Havish Chennamraj
903201642

CSE 7641 – Assignment 3 Report

1 Abstract

The report explores 2 unsupervised learning algorithms namely 1.) K-Means and 2.) Expectation Maximization and 4 dimensionality reduction Techniques: Principal Component Analysis, Independent Component Analysis, Random Projections and Factor Analysis. It also explores how the dimensionality reduction techniques affect Supervised Learning techniques like Neural Networks w.r.t to performance, time etc.

2 Dataset

The same datasets from Assignment 1 have been used. They are provided by Kaggle and described below.

2.1 Stay Alert! dataset from the The Ford Challenge [1]

The dataset consists of results of a number of "trials", each one representing about 2 minutes of sequential data that are recorded every 100 ms during a driving session on the road or in a driving simulator. The trials are samples from some 100 drivers of both genders, and of different ages and ethnic backgrounds. Every sample in the dataset has 30 features which are continuous values representing physiological, environmental and vehicular data. It's also got a binary variable "IsAlert" around which the classification problem is built. The training dataset has around 600k records and the test dataset has around 121k records to work with.

2.2 Otto dataset from Otto Group Product Classification Challenge [2]

The dataset is a collection of transactional data of the Otto Group, one of the world's biggest e-commerce companies. It has around 93 features for about 200,000 products. Each row in the dataset corresponds to a single product and the 93 features which represent count of different events. The feature set is a mixture of binary and continuous valued variables. Additionally, the dataset has a "target" feature which is of particular interest to our classification problem, and has 9 different values to it each corresponding to a particular product category in the company like fashion, electronics etc.

Please note that both datasets are very large and in the interest of time they've been heavily subsampled for all of the experiments.

3 Experiments and Analysis

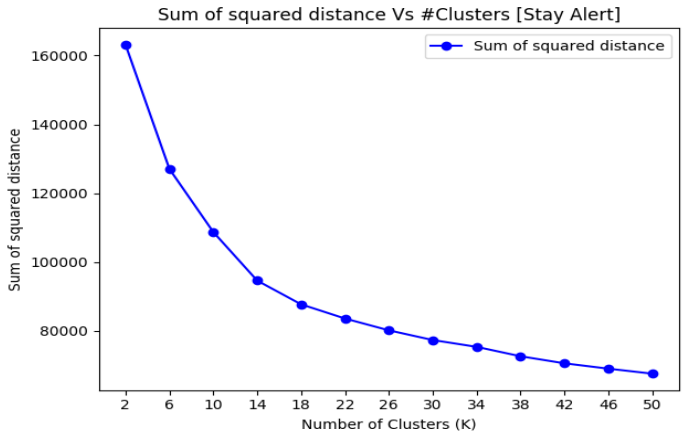
3.1 Clustering without dimensionality reduction

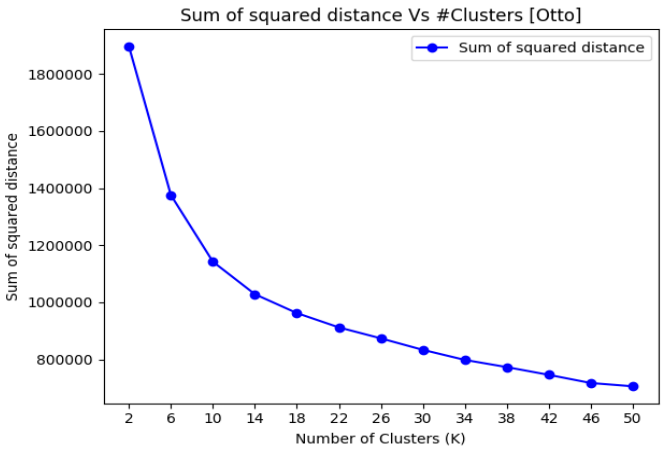
3.1.1 Clustering without dimensionality reduction: K-Means

K-Means is an iterative unsupervised clustering algorithm which tries to partition the given data into k clusters or subgroups where k is defined by the user such that every sample strictly belongs to **only one of the groups.** K-Means assigns data-points to clusters with the least distance from the cluster centroid (arithmetic mean of all data points in the cluster) to the data point itself. The algorithm is briefly summarized as follows:

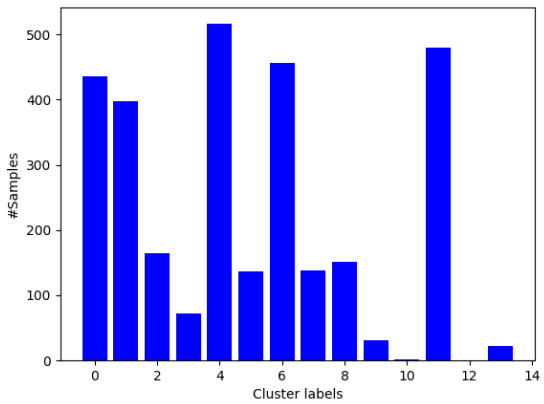
1.) Define the number of clusters 'K'
2.) Initialize the first K cluster centroids by choosing 'K' data samples at random without replacement.
3.) For every point, compute its distance from all of the 'K' cluster centroids and assign it to the cluster with the least distance. Note that the distance function is also chosen by the user.
4.) Recompute the cluster centroids which is basically the arithmetic mean of the points across all dimensions.
5.) Repeat steps 3 & 4 until convergence, in other words until the cluster centroids do not change.

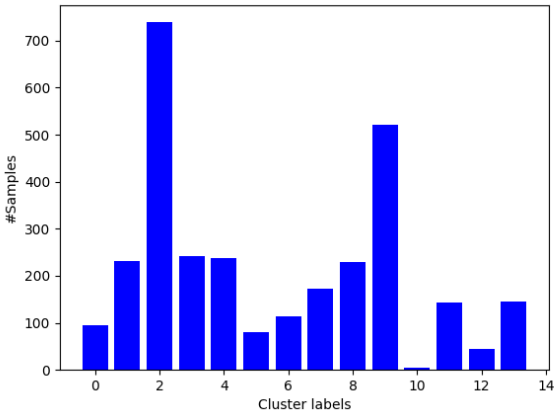
The **elbow method** is used to pick the best value of 'K' based on the sum of squared distances (SSE) between data points and their allocated clusters' centroids. I've chosen **Euclidean distance** because it is fairly simple and gave the best results for K-NN classifier in Assignment 1 indicating that Euclidean distance does a good job in modelling distance between data samples in both datasets. As the name 'elbow' suggests, the idea of the method is to choose k at a point where the SSE stops decreasing sharply and tends to flatten out forming an elbow. The graphs below show the SSE values for varying K.





