Exploring Clustering and Dim Redn Algorithms

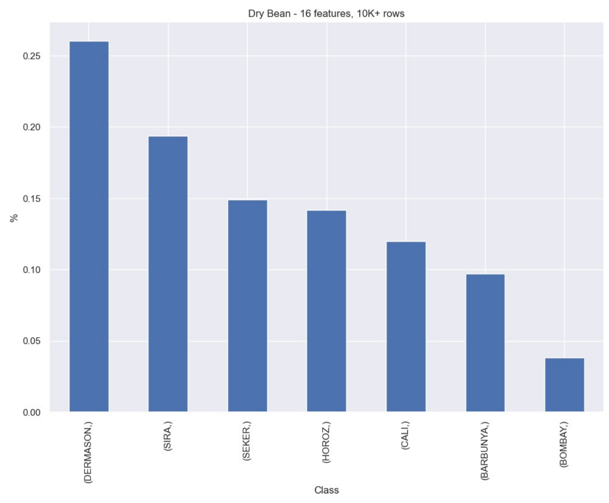
By: Pburra6

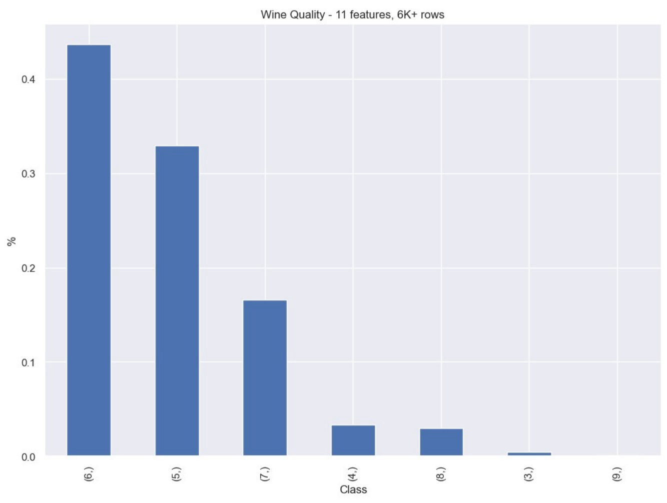
Datasets

For the project two datasets were chosen *Dry Beans* and *Wine Quality* sourced from UCI ML data repo. The datasets have all numeric features to circumvent requirement of any special treatment for categorical features. The datasets also have a target variable defined to help verify the unsupervised clustering against ground truth.

In pair plots of *Dry Beans* exhibit clear clustering and strongly correlated features. There are no outliers. *Wine Quality* data set however does not demonstrate clear clustering across features, no strong correlation and has outliers. This should provide for a good contrast on the algos performance.

Datasets are also imbalanced hence all the groups may not be identified in clustering due to the lack of datapoints.





Clustering

I chose KMeans and Gaussian Mixture Models (GMM) to explore the difference in behaviour when the datapoints are:

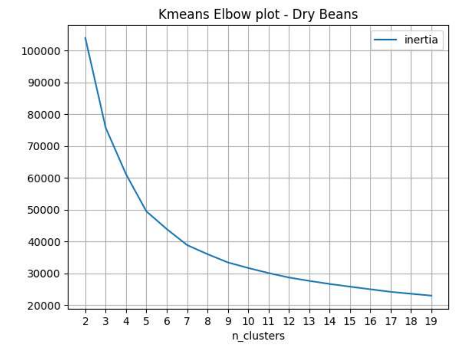
1. Not circularly clustered.
2. Are very close making clustering difficult.

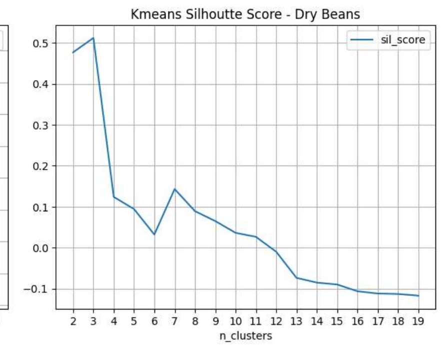
KMeans works well on groups that are circular due to Euclidean distance metric so has limitations unless there are odd shape groups. GMM can adapt using elliptical distb. GMM also deals better with clusters that are too close or overlap. Overall, I expect Gaussian to perform as well as, if not better than KMeans.

