HW3: Unsupervised Learning
Kunal Agarwal

1. Objective:

In this assignment, the objective is to apply unsupervised learning algorithms like clustering and dimensionality reduction. In the first part, we run various clustering algorithms on the datasets and note the observations. In the second part, dimensionality reduction (PCA) has been used to reduce to feature space for both the datasets and how this affects clustering is studied. Then in final part, we study how the clustering algorithms and the dimensionality reduction (PCA) affect the accuracy of a dataset by passing the transformed data through a Neural Network.

2. Datasets:

Phishing Website Dataset and Gene Expression datasets from HW1 has been chosen again. The phishing website dataset has 11055 instances and 30 attributes. The response variable has 2 classes indicating whether the website is phishing or legitimate. Phishing websites are threat to online security. These websites have the layout of legitimate website and it is important for internet browser and firewall to detect it. Therefore, building machine learning models to predict phishing websites is of practical importance for cyber security. Gene Expression dataset deals with prediction of gene expression level based on histone modification signals. The gene expression dataset had 500 features and 2 classes: high expression (1) and low expression (0). Derived variables (15) were created by taking sum, mean and median for each individual histone modification signals.

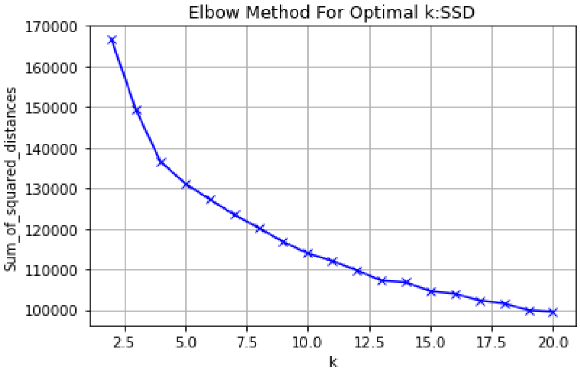
These datasets were chosen to compare the effect of high and low feature space on performance of clustering and dimensionality reduction. Gene expression dataset has highly dimensional data (515) features whereas phishing dataset has just 30 features. All the implementation has been done using Scikit Learn library of Python.

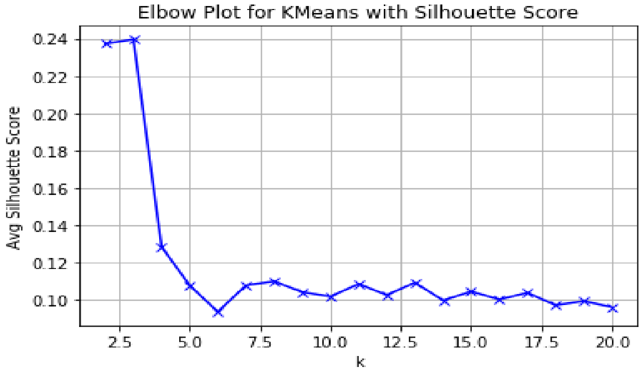
Note: Gene Expression dataset was scaled before passing to any algorithm discussed below. All variables (categorical) in phishing website data had values between -1 and 1. Therefore, it wasn’t scaled.

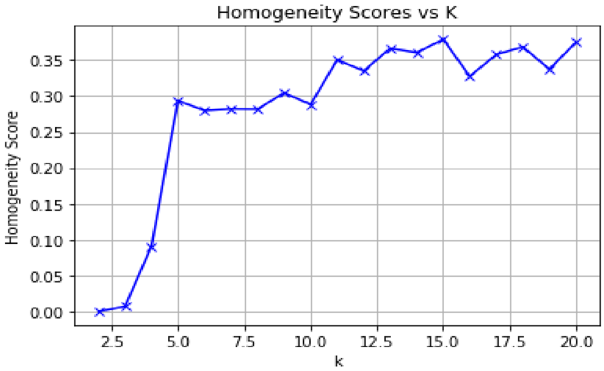
3. Clustering:

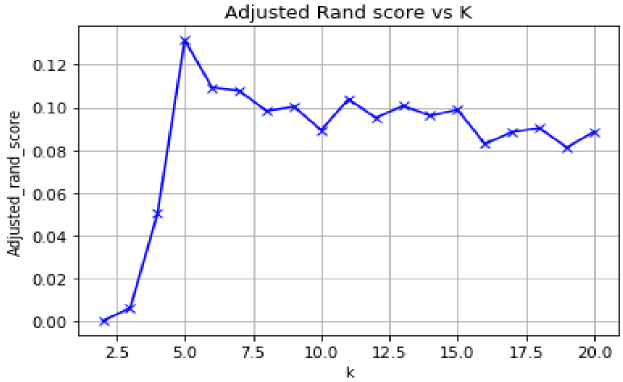
In this part, K-means and Gaussian Mixture Model (based on Expectation Maximization) have been applied on the two datasets. For k-means, optimal **k** (number of clusters) has been chosen based on elbow method using metrics like Sum of Squared Distances and silhouette score. Adjusted Rand Score and Homogeneity score has been computed to see the distribution of true class labels in resultant clusters. In case of GMM, AIC and BIC values have been used to arrive at optimal **n_components** (number of clusters).

Phishing Website Dataset

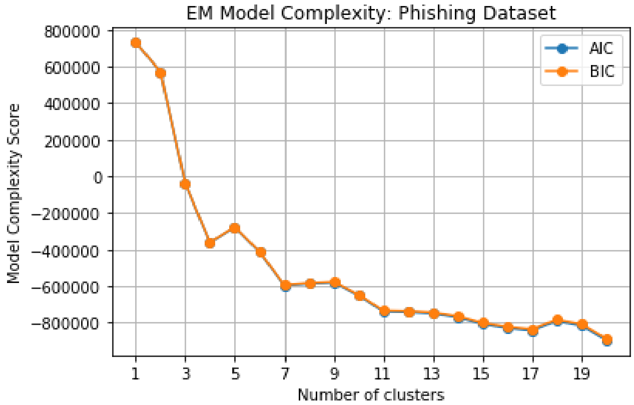


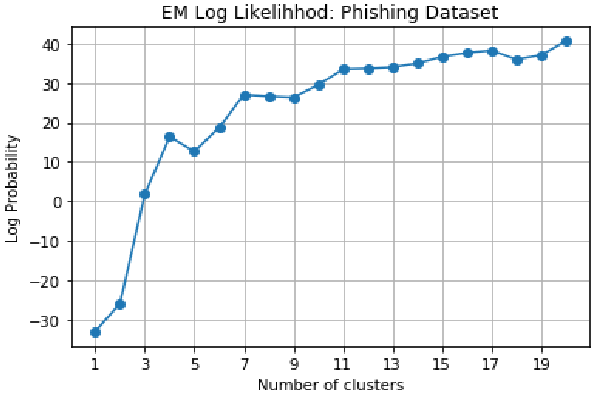






As can be seen from the above elbow plot for SSD, there is a elbow k=4 but due to highest ARS score (highest similarity between true labels and obtained clusters) and fairly high homogeneity score at K=5, I have decided to choose 5 as optimal k-value. Though, silhouette score is low at K=5 compared to initial values but at initial values on further analysis we found the presence of clusters below average silhouette which is not good. This reveals that there is presence of underlying clusters within the same class data points.

