Data Mining Report

36155273

MSc Data Science SCC403: Data Mining

***Abstract*— This paper aims to briefly describe the methodologies of current clustering and classification machine learning methods and give a discussion of their advantages and limitations. Specific clustering algorithms such as KMeans, GMM, and Spectral Clustering were implemented in python [20] on a climate dataset, to show true evaluations of clustering performance. Binary classification of an image stream was also performed using SVMs and Linear Regression models, again using well-known metrics to compare classification accuracy. The overall results were that KMeans gave the best fit however most clustering was disappointing, whereas classification was very high performing with SVM (using RBF kernel) giving excellent results.**

I. Introduction

The aim of this paper was to explore clustering and classification algorithms on two separate datasets. The first was the clustering of a climate dataset used contained 1763 rows and 18 features, covering minimum, maximum and mean values of aspects of the climate in Basel, from winter and summer months, such as temperature, humidity, pressure, wind gusts and wind speeds. Further features on precipitation, snowfall and sunshine duration were also a part of the data set. The second was the binary classification of a small, labelled dataset on the shapes of a car and motorbike from a given video stream. It was important to note that the data was already linearly separable and so was greatly simplified.

There are many different clustering techniques, but they can be mainly broken down into four broad categories [1,2]:

a) Centroid/partition-based: Uses initialised centroids and iteratively partitions the data into clusters until a threshold is reached. The popular KMeans method was explored in this paper.
b) Density-based: Starts with a single data point and iteratively creates clusters by finding differing densities of points, normally based off their distances from one another. DBSCAN is an example of this, although it was not explored because these methods usually struggle with high-dimensional data.
c) Hierarchal: Agglomerative or divisive method that either continues to build the clusters up from a bottom-up approach or divide the dataset into respective clusters from a top-down approach. These are simple and logical approaches focusing on the linkages between points but are very computationally taxing and lack performance in comparison, so were not explored further.
d) Model-based: Assumes that the data is from a certain distribution and uses a maximum likelihood approach to find the best clusters. These methods do not assume that points centre around the centroids, unlike KMeans. Gaussian Mixture Modelling was studied further in this report.

Of course, some methods do not belong strictly to one of these categories however they do often build off the principles. For example, spectral clustering stems from graph theory, but often uses KMeans to find the clusters after the data points have been embedded into a space using eigenvectors to make them more 'obvious' [3]. It often produces much more accurate clusters at a cost of higher time complexity, due to it using an affinity matrix that calculates the similarity/ correlation between all the features. The data was high dimensional and so the 'curse of dimensionality' had to be dealt with, ensuring feature extraction and reduction was performed sufficiently before applying any model [4].

Classification is a supervised learning problem, unlike clustering, where the labels for each data point are provided. The methods use this labelling and the relation between the features and labels to produce a model that can predict the class of new data. There are many types of classification methods: feature selection, probabilistic, decision trees, rule-based, instance-based, SVM and neural networks [5]. Feature selection must be done at the beginning of every method and so is more general than the rest. It involves testing to find the best features that describe the data and reducing them down to decrease the computational load. Probabilistic methods such as logistic regression use statistical inference for their decisions and are explored further in this report. Rule-based methods and decision trees are very similar, both creating very robust and easily trained models however rule-based is more of a generalisation, whereby partitioning of the data is not strict like decision trees. Instance based learning is often referred to as 'lazy learning' because the test data is often derived straight from the training data without any modelling required. K-nearest-neighbour is a distance-based approach that often performs well on simple datasets despite its minimalism. SVM was another method explored in this report, relying on linear methods to split the data by a line that optimises the separation of data for better generalisability. Finally, neural networks are a modern development that emulate the biological properties of the human brain, strengthening the connections between layers of created nodes that model the data, leading to usually excellent performance on complex data. This was not explored as it is much too complicated for the simple problem faced.

This report will go through the stages of pre-processing the two datasets and explain how every model was produced, discussing the limitations, benefits, and overall performance of each.

II. Pre-processing

A. Climate Data

Firstly, a simple search for null values in the dataset was performed to detect if any rows contained missing data that might need to be dealt with. Zero null values were found, and so no further investigation was needed.

Next, the distributions of the features were plotted to investigate any statistical trends in the data that could help with the next steps in pre-processing (e.g., decision to standardise or normalise) and produce a base of understanding for any assumptions made during model fitting. The histograms in Appendix A1.1 show that the distributions of temperature, humidity, wind, and pressure were all approximately Gaussian (the means are shown here for simplicity, but min and max values also followed the same pattern). In particular, there was a potential multivariate normal distribution in Temperature, which is logical as the data comes from summer and winter separately, therefore likely to have noticeable differences.

The remaining three features on precipitation, snowfall and sunshine duration were also assessed in this method. Precipitation and snowfall histograms showed an overwhelming number of low values in each column with a few extreme values also. After investigation, 80% of rows in the precipitation feature contained values < 2mm, and 93% of the snowfall feature contained values of zero. Due to the high number of low values in each of these features, a choice was made to remove them from the dataset. This was because they were very unlikely to provide any influence on the models training since there is an extremely large skew towards minimal values, resulting in little information gain. Models were later created including these features and influence was indeed insignificant, but this will not be included. On the other hand, sunshine duration, other than the high value for zero, showed a clear uniform distribution due to the linear decrease in this factor in northern hemisphere countries as the year progresses. This could provide a linear relationship with other features and importantly provide information for the clustering models therefore this feature was kept. The high zero count again was potentially problematic however they only account for 11% of the data and it was reasonable that winter days could have no sunlight.

There is clear high correlation between the min, max and mean features which is logical, looking at Appendix A1.2. Although lower, there is also a clear relationship between sunshine duration, humidity, and temperature. The correlation here is not significant enough to warrant the further removal of features however a technique needs to be applied to account for the high correlation between min, max and mean values for different features. One technique would be to use Principal Component Analysis (PCA) to reduce the number of features.

To use this process, the data must first be centralized and scaled. As shown, the data is mainly Gaussian and so standardisation was very appropriate and the preferred method over normalisation in this case. Although normalisation would work, it was much more suitable to use standardisation and did produce much clearer results in the models later made. Standardisation centres the data by subtracting the mean and then scales it by dividing by the standard deviation. This is done for many reasons: 1. The features have different scales, so it was important to scale them for equal weighting, 2. PCA requires data to be centralized and scaled for equal weighting due to its use of the standard deviation in calculating the projections of the new axes, 3. Outlier detection can easily be done on the transformed data (PCA is sensitive to outliers too).

To detect these global outliers, each “mean” feature was analysed and any values greater than 3 or less than -3 were removed due to them being too extreme. The mean features were chosen because they show the distribution better, compared with min and max which are much more susceptible to large variation. Finally, sunshine duration was not investigated for outliers due to its uniform distribution (standardisation outlier detection would not be fair). In total, 43 rows were dropped (39 due to wind, 2 humidity, 2 pressure and 0 from temperature). Note that no outliers from temperature could be due to the multinormal distribution it shows as there are clearly 2 means for the split and so this could have been investigated further.

Finally, PCA was applied (see Appendix A1.4 for component variation). The first 4 principal components were chosen for further use because they explain almost 90% of the variance in the data and there is not much information gained by choosing more components. This would also increase the dimensionality of the data leading to higher time complexities in model fitting and the trade-off for this extra accuracy was not worth it.

