Data Science Project
Jonathan Rees
ST20166609

Clustering Local Authorities based on administrative data

Abstract

Cluster analysis via machine learning provides the opportunity for computer algorithms to discover patterns within data that were previously undetected by human analysis. This project uses several clustering algorithms in order to group UK Local Authorities together into clusters, which can then be used by the Office for National Statistics in their yearly population estimates.

The Office for National Statistics currently creates Local Authority groupings by geographical closeness of authorities, but this risks under or over-estimating certain key demographics in the UK's national population. Grouping LA's instead based on commonalities found in administrative data will allow for more accurate population estimates to be generated.

Data derived from administrative sources was used in assessing 4 different clustering algorithms. The data was standardised across each algorithm, with dimensionality reduction applied via Principal Component Analysis. The validity of these clusters was then assessed using the silhouette score metric, as well as via manual evaluation from subject experts at the ONS. Results from these two evaluation methods were then compared in order to find the most successful techniques.

Both K-means and Self Organising Map algorithms achieved similar silhouette scores, with Hierarchical clustering receiving lower scores and the Tensorflow K-means estimator proving to be too computationally expensive. The results of the project show that while each algorithm achieved similar groupings of Local Authorities, these groupings scored relatively low silhouette scores. Manual evaluation was similarly unsuccessful, with the subject experts being unable to identify any clear patterns in the clusters. When compared to current LA classification groups used by the ONS the generated clusters did not appear to correlate to any particular classifications. Despite this the project does provide some valuable insight in how the data influences the clustering algorithms, and the generated clusters will be stored for use in future projects of a similar nature. Recommendations for future work include

using data with higher dimensionality and a wider array of origin data sources, as well as investigating additional clustering algorithms.

Acknowledgements

I would like to thank Jo Maimaris for her assistance in the selection and continuation of this project.

I would also like to thank Steve Smallwood and Neil Park for their assistance in validating outputs and providing an expert viewpoint.

Table of Contents

Introduction

The most common way for governments across the world to understand their population distribution and makeup is the use of a Census survey to collect this information. A problem arises in that Census' are generally collected on a 10 year basis. This allows for significant time to pass before changes in a population can be noticed by the government. The solution that many governments are moving towards is producing population estimates in the years between Census'. These estimates are often generated from previous census', with rules applied to increase the population year by year in a manner stratified to account for different demographics. Many statistical organisations around the world are beginning to lessen their reliance on a decennial Census by moving towards more modern data collection techniques. Jon Bryant, of Stats New Zealand states that *"The conceptually simplest alternative ... is to take a single administrative data source, such as a list of people enrolled within the health system, and to adjust for known deficiencies" (Bryant J, 2015).* The papers from Stats New Zealand have proved instrumental in the ONS's direction towards admin-based population estimation. The types of administrative data used by the ONS includes individual-level data records from Doctor's surgeries, educational institutes, the NHS, and HMRC tax records. Essentially for this study Administrative Data includes any data source that is not an ONS survey.

The UK is also beginning to look at ways to produce population estimates using administrative data sources. These between-census estimates are known as Admin Based Population Estimates (ABPEs). As with all estimation practices the ABPE's have some noticeable flaws, notably trends in under or over coverage for certain demographics, with these coverage patterns often differing greatly between local authorities.

This project aims to use cluster analysis, an unsupervised machine learning technique, to group together English and Welsh Local Authorities into "Estimation Areas" (EAs). Estimation areas are used by the Office for National Statistics (ONS) to adjust their ABPE's to account for these under or over coverage errors in certain demographic groups.
The ONS currently produces EAs based on geographical closeness. The need for these new EA groupings arises when the current method results in Local Authorities (LAs) being grouped with statistically dissimilar LA's. For instance, this can result in the grouping of local authorities with very different populations in regards to age/sex distributions. This can result in key demographics being misrepresented in official population estimates. These new groupings would theoretically result in more demographically similar LA's being put together, so that ABPE adjustments would be more accurate.

This project instead seeks to utilise machine learning clustering methods to discover groups of similar LAs, by identifying patterns within the administrative data. The primary data features used in the cluster analysis will be different types of *churn.* Churn is described by Scanlon, Travers and Whitehead as *"...mobility of all types and measures the net effects of all types of move [on a given population]. Moreover, household moves are identified as 'churn' only if they involve moving across borough boundaries" (Scanlon k et al, 2010)*, and has been identified as a key factor in potential over-estimation of the UK population. As such clustering using this data will provide unique insight into how best to account for this over-estimation. Other features will also be included where possible, such as rates of communal establishments, and average ages of LA populations. These features are theoretically not as important as the churn features, but age of population is an important thing to consider when looking at population movement.

Throughout this project, different machine learning techniques will be used and validated. The project will visualise data outputs and appraise the pros and cons of each technique. The project aims to produce a set of Estimation Areas, clustered from the local authorities in England and Wales. Various methods will be used to validate the final clusters, including statistical analysis as well as manual validation by colleagues at the ONS with expert domain knowledge on UK Local Authorities. If successful, these clusters will potentially be used by the ONS to adjust their ABPE's and produce estimates that have a more accurate representation of the UK population on a Local Authority level.

Literature Review

Admin based population estimates

The UK's Office for National Statistics (ONS) has recently been using administrative data (i.e. GP, NHS, and Schools data) to create Admin Based Population Estimates. The aim of these estimates is to reduce reliance on the decennial census *(Blackwell L, 2020),* and to provide population estimates for non-Census years. The ONS has currently iterated the ABPEs up to version 3.0, with each version updating inclusion rules for the estimation process. At the time of writing ABPEs have been produced for the years of 2016-2020. Each year is based on the previous year's ABPE, with estimated changes applied for births, deaths, LA to LA moves (internal migration) as well as international migration.

The ABPEs exist currently as an iterative project that are being continually worked on and improved, and as such are not yet relied upon for official statistics in the same way that Census data would be. It is understood that the ABPE's have an inherent level of statistical uncertainty. In a 2021 publication, the ONS' Ann Blake states that *"Both ABPE versions showed coverage patterns with local authorities (LAs) that were relatively stable over time in comparison with the Mid Year Estimates. However, there were considerable differences in the coverage patterns across different LAs. Our analysis explores what might be contributing to the variation in coverage patterns, for example, student moves to and from university." (Blake A, 2021)*.

This indicates that certain LAs are harder to estimate than others. As such the ONS has worked to identify these LAs, as well as the key factors and population groups that cause them to be difficult to estimate. Blake goes on to identify population areas with high percentages of students and communal establishments (such as halls of residence or military bases) as key factors in this, as these kinds of residences result in a high rate of movement year on year and can be difficult to capture in admin data. One reason for this includes students regularly moving location and being less likely to register with local services such as GP surgeries. To account for these movement-related changes in coverage between LAs, the ABPEs use estimation areas (groupings of similar Local Authorities that are adjusted together to give a more accurate population estimate) based on geographical constraint. With these estimation areas observed, the population estimates are then adjusted on an age and sex basis to more accurately represent changes in the population, with each estimation area being adjusted differently.

The adjustments that are applied to the population estimates are derived from the ONS' Population coverage survey. This survey is sent out to the public as a way of assessing demographic distributions in the UK. The results of this survey then inform the ONS in its adjustment of the raw ABPE estimate counts. These adjustments are made at an estimation area level rather than LA level as to ensure that there is an adequate sample size. The estimates are then quality assured using demographic analysis, census information, and survey data. *(Abbot O, 2009)*.