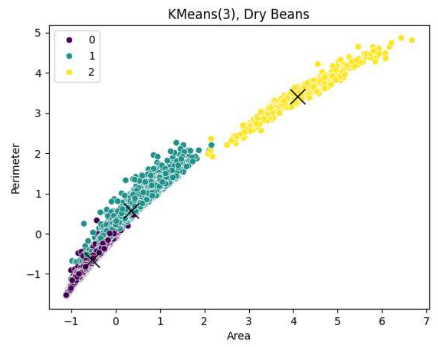
Dry Beans

Neither managed to capture the ground truth of the 7 types of seeds, although we see a small spike in silhouette scores at K=7. From the data visualisation – we can see that most clusters overlap at current dimensions. Neither clustering method performs well. The comparison of the clustering against original label across all features is available in the python notebook – (*kmeans/exp_max)_dry_beans.ipynb*

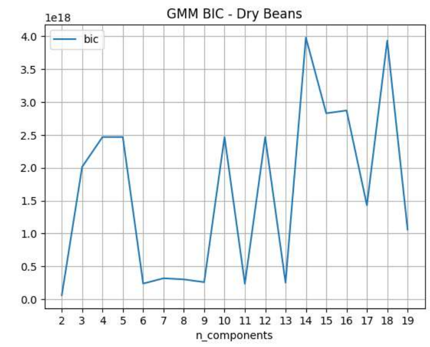
KMeans

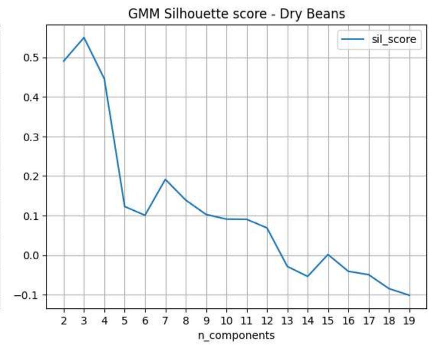


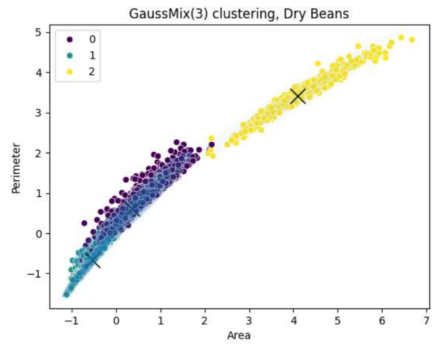




Gaussian Mixture



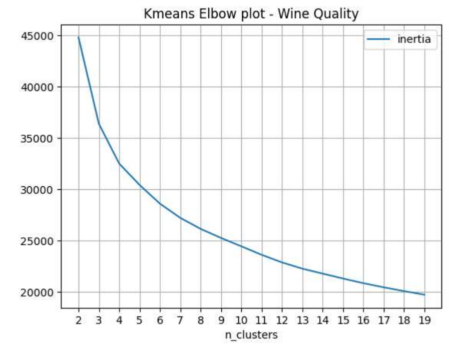


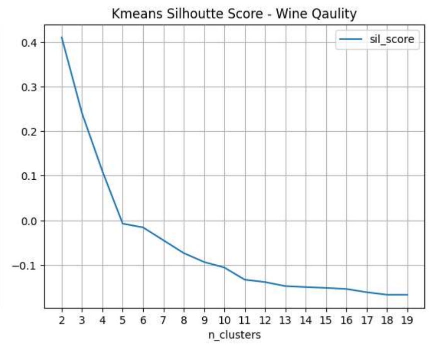


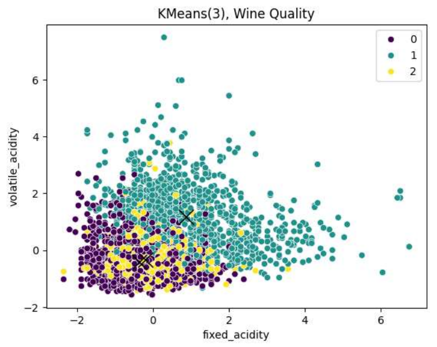
Wine Quality

Wine Quality data has challenge of no obvious grouping and significant overlap of the groups. This lack of pattern challenges both clustering algorithms. Silhouette score starts low (<0.5) and goes negative. The comparison of the clustering against original label across all features is available in the python notebook. - – (*kmeans/exp_max)_wine_quality.ipynb*

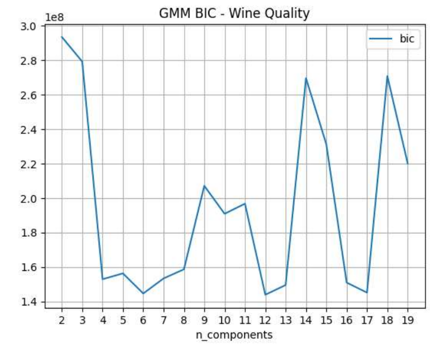
KMeans

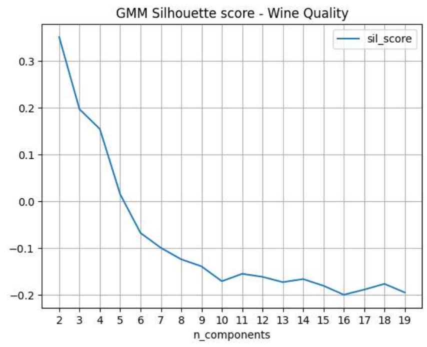


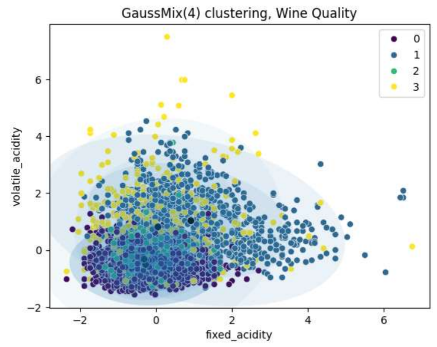




Gaussian Mixture







Dimension Reduction

Dimension Reduction hopes to tease out patterns within data by identifying the underlying variables.

