Unsupervised Learning and Dimensionality Reduction

Younes EL BOUZEKRAOUI
ybouzekraoui3@gatech.edu

***Abstract*—The Purpose of this project is to explore unsupervised learning algorithms, two clustering methods, k-means and Expectation maximization are tested to two data sets, Four dimensionality reduction algorithms are applied, and then the performance of the clustering methods is tested again on the resulting data sets, finally , a neural network from a previous assignment is tested and evaluated after the dimensionality reduction.**

1 TOOLS AND ENVIRONMENT

The programming language used in this assignment is Python with Numpy, Matplotlib, Pandas, Scipy and Sckit learn libraries.

The programming environment is Jupyter Notebook, and the code is available in a jupyter-notebook format and also in html format.

2 DATASETS

2.1 Dataset1 : Gene expression cancer RNA-Seq

This Dataset is subset of the RNA-Seq (HiSeq) PANCAN data set, It is the extraction of gene expressions at random from patients with various forms of tumors: BRCA, KIRC, COAD, LUAD and PRAD.

The Data set contains 801 instances stored row-wise. and 20531 attributes the RNA-Seq gene expression levels measured by illumina HiSeq platform.

In this dataset we already have the true label of each instance and we know that there are 5 classes , and we will use them to evaluate our clustering algorithms.

We have choose this dataset because it contains a very high number of attributes and we want to see how the algorithms will perform in this situation, some algorithms are not able to run on this dataset because it will take a long time, so this dataset is used only in some specific algorithms bellow

2.2 Dataset2 : Credit Card Dataset

This Dataset shows the usage patterns of around 9000 active credit card users over the last six months. It contains 8950 instances and 18 attributes which are behavioral characteristics at the consumer level which is a small number of attributes in comparison with the previous dataset.

In this dataset we do not have the labels so we are exactly in the framework of unsupervised learning and we don't know how many label classes do we have.

2.3 Dataset3 : Churn Dataset

This is the dataset used in the Assignment1.
Churn Prediction for a Telecommunication company by analyzing all relevant customer data and how they use the service and predicting which customers are most likely to cancel a subscription, it contains 2666 instances with both numerical and categorical data , and it contains 77 attributes after One-Hot Encoding the categorical data.

We have the true labels that are (True/False) and we will use them to evaluate the performance of our clustering algorithms.

3 CLUSTERING ALGORITHMS

In order to study clustering algorithms , there is multiple metrics that helps evaluate the clustering method and get insight on the performance of the algorithm.

- **The Elbow SSE method :** The Elbow method is a visual way for determining the consistency of the optimal number of clusters by evaluating the difference of the sum of square error (SSE) of the clusters, the biggest deviation forming the angle of the elbow indicates the best cluster number.
- **The silhouette coefficient :** A measure of cluster cohesion and separation, It uses The distance between the data point and the other points in the cluster. to determine how well a data point fits into its given cluster.

3.1 K-means clustering

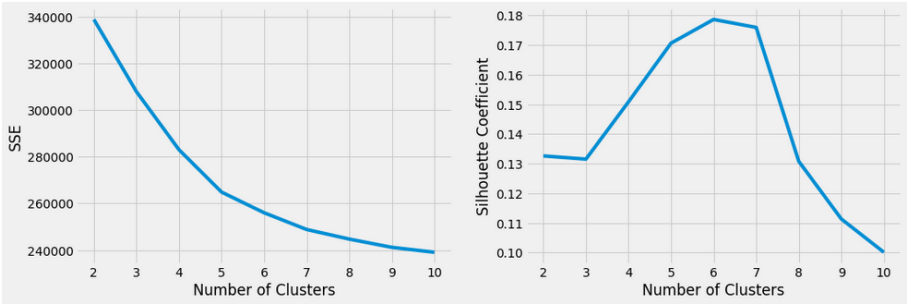
We used the Kmeans function of the Sklearn library to test and evaluate this clustering algorithm, we tested the algorithm for different values of the number of clusters (from 1 to 11)