As it can be observed from the above graphs, there is not a distinctive elbow formation in both the graphs. However, since SSE decreased by the same amount from '2-14' as well as '14-50' and tends to flatten out for higher values of K in both the graphs, it can be approximated that 14 is the best value of 'K' for both the datasets. Below plots show the distribution of samples across the 14 clusters for both datasets.





As we can notice the distribution of the samples across clusters is also un-even with some clusters having more samples than others and nothing much can be inferred about the performance of K-Means w.r.t both datasets from this.

Dataset	**Silhouette Score**
Ford Stay Alert! Dataset	0.12371168898546159
Otto Dataset	0.17184999738157838

Silhouette Coefficient Score [3] was then used to evaluate the performance of K-Means. Silhouette Score is the measure of how similar a given sample is with its assigned cluster as compared to other clusters. For a given sample, the score is defined as follows:

Silhouette Score (S) = $(b^i - a^i) / max(b^i, a^i)$ where,
a^i - Average distance from all data points in the same cluster
b^i – Average distance from all data points in the closest cluster.

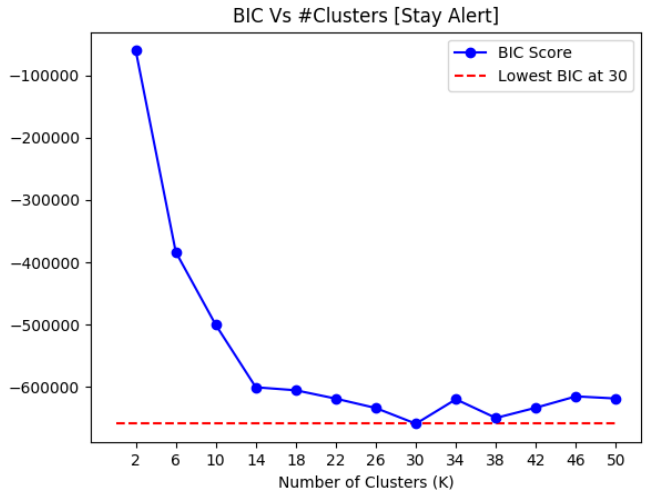
Like mentioned above, the score is defined for every sample and mean Silhouette Score across all samples was used for the evaluation. The coefficient can take values from -1 to +1; value close to 0 indicates that the samples are very close to neighboring clusters, close to -1 indicate that the samples are assigned to the wrong clusters and values close to +1 indicate that samples are pretty far off from the neighboring clusters. Obviously, higher the value of mean Silhouette Coefficient, the better the performance of the K-Means. The Silhouette scores for both the datasets are reported in the table above. Values of 0.124 and 0.172 for Ford and Otto datasets respectively indicate that most of the data samples are pretty close to boundaries of neighboring clusters and thus the clusters aren't very well separated. *The lack of a distinctive elbow formation coupled with very mediocre Mean Silhouette Coefficient suggests that K-Means is not a good clustering algorithm for both the datasets. This might be because of K-Means getting stuck at a local optimum or perhaps outliers in data. It might also be possible that the data cannot be partitioned into distinctive spherical clusters which K-Means strives for inherently.*

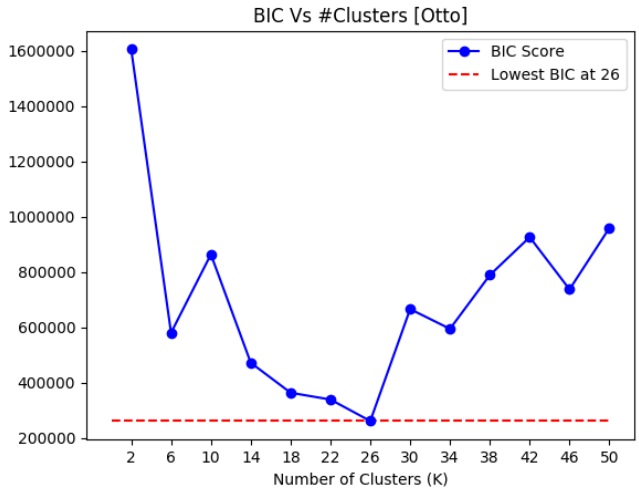
We then use Adjusted Mutual Information (AMI) [3] to measure how well the cluster assignments of the data samples align with true assignment, assuming the true assignments are the corresponding class labels. AMI is bounded between 0 and 1, and higher the value, the better is degree of agreement between the assignments. AMI scores came out to be 0.26 for Ford Dataset and 0.05 for Otto. Again, mediocre performances at best. However, this is interesting for the following reason: AMI scores will be higher if the data points are linearly separable with a bunch of hyperplanes, and it was observed from the first assignment that SVM achieved an accuracy of ~84% on Ford dataset as compared to ~75% for Otto dataset thus making our results here of having a higher AMI score for Ford dataset is consistent with our observation.

It is also important to note that Euclidean distances may become inflated in high-dimensional domains and thus dimensionality reduction techniques like PCA and ICA might provide us with better results. This shall be explored in the later sections of the report.

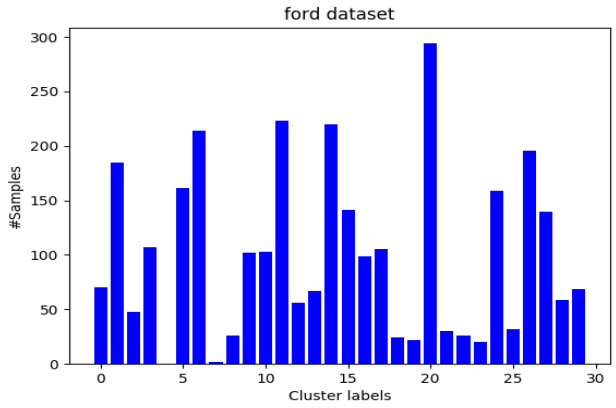
3.1.2 Clustering without dimensionality reduction: Expectation Maximization.