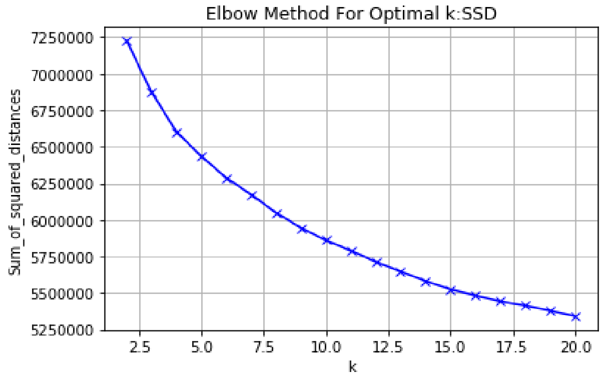


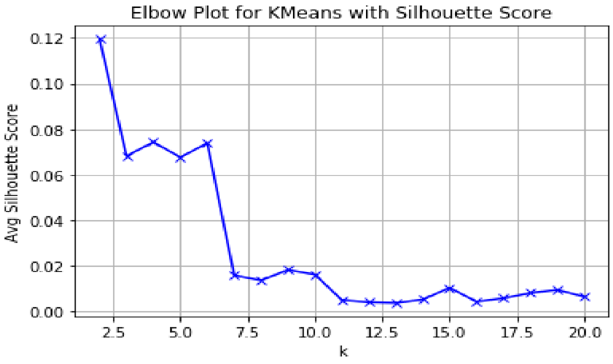


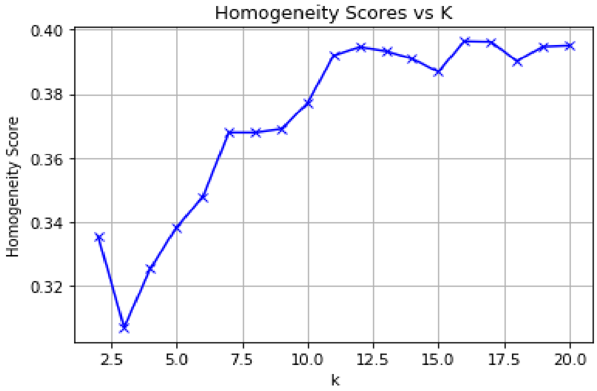
Now we see the results for EM algorithm GMM on the phishing dataset. The log likelihood keeps increasing but the rate of increase has slowed down around n_componets=4. Also, we can see from EM Model Complexity vs number of clusters graph, there is a clear elbow at n=4. Therefore, based on AIC/BIC values **n=4** has been chosen as optimal value. This confirms that there is presence of underlying clusters within same label data points. Another interesting observation is that the AIC and

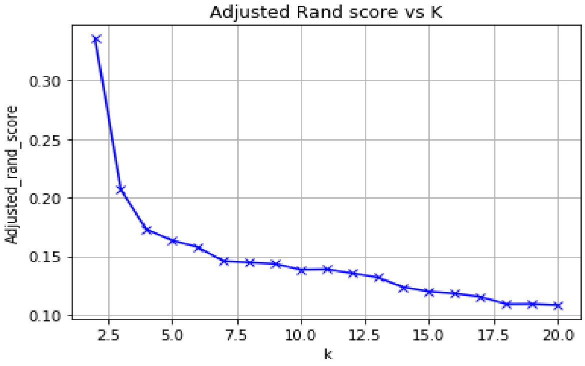
BIC scores overlap. BIC penalizes more heavily for highly complex models than AIC but since the number of attributes is less in this case, it might have caused them to have same values.

