Assignment 3: CS7641 - Machine Learning Yogesh Edekar(GTUID – yedekar3)

March 27, 2021

1. Introduction

As part of assignment 3 I am trying to analyze the applications of supervised learning algorithms to find out similarities between the features using the clustering algorithms of K means clustering and expectation maximization to determine the clusters in the chosen datasets. Then I apply the dimensionality reduction algorithms Principal Component Analysis (PCA), Independent Component Analysis (ICA), Randomized Projections and Univariate Feature Selection (UVFS) for feature selection.

2 Data Sets

The data sets used for analyzing the supervised algorithms are a) Term deposit dataset with 17 features identifying a successful term deposit at downloaded from Kaggle and b) income evaluation dataset with 14 features that are impacting the income of a person also downloaded from Kaggle.

2.1 Term deposit dataset

The term deposit data set consists of 5873 people of the selected population who are not opening the term deposit account at the bank and 5289 people of the selected population that are opening the term deposit account at the bank. Apparently the distribution of labels seem pretty even for the dataset and hence the expectation is we will have well defined clusters easily. We will validate our assumption during our clustering analysis.

2.2 Income evaluation dataset

Income evaluation dataset consists of a total of 32562 population count where based on various features the income of an individual being greater than 50,000 or less than or equal to 50,000 is determined. However we see certain disparity in this dataset where 76% of the records fall in the <= 50,000 category and remaining fall in the < 50,000 category. Finding clusters in this dataset due to the disparity of the records might be difficult. We will validate this assumption during our clustering analysis.

3 Clustering

Clustering is the process of grouping the instances into group so that the instances into a group are more similar to each other than the instances that belong to the other group. The two clustering algorithms that are analyzed as part of this assignment are K Means clustering and Expectation Maximization (EM).

3.1 K-Means clustering

3.1.1 Algorithm

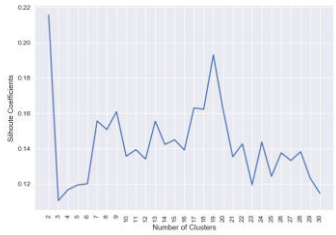
We take k centroids at random and try to partition the data points into k clusters around the chosen centroids using a distance metric such as Euclidian distance, Manhattan distance etc. Based on the clusters formed we revaluate the centers to see if we can get any better centers such that the points in the cluster are more similar to each other. We repeat this process until we find better centroids than current ones. The algorithm converges when we can't find any better centroids and we have obtained k clusters.

3.1.2 Dataset analysis

The KMeans class of the sklearn.cluster package is used for performing the K-means clustering on deposit and income dataset. The Silhouette score has been used to identify the quality of clustering. The observations are as follows.

3.1.2.1 Term Deposit Dataset

The plots to determine the k value for term deposit dataset are given below



Silhouette Score

Adjusted rand Score

The number of clusters given by Silhouette score were then compared with the actual labels for validations using the adjusted rand score to identify the appropriate value of k that gives an optimal rand score and based on the rand score provided the value of k (number of clusters) is determined to be 2. The readings for k and corresponding rand scores are given in the following table 1.

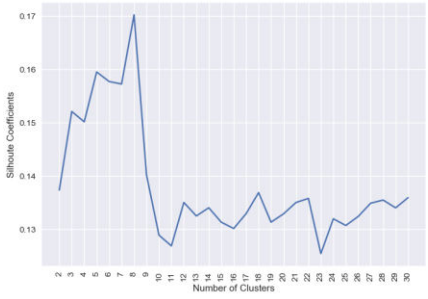
Method	Silhouette Score
Number of clusters (k)	2
Adjusted Rand Score	0.050

Table 1 : k and adjusted rand score for term deposit dataset

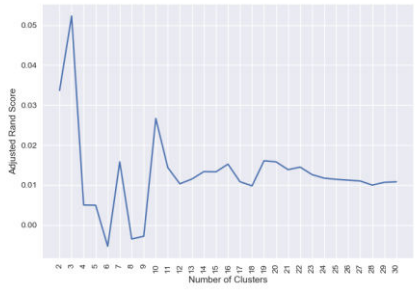
We will be using a **cluster count of 2** for the Term deposit dataset.