In order to find the optimal number of clusters we used two methods/metrics :

- The elbow method
- The silhouette coefficient

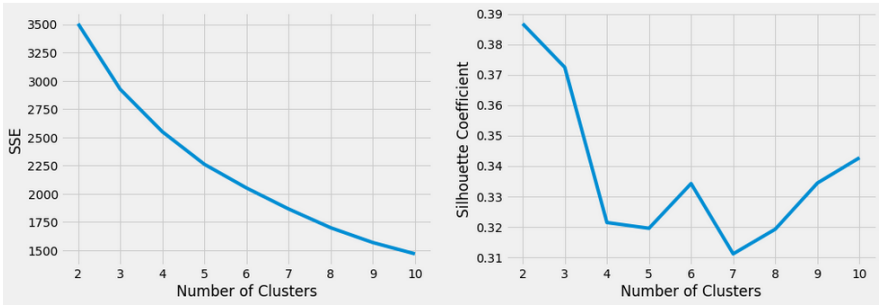
Rather than one being chosen over the other, we used the two methods as complementary evaluation tools, we ran the k means algorithms with multiple values of k and saved the SSE and Silhouette coefficient metric.

3.1.1 *Dataset1*



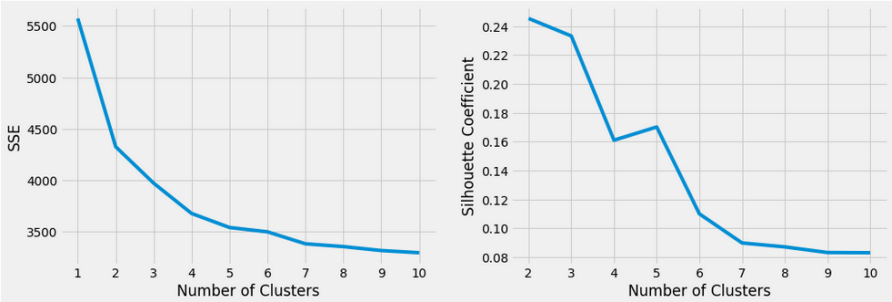
We we can from the figures bellow there is an elbow at 5 clusters , this point is a reasonable trade-off between the number of clusters and the error. However The Silhouette Coefficient gives us a different results it reaches it's maximum at 6 clusters but it's value for 5 and 7 clusters is also high , so in order to use the two methods complementary, we keep tree values to evaluate the clustering algorithm (5 clusters, 6 clusters and 7 clusters).

3.1.2 *Dataset2*



For the second dataset we can see that there is an elbow at 3 clusters however the Silhouette Coefficient reaches it's maximum at 2 clusters , so let's keep both the values (2 clusters and 3 clusters)

3.1.3 *Dataset3*



On the Churn the results of both the metrics seems coherent, we can see an elbow of the SSE graph at 2 clusters and the silhouette coefficient also at it's maximum at 2 clusters

3.2 Expectation Maximization

To test and evaluate the expectation maximization algorithms we used the Gaussian-Mixture function of Sklearn.

And in order to evaluate and find the best number of clusters, we used the Silhouette Coefficient and the AIC/BIC scores AIC and BIC are used to select the best model and in our case find the best number of clusters. AIC means Akaike's Information Criteria and BIC means Bayesian Information Criteria. The AIC seeks to find the model that best describes a high-dimensional reality that is unknown. This means that reality is never among the candidate models under consideration. BIC, on the other hand, seeks out the TRUE model among the candidates.

