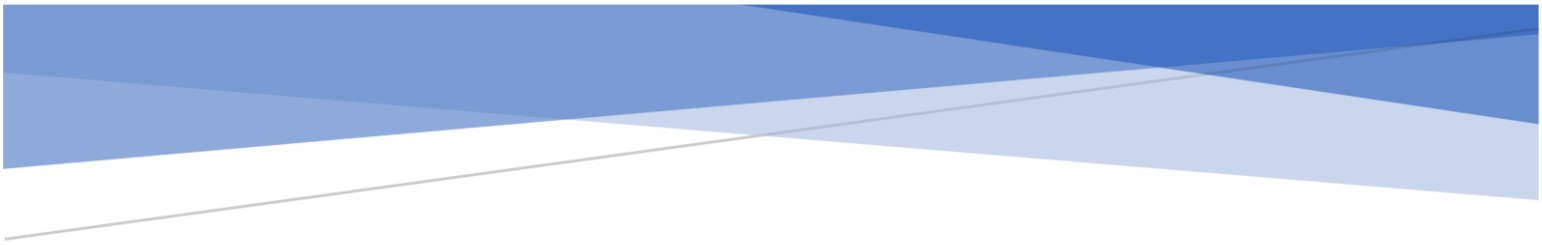

IBM DATA SCIENCE PROFESSIONAL CERTIFICATE

Applied Data Science Capstone: Analysis of towns in the UK using population and venues data

Abstract

The aim of the Applied Data Science Capstone of the IBM Data Science Professional Certificate is to leverage Foursquare location data to analyze cities and solve a problem. I performed regression analysis on venue and geographical data in the UK to determine if there is a relationship between the amount of venues of different venue categories and the population of towns in the UK. Using classification and clustering machine learning algorithms on the venues and geographical data, I also classified cities in the UK according to relative amount of different venues, which would be helpful analysis for someone potentially aiming to start a business or settle in a city in the UK.

Victor George

1.1BACKGROUND

The United Kingdom (U.K.) is a nation made up of the political union of England, Scotland, Wales and Northern Island. The United Kingdom is currently one of the most powerful and influential countries in the world and is currently the 5th ranked economy in the world by Gross Domestic Product (World Population Review, 2020). Major drivers of the U.K. economy are the tourism, transport, energy and science and technology industries and thousands of businesses associated with these major industries operate throughout the U.K. The U.K. is also one of the top tourist destinations in the world, and London yearly attracts one of the highest numbers of international visitors in the world. The U.K. has a very rich history and cultural diversity, and as one of the world's largest economies is one of the hubs for immigration, settlement and business.

1.2 BUSINESS PROBLEM

Towns in the U.K. exhibit a considerable degree of variation for a variety of sociocultural and geopolitical reasons. This project will use machine learning techniques to conduct two main analyses:

1. Investigate the relationship between the total number of venues of certain venue categories in a town and its population
2. Classify towns in the U.K. according to shared venue category characteristics.

The insight gained from this analysis would be of potential interest to someone aiming to immigrate to the UK or to establish a business or branch of a multinational business in the U.K.

2.) DATASET AND SOFTWARE

This project was performed in *Jupyter Notebook*™ using the *Pandas* and *NumPy* libraries for data analysis and *Scikit-learn* for machine learning. The full Jupyter Notebook code can be viewed **here**. The data used for this project is Foursquare venues data and geographic data for all the postcodes in the U.K. The geographic data hosted on consists of the 3096 UK postcodes as of the time this data was compiled and for each of them, their geolocation, the population and the number of households The total population in this DataFrame Is 63,153,528 about 4 million less than the live UK population of about 67.9 million (Worldometer, 2020). Each town has multiple postcodes and grouping the data by 'Region' results in **424** unique towns.

Region	Latitude	Longitude	Easting	Northing	Postcodes	Active postcodes	Population	Households
Aberdeen	1086.4526	-41.650320	7381353.0	15392398.0	23481.0	6855.0	245387.0	112158.0
Aberdeenshire	974.0809	-42.558970	6285182.0	13996142.0	13255.0	8915.0	223979.0	93341.0
Adur	101.6721	-0.500124	1046641.0	211033.0	1055.0	739.0	32639.0	14269.0
Allerdale	328.0630	-20.325460	1863739.0	3193442.0	4813.0	4127.0	98058.0	43111.0
Amber Valley	212.1896	-5.628090	1759396.0	1400721.0	4805.0	2904.0	131664.0	56522.0
...	...	...	...	...	...	...	...	...
Wychavon	208.5272	-8.033620	1598138.0	992687.0	4572.0	2888.0	95934.0	40902.0
Wycombe	413.2197	-6.068381	3887867.0	1564660.0	6381.0	4172.0	157892.0	61856.0
Wyre	161.6552	-9.030260	1001113.0	1331045.0	3318.0	2592.0	98490.0	44100.0
Wyre Forest	209.4953	-9.094130	1525899.0	1100490.0	3970.0	2996.0	100374.0	44004.0
York	1079.3580	-21.104093	9241582.0	9056868.0	19840.0	7902.0	271657.0	113371.0

424 rows × 8 columns

Figure 1 - Geographic Data

The venues data (Figure was created by using Foursquare © API to fetch the nearest 100 venues within a 5 km radius of each postcode in the geographic data. A total of 265932 venues were returned by this request.

	Town/Area	Town/Area Latitude	Town/Area Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Aberdeen	57.1124	-2.243510	Northern Lights Lounge	57.199933	-2.202709	Airport Lounge
1	Aberdeen	57.1034	-2.272700	Northern Lights Lounge	57.199933	-2.202709	Airport Lounge
2	Aberdeen	57.2084	-2.200920	Northern Lights Lounge	57.199933	-2.202709	Airport Lounge
3	Aberdeen	57.1865	-2.119590	Northern Lights Lounge	57.199933	-2.202709	Airport Lounge
4	Aberdeen	57.2083	-2.089800	Northern Lights Lounge	57.199933	-2.202709	Airport Lounge
...	...	...	...	...	...	...	...
265927	York	53.9599	-1.090960	Sant Angelo	53.928397	-1.385169	Wine Bar
265928	York	53.9267	-0.815105	Wildes	53.960637	-1.082206	Wine Bar
265929	York	53.9267	-0.815105	Field And Fawcett	53.956005	-1.014766	Wine Bar
265930	York	54.1544	-1.080350	National Centre for Birds of Prey	54.241132	-1.076086	Zoo
265931	York	54.0846	-0.947672	National Centre for Birds of Prey	54.241132	-1.076086	Zoo

265932 rows × 7 columns

Figure 2 - UK Venues Data from Foursquare ©

2.2 Data Preparation

Before analysis could be performed on the data, the geographic data and venues data were merged into one dataset. In the venues data, for some towns, a venue was repeated multiple times, with the same latitude and longitude. This was removed by creating a unique name for each record, by combining the venue name, town, venue latitude and venue longitude in a separate column and removing the duplicates of this unique name. After removing duplicates, 76318 venues from 419 venue categories in 424 towns remained (Figure 3).

	Town/Area	Town/Area Latitude	Town/Area Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Aberdeen	57.1124	-2.243510	Northern Lights Lounge	57.199933	-2.202709	Airport Lounge
5	Aberdeen	57.1269	-2.136440	No 10 Tavern	57.143409	-2.121221	Pub
6	Aberdeen	57.1269	-2.136440	CASC ABERDEEN	57.145874	-2.097568	Pub
7	Aberdeen	57.1269	-2.136440	Revolucion de Cuba	57.147413	-2.101209	Pub
8	Aberdeen	57.1269	-2.136440	No. 1 Bar/Grill	57.143286	-2.119574	Pub
...	...	...	...	...	...	...	...
265899	York	53.9581	-1.071130	Goji Vegetarian Cafe and Deli	53.961740	-1.079906	Restaurant
265918	York	53.9484	-1.121640	Sant Angelo	53.928397	-1.385169	Wine Bar
265919	York	53.9082	-0.829513	Wildes	53.960637	-1.082206	Wine Bar
265920	York	53.9082	-0.829513	Field And Fawcett	53.956005	-1.014766	Wine Bar
265930	York	54.1544	-1.080350	National Centre for Birds of Prey	54.241132	-1.076086	Zoo

76318 rows × 7 columns

Figure 3 - UK Venues Data after removing duplicates.

2.1.1. Feature Scaling

After removing duplicates, the venues categories, which are categorical variables[1] were converted to dummy variables[2] using one-hot encoding. The first column in the resulting DataFrame[3] is the town where the venue exists, and the rest of the columns correspond to all the venues in the dataset. For each venue (row), the number 1 is assigned to the venue category it corresponds to, and 0 is assigned to every other venue category.

	Town/Area	Theater	Pub	Restaurant	Coffee Shop	Brewery	Museum	Park	Movie Theater	History Museum	Italian Restaurant
139186	Liverpool	0	1	0	0	0	0	0	0	0	0
139616	Liverpool	0	0	1	0	0	0	0	0	0	0
140300	Liverpool	0	0	0	1	0	0	0	0	0	0
140384	Liverpool	0	0	0	1	0	0	0	0	0	0
140902	Liverpool	0	0	0	0	0	0	0	0	0	0
141751	Liverpool	0	0	0	0	0	0	0	0	0	1
141855	Liverpool	0	0	0	0	0	0	0	0	0	1
141900	Liverpool	0	0	0	0	0	0	0	0	0	0
142077	Liverpool	0	0	0	0	0	0	0	0	0	0
143366	Liverpool	0	0	1	0	0	0	0	0	0	0
143387	Liverpool	0	0	0	0	0	0	0	0	0	0
143831	Liverpool	0	0	1	0	0	0	0	0	0	0
143905	Liverpool	0	0	0	0	0	0	0	0	0	0
144064	Liverpool	0	0	0	0	0	0	0	0	0	0

Figure 4 – One hot encoded venue data for a few venues in one of the towns, Liverpool. A few of the 351 columns were selected for visualization, so of the 14 venues in the figure, only 7 venues have a category.

To make the frequency of venue categories comparable across towns, two DataFrames were produced. In the first DataFrame, DataFrame I (Figure 5), for each town, the dummy variable was summed to obtain the total number of venues in that category . The venue category totals for each town were then normalized by dividing by the maximum venue category total for that venue category giving the values a range between 0 and 1.

Town/Area	Accessories Store	Airport	Airport Lounge	Airport Service	Airport Terminal	Amphitheater	Antique Shop	Apres Ski Bar	Aquarium	Arcade	...	Wine Bar	Wine Shop	Winery	Wings Joint	Women's Store	Yoga Studio	Zoo	Zoo Exhibit	Population (standardized)	Households(standardized)
Aberdeen	0.0	0.0	0.142857	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.712070	0.882569
Aberdeenshire	0.0	0.0	0.285714	0.0	0.0	0.0	0.0	0.0	0.5	0.0	...	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.554003	0.549234
Adur	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.858766	-0.851495
Allerdale	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.5	0.0	...	0.00	0.0	0.0	0.0	0.0	0.0	0.2	0.0	-0.375742	-0.340570
Amber Valley	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.127610	-0.103000
Anglesey	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.5	0.0	...	0.00	0.0	0.0	0.0	0.0	0.0	0.2	0.0	-0.584748	-0.562304
Angus	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.0	0.0	0.0	0.0	0.0	0.2	0.0	-0.338004	-0.284929
Antrim	0.0	0.0	0.142857	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.800776	-0.828855
Archiestown	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-1.099758	-1.104264
Ards	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.00	0.0	0.0	0.0	0.0	0.0	0.2	0.0	-0.650299	-0.666253

10 rows × 354 columns

Figure 5 – DataFrame I: Normalized Venue Category Totals

[1] A variable that can take several different nominal values based on a qualitative property (Wikipedia, 2020).
[2] Or indicator variable. A variable that takes the value 0 or 1 to indicate that it belongs to a category.
[3] 2-dimensional labeled data structure with columns of potentially different types (The Pandas Development Team, 2014).

In the second DataFrame, DataFrame II (Figure 6), for each town, the sum of the dummy variables in each category, was divided by the total number of venues to obtain the relative number of venues in that category to other venues or the **venue category density**. For example, if a town has 100 venues, and 20 of them are pubs, the venue category density for Pubs in that town would be 20/100 = 0.2. So, 20% of the venues in that town are Pubs. The venue category density for each town was then normalized by dividing by the maximum venue category density for that venue category. This gave venue category densities normalized values ranging between 0 and 1.

Town/Area	Accessories Store	Airport	Airport Lounge	Airport Service	Airport Terminal	Amphitheater	Antique Shop	Apres Ski Bar	Aquarium	Arcade	...	Wine Bar	Wine Shop	Winery	Wings Joint	Women's Store	Yoga Studio	Zoo	Zoo Exhibit	Population (standardized)	Households(standardized)
Aberdeen	0	0	0.120482	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0.712070	0.882569
Aberdeenshire	0	0	0.13986	0	0	0	0	0	0.328671	0	...	0	0	0	0	0	0	0	0	0.554003	0.549234
Adur	0	0	0	0	0	0	0	0	0	0	...	0.533981	0	0	0	0	0	0	0	-0.858766	-0.851495
Allerdale	0	0	0	0	0	0	0	0	0.606452	0	...	0	0	0	0	0	0	0.154839	0	-0.375742	-0.340570
Amber Valley	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	-0.127610	-0.103000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Wychavon	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0.301255	0	-0.391424	-0.379702
Wycombe	0	0	0	0	0	0	0	0	0	0	...	0	0	1	0	0	0	0	0	0.066046	-0.008510
Wyre	0	0	0	0	0	0	0	0	0	0	...	0.814815	0	0	0	0	0	0	0	-0.372552	-0.323051
Wyre Forest	0	0	0	0	0	0	0	0	0	0	...	0.37931	0	0	0	0	0	0.165517	0.172414	-0.358641	-0.324751
York	0	0.0323129	0	0	0	0	0	0	0	0	...	0.561224	0	0	0	0	0	0.0816327	0	0.906036	0.904057

424 rows × 354 columns

Figure 6 – DataFrame II – Normalized Venue Category Density.

In the geographic data dataset, the Population and Households columns were summed for each town and normalized. This gives them a comparable scale to the Venue Categories. The geographic data was joined with the normalized venue category totals to produce DataFrame I. The geographic data was also joined to the DataFrame with the normalized venue category densities to produce DataFrame II. Figure 7 shows the distribution of population and one of the venue categories, 'Pub', in both datasets.

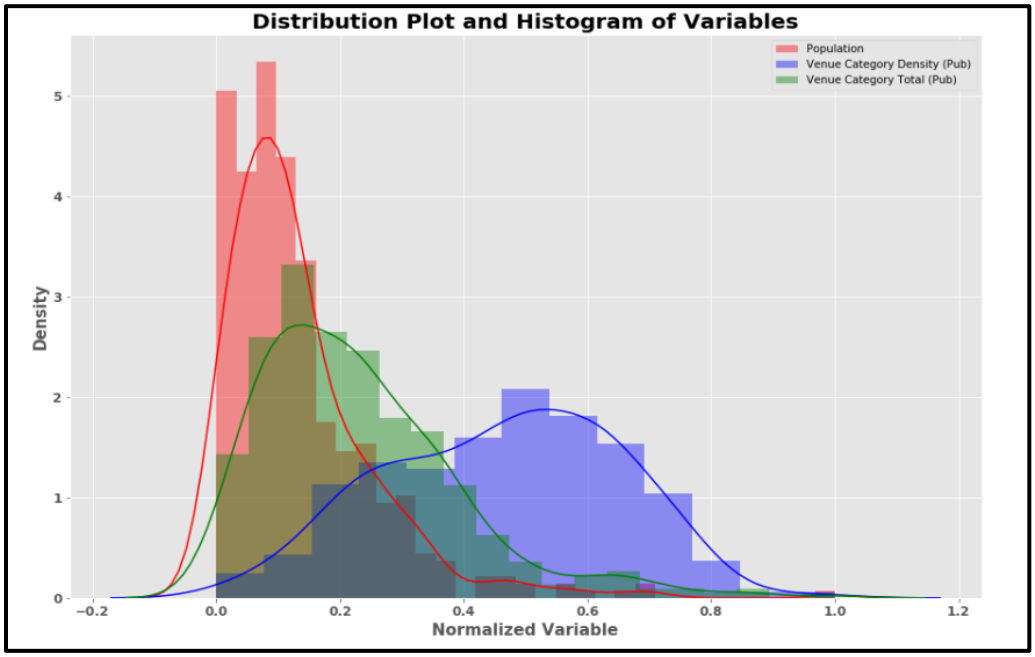

Figure 7 - Distribution of Normalized Variables

3.) METHODOLOGY

3.1 Exploratory Data Analysis

An exploratory analysis of the data was performed to obtain a general understanding of the relationships within the data. In DataFrame I, a correlation between each of the venue category totals – the features, and the normalized population was performed to investigate if there is a relationship between the total number of venues in a venue category and the population of a town. This was quantified using the Pearson coefficient and the P-value. The higher the magnitude of the Pearson coefficient, the higher the correlation with the population. The lower the P-value, the higher the certainty in the correlation coefficient, or the more statistically significant the feature is. A feature with a P-value of <0.001 is generally considered statistically significant. Table 1 shows the features with the lowest ten P-values, which are also the features with the highest ten Pearson coefficients. This analysis also helped to identify features which had no correlation with population. Some features which were not strongly correlated, or generally had Pearson coefficients of less than <0.3 and were considered redundant, were merged in the original dataset, and the steps in Section 2 repeated to reproduce the analysis sets with three hundred and fifty-one features.