Although PCA is highly effective for dimensionality reduction (have reduced from 16 features down to 4) and allows the data to be visualised in 2 dimensions, it does have its limitations. The main being that making deductions from the clusters one creates is much more difficult because the original features are lost. However, some intuition from these features can still be found by looking at the loadings plots (Fig. 1) from the correlation between the principal components and the original features (see appendix A1.5) [6].

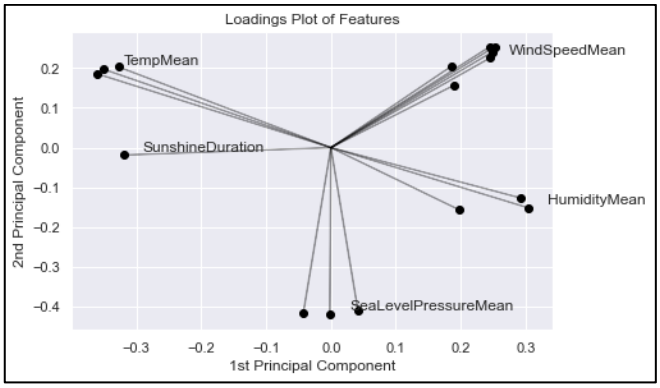


Fig. 1. Loadings plot for original features on PC1 and PC2 [19]

The highly correlated features (min, max, mean) are grouped together, shown by just the mean labels, and there is already a basis of understanding for how the data may be clustered.

B. Car & Motorbike Video Stream

Most of the pre-processing for this dataset had already been completed by the supplier, where a video stream of an arial view of a car and motorbike was taken. Boxes were drawn around these objects, and in each frame, the length and width of the boxes were recorded and assigned a label from human assistance. An area of each box was calculated also, but since the correlation between the other two variables was extremely high (>90%), this feature was removed. The first 16 rows of the dataset were also removed as these frames only included the presence of one object, the motorbike, which could lead to selection bias in classification where one label gets more training than the other. If no car was in the frame, then it was impossible not to label it as a motorbike, hence the population is not representative. The remaining 172 rows are thought as 86 pairs of objects in the frame.

III. Clustering

A. KMeans

Firstly, a KMeans model was applied. This is a very robust algorithm, is simple to implement and understand, and generally has good clustering performance. It is a distance-based metric and so assesses how 'similar' data points are geometrically through a distance measure (Euclidean was used here). The clusters are formed by repeatedly calculating new centroids, based off the mean of the points assigned to those clusters, until some threshold defined is reached. A limitation of this method, like many clustering algorithms, is the choice of k (the number of clusters/ centroids) must be stated initially. This k was found using several different methods (Fig. 2) [7]:

a) Elbow method: manually look for an 'elbow' or kink in the within-cluster-sum-of-squares (WCSS)/ inertia. From the graph, 3 or 4 clusters could be the best.
b) Silhouette scores: evaluates how good the clustering is by using the pairwise distance of intra-cluster distances and the nearest-cluster distances for each cluster. In this case, there is a clear spike at k=3 where the score is maximized.
c) Calinski-Harabasz Index (CH Index) is a "variance ratio criterion" [8]. Intuitively, it measures clustering fit through the tightness of the cluster and the rigidity between clusters. The aim is to maximize this metric and so the graph shows k=3 is best.
d) Davies-Bouldin Index (DB Index) shows the similarity of clusters based on the data density, which decreases with distance, and can find the appropriateness of partitions [9].The aim is to minimize this score and so k=3 is again best.

The gap statistic was another metric used and is discussed further in appendix B1.2 [10].

Other limitations of this algorithm are that it is not sensitive to the density of the data and so no clusters can overlap, and the initial setting of the centroids (although random) can lead to unwanted results from outlier influence.

In conclusion, k=3 was clearly the correct choice, and so the results were plotted in Fig. 3, indicating the different assigned labels by colour, and using the first two principal components as axis. Meaning can be abstracted from these clusters by using the loadings plot from Fig. 1. The direction of the original features indicates the correlation to the first

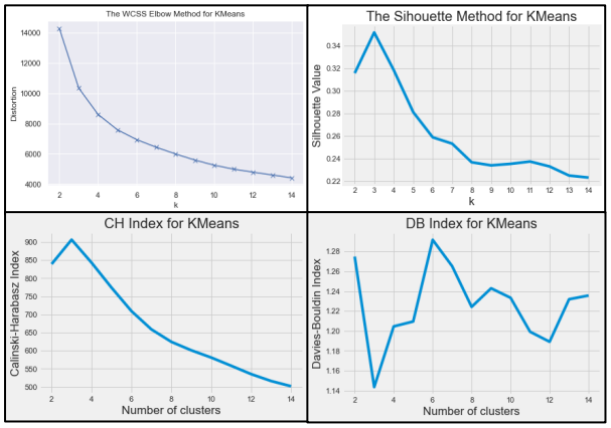


Fig. 2. Finding appropriate k for KMeans via Elbow Method (top-left), Silhouette Score (top-right), Calinski-Harabasz Index (bottom-left) & Davis-Bouldin Index (bottom-right).

two components therefore one can infer that the blue cluster shows windier, winter (shorter sunlight duration) days with colder temperatures, yellow shows hotter summer (longer sunlight duration) days with little wind and red indicates high pressure days (likely for both seasons) as pressure is inversely proportional to high wind. It is interesting to note that humidity is higher in colder months in Basel.

In addition, Mini-Batch KMeans was also explored to notice any difference in results. This is a much more computationally efficient algorithm as it takes samples of the data at each iteration in the algorithm, instead of the whole dataset, meaning less distance measures are taken, but as a result some accuracy is lost [11]. Because the size of the dataset is relatively small, the results of clustering were extremely similar (slight decrease in scores from KMeans).

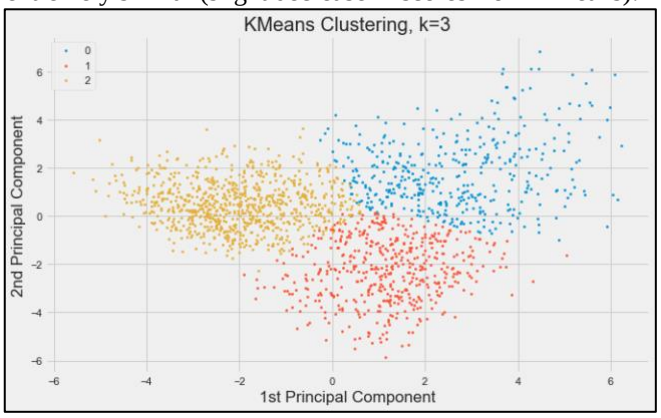


Fig. 3. KMeans clustering (k=3) for climate data set.

B. Gaussian Mixture Model (GMM)

As mentioned earlier, the marginal distributions of the main features (excluding Sunshine Duration) have a Gaussian nature. In particular, temperature shows a clear bimodal distribution by the separation of peaks, and, although not clear, there is potentially the same trend in the other features (especially Humidity) where two peaks have potentially merged. This gave motivation to explore a Gaussian Mixture Model (GMM) due to its fundamental assumption that the data is a result of sampling from distinct sub-populations; in this case, the dataset is from summer and winter seasonal climate data, two very contrasting periods of time.