Gene Expression Dataset

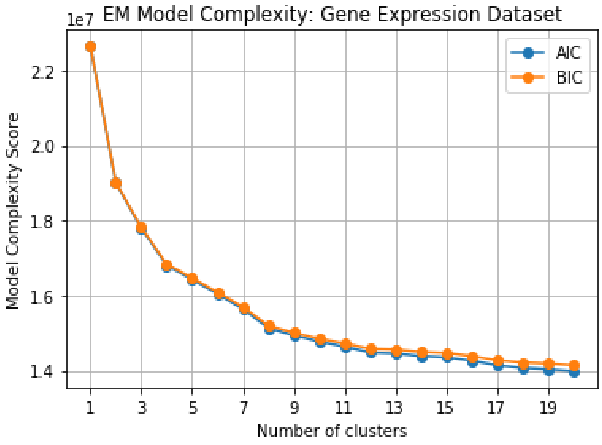

Elbow Method For Optimal k:SSD

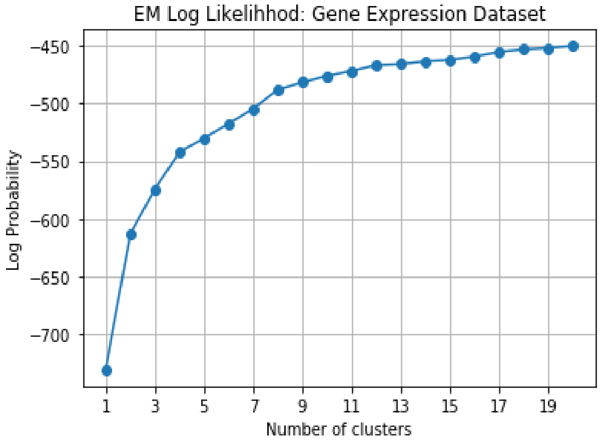

Elbow Plot for KMeans with Silhouette Score


Homogeneity Scores vs K


Adjusted Rand score vs K

In gene expression dataset as can be seen from SSD vs K graph, there is no presence of clear elbow in the graph. Also, silhouette score is highest at k=2 revealing as k increases clusters separation distance is decreasing and adjusted rand score is also highest at k=2 revealing highest similarity between true labels and obtained clusters at k=2. Therefore, for gene expression dataset **k=2** has been chosen as optimal value.


EM Model Complexity: Gene Expression Dataset


EM Log Likelihhod: Gene Expression Dataset

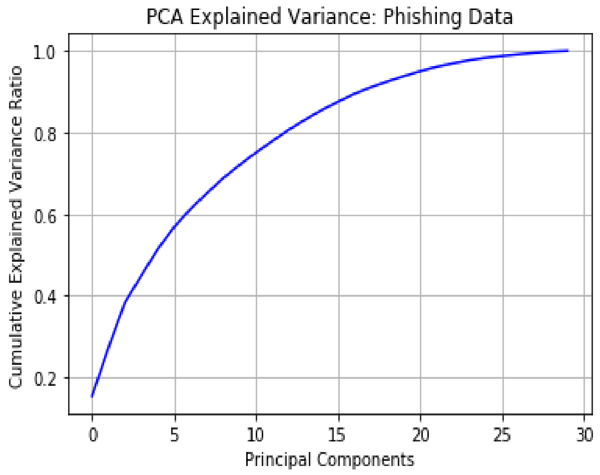
In gene expression dataset, for AIC/BIC vs n_components graph, there is a clear presence of elbow at n=2 and n=4 and the rate of decrease in AIC/BIC decreases after this point. Therefore, in this case **n=4** has been chosen as the optimal value. In GMM, 4 clusters are optimal which indicate there might

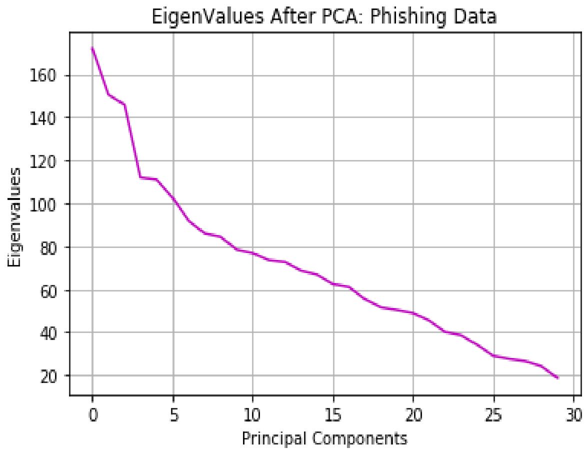
be because within High Expression and low expression class there can be subdivisions as very high, high, low, very low expression level.
This also shows the difference between k-means and GMM. GMM is less restrictive and in GMM clusters can take ellipse shapes which is not the case in k-means which can only take circular shapes. It is possible that therefore in this case GMM does better job.

