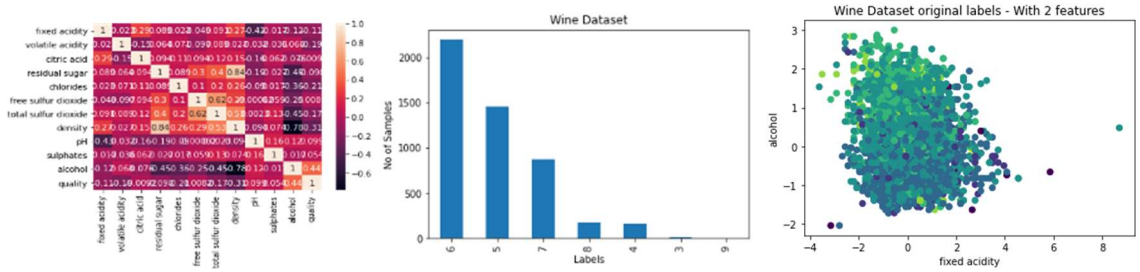
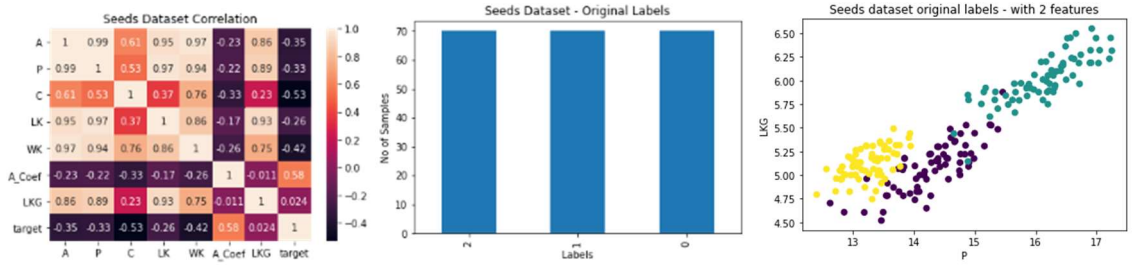
Unsupervised Learning Algorithms

1. DATASETS

White Wine quality dataset from UCI ML repository: This dataset is suitable for dimensionality reduction since it has around 11 features and has 7 labels with different distributions of samples. From the correlation heat map, we can find that many features have equal correlation to the target variable, so it must be interesting to see how the dimensionality reduction and clustering algorithms would perform on the dataset.



Wheat Seeds dataset from Kaggle: This dataset has 7 features and 3 labels with equal distribution of samples. The features have good correlation with the target and there is correlation within themselves too. With its simplicity, it would be easy to visualize the clusters, and better understand the workings of the algorithms.



2. CLUSTERING

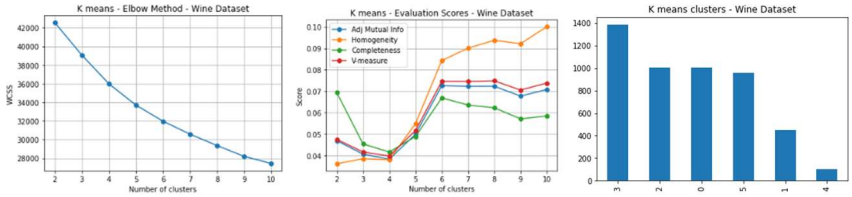
2.1. K means

K means clustering picks k centres, claims the closest points and centres are recomputed by averaging the cluster points across the dimensions, this is repeated until convergence. I have used the sklearn's implementation of KMeans algorithm with Euclidean distance as the distance measure because the error will always go down, not up.

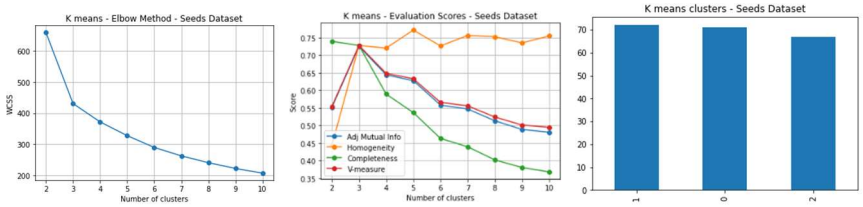
I have used the following evaluation metrics for the clustering algorithms – Adjusted mutual information score gives the agreement/mutual information between cluster and labels. Homogeneity score says how much each cluster includes samples of same label. Completeness score represents if the samples belonging to a label are part of the same cluster. V measure score is the harmonic mean of homogeneity and completeness.