This is a probabilistic clustering method, relying heavily on the underlying statistical assumptions of the data, which contrasts heavily to KMeans, and the general approach used to find suitable parameters is called Expectation-Maximisation (EM). This algorithm initialises with a model and then iteratively maximises the likelihood by taking the previous model as a basis for estimation. This continues until convergence is met based upon a threshold [12]. EM can be seen as a 'soft' version of KMeans, where it uses a similar approach, however calculating centroids using a probabilistic method on the density, which leads to no strict boundaries and the possibility of overlapping clusters. Like KMeans, this approach requires the initialisation of the number of components/clusters, k. The same approaches used for KMeans can be performed to find this k however the most reliable method for GMMs is using the Bayesian Information Criterion (BIC). Maximum Likelihood Estimation is logically the method one would think of when trying to find

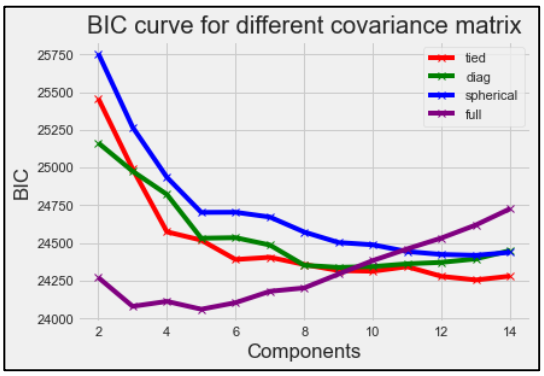


Fig. 3. BIC curves to find the choice of k and covariance matrix

the best k, however the likelihood will always pick the highest dimensionality, due to its increasing nature. This will lead to overfitting of the data if allowed. The BIC considers the complexity and penalises models with higher number of parameters [13]. Akaike's Information Criterion (AIC) has a similar nature however does not penalise as harshly and because generalisation is of most importance, BIC was used. GMMs use a covariance matrix in the algorithm and the choice of this changes the shape of the clusters. "Tied" is a general covariance matrix for all features, "diagonal" just allows different diagonal covariance matrices for each feature, "spherical" means each feature only has its own variance, and "full' gives a unique covariance for each feature. From Fig. 3, the BIC is lowest for "full", which is a more complex method but works well with the small size of the dataset. You can see how the BIC increases with k for this choice compared to the other methods, due to the increase in complexity. The lowest values are at 3 and 5, but using Occam's Razor, the simpler k=3 is chosen again. This is further evidenced in Appendix B1.5.

There is a clear difference in the way the clusters have been formed, in comparison to KMeans, displaying overlaps and possibly allowing for more variation in features (Fig. 4). The fundamental structure and inference from the original features from the clusters is the same as KMeans, however there is more of a spread in the upper left (red) cluster, resulting in less points clustered in the upper right (yellow).

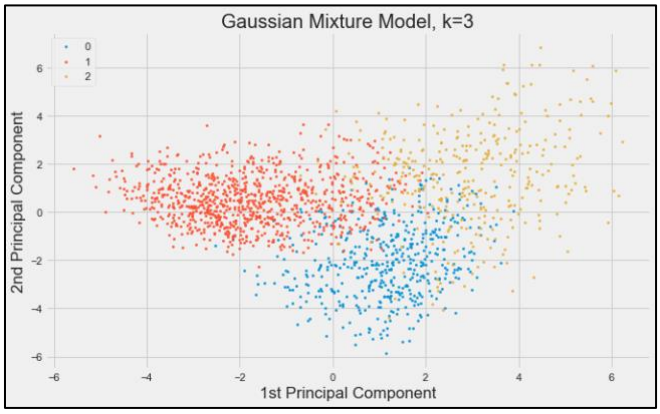


Fig. 4. GMM for 3 clusters

C. Spectral Clustering & BIRCH

These two algorithms were further implemented for more comparison. Spectral clustering is normally useful on very high dimensional and complex data so the dataset used may not qualify but was performed anyway to check for any improvement in clustering performance. On the other hand, BIRCH is an online learning algorithm that drastically improves performance and is an alternative to Mini Batch KMeans, based on a hierarchal approach to finding centroids. The process for fitting the models and results is included in appendix B1.6 and appendix B1.8.

D. Summary & Comparison of Methods

In summary, despite it being the simpler of the algorithms, KMeans performed the best clustering of the dataset, giving the best value for all 3 comparable statistics (Fig. 5). This means the clusters were denser and contained more 'similar' points. It is important to note that all models performed similarly, and these scores only show theoretical values for how well the data is clustered; there is no true accuracy because there are no corresponding labels for this dataset. The CH Index is a ratio and so this metric is purely dependent on the data, whereas the DB Index has a minimum of 0 and so the clustering performed on this data is quite poor. This is further exemplified by the low Silhouette Scores, which could have ranged from -1 to 1 (-1 being no clusters at all, and 1 being perfectly separated). This is due to the original data provided being very dense and containing many features, so it is hard to partition and potentially visualize in 2 dimensions.

Time complexity was not a problem however, for completeness, the relative times of training were still considered. BIRCH and Mini-Batch KMeans showed their known efficiency, but surprisingly the GMM was fastest. Spectral clustering was extremely slow in comparison to all other methods due to its complexity and BIRCH was clearly the worst at clustering.

Clustering Method (k=3)	**Metrics (4 s.f.)**			
	Silhouette Score	***CH Index***	***DB Index***	**Time (ms)**
KMeans	0.3515	906.4	1.143	65.50
Mini-Batch KMeans	0.3494	903.3	1.157	34.85
GMM	0.3242	753.5	1.250	20.21
Spectral Clustering	0.3254	690.7	1.205	709.7
BIRCH	0.3042	665.7	1.449	51.89

Fig. 5. Table of Evaluation Scores for Clustering Techniques.

IV. Classification

The classification techniques explored were chosen due to the data being linearly separable, meaning that a straight line can be drawn between the points to separate them entirely. Both SVM and Logistic Regression require this assumption and perform extremely well on binary datasets like the video stream worked with. To assess the performance of the algorithms, the data was split into a training set (70%) and a test set (30%). Since the data was sparse but with distinct clusters, a large percentage was assigned to test to show the effect of missing values on the classifiers and give a more educational viewpoint of training. To keep a consistency across results, the same splits were given for both models, based on a constant random seed. For final evaluation, however, 10-fold cross-validation was performed.