Feature	Pearson coefficient	P-value
Households(normalized)	0.996	0
Theater	0.505	8.35E-29
Pub	0.484	2.98E-26
Restaurant	0.470	1.22E-24
Coffee Shop	0.441	1.20E-21
Museum	0.413	7.32E-19
Park	0.392	4.83E-17
Indian Restaurant	0.392	5.47E-17
Movie Theater	0.390	7.62E-17
History Museum	0.378	7.34E-16

Table 1 – Features with the lowest 10 P-values

The features which were most strongly correlated with population were not necessarily the highest frequency features in the dataset. By summing the dummy variable for each feature, the frequency of each feature in the dataset could be determined. Figure 8 shows the top 10 features by frequency in the dataset.

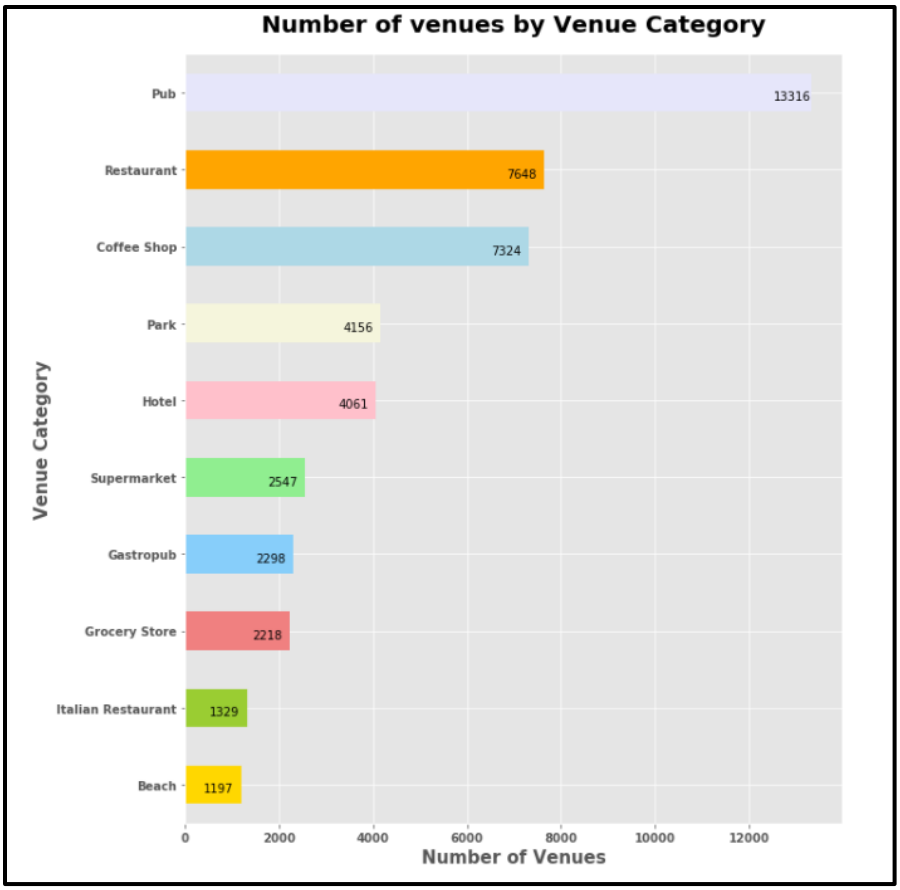

Figure 8 – Top 10 venue categories by frequency in the dataset

The top 10 venue categories by frequency account for 60% of all the venues in the dataset (Figure 9)

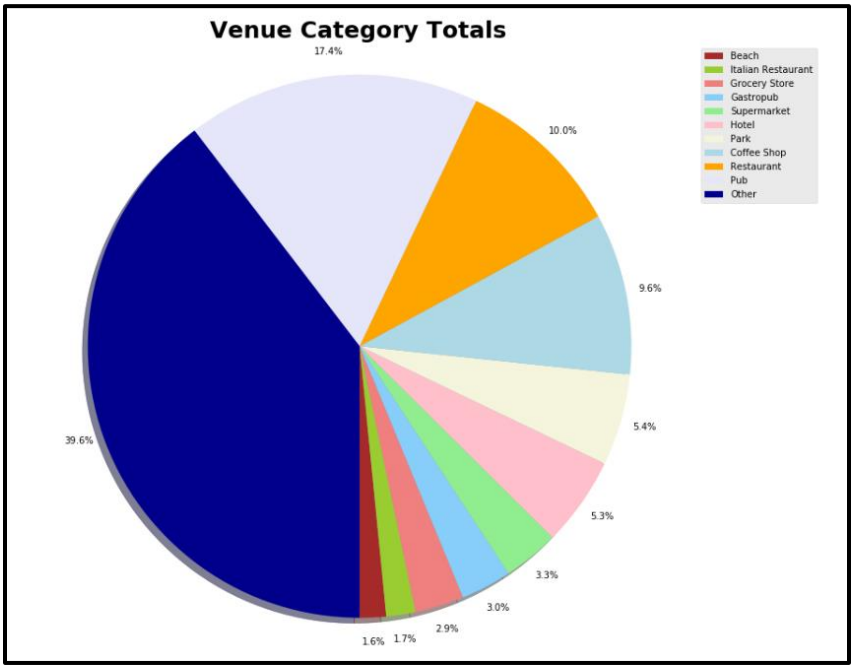

Figure 9 – Pie chart showing the total number of venues per venue category.

It would be expected that the highest frequency venue categories are the most highly correlated with population. However, a comparison between the highest frequency venue categories and the venue categories most highly correlated to population, reveals some categories, for example, Beach, which have a high frequency in the dataset but are not as highly correlated with population.

Feature	Pearson Coefficient	P-value
Pub	0.473	5.35E-25
Restaurant	0.470	1.22E-24
Coffee Shop	0.441	1.2E-21
Park	0.392	4.83E-17
Italian Restaurant	0.369	3.83E-15
Gastropub	0.299	3.48E-10
Hotel	0.294	6.91E-10
Supermarket	0.293	7.56E-10
Grocery Store	0.257	7.84E-08
Beach	0.142	0.003449

Table 2 - Pearson coefficient and P-value for Top 10 venues by frequency

Figure 10 shows box plots for the top 10 venue categories by frequency in the dataset.

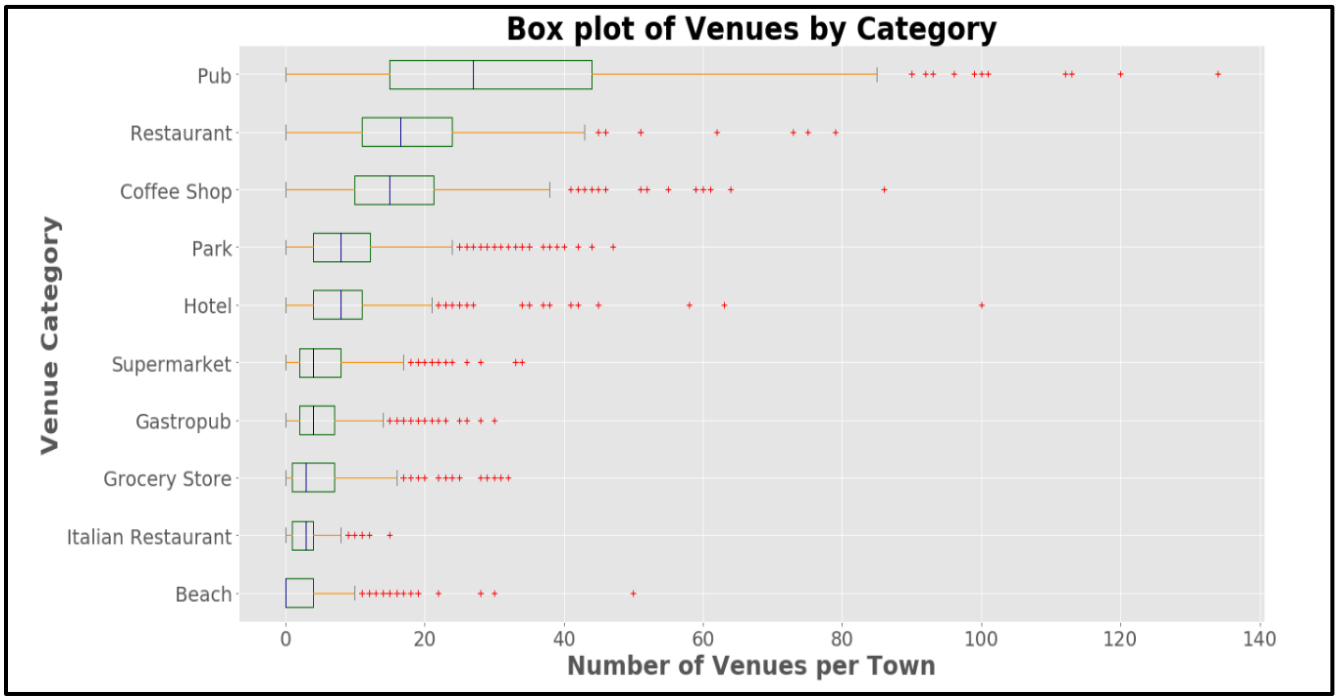

Figure 10 - Box plot of Top 10 venue categories

The average town in the UK has 180 venues. Of the 180 venues, on average 31 are pubs (~17%), 18 are restaurants (10%) and 17 are coffee shops (~9.4%) and 42 belong to the other of the top 10 venue categories (23%). So, these top 10 venue categories account on average for about 62.4% of the venues in a town. Hence, these venues categories are considered important features in the makeup of a town. The distribution of total venues in each venue category is shown in Figure 11.

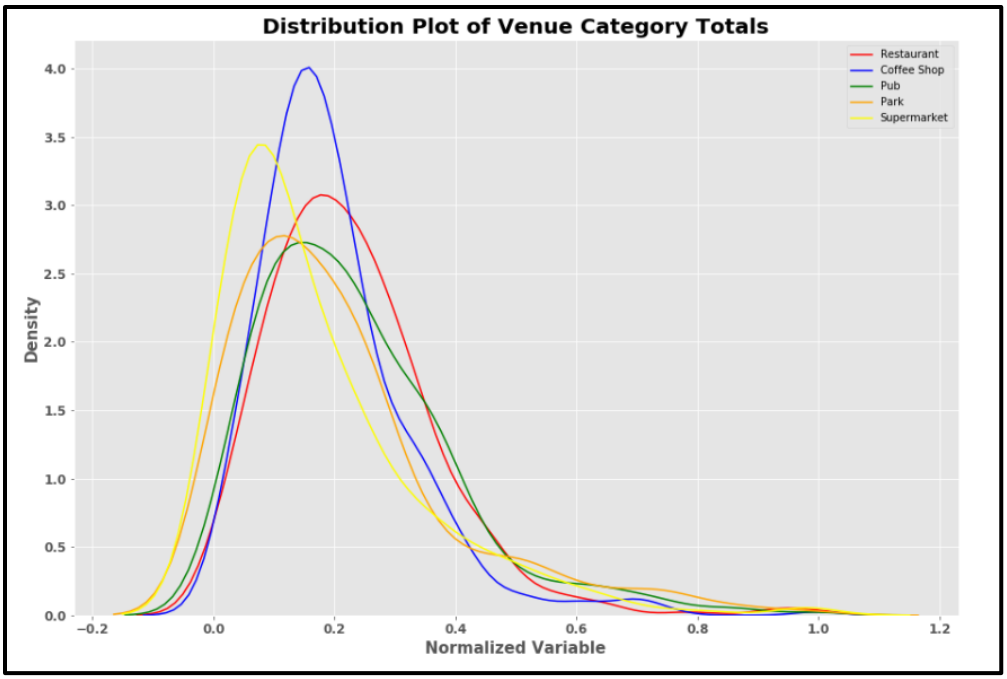

Figure 11 – Distribution of some of the top venue category totals

Following analysis of the venue categories, the population was analyzed by plotting a histogram of the population (Figure 12). The distribution of the population resembles a log-normal distribution. Seventy-five percent of the towns have populations which are less than a standard deviation away from the mean. It was then divided into six evenly spaced bins (Figure 13), which capture the variation in the population. Each town was then assigned to a bin. The majority of the towns (309 of 424) belong to Bin 1, which has a maximum population near the 75% percentile.

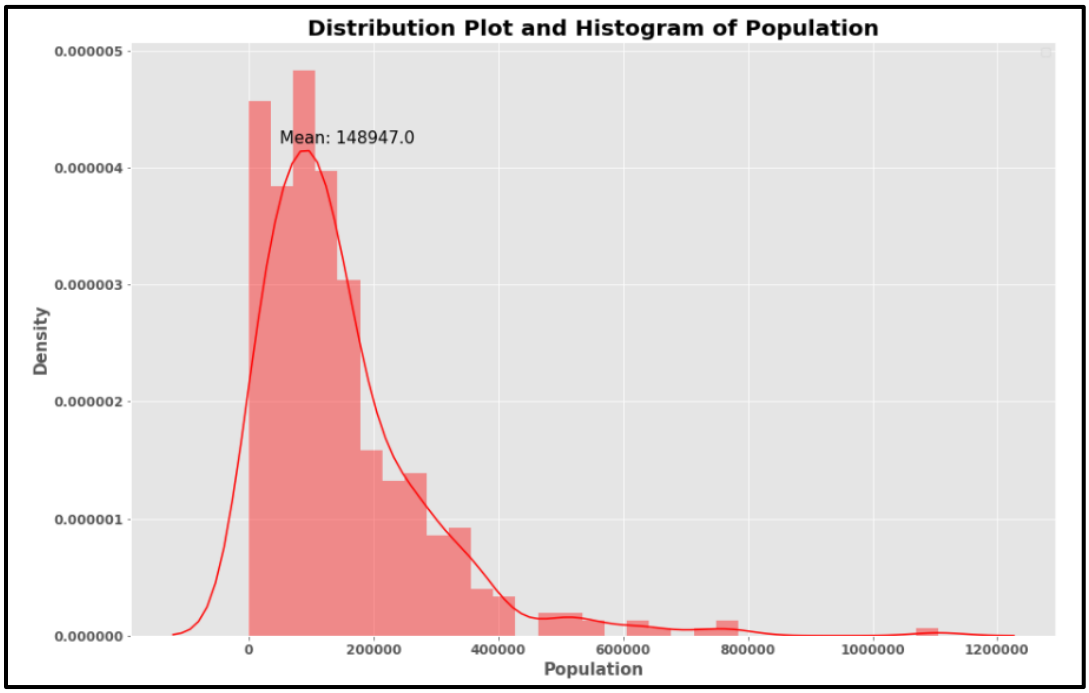

Figure 12 - Population Histogram and Distribution Plot

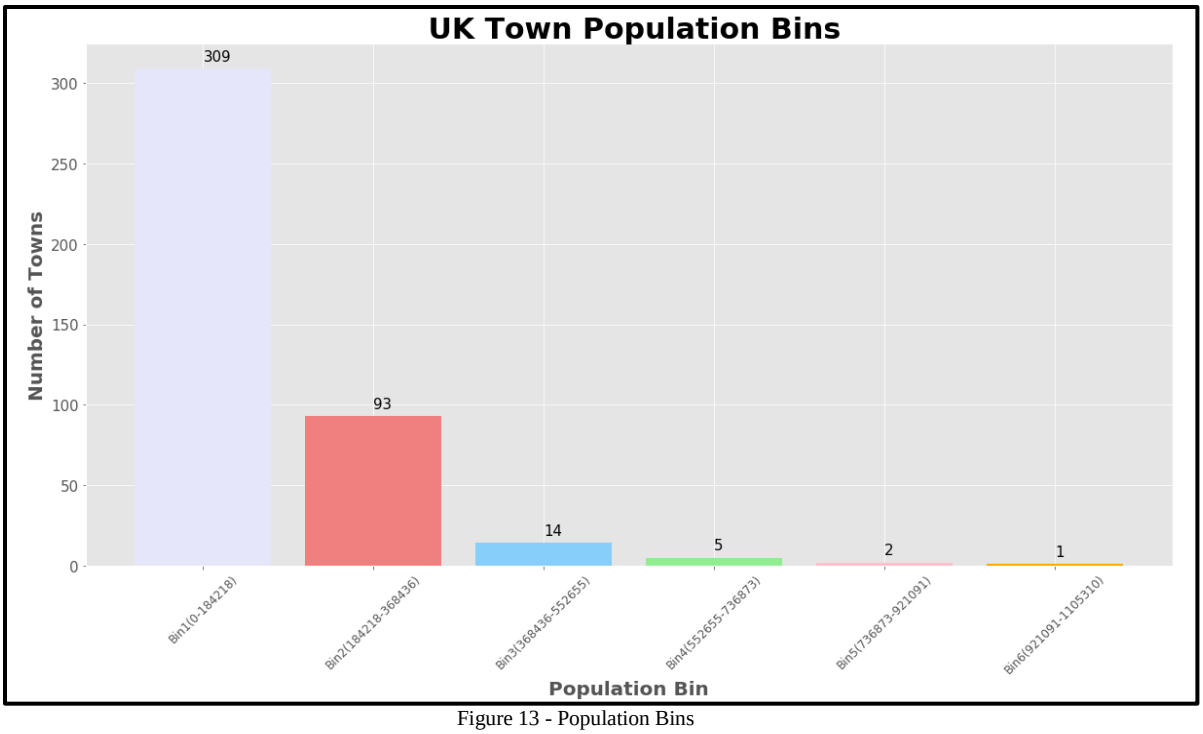

Figure 13 - Population Bins

3.2 Data Analysis

The series of analyses which were carried out on the data are:

1.) Regression
2.) Clustering
3.) Classification

3.2.1– Selecting Model Parameters

To determine the optimal input parameters for some of the regression and classification models, an Exhaustive Grid Search was performed. An Exhaustive Grid Search roams a hyper parameter space, i.e. a grid of specified parameters which are used to control the learning process, and considers all hyparameter combinations, to obtain a dictionary of hyperparameters which results in the highest cross-validation accuracy (Scikit-Learn Developers, 2019).

3.2.2 Regression

3.2.2.1 Regression Feature Selection

The features in DataFrame I, which were most highly correlated with the population were selected to perform Regression. A minimum threshold of around 0.3% for Pearson Coefficient and maximum threshold of 0.001 for P-value, in conjunction with intuitive understanding of the feature was used to select suitable features. Households is perfectly (99.99%) correlated with Population, so this was eliminated as there is not enough variation to make it an interesting independent variable. Since Italian Restaurant and Indian Restaurant have such a high correlation to population whereas all other types of restaurants are not considerably highly correlated, these features were not merged with Restaurant feature. The features selected for regression were:

Regression Features
Theatre
Pub
Restaurant
Coffee Shop
Museum
Park
Indian Restaurant
Movie Theater
History Museum
Italian Restaurant

Table 3 - Regression Features

3.2.2.2 – Regression Model Evaluation

The accuracy of the models was evaluated using the Mean-Square Error (MSE) and R-Square evaluation metrics. The mean-squared error is the sum of the squared prediction errors (difference between true value and predicted value). The larger this value, the less accurate the predictions are.

The R-Square, also known as the Coefficient of Determination, measures how close the data is to the fitted regression line. It also quantifies how much of the variation in the quantity is described by a model. The formula for R-Square is:

$$R^2 = 1 - \frac{MSE\ of\ regression\ line}{MSE\ of\ average\ of\ the\ data}$$

R^2 mainly ranges between 0 and 1. 1 corresponds to a perfect fit, (100% of the variation of the quantity is described by this model), 0 corresponds to no fit (0% of the variation of the quantity is described by the model, i.e. the model is equal to the average of the data) and less than zero usually signifies overfitting..

In order to perform an in-sample evaluation of the regression models, each model was trained using all the samples of the feature matrix and their target values – the population, and target values were predicted for all the samples. The R-Square and MSE were then computed.

Out of sample evaluation was performed using Cross Validation. Cross Validation splits the dataset into equal sized partitions where one partition is used as the test set and the rest forms the training set. Predictions are performed on the test set and the process is iterated until each partition has been used as the test set. At each train/test split, the accuracy is computed, and the mean value of the accuracy describes the general out of sample performance of the model.

Distribution plots were also generated to compare the distribution of predictions to the distribution of the actual target value for each of the models.

3.2.2.3 – Simple Linear Regression

Simple Linear Regression was performed to determine if there are linear functions which describe the relationship between a single feature and population using DataFrame I. This was performed for five of the regression features - Theatre, Pub, Restaurant, Coffee Shop and Movie Theater. These formed separate feature vectors.

3.2.2.4 – Multiple Linear Regression

Multiple Linear Regression (MLR) was performed on DataFrame II, to determine a linear function in terms of all the features which describes the relationship between all the features and population. The MLR function is of the form:

$$\hat{Y} = a\widehat{X1} + b\widehat{X2} + c\widehat{X3} + \cdots + n\widehat{XN} + p$$

OR

$$\hat{Y} = \widehat{coefficient}\ vector \cdot \widehat{feature}\ matrix + intercept$$

Where $\hat{Y}$ is the target variable vector, a, b, c…n are the coefficients of the linear function, $\widehat{X1}$, $\widehat{X2}$…$\widehat{XN}$ are individual feature vectors and p is the intercept.

The Linear Regression model was trained on the feature matrix, consisting of all 10 regression features and the corresponding true values for the population and values for the population were predicted using the model for all the samples of the feature matrix.

3.2.2.5 – Linear Regression with Polynomial Features

The relationship between all the features and the population can also be described with an n-degree polynomial transformation of all the features. In fact, the relationship between individual features and population may be better approximated by non-linear functions (Figure 14).

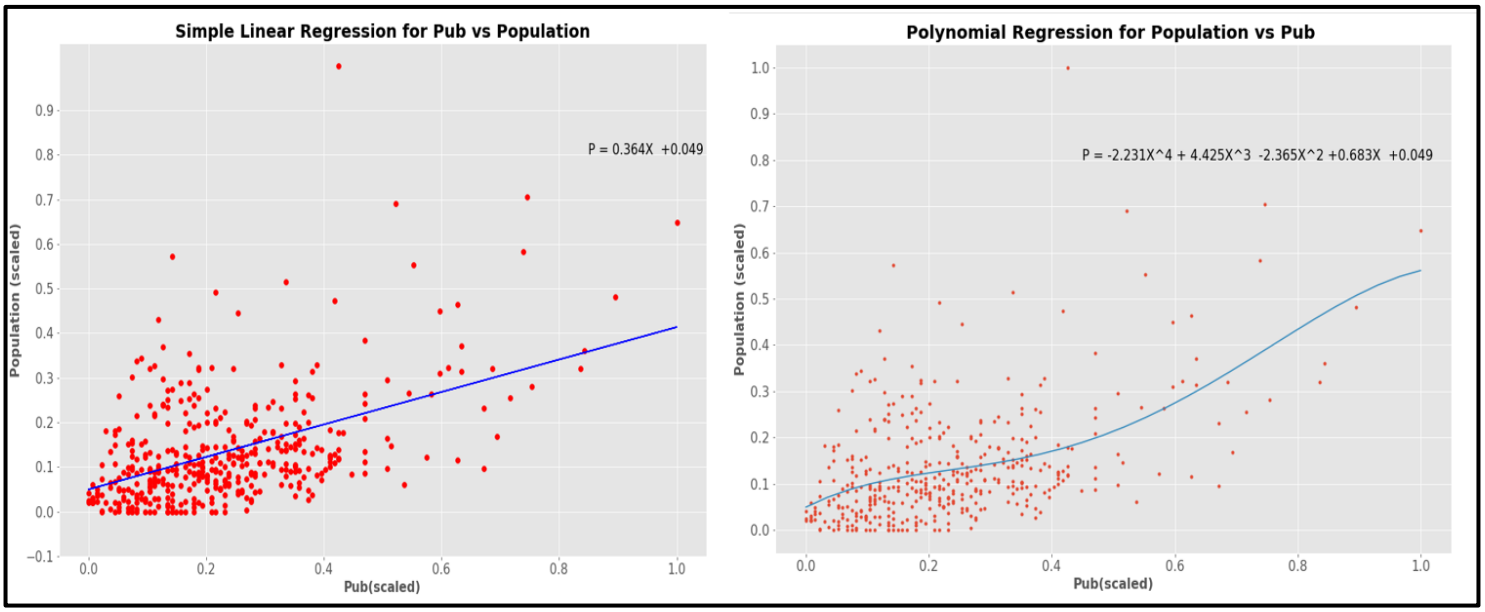

Figure 14 - Simple Linear Regression vs. Polynomial Regression for individual feature (Pub)

The order for the polynomial feature transformation was decided by plotting the cross-validation accuracy as a function of polynomial transformation order and selecting the order which corresponds to the peak of this curve (Figure 15).

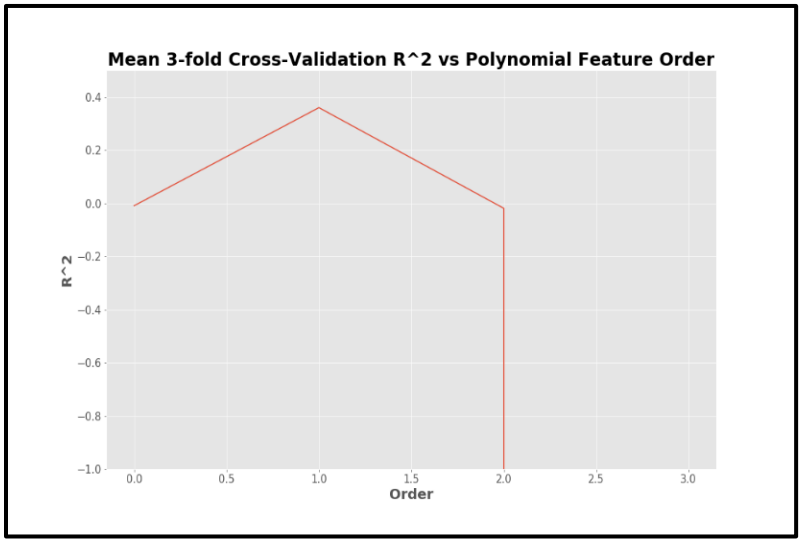

Figure 15 - Selecting order for Polynomial Transformation

The curve indicates that a 1st-degree polynomial transformation should be chosen. A first-degree polynomial transformation is identical to the Multiple Linear Regression Model, as the features are all first order. However, a 2nd-degree polynomial transformation was also used for comparison. Under a 2nd-degree polynomial transformation, the 10 features became 66 features where the maximum degree of an individual feature is 2. A linear regression model was then fit to which assigned coefficients to these 66 features.

Original Features	
x0	Theater
x1	Pub
x2	Restaurant
x3	Coffee Shop
x4	Museum
x5	Park
x6	Indian Restaurant
x7	Movie Theater
x8	History Museum
x9	Italian Restaurant

Table 4 - Original 11 Regression Features

POLYNOMIAL FEATURES										
0	1	2	3	4	5	6	7	8	9	10
1	**x0**	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**
11	12	13	14	15	16	17	18	19	20	21
x0^2	**x0 x1**	**x0 x2**	**x0 x3**	**x0 x4**	**x0 x5**	**x0 x6**	**x0 x7**	**x0 x8**	**x0 x9**	**x1^2**
22	23	24	25	26	27	28	29	30	31	32
x1 x2	**x1 x3**	**x1 x4**	**x1 x5**	**x1 x6**	**x1 x7**	**x1 x8**	**x1 x9**	**x2^2**	**x2 x3**	**x2 x4**
33	34	35	36	37	38	39	40	41	42	43
x2 x5	**x2 x6**	**x2 x7**	**x2 x8**	**x2 x9**	**x3^2**	**x3 x4**	**x3 x5**	**x3 x6**	**x3 x7**	**x3 x8**
44	45	46	47	48	49	50	51	52	53	54
x3 x9	**x4^2**	**x4 x5**	**x4 x6**	**x4 x7**	**x4 x8**	**x4 x9**	**x5^2**	**x5 x6**	**x5 x7**	**x5 x8**
55	56	57	58	59	60	61	62	63	64	65
x5 x9	**x6^2**	**x6 x7**	**x6 x8**	**x6 x9**	**x7^2**	**x7 x8**	**x7 x9**	**x8^2**	**x8 x9**	**x9^2**

Table 5 – Transformation into 66 Polynomial Features

3.2.2.6 – Ridge Regression

Linear Regression with 2nd-degree polynomial features transformation produces a function with higher in-sample accuracy than simple linear regression or multiple linear regression and a distribution of the predicted values which more closely matches that of the true values, but the coefficients may not reflect the true relationships in the data and the model is liable to overfitting. Additionally, the calculated function for each train/test split is not always as accurate since for each train/test split, a function is derived from only the training features and training target variable and it is tailor-made for that training set. Ridge Regression is a linear regression model that tunes the original model, to prevent overfitting, by using the parameter, alpha, to control the magnitude of the coefficients and improve out-of-sample accuracy. Alpha was found using an Exhaustive Grid Search.

Ridge Regression is particularly useful for polynomial regression or linear regression with higher order features as alpha controls the size of the coefficients of the higher order features. It doesn't have a significant effect on 1st-order features or simple linear regression. Hence, Ridge Regression was used to determine an alternative linear regression with 2nd-order polynomial feature transformation model.

3.2.2.7 – Regression Results

Model	R-Square (In Sample)	R-Square (Out of Sample - 4-fold X-validation)	Mean Square Error (In Sample)	Mean Square Error (Out of Sample - 4-fold X-validation)
Polynomial Features Degree 2 with Ridge Regression	0.432	0.398	0.009	0.009
Multiple Linear Regression with Ridge Regression	0.415	0.380	0.009	0.009
Multiple Linear Regression	0.418	0.368	0.009	0.009
Simple Regression (Theater)	0.255	0.235	0.011	0.011
Simple Regression (Restaurant)	0.221	0.188	0.012	0.012
Simple Regression (Pub)	0.234	0.186	0.012	0.012
Simple Regression (Coffee Shop)	0.195	0.172	0.012	0.012
Simple Regression (Movie Theater)	0.152	0.118	0.013	0.013
Polynomial Features Degree 2	0.531	-0.026	0.007	0.016

Table 7 – Regression Accuracy for Various Models. In-sample refers to the full dataset being used as both the training and testing set. Out-of-sample refers to the dataset being split into a training and testing set which are fractions of the full dataset. See Appendix I for more details.

The Simple Linear Regression Models are the worst performing overall, and performance generally decreases with decreasing correlation of the individual feature to population. The best regression model based on out-of-sample accuracy is the Ridge Regression tuned Linear Regression with polynomial features model, which has an out-of-sample R-Square score of ~0.398 and in-sample R-Square of 0.432. Although the regular model has the highest in-sample accuracy, it is overfit and has the worst out-of-sample accuracy. The Ridge Regression tuned model has 66 coefficients which makes it a more complicated model than the Multiple Linear Regression model which only has 11 coefficients. The Multiple Linear Regression without Ridge Regression model performs only slightly worse than the 2nd Degree Polynomial Features with Ridge Regression and Multiple Linear Regression with Ridge Regression model with an R-Square of ~0.368. However, distribution plots show that the Multiple Linear Regression distribution fits the distribution of the actual values most closely.