3.1.2.1 Income Evaluation Dataset

we get a value of k = 8 using the Silhouette score for the income evaluation dataset and when we check the adjusted rand score for the same value the optimal k is found out to be 3. Since for k the adjusted rand score seems negative we will take the higher value indicated by adjusted rand score **k = 3**



Silhoutte Score (k = 8)



Adjusted rand Score (k=3)

3.2 Expectation Maximization

3.2.1 Algorithm

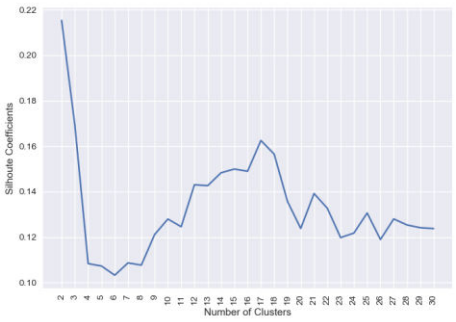
Expectation maximization uses probability distributions. It works in estimation phase and maximization phase. The estimation phase estimates missing variables in the dataset and the maximization phase maximizes the parameters in presence of data. This process is repeated iterative until we get the desired number of clusters or data cannot be further maximized.

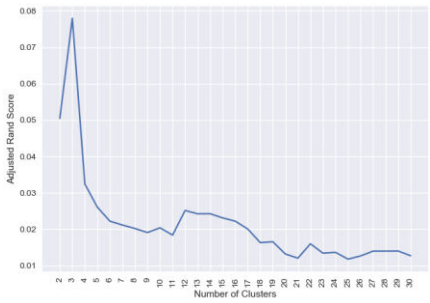
3.2.2 Dataset Analysis

The *GaussianMixture mixture* class of the sklearn.mixture package is used for performing the expectation maximization using gaussian mixture on deposit and income dataset. The Silhouette score has been used to identify the quality of clustering. The observations are as follows.

3.2.2.1 Term Deposit Dataset

The plots used to determine the value of k for term deposit dataset are given as below.





Silhouette Score *Adjusted Rand Sc*

The number of clusters given by Silhouette score were then compared with the actual labels for validations using the adjusted rand score to identify the appropriate value of k that gives an optimal rand score and based on the rand score provided the value of k (number of clusters) is determined to be 3. The readings for k and corresponding rand scores are given in the following table 2.

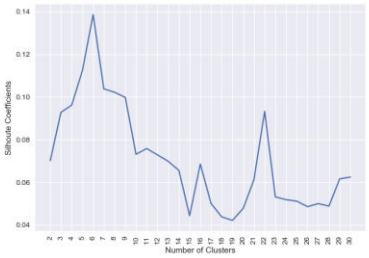
Method	Silhouette Score	Silhouette Score
Number of clusters (k)	2	3
Adjusted Rand Score	0.050	0.08

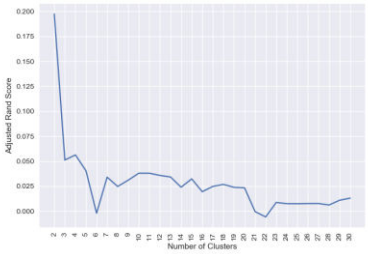
Table 2 : k and adjusted rand score for term deposit dataset using EM

We will be using a **cluster count of 3** for the Term deposit dataset for expectation maximization due to high adjusted rand score for the value of k.