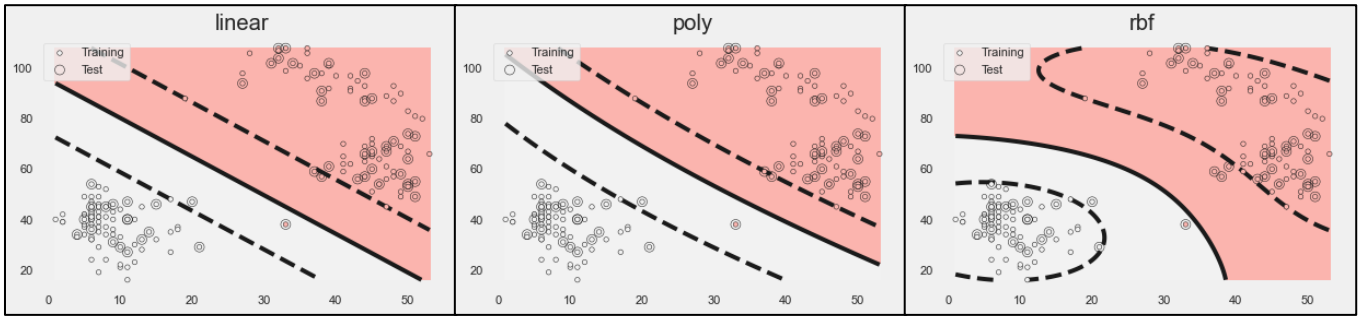


Fig. 6. SVM for different kernels (x-axis= width, y-axis=length)

A. *Support Vector Machine (SVM)*

This algorithm attempts to maximise the distance between points of different labels and then creates a line, in this case, or a hyperplane in higher dimensions, that splits the data into two different classes. The samples that lie on these maximised lines (see Fig. 6 dashed lines) are called the support vectors; these are the points used to find the optimal weighting, w, of the line of separation. Any data that lies on the wrong side of the line is said to be misclassified.

Initially, a linear SVM model was produced (Fig. 7) [21]. There is a clear separation of the data by the straight line and the margin is maximised. There were a few points that sit closer to the boundary and could easily be misclassified if taken as a part of the test set. In this case, two test points, one from each class, are within the constraints of the support vectors and are therefore at risk of being misclassified by the line. In fact, both objects are classified as motorbikes, when only one is in truth. This leads to a drop in accuracy for this random subset of the data. The confusion matrix (Fig. 7) best shows this.

SVM is very robust as it can take different forms if the kernel is changed to a non-linear function which transforms the data into a higher dimensional feature space, allowing a new linear separating hyperplane to be found [14]. Despite the data already being linearly separable, implementations of a radial basis function (rbf) and quadratic (poly) kernel were also implemented as future data, for example, could show there is a more complex decision boundary than initially thought. The quadratic kernel allows for curvature in the line; in this case just a simple curve, but there is no limit on complexity, depending on the problem. On the other hand, the RBF kernel is more generalised. It calculates, using a distance metric, the similarity of points based off the assumption of an underlying gaussian distribution. Specifically, the function is exp $(\frac{distance}{2*variance})$ and so the metric outputted can be seen as dissimilar if it is far from the mean (too many standard deviations away). This variance was carefully chosen in modelling by varying the gamma parameter in sci-kit learn's implementation; a low gamma was chosen (0.0008) which represents a high variance and grants a softer and more generalised decision boundary. Having to define hyperparameters and the kernel can give flexibility but also be a limitation of SVM. Furthermore, the algorithm does not perform well computationally for larger data or for multiclass classification problems [15].

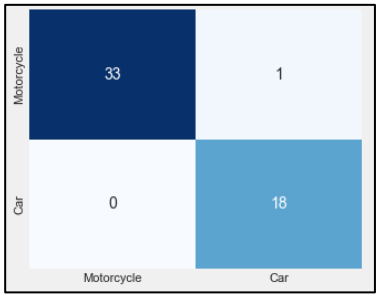


Fig. 7. Confusion Matrix for all classifiers. (Predicted y-axis, True x-axis)

B. *Logistic Regression (LR)*

LR is a more statistical approach than SVM, and despite it being called 'regression', it does work well as a classifier of binary data [16]. It models the probability by fitting a curve based on the log odds (logit function) of a class occurring and then uses maximum likelihood for estimation of the binary outcome [17]. This leads to a straight line, very similar to the linear SVM, that splits the dataset (Fig. 8) and the coefficients can give clearer interpretability than other methods. The confusion matrix for this example is the exact same as the SVM (Fig. 7), as only one point is misclassified.

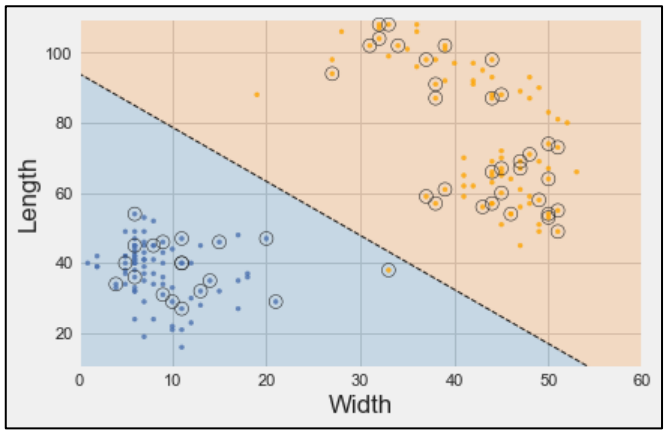


Fig. 8. Logistic Regression Classification [22]

The assumptions of linear regression give it it's weaknesses. the main one being that there is independence between the features [18]. In this case it is assumed and still works well, however the correlation between length and width is very high in fact (~70%) and could lead to problems with more data. Bigger sample sizes lead to better performance and therefore could be a limitation of this method due to only having small data. Therefore, there is potential overfitting.

C. *Summary & Comparison of Methods*

In conclusion, all methods show there is a clear distinction of objects based on their area and classification of these two objects scores very highly in every metric. 10-fold cross validation was performed, splitting the data into 10 distinct, equal-sized subsets randomly. Each subset (10% of the data) is removed iteratively, and the rest is used for training. An average score for each metric can be calculated from the individual scores on each subset left out. This gives a very representative scoring of classification and is used due to the

small size of the dataset. Just taking one split for training and testing would not be demonstrative of classifier performance.

The accuracy gives the number of correct predictions from all predictions made, which is useful as the target variables are very balanced. The most accurate classifier is the RBF SVM. Precision gives the proportion of true predictions for each predicted class (SVM gives top-class precision). Recall takes the proportion of correct predictions for each true class (RBF SVM and LR give the best recall). The F1 score gives a metric that combines both precision and recall for overall performance and is the best indicator along with accuracy. Although very similar, RBF SVM and LR give the highest scores. There is not much between the methods however the RBF kernel does have the best performance overall due to the run time of LR being much slower.

Classifier	**Metrics (2 d.p.)**				
	Accuracy	***Precision***	***Recall***	***F1***	**Time (ms)**
SVM (linear)	98.86%	97.84%	100%	98.75%	0.51
SVM (poly)	98.86%	96.9%	100%	98.89%	0.55
SVM (rbf)	99.41%	98.75%	100%	99.6%	0.65
Logistic Regression	98.82%	99.17%	100%	99.47%	5.2

Fig. 9. Table of Evaluation Scores for Classification Techniques.

V. Conclusion

In summary, several different clustering and classification techniques have been explored and evaluations of their performance have been completed on two chosen datasets. More investigation on the climate dataset could have been performed, in particular, looking at the bimodal nature of temperature, and possibly using other ways of outlier removal on the features as the methods used were not fully robust.

Despite an overall poor performance in clustering, there was some inferences to made on the three clusters chosen by all algorithms, each having their own subtle variations. KMeans performed best despite its simplicity but all models had similar metrics in evaluation. If given more time and room for exploration, the use of the gap statistic would have been implemented for more clustering methods than just KMeans as it provides a unique outlook on the data in comparison to the main known metrics. Furthermore, other distance metrics than the Euclidean distance could be explored for KMeans, and in general, more hyperparameters on all models could be investigated further to tweak performance and find better models. The variation in relative time complexity was also explored showing that for example, Spectral Clustering did not perform well. Visualisations of the data in more than two dimensions could also be studied and could result in more obvious inference in the clustering.