To choose value of k, I have used the elbow method where for k values from 2 to 10, we compute within cluster sum of squares(wcss) and choose the elbow at which wcss lowers.

For the wine dataset, we can see that wcss values level off around k=6 and even all the scores climb up. The evaluation scores are very low around the range of 0.08, clustering doesn't seem suitable, maybe because the features by projection are not able to form distinct groups of the labels.



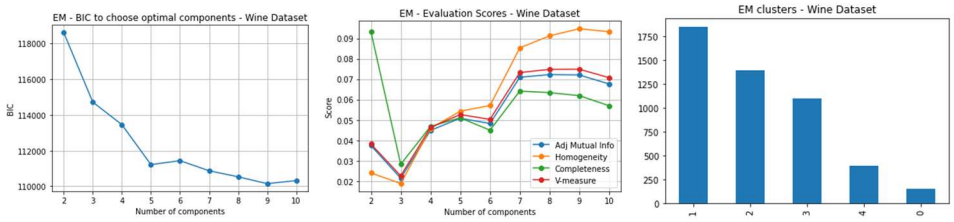
For the seeds dataset, we can see at k=3, the wcss reduces significantly and all the scores are high (0.74). This is because projecting the features clearly separates the samples into groups, thus very well suited for clustering.



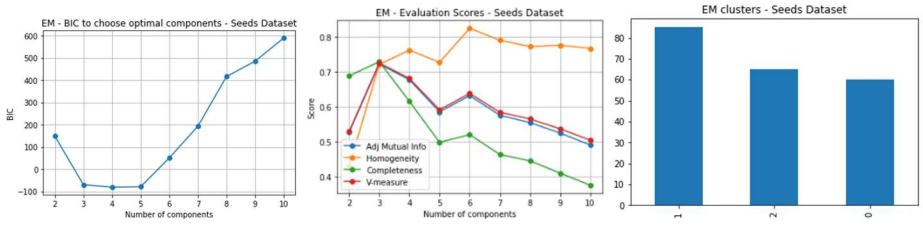
2.2. Expectation Maximization

Expectation maximization algorithms calculates probabilities of points belonging to the clusters i.e., maximizing the likelihood of the data in the clusters. I have used sklearn's Gaussian Misture models. To choose the value of k (number of components) I have used the Bayesian Information criterion(BIC) which says how well our model predicts the underlying distribution of the data. We take the value at which the score lowers significantly.

For the wine dataset, we can see BIC lowers at k=5 and then stabilizes. It is interesting to see the scores reach high at k=7, because BIC adds a penalty as the number of components increase. From the results, we can see EM has done a better job at clustering the points, which comes close to the original labels.



For the seeds dataset, we can see how the BIC graph is inverted, because we have well defined 3 labels. From the result, it does bad compared to K means, this must be because, the outlier points are taken into wrong clusters due to their little variations in their probability scores.



3. DIMENSIONALITY REDUCTION

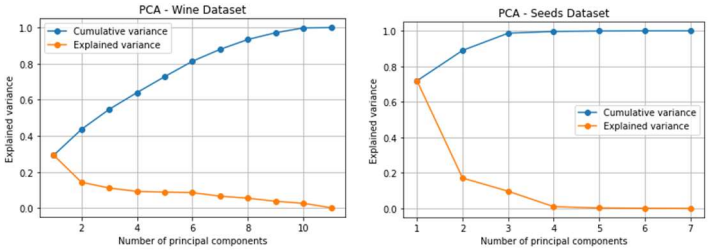
3.1. Principal Component Analysis (PCA)

PCA technique projects the dimensions on a space and tries to capture the variations between the features using lesser dimensions. I have plotted the cumulative variance explained using the number of principal components and

the variance individually explained by each principal component. I have chosen number of components where cumulative variance is more than 85%.