3.2.2.2 Income Evaluation Dataset

The plots used for determining the value of k for income evaluation dataset are as follows





Silhouette Score Adjusted Rand Score

The number of clusters given by Silhouette score were then compared with the actual labels for validations using the adjusted rand score to identify the appropriate value of k that gives an optimal rand score and based on the rand score provided the value of k (number of clusters) is determined to be 2. The readings for k and corresponding rand scores are given in the following table 2.

Method	Silhouette Score	Silhouette Score
Number of clusters (k)	6	2
Adjusted Rand Score	0	0.2

We will be using a **cluster count of 2** for the income dataset for expectation maximization due to the fact that adjusted rand score is higher for k = 3 with a negative rand score eliminating the k count to be 6.

4 Dimensionality Reduction

The number of input variables in a dataset is referred to as it's dimensionality. Dimensionality reduction technique is used to reduce the number of input variables in a dataset to over the so called Curse of Dimensionality. With more dimensions our predictions can become accurate but at the same time we need more and more data requiring us to add to the computation time.

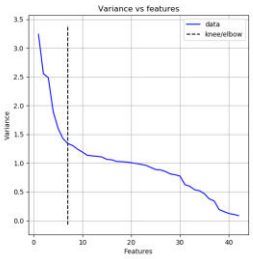
4.1 Principal Component Analysis (PCA)

4.1.1 Algorithm

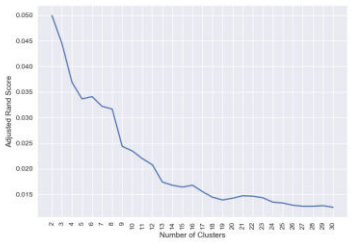
The major target of PCA is to reduce the dimensionality of the features of a dataset but by persisting most of the information in the dataset. When we reduce dimensions a little accuracy will be lost with the data however if we retain the dimensions that increase or decrease together i.e. the dimensions that have positive co-variance then we get the so called Principal Components from the given features. Thus PCA tries to maximize projection such that maximum co-variance is obtained in the first few projections so that we can choose these projections for our dataset.

4.1.1.1 Term Deposit dataset

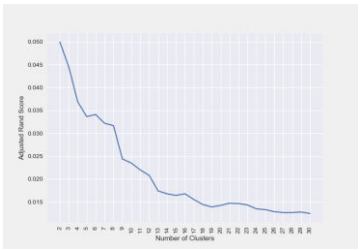
Given below are the graphs obtained for PCA dimensionality reduction over the term deposit dataset and the results of applying PCA for clustering



PCA count = 7



KMeans rand score



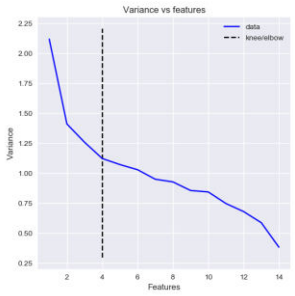
EM rand score

After obtaining different covariance values with PCA we use the elbow method to determine the count of transformed dimensions to be used for PCA. The dimensions to be used come out to be 7 for the term deposit dataset. We can see that with 7 dimension we get a rand score of 0.05 for k means. Below table captures all the data for PCA transformed dimensions for income analysis dataset using clustering

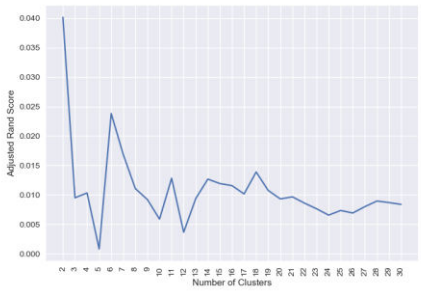
Metric	PCA	KMeans	EM
Rand Score	0.05	0.05	0.08
Run time	0.019	0.2	0.2
Cluster count	2	2	3

4.1.1.1 income Analysis dataset

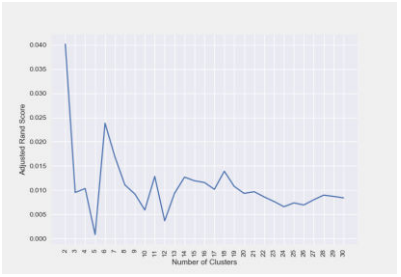
Given below are the graphs obtained by applying PCA over the income analysis dataset and the results obtained by applying PCA over the dimensions and then performing clustering over the transformed dataset.