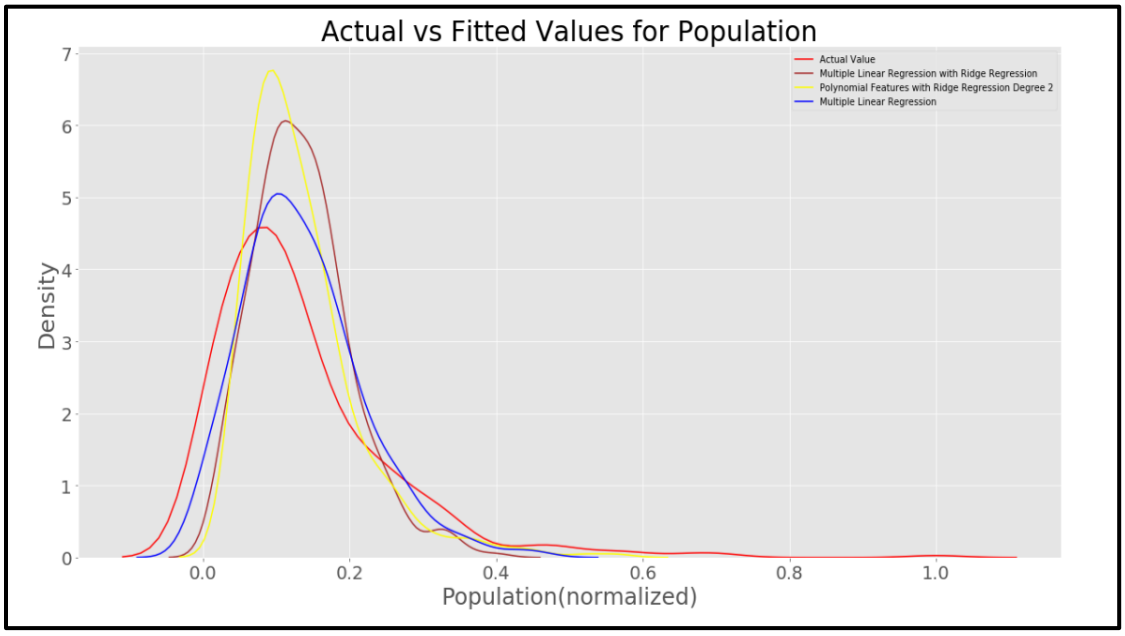

Figure 16 – Distribution Plots showing distribution of the predictions of best linear regression models vs. the true values

As the original MLR model is only marginally less accurate than the Ridge Regression tuned MLR and the Ridge Regression tuned Polynomial Features Degree 2 model but better fits the distribution of the true population values, and is much simpler, this would be the preferred model for predicting town population based on venue category totals. The Multiple Linear Regression Model has the formula:

$$\begin{pmatrix} P_1 \\ \vdots \\ P_m \end{pmatrix} = \begin{pmatrix} X1_1 & X2_1 & X3_1 & X4_1 & X5_1 & X6_1 & X7_1 & X8_1 & X9_1 & X10_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X1_m & X2_m & X3_m & X4_m & X5_m & X6_m & X7_m & X8_m & X9_m & X10_m \end{pmatrix} \begin{pmatrix} 0.110 \\ 0.037 \\ 0.072 \\ 0.064 \\ 0.056 \\ 0.084 \\ 0.041 \\ 0.083 \\ 0.082 \\ 0.032 \end{pmatrix} - 0.007$$

Feature Matrix

Intercept

Coefficient Vector

- P is the normalized population
- X1 to X10 (see Table 5) are the normalized venue category totals of the 10 regression features (Table 3)
- m = Town ID. There are 424 towns in this dataset.

3.2.3 – Clustering

Rather than attempt a user-defined classification scheme according to metrics other than population bins, two machine learning clustering methods were used to define town clusters as classes using DataFrame II (venue category density data), namely, K-Means and Density Based Spatial Clustering of Applications with Noise (DBSCAN).

3.2.3.1 – K-Means

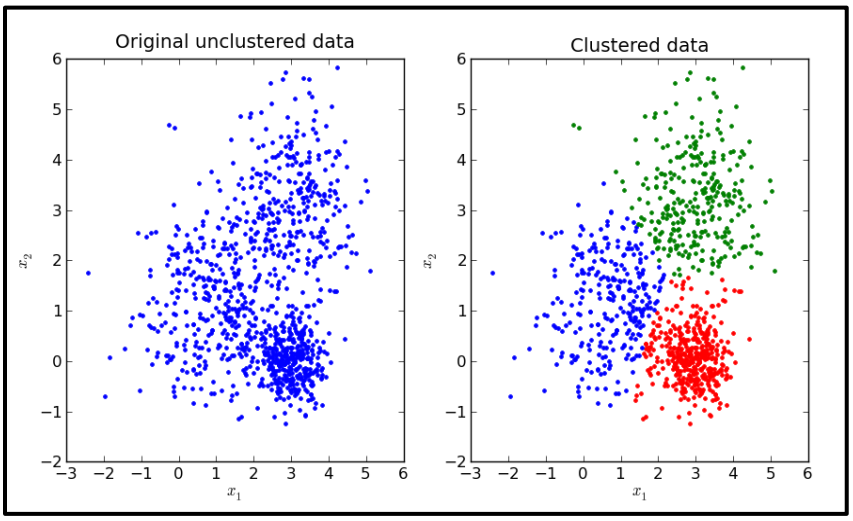

Figure 16 - K-Means Clustering. Source: https://i.stack.imgur.com/cIDB3.png

K-Means is a partition-based clustering method which is used to determine optimal clusters of data points based on the overall similarity of their features. The K-means algorithm minimizes the distance of points from their cluster centroid and maximizes the distance between the centroids. The number of clusters, K, is pre-selected, which places K centroids in the data points randomly whose positions are iteratively repositioned until they no longer move, and the clusters are most dense. At this point the algorithm has converged to its local optimum, based on the initial state. In order to achieve the global optimum, the algorithm is run multiple times with different initial states.

3.2.3.2 – Density Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN is a density-based clustering algorithm which clusters points based on density within a specified radius. The DBSCAN algorithm requires two parameters, the radius of the neighbourhood (or *epsilon*) and the minimum number of neighbours (M). The radius of the neighbourhood is the maximum distance between two samples for one to be considered in the neighbourhood of another (Scikit-Learn Developers, 2019). The minimum number of neighbors is the number of samples in a neighborhood needed to define a cluster. DBSCAN is particularly useful due to its ability to form arbitrary shaped clusters and to detect outliers/noise. It is usually effective for clustering spatial data, such as clustering data points on a map.

3.2.3.3 – Principal Component Analysis

Principal Component Analysis was performed on the feature matrix to project the feature matrix onto a lower dimensional feature matrix which preserves the statistical information of the original feature matrix. These principal components are uncorrelated features which maximize the variance and are functions of the original features (Wikipedia, 2020). The principal components were used to visualize the results of clustering in the 2-D or 3-D principal component space.

3.2.3.4 – Feature Agglomeration

For K-Means, clustering was based on the venue category density and population. For DBSCAN, latitude and longitude were added to the feature matrix to add geospatial variance since DBSCAN is normally effective for spatial data. To optimize the clustering process, the dimension of the feature matrix was reduced by merging similar features. The resulting feature matrix, in many cases contains new features which are the merged features combined into single values where the variance of the features that were merged is minimized. This helps eliminate noisy features which can compromise effective clustering.

3.2.3.5 – Selecting Model Parameters

The Silhouette Coefficient is a measure which describes the overall intra-cluster homogeneity and inter-cluster dissimilarity. Intra-cluster homogeneity is quantified by the Euclidian distance between the data points in the cluster and the cluster centroid, and inter-cluster dissimilarity by the Euclidian distance between a sample and the nearest cluster that sample doesn't belong to. The Silhouette Coefficient describes the relationship between these two. Optimal clusters have high intra-cluster homogeneity and are well separated, i.e. dissimilar from other clusters. The Silhouette Coefficient ranges from -1, where -1 signifies, the data point likely belongs to the wrong cluster, 0, that the clusters overlap and 1 that the clusters are homogenous and well separated (Scikit-Learn Developers, 2019). As there was no ground truth for class assignments, for each clustering method, the model hyperparameters and dimension of the feature matrix, were selected to generate clusters which made the most intuitive sense and maximized the Silhouette Coefficient. For example, two clusters, doesn't seem to intuitively, describe the range of variability of towns in the UK, despite generally high Silhouette Coefficients so parameters which specified or generated greater than two clusters were selected.

3.2.3.6 – Selecting Cluster Model

A range of model parameters were tested for each model and the resulting Silhouette Coefficient calculated, as shown in Appendix II. The K-Means Model parameters were selected to produce the highest possible Silhouette Coefficient given that there are more than 3 clusters, more than 5 features and more than 5 initializations to reach ensure the global optimum is reached.

K-MEANS MODEL PARAMETERS	
Number of Clusters	5
Number of Features	6
Number of Initializations	7
Algorithm	'Elkan'
Silhouette Coefficient	0.313

Table 8- Selected K-Means Model Parameters

DBSCAN parameters were selected to produce the highest possible Silhouette Coefficient given that the number of generated clusters, excluding the outliers is greater than 3 and the minimum number of samples is greater than 5. The six features in this model corresponded to the Latitude, Longitude, and four other merged features of statistical significance.

DBSCAN CLUSTERING MODEL PARAMETERS	
Epsilon	0.3
Minimum Samples	8
Number of Features	7
Silhouette Coefficient	0.344

Table 9 - Selected DBSCAN Clustering Model Parameters

3.2.3.7 – Clustering Results

3.2.3.7.1 – K-Means Clusters

The K-Means model generated 5 clusters of towns, labeled 0 to 4 as shown in Figures 17-18. For each K-Means run, the labels 0 to 4 are interchangeable but the makeup of the clusters is fixed (+/- 1 town depending on the initial state).

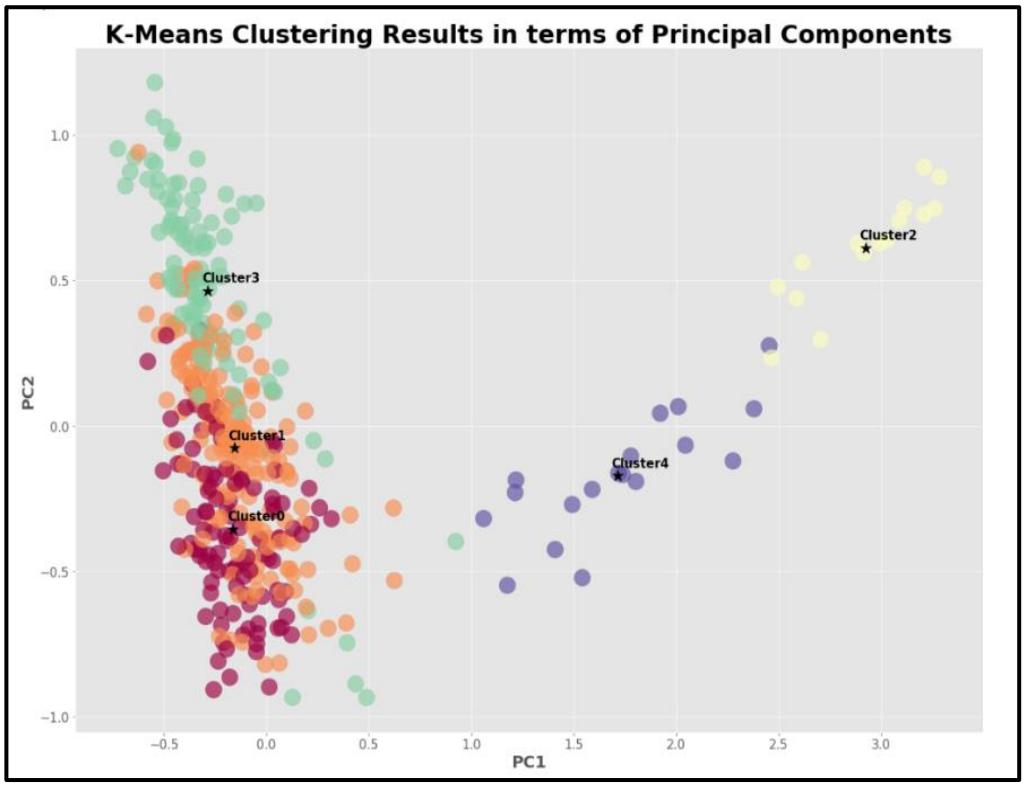

Figure 17 - K-Means Clusters in terms of Principal Components for UK towns based on relative number of venue categories density

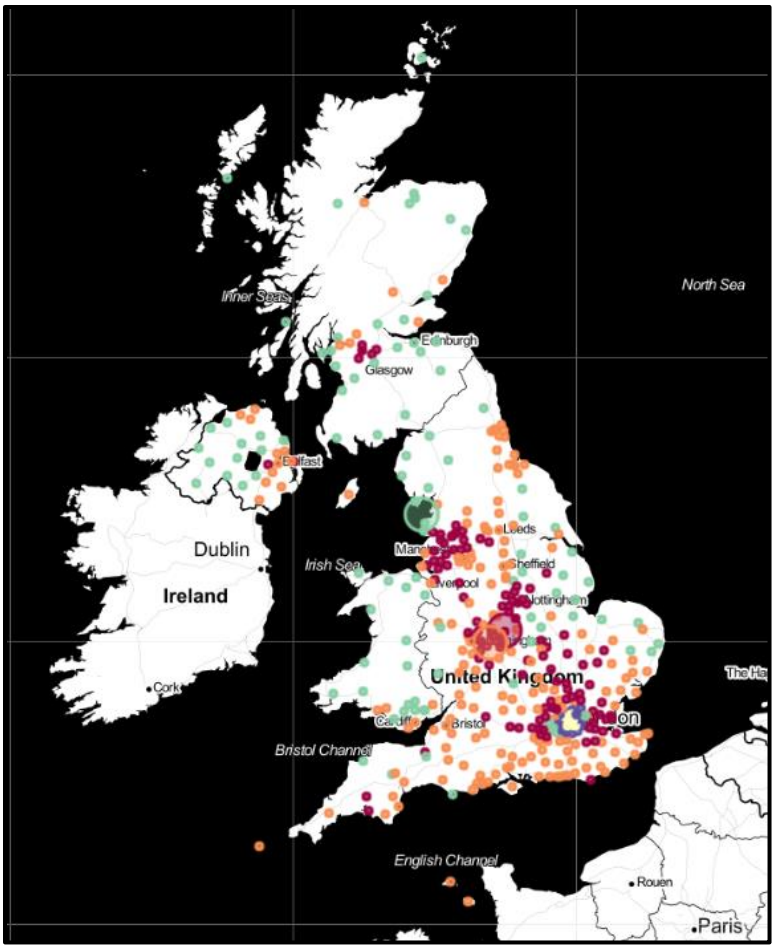

Figure 18 -Map of K-Means Clustered Towns in the UK: Red – Cluster 0, Orange – Cluster 1, Yellow – Cluster 2, Green – Cluster 3, Purple– Cluster 4. Larger circles indicate geographical cluster centers

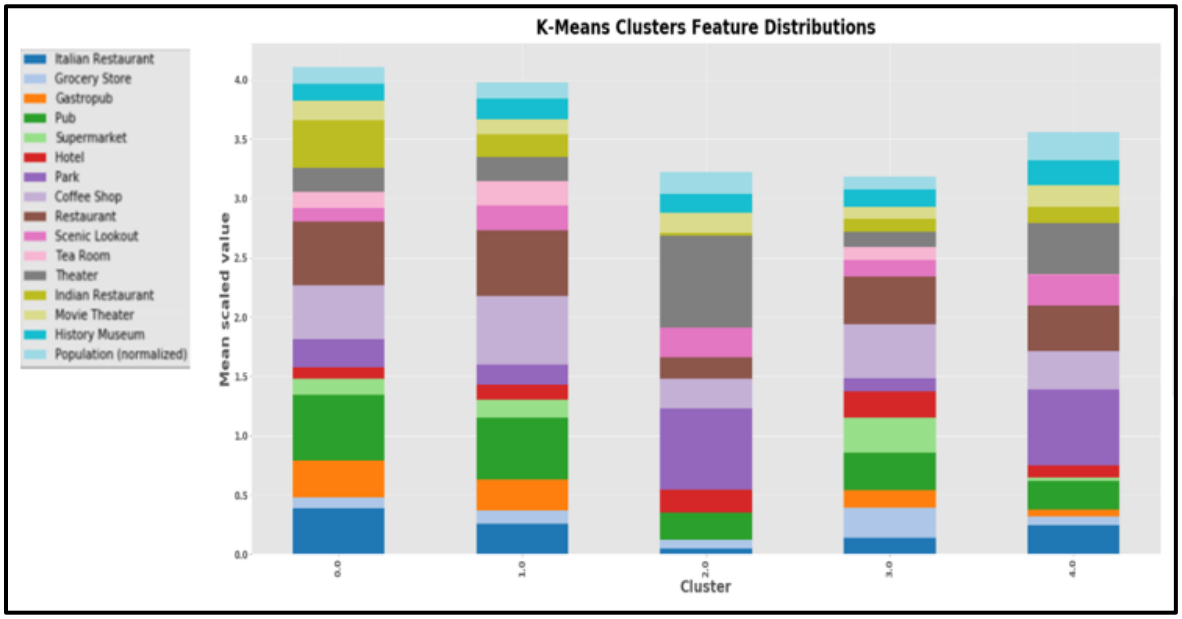

Figure 19 – Stacked chart of Average Features for K-Means Clusters in terms of normalized venue category density

Cluster 0

This cluster is comprised of 110(+/-1) towns located primarily in England and a small cluster of towns near Glasgow, Scotland. Some of the most populous towns in this Cluster include, Birmingham and Liverpool, which are two of the most populous towns in the UK, Nottingham, Wigan and Coventry. These towns have a high density of gastropubs, Indian Restaurants and Italian Restaurants.

