustering = yeg_grouped_sum.drop('Neighbourhood', 1)

  # run k-means clustering
  kmeans = KMeans(n_clusters=kclusters, random_state=0).fit(yeg_grouped_clustering)


  # check cluster labels generated for each row in the dataframe
  kmeans.labels_[0:10]
```

```
: array([0, 0, 0, 4, 0, 3, 4, 0, 0, 0], dtype=int32)
```

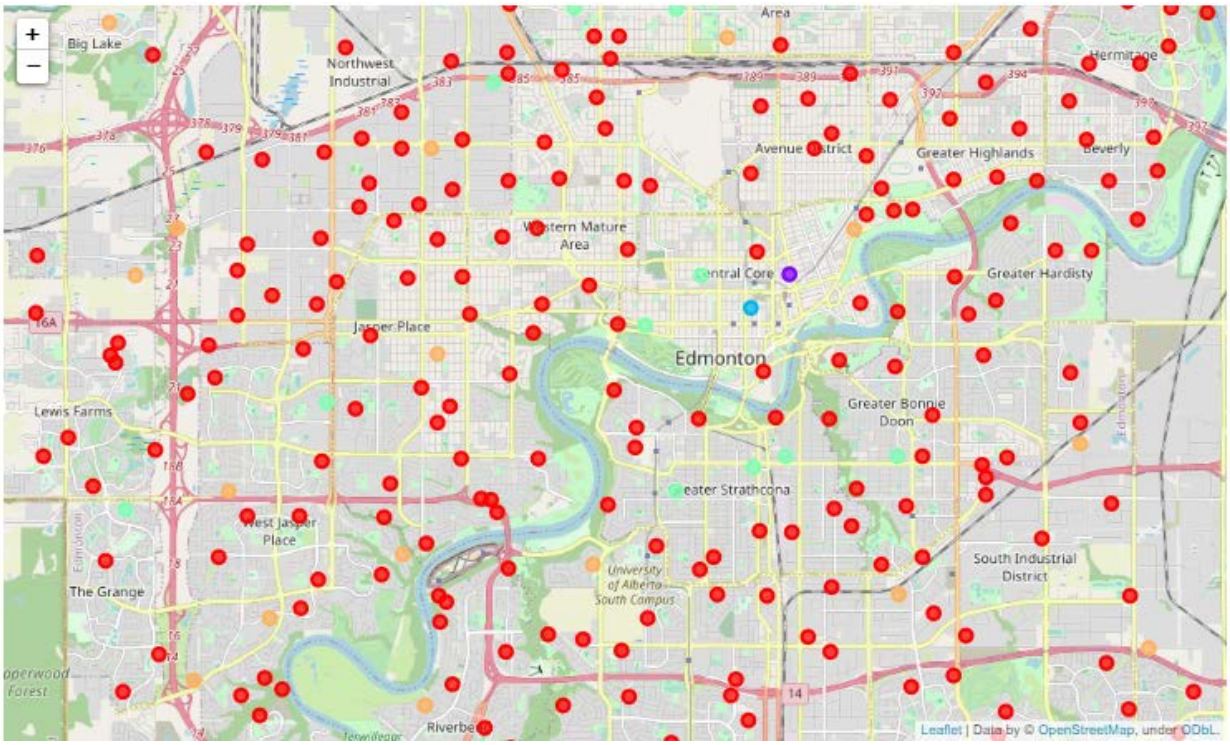
Following is a summary of the resulting clusters

```
# Summary of cluster labels
yeg_merged['Cluster Labels'].value_counts()
```

```
0    339
4     25
3     18
2      1
1      1
Name: Cluster Labels, dtype: int64
```

Data Mapping

Using the Map() function from the folium library, all neighbourhoods are plotted on Edmonton's map with different coloured map markers according to the cluster labels.



Results and Discussion

The objective of the business problem was to provide evidence in support of choosing the best neighbourhood for opening of businesses in Edmonton, Alberta, Canada. To provide a more complete picture variables beyond population and venue density were used, which were safety (through criminal occurrences), property value (Assessment values) and Area of the neighbourhoods.

Data showed us that some of neighbourhoods had 0 population as they were industrial areas, however they were still included because they still had venues. Still there were many neighbourhoods that has very little to no venues.

We also found that more populated neighbourhoods had a higher correlation with more entertainment related (restaurants, theatre etc.) venues, whereas neighbourhoods with higher property values had higher correlations with businesses such as Construction & Landscaping.

Constructions & Landscaping was the venue category with the highest count of venues and Downtown neighbourhood had most venues.

Clustering the venues data showed patterns and helped in validating some of the prior notions. Most of the city neighbourhoods were clustered together, showing a lack of diversity, still the clusters would help a user to identify which neighbourhoods could use new businesses and provide much more robust information and evidence to make decision.

Conclusion

This Capstone project has met it's outlined objective of providing a user with information to support their decision in opening a new commercial establishment in an Edmonton neighbourhood. This project achieved this through acquiring data, wrangling data, normalizing data, running machine learning algorithms, mapping data, and other statistical operations such as correlations testing. This project shows the advantage of open data and how through data science techniques advanced analysis can be done at no cost, with basic IT resources and on a quick turnaround.