PCA wont work well when the features have nonlinear relation or no clear relationship at all. This is evident for the Wine Quality data where PCA does not result in any distinguishable clusters. Reconstruction error also remains high until many components have been included. PCA does however clearly segregate the clusters for Dry Beans dataset owning to its correlated features.

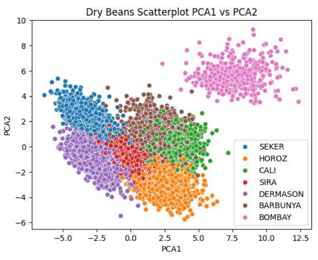
ICA has limitations if the sources have gaussian distribution. This again leads to poor clustering for Wine Quality data where the distribution of the features is approximately gaussian in pairplots.

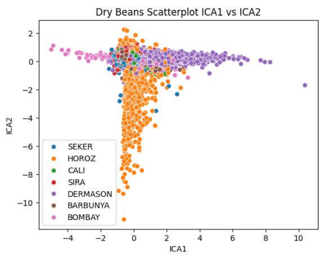
Random Gaussian Projection and t-SNE work well when the initial number of features are large, especially t-SNE which works best for features greater than 50. The datasets considered here may not be sufficiently high dimensional to see the benefits. t-SNE requires significant tuning with perplexity and learning rate. t-SNE has been slowest of the clustering algorithms as well.

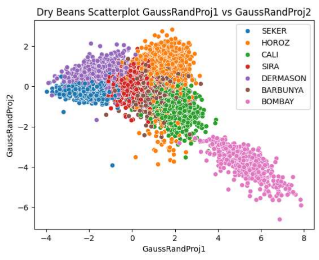
An interesting observation can be made for Wine Quality data. Considering there is no clear pattern in original and dim reduced data between features and class, one can question if the classification is random. i.e Perception wine quality may depend on other factors (marketing, perceived quality based on age) than physical properties.