While the current system of ABPEs appear to be working to an acceptable level of error, the grouping of LAs into estimation areas has been identified as an area for improvement. Whereas LAs are currently grouped together based solely on geographical constraint, this project aims to create new estimation areas, with LAs grouped together via unsupervised clustering with a machine learning algorithm. The hypothesis here is that local authorities that are geographically separate can often have more in common demographically than LAs that are very close to one another (particularly considering age and sex distributions). As such the adjustment process would be more accurate when applied to these newly grouped estimation areas due to them having more similar characteristics than those in the previous EAs.

The ONS identified Age, Sex and Over/Undercoverage log scaling factors (LSF) as good data features to interrogate for clustering local authorities (in separate projects). Another primary attribute that has been identified for grouping estimation areas is population churn. Population churn is defined as *"…the total number of people moving in and out of an area, divided by the size of the population in that same area. More to the point, it is the population turnover rate for a specific community" (Cashen M, 2004).* Certain groupings of LAs may have similar rates of in or out churn depending on things such as university attendance, populations of transient workers, or the existence of military bases in their boundaries. Previous ONS work has identified churn as a key indicator of over coverage, as well as being a determinant in the quality of administrative data. This effect on admin data quality can be attributed to people not quickly updating their administrative data (doctors surgeries etc) when moving in and out of different local authorities, resulting in a lag in movement when the data is viewed longitudinally.

Methods for population clustering

A number of options are available for cluster analysis. One of the most well known and utilised algorithms available is the k-means method. K-means is a centroid based clustering method that is widely used due to its efficiency and ease of interpretation. K-means is however very sensitive to outliers and "noisy" data, so this must be taken into account during any data cleaning and standardisation that takes place *(Gan G, Ng M, 2017).* K-means requires the user to define the number of clusters *k*, and as such relies on some domain knowledge to decide upon the optimal number of clusters, although methods for deducing the optimum *k* value do exist, such as the use of an "elbow plot". *(Sammouda, R. & El-Zaart, A, 2021).*

This project also aims to utilise some more complex clustering techniques, such as Artificial Neural Networks. In recent years ANN's have been used extensively to solve clustering problems. The self-organising map (SOM) introduced by professor Teuvo Kohonen is one notable ANN that uses a competitive learning approach rather than error correction as with many other approaches. This technique can be easily implemented using a python package, and will be one of the techniques appraised in this work. *(Kohonen T, 1998).*

The project will also investigate the potential uses of more complex ANN packages such as Google's Tensorflow.

Following on from the creation of new estimation area clusters, the question of validation arises. How do we know that the clusters created are of good quality, or that the LAs included in each cluster are in fact statistically similar. The problem that arises with clustering lies in its unsupervised nature. How do we validate the separation of points into groups, if we have no prior knowledge to base our validation on? Thankfully, methods exist to validate clustering models, mostly depending on the structure of the clusters. Halkidi, Batistakis and Vazirgiannis, in their 2001 paper outline 3 groups of approaches for cluster validation. *External* validation compares the cluster partitions to prior partitioning knowledge (not dissimilar to validation of supervised learning methods). *Internal* validation focusses on the structure of the clusters, such as data point density, centroid distance, cluster and compactness. Finally *relative* validation compares the model's performance with that of models produced from separate subsets of the data. *(Haldiki M et al, 2001).*

The simplest of these options for our purposes is *Internal* validation. Internal validation comprises a family of techniques *(Dalton L et al 2009)* based on the assumption that each point is close to members of its own cluster, and far from members of all other clusters. Some of these metrics include Dunn's indices, Figure of Merit, and the Silhouette Index. This project will implement several of these assessment metrics, along with the manual assessment of clusters from an ONS colleague with specialist domain knowledge to select a "final" set of EA clusters.

Methods

Data Normalisation

Data normalisation is highly important in unsupervised machine learning. When clustering points together, the closeness of datapoints to their cluster centroids is an important metric to for validating clusters. As such, when we have outliers present in our data that have comparatively large distances from other datapoints or centroids, the cluster validation metrics can give less acceptable results. The first step of normalisation is to remove these clear outliers. In the data used in this project, only one point makes itself apparent as a true outlier. The city of London has much greater ABPEv3 churn than any other local authority. As such the City of London has been stripped from the data for this project, and will have to be considered as it's own entity during any potential ABPE alterations later.

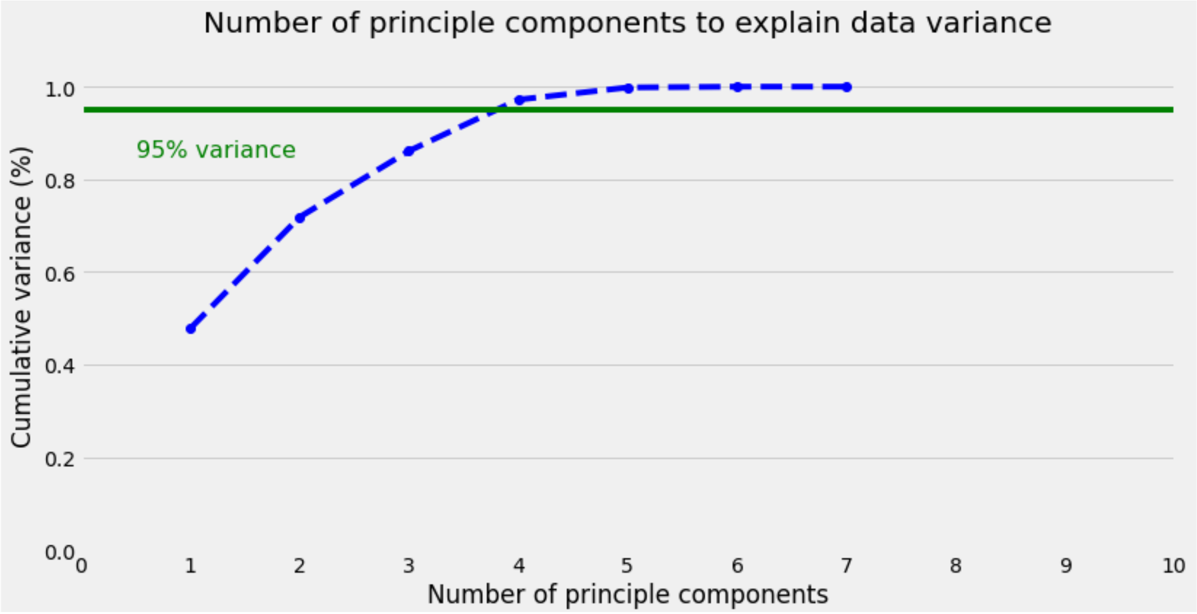
Following the removal of outliers, data standardisation methods were employed in order to bring the data down to a smaller range that is more appropriate for clustering. The StandardScaler function from SKLearn has been used to scale the data down by subtracting the sample mean and scaling to unit variance using the standard deviation. (Pedregosa *et al., 2011*). Theoretically this process will make it easier for the algorithms to group points together into clusters.

Following the data standardisation, Principal Component Analysis (PCA) has been used to reduce the number of features in the dataset. While our data only has 7 features, which is not a large amount for machine learning, some of the data features are derived from the same data and as such have a very high level of collinearity.