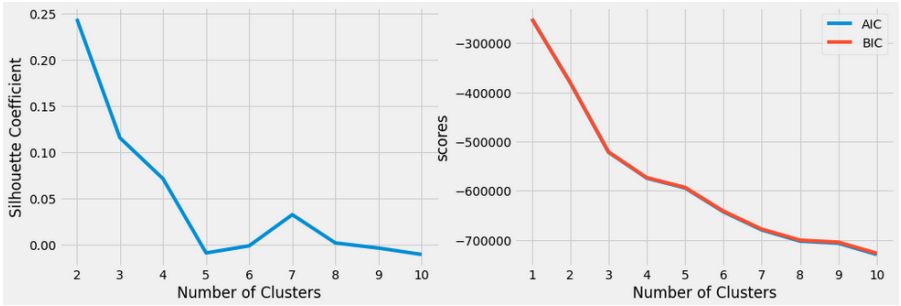


Figure 1—Silhouette Coeficient and AIC/BIC scores for dataset 2

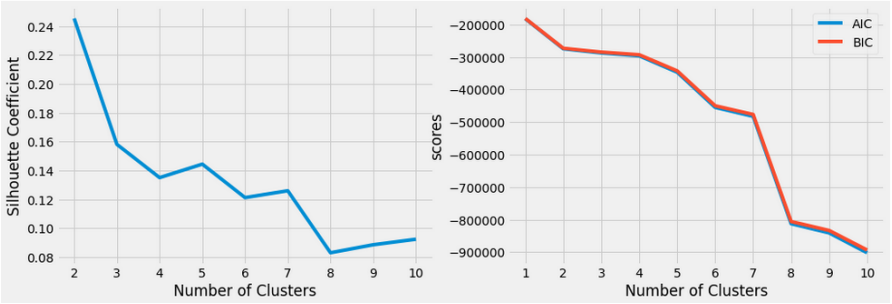


Figure 2—Silhouette Coeficient and AIC/BIC scores for dataset 3

The same analysis as before we prefer higher silhouette coefficient, but we prefer lower AIC and BIC

4 DIMENSIONALITY REDUCTION ALGORITHMS

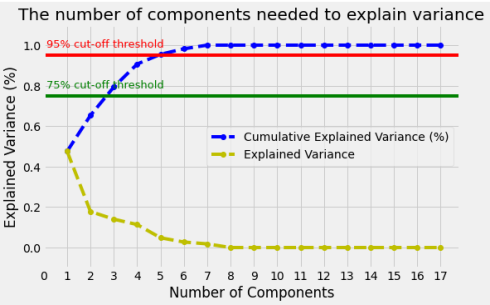
Our ultimate goal is to develop simple models that can be performed rapidly and explained easily. Our model becomes more complex as the number of features increases, and the explainability diminishes. There exists multiple methods to reduce the high dimensionnality, and in this section we will explore 4 algorithms for Dimensionality reduction

4.1 PCA

Principal Component Analysis is an excellent way for reducing the dimensionality of the data when dealing with these complex data-sets, It reduces a large number of potentially correlated variables to a smaller number of uncorrelated variables called principle components.

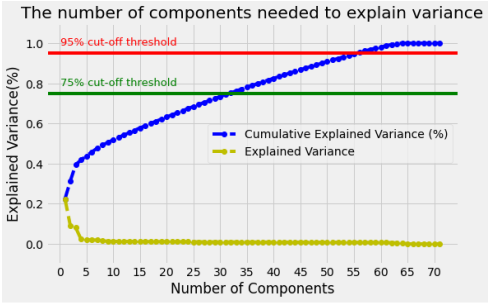
In order to evaluate the PCA algorithm and choose the optimal number of principle components to keep we used the explained variance Ratio metric, which evaluates percentage of variance that is attributed by each of the selected components and this shows the usefulness of each principal component.

4.1.1 *Dataset2*



We can see from the figure above that 5 principal components are sufficient to cover 95% of the information and even 3 components can achieve 75% of the explained variance, so we were able to go from 17 attributes to only 5 or 3 components

4.1.2 *Dataset3*



For the third Dataset , PCA is not realy performing well in term of dimensionality reduction , as we can see the cumulative explained variance needs higher number of components 55 to cover 95% percent of the information and around 35 components to cover 75% of the information which is high in the case where we have 77 attribute.