In terms of classification, the results were outstanding, and all algorithms chosen performed very well, mainly because the data was binary and linearly separable. The SVM with RBF kernel was decidedly the best fit method. SVM algorithms outperformed the logistic regression model in time complexity but were otherwise similar. More distinct and complex methods such as neural networks and decision trees are an area of interest that should be researched more, but instead on more complex data, as they were not justified in this case.

VI. References

[1] Aldenderfer, M. S. and Blashfield, R. K., 1984. Cluster Analysis. Quantitative Applications in the Social Sciences. Thousand Oaks, CA: SAGE Publications, Inc. Available at: <https://dx.doi.org/10.4135/9781412983648> [Accessed 19 Dec 2022].

[2] Rodriguez, M.Z. et al. (2019) 'Clustering algorithms: A comparative approach', PloS one, 14(1), pp. e0210236–e0210236. doi:10.1371/journal.pone.0210236.

[3] Liu, J. and Han, J. (2014) 'Spectral Clustering', in Data Clustering. 1st edn. CRC Press, pp. 177–199. doi:10.1201/9781315373515-8.

[4] Aggarwal, CC, & Reddy, CK (eds) 2013, Data Clustering : Algorithms and Applications, CRC Press LLC, Philadelphia, PA. Available from: ProQuest Ebook Central. [19 December 2022].

[5] Aggarwal, C.C. (2015) *Data classification : algorithms and applications*. 1st edition. Boca Raton: CRC Press.

[6] Sanguansat, P. (2012) Principal Component Analysis. Place of publication not identified]: IntechOpen.

[7] Y. Liu, Z. Li, H. Xiong, X. Gao & J. Wu 2010, "Understanding of Internal Clustering Validation Measures", - *2010 IEEE International Conference on Data Mining*, pp. 911.

[8] Caliński, Tadeusz & JA, Harabasz. (1974). A Dendrite Method for Cluster Analysis. Communications in Statistics - Theory and Methods. 3. 1-27. 10.1080/03610927408827101.

[9] Davies, D.L. and Bouldin, D.W. (1979) 'A Cluster Separation Measure', IEEE transactions on pattern analysis and machine intelligence, PAMI-1(2), pp. 224–227. doi:10.1109/TPAMI.1979.4766909.

[10] King, W.M., Giess, S.A. and Lombardino, L.J. (2007) 'Subtyping of children with developmental dyslexia via bootstrap aggregated clustering and the gap statistic: comparison with the double-deficit hypothesis', International journal of language & communication disorders. Received 9 November 2005; accepted 4 May 2006, 42(1), pp. 77–95. doi:10.1080/13682820600806680.

[11] Bejar, J., 2013. K-means vs Mini Batch K-means: a comparison, Available at: http://hdl.handle.net/2117/23414 [Accessed December 19, 2022].

[12] Reynolds, D. (2015) 'Gaussian Mixture Models', in Encyclopedia of Biometrics. Boston, MA: Springer US, pp. 827–832. doi:10.1007/978-1-4899-7488-4_196.

[13] Schwarz, G. (1978) 'Estimating the Dimension of a Model', The Annals of statistics, 6(2), pp. 461–464. doi:10.1214/aos/1176344136.

[14] Huang, X., Shi, L. and Suykens, J.A.K. (2014) 'Asymmetric least squares support vector machine classifiers', Computational statistics & data analysis, 70, pp. 395–405. doi:10.1016/j.csda.2013.09.015.

[15] Patle, A. and Chouhan, D.S. (2013) 'SVM kernel functions for classification', in 2013 International Conference on Advances in Technology and Engineering (ICATE). IEEE, pp. 1–9. doi:10.1109/ICAdTE.2013.6524743.

[16] Patle, A. and Chouhan, D.S. (2013). 'Background: Logistic Regression' in 'SVM kernel functions for classification', in 2013 International Conference on Advances in Technology and Engineering (ICATE). IEEE, pp. 1–9. doi:10.1109/ICAdTE.2013.6524743.

[17] Lorena, A.C., Jacintho, L.F.O., Siqueira, M.F., Giovanni, R.D., Lohmann, L.G., de Carvalho, André C. P. L. F. & Yamamoto, M. 2011, "Comparing machine learning classifiers in potential distribution modelling", *Expert Systems with Applications*, vol. 38, no. 5, pp. 5268-5275.

[18] Osborne, J.W. (2015) Best practices in logistic regression. Los Angeles: SAGE.

[19] Bedre, R., (2021). 'Principal component analysis (PCA) and visualization using Python (Detailed guide with example)', November 7. Available at: https://www.reneshbedre.com/blog/principal-component-analysis.html (Accessed 19 December 2022).

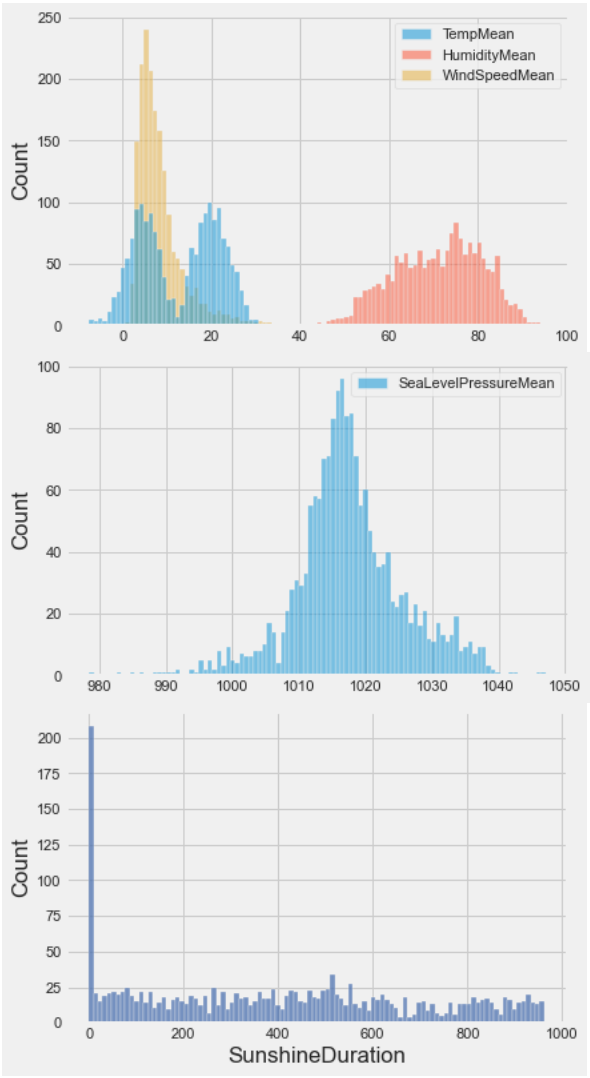
[20] Pedregosa, F. et al., 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct), pp.2825–2830.

[21] Angelov, P., (2022) SCC403 Data Mining, Week 7 Lab: SVM code.

[22] Christian, (2020). 'Plotting the decision boundary of a logistic regression model', September 17. Available at: https://scipython.com/blog/plotting-the-decision-boundary-of-a-logistic-regression-model/ (Accessed 19 December 2022).