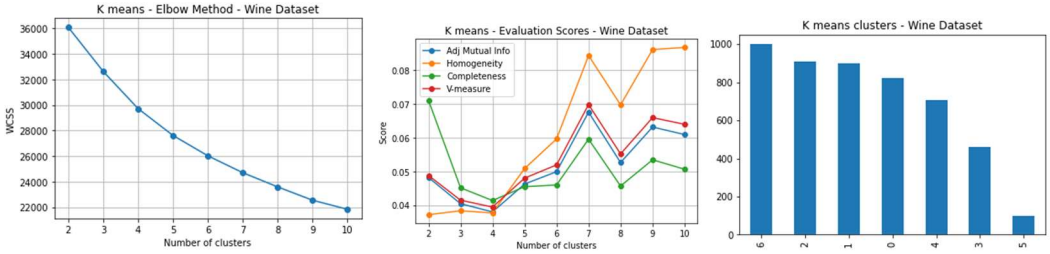
For the wine dataset, with 7 principal components it captures about 88% of the variance in the 11 original features.

For the seeds dataset, with just 2 components it captures about 89% of the variance in the 7 original features. It is very interesting to note here how the PC1 alone captures 70% of the variance. PCA is very effective in such datasets, where without losing much information we can reduce dimensions, and accurately capture the data.

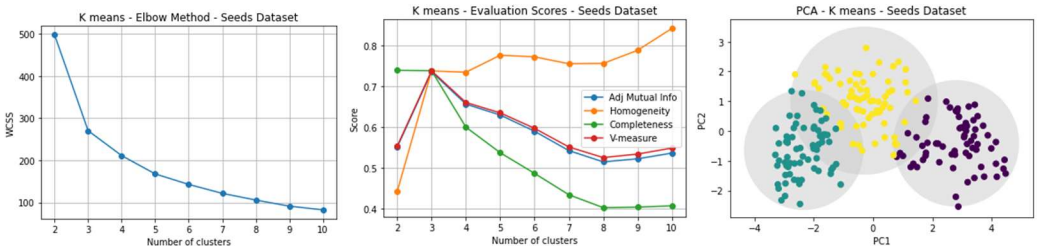


PCA with K means

For the wine dataset, I chose k=7, but clustering still does not depict the original labels, but we are able to get scores similar after dimensionality reduction. In that sense, PCA has done well.

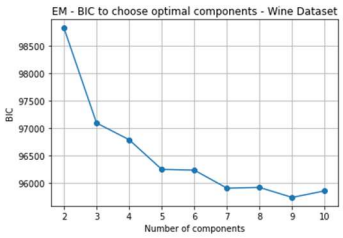


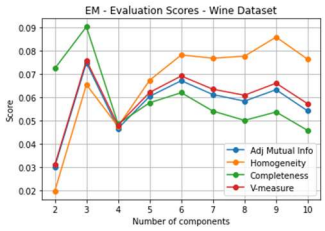
For the seeds dataset, we see very low wcss scores because the data points are clearly defined in groups. I chose k=3, using PCA with just 2 components we could get scores around 0.72. We were also able to visualize the data using PCA. We can see how the 3 clusters are clearly captured by k means.

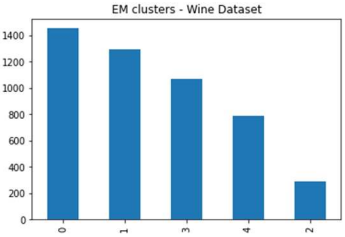


PCA with EM

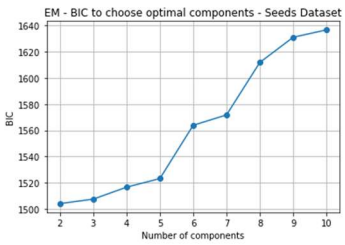
For wine dataset, EM has performed better than K means here as well, but still does not depict the original labels well with very low scores.

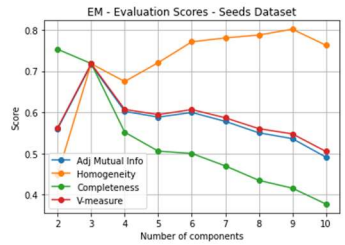


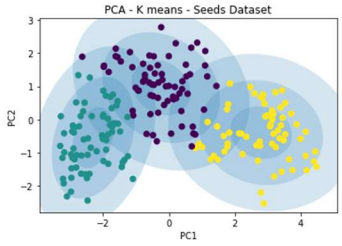




For seeds dataset, using PCA, we can see the distributions and clusters captured by EM. It is very interesting to note how the green labels have a oval shaped cluster, exactly depicting the original distributions compared to Kmeans. This is one of the important advantages of EM over K means, which can capture non-circular clusters.