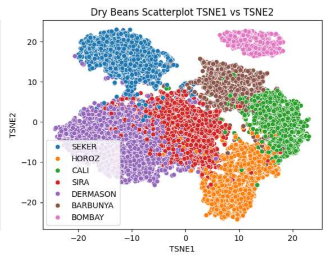
Dry Beans

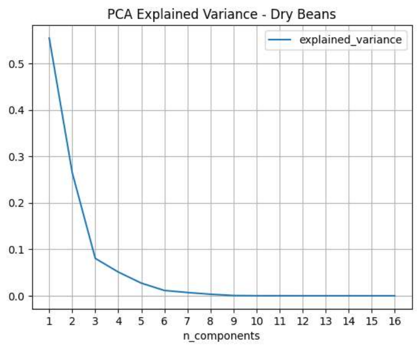
TODO: Rerun the random projections multiple times!

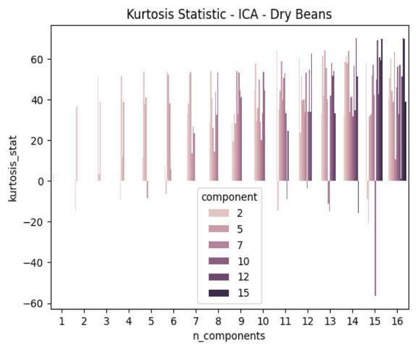


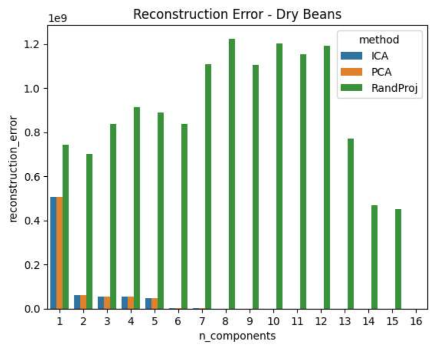




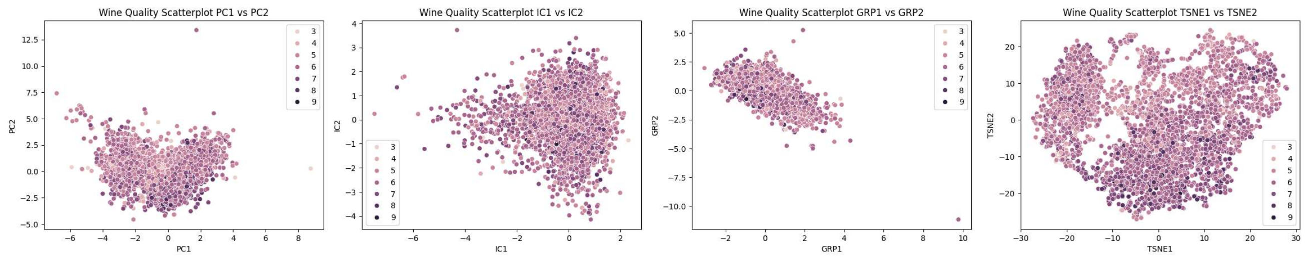


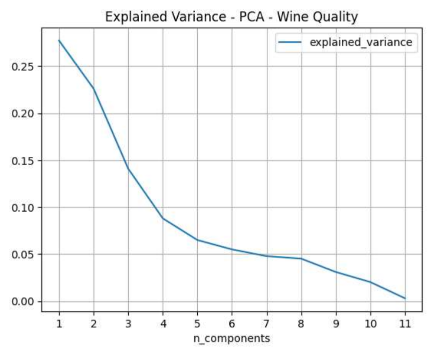


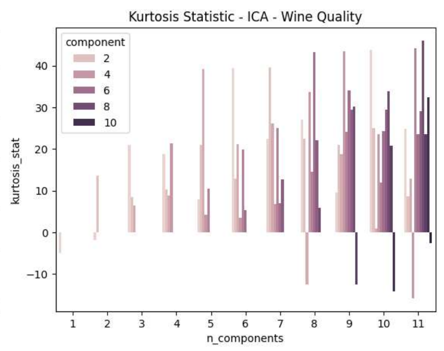


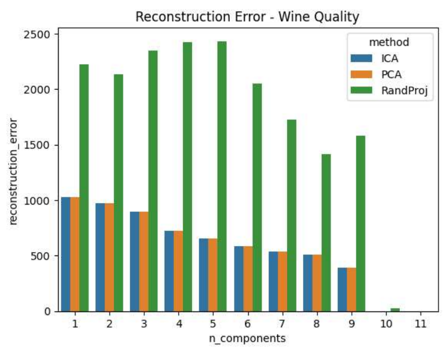


Wine Quality









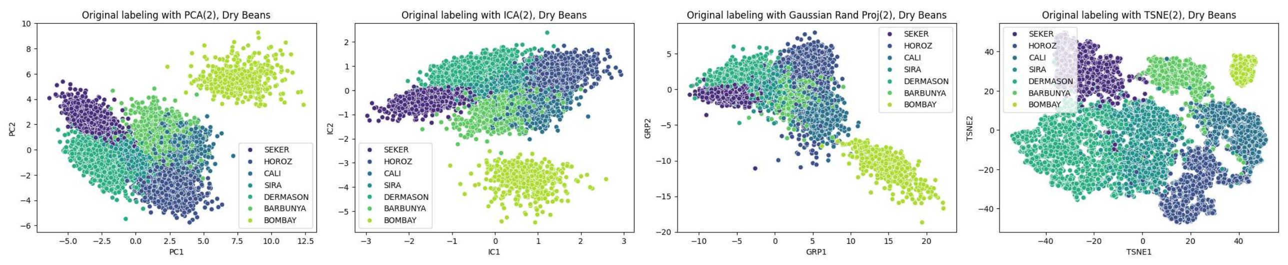