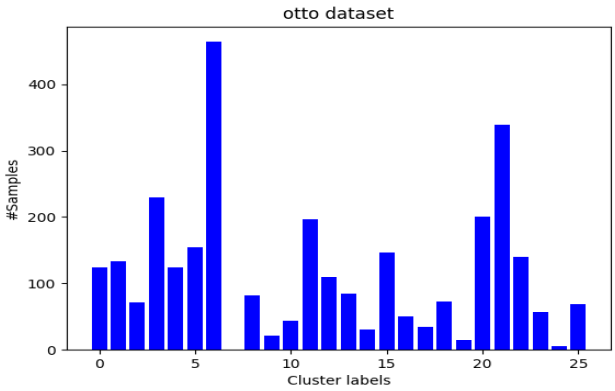
Expectation Maximization is an algorithm that finds 'K' distributions of data such that the likelihood (LL) of data given distributions is maximized. EM alternates between Estimation step which involves calculating the probability that a point belongs to a cluster (which happens to be a straight 0 or 1 in the case of K-Means) and Maximization step which involves computing the new distributions a.k.a means of the Gaussian distributions using the Maximum Likelihood Estimate. The goal of EM clustering is to estimate the means and standard deviations for each cluster so as to maximize the likelihood of the observed data (distribution). Gaussian Mixture Models (GMM) from 'sklearn' library have been used to implement this 'soft' clustering on both the datasets. Unlike in K-Means, Silhouette scores cannot be used here to evaluate performance because they work well only for Spherical Clusters and EM has no such spherical bias towards the clusters it generates. Instead, the Bayesian Information Criterion (BIC) was used to find the best 'K', lower BIC values being better. Below plots show the values of BIC for varying number of clusters.





As it can observed from the above graphs, the points of minima are K=30 and 26 for Ford and Otto datasets respectively which means these are our final values for number of components on which the EM algorithm should run for the corresponding datasets. The distribution of data samples across clusters is uneven just like K-Means thus stopping us from drawing any meaningful inferences from the plots below.



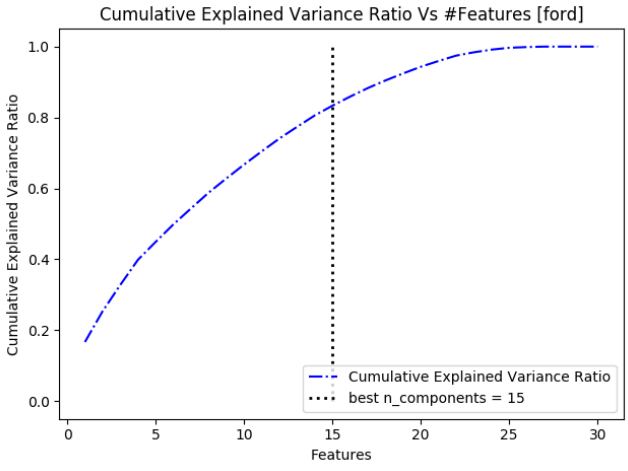


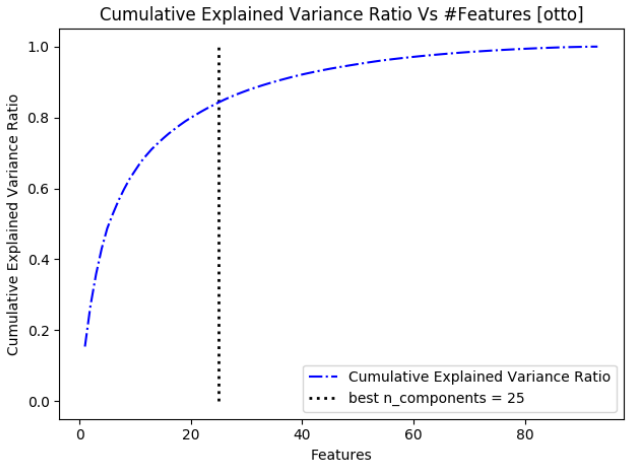
Again, to evaluate how well the cluster labels from EM align with the ground truth labels which are the corresponding class labels of the data samples, we used Adjusted Mutual Information (AMI). AMI score for Ford came out to be 0.258 and 0.04 for Otto which is very similar to the scores from K-Means. The only inference I can make from this is that data samples in the Ford dataset are more separable using hyper-planes compared to Otto dataset. However, both the scores are mediocre which suggests that it is difficult to cluster both datasets using EM just like K-Means. However, eliminating noisy features and transforming the feature set using dimensionality reduction technique might make the dataset more separable and cluster-able. We shall evaluate this in the later sections of the report.

3.2 Dimensionality Reduction

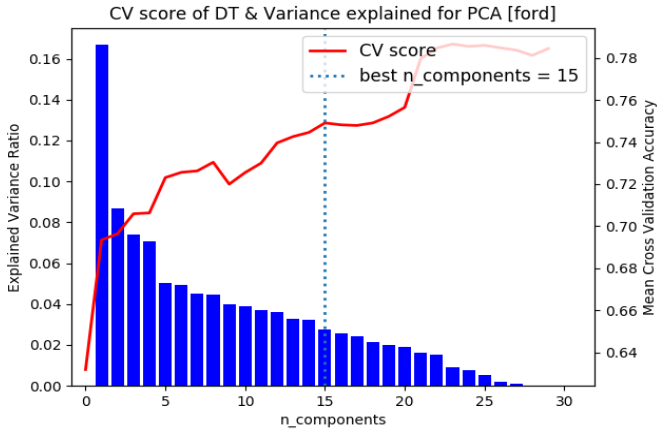
3.2.1 Principal Component Analysis (PCA)

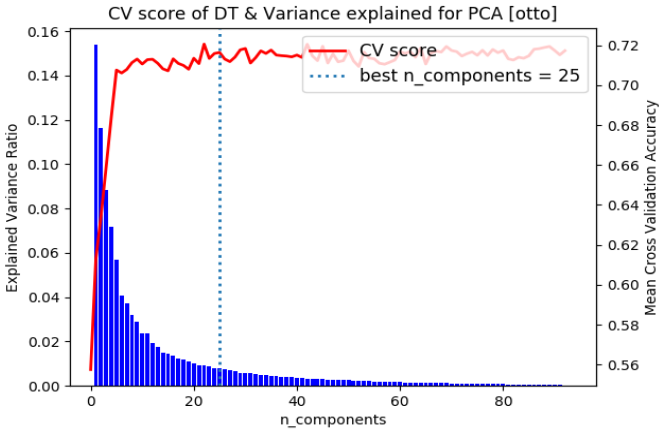
Principal Component Analysis finds basis vectors that 'best explain' the variance in the data with the first (highest ranked) basis vector (1st PC) best fitting the variance in the data. Following PCs have the same criteria but are orthogonal to each other. The proportion of total variance in the data explained by each component is the corresponding eigenvalue of the covariance matrix. Total Cumulative Variance explained is plotted for increasing number of features.