4.2 ICA

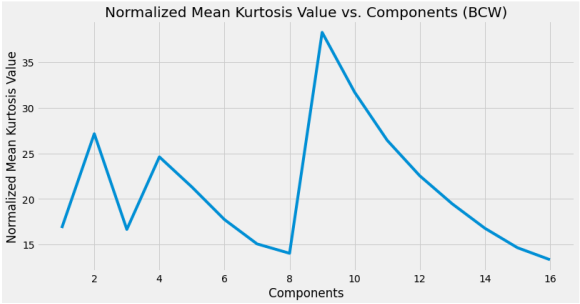
Independent Components Analysis (ICA) is an interesting method for dimensionality reduction. It assumes that each data sample is made up of a number of independent components, and it seeks to identify these components.

We tested the FastICA function of Sklearn on the data-sets and used the normalized Mean Kurtosis value as a metric to find the best number of components.

The essential key for estimating the ICA model is the Non-gaussianity thus the Mean

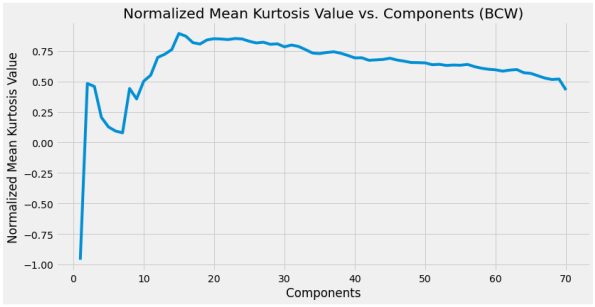
Kurtosis value is a quantitative measure of non-gaussianity of the components. This metric will help us find component that maximize the Kurtosis which means components that are highly non-normally distributed.

4.2.1 *Dataset2*



We can see from the figure above that the normalized Mean Kurtosis reaches it's maximum value at 9 components. In comparison to the PCA where we were able to keep 95% of the information with only 5 components, the PCA is giving better results in dimensionality reduction than the ICA, this is what we will confirm later when we will do the clustering after the the dimensionality reduction.

4.2.2 *Dataset3*



The observation as before, the normalized Mean Kurtosis reaches it's maximum value at 15 components, which is a very good value in comparison to the PCA were we needed much higher number of components

4.3 Randomized Projections

Random projection is a simple and computationally efficient method of reducing data dimensionality by exchanging a regulated proportion of error for faster processing and smaller models.

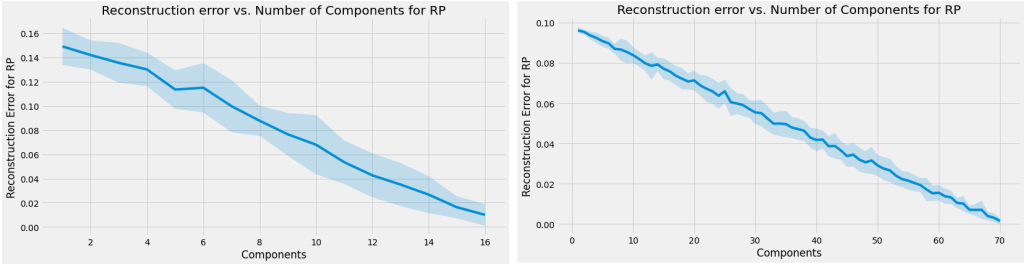


Figure 3—Reconstruction Error vs Number of Components for Dataset2 on the Left and dataset3 on the Right

In Both Data sets we can see that the reconstructed error decreases linearly with the number of component so this makes it difficult to choose the number of components to choose, because we cannot choose the lowest reconstruction error as it corresponds to a high number of components (All the attributes), and let's remind us that the purpose of this part is to reduce the dimensionality.