Clustering with Dim reduction

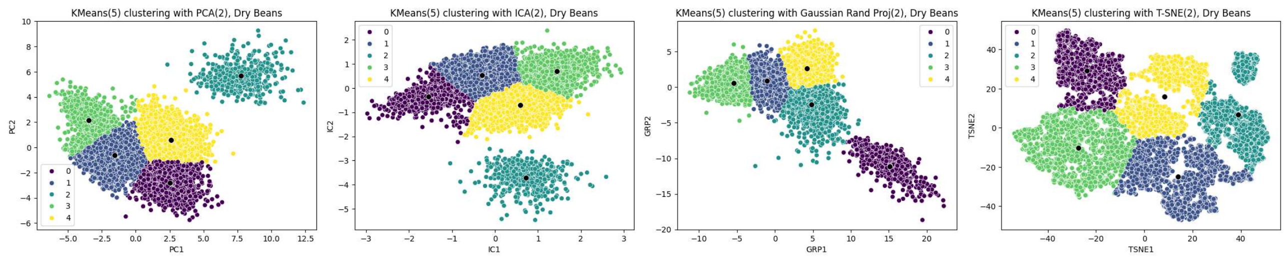
Following the performance of the dimensionality reduction GMM offers the best performance for the Dry Beans data after the PCA or ICA. The key reasons being:

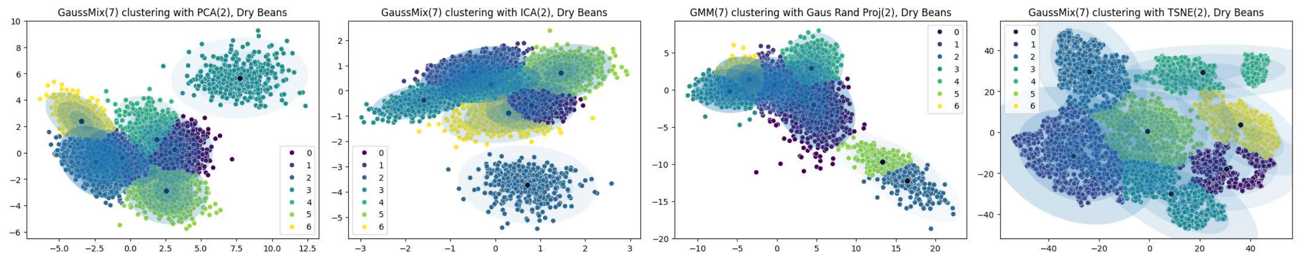
1. The clusters are well defined but skewed shape of cluster is where GMM outperforms KMeans
2. The clusters are very closely spaced with some overlap – this again gives soft clustering algos an edge.