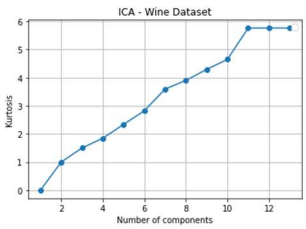


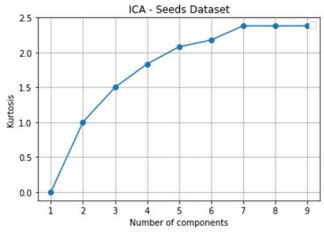


3.2. Independent Component Analysis (ICA)

ICA decomposes the original features and creates components which are statistically independent from each other. We can evaluate the model by measuring the non-gaussianity, which is measured by maximizing the kurtosis. I have used sklearn's FastICA algorithm.

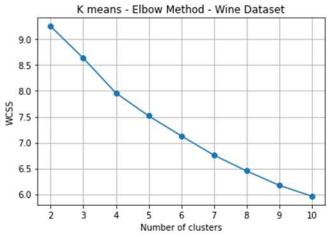
Comparing we can see how kurtosis values are high for the wine dataset and lower in range for the seeds dataset. We can infer that the wine features are more independent of each other originally and are also non-gaussian.

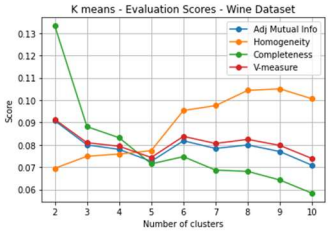


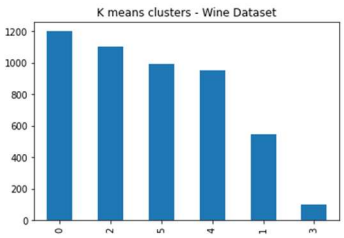


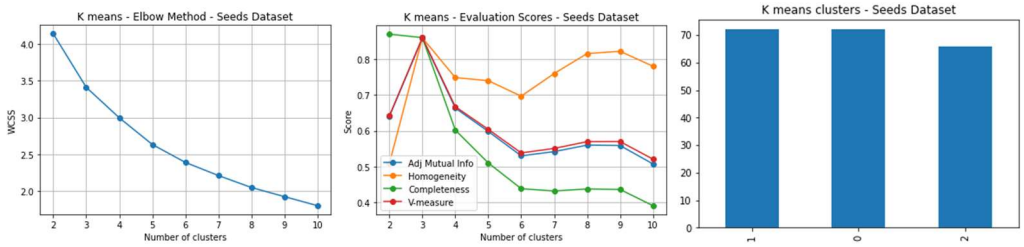
ICA with K means

The wcss scores are very low in range with ICA, which depict ICA generates more compact clusters. The evaluation scores and clusters are similar to previous methods.



