Waleed Sial wxs180004

Applied Machine Learning

Assignment 4

Dataset: The first dataset is the Appliance Energy Dataset and the 2nd dataset is credit card payment default dataset.

K means Clustering

Question: Should we separate X & Y variables in k means clustering?

Answer: To my understanding, k means is an unsupervised method, and, in this case, we don't have any notion of labelled data. Therefore, we should not separate the variable y. However, since in our case we know the classes, later we can use the y class labels for comparison purposes.

Unsupervised learning is learning about the data, exploring and analyzing it. To see how correct our clusters are formed, we can test the clusters and see if a data point is assigned to a cluster which is like the class label we have.

Visualization of Clustering

We can make clusters from any number of features; however, we will not be able to visualize them using all features if the number of features is more than 2 or 3.

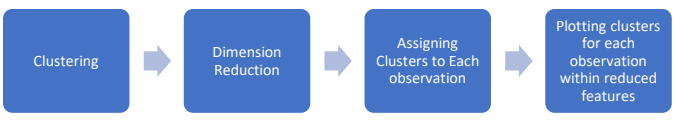
Question: Can we visualize data after applying clustering?

It must be noted that here we are applying dimensionality reduction after the k means clustering so that we can plot it, the clusters were obtained using all the features without any dimensionality reduction.

Once we have the cluster labels, then we will apply dimensionality reduction on the data, reduce the dimensions from 28 to 1,2 or 3 since we can plot till 3 dimensions. In our assignment we have used 3 dimensions.

<u>Visualization Process Pipeline</u>

Following is the pipeline which we have followed, for dimension reduction I have used T-sne for the first experiment:

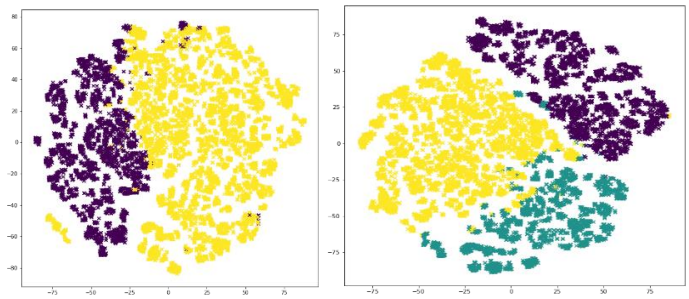


For the 1st task of creating clusters and visualizing them, we will try for different number of clusters and see if we can make sense of the under-lying data:

Clusters = 2 & 3

Question: What does these cluster plot below indicate?

Answer: It is difficult to explain the clusters since we have 28 features in plain words, however we see that in the both the cases the clusters are well formed generally, they are sort of mixed at the boundaries and in case of 2nd diagram we have some sort of distant mixing as well.



Question: Can we check the accuracy of clustering?

Answer: Yes, there are couple of methods which we can employ to get a notion of accuracy. We will explore those methods in detail however in our case, we can test using our class labels.

In our earlier assignments, we defined Appliance energy usage as our Y variable which we classified as High or low appliance energy usage. It is therefore intuitive to see if our clusters are like our class labels.

Question: How to compare our cluster labels with class labels in Appliance energy case?

Answer: Clustering labels are not equivalent to class labels but what we can try to develop a relation.

We shall see which clusters are assigned to which datapoints and what kind of classes they have. In other words, we want to see if the cluster assignment has any similarity with class labels or not.

For example, in our case we have the following value **counts for our class labels:**

```
Class 0    14524
Class 1     5199
```

Cluster Labels:

```
Cluster 1    13301
Cluster 0     6422
```

It must be noted that Cluster 1 is, and Cluster 0 are just two clusters, it does not mean that Cluster 0 means it is low appliance energy usage group. Clustering is not at all classification. Though they are comparable in some terms, which we shall describe below:

Now we see that the bigger cluster can correspond to class 0, we can compare the indexes of class label 0 and cluster label 1 and see how many of them match.

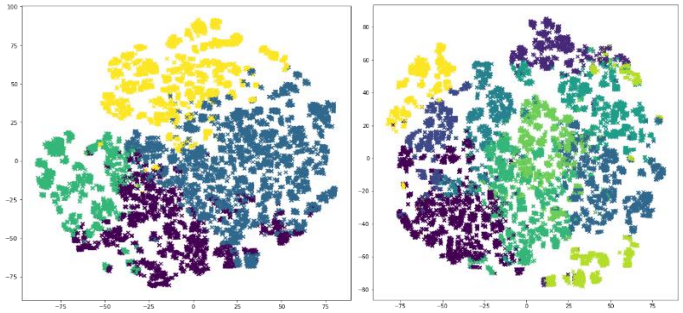
The number of indexes which had class label 0 and cluster label 1 are 10196. If we divide this number by the total number of class 0 index than we get the following statistics:

0.702 i.e. approximately 70 % of the class 0 labelled observations are in cluster 1.

Where as for the observations with class label 1, 40 % were in cluster 0.

Analysis: This means that both the clusters have some sort of overlapping with respect to our original class labels and they are not very perfect. The clusters predominantly are like the class labels in our case.

Increasing the number of clusters can be done to see visually if more subgroups are being formed.



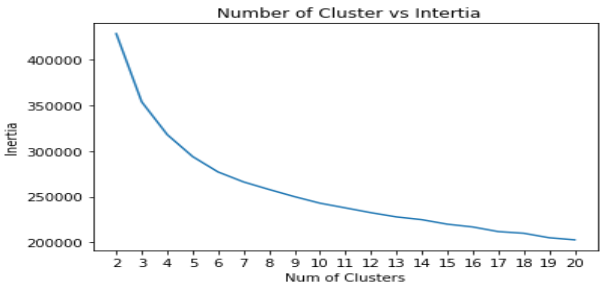
The above plots show the results when we increased the number of clusters arbitrarily to higher number. Therefore, this shows that we need to have some notion of a good number of clusters.

Question: How do we determine ideal number of clusters?

Answer: There are multiple methods which we employ to evaluate the formed clusters.