As we can notice, the total variance explained rises rapidly by including the first few features. A good rule of thumb is to select the best number of features is to select *best 'n_components'* such that it accounts for atleast 85% of the total variance since we don't have an elbow in the plots. Following that rule gives us 15 and 25 as *best 'n_components'* for Ford and Otto dataset respectively. Now having finalized on the best number of features, let's see how the PCA transformed feature set performs w.r.t to a Decision Tree Classifier on both datasets.



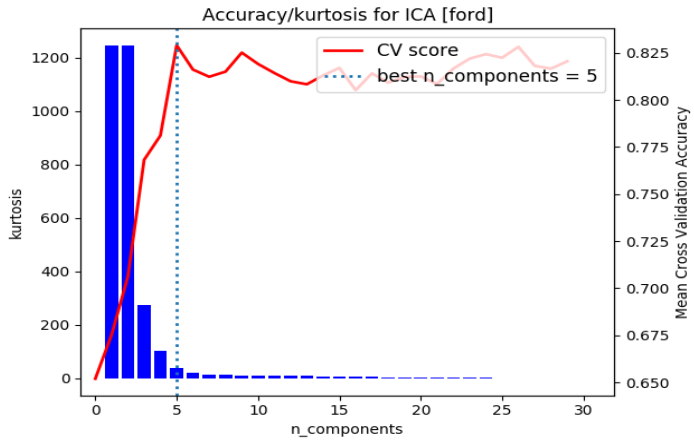


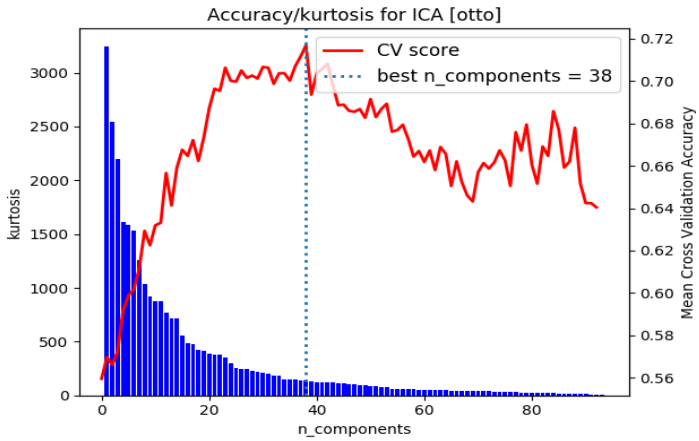
As we can notice, the choice of 'best n_components' seems to strike a healthy balance in the Ford Dataset: PCA managed to remove most of the irrelevant features whilst not compromising on the performance of the classifier. We can still squeeze out a little more performance (from ~0.74 to ~0.78) for the classifier by including features (16-22) at the cost of slightly higher training times. This is probably because of outliers in the dataset where the decision to arrive at a given class label also depends on relatively unimportant features which most of the data samples do not care about. It should be noted that PCA's performance for Otto dataset is commendable as it not only managed to shave off around 75% of the feature set but also achieved the full potential of the classifier with the reduced feature set as we can notice from the plot above.

Clearly, the nature of the classification problem plays an important role in the evaluation of Dimensionality Reduction Techniques. If for a given classification task, 'global properties' are more important, they are more easily extracted by PCA than ICA. However, if the classification problem relies more on spatially local localized features such as time-series etc. ICA would a better job. For the Otto classification problem, I believe it relies heavily on global features like price of the product, category, weight to classify products which explains the good performance of PCA.

3.2.2 Independent Component Analysis (ICA)

ICA tries to reduce a multivariate feature set into independent components by maximizing the statistical independence between the said components. ICA inherently assumes that the multivariate feature set is a collection of statistically independent and non-gaussian components and hence achieves the transformation by trying to maximize Kurtosis. Kurtosis is the measure of non-gaussianity. It measures the 'spikiness' of the distribution and is 0 for a gaussian distribution; positive values indicate a 'spikier' distribution than gaussian and negative values indicate a 'flatter' distribution than gaussian. 'FastICA' from sklearn library was used for this purpose. The independent components are sorted by kurtosis values from highest to lowest and CV score of Decision Tree Classifier is also plotted alongside to determine the 'best n_components'. We choose the number of components by following the elbow rule with respect to the CV score of the Decision Tree Classifier.