Cluster 1

This cluster comprises of 180(+/-) towns, scattered throughout England, primarily middle to southern England, and a few located in eastern Northern Ireland. Some of the most populous towns in this cluster are Leicester, Leeds, Hampshire, Sheffield and Cornwall. These towns are distinguishable by the high density of coffee shops and tea rooms.

Cluster 2

This cluster contains 15(+/-1) towns in the urban areas of the London metropolitan area. Some of the most populous towns in this cluster are Camden, Hackney, Kensington & Chelsea and Hammersmith & Fulham. These towns are characterized by a high density of theaters, and parks.

Cluster 3

This cluster consists of 101(+/-1) towns, scattered throughout England and Scotland and western Northern Ireland. Some of the most populous towns in this cluster are Norfolk, Edinburgh, Exeter, York, Somerset and Aberdeen. Many coastal towns belong to this cluster. These towns are characterized by a high density of supermarkets, hotels and grocery stores.

Cluster 4

Cluster 4 consists of 18(+/-1) towns in the suburbs of the London metropolitan area. Some of the most populous town in this cluster are Croydon, Enfield, Greenwich and Hounslow. The main distinguishing characteristic of these towns is a very high average population.

Further Observations

Clusters 0,1 and 3 have a similar density of coffee shops and restaurants, which suggests there is little variation in the density of coffee shops, pubs restaurants per town across the majority of the UK. However, towns in Cluster 2, the urban areas of London have a much lower density of coffee shops and restaurants than in Clusters 0, 1 and 3, which is an indication of competition of many industries with the food and beverage hospitality industry in London

The average venue category densities for each cluster in terms of percentage of total venues can be seen in Appendix III.

3.2.3.7.2 – DBSCAN Clusters

The DBSCAN model generated 5 clusters (including the outlier cluster) labeled -1 through 4. As shown in Figures 20-21 two of the principal components appear to correspond to the latitude and longitude and the map in Figure 22, shows that they are generally clustered according to geographic location. However, one of the clusters has a collection of 29 towns with PC1 values greater than 1, which correspond to towns in the London metropolitan area, which are very different from towns everywhere else in the UK. DBSCAN is not able to separate these towns as an individual cluster based on the parameter requirements.

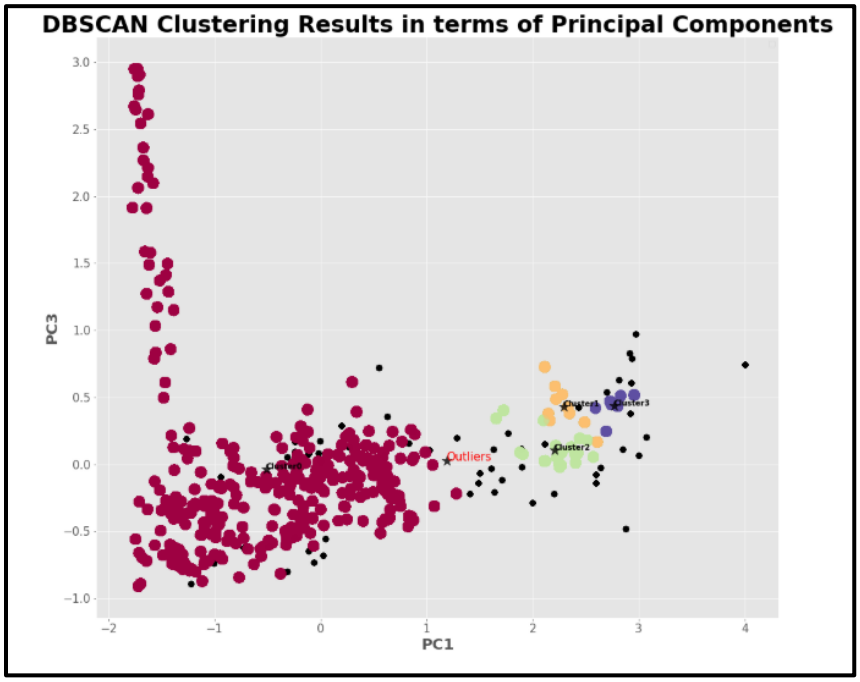

Figure 20 – DBSCAN Clusters in terms of Principal Components

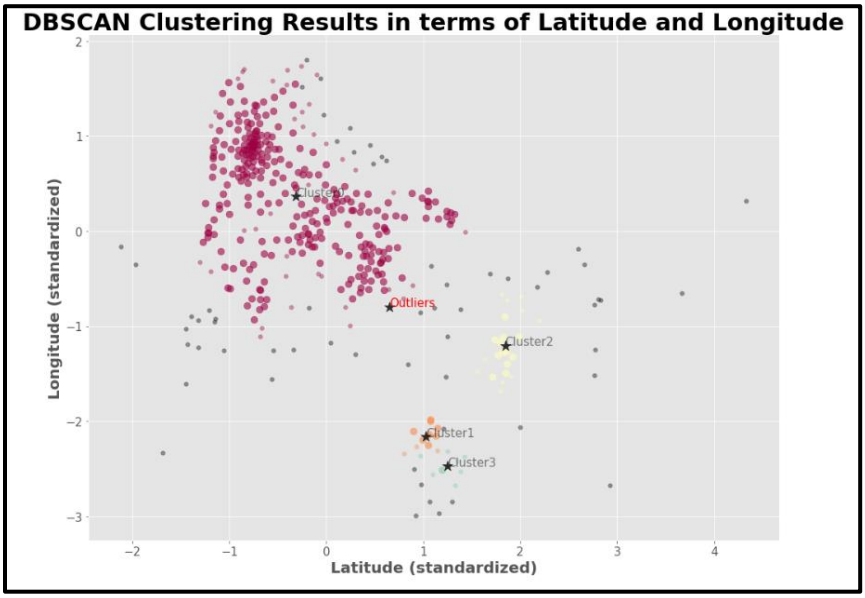

Figure 21 - DBSCAN Clusters in terms of Latitude and Longitude

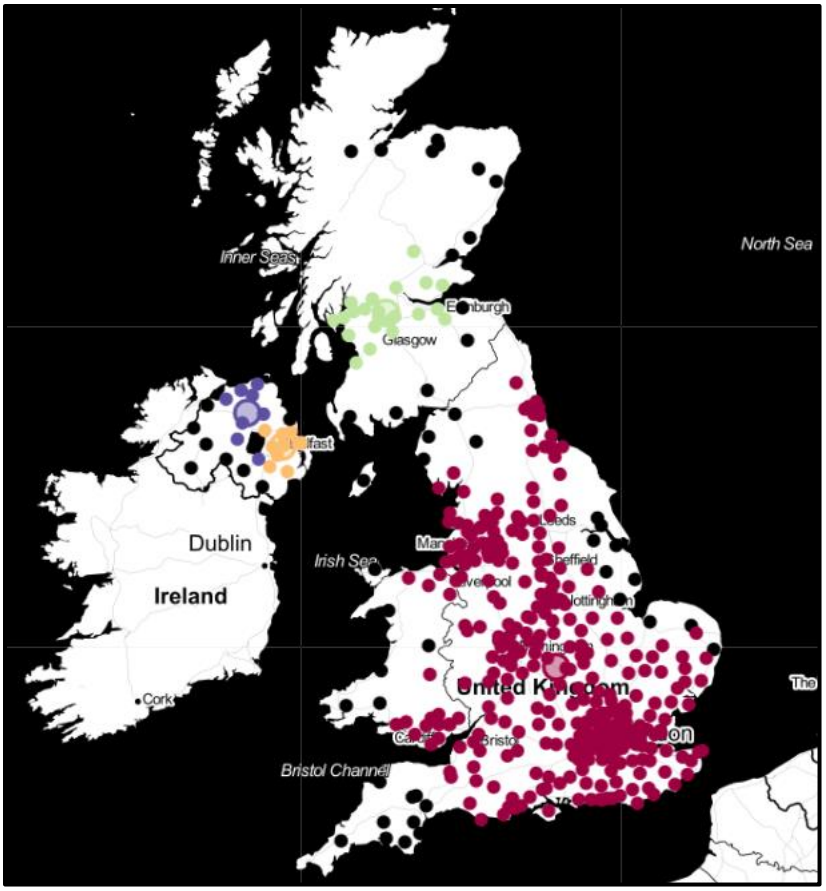

Figure 22 - Map of DBSCAN Clustered towns in the UK

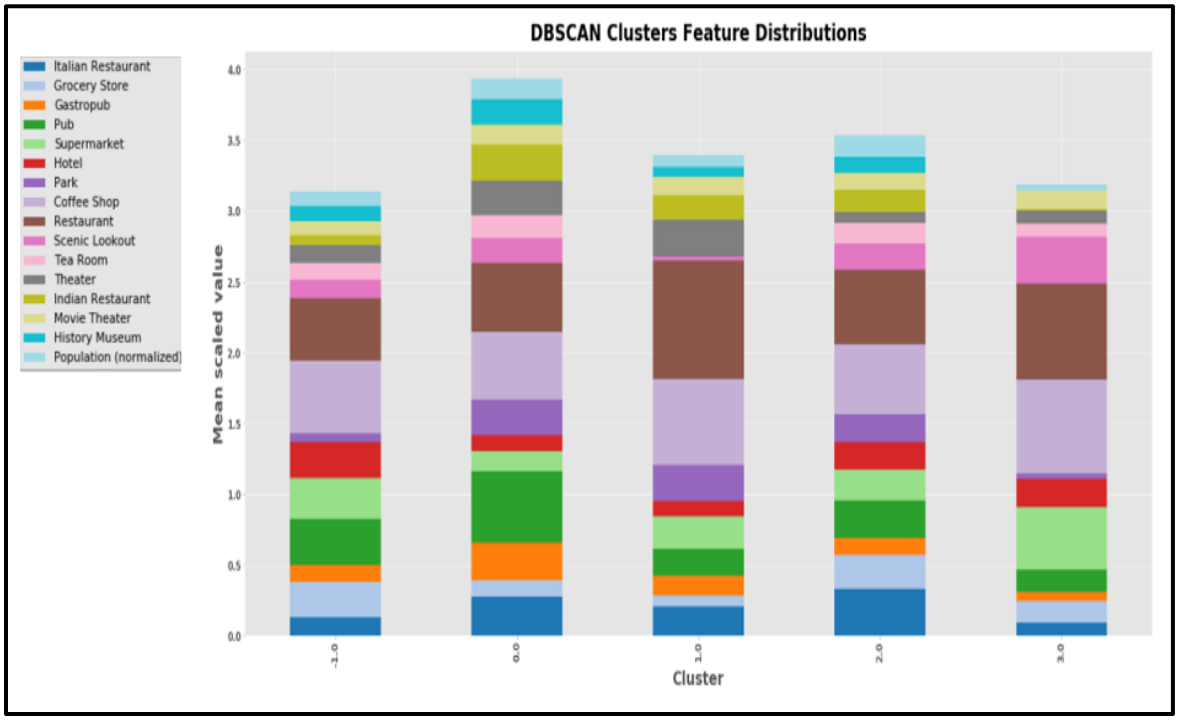

Figure 23 - Average Features for DBSCAN Clusters

Cluster -1

Cluster -1 is the outlier cluster. There are 60 outlier towns scattered around Scotland, England and Northern Ireland and particularly in coastal areas and away from the major towns. However, three major towns, Norfolk, in southeast England and Cornwall and Exeter, in Southwest England, which are in the top 30 towns in the dataset by population, and Aberdeen, which is a major town in Scotland belong to this cluster. These towns are do not appear to be characterized by any distinguishing feature.

Cluster 0

Cluster 0 is comprised of 324 towns which are scattered around England and eastern Wales. These towns have a high density of pubs, gastropubs, Indian Restaurants and History Museums.

Cluster 1

Cluster 1 comprises of 10 towns located in southeastern Northern Ireland. These towns have a very high restaurant density. Similar to towns in England, they have a high density of parks and theaters.

Cluster 2

Cluster 2 is comprised of 22 towns located in central Scotland. There isn't a feature which particularly differentiates these towns from other towns aside from location.

Cluster 3.

Cluster 3 is comprised of 8 towns in northeastern Northern Ireland. These towns have a higher density of supermarkets and scenic lookouts to towns in other clusters.

The average venue category densities for each cluster in terms of percentage of total venues can be seen in Appendix III.

3.2.4 – Classification

3.2.4.1 – Classification Feature Selection

The features selected for classification were those which were selected for regression, a few of the highest frequency features which aren't as highly correlated with the population, and the population. The following 16 features were selected:

Classification Feature Set
Theater
Pub
Restaurant
Coffee Shop
Museum
Park
Indian Restaurant
Movie Theater
History Museum
Grocery Store
Gastropub
Supermarket
Hotel
Scenic Lookout
Tea Room
Population

Table 10 – Classification Features

Classification was performed using DataFrame II for three different types of class labels:

1.) Population Bins as Class Labels
2.) K-Means Clusters as Class Labels
3.) DBSCAN Clusters as Class Labels.

For each of these scenarios, the following machine learning algorithms were used to classify the towns into predefined classes to see which one(s) would be effective:

I. Support Vector Machines (SVM)
II. K-Nearest Neighbor (KNN)
III. Decision Tree (DT)
IV. Random Forest (RF)

3.2.4.2 – Support Vector Machines

The SVM algorithm classifies data points by finding a separator between classes with maximum margin by *kernelling* - mapping the data into a higher dimensional feature space where the classes are separable. The two most popular kernels are the Linear kernel and the Radial Basis Function (RBF) kernel which are defined mathematically, where x is the Feature Matrix, as:

$$Linear\ Kernel: \langle x, x' \rangle^4$$

$$Radial\ Basis\ Function: \exp(-\gamma \parallel x - x' \parallel^2)$$

Each kernel requires a parameter, Cost (C), which describes the tradeoff between the prediction error and the simplicity of the decision surface (Scikit-Learn Developers , 2019). The RBF kernel also requires the parameter, gamma, which describes the influence of a single training example on class prediction (Scikit-Learn Developers , 2019). The RBF kernel is the de-facto kernel in many problems and can handle features which are non-linearly related to class labels but can be computationally expensive whereas a linear kernel is much faster and is preferred when there is a linear relationship between features and labels and in situations where the number of features is much higher than the number of observations (Hsu, 2003).

3.2.4.3 – K-Nearest Neighbor

The K-Nearest Neighbor algorithm classifies points based on their similarity to other points. A value of K is selected and the distance of an unclassified data point from all other data points in calculated. The algorithm searches for the K classified data points in the training set, which are nearest this data point. The class for this unclassified data point is then predicted, using the most popular class in the K-nearest neighbors to this point.