Daniel Larose describes PCA as seeking to *'explain the correlation structure of a set of predictor variables, using a smaller set of linear combinations of these variables. These linear combinations are called components, The total variability of a data set produced by the complete set of m variables can often be mostly accounted for by a smaller set of k linear combinations of these variables' (Larose, D 2015).*

PCA allows us to reduce our data into primary vectors known as 'eigenvectors' that carry the primary characteristics of the data. PCA itself is closely linked to clustering techniques, and as such requires a predefined value *x* to determine the number of principle components produced. As with clustering the ideal value of *x* must be identified and justified using statistical methods.

SKLearn's PCA feature has a function called .explain_variance_ratio(). This function gives us the amount of variance that each eigenvector is 'responsible' for. As a result, we can use the outputs of this function to see how many principal components are needed to account for 95% of our feature variance. The graph below shows that this value is reached at 4 principal component features. As such, we will then use the PCA method with an argument of n_components set to 4. This will transform our data into a 4-feature dataframe, which will hopefully cut out 'noise' from the dataset, as well as making our clusters substantially easier to visualise in the evaluation phase.



Algorithm Choices

With the data suitably pre-processed, the next step of the project is to outline and implement the different clustering algorithms to be used. The project will use a selection of four different approaches, with varying levels of complexity. The ONS's policies regarding statistical outputs puts a high value on the *explainability* of their reports. As such, they focus on using more simple algorithms such as k-means and hierarchical clustering. The benefits of this work now being undertaken as an external research project are that the project can consider the use of more technically advance algorithms, without undermining the need for the ONS to be able to explain their methods to stakeholders.

The first two clustering methods this project will consider are K-Means and Hierarchical clustering. These methods are both used within the ONS due to their easily explainable nature, ease of use, and low level of computational resource requirements. They can both be imported via the SKlearn package and used quickly with relatively few lines of code.

Following these the project will also implement a Self Organising Map (SOM) method, as well as a Tensorflow implementation of the K-means algorithm. The SOM algorithm operates in a similar fashion to neural networks, adjusting the values in a Numpy matrix iteratively until a pre-defined *n* number of iterations has passed.

Cluster validation

When undertaking unsupervised clustering, the first question to ask is always how many clusters to produce. The optimal number of clusters is generally down to the user to specify, but some techniques are available to give insight into what may be an optimal number of clusters. Many of the metrics provided by the SKlearn package require a 'truth' of sorts for the clusters to be compared against. For instance this could be a pre-defined class that gets removed from the data before clustering is undertaken (for example this is often done with a source such as the iris dataset for machine learning training purposes). Unfortunately our clustering is completely unsupervised, as we have no previously defined truth columns. As such we will use a combination of the elbow method and silhouette method to decide upon our number of clusters for each implementation.

The elbow method creates multiple cluster sets where number of clusters *k* ranges between two given numbers (for this project I have used the range of 2 to 15), and compares the average distortion of all clusters value of *k* (This representation of cluster number is most common with k-means, but I will use the descriptor of *k* for all algorithms in this project for consistency). This method is very popular with simple K-means implementations, but only focuses on the distortion metric (The sum squared error of each point from it's given cluster centroid), whereas the silhouette method compares a point's similarity to it's own cluster with it's dissimilarity from other clusters (Known respectively as cohesion and separation). Due to using two different metrics, the silhouette method is often considered the more comprehensive measure of clustering success, and will be the main measure that this report uses to judge cluster validation.

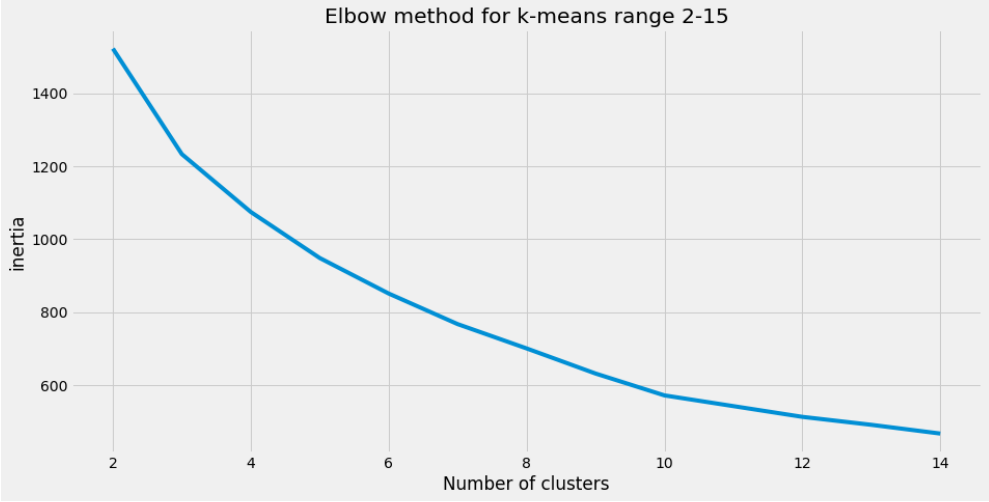
An unfortunate outcome when using these methods is that the clusters often show greater silhouette scores at very low cluster numbers, such as 2 and 3. For the purposes of population estimation this is not suitable, as we need to work at a much more granular level. A similar project was undertaken in 2011 using different population data. That previous project clustered LA's into 8 groups known as 'Supergroups' with distinct demographic characteristics. Considering this it, this project will discount a number of clusters lower than 8 in hopes of decreasing cluster size. Hopefully clusters of below 30 LA's would be much easier to adjust for as well as potentially having more specific commonalities between LA's that share each cluster.

<u>K-Means clustering</u>

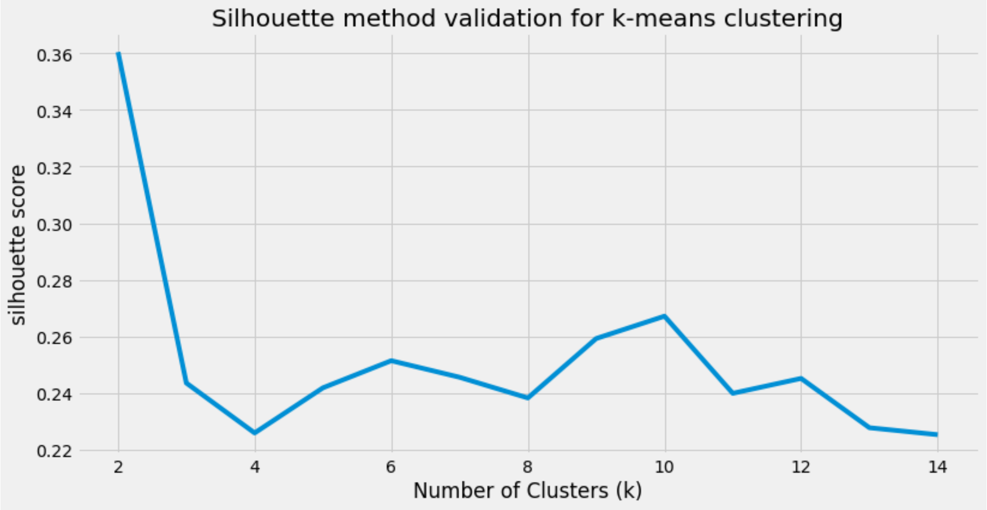
The first algorithm used is the K-means clustering algorithm. K-means is arguably the simplest and most well known clustering algorithm available to us. This ease of use along with its reliability in producing viable clusters has resulted in K-means being well used within the ONS. The aim of using K-means first is to potentially produce a cluster set that the ONS would be comfortable using in official population estimates, as well as having a 'benchmark' to compare our more complex methods to. This will help us understand whether utilising more complex methods of clustering warrants the increased difficulty, or whether using tried and tested methods gives us equally suitable cluster results.