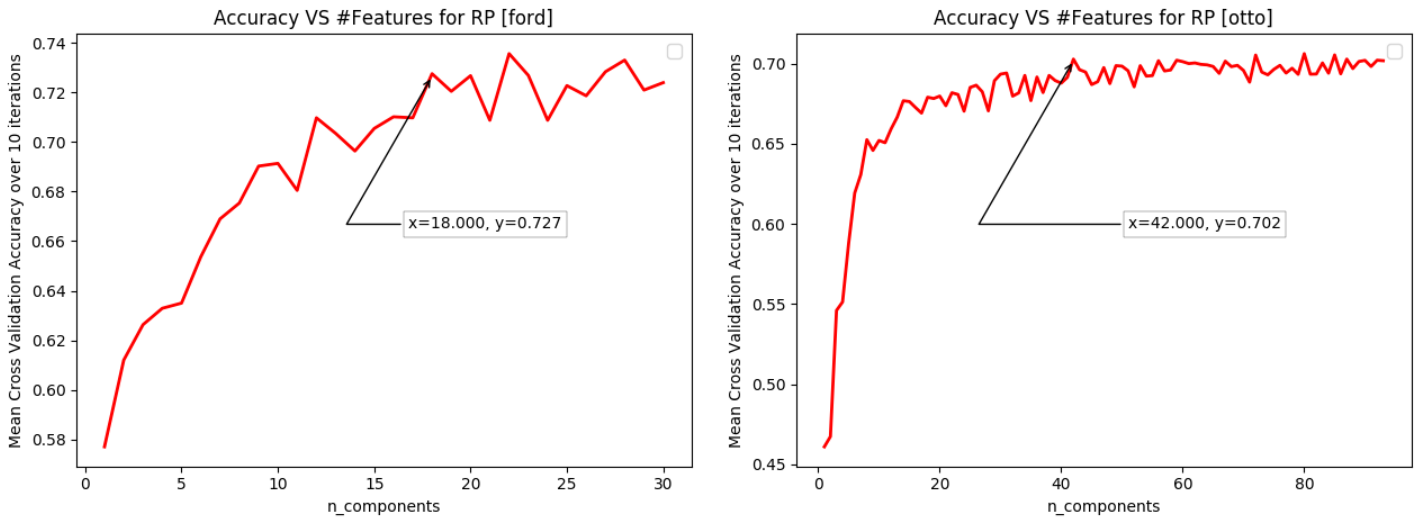


As we can notice, the CV score of the classifier tends to flatten out at ~81% beyond the top 5 features for Ford dataset and achieves the best performance for n_components=5. The behavior is slightly different for the Otto dataset; as we can notice the plot for CV score of the classifier against the number of features takes a 'hill' like shape. The classifier achieves its full potential of ~72% accuracy at 38 features, adding features beyond this seems to hamper the performance of the classifier. It is probably because these features constitute noise and incorporating them into the model only adds to the confusion of the classifier. It might also be a classic case of overfitting; adding more features makes the model highly sensitive to the training data, thereby hampering its ability to generalize on the validation/test set.

Like mentioned above in the PCA section, PCA tends to perform better when the classification task relies more on global features and ICA performs better if the features are spatially localized and the 'Ford' dataset classification problem happens to be of the latter type. The data samples for the Ford dataset are the results of a number of "trials", each one representing about 2 minutes of sequential data that are recorded every 100 ms during a driving session on the road or in a driving simulator and since it being a time-series dataset, there's a degree of locality in the data samples with respect to time which explains why ICA performs so well for "Ford" dataset.

3.2.3 Random Projection (RP)

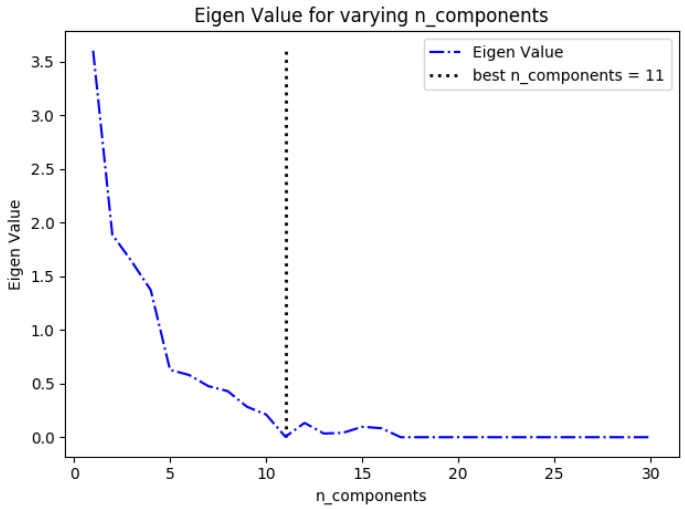
Random projection (RP) achieves dimensionality reduction by projecting data samples from a n-dimensional space to a k-dimensional subspace such that k << n. It is a simple and computationally cheap way to reduce dimensions trading controlled error for faster computation and smaller models. Obviously, the reduction technique being inherently random, Cross Validation Scores of the Decision Tree Classifier were averaged across 10 iterations of the same experiment. Below are the plots for the same. The elbow method was used to find the best n_components for RP on the plots where the CV score the DT Classifier was plotted against the top 'n' features (n_components) produced by Random Projections.

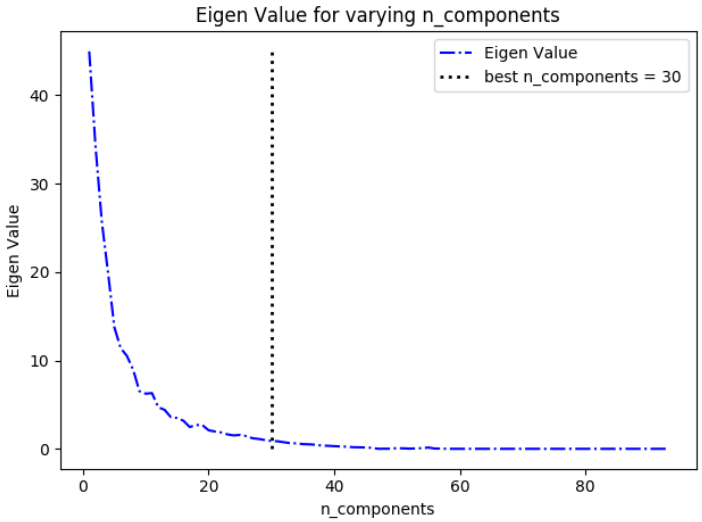


The best 'n_components' turned out to be 18 and 42 for Ford and Otto datasets respectively. We can clearly see that beyond the best 'n_components', CV score of the classifier tends to flatten out in both cases thus leaving us with no incentive in broadening our feature set.