PCA components = 4



KMeans rand score



EM rand score

After obtaining different covariance values with PCA we use the elbow method to determine the count of transformed dimensions to be used for PCA. The dimensions to be used come out to be 4 for the income evaluation dataset. We can see that with 7 dimension we get a rand score of 0.05 for k means. Below table captures all the data for PCA transformed dimensions for income analysis dataset using clustering. For both datasets we clearly see the time reduction for getting same number of clusters as expected.

Metric	PCA	KMeans	EM
Rand Score	0.05	0.05	0.08
Run time	0.019	0.2	0.2
Cluster count	2	2	3

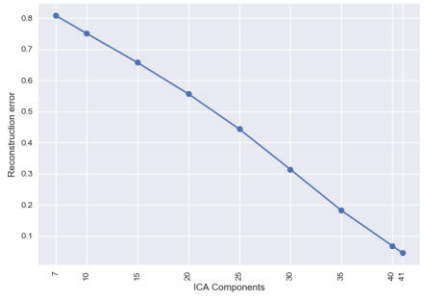
4.2 Independent Component Analysis(ICA)

4.2.1 Algorithm

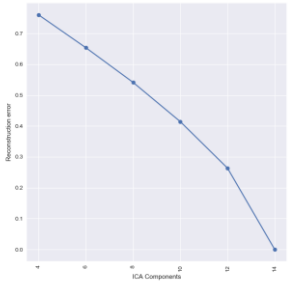
ICA transforms the input dimensions to independent non gaussian signals. Hence if the input dimensions have positive kurtosis indicating a non gaussian distribution then ICA can give us the corresponding independent signals. Here we apply ICA and check for the reconstruction error to determine the components to be used for ICA so that the features can be reconstructed from the generated distribution.

4.2.2 Analysis

The reconstruction error plots for deposit and income dataset are given as follows



Deposit dataset recon error



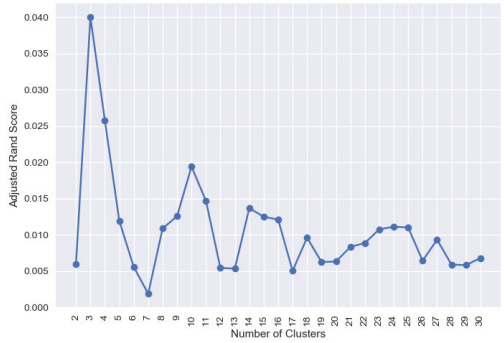
Income analysis recon error

The kurtosis for both the datasets have maximum of non negative values 36 for term deposit dataset and 11 for income analysis dataset indicating a non gaussian distribution. As expected value of n_components equal to all the dimensions gives us the least reconstruction error. However looking at the above graphs we can say that for deposit data set with components = 35 and for income dataset with components = 12 we get an acceptable reconstruction error hence we will use **deposit ICA component value = 35** and **income ICA component value = 12** for analyzing the results of clustering.