3.2.4.4 – Decision Tree

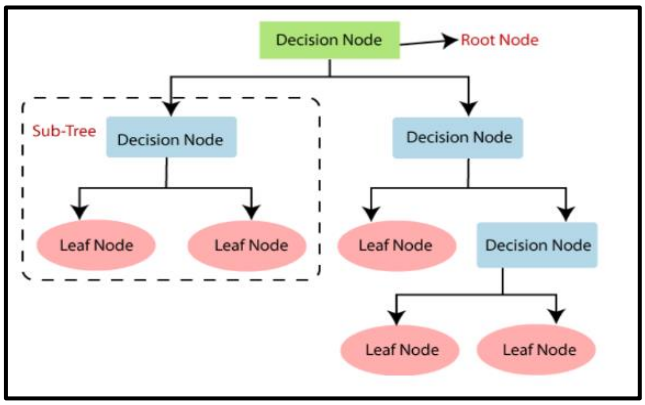

Figure 24 – Concept of a Decision Tree. Source: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

[4] This is also known as the Gramian Matrix, the inner product of the feature matrix

The Decision Tree algorithm (Figure 24) classifies data points by making decisions about which class a data point belongs to at each node in the decision tree. At each node, the dataset is split according to best feature and criterion which increases the purity of the samples in successive nodes. Each node should consist data points which primarily belong to one category of the data, and a completely pure node, consists of only that category of the data. The purity of nodes is quantified either by a value called entropy, which describes the randomness of the node, or gini impurity which describes the probability that an incorrect classification will be made. At each split the entropy/gini impurity decreases and should very close to zero or zero at the leaf node, at the bottom of the decision tree, where the majority of, or all samples belong to one class and a classification is made. A decision tree with good out-of-sample accuracy can then be used to predict unclassified data points.

3.2.4.5– Random Forest

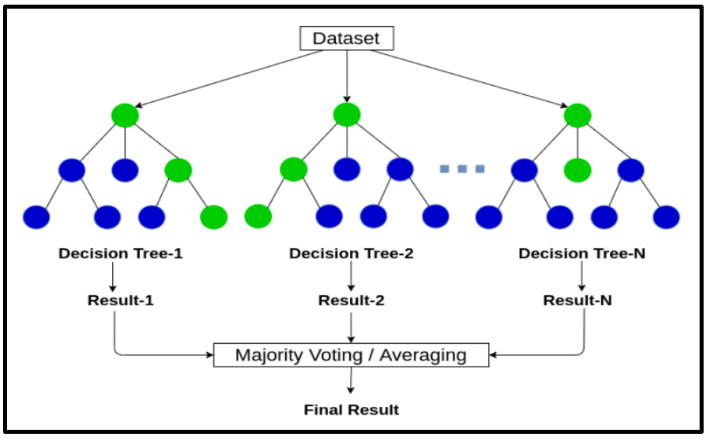

Figure 25 - Random Forest Diagram. Source: https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/

The Random Forest algorithm (Figure 25) builds multiple decision trees on random subsets of the data and averages the predictions for an unclassified sample to make accurate predictions. This decorrelated collection of decision trees creates a "random forest". The most common prediction for a sample from this collection of decision trees, becomes the prediction for the sample. Random Forests help prevent overfitting and are generally more accurate than a single decision tree.

3.2.4.6 – Learning Scenarios

The four ML algorithms were used on DataFrame II to predict class labels in five different scenarios:

A.) Model with maximum in-sample accuracy, trained on the full dataset, predicting on the full dataset.
B.) Model with maximum out-of-sample accuracy for a specified train/test split[5], trained on the training set, predicting on the training set.
C.) Model with maximum out-of-sample accuracy for a specified train/test split, predicting on the test set.
D.) Ideal model for maximum out-of-sample accuracy on the dataset derived from Grid Search, performing 4-fold cross validation.
E.) Ideal model for maximum out-of-derived from Grid Search, predicting on the full dataset.

[5] The train/test split was chosen by using the ideal model of each algorithm to predict on a range of test sizes from 10% to 70%. The test size which generates the highest recall for each model was averaged, to determine a train/test split which is suitable for all algorithms.

3.2.4.7 – Classification Evaluation

Classification was evaluated using the following metrics

I. Accuracy Score (Jaccard Similarity Score) – Fraction of correctly predicted classes to the union of the true classes and predicted classes.
II. Precision – True Positives/ (True Positives + False Positives)
III. Recall - True Positives/ (True Positives + False Negatives)
IV. F1 Score – (2 x (Precision x Recall))/ (Precision + Recall)

For each scenario, a confusion matrix which shows the recall for each class, was plotted, to visualize the accuracy of the predictions.

3.2.4.8 – Classification Results

Table 11 shows the results of classification for each of the models and learning scenarios. For each model, the weighted average Accuracy Score for all classes is shown. For more details on each model see Appendix IV.

Class Type	Train/Test Split	Model	Data Model is Predicting on	KNN Accuracy Score	SVM Accuracy Score	DT Accuracy Score	RF Accuracy Score
Population Bins	No split	Model with maximum in-sample accuracy	Full Data	1.000	0.991	1.000	0.998
	Train: 86% Test 14%	Model with maximum out-of-sample accuracy for split	Training Data	1.000	0.989	0.992	0.978
		Model with maximum out-of-sample accuracy for split	Testing Data	0.767	0.967	1.000	0.917
	4-Fold Cross Validation	Ideal Model from Grid Search	Cross Validation Test Data	0.795	0.930	0.988	0.949
	No split	Ideal Model from Grid Search	Full Data	1.000	0.981	1.000	1.000
K-Means Classes	No split	Model with maximum in-sample accuracy	Full Data	1.000	0.962	0.998	0.991
	Train: 75% Test 25%	Model with maximum out-of-sample accuracy for split	Training Data	1.000	0.906	0.821	0.994
		Model with maximum out-of-sample accuracy for split	Testing Data	0.802	0.877	0.651	0.783
	4-Fold Cross Validation	Ideal Model from Grid Search	Cross Validation Test Data	0.799	0.870	0.625	0.724
	No split	Ideal Model from Grid Search	Full Data	1.000	0.955	1.000	0.983
DBSCAN Classes	No split	Model with maximum in-sample accuracy	Full Data	1.000	0.974	1.000	1.000
	Train: 84% Test 16%	Model with maximum out-of-sample accuracy for split	Training Data	1.000	0.921	1.000	0.992
		Model with maximum out-of-sample accuracy for split	Testing Data	0.941	0.912	0.853	0.912
	4-Fold Cross Validation	Ideal Model from Grid Search	Cross Validation Test Data	0.889	0.882	0.828	0.865
	No split	Ideal Model from Grid Search	Full Data	1.000	0.939	1.000	1.000

Table 11 – Classification Results

3.2.4.8.1 - Population Bins

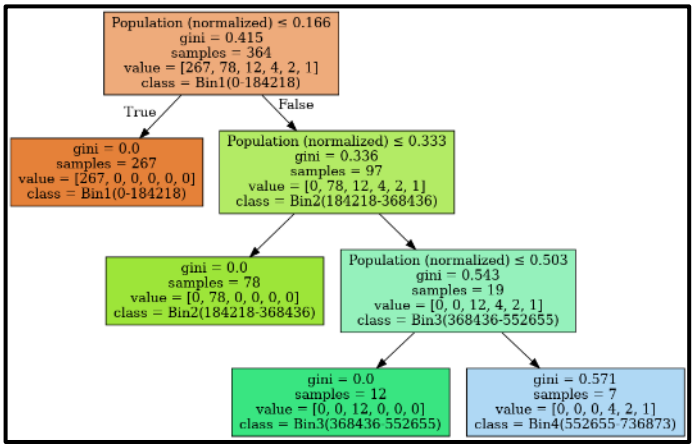

Figure 26 – Training Decision Tree Model (86% of classification feature set and corresponding population bins)

It is evident from (Table 13) that the Class Type which results in the highest overall accuracy is the Population Bins. Figure 26 shows the Decision Tree Model trained on 86% of the data for Population Bins classes.

This illustrates that the decision-making process is simple, as the only splitting feature is the Population. The decision criterion is also quite accurate. Bin 1 has a range of 0 – 184218. At the root node, towns with a normalized population of less than 0.166 are assigned to Bin 1. The maximum population is 1,105,310, in Birmingham, and 0.166 corresponds to a population of 0.166 * 1,105,310 = 183481.46 and so it follows for the other nodes (<0.333[368068]; <0.503[555970]). All the leaf nodes except one are pure. If the test set contains any towns which belong to Bin 5 or 6, prediction will fail because any town with a normalized population >0.503 is assigned to Bin4.

On the other hand, the ideal model derived from the Grid Search, shows that the main splitting feature is the Population but for Bins 5 and 6, which have two and one sample respectively, the splitting feature is not Population. It appears the very low sample size in those bins with the highest population, compromises the algorithm's ability to make an intelligent classification although the model still achieves high out-of-sample accuracy.

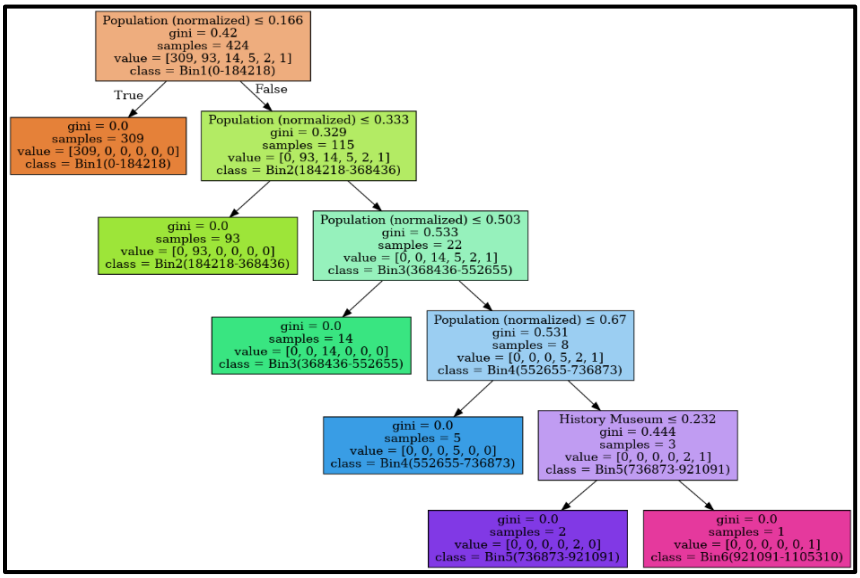

Figure 27 - Grid Search Derived Ideal Decision Tree Model

For SVM and RF, there is also high accuracy, due to the fact that the Population is clearly correlated with the Population Bin. Evidently there are no other features which have a strong enough contribution towards the population bin a town belongs to. DT, SVM and RF attempt to establish a relationship between the classes and the features, whereas KNN just relies on using classified samples in the neighbourhood (by Euclidian distance) of unclassified samples. Hence, KNN is the worst model for Population Bin classes.

3.2.4.8.2 - K-Means

For the classes derived from K-Means Clustering, the out-of-sample accuracy across the four models is roughly 77%. Figure 28 shows the Decision Tree Model trained on 75% of the data (318 towns) where the classes are the K-Means Clusters. Figure 29 shows the confusion matrix for this model.

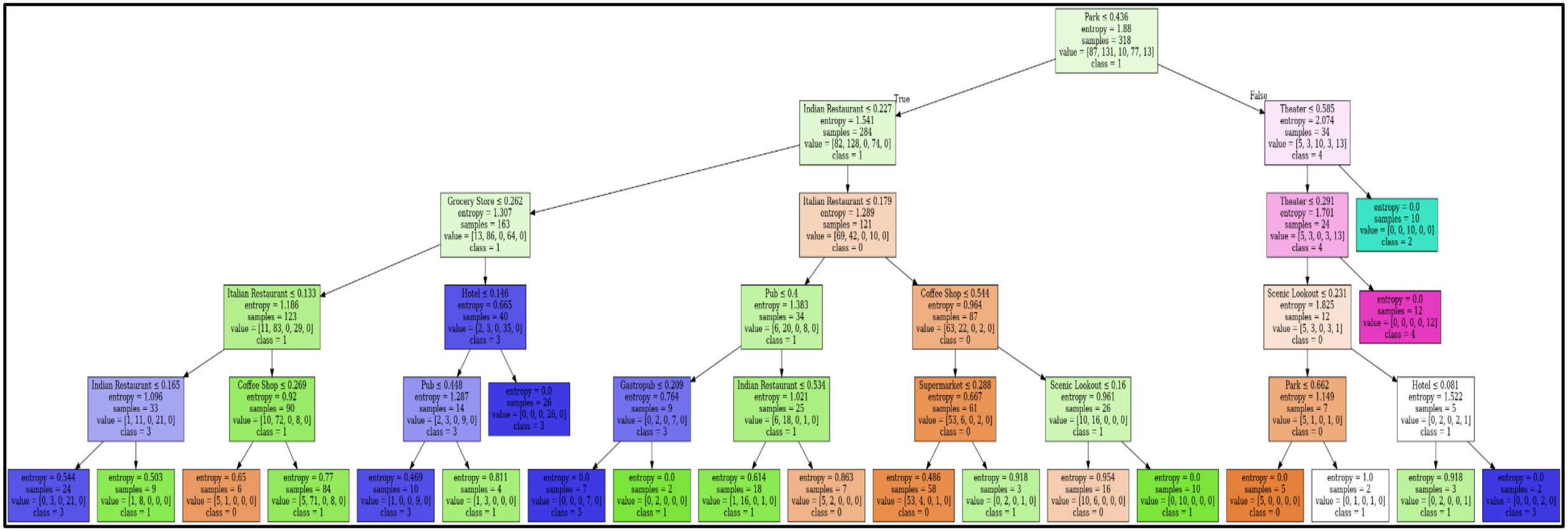

Figure 28 - Training Decision Tree Model for K-Means Cluster Labels (75% of classification feature set and corresponding K-Means Clusters)

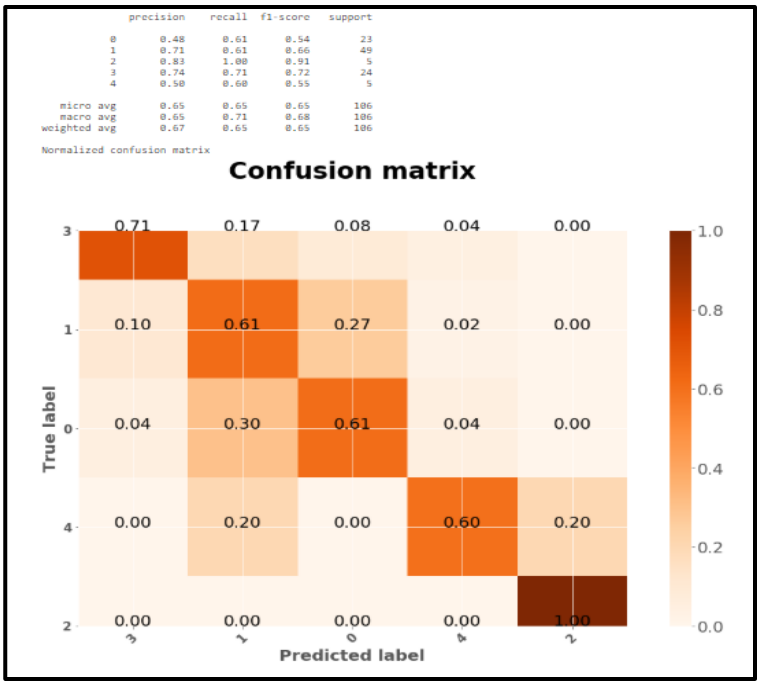

Figure 29 – Confusion Matrix for Decision Tree Model with K-Means Cluster Labels. Trained 75% of Classification Feature Set and tested on the remaining 25%.

This decision tree model is much more complex than the one with Population Bins as classes, as classification depends on many more features. The weighted average recall on the test set is about 65%. The decision tree indicates there is some relationship between the features used for classification and the class, but the relationship is not very clear. Furthermore, many of the leaf nodes are impure which would make prediction unreliable.

There isn't a fixed ideal decision tree generated from this classification feature matrix. Each time a Grid Search is run to generate the model, slightly different parameters are returned. The ideal model decision tree (DT_V1, DT_V2) is more complex than the training set decision tree. Different versions of the ideal model have some different nodes and splitting criteria. Examining the decision nodes of different runs of the ideal model the training set decision tree model (Figures 28) can highlight certain common splitting features and criteria such as Indian Restaurant or Park density at the root node but the general relationship between features and their class is difficult to observe.