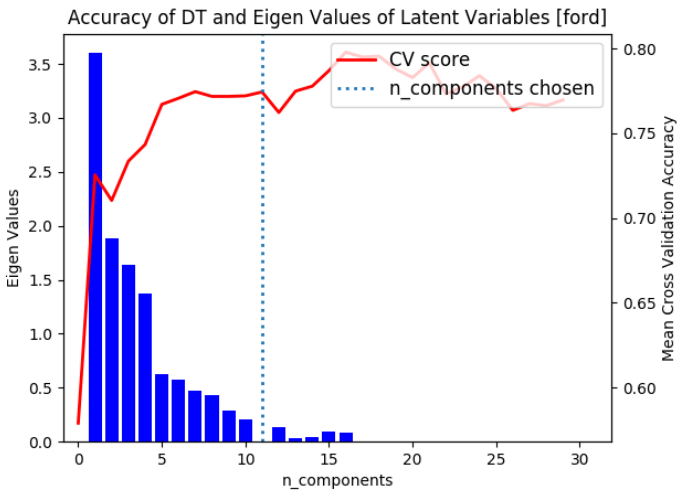
3.2.4 Factor Analysis (FA)

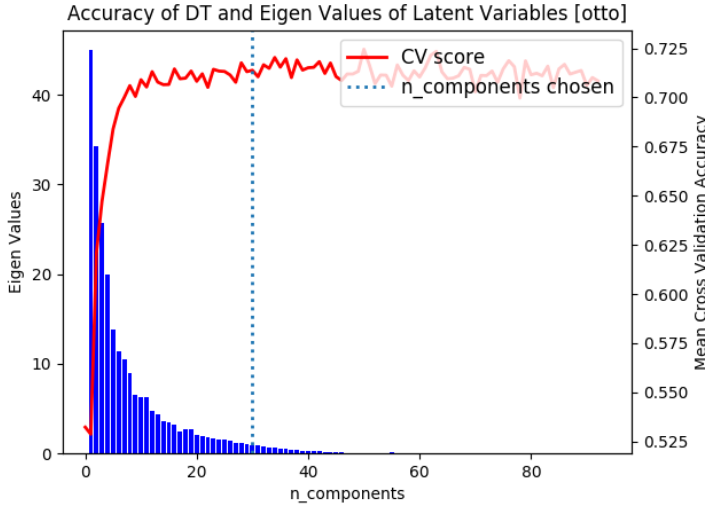
Factor Analysis[5] is quite similar to PCA. The objective of FA is to find intercorrelations among 'n' features by coming up with set a common factors 'm' such that m << n. The factors are random variables that cannot be observed, counted or measured directly, but which are presumed to exist in the population and hence they are in the experimental sample and hence sometimes referred to as latent variables. In order to find the best 'n_components' in FA we need to figure out the best 'n' hidden variables that influence the behavior of the samples in the dataset which basically boils down to a matrix reduction problem just like PCA. To do this, we plot the eigen values for all the hidden variables in descending order from left to right and use the elbow method to determine the best 'n_components'.





There is a distinctive elbow formation in both the plots above at n_components= 11 & 30 for Ford and Otto datasets respectively. Essentially, any of the latent variables beyond the best 'n_components' is technically pointless because they don't capture the interrelations between data samples as the algorithm intends to, considering their eigen values are zeroes.



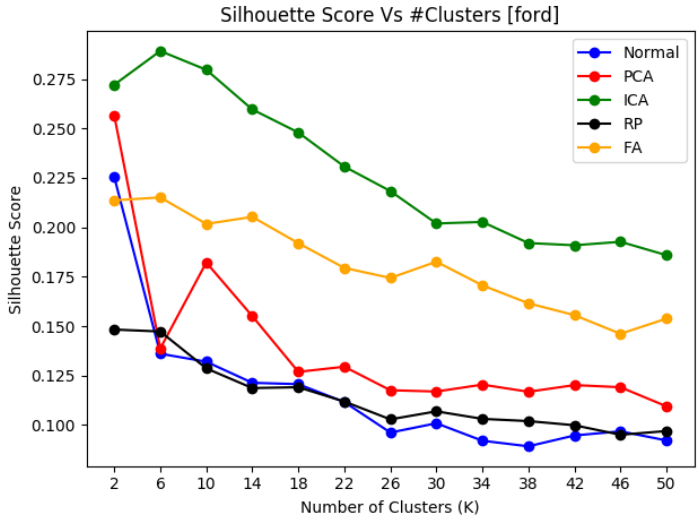


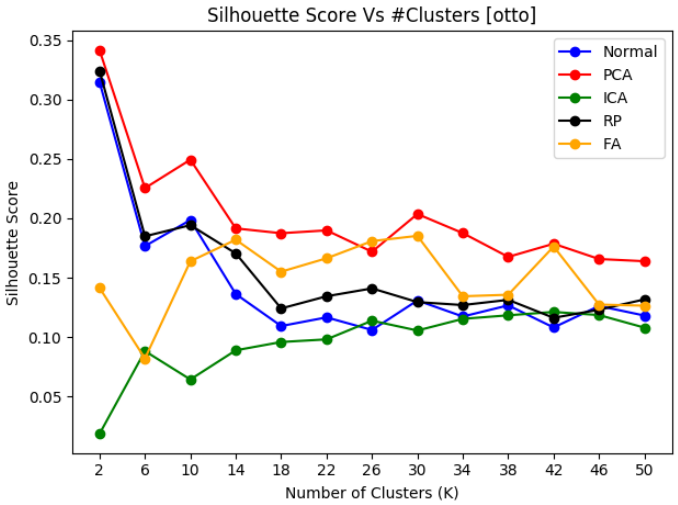
Our choice of best n_components seems to agree with respect to the performance of DT classifier almost achieving the best CV score for the classifier while significantly pruning the dimension space for both the datasets.