4. Dimensionality Reduction (PCA):

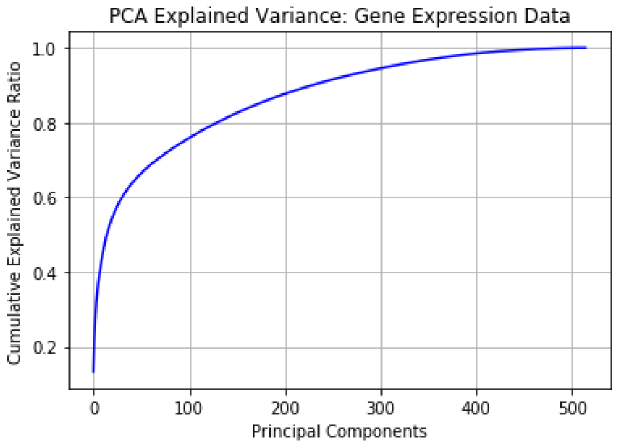
In any dataset, it may happen that not all the features are significant. Moreover, if the dataset (such as Gene Expression above) contains a large amount of features, it leads to the Curse of Dimensionality. This is where dimensionality reduction helps by reducing the feature space. The reduction is usually a step prior to supervised steps. Here we apply Principal Component Analysis (PCA) and analyze the impact on performance of clustering.

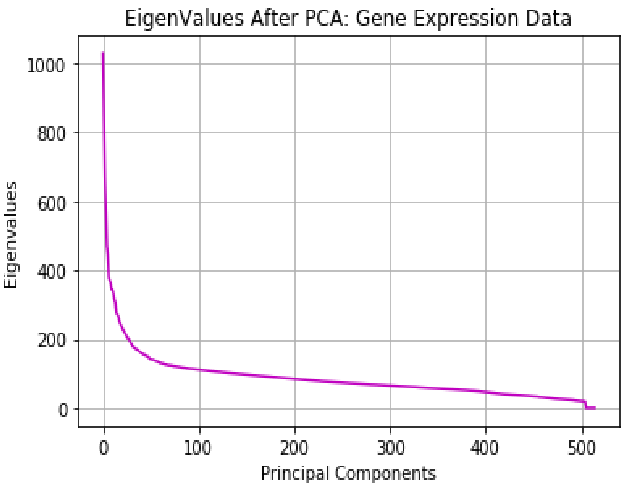
PCA aims to find the directions of maximum variance in high-dimensional data and projects it onto a new subspace with equal or fewer dimensions that the original one. We implement PCA using sklearn library and try to find the optimal number of components using 2 curves: Cumulative Explained Variance Ratio vs number of components and Eigenvalues vs components curves.





In phishing dataset, as it can be seen from the graph above, 95% of the variance in the data is explained by 21 principal components and rest 9 components explain just 5%. EigenValues for components decreases indicating less important components at end. Therefore, **n_components = 21** is the optimal value and I reduced the original phishing data into 21 dimensions.