Inertia: Internal Evaluation

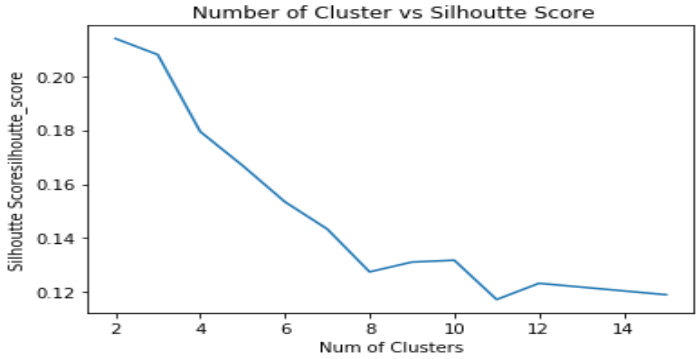
One built in attribute in case of Sklearn is inertia which is sum of squared distanced of samples from their nearest cluster center. Ideally, we want to minimize it and we want to select the number of clusters accordingly, we can evaluate, plot and select using elbow method.



The plot shows that initially when the number of clusters is low, the data points are located far away from the centers and as we increase the number of clusters the distance continue to decrease. It must be noted that that inertia is sum of squared distanced of all samples hence it is a high value. Using Inertia, we can try with clusters between 6-9.

Silhouette score Evaluation

A higher silhouette score indicates a model with well defined clusters. Silhouette score varies between -1 and +1 Following is a plot of silhouette score, however this plot may need some improvements. A score which is near to 1 indicates that the instances are clustered together and are far from the other clusters, however in our case we see that we don't the clusters which are far from each other. As we increase the cluster number, the distance of data points in a cluster from the other clusters is bound to decrease which is shown by the below plot as well.

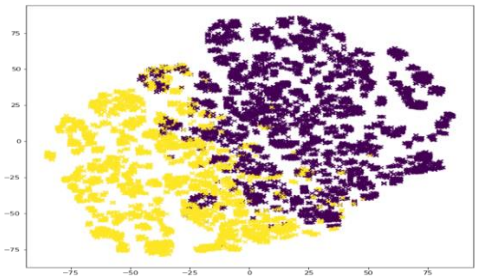


Expectation Maximization

EM instead of hard assignment of data points assigns probabilities to the data points. We assume that data points were generated by gaussian distributions and we try to find those distributions.

In this section, we will use gaussian mixture class of sklearn to make clusters and once we have the clusters, we shall plot those clusters in reduced dimensions.

Following is the case of when we assumed number of gaussians were 2:



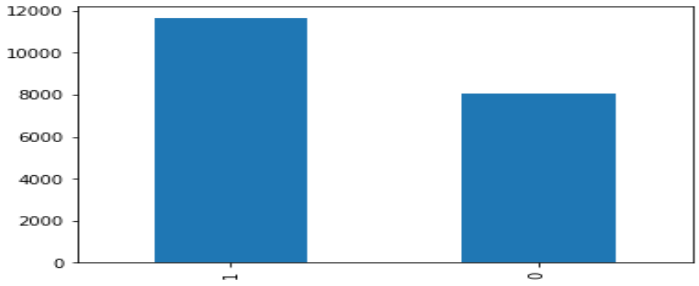
Visually, the resultant clusters exhibit some similarity with the k means clusters however we shall check the goodness of the clusters using empirical methods.

For 2 gaussians, we can check if they line up with the class labels or not. For the rest of the experimentation it is not suitable since we only have 2 classes.

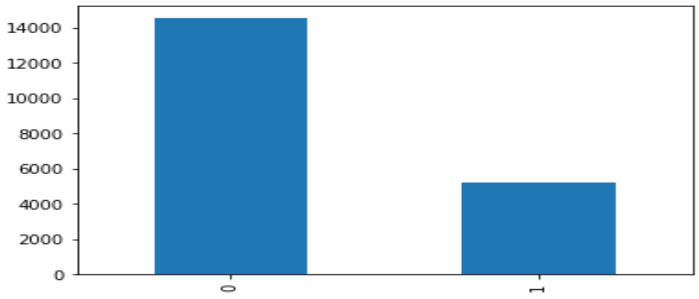
Question: How close are the clusters to the original class labels?

Answer: For every data point, we have a class label. Once we have the estimated gaussians, we can than find out how many of the data points of let's say class 'A' are in assigned a gaussian.

For our example of 2 gaussians, we have following distribution of our **gaussian assignment** to all data points:



Where as if we compare it with our **class labels distribution**, we see following distribution:



We can assume intuitively that class 0 and gaussian 1 have similarity and we can find out the number of common data points between class 0 and gaussian 1 to get a notion of how close the EM method is to original class distribution. ***60 % of the data points which have class 0 have been assigned gaussian 1.***

Questions: How do we know which component has more weight?

Answer: Weight is the probability of a data point joining that gaussian. Gaussian mixture class in sklearn has an attribute for weights which shows the associated weight for each component. For example, in the case of 2 gaussian mixtures following weights were returned, with one component is slightly more dominant:

0. 59132533	0.40867467

Where as in case of 3 gaussians we have following weights returned for each gaussian:

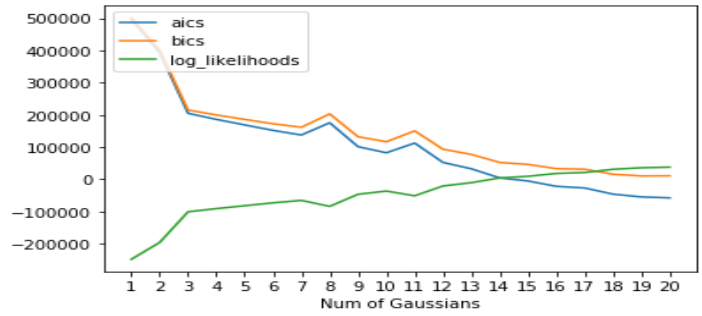
0.21208164	0.57346459	0.21445377

With 3 gaussian, we get one more dominant than other 2.

Selection of Gaussians

We cannot use silhouette scores with EM as we did with k means because it is not very reliable in case when clusters are not well formed. Therefore, we should use BIC, AIC or similar criteria.

AIC, BIC, Log-likelihood For BIC & AIC we prefer the lower score where as the score function gets us per-sample average log-likelihood of the given data X, we take its product with the number of samples to get the final log likelihood score of the model.