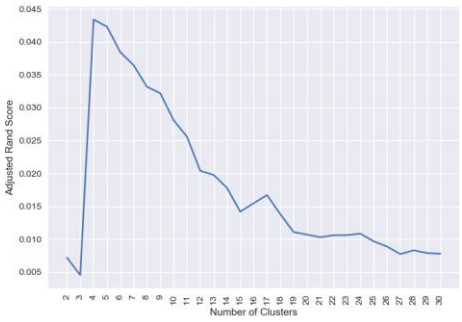
4.2.3 Clustering Analysis

4.2.3.1 Deposit dataset

The rand scores for term deposit dataset for KMeans and Expectation Maximization algorithm using ICA component = 35 are as follows



KMeans adjusted rand score



EM adjusted rand score

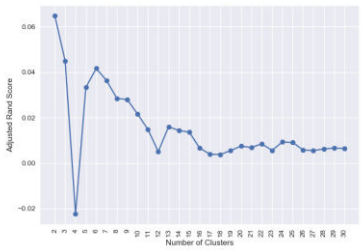
When we run the clustering algorithm with ICA with reconstruction error of around 20% we see that the rand scores for ICA are very close to KMeans score but they are lagging for the estimation maximization algorithm. Hence ICA over the deposit dataset does not give us significant gain when we try to aim for maintaining variance in the data. Hence value of k given by ICA against KMeans and EM is given in the table below.

Algorithm	KMeans	Expectation Maximization
# of clusters (k)	2	3
# of clusters with ICA	3	4
Adjusted rand score	0.05	0.078
Adjusted rand score with ICA	0.040	0.042

Thus we see that using ICA here for dimensionality reduction does not help us in improving performance.

4.2.3.2 Income evaluation dataset

The rand scores for income evaluation dataset for KMeans and Expectation Maximization algorithm using ICA component = 12 are as follows



KMeans adjusted rand score

EM adjusted rand score

When we run the clustering algorithm with ICA with reconstruction error of around 20% we see that the rand scores for ICA are very close to KMeans score but they are lagging for the estimation maximization algorithm. Hence ICA over the income dataset does not give us significant gain when we try to aim for maintaining variance in the data. Hence value of k given by ICA against KMeans and EM is given in the table below.

Algorithm	KMeans	Expectation Maximization
# of clusters (k)	3	2
# of clusters with ICA	3	2
Adjusted rand score	0.055	0.2
Adjusted rand score with ICA	0.040	0.042

Thus we see that using ICA here for dimensionality reduction helps us in improving performance for expectation maximization algorithm.

4.3 Random Projection(RP)

4.3.1 Algorithm

The core idea behind random projection is given in the Johnson-Lindenstrauss lemma, which states that if points in a vector space are of sufficiently high dimension, then they may be projected into a suitable lower-dimensional space in a way which approximately preserves the distances between the points.

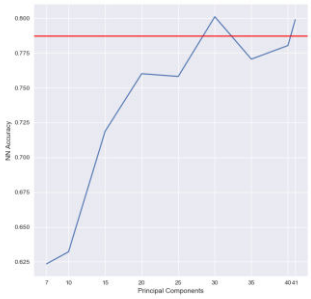
4.3.2 Analysis

After applying random projections using Gaussian Projection and Sparse random projection since ICA gave us the results indicating the datasets to be non gaussian we use the Sparse Random projection for effective projection of data to lower dimensions.

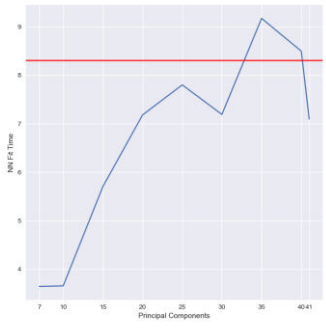
4.3.3 Clustering Analysis

4.3.3.1 Deposit dataset

I ran random projection algorithm for multiple random components from the list [7, 10, 15, 20, 25, 30, 35, 40, 41]. Then the results after dimensionality reduction were used for training a neural network and we found out the **component value** to be used for random projections to be **30** since with 30 random components we could see that we are getting better accuracy with less time as shown below.