Dry Beans

Original vs KMeans vs GMM



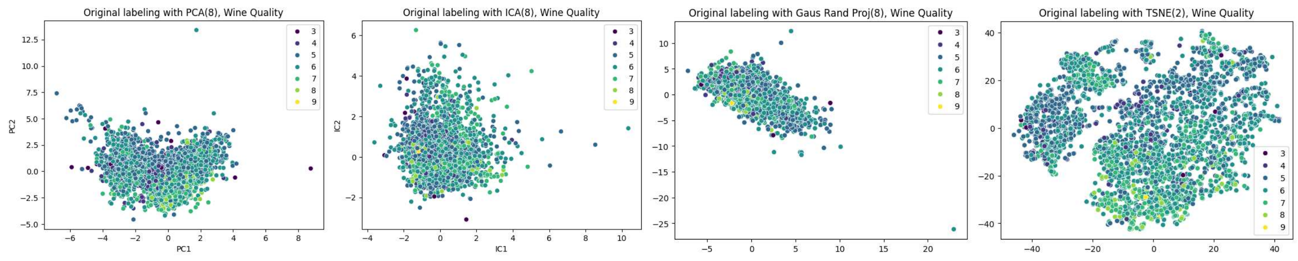


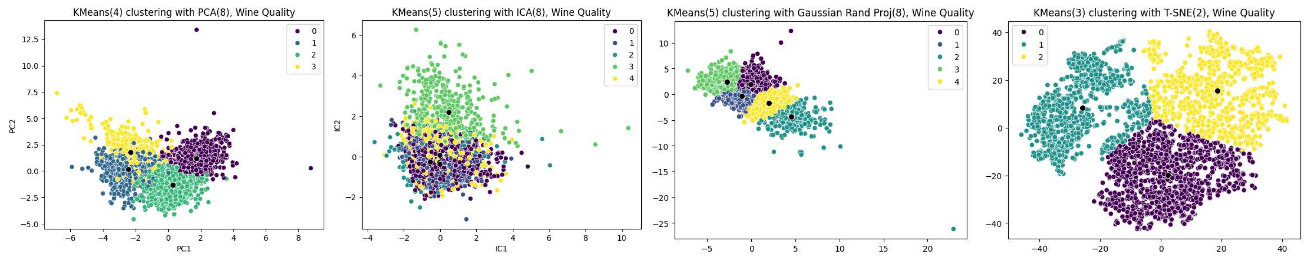


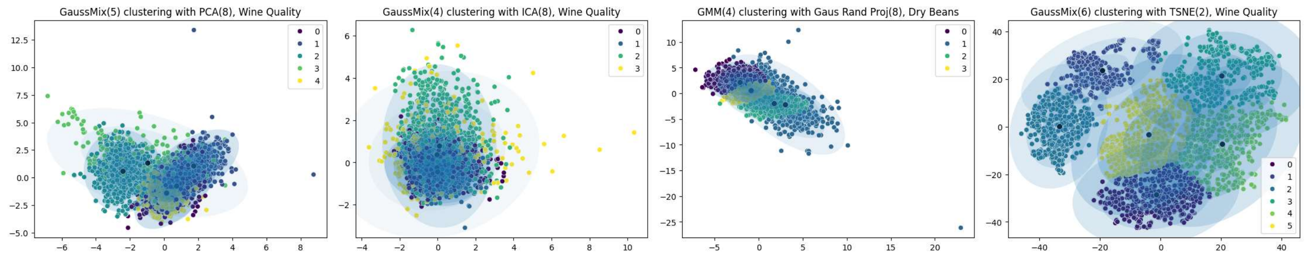
Wine Quality

Dim reduction required almost all the features (8) to keep the reconstruction err low. The projected data does not demonstrate any significant relations hence the clustering algorithms do perform well.