The K-means algorithm works by partitioning the datapoints as to minimise the within-cluster sum of squares (WCSS), sometimes referred to as Distortion or Variance. The algorithm starts by assigning *k* random datapoints as cluster centroids, then assigns each datapoint to its closes centroid by some distance measure (typically Euclidean). The process is then repeated multiple times with the centroids gradually moved to the centre of their respective clusters. This then achieves the lowest possible WCSS for this number of *k (Zalik k, 2008).*

The Elbow method for this run of K-means implementation gives an elbow point at 10 clusters, with an inertia of 571.

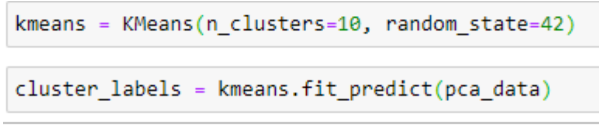


When performing the silhouette evaluation method, the output also identifies a peak in silhouette score at *k*=10 clusters with a score of 0.267. These two methods corroborating one another gives us a good indication that 10 clusters would provide us with datapoints that are reasonably well separated.



Generally speaking, silhouette coefficients over 0.5 show a well organised cluster structure, whereas values between 0 and 0.5 show loosely structured or overlapping clusters. For our clustering task, this is somewhat expected due to the nature of the data. The churn based variables are unlikely to be well segregated in their feature space, as our churn variables are rates that are impacted by any number of variables within an LA. Given this, a silhouette score of around 0.25 is acceptable. Following these test results I proceeded to run a final k-means instance and produce an output file with each LA assigned to one of 10 clusters. This will be manually evaluated at the end of the project along with outputs from each of the other algorithms, in order to decide on the best performing model.

The following code was used to produce the final set of clusters. The cluster index list can then be appended to a Pandas dataframe as a new column, then passed to a Plotly chart for final visualisation.



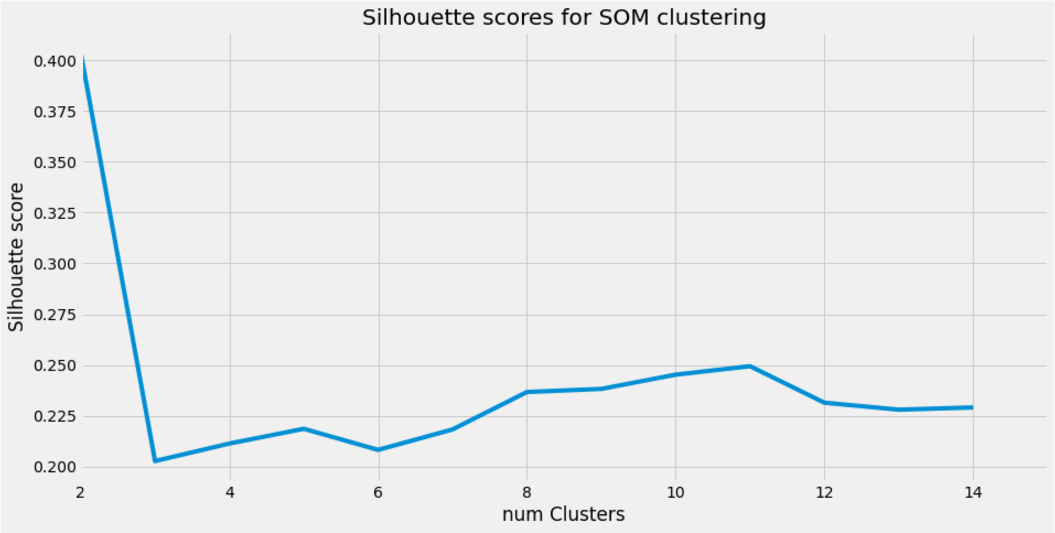
```
kmeans = KMeans(n_clusters=10, random_state=42)
```

```
cluster_labels = kmeans.fit_predict(pca_data)
```

Minisom implementation

The Self Organised Map (SOM) is a neural-network based clustering algorithm first implemented by Teuvo Kohonen in the 1980s. The SOM is an Artificial Neural Network (ANN) that operates slightly differently from the commonly used ANN methods such as gradient descent based error correction. The SOM operates with a competitive learning approach where the map's nodes are compared with the weight vectors, and moved depending on their similarity with the winning vector. This is repeated for *x* number of iterations, and when finished the map will have formed clusters, with each datapoint moved closer together and assigned index labels depending on their closeness to one another.

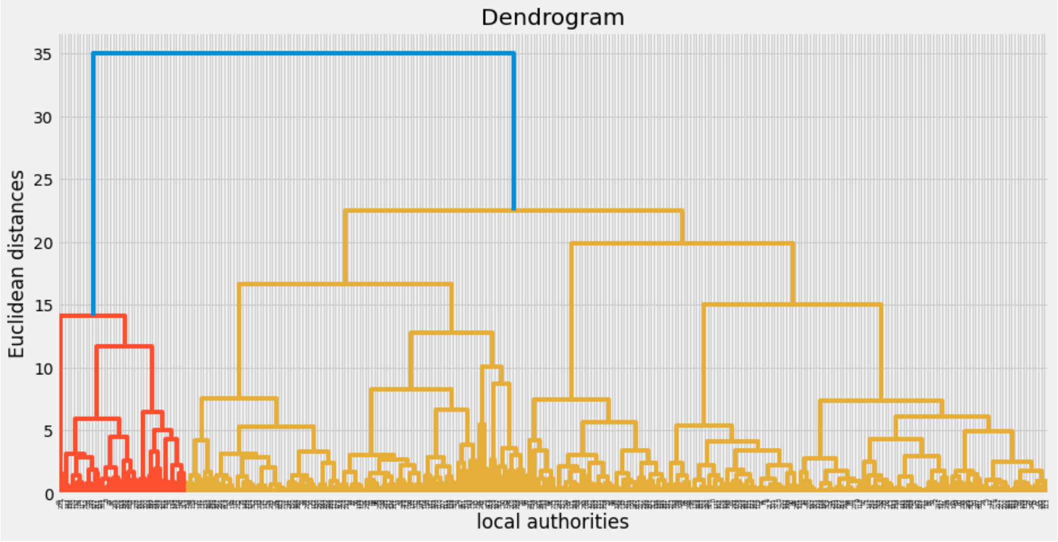
To achieve the silhouette scores for this method, I looped through the range of 2 to 15, as with the k-means approach, and calculated the score using sklearn.metrics for each loop. The silhouette score can be accessed by passing the cluster index labels along with the initial input data to the sklearn metrics package. The chart below outlines the silhouette scores for each number of *k* clusters. As we can see the scores reached a second peak of 0.25 at 11 clusters. This is in the same ballpark as our k-means result of 10, as well as a peak score that is very similar to the k-means implementation. This indicates that the clusters found are 'real' and not due to some coincidence with the algorithm. Following on from this I then used the code to produce outputs for 11 clusters, which will be evaluated manually at the end of the project.