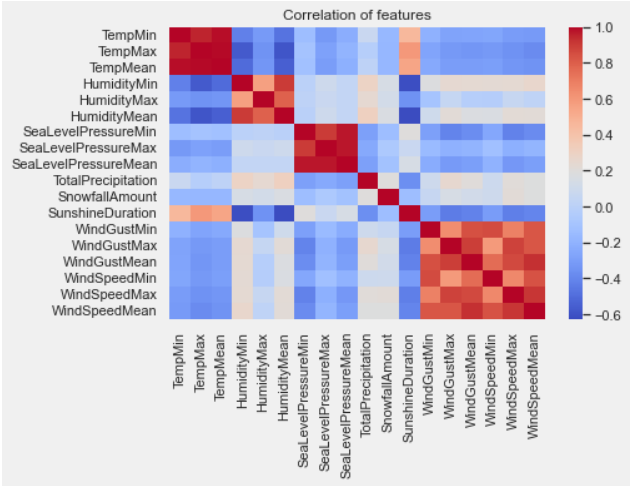
APPENDIX

A1.1



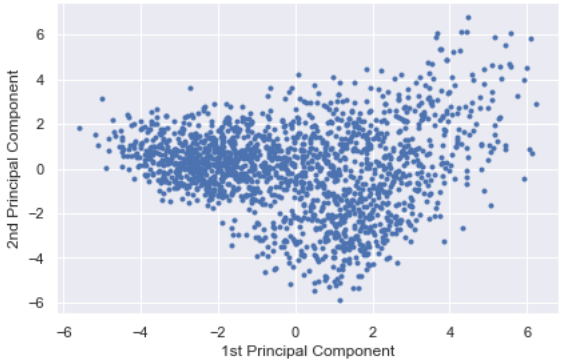
Feature distributions. Clear gaussian distributions in main features and sunshine duration with uniform. Note potential bimodal distribution in temperature and Humidity.

A1.2



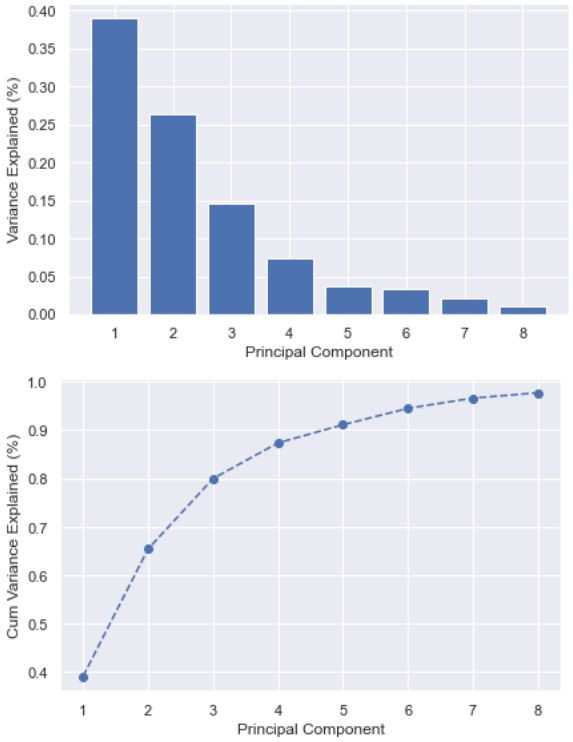
Correlation matrix of all features. Note high correlation between grouped min, mean and max features. Also sunshine duration correlated with humidity and temperature.

A1.3



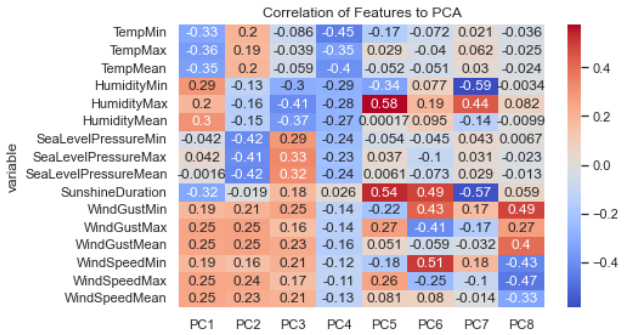
Distribution of data in two dimensions.

A1.4



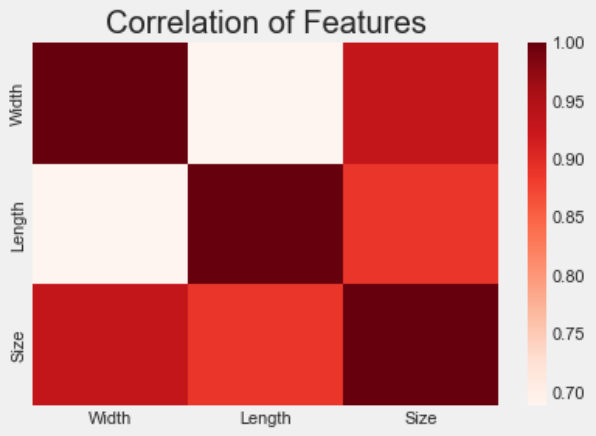
Principal Component variance explained and cumulative variance (scree plots). 4 components chosen due to cumulative variance = 90%.

A1.5



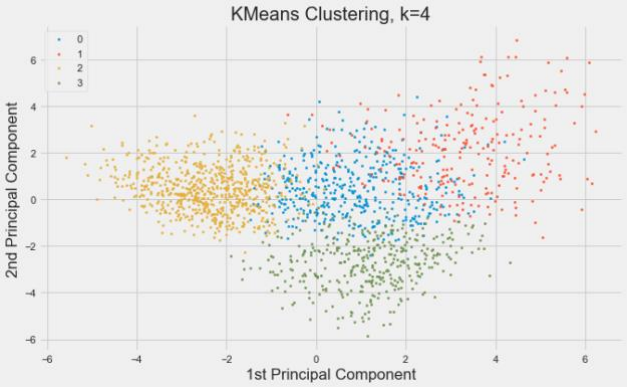
Correlation matrix of features to principal components, used to make loadings plots.

A2.1



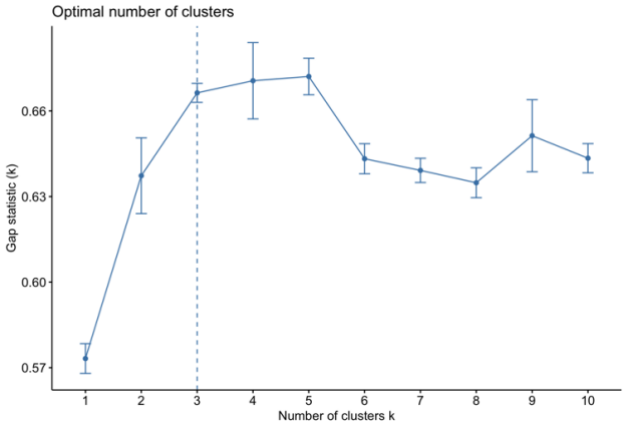
Correlation of features in video stream dataset. Extreme correlation between size and other 2 features but also high correlation between them too.

B1.1



Plot of KMeans with 4 clusters as was close on metrics, to show comparison.