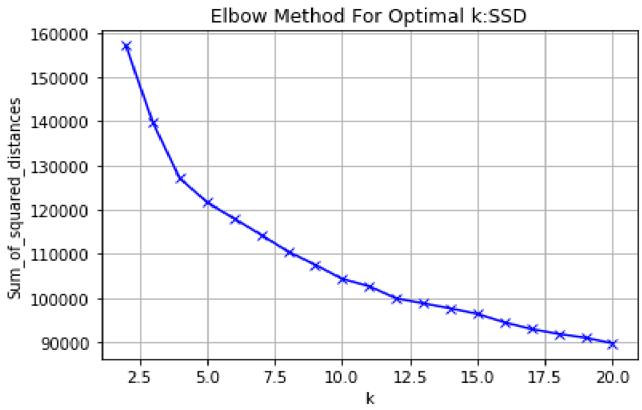


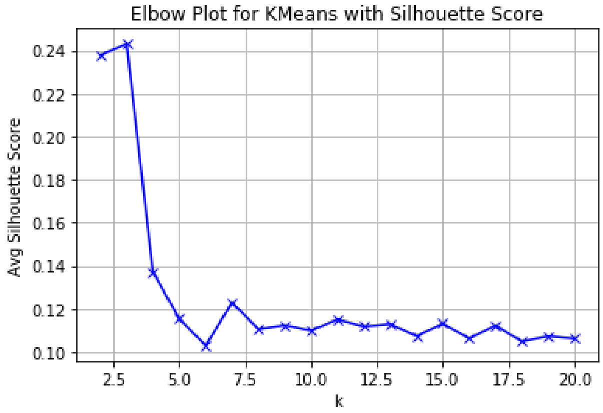


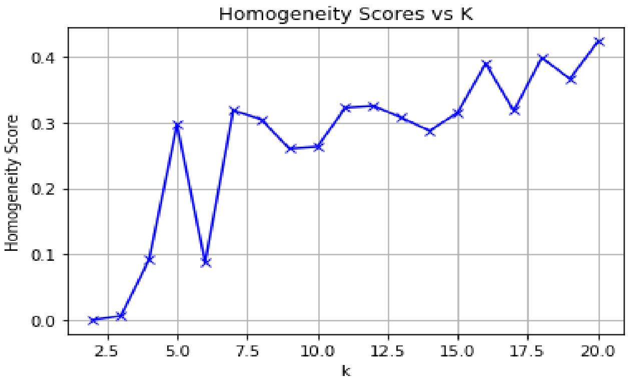
In case of Gene Expression dataset which is a very high dimensional data compared to phishing dataset, it can be seen that more than 75% variance can be explained by only 100 components. This shows that lot of features in the dataset are redundant, correlated or simply insignificant and significantly less features can explain the data perfectly. For choosing optimal component value, I considered 230 (90% variance) and 310 (95% variance) values but selected **n_components = 230** as the optimal value. It is because taking extra 80 components for 5% variance (0.06% per component) is not optimal.

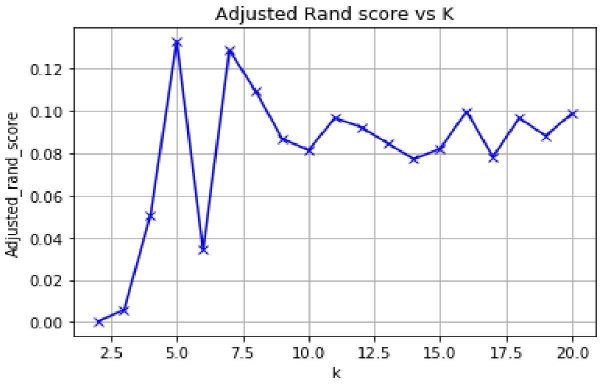
5. Clustering After Dimensionality Reduction:

Phishing Dataset



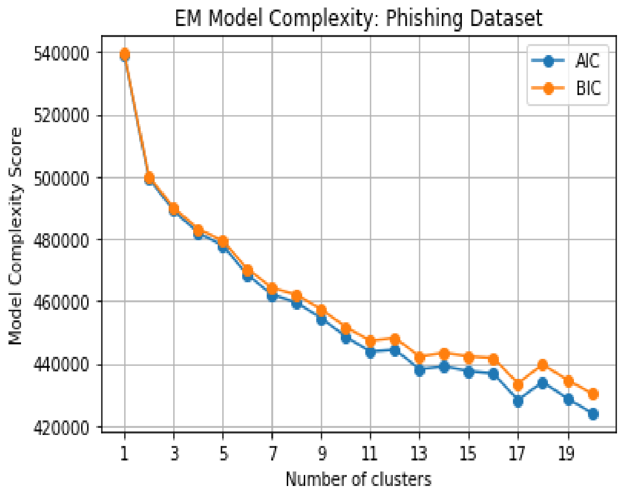


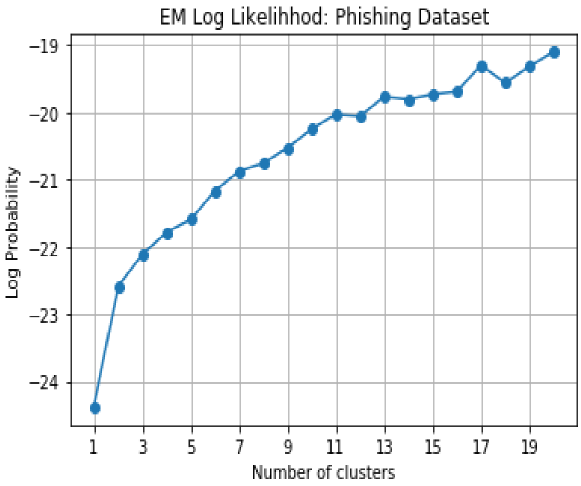




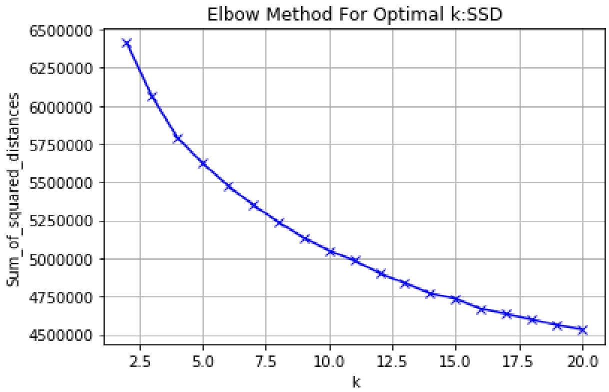
For k-means clustering on phishing data after PCA, we observe that we find almost similar behavior of all the graphs and find same clusters (**k=5**) which shows that transformed dataset covers almost the entire behaviour for the dataset.

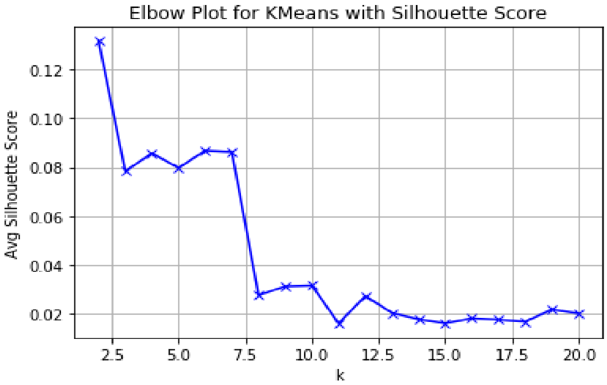
In GMM, there is some difference compared to the results before PCA. Now we see an elbow at **n_components** = **2** compared to 4 before PCA. However, since we know we have 2 target classes, so probably, having reduced features here leads to a better EM result (perhaps overcomes the Curse of Dimensionality?) and thus yields better EM clusters.