3.3 Clustering with dimensionality Reduction

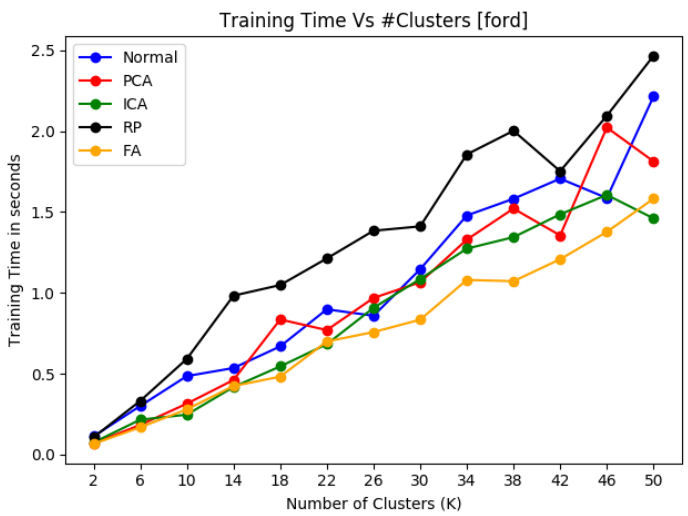
3.3.1 Clustering with dimensionality Reduction: K-Means

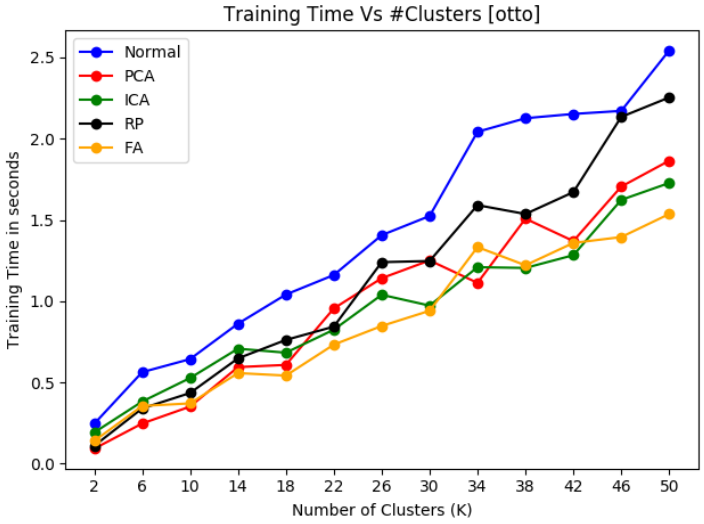
To evaluate the performance of the K-Means on the feature reduced datasets, I chose Silhouette score. Scores like Adjusted Rand Index, completeness, homogeneity etc. require ground truth labels for calculation which we don't have since our datasets were not geared towards that in the first place. Unless, someone manually annotates the data samples it is actually difficult to obtain these ground truth labels. Since Silhouette score has no such prerequisite, it is ideal to estimate the performance of K-Means in this case. Silhouette Score is the measure of how similar a given sample is with its assigned cluster as compared to other clusters and a higher Silhouette score relates to a model with better defined clusters.





As we can notice, ICA outperforms the rest in Ford dataset while PCA does the same in Otto dataset. We've already explored the reasons as to why ICA and PCA are the best suited Dimensionality Reduction techniques for Ford and Otto respectively. I suspected that since Ford is composed of time-sequential data recorded every 100ms which is spatially localized and composed of physiological, environmental and vehicular data which are relatively independent of each other, ICA is best suited algorithm. While on the other hand Otto classification problem being about classifying products on an organization level depends on global features like price, weight etc. which PCA is best suited for and the plots above agree with this conjecture. Applying the elbow technique for the above plots gives us K=14 as the ideal number of clusters for K-Means on Otto dataset for all the algorithms. However, there doesn't seem to a distinctive elbow formation for Ford dataset but we can approximate the ideal number of clusters to 26 as all lines tend to flatten out from 26.

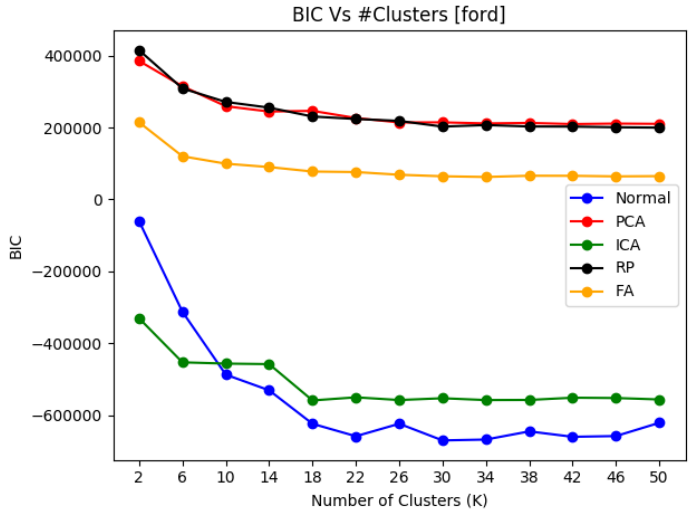


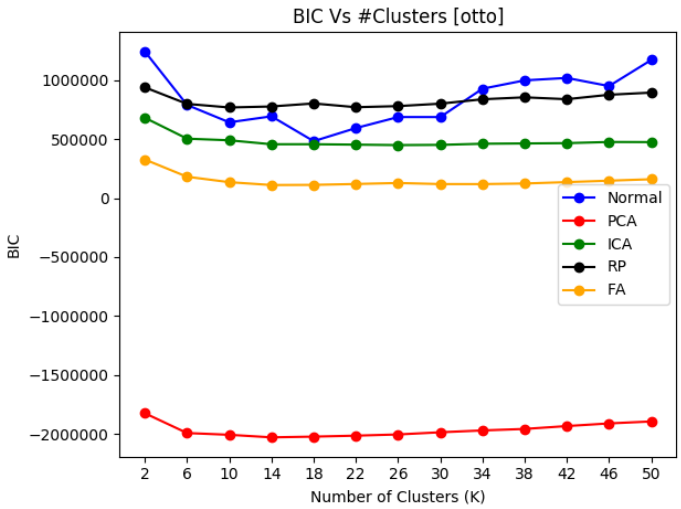


The graphs plot training time of K-Means against the value of K for normal as well as transformed datasets. No surprises here, we can see the time increases with the value of K as it takes more time reach convergence. Also, the dimensionally reduced datasets have lesser training time than the original space which is also expected.