Elbow Method:

It must be noted that Elbow method is a heuristic following this method we see that 3 gaussians can be considered the elbow point. I have plotted the log likelihood for comparison purposes which shows that the increase in the score starts flattening at 3 gaussians. Considering the behavior of these 3 curves we can safely assume that **selecting 3 gaussians is a better approach**. However, there are other advanced methods as well which we can further explore.

Automatic Optimal Cluster Number

If we have some domain knowledge about the dataset and we can safely assume that clusters cannot be greater than some number, we can use **BayesianGaussianMixture**. Using this method, we can eliminate un-necessary clusters. Following are the weights for number of clusters.

1	2	3	4	5	6	10	11
0.07	0.11	0.08	0.05	0.07	0.05	0.1	0.1

Dimensionality Reduction

As the first algorithm for this step, we shall try PCA. For the experimentation purposes, we shall start with 1 principle component and will increase.

Question: How many principle components we should try for?

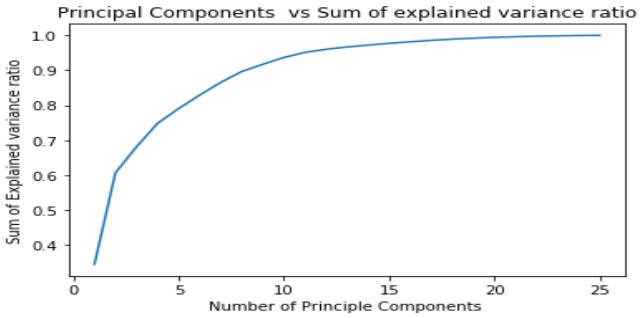
Answer: Ideally, we want maximum Variance (information) to be retained therefore for our case we should try for 90 % of variance retained.

For this purpose, we can try first for 26 components, sklearn provides us with percentage of variance explained by each of the component. Using that, we can sum up the variance

explained and that should be in range of 80 to 90%, at least for our experimentation purposes.

Number of Principle Components and Explained Variance

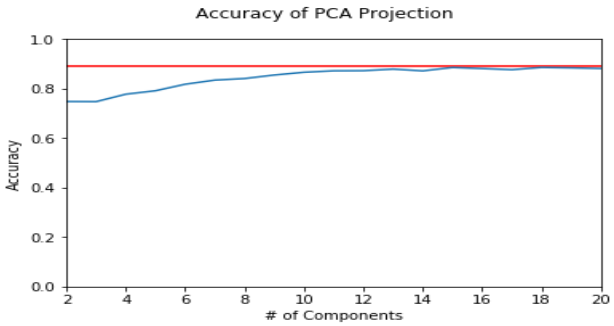
The following graph shows relationship between number of principle components and explained variance sum. This shows that maximum variance is retained when we have 25,26 components but that kills the purpose of dimension reduction.



Following table accompanies the above graph. From the table and graph we can safely select 8,9 principle components to retain almost 90 % variance of the dataset.

Num of Components	Variance Retained
1	0.3452425147185698
2	0.6068434331038963
3	0.6811899501351539
4	0.7483530994850867
5	0.790455305592957
6	0.8291003867888574
7	0.8653337617162418
8	0.8964340256373474
9	0.9170801656189046

Now in the following section, we shall use the transformed features, train our neural network on the transformed features produced by PCA and compare the results with the base line. The base line is obtained by training the neural network on all the features and using best hyperparameters



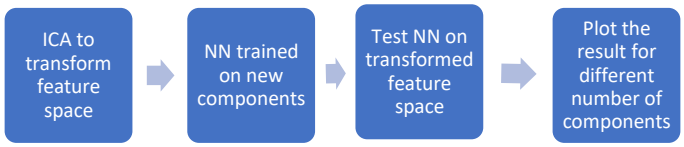
This result is in line with the variance retained plot. ***We see that we can get close to the best performance with features as low as 9.*** At that point and onwards, we are very close to the best performance of neural network with full feature space. **Please note that Neural network best score with full feature set is 0.889**

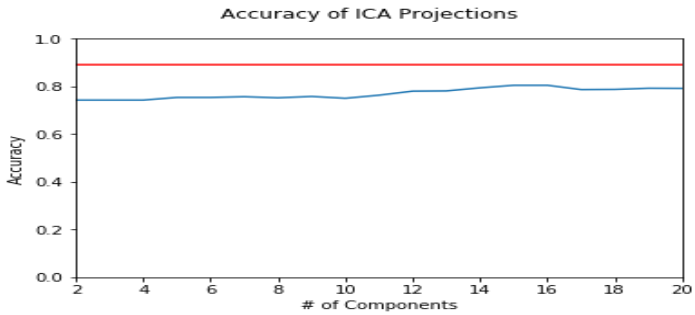
# of Components	Neural Network Accuracy
2	0.74803
3	0.74752
4	0.77769
5	0.79163
6	0.81799
7	0.83472
8	0.84106
9	0.85525

This conclude our discussion of PCA dimension reduction and corresponding performance of Neural network using reduced feature space.

ICA

In sklearn we have ICA implemented as FastICA. Unfortunately, we don't have ratio of explained variance for this method. Therefore, for this section the process of selection, evaluation of features will be done by comparing the performance of neural network. We will fit the neural network with different number of components and compare the results with the best results of full features dataset.