B1.2

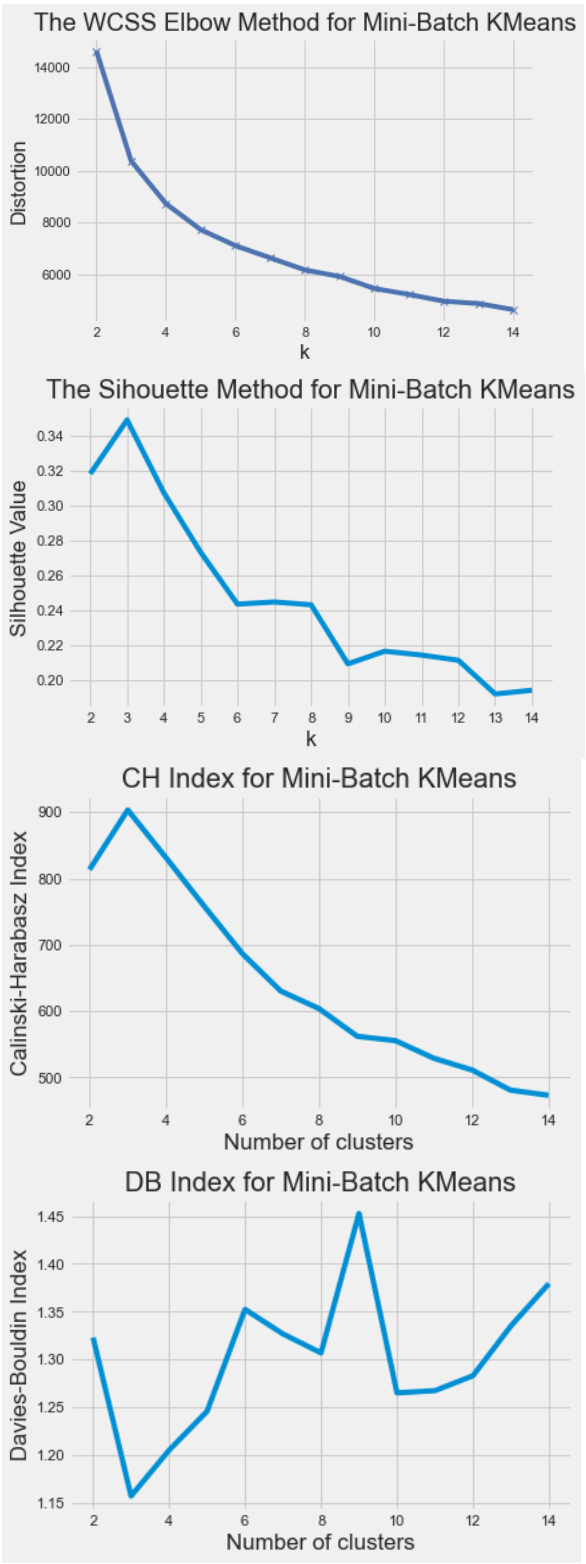


Gap Statistic for KMeans. Created using NBClust package after importing dataset into R.

It is a metric that looks at the within cluster sum of squares, like in the elbow method, and evaluates a high confidence in the k that significantly changes this metric. It is a formalisation and automation of the elbow method, finding the "kink". K=3 is determined to give the "tightest" clusters, just as the other metrics suggested.

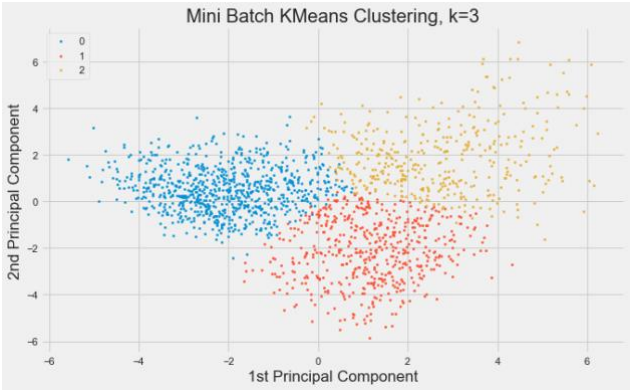
```r
##very short code to calculate gap statistic in R on KMeans

#import dataset after pre-processing in python
data = read.csv("PCAdata.csv")
#library that allows us to calculate gap statistic on kmeans
library(NbClust)
#initialise seed
set.seed(1)
#this function does the gap statistic on the dataset, using a bootstrap method
fviz_nbclust(data, FUN = kmeans, nstart = 1,  method = "gap_stat", nboot = 10)
```

B1.3



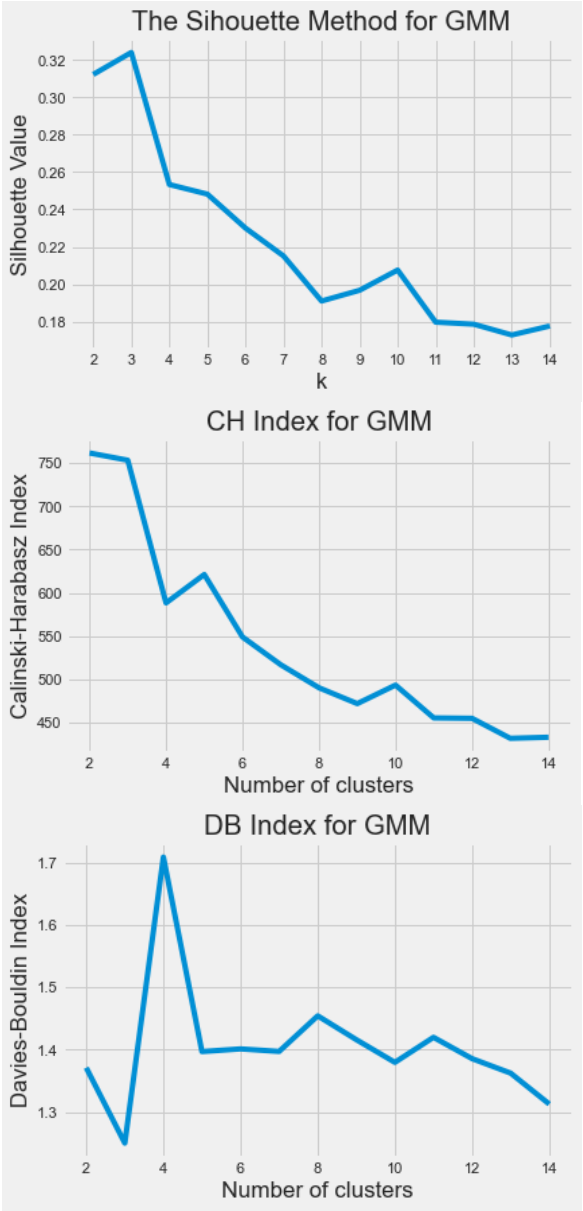
Metrics for Mini-Batch KMeans- very similar to KMeans with slight decrease in performance. K=3 obvious choice.