So I decided to choose a value for number of components which matches the k value in PCA , for at least performing comparisons.

4.4 Sparse PCA

To handle clustering and feature selection challenges, we apply sparse principal component analysis (PCA). Sparse PCA looks for sparse factors, or linear combinations of data variables, that can explain the most variance in the data with the fewest number of nonzero coefficients.

This method can be evaluated in the same way as PCA by looking at the explained variance by each component.

The Sparse PCA can be more useful than PCA is some cases. Ordinary PCA has the drawback that the principal components are almost always linear combinations of all input variables. Sparse PCA solves this problem by identifying linear combinations with only a few input variables.

5 CLUSTERING AFTER DIMENSIONALITY REDUCTION

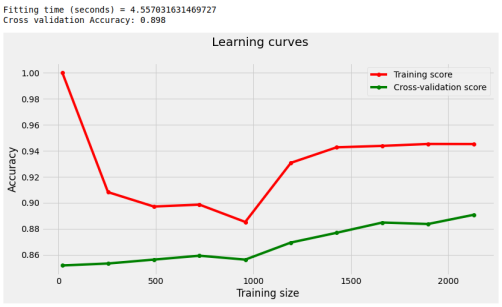
In this section, I conducted experiments with all possible reduction and clustering method combinations. Because there are so many outcomes, I decided to focus on the

most interesting and unique ones.

6 NEURAL NETWORK WITH DIMENSONALITY REDUCTION

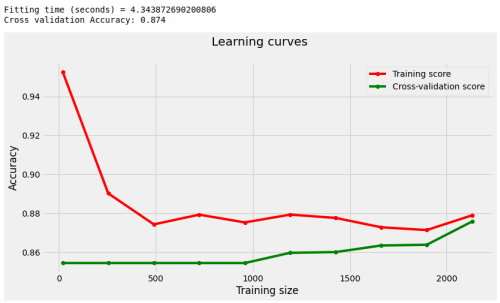
In this Part we apply the neural network implemented on Assignment 1 that contains 3 hidden layers of 64 unit and Relu activation function. The Cross Validation Accuracy obtained on Assignment 1 was 0.880.

6.1 Reduction using pca



Using the PCA reduction we were able to obtain a higher accuracy 0.898 , this might be explained by the fact that PCA does not conserve all the information but only 75% -80% in our case with 35 principal components instead of 77 attributes, and helps to avoid over-fitting (this can be seen from the learning curves) because the learner tries to learn only the important information that the PCA kept .

6.2 Reduction using Ica



However using the reduction with ica the cross validation accuracy is lower than it's value before the reduction, and also the graph shows that the cross validation score can be improved, and this might be a sign for underfiting, and that ICA removed some relevant informations for the learner

7 NEURAL NETWORK WITH CLUSTERING

In this part we did the clustering using kmeans and EM of the dataset resulting from PCA reduction, and tested again the performance of the neural network

Figure 4—Learning curves of The neural network after applying k-means [Left figure] and EM [Right figure]

We can see from the figures above that both K-means and Expectation maximization (EM) Clustering methods gave the same accuracy that before but in term of time performance , the neural network was fast this time , this can be explained by the fact that the columns that are added to the data-sets containing the cluster assigned to each instance actually helped the neural network to do the classification, and that's why it performed better in terms of time. I was expecting that it will also perform better in term of cross validation accuracy, but it was not the case and this might be due to an incorrect labeling using the clustering previous clustering methods, or maybe an other reason , or maybe my expectation is incorrect.

8 CONCLUSION

In conclusion we can say that The major takeaway from the preceding experiments is that dimensionality reduction and clustering algorithms can reduce the processing time of algorithms while preserving accuracy. Further study on the data-sets is required to make uderstand exactly what makes a method better than the other.

"A baby learns to crawl, walk and then run. We are in the crawling stage when it comes to applying machine learning."
–Dave Waters–