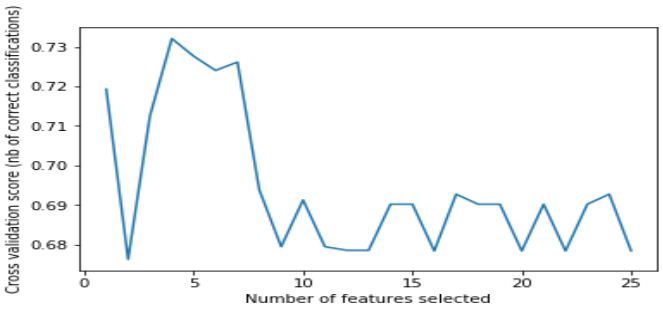


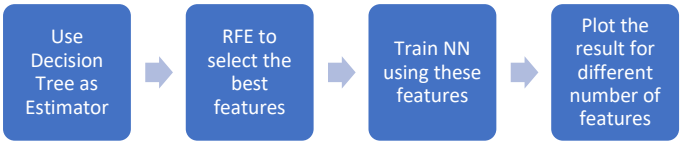
The above plot shows that with ICA projections we are closest to best performance neural network at around 15,16 components, however ICA is particularly slow and needs hundreds of thousands of iterations to converge. Following table is a summarized version of scores for different components numbers:

# of components	Score
2	0.742
15	0.804

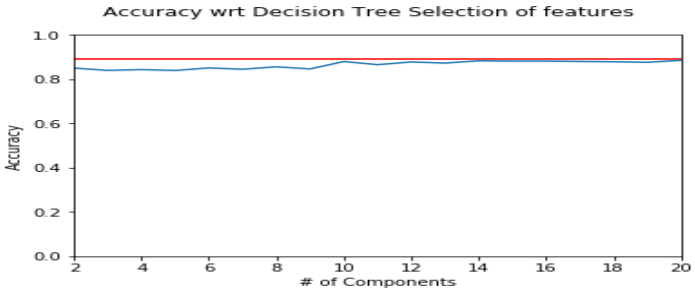
Decision Tree Recursive Feature Elimination This method takes an estimator and returns minimum number of estimators that are optimal. For example, for our case we have selected decision tree as an estimator and following

results are obtained:





We looped over number of features to be selected. Those features were selected by recursive elimination algorithm which used decision trees as its estimator.



Once these features were obtained, we reduced the original dataset to the features obtained by RFE and then ran Neural network onto that dataset. The resultant plot is hence accuracy of neural network using limited features obtained by RFE and Decision tree. This shows that 10 and 10 onwards gets us very close to the baseline performance.

Randomized Projections

In randomized selection we project the data into lower dimensions using random linear projections. Basically, it has faster computation and it gives away very less accuracy. The sklearn module has an auto feature but it does not always reduce the number of features. It depends upon the size of the dataset and the EPS parameter which is a positive float. **Even if we select EPS = 0.99 and select our full dataset it still returns 237 features.** Therefore, instead of using the automatic selection in Random selection, we can give the number of components to which we want our feature space to reduce to and further test the performance by running Neural network or similar algorithm for this. *Most of the times, dimensionality reduction is basically a preprocessing step for a downstream classifier or some other algorithm.*

Further Research

Sklearn random projection class has another module called `johnson_lindenstrauss_min_dim` which is basically used by both random modules of sklearn(Gaussian & Sparse). Now, ideally this function will return "a safe number of components to randomly project to". However, funny thing is that when we apply this method we get '8476' dimensions. As explained earlier, this depends upon dataset and EPS, but question is that we did not reduce anything here! This means that we cannot make any assumption about preservation of pairwise distance. In the following sections we will explore this further.

Question: How do we select/evaluate the number of components?

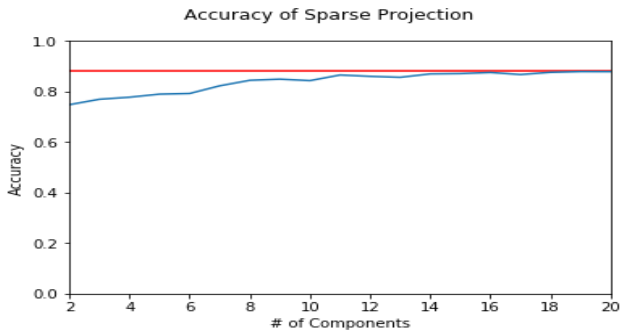
Selection of component can be coupled with the performance on neural network. We can try for different number of components and see which number of the component gets us best performance as compared to neural network performance.

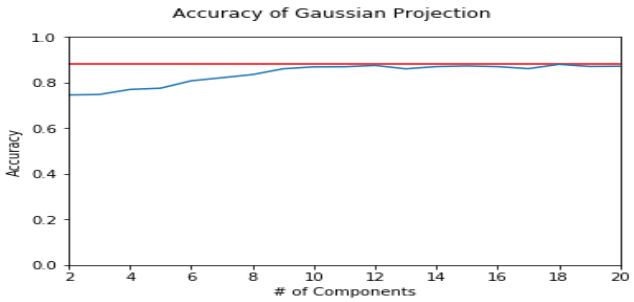
Neural Network Performance with Random Projections

Our baseline score, achieved by utilizing the original feature space and best parameters obtained in previous assignment is: **0.8899873257287706**

And following the graph where we compared the accuracy which we shall obtain on test data while trying for different number of components.

Gaussian Vs Sparse Projections For comparison purposes, I have included both types of projection gaussian and sparse. Sklearn random random projection implements these two modules. Both show that we achieve the base line accuracy with 18 components.





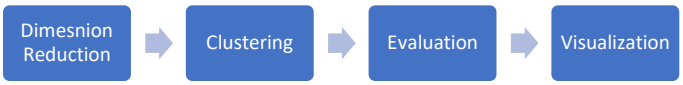
Using these kinds of curves, we can evaluate and select different number of components.

Projection	Components	Score in reduced	NN Score
Gaussian	18	0. 8823827	0.889
Random	19	0.8844106	0.889

In case, even if we want to reduce further at the cost of decrease in accuracy we can come as down to 11 components with accuracy around 0.87122. Hence this is the way that we evaluate the performance of our model and select the number of features accordingly.

Clustering After Dimension Reduction

In the following section we shall proceed as follows:



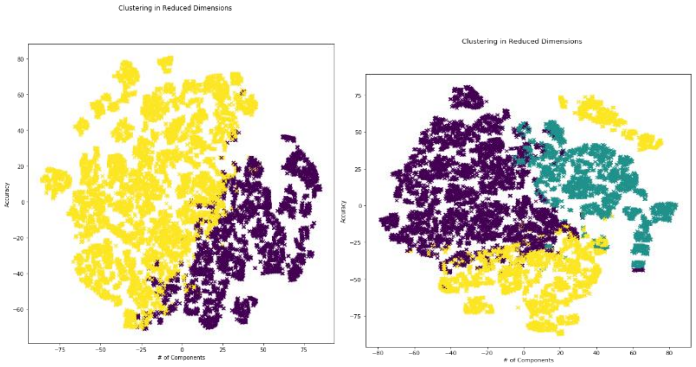
To reduce the dimensions, we shall utilize our findings in the previous section and reduce the dimensions accordingly. For example, in PCA we can get good score from 6 to 9 dimensions. Let's start with that:

PCA & Clustering

We will use PCA to reduce the dimensions to let's say 6 and then using these dimensions we do clustering. It must be noted that we don't have any notion of training and testing in case of unsupervised learning, the number of dimensions to reduce to we have selected based on the experiments we did with neural network.