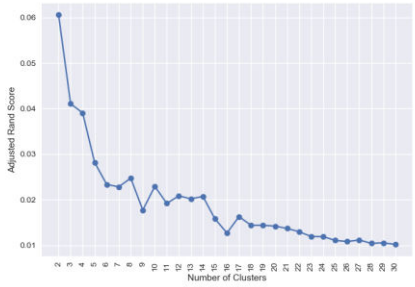
accuracy 0.8



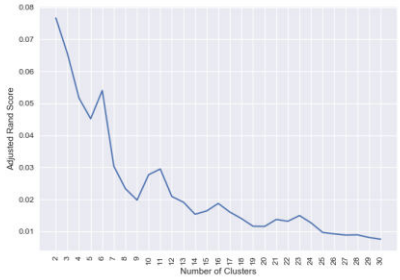
fit time 7.2 seconds

From the adjusted rand scores as mentioned below in the table it is pretty much evident that Radom projections perform well in this case for deposit dataset for k means algorithm whereas for EM algorithm they perform equally well.

Clustering algorithm	KMeans	EM
# Clusters with ICA	2	3
# Clusters Original	2	2
Original clustering scores	0.05	0.078
ICA clustering scores	0.06	0.078



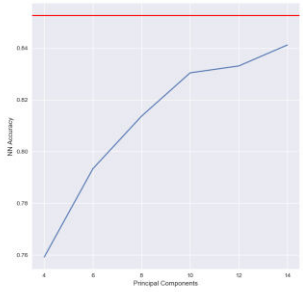
KMeans adjusted rand score



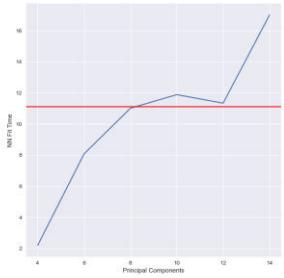
em adjusted rand score

4.3.3.2 Income dataset

I ran random projection algorithm for multiple random components from the list [4, 6, 8, 10, 12, 14]. Then the results after dimensionality reduction were used for training a neural network and we found out the **component value** to be used for random projections to be **8** since with 8 random components we could see that we are getting accuracy closer to baseline neural networks.

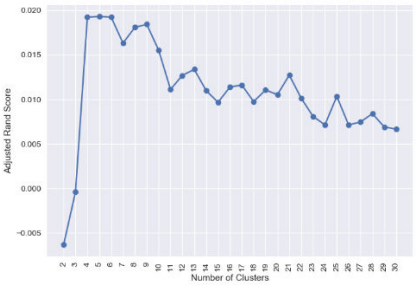


Accuracy 0.81



fit time 11 seconds

From the adjusted rand scores as mentioned below in the table it is pretty much evident that Radom projections do not perform well in this case for income dataset for k means algorithm as well as EM algorithm.



KMeans adjusted rand score = 0.019

EM adjusted rand score

Clustering algorithm	KMeans	EM
# clusters with ICA	5	4
# clusters with original	3	2
Original clustering scores	0.055	0.2
ICA clustering scores	0.019	0.087

4.4 Univariate Feature Selection (UVFS)

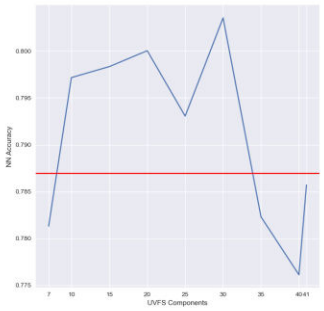
4.4.1 Algorithm

UVFS selects features by choosing features based on univariate tests. I have used the SelectKBest algorithm to choose k best features which were passed to the function and value of k is validated using neural network as follows

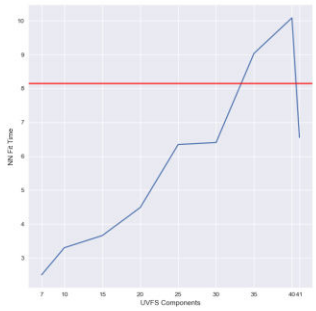
4.4.2 Analysis

4.4.2.1 Deposit dataset

I ran random projection algorithm for multiple random components from the list [7, 10, 15, 20, 25, 30, 35, 40, 41]. Then the results after dimensionality reduction were used for training a neural network and we found out the **component value** to be used for random projections to be **30** since with 30 random components we could see that we are getting better accuracy with less time as shown below.