The ideal RF, KNN and SVM models all perform better than DT. RF is a collection of decision trees, and out-of-sample accuracy goes up from ~63% to ~72% when an RF model is used. Since the relationships between the features and the K-Means class assignment is very complex, a singular decision tree is not reliable enough. The KNN ideal model, which involves 20 neighbours, has an out-of-sample accuracy of about ~80% which is even better than RF and the best model is the SVM with an out-of-sample accuracy of ~87% with a Cost of 62. Accurate classification into K-Means classes requires high computational cost and it appears the upper limit is less than 90%. SVM is the most powerful of the algorithms because it is able to create decision hyperplanes to separate the samples into classes based on the complex relationships between the features. The SVM model used an RBF kernel, which may mean the relationship between the features and the class labels is non-linear. The general relationship between the features of towns and the K-means class they belong to may be visualized in a Random Forest but for accuracy it is best to us SVM.

The K-Means model used to generate the classes, was generated using only six features as feature reduction to six merged features was one of the processes needed to generate the best clusters. Thus, the decision tree model, determines relationships between the features being used for classification and the classes regardless of the features which generated the clusters. This feature matrix is user-determined, and it may be possible to achieve similar or better levels of accuracy using other features or allowing the algorithm to determine splitting features from a feature matrix of all three hundred and fifty-one features.

3.2.4.8.3 - DBSCAN

For DBSCAN, the out-of-sample accuracy for the models is roughly 89%. Figure 30 shows the DT model trained on 84% of the dataset, where the classes are the DBSCAN cluster labels. This decision tree is slightly more complex than the training set decision tree for the K-Means Cluster Labels, but the leaf nodes are purer and accuracy on the test set is higher ~85%. The confusion matrix (Figure 31) shows that prediction is best for Cluster 0 (96% recall), which contains the majority of the towns, but for the other clusters it does not perform as well. For Clusters -1, 1, and 2, the recall is 55%, 0% and 67% respectively.

Different iterations of the ideal decision tree (DT_DBV1, DT_DBV2) exhibit much less variation in decision nodes and splitting criteria than those of K-Means and the structure remains relatively constant. There appears to be a distinct relationship between the features and the class predictions. The DBSCAN algorithm, incorporates geospatial variance, and as evidenced by the results of clustering (Figure 20-23), clusters appear to be heavily dependent on location even though five other features were used to generate the clusters. Thus, the results of the DT model may be establishing some relationship between the venue category densities of towns and whether they belong to one of the four defined geographical locations which could be useful insight. As with K-Means classes the best models are KNN and SVM followed by RF and DT. However, unlike with the K-Means classes, all the models have about the same accuracy and DT is only marginally worse than the other three. This may be due to the fact that the DBSCAN classes are better separated in principal component space than K-Means classes where three of the five classes overlap which makes classification easier for all the algorithms.

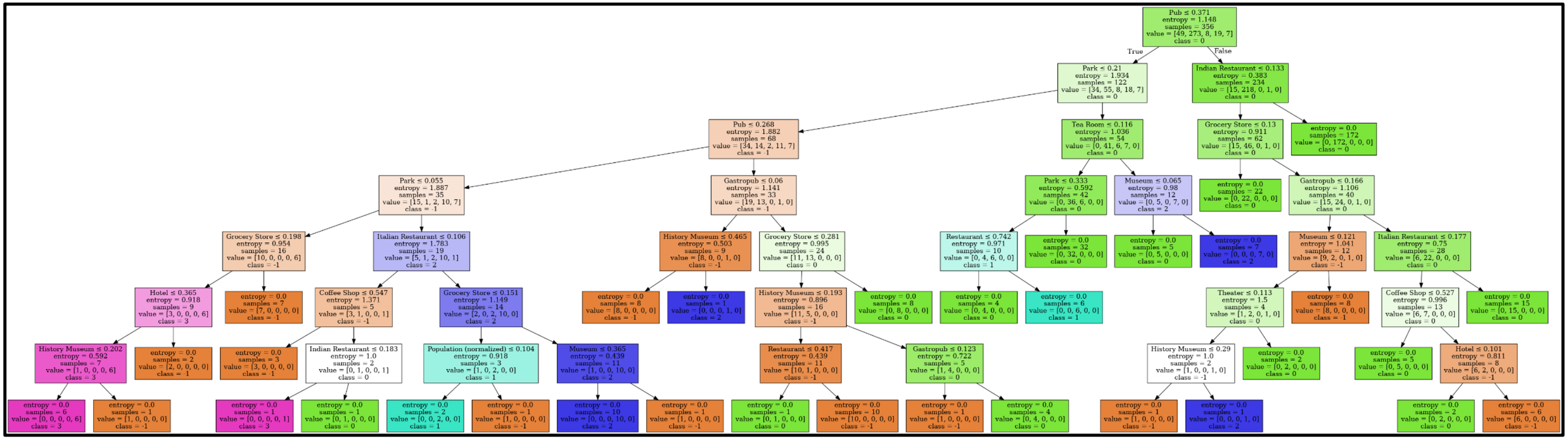

Figure 30 - Training Decision Tree Model with DBSCAN Cluster Labels as classes (84% of classification feature set and corresponding DBSCAN Clusters)

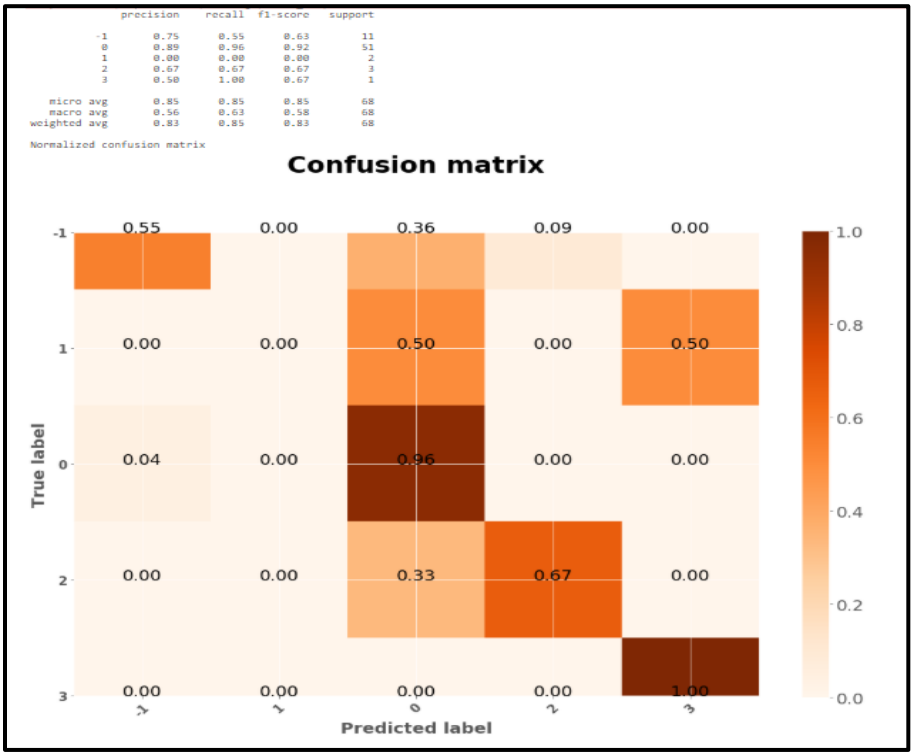

Figure 31 Confusion Matrix for Decision Tree Model with DBSCAN Cluster Labels as classes. Trained on 83% of Classification Feature Set and tested on the remaining 17%.

CONCLUSIONS & RECOMMENDATIONS

Regression techniques aimed to establish a relationship between the total number of certain venue categories per town and the population of the town. The best model had an R-square value of less than 0.5 which indicated that in the UK, there is no significant relationship between the number of certain venue categories in a town and its population. There is no particular correlation between the venue category densities and the population, so using venue category density would not improve results. Classification with Population Bins, as classes, also did not reveal any relationship between venue category densities in a town and its population bin except for Bins 5 and 6. This may indicate that, in the UK, population doesn't really drive business. Perhaps examining the relationship between venue categories totals and *population density*, or an economy related attribute would yield better results.

The clusters derived from K-Means represent one way of classifying towns in the UK. In deciding a town to settle in or establish a business, towns from another country, which fit the desired profile, could be classified using the SVM model and it would be assigned to one of these clusters. Any UK town in the cluster would likely share similarities with the foreign towns.

The accuracy scores of between 60-90%, demonstrate that machine learning algorithms are able to establish relationships between variables in the data when optimized for best performance. One future consideration would be to use towns from a totally different country for prediction. The towns and their assigned clusters could be displayed in a scatter plot in terms of their principal components, or the features for each cluster could be averaged. If the scatter plot or mean features for the predicted towns, were a reasonable match those of the training UK set, this would mean that the classification models, could be used to compare towns in the UK, with other towns around the world.

As would be expected, the London Metropolitan Area is very different from other regions in the UK. However, aside from London, the UK does not exhibit significant regional segmentation in the makeup of its towns. The DBSCAN algorithm generated clusters which were evidently separated predominantly according to the main geographical regions in the U.K but there aren't many major differences between the clusters. As shown in the DBSCAN PCA plot (Figure 20), the towns in the London cluster are the ones that exhibit considerable differences aside from geographical location. There isn't a combination of parameters for DBSCAN which results in multiple well-defined clusters in different geographical regions in the UK, with many towns in each cluster, and minimal outliers. The major clusters tend toward the three main geographic regions in the UK.

Incorporating latitude and longitude into K-Means clustering produces some finer geographical divisions in comparison to DBSCAN and well-defined large clusters in different geographical regions as shown in Figures 32-33.

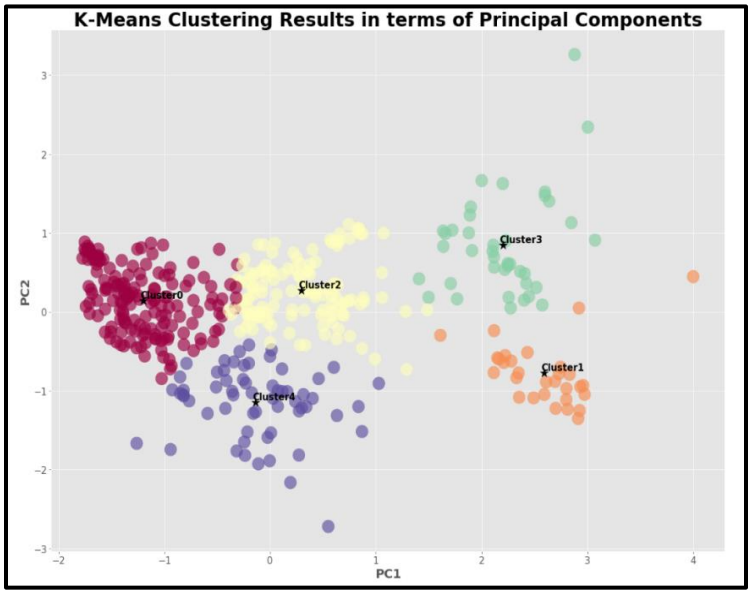

Figure 32 – K-Means Clusters with Latitude and Longitude added to Feature Matrix

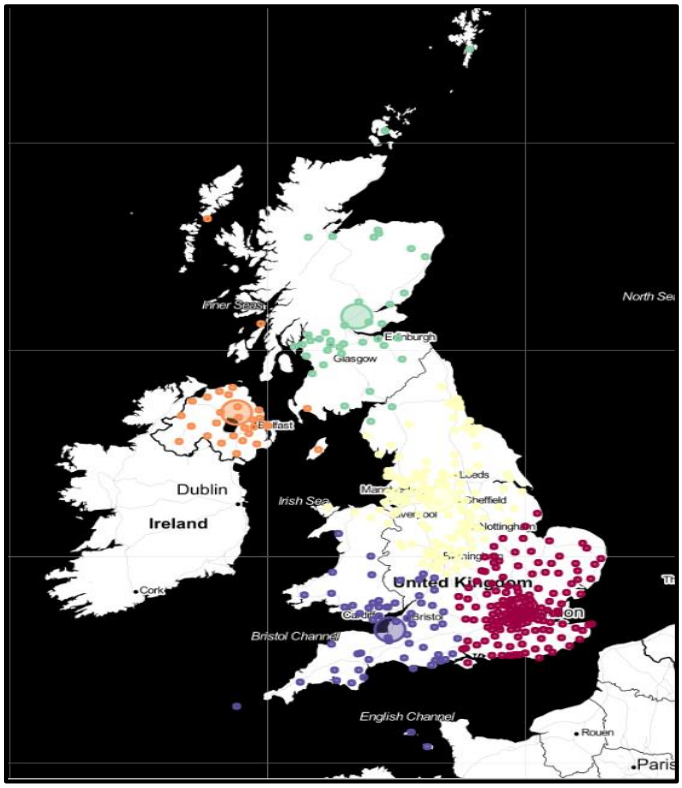

Figure 33 – Geographical Distribution of K-Means Clustered towns with Latitude and Longitude in Feature Matrix.

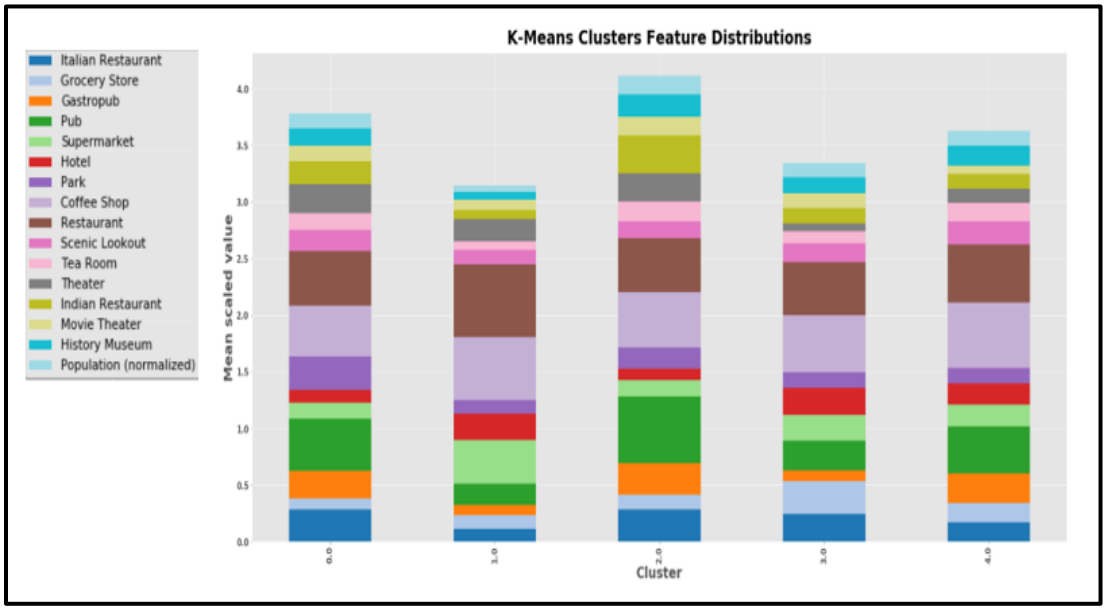

Figure 34 – K-Means Cluster Feature Distributions with Latitude and Longitude in Feature Matrix

There also appears to be enough differences between the clusters to suggest towns are segmented geographically according to their makeup (Figure 34). If so, classification could reveal what these differences are, and for a foreign town, be used to match it with similar towns in the UK based on geographical location.

These differences could also just be the average features, for each location defined by the K-Means Clusters. Perhaps, to really understand the regional variation of towns K-Means or DBSCAN could be used to generate geographical divisions of the UK using the latitude and longitude feature. For each geographical region, K-Means would be run again based solely on venue category density and a breakdown of types of towns within each geographical region of the UK would be achieved and potentially this would boost classification accuracy.

Overall, this project demonstrated the efficacy of using machine learning algorithms to gain a level of insight about towns on the U.K. There are many more insights which would be gleaned through further investigation and if tailored to the specific needs of the interested party.