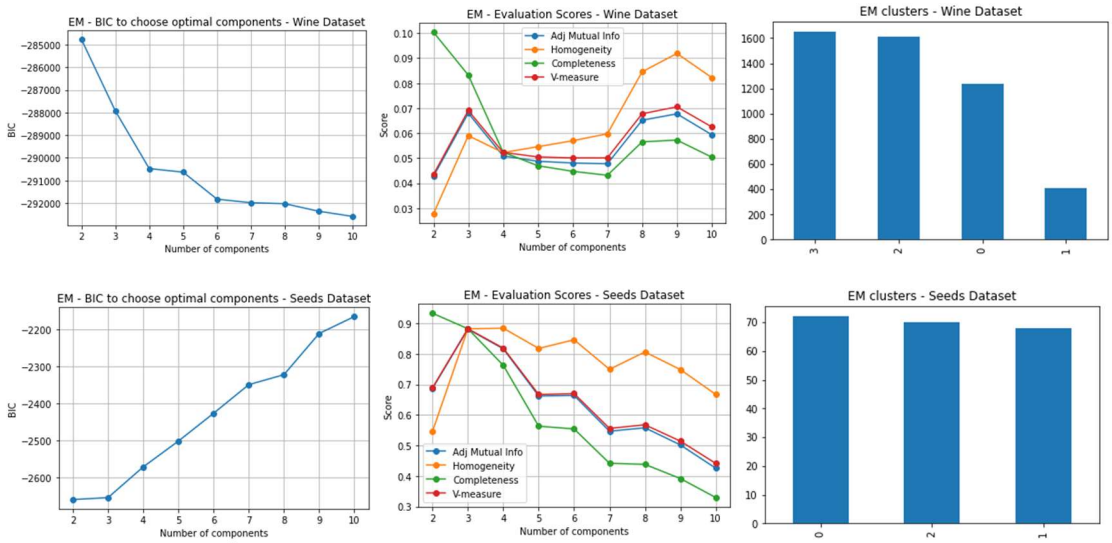






ICA with EM

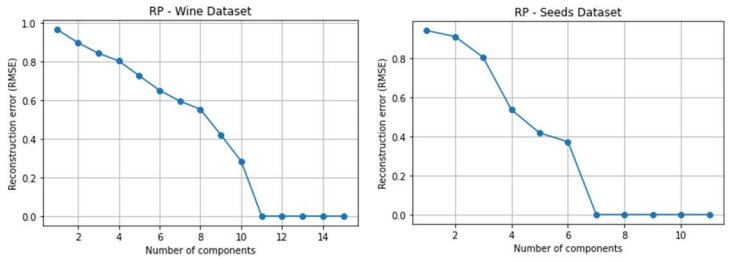
For wine dataset, ICA=10 components has low evaluation scores compared to PCA=7 components. It seems to perform better on the seeds dataset with good evaluation scores.



3.3. Randomized Projections (RP)

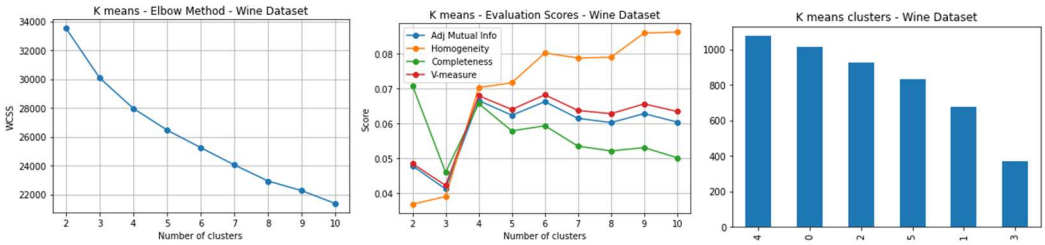
RP is a computationally effective technique which projects the data randomly onto a lower-dimensional space. Reconstruction error(RMSE) between the original features and the newly constructed dimensions is measured. We can note that when no of components = no of features, the error reaches 0.

Here I have chosen, wine=9 components, seeds=5 components using the elbow method.

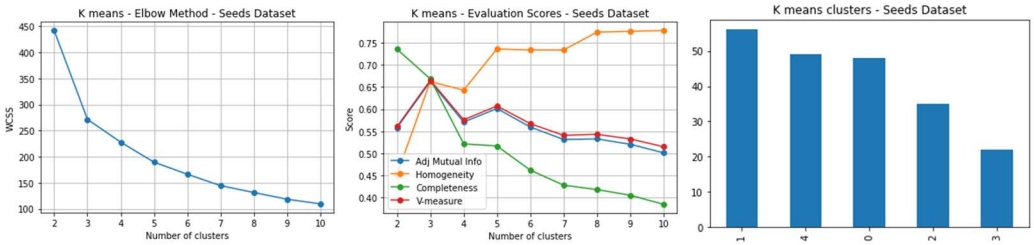


RP with K means

For wine, there is not much difference in performance compared to previous methods.