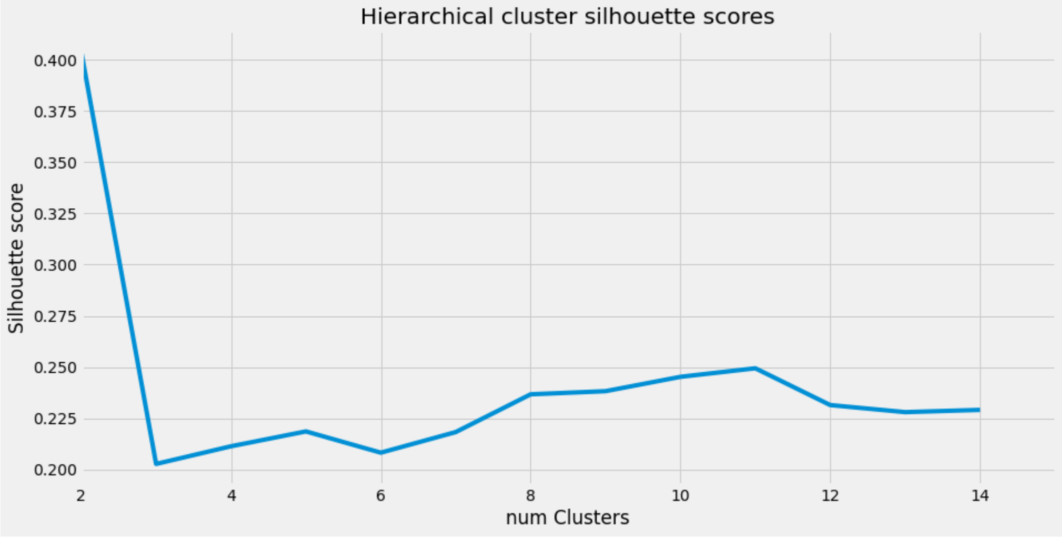
Hierarchical clustering implementation

Hierarchical clustering is another fairly straightforward technique, employed in via the SKlearn package, similarly to k-means. Hierarchical clustering operates by first treating each datapoint as it's own cluster, then iteratively bringing the closest two points together as a cluster, reducing cluster numbers but increasing points within those clusters, until the specified number of *k* is met.

One method for assessing the optimal number of clusters with this technique is called a *dendrogram*. Dendrograms display the joining of datapoints into clusters, as well as the distance between points when they are joined. Dendrograms do not give a perfect representation of the optimal cluster number, but can be interpreted by looking for a point where the vertical distance between cluster joins is greatest.



As we can see the dendrogram indicates that two clusters is ideal, but 2 clusters does not serve our purpose particularly well. We can however see that reasonable clusters are also present around the 7-14 range, with fairly long vertical distances between joins. Following this I will also use the previously mentioned silhouette metric to decide on the final number of clusters for this model.



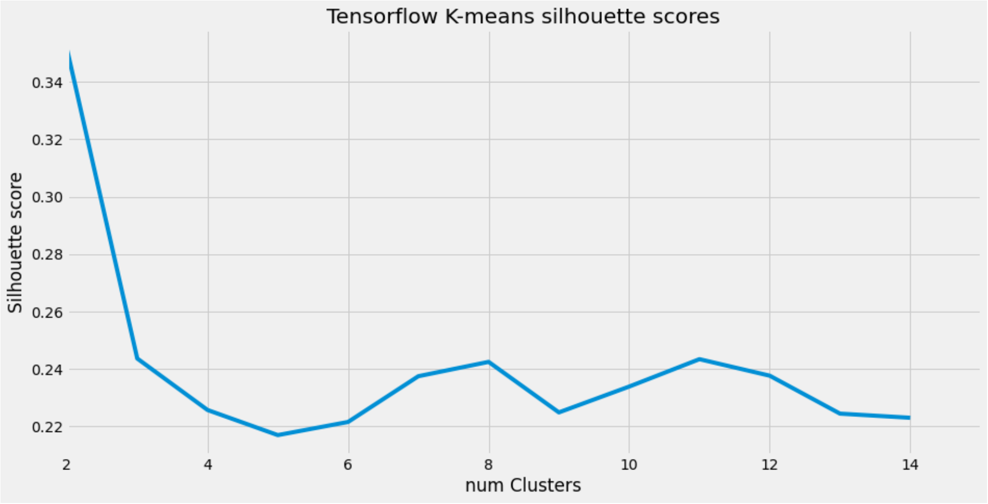
Again, we see a slight peak in score at 11 clusters (Similar to the number in the previous models) at 0.25. The fact that both the optimal number of *k* and the highest silhouette score are similar to those seen in previous clustering implementations provides more evidence that the clusters we are seeing are legitimate, and not a coincidental result. As with previous algorithms, manual validation of clusters will hopefully provide more insight about the common features of LA's in the same clusters. Clusters from different algorithms will also be compared to confirm or deny whether the clusters contain the same (or at least similar) local authorities.

Tensorflow K-means

The Tensorflow package is a machine learning platform developed by Google. It has a particular focus on the operation of Artificial Neural Networks. While Tensorflow mainly focuses on the operation of deep learning to carry out regression or classification tasks, it can also operate the K-means algorithm. The benefit that Tensorflow has over traditional k-means the ability to generate clusters over *n* 'epochs'. The hope with using this version of k-means is that the increased number of iterations will give a more accurate set of clusters, with improved validation scores.

To find the ideal number of clusters the silhouette methods was again used for the range of 2 to 15. The number of iterations for each k-means run has been set to 50. Ideally this number would be higher, but running this process through 13 times is fairly time intensive, so a smaller number has been selected in the hopes of achieving an acceptable silhouette score whilst keeping computational resource to a minimum.

The results for the silhouette method using the Tensorflow package are shown in the plot below.



Given that the second peak of the silhouette scoring method occurs at a similar point to our previous algorithms (k = 11), the final clusters using the Tensorflow version of K-means will be set at k = 11. The algorithm was run once more with these parameters to produce an output set of clusters.

The initial thoughts for a Tensorflow iteration of k-means, was that it's increased computational power would lend an increased accuracy to the k-means algorithm. Unfortunately this does not appear to be the case. Ultimately the project does not have the resource to investigate why this may be. There is a possibility that more tuning could give better results, but in all likelihood the data simply is not of a high enough dimensionality to require computational complexity that Tensorflow can provide.

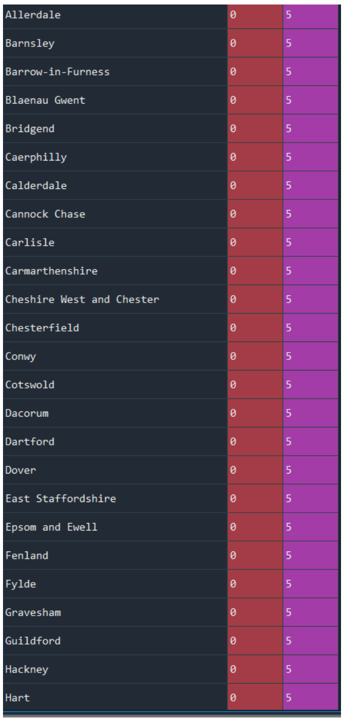
There may be more work to be done here, as Tensorflow takes a selection of other hyperparameters that were not investigated in this work. These include epoch number, learning rate and sigma value. Provided with more time this may have been an interesting avenue of investigation, but with the limited resource for this problem the decision has been made to not investigate the Tensorflow implementation any further. That being said, outputs of cluster visualisation will be produced as with the previous algorithms for the sake of consistency.