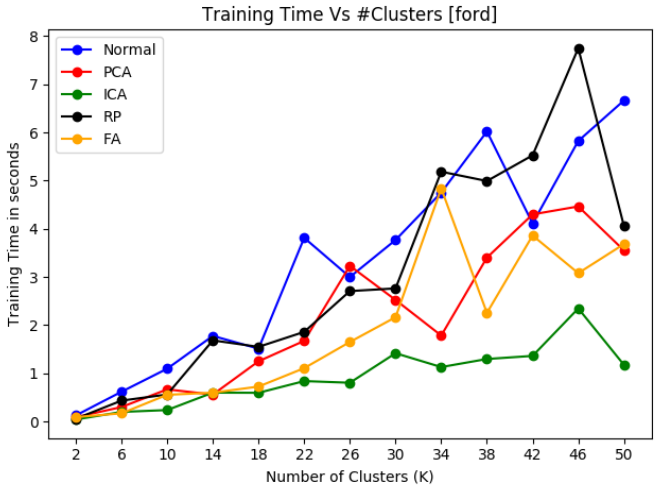
3.3.2 Clustering with Dimensionality Reduction: Expectation Maximization (EM)

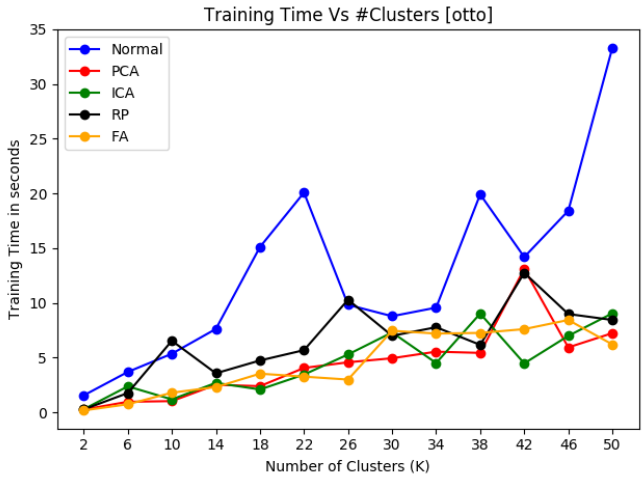
I've chosen BIC to evaluate the performance of EM algorithm for all the transformed and original dataset. BIC is a criterion for model selection among a finite set of models. It is based, in part, on the likelihood function, and helps in preventing overfitting by adding penalty for every parameter added to the model. Note that lower BIC scores are better, and we choose k as the one which gives the least BIC score.





We can see that the behavior of the feature transformed datasets and the original one is the same for Otto dataset and almost similar in Ford. The plots tend to collectively agree on K=18 being the best choice of K for Otto dataset, this means that even after feature transformation and reduction BIC suggests that 18 clusters have a normal distribution that matches the underlying assumptions of GMM and the model generated from PCA transformed data clearly outperforms the rest with a substantially lower BIC score. For Ford dataset, there seems to be a reasonable amount of agreement on the value of K among PCA, ICA, RP and FA with it being around 18, while on the original dataset the optimal value of K is 30. Also, the model generated from the original dataset seems to be the best one with ICA coming as a close second for Ford.





Just like in the case of K-Means, no surprises here. The training time tends to increase with the value of K and EM seems to take longer to train on the original dataset as compared to the reduced datasets of PCA, ICA, RP & FA because of the difference in the scope of the dimension space.

4 Dimensionality Reduction and Neural Networks

In this experiment we choose Otto dataset to train neural networks for all the dimensionality reduction algorithms because in the assignment on supervised learning, NN almost performed the best. Although it would have been ideal to find the ideal hyperparameters like #neurons, #layers, activation function etc. for all of the transformed datasets and then evaluated the performance of models, in the interest of time and space in the report, we train all the neural networks on the same hyperparameters we deemed best for the original Otto dataset in assignment 1 which happen to be: #Neurons=7, #Layers=1, #Iterations=200 and Activation function='relu'. Following are the training times and training and testing accuracies for each of the datasets.

Dataset	Training Time in seconds	Training Accuracy	Testing Accuracy
Original	14.2777791023254395	0.8641428571428571	0.7746666666666666
PCA (n_components = 25)	10.294281005859375	0.7901428571428571	0.7643333333333333
ICA (n_components = 38)	11.333997106552124	0.7325714285714285	0.7206666666666667
RP (n_components = 42)	11.3116209506988525	0.7974285714285714	0.7583333333333333
FA (n_components = 30)	11.029448986053467	0.7904285714285715	0.756

The training time of the transformed datasets is obviously lower because of the reduced dimension space allowing the NN to reach convergence faster. Al the reduction techniques perform almost as good as the original dataset which suggests that the projected data in all of the cases still has enough discriminative features which the NN is able to find. Among the dimensionality reduction algorithms, PCA performs the best. This is because among all dimensionality reduction algorithms, PCA gives the best low-rank approximation of the data. Thus, information loss is minimal and the neural network still works well.

5 Dimensionality Reduction with clustering algorithms and Neural Networks

	K-Means			Expectation Maximization		
Dataset	Training Time in seconds	Training Accuracy	Testing Accuracy	Training Time in seconds	Training Accuracy	Testing Accuracy
Original	14.368	0.866	0.776	14.214	0.857	0.773
PCA (n_components = 25)	10.954	0.799	0.770	10.940	0.791	0.765
ICA (n_components = 38)	11.218	0.729	0.71	11.021	0.717	0.708
RP (n_components = 42)	11.336	0.787	0.755	11.239	0.790	0.754
FA (n_components = 30)	11.204	0.788	0.752	11.014	0.796	0.754

This experiment is to evaluate the effect adding cluster labels to the feature set on the performance of NN. Just like the above experiment, although not ideal, we adopt the same hyperparameters that we found in Assignment 1 for NN. Also, the value of k for clustering algorithms on the transformed datasets is set in accordance with our findings in 3.3.1 and 3.3.2. We can notice that, there is a very slight bump in the performance of NN on adding cluster labels from K-Means and EM to the PCA transformed dataset while the others remain fairly constant. I was inclined to believe that nature of the dataset which can be characterized by the cluster labels should matter and consequently result in better classification. However, save a minor bump for PCA it doesn't appear that cluster labels provide a major boost in anyway. Although our transformations and reductions do not seem to have any significant impact here, it is not necessarily indicative of these algorithms and techniques having the little overall effect in general.

6 References

[1] https://www.kaggle.com/c/stayalert/data
[2] https://www.kaggle.com/c/otto-group-product-classification-challenge/data
[3] https://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient
[4] http://fourier.eng.hmc.edu/e161/lectures/ica/node4.html
[5] https://research-repository.griffith.edu.au/bitstream/handle/10072/366058/02Main.pdf?sequence=1