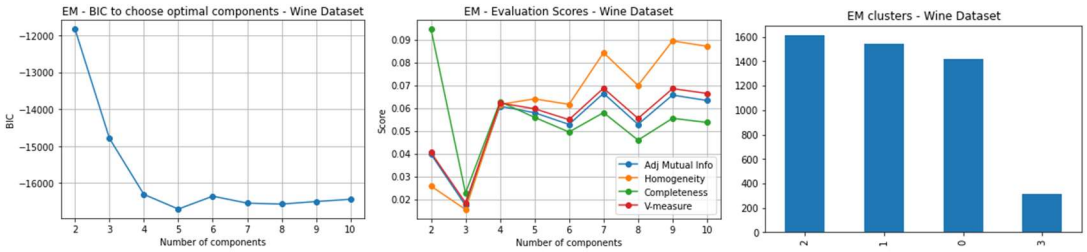


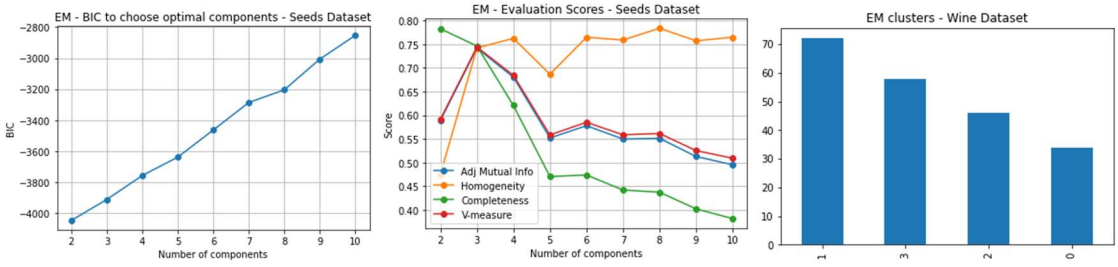
For seeds, the randomness in the projections has not yielded better results. We might have some important information that differentiated the clusters.



RP with EM

For wine, it is interesting how with 4 clusters the BIC almost stabilizes. Even after losing information of about 0.4 by random projections, there are clearly defined clusters with similar scores to previous methods.

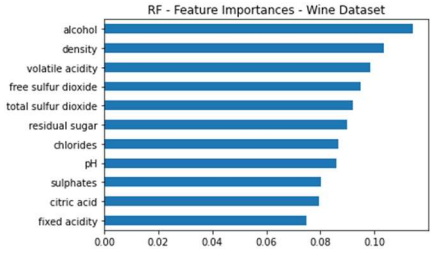


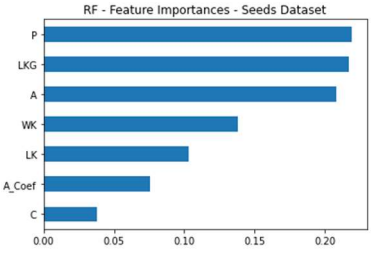


3.4. Random Forest Feature Selection

Random Forests are popular feature selection methods, which by fitting the data into the model, gives feature importance scores. I have used the SelectFromModel method from sklearn to select the features which chooses features over the threshold which is the mean of their feature importance scores.

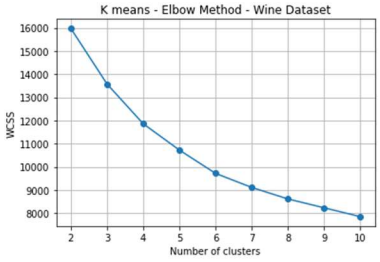
For wine, the features selected were 'volatile acidity', 'free sulfur dioxide', 'total sulfur dioxide', 'density', 'alcohol' which we can see high feature importances. For seeds, features selected were ‘A', 'P', 'LKG'.

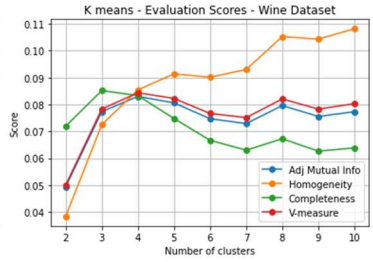


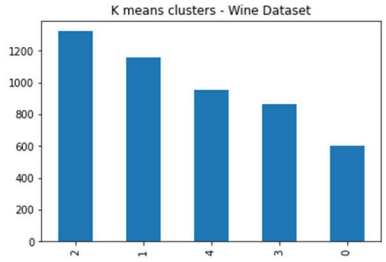


RF with K means

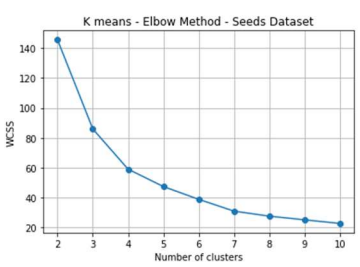
For wine, there is no significant difference in the performance of the k means with RF feature selection.