Results

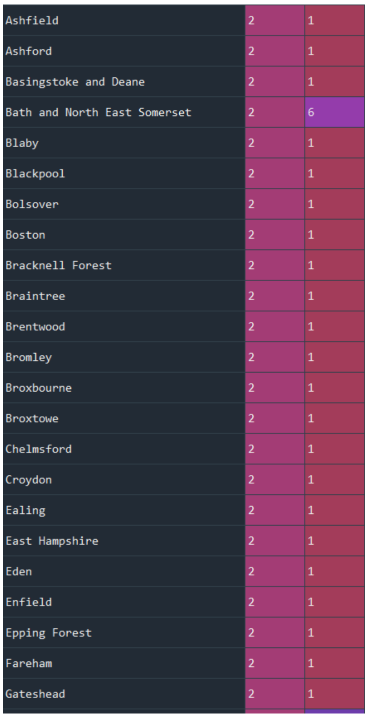
When considering the output clusters of these algorithms, no truly ‘correct’ method for validating them truly exists. The hypothesis for this piece of work is that the churn variables would provide a way to identify “harder to count” populations such as students, military staff, and non-permanent workers. Considering this, the identification of trends within our clusters will be difficult to interpret for anyone without a strong knowledge of the demographic breakdowns of the UK’s LA’s.

To solve this problem, the cluster groupings were exported from the code as a .csv file, and sent to two colleagues in the ONS that have been identified as having an expert understanding of UK demographics across LA’s. Rather than sending all four files for evaluation, both colleagues were sent one cluster set each, as to limit time investment for those unaffiliated with this project. The cluster groupings chosen for this were the K-means and SOM cluster sets, as they both received slightly higher silhouette scores than those achieved by the hierarchical and Tensorflow methods, and also represent one simple method (K-means) and one more complex method (SOM). Hopefully this will help us answer the question of whether more computationally advanced clustering methods produce better results.

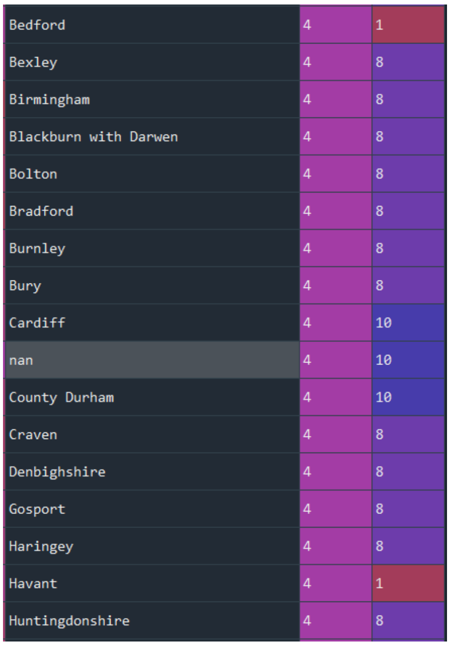
When comparing these two cluster groups, we can see very clear commonalities between them. Whilst the random nature of cluster centroid starting points mean that the cluster numbers are not shared, there are many local authorities that appear in the same cluster in both the K-means and SOM cluster groupings. This can be viewed by joining the two output dataframes on their shared LA name column, and viewing the dataframe via the Python IDE.



Allerdale	0	5
Barnsley	0	5
Barrow-in-Furness	0	5
Blaenau Gwent	0	5
Bridgend	0	5
Caerphilly	0	5
Calderdale	0	5
Cannock Chase	0	5
Carlisle	0	5
Carmarthenshire	0	5
Cheshire West and Chester	0	5
Chesterfield	0	5
Conwy	0	5
Cotswold	0	5
Dacorum	0	5
Dartford	0	5
Dover	0	5
East Staffordshire	0	5
Epsom and Ewell	0	5
Fenland	0	5
Fylde	0	5
Gravesham	0	5
Guildford	0	5
Hackney	0	5
Hart	0	5



Ashfield	2	1
Ashford	2	1
Basingstoke and Deane	2	1
Bath and North East Somerset	2	6
Blaby	2	1
Blackpool	2	1
Bolsover	2	1
Boston	2	1
Bracknell Forest	2	1
Braintree	2	1
Brentwood	2	1
Bromley	2	1
Broxbourne	2	1
Broxtowe	2	1
Chelmsford	2	1
Croydon	2	1
Ealing	2	1
East Hampshire	2	1
Eden	2	1
Enfield	2	1
Epping Forest	2	1
Fareham	2	1
Gateshead	2	1

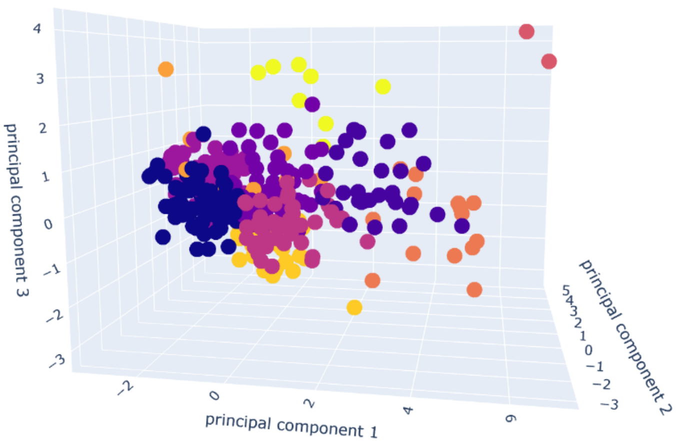


Bedford	4	1
Bexley	4	8
Birmingham	4	8
Blackburn with Darwen	4	8
Bolton	4	8
Bradford	4	8
Burnley	4	8
Bury	4	8
Cardiff	4	10
nan	4	10
County Durham	4	10
Craven	4	8
Denbighshire	4	8
Gosport	4	8
Haringey	4	8
Havant	4	1
Huntingdonshire	4	8

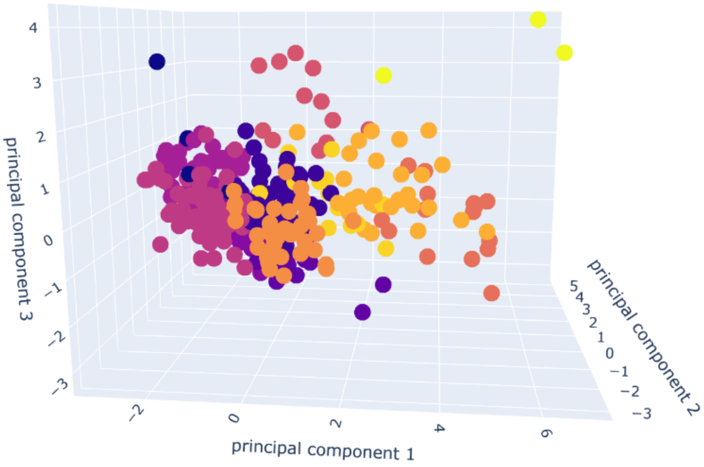
While there are differences in the clustering, more often than not both of these algorithms placed the same LA’s together into groups. This indicates that the trends between datapoints were ‘visible’ to different methods of clustering, and as such are true trends apparent in the data.

For further analysis of the clusters, 3d Plotly scatterplots were generated for each output. These are interactive HTML plots, and as such cannot be displayed here in their interactive form (However they will be included with the code outputs for further interrogation). It’s also worth bearing in mind that due to cluster numbers being different with each algorithm, the datapoints in each plot will be slightly different.