NN accuracy 0.8



NN fit time 6.5 seconds

The adjusted rand scores for deposit dataset can be given as follows, this indicates that UVFS performs well from time standpoint with KMeans for deposit dataset with equal accuracy but same is not the case with EM.

Clustering algorithm	KMeans	EM
# Clusters with ICA	2	7
# Clusters Original	2	2
Original clustering scores	0.05	0.078
ICA clustering scores	0.05	0.055

4.4.2.2 Income dataset

I ran random projection algorithm for multiple random components from the list [4,6,8,10,12]. Then the results after dimensionality reduction were used for training a neural network and we found out the **component value** to be used for random projections to be **10** since with 10 random components we could see that we are getting closest accuracy with less time as shown below. We can see here that UVFS does not match with neural network accuracy but does a good job of clustering with EM.

Clustering algorithm	KMeans	EM
# Clusters with ICA	2	7
# Clusters Original	2	2
Original clustering scores	0.05	0.078
ICA clustering scores	0.05	0.175

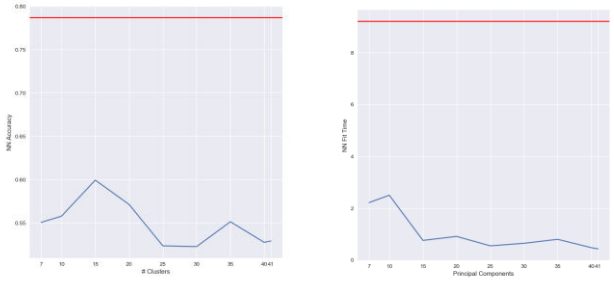
5 Neural Network Analysis

The cross validation accuracy and fit scores obtained for dimensionality reduced datasets with Neural network are as follows. This gives is a clear understanding about utility of dimensionality reduction in unsupervised learning for deposit dataset. This dataset has 41 features which is more from the selected datasets hence it is a better dataset to analyze the effects of dimensionality reduction.

DR algorithm	PCA	ICA	RP	UVFS
Best accuracy	0.82	0.82	0.80	0.83
Best time	8.2 S	5 S	7.1 S	6.2 S
Components Used	30	30	30	30

Thus from above observations we can say that for term deposit dataset can say that all the dimensionality algorithms perform almost equally well with very minute difference between each of them. It can be said that ICA performs well with almost the same accuracy gaining about a second with the maximum score contributor UVFS. All the plots for this can be found out in the plots folder in the source code.

6. Clustering as dimension reduction



I ran the KMeans algorithm over a range of k as [7, 10, 15, 20, 25, 30, 35, 40, 41] and used the clusters generated from that value of k as input to the neural network. Above graphs represent the accuracy score (left) and for time (right) with baseline score of neural network as red line. We can clearly see that even if with the clusters the algorithm runs much quicker in less than 2 seconds almost every time we get a very poor accuracy around 0.55 every time. This indicates that the clusters generated from clustering methods do not work well in modelling

7. Conclusion

Overall from all the experiments we can say that dimensionality reduction reduces the running time and hence computational complexity of the underlying learning model at the cost of accuracy. If the drop in accuracy is not huge then we can use dimensionality reduction algorithms for preprocessing in learning process to save on resources. If we manage to improve accuracy or keep it the same with less time then dimensionality reduction should be used as a preprocessing method

3.3 References

https://realpython.com/k-means-clustering-python/

https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60

https://towardsdatascience.com/independent-component-analysis-ica-in-python-a0ef0db0955e

https://www.youtube.com/watch?v=bfS7JAjiOMI