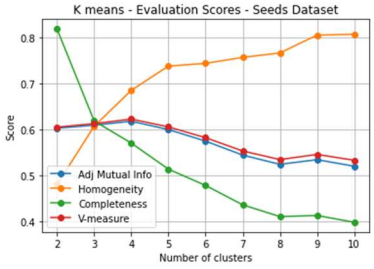


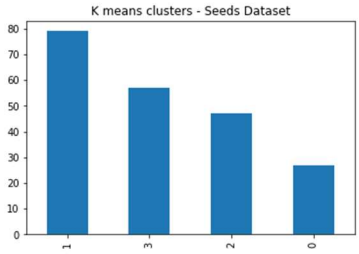




For seeds though, the performance is worse compared to other methods. The main reason could be, there is some significant information left in the other features which contribute to differentiating the clusters.

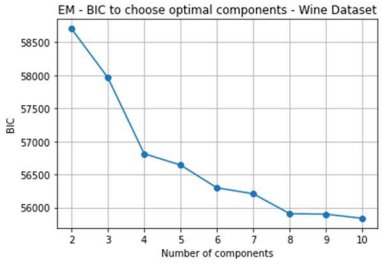


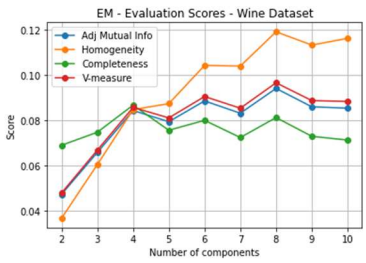


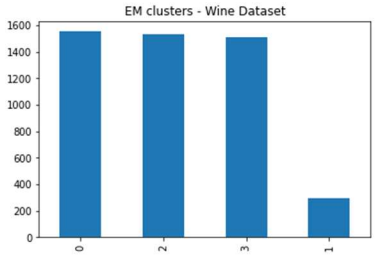


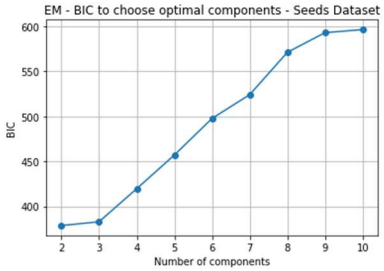
RF with EM

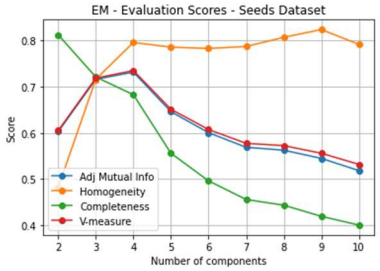
For wine, the results depict how the features selected group the data majorly over 3 clusters, though it doesn't come any closer to the original labels, we can note here that the decision trees of the random forests must have been built on with these most important features. The remaining features must be distinguishing the data implicitly inside these clusters.

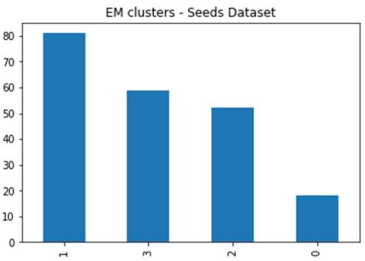












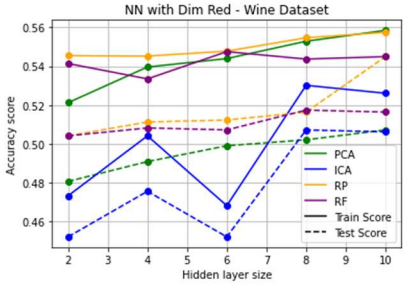
4. COMPARITIVE ANALYSIS

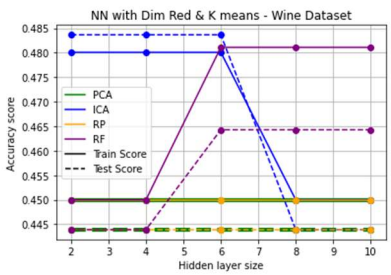
For wine dataset, the highest performance is yielded by ICA with K means, RF with K means, RF with EM with v-measure score of 0.082. The lowest score yielded is 0.05 by ICA with EM. Overall the clustering methods do not do a good job and are not suitable on this dataset.