Below is the plot for the K-means cluster outputs.



And the following is the same plot for the clusters made with the Minisom self organising map package.



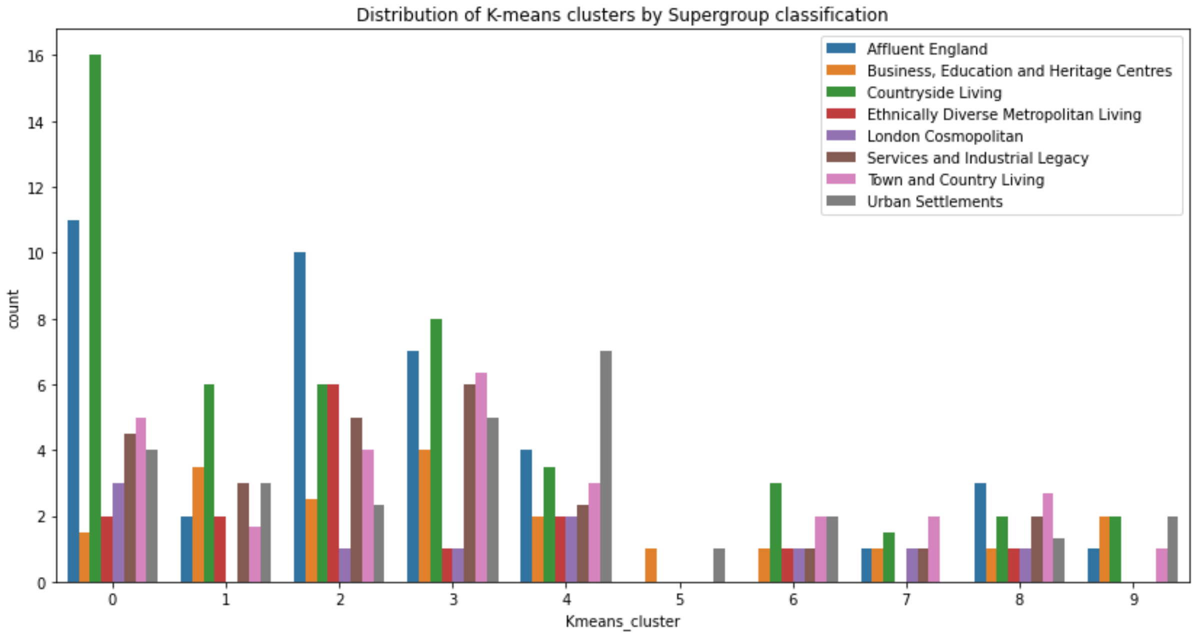
While the colours of each cluster is slightly different, we can clearly see areas of datapoints clustered together similarly in both sets.

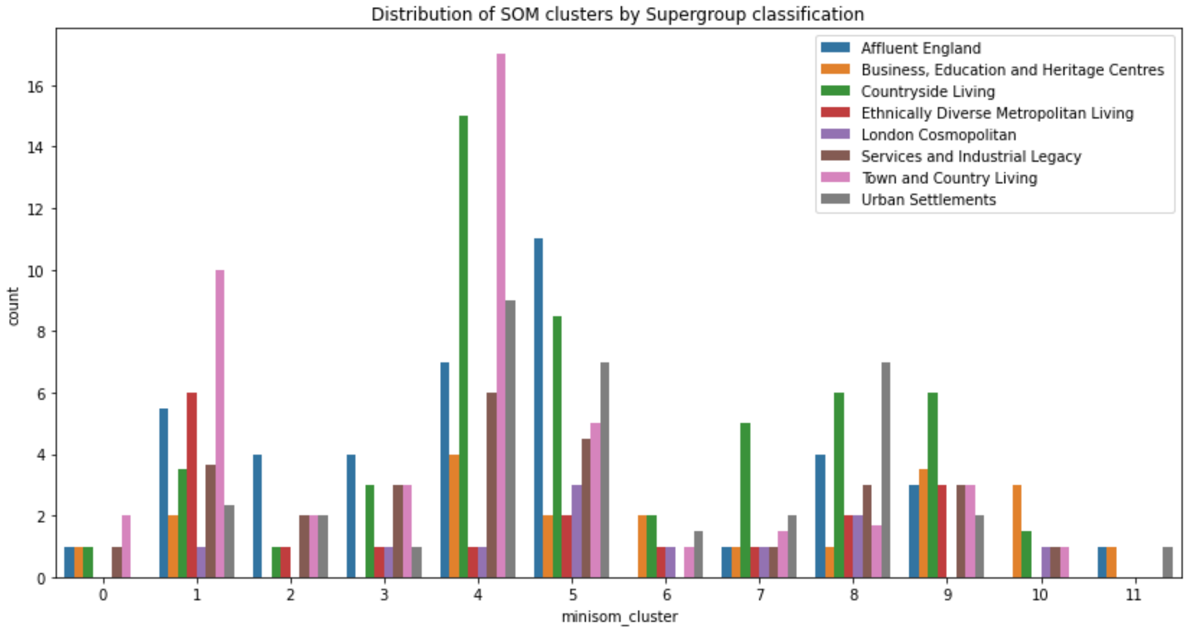
Manual Evaluation

The final outputs of this clustering work consist of cluster-specific spreadsheets with one Local Authority name column and one cluster id column. The most time efficient method of validation for these clusters is to submit them for manual evaluation by colleagues with an expert understanding of the demographic breakdown of Welsh and English LA's. The K-means file and SOM file were both submitted to two such colleagues for assessment.

Initial feedback was that no distinct patterns could be witnessed in the clusters. Expectations were that clusters would form consisting of authorities with similar characteristics regarding average age, industry types and ethnic makeup, but from manual inspection by a qualified expert this does not seem to be the case. This does not however mean that there are no common features between clustered LA's.

Another method considered for valuation is the comparison of clusters to different LA classifications generated by the ONS. One such classification is the ONS' Supergroup and Subgroup systems. These classifications are used by the ONS to designate a label to the general makeup of a local authority, such as "Affluent England", "Town and Country Living", or "Urban Settlements". With the cluster datasets joined onto the open source datasets for these classifications, it becomes more apparent that the makeup of the clusters does not match with any particular Supergroup classification. Here we see that each cluster is made up of a number of different supergroups (Often containing at least one of each).