Clusters 2,3 in Reduced Dimensions number 6

Above is the example of clusters that were formed using 6 dimensions. For 2 dimensions we see that we are still getting well formed clusters though not well separated where as in case of 3 clusters we have some mixing up.



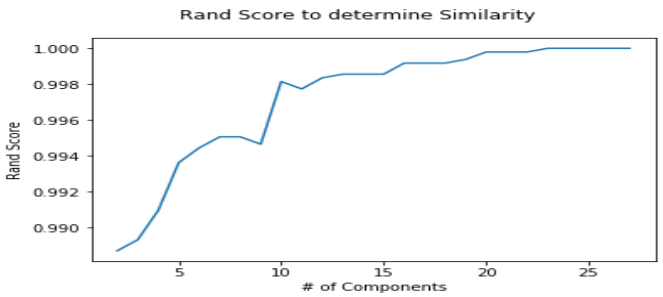
Question: **How to empirically test the higher number of clusters?**

Answer: We can test the 2 clusters and see how much of they can be like our class labels, but this methodology does not work with higher dimensions since we don't have much to compare with. However, there can be other metrics which we can test that how well a cluster is formed.

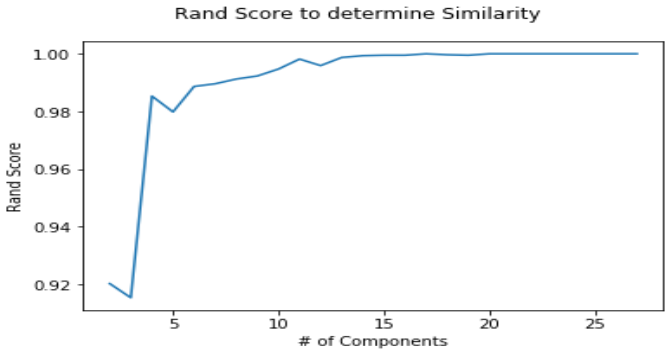
Adjusted Rand Score: Comparing Clusters in Reduced vs Full Feature Space

This method can be utilized to compute similarity between two clusters. Using this method, we can find the similarity between same number of clusters being produced in higher and lower dimensions. This will help us differentiate the clusters produced in both cases:

# of Clusters = 2,3



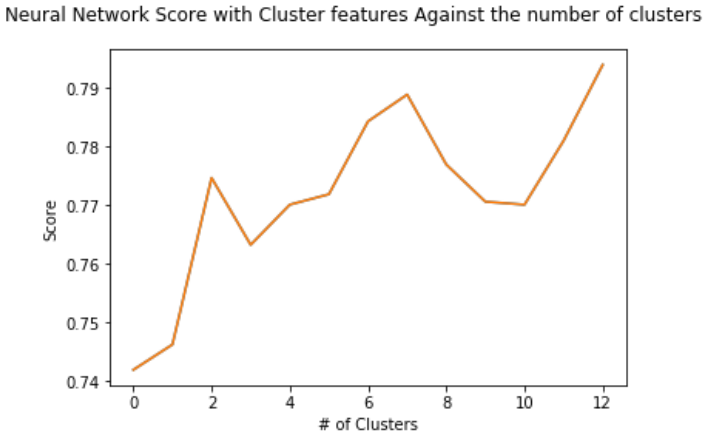
# of Clusters = 3



The above plot shows Rand score for 2 and 3 clusters, we see that the clusters produced in reduced space become quite similar near 6 and increases further as well. For 2 & 3 number of clusters, we get reasonably good similarity at 6 features. **Using Clustering Results as new Features**

In this section, we are asked to use clustering results as features. **Clustering as Dimension Reduction**

Using clustering results to cluster the data and use it as supervised learning is sort of dimension reduction using clustering results. Now the clustering results can be simply cluster assignments of each data point. It can also be some metrics related to cluster properties. For example, each data point distance from each cluster. Let's say that there are 4 clusters than we can have 4 features with each being the distance from each of the centroid. In the following section we shall see the results for different number of clusters.

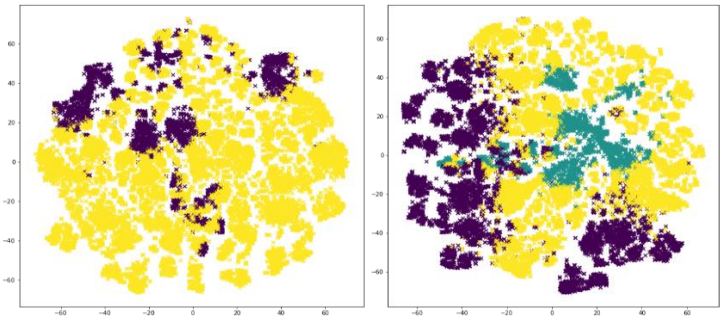


The above plot shows the result of neural network on test data set obtained by using the clustering results as the features. It shows that as we increase the clusters, our performance may increase which is plausible since increase in clusters means more and more features. However, from dimension reduction point of view and using clustering as a mean of turning unsupervised learning into supervised learning, good performance is achieved at 7 clusters.

Dataset 2

Clustering

K-Means: For the sake of brevity, the number of clusters for this section are 2 & 3.



Visually, we see that clusters in this dataset are not well formed with the selected number. Perhaps, we can later try to find a suitable number of clusters and see if the plot makes sense.