Original vs KMeans vs GMM







Neural Networks with Dim reduction

Dry Beans

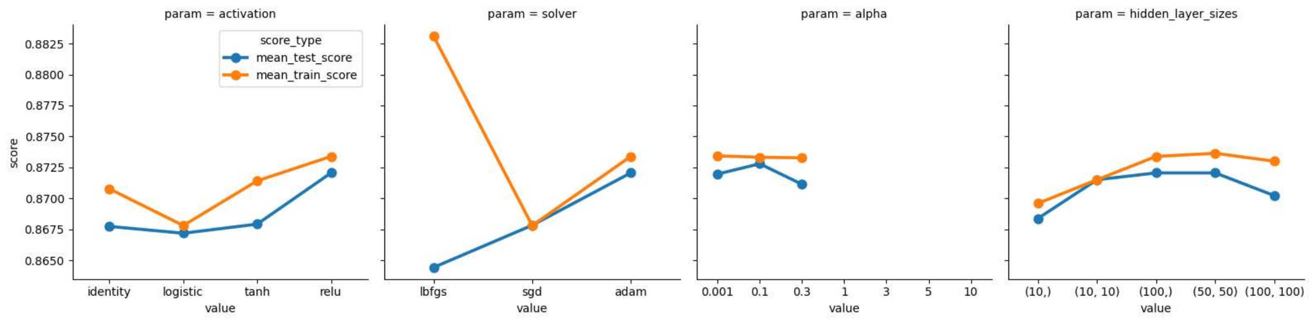
Summary of the best model, train times and scores

Model	Grid Search	Best Param	Test Score	Train Score
NN	20 min	`activation: 'relu', alpha: 0.1, hidden_layer_sizes: (100,), solver: 'adam'`	0.9305	0.9232
PCA(16) + NN	22 min	`activation: 'relu', alpha: 0.001, hidden_layer_sizes: (50, 50), solver: 'adam'`	0.9309	0.9251
PCA(2) + NN	4 min 40s	`activation: 'relu', alpha: 0.1, hidden_layer_sizes: (100), solver: 'adam'`	0.8728	0.8737
ICA(6) + NN	21 min	`activation: 'relu', alpha: 0.1, hidden_layer_sizes: (150,), solver: 'adam'`	0.9310	0.9214
ICA(2) + NN	7 min	`activation: 'relu', alpha: 0.001, hidden_layer_sizes: (100,100), solver: 'adam'`	0.8724	0.8748
GRP(8) + NN	26 min	`activation: 'tanh', alpha: 0.001, hidden_layer_sizes: (50, 50), solver: 'adam'`	0.9031	0.9023
GRP(2) + NN	6 min	`activation: 'tanh', alpha: 0.001, hidden_layer_sizes: (100,100), solver: 'adam'`	0.8066	0.7973
t-SNE(2)+ NN	4 min	`activation: 'tanh', alpha: 0.001, hidden_layer_sizes: (50, 50), solver: 'adam'`	0.9098	0.7535