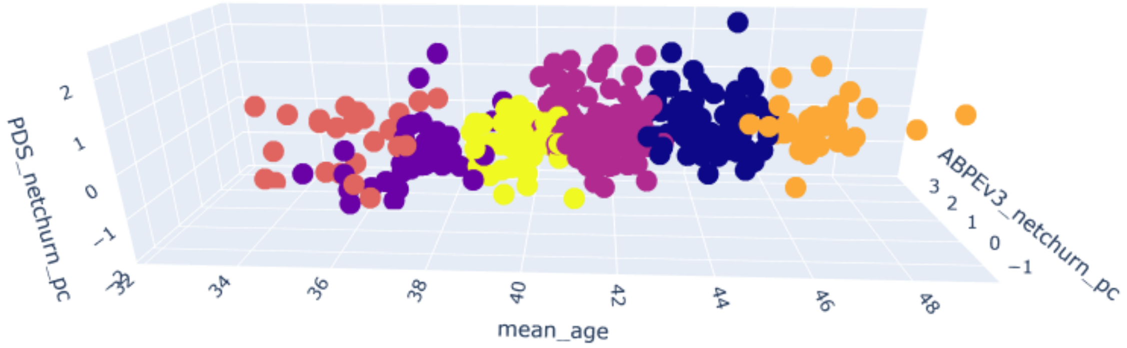




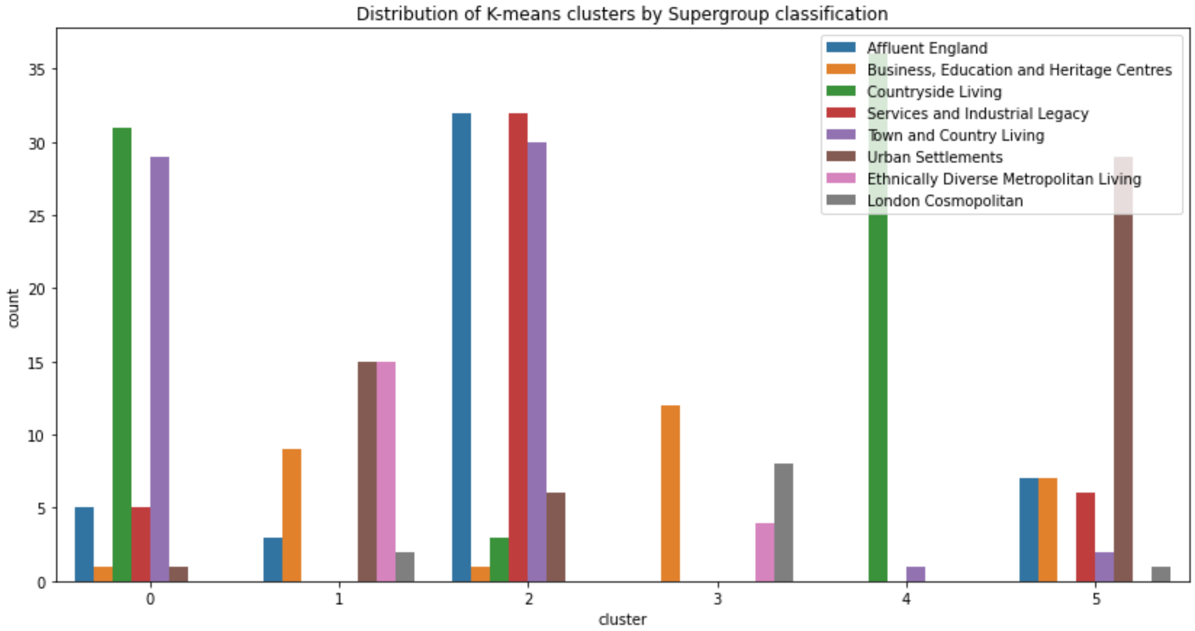
Considering that no clear correlation is shown between cluster groupings and any other classification assigned by the ONS, one potential next step is to repeat the clustering process with a different number of clusters. Of course repeating this with each algorithm would be very costly in terms of time and human resource, so only one algorithm will be selected as one last point of investigation to see if any more substantial patterns can be found within the clusters.

For this I have selected the k-means algorithm simply due to it's ease of use and reliability. Looking again at the *optimal-k* validation plots, the silhouette method displayed a score of 0.311 at *k*=6. The score was still not particularly high, but as a final exploratory step, generating these new clusters could give valuable insight into the best way to group UK local authorities.

After running the clusters with 6 centroids, the clusters were visualised and exported much in the same way as the previous sets. Below is a 3d visualisation of the clusters in Plotly's 3d-scatter function.



Here, we can see fairly well-defined cluster structures, however the clusters are clearly very strongly influenced by the mean_age column. Comparing these new clusters in a grouped bar chart can also provide some interesting findings.



While not perfect, here we can see a much clearer trend in certain clusters being made up primarily of one or two supergroup types. For instance, cluster 4 is primarily Countryside living, and cluster 5 is primarily urban settlements. Unfortunately this link may or may not be a result of this algorithm being run on the raw data, with age range being a key factor when considering the demographic layout in each particular supergroup. This possibility is further supported by the 3D Plotly scatter plot. As a result of this uncertainty, more testing should be done on the links between cluster layout, population age, and similarities to UK supergroup classifications.

Conclusion

This study aimed to use machine learning to cluster local authorities into groupings of local authorities that bore similarities based on several different types of churn variable. Considering the results mentioned above, it is difficult to claim that the clustering has been completely successful or not. While the clusters created bore no concrete resemblance to existing groupings produced by the ONS, it is important to understand that these groupings are in and of themselves not perfect classifications. While no obvious trends between clustered LA's was obvious in terms of demographic makeup, this work serves as an important piece of exploratory analysis, and may lead to more extensive clustering projects in the future.

The project has provided some interesting insight into the ML clustering process. The fact that separately generated cluster sets were very similar indicates that patterns are in fact hidden in the data but are difficult to detect from a human perspective. This thought is reinforced by the fact that fundamentally different algorithmic methods identified very similar clusters. The outcome is ultimately a positive one, despite it being difficult to interpret. With this in mind it will be useful to hold the cluster groupings for consideration in any future work, as future analysis of the Local Authority groupings may lead to a better understanding of common traits within the clustered LA's.

An additional possibility is that the data used in this project simply was not complex enough enough (Either in feature size or data complexity) to allow the algorithms to clearly separate the datapoints into neat clusters. One of the best uses of time in carrying this work forward would be to increase the number of features. This data only took churn variables from the PDS and SPDv3, whereas similar variables could be derived from other administrative data sets, such as the Higher Education Statistics Agency (HESA) datasets. It is also important to consider that the 3 PDS variables are derived from the same source, and as such strongly connected. While there may be reasons for not including more data in the current state of the project (data security issues for instance), future iterations may be able to implement these changes. This variation in an improved dataset could give more information for the clustering algorithms to interpret, and display much more consistent cluster groupings.

References

(2021). *A Beginner's Guide to UK Geography.* Retrieved from http://www.nationalarchives.gov.uk/doc/open-government-

Abbot, O. (n.d.). *2011 UK Census Coverage Assessment and Adjustment Methodology.*

Agrawal, S., & Phillips, D. (n.d.). *Catching up or falling behind? Geographical inequalities in the UK and how they have changed in recent years.* Retrieved from www.nuffieldfoundation.org

Basel Abu-Jamous. (2015). *Integrative Cluster Analysis in Bioinformatics.*

Blackwell, L. (n.d.). *Admin-based population estimates and statistical uncertainty: July 2020.*

Cashen, M. (2004). *Population Churn: The Migration Flow Of Florida.* Retrieved from http://library.ucf.edu

Dalton, L., Ballarin, V., & Brun, M. (2009). *Clustering Algorithms: On Learning, Validation, Performance, and Applications to Genomics.*

Halkidi, M. (2001). *On Clustering Validation Techniques.*

Kohonen, T., & Oja, E. (1998). *Neural Computing & Applications Visual Feature Analysis by the Self-Organising Maps.* Springer-Verlag London Limited.

Pedregosa FABIANPEDREGOSA, F., Michel, V., Grisel OLIVIERGRISEL, O., Blondel, M., Prettenhofer, P., Weiss, R., . . . Duchesnay EDOUARDDUCHESNAY, F. (2011). *Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot.* Retrieved from http://scikit-learn.sourceforge.net.

Scanlon, K., Travers, A., & Whitehead, C. (2010). *Population churn and its impact on socio-economic convergence in the five London 2012 host boroughs.* Retrieved from www.communities.gov.uk