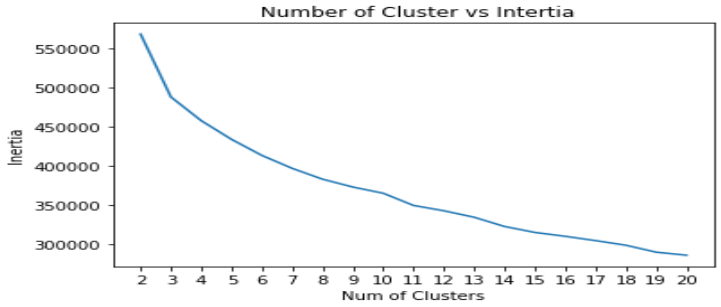
Clusters & Class labels: To answer the question whether the clusters align with our class labels:

% Class 0 & Cluster 1 Common	**84 %**
% Class 1 & Cluster 0 Common	**14 %**
% Class 1 & Cluster 1	**85 %**

Question: Does these statistics help us in any way?

Answer: No, this shows that clusters are not formed considering the class labels. Cluster 1 contains data points which have both class 1 and class 0. We don't have well formed clusters.

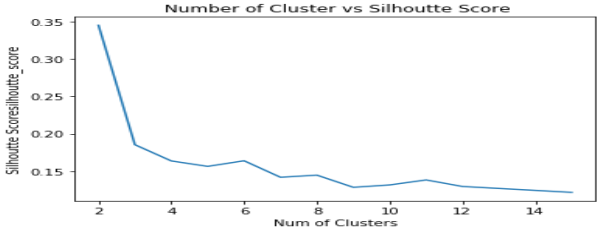
Best Number of Clusters For K-means clustering, we can utilize inertia and silhouette scores for finding suitable number of clusters: **Inertia**



If we follow the intuition of Elbow method, we see that we see a clear dip from 2 to 3 clusters, however we saw that even at 3 clusters we don't have well-formed clusters, even though Inertia continues to decrease, and it makes perfect sense as well since the distance decreases with increase in clusters. However, from data analysis point of view it is not of great help.

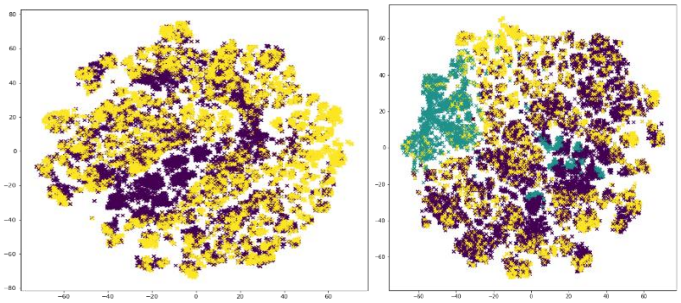
Silhouette Score

The following plot is kind of telling the same tale with different number, we prefer silhouette score to be as near possible to +1 however it seems that we don't really have perfect clusters. We see that silhouette score decreases from 2 to 3 clusters. It seems that having a smaller number of clusters is favorable for this dataset. The highest silhouette score we have is for 2 clusters. If we compare inertia with Silhouette score, the graph is comparatively easy to read for Silhouette.



EM

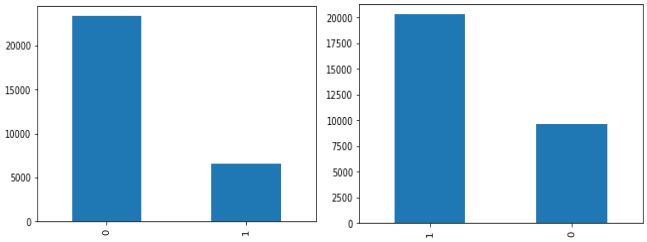
Following plots show 2 and 3 gaussians, in both the cases we have very mixed up gaussians.



It seems that the underlying assumptions of independent gaussians is not holding in this case. However, we can see the weights, AIC, BIC for having more understanding. The following table shows **top 3 weights** when we used 16 gaussians:

0.27	0.11	0.1

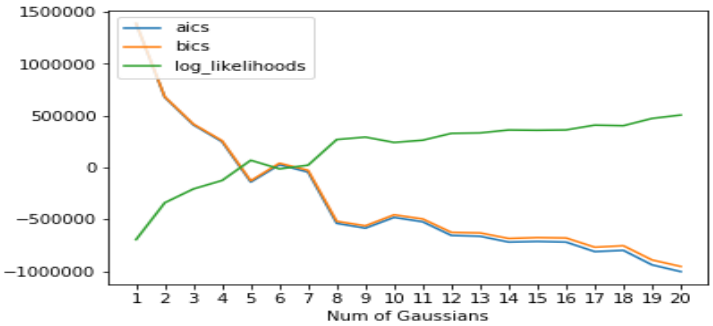
Class & Cluster Distribution (Left is class & Right is cluster)



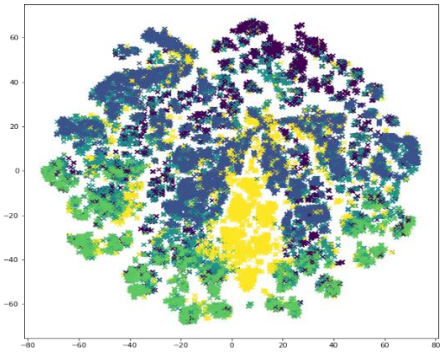
Interestingly, it seems that the cluster 0 and class 1 has similar distribution. Let us see how many of credit card defaults (class 1) have been assigned gaussian 0. However, only 21 % of the data points which had credit card default were part of group 0. Maybe with increase in number of clusters we can eventually have a clean group, but we must always remember that here we know what the actual class labels are.

AIC & BIC

As explained earlier, we cannot use the silhoutte score for EM, therefore to evaluate the gaussians we use aic, bic. If we follow the intuition of Elbow method we see that with 5 clusters we get the best score.



However, even with 5 gaussians we see the following plot, which is not very intuitive.