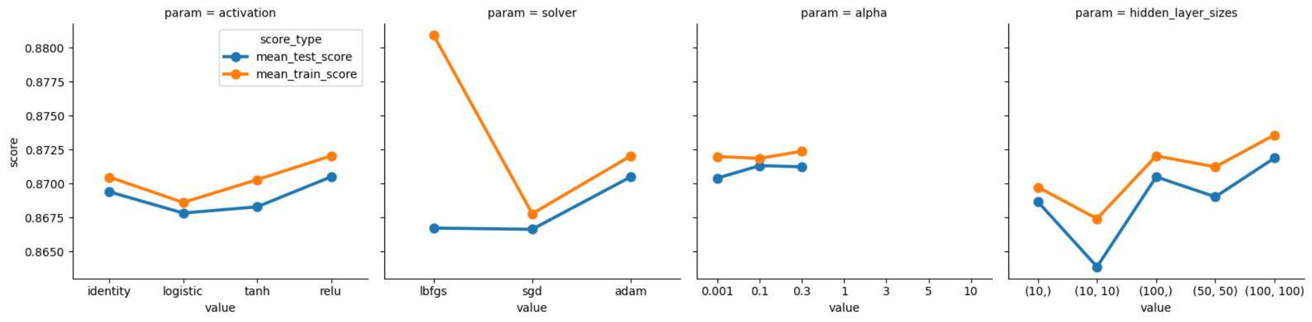
The expectation of applying dimensionality reduction is an improved train time and score of the neural network. The more the components are retained the better the testing score, but training time is as slow as original NN. Dropping to just 2 components slashes the training to a fifth without a significant compromise in train and test scores.

Test scores for GRP and t-SNE however saw a significant drop in test score when using 2 components.

PCA(2)

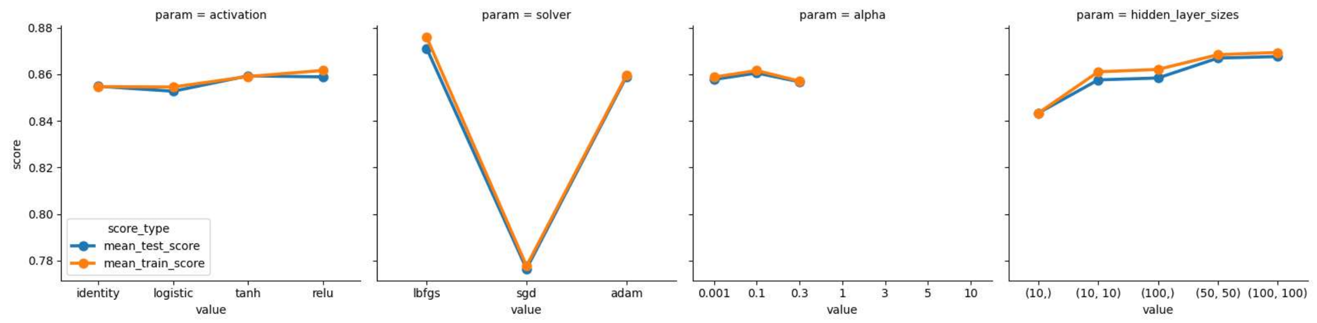


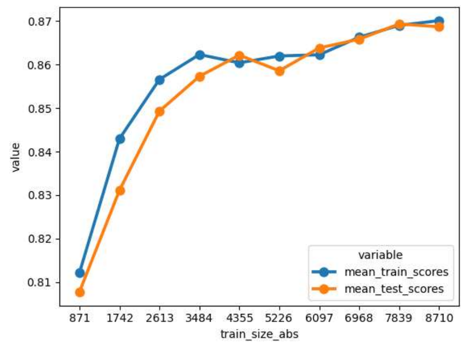
ICA(2)



Neural Network with Clustering

KMeans





References

https://towardsdatascience.com/kmeans-hyper-parameters-explained-with-examples-c93505820cd3

https://www.analyticsvidhya.com/blog/2021/05/k-mean-getting-the-optimal-number-of-clusters/

https://towardsdatascience.com/gaussian-mixture-model-clusterization-how-to-select-the-number-of-components-clusters-553bef45f6e4

https://towardsdatascience.com/gaussian-mixture-model-clearly-explained-115010f7d4cf

https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

https://scentellegher.github.io/machine-learning/2020/01/27/pca-loadings-sklearn.html

https://towardsdatascience.com/interesting-projections-where-pca-fails-fe64ddca73e6

https://towardsdatascience.com/introduction-to-ica-independent-component-analysis-b2c3c4720cd9

https://medium.com/@violante.andre/an-introduction-to-t-sne-with-python-example-47e6ae7dc58f

https://distill.pub/2016/misread-tsne/

https://www.thekerneltrip.com/statistics/tsne-vs-pca/