APPENDIX I – REGRESSION MODELS

Model	R-Square (In Sample)	R-Square (Out of Sample - 4-fold X-validation)	Mean Square Error (In Sample)	Mean Square Error (Out of Sample - 4-fold X-validation)	Intercept	Coefficients
Polynomial Features with Ridge Regression Degree 2	0.432	0.398	0.009	0.009	0.041	66 coefficients shown below
Multiple Linear Regression with Ridge Regression	0.415	0.380	0.009	0.009	0.004	11 coefficients shown below
Multiple Linear Regression	0.418	0.368	0.009	0.009	-0.007	['0.24744']
Simple Regression (Theater)	0.255	0.235	0.011	0.011	0.061	['0.40753']
Simple Regression (Restaurant)	0.221	0.188	0.012	0.012	0.042	['0.36423']
Simple Regression (Pub)	0.234	0.186	0.012	0.012	0.049	['0.42058']
Simple Regression (Coffee Shop)	0.195	0.172	0.012	0.012	0.050	['0.25947']
Simple Regression (Movie Theater)	0.152	0.118	0.013	0.013	0.082	11 coefficients shown below

Polynomial Features Degree 2 with Ridge Regression

['0.0' '0.0439' '0.01589' '0.02316' '0.0174' '0.01876' '0.03855' '0.01482'

'0.02404' '0.02338' '0.01592' '0.03407' '0.01501' '0.00592' '0.00843'

'0.01282' '0.02434' '0.00877' '0.03067' '0.02259' '0.00829' '0.01481'

'0.0117' '0.01117' '0.01036' '0.00826' '0.01577' '0.01867' '0.00982'

'0.00927' '0.00819' '0.00765' '0.00898' '0.00582' '0.011' '0.01101'

'0.00664' '0.00783' '0.00713' '0.00744' '0.00475' '0.00916' '0.01072'

'0.0043' '0.00678' '0.00616' '0.00593' '0.0054' '0.02132' '0.00458'

'0.00716' '0.01267' '0.0097' '0.01235' '0.01295' '0.0069' '0.01832' '0.01376'

'0.01418' '0.01509' '0.02646' '0.01204' '0.00671' '0.01213' '0.0055' '0.01105']

Multiple Linear Regression with Ridge Regression

['0.10353' '0.04992' '0.05832' '0.05011' '0.05374' '0.07099' '0.04256'

'0.07025' '0.06115' '0.03775']

Multiple Linear Regression

['0.10974' '0.03688' '0.07237' '0.06393' '0.05559' '0.08412' '0.04077'

'0.08287' '0.08284' '0.03247']

APPENDIX II – SILHOUETTE COEFFICIENTS

K-Means Silhouette Coefficients

A total of 6187 Silhouette Coefficients were calculated for the number of clusters ranging from 2 to 14, number of features ranging from 1 to 14, the 'elkan' and 'auto' algorithm, and number of initializations ranging from 1 to 14, These are the Top 30 Silhouette Coefficients where the number of clusters is greater than 3, number of features is greater than 5, and number of initializations is greater than 5. Initializations greater than 5, was selected so the global maximum is reached.

ID	Silhouette Coefficient	Number of Clusters	Number of Features	Algorithm	Initializations
1524	0.3126	5	6	elkan	7
1526	0.3126	5	6	elkan	8
1537	0.3126	5	6	auto	13
1523	0.3125	5	6	auto	6
1525	0.3122	5	6	auto	7
1527	0.3122	5	6	auto	8
1528	0.3122	5	6	elkan	9
1530	0.3122	5	6	elkan	10
1531	0.3122	5	6	auto	10
1532	0.3122	5	6	elkan	11
1533	0.3122	5	6	auto	11
1535	0.3122	5	6	auto	12
1536	0.3122	5	6	elkan	13
1538	0.3122	5	6	elkan	14
1539	0.3122	5	6	auto	14
1522	0.3121	5	6	elkan	6
1529	0.3121	5	6	auto	9
1534	0.3121	5	6	elkan	12
1133	0.3096	4	9	auto	7
1136	0.3096	4	9	elkan	9
1140	0.3096	4	9	elkan	11
1143	0.3096	4	9	auto	12
1144	0.3096	4	9	elkan	13
1146	0.3096	4	9	elkan	14
1147	0.3096	4	9	auto	14
1131	0.3095	4	9	auto	6
1145	0.3092	4	9	auto	13
2000	0.3066	6	6	elkan	7
2005	0.3066	6	6	auto	9
2007	0.3066	6	6	auto	10

DBSCAN Silhouette Coefficients

A total of 343 Silhouette Coefficients were calculated for epsilon ranging from 0.3 to 0.9, minimum number of samples ranging from 3 to 9, and number of features ranging from 3 to 9. The minimum number of samples cannot be too low, or no clusters will form, and it can't be too high or else all samples will fall into one cluster. These are the top 20 Silhouette Coefficients, where the minimum number of samples is greater than 5, and the number of clusters excluding the outliers is greater than 3.

ID	Silhouette Coefficient	Epsilon	Minimum number of samples	Number of Clusters excluding outliers	Percentage Outliers	Number of Features
39	0.3438	0.3	8	4	14.15	7
40	0.3431	0.3	8	4	14.15	8
21	0.3013	0.3	6	4	6.60	3
22	0.3006	0.3	6	4	6.60	4
35	0.2906	0.3	8	4	8.73	3
28	0.2902	0.3	7	4	7.78	3
36	0.2899	0.3	8	4	8.73	4
29	0.2895	0.3	7	4	7.78	4
23	0.2832	0.3	6	4	7.78	5
30	0.2787	0.3	7	4	8.25	5
37	0.2656	0.3	8	4	10.38	5
24	0.2627	0.3	6	4	8.02	6
31	0.2583	0.3	7	4	8.49	6
25	0.2440	0.3	6	4	8.73	7
26	0.2431	0.3	6	4	8.73	8
27	0.2027	0.3	6	5	11.56	9
48	0.1733	0.3	9	4	19.81	9
32	0.0385	0.3	7	4	11.08	7
33	0.0355	0.3	7	4	11.32	8
34	-0.0044	0.3	7	4	14.86	9

APPENDIX III – CLUSTER DATA

K-Means Cluster Labels	Italian Restaurant	Grocery Store	Gastropub	Pub	Supermarket	Hotel	Park	Coffee Shop	Restaurant	Scenic Lookout	Tea Room	Theater	Indian Restaurant	Movie Theater	History Museum	% of All Venues	Average Population
0	2.66	1.84	3.77	20.48	2.69	3.73	6.36	8.86	10.92	0.52	0.53	1.15	2.78	0.89	0.75	67.93	151631.71
1	1.74	2.37	3.18	19.09	3.15	4.75	4.59	11.24	11.30	0.94	0.83	1.11	1.37	0.64	0.94	67.23	148471.09
2	0.31	1.55	0.00	8.65	0.04	7.47	18.32	4.84	3.62	1.13	0.00	4.32	0.17	0.87	0.86	52.14	206419.13
3	0.94	5.41	1.74	11.73	5.90	8.55	3.04	8.85	8.18	0.63	0.44	0.70	0.75	0.56	0.78	58.20	117460.93
4	1.66	1.52	0.69	9.16	0.47	4.06	17.09	6.34	7.83	1.16	0.02	2.40	0.94	0.98	1.09	55.40	266077.94

DBScan Labels	Italian Restaurant	Grocery Store	Gastropub	Pub	Supermarket	Hotel	Park	Coffee Shop	Restaurant	Scenic Lookout	Tea Room	Theater	Indian Restaurant	Movie Theater	History Museum	% of All Venues	Average Population
-1	0.89	5.21	1.43	12.12	5.81	9.8	1.63	10.16	8.92	0.58	0.47	0.7	0.51	0.5	0.58	59.31	112900.6
0	1.89	2.34	3.19	18.84	2.8	4.48	6.56	9.36	9.9	0.81	0.64	1.34	1.79	0.75	0.96	65.65	158532.6
1	1.41	1.63	1.69	7.07	4.51	4.35	6.74	11.97	16.85	0.13	0	1.46	1.23	0.67	0.39	60.1	89985.8
2	2.25	5.11	1.39	9.95	4.31	7.62	5.25	9.66	10.63	0.84	0.59	0.43	1.09	0.64	0.6	60.36	169737.7
3	0.64	3.15	0.81	5.66	8.8	7.91	1.02	12.99	13.76	1.5	0.36	0.52	0.08	0.68	0	57.88	47606.4

APPENDIX IV – CLASSIFICATION MODEL PARAMETERS & RESULTS

Class Type	Train/Test Split	Model	Data Model is Predicting on	Model Parameters KNN	Model Parameters SVM	Model Parameters DT	Model Parameters RF	KNN Accuracy Score	SVM Accuracy Score	DT Accuracy Score	RF Accuracy Score
Population Bins	No split	Model with maximum in-sample accuracy	Full Data	Number of Neighbours: 1, Algorithm: 'Auto', Weights: 'Distance'	Kernel: Linear, Gamma: Scale, C >69	Purity Criterion: Max Depth: 5	Purity Criterion: Gini, Max Depth: 9, Max Features: 'Sqrt', Number of Estimators: 10	1.000	0.991	1.000	0.998
	Train: 86% Test 14%	Model with maximum out-of-sample accuracy for split	Training Data	Number of Neighbours: 4, Algorithm: 'Auto', Weights: 'Distance'	Kernel: Linear, Gamma: Scale, C = 60	Purity Criterion: Max Depth: 3	Purity Criterion: Gini, Max Depth: 9, Max Features: 'Sqrt', Number of Estimators: 1	1.000	0.989	0.992	0.978
		Model with maximum out-of-sample accuracy for split	Testing Data	Number of Neighbours: 4, Algorithm: 'Auto', Weights: 'Distance'	Kernel: Linear, Gamma: Scale, C = 60	Purity Criterion: Max Depth: 3	Purity Criterion: Gini, Max Depth: 9, Max Features: 'Sqrt', Number of Estimators: 1	0.767	0.967	1.000	0.917
	4-Fold Cross Validation	Ideal Model from Grid Search	Cross Validation Test Data	Number of Neighbours: 4, Algorithm: 'Auto', Weights: 'Distance'	Kernel: Linear, Gamma: Scale, C = 21	Purity Criterion: Gini, Max Depth: 28 (max depth fluctuates)	Purity Criterion: Gini, Max Depth: 9, Max Features: 'Sqrt', Number of Estimators: 11 (Max Depth fluctuates each grid search)	0.795	0.930	0.988	0.949
	No split	Ideal Model from Grid Search	Full Data	Number of Neighbours: 4, Algorithm: 'Auto', Weights: 'Distance'	Kernel: Linear, Gamma: Scale, C = 21	Purity Criterion: Gini, Max Depth: 28 (max depth fluctuates)	Purity Criterion: Gini, Max Depth: 9, Max Features: 'Sqrt', Number of Estimators: 11 (Max Depth fluctuates each grid search)	1.000	0.981	1.000	1.000
K-Means Classes	No split	Model with maximum in-sample accuracy	Full Data	Number of Neighbours: 1, Algorithm: 'Auto', Weights: 'Uniform'	Kernel: RBF, Gamma: Scale, C = 91	Purity Criterion: Entropy, Max Depth: 10	Purity Criterion: Entropy, Max Depth: 8, Max Features: 'Sqrt', Number of Estimators: 27	1.000	0.962	0.998	0.991
	Train: 75% Test 25%	Model with maximum out-of-sample accuracy for split	Training Data	Number of Neighbours: 7, Algorithm: 'Auto', Weights: 'Distance'	Kernel: RBF, Gamma: Scale, C = 8	Purity Criterion: Entropy, Max Depth: 5 (max depth fluctuates)	Purity Criterion: Entropy, Max Depth: 8, Max Features: 'Sqrt', Number of Estimators: 91	1.000	0.906	0.821	0.994
		Model with maximum out-of-sample accuracy for split	Testing Data	Number of Neighbours: 7, Algorithm: 'Auto', Weights: 'Distance'	Kernel: RBF, Gamma: Scale, C = 8	Purity Criterion: Entropy, Max Depth: 5 (max depth fluctuates)	Purity Criterion: Entropy, Max Depth: 8, Max Features: 'Sqrt', Number of Estimators: 91	0.802	0.877	0.651	0.783
	4-Fold Cross Validation	Ideal Model from Grid Search	Cross Validation Test Data	Number of Neighbours: 20, Algorithm: 'Auto', Weights: 'Distance'	Kernel: RBF, Gamma: Scale, C = 62	Purity Criterion: Entropy, Max Depth: 24 (These parameters fluctuate with each run)	Purity Criterion: Entropy, Max Depth: 8, Max Features: 'Sqrt', Number of Estimators: 9 (These parameters fluctuate for each Grid Search)	0.799	0.870	0.625	0.724
	No split	Ideal Model from Grid Search	Full Data	Number of Neighbours: 20, Algorithm: 'Auto', Weights: 'Distance'	Kernel: RBF, Gamma: Scale, C = 62	Purity Criterion: Entropy, Max Depth: 24 (These parameters fluctuate with each run)	Purity Criterion: Entropy, Max Depth: 8, Max Features: 'Sqrt', Number of Estimators: 9 (These parameters fluctuate for each Grid Search)	1.000	0.955	1.000	0.983
DBSCAN Classes	No split	Model with maximum in-sample accuracy	Full Data	Number of Neighbours: 1, Algorithm: 'Auto', Weights: 'Distance'	Kernel: RBF, Gamma: Scale, C =68	Purity Criterion: Entropy, Max Depth: 8	Purity Criterion: Entropy, Max Depth: 13, Max Features: None, Number of Estimators:36 (This fluctuates)	1.000	0.974	1.000	1.000
	Train: 84% Test 16%	Model with maximum out-of-sample accuracy for split	Training Data	Number of Neighbours: 9, Algorithm: 'Auto', Weights: 'Distance'	Kernel: RBF, Gamma: Scale, C = 92	Purity Criterion: Entropy, Max Depth: 12 (Max Depth fluctuates)	Purity Criterion: Entropy, Max Depth: 13, Max Features: None, Number of Estimators:8 (This fluctuates)	1.000	0.921	1.000	0.992
		Model with maximum out-of-sample accuracy for split	Testing Data	Number of Neighbours: 9, Algorithm: 'Auto', Weights: 'Distance'	Kernel: RBF, Gamma: Scale, C = 92	Purity Criterion: Entropy, Max Depth: 12 (Max Depth fluctuates)	Purity Criterion: Entropy, Max Depth: 13, Max Features: None, Number of Estimators:8 (This fluctuates)	0.941	0.912	0.853	0.912
	4-Fold Cross Validation	Ideal Model from Grid Search	Cross Validation Test Data	Number of Neighbours: 3, Algorithm: 'Auto', Weights: 'Distance'	Kernel: RBF, Gamma: Scale, C = 14	Purity Criterion: Entropy, Max Depth: 38 (max depth fluctuates with each run)	Purity Criterion: Entropy, Max Depth: 13, Max Features: None, Number of Estimators: 15 (These parameters fluctuate for each Grid Search)	0.889	0.882	0.828	0.865
	No split	Ideal Model from Grid Search	Full Data	Number of Neighbours: 3, Algorithm: 'Auto', Weights: 'Distance'	Kernel: RBF, Gamma: Scale, C = 14	Purity Criterion: Entropy, Max Depth: 38 (max depth fluctuates with each run)	Purity Criterion: Entropy, Max Depth: 13, Max Features: None, Number of Estimators: 15 (These parameters fluctuate for each Grid Search)	1.000	0.939	1.000	1.000

References

Bell, C. (2020). *UK Postcodes* . Retrieved from https://www.doogal.co.uk/UKPostcodes.php

Hsu, C.-w. &.-c.-J.-W.-C.-J. (2003). *A Practical Guide to Support Vector Classification* .

NCSS Statistical Software. (n.d.). Retrieved from https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Hierarchical_Clustering-Dendrograms.pdf

Scikit-Learn Developers . (2019). *Support Vector Machines*. Retrieved from https://scikit-learn.org/stable/modules/svm.html#svm-kernels

Scikit-Learn Developers. (2019). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html).

Scikit-Learn Developers. (2019). *sklearn.metrics.silhouette_score*. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

Scikit-Learn Developers. (2019). *Tuning the hyper-parameters of an estimator*. Retrieved from https://scikit-learn.org/stable/modules/grid_search.html

The Pandas Development Team. (2014). *Intro to data structures*. Retrieved from https://pandas.pydata.org/pandas-docs/stable/getting_started/dsintro.html

Wikipedia. (2020). *Categorical variable*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Categorical_variable

Wikipedia. (2020). *Principal component analysis*. Retrieved from https://en.wikipedia.org/wiki/Principal_component_analysis

World Population Review. (2020). *GDP Ranked by Country 2020*. Retrieved from World Population Review: https://worldpopulationreview.com/countries/countries-by-gdp

Worldometer. (2020). *U.K. Population (LIVE)*. Retrieved from Worldometer: https://www.worldometers.info/world-population/uk-population/