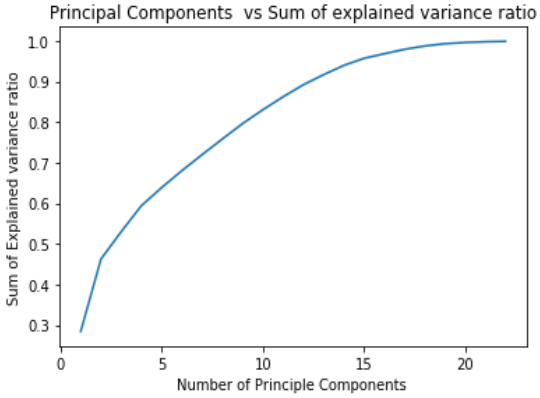


As the next step, we shall proceed to Dimension reduction.

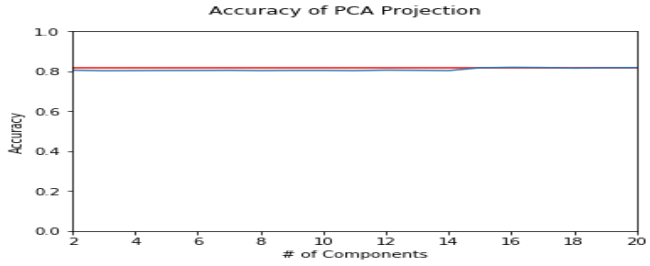
Dimension Reduction

In this section, we shall reduce the dimensions and use the neural network with the reduced dimensions to evaluate performance

PCA & Variance The below plot shows that if the goal is to retain maximum variance. With 9,10 components we get the maximum retained variance.



We can run NN on the reduced dimensions and see which number of components gets us closest to the best performing network which we had obtained in previously. The following plot shows the test accuracy with reduced features(components) compared with the base score of neural networks (in Red) on full feature set. The full model score is **0.818266**.

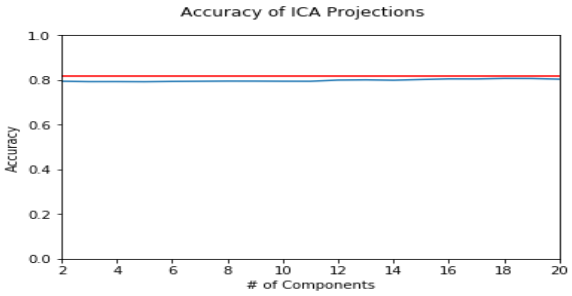


In the above plot blue line is reduced features performance, we see that with the increase in principle components the reduced model catches up to the full model.

```
0.8038666666666666,
0.8178666666666666,
0.8201333333333334,
0.8192,
0.816,
0.818.
```

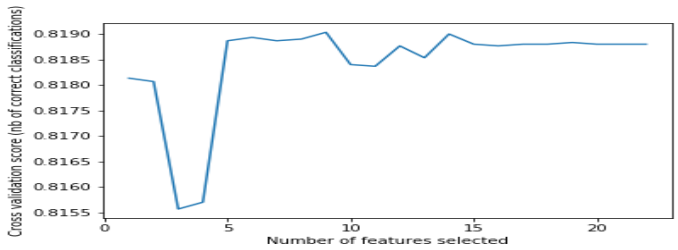
For the sake of brevity, I have included the above screenshot which shows that in this dataset, the reduced model not only catches up to the full featured model but also the accuracy improves as compared to the full featured model. In practice, PCA is just projecting the data, it is not adding anything however it removes noise and helps improve Neural network like algorithms to converge faster.

ICA & Neural Network

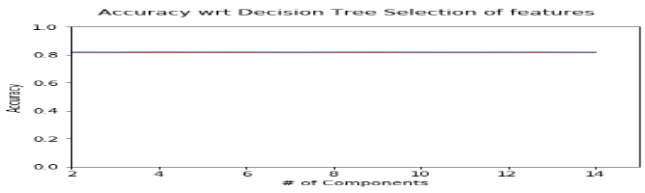


In case of ICA, we see the similar but not same behavior as of PCA as early as we saw in PCA. Though ICA still performs close to the baseline model (in red) when the independent components increases to 18,19 however it does not improve the accuracy and it does not reach to the baseline performance with less features as we saw in case of PCA.

Decision Trees & Neural Networks As explained in the 1st dataset section, we can use recursive elimination method with Decision trees as estimator to select the features. The recursive method returns the cross validated results. The following plot shows that with 5 features, we can get very reasonable accuracy though the optimal number of features returned by the method is 10.



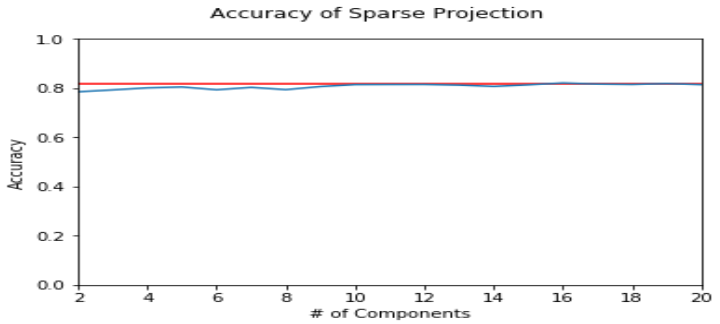
This method also returns a ranking of the optimal features and importance of the features as well. We see that most *important feature for credit card payment default is Pay_0 which is basically the payment in the last month.* Using the same recursive elimination algorithm, we can select a subset of the features. We can then use this reduced feature space to train our neural network and see how our neural network performs against our baseline model.

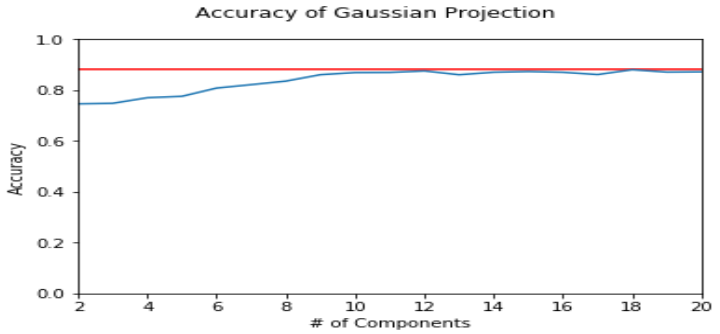


In recursive algorithm we select the minimum number of features to select and as a result we get the optimal numbers. However, we see from the above plot that in this dataset we can fairly do very well with as less as 2-4 features. Again, using the ranking attribute we can select the top most features which shows that the most important features are related to payment history. **PAY_0, PAY_5, PAY_AMT2 and these 3 features gets us accuracy of 0.82**