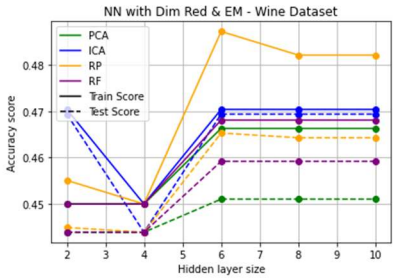
For Seeds dataset, ICA with EM gives the highest v-measure score of around 0.9. Even ICA with K means gives a score of 0.87. The least score is given by RF with K means of 0.61. But it was interesting to note that PCA reduced the whole set of 7 features to just 2 components producing scores of 0.72 with both clustering methods.

5. NEURAL NETWORKS

I have used the wine quality dataset from assignment 1 for these experiments. I have used the sklearn's MLPClassifier and evaluated the performance using the accuracy score. I split the data into train set and test set by 80-20 and recorded both scores.







Neural networks with Dimensionality reduction: Running the neural networks with the features(reduced dimensions) generated by PCA,ICA,RP,RF. We can see that Random Projections has performed best, especially at layer size 10, the test score reaches the maximum value of 0.54. ICA has performed least in comparison.

Neural networks with Dimensionality reduction & clustering: Using the reduced dimensions generated by PCA,ICA,RP, RF I generated cluster labels using the clustering algorithms K means, EM. Now using these cluster labels as features, we run the neural networks and measure the performance.

K means:

We can see that the accuracy scores have dropped using the cluster labels compared to the dimensions as features. But it is interesting to note that with only the cluster labels as a feature, though all the information about the data has been compressed onto a single feature, the accuracy scores vary only by 0.1 range.

We can recall from assignment 1 that using the same dataset with its original features, we were able to achieve an accuracy score of around 0.5 using neural networks. Clustering can be very useful in these cases where, comparing running neural networks with just 11 original features and just 1 feature, the running time significantly come down.

Though applying clustering methods on the wine dataset yielded poor results, using clustering as means of dimensionality reduction has yielded good results.

K means with ICA gives us the maximum test score here at layer size 2. PCA and RP have consistent scores over all layer sizes. We still need to note here that the axis of comparison here is between 0.44 to 0.48.

EM:

Here, EM with ICA gives the maximum test score of 0.47. RP comes second in terms of both train and test scores. It is interesting to note that, projecting the data on a random low-dimensional space is as efficient as other algorithms. It is computationally better and seems to perform well in terms of accuracy as well.

EM and K means produce similar performances in the neural networks.

6. CITATIONS

Maklin, Cory. "K-Means Clustering Python Example." *Medium*, Towards Data Science, 21 July 2019, towardsdatascience.com/machine-learning-algorithms-part-9-k-means-example-in-python-f2ad05ed5203.

VanderPlas, Jake. "In Depth: Gaussian Mixture Models." *In Depth: Gaussian Mixture Models | Python Data Science Handbook*, jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html.

"Legend Guide¶." *Legend Guide - Matplotlib 3.2.1 Documentation*, matplotlib.org/tutorials/intermediate/legend_guide.html.

"2.3. Clustering¶." Scikit, scikit-learn.org/stable/modules/clustering.html#clustering-evaluation.

Hightower, Mallory. "When Clustering Doesn't Make Sense." Medium, Towards Data Science, 18 Apr. 2019, towardsdatascience.com/when-clustering-doesnt-make-sense-c6ed9a89e9e6.

GmbH, RapidMiner. "Expectation Maximization Clustering (RapidMiner Studio Core)." Expectation Maximization Clustering - RapidMiner Documentation, docs.rapidminer.com/latest/studio/operators/modeling/segmentation/expectation_maximization_clustering.html.