B1.4



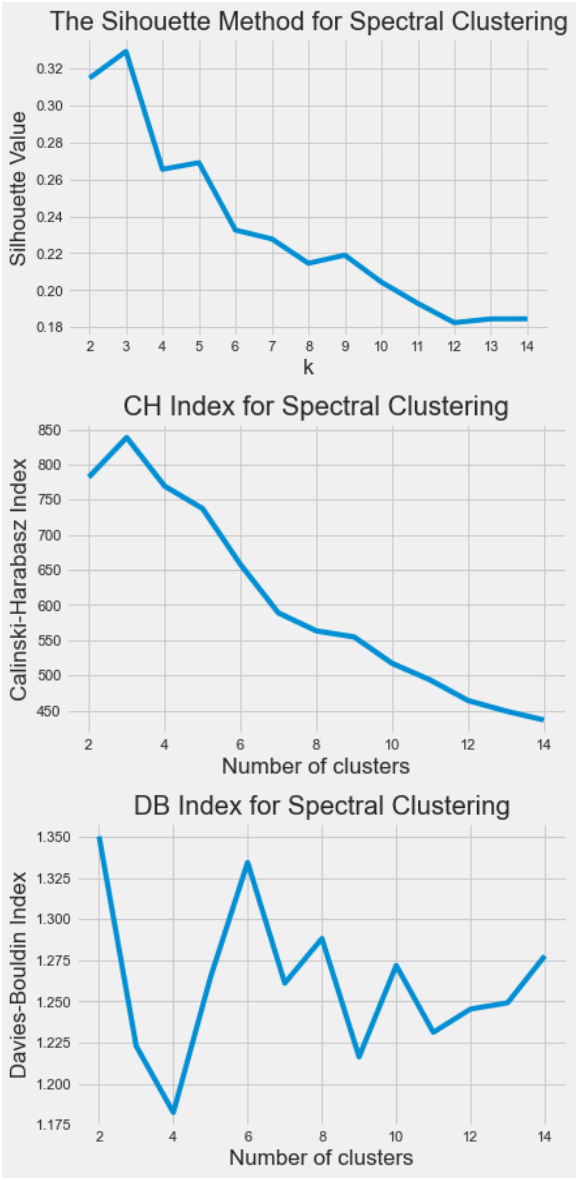
Mini-Batch KMeans plot of clusters- extremely similar to KMeans.

B1.5



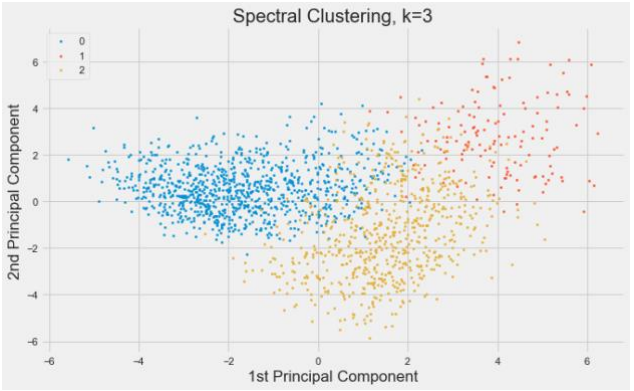
Metrics for GMM to back up choice of k=3 for model. CH value is actually best at k=2 so not the most robust model.

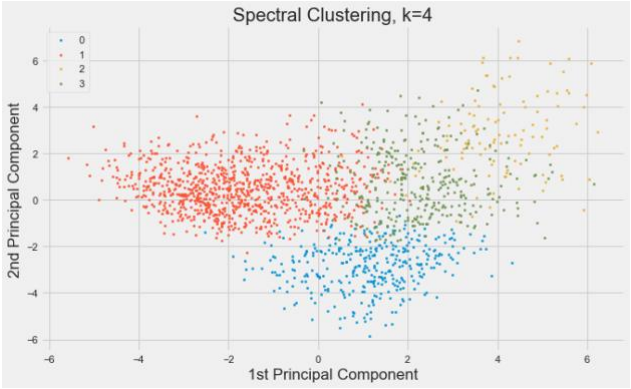
B1.6



Metrics for Spectral Clustering. K=3 chosen however k=4 has better DB index.

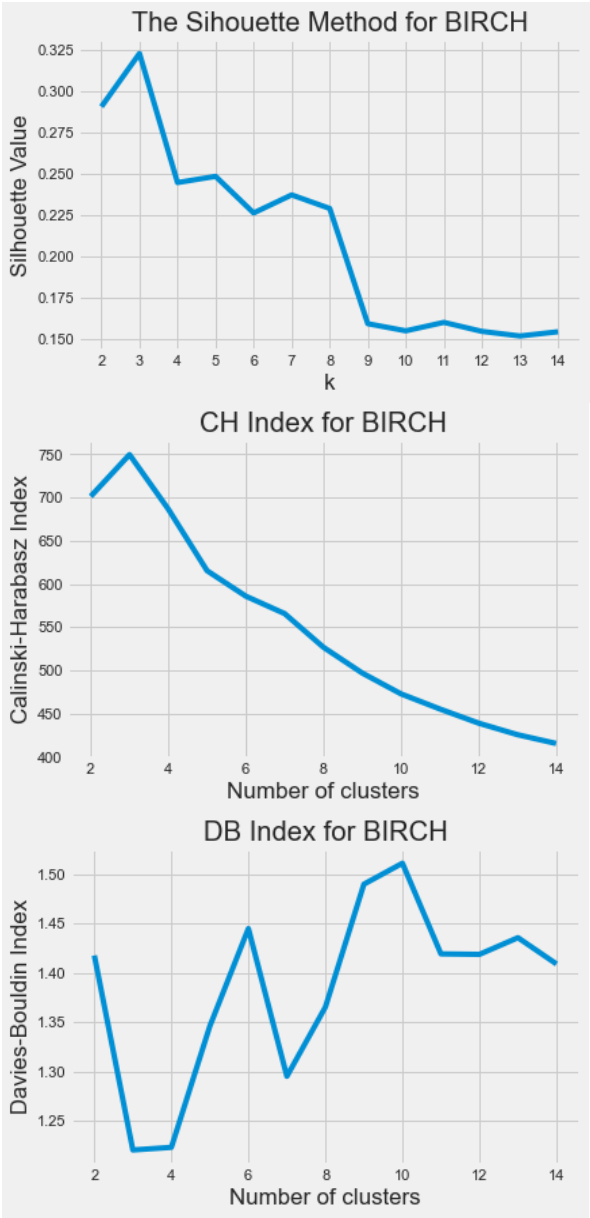
B1.7





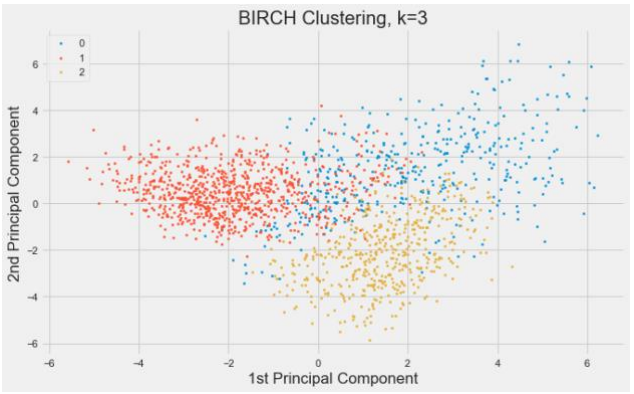
Plots for Spectral Clustering clusters, k=3 and k=4. More crossover in points, favoring smaller clusters in the upper right corner.

B1.8



Metrics for BIRCH. Worst performance but k=3 chosen again

B1.9



Much more crossover in clusters, probably due to lower performance.