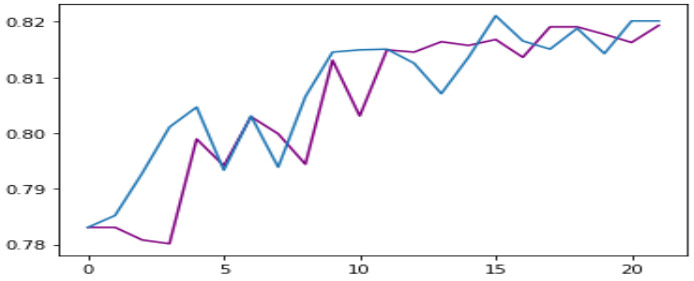
Neural Network Performance with Random Projections

Our baseline score, achieved by utilizing the original feature space and best parameters obtained in previous assignment is: **0.818** ***Gaussian Vs Sparse Projections***





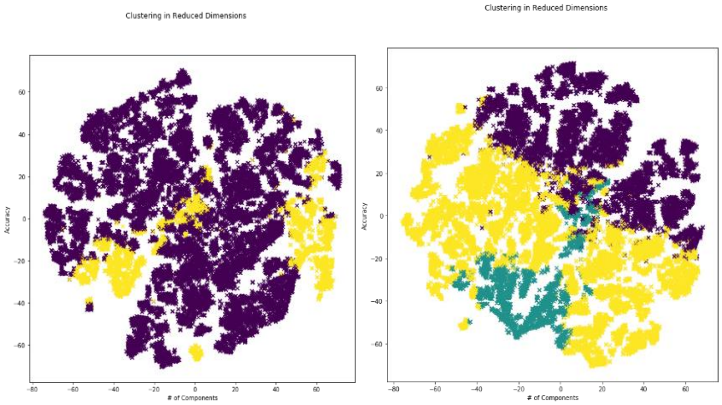
The above plot shows that we cannot assume nature of the gaussians. The above plots show that in case of Sparse projections we get closer to the baseline performance with less components as compared to gaussian projections. **This is evident below with the Sparse Random projections (blue curve) being better than purple at the greatest number of components.**



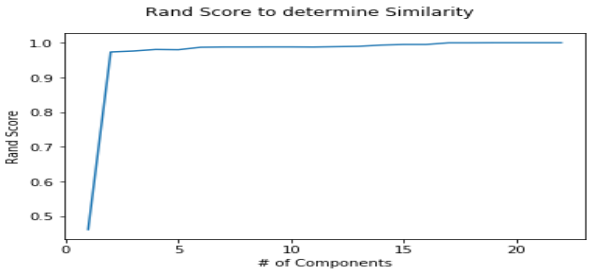
Projection	Components	Highest Score in reduced	NN Score
Gaussian	22	0. 819	0.818
Random	19	0.821	0.818

Clustering After Dimension Reduction

For clustering after dimension reduction, we will be reducing to the dimensions which were optimal as per our experiments before. In our experiments we have seen for PCA we can get high retained variance with 9 components and the cluster results for 2 and 3 are as follows:

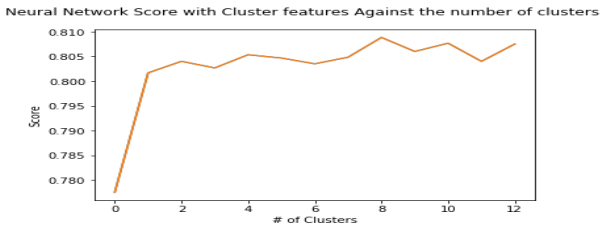


To test how clusters were formed in full and reduced feature space we can utilize adjusted rand score and interestingly it shows that we can get similar clusters as we got in full feature set with as low as 2,3 components:



Using Clustering Results as new Features

In this section, we will create new feature matrix by utilizing the newly formed clusters. As we did in the first dataset, we create clusters and using the cluster centers we create new features. The new features are basically distance of each data point from different number of clusters. Using this approach, for different number of clusters, we initialize neural network and see neural network performances using clustering results.



The above plot shows that just by using clustering results, with clusters as low 2 we get a very good result on neural network.

Summary

We experienced that with clustering algorithms it is not just enough to run the clusters. We need to know the underlying assumptions of different kinds of clustering algorithms as well. For example, we saw that in the 1st dataset both EM and k means produced similar results where as that was not the case so much in case of 2nd dataset.

To visualize the results of clusters, we cannot use all the features and even for visualization we must do dimension reduction, for this purpose in both the datasets a pipeline was built which first created the clusters in all or required number of features and then to plot the clusters we reduced the dimensions as well. For this purpose, T-SNE was used which is an advanced dimension reduction technique specially used for cluster visualization. It was recently shared by Hinton in 2008.

In case of Dimension reduction, once again, different dimension reduction methods have different assumptions and there is not one fit for all the kinds of datasets. Dimension reduction by decreasing the dimensions is sometime not useful when we are losing lot of information but in case of our datasets, we saw that we retained most of the information and underlying neural network performances with very a smaller number of features.

In case of last step, the main purpose was to use the clustering results. Now a naïve or a very simple approach will be to just use the clusters and to use them to predict the class labels. However, I did not use this approach rather I used which can be categorized as centroid method. In this method, I used the cluster distances from the data points. Each data point has some distance from each cluster, let's say if the clusters are 7 than each data point has a distance with each cluster center. As a result, we get a new feature matrix which has 7 columns. This approach proved to be very successful considering that we are converting unsupervised learning into supervised learning and we obtained a score as high as 0.78 in the first dataset where the best performance was 0.89 and in case of 2nd dataset we had the performance of 0.71. We also researched and experimented that using this method i.e. cluster results how many of the clusters can get us close the best performance of the neural network.

At the end, it looks like that no one approach is fit for all the kinds of different problems and unsupervised learning can be extremely helpful in case when we don't have any information. For example, in case of credit card default, even if the clustering results had 21 % of the default predicted as a separate cluster it helps business a lot even if 21 % of loss can be controlled.