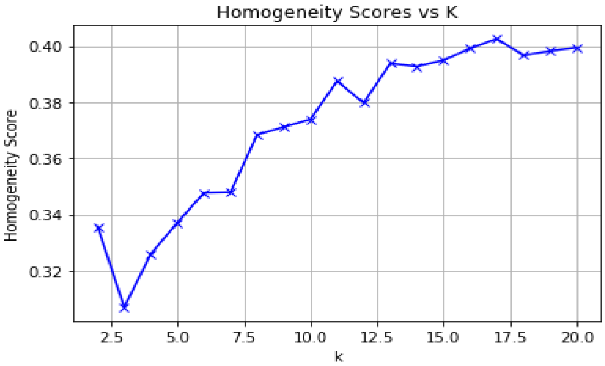


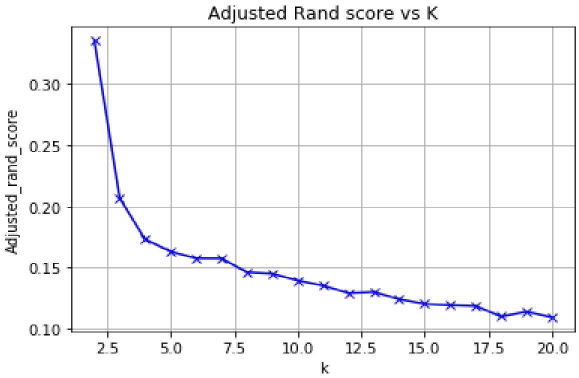


Gene Expression Dataset

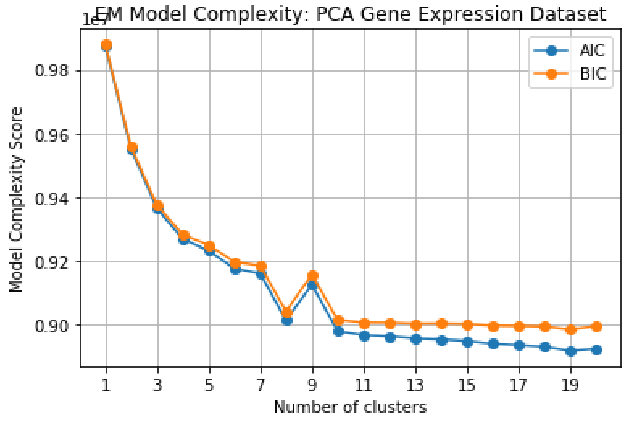


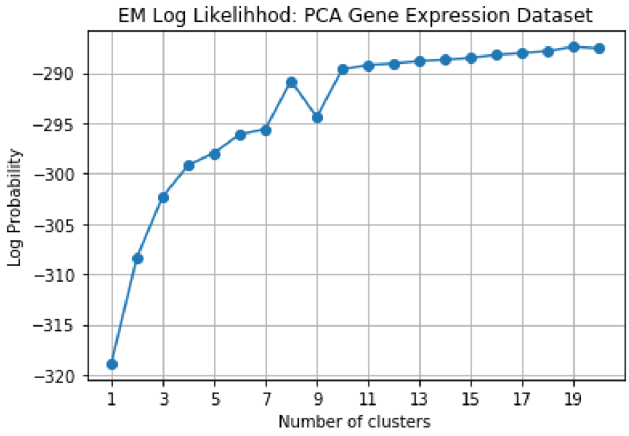






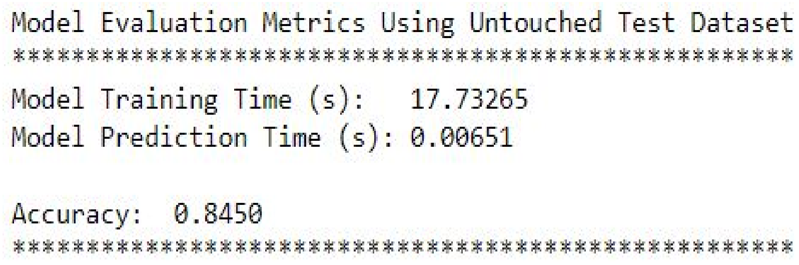
For expression dataset also, both in k-means and GMM the trend is same as before PCA and **k=2** and **n=4** is still optimal value. This signifies that the reduced dataset captures the original dataset features quite well, and therefore this shouldn't change the shape of the clusters.





6. Neural Network Performance:

In this section, we try to analyze the effect of dimensionality reduction (PCA) and clustering on neural network performance. Gene Expression data has been used for this purpose. The full dataset was splitted into training (70%) and testing set (30%). The dataset was then scaled. Then, I tuned parameters for neural network using test accuracy on original non-reduced dataset. The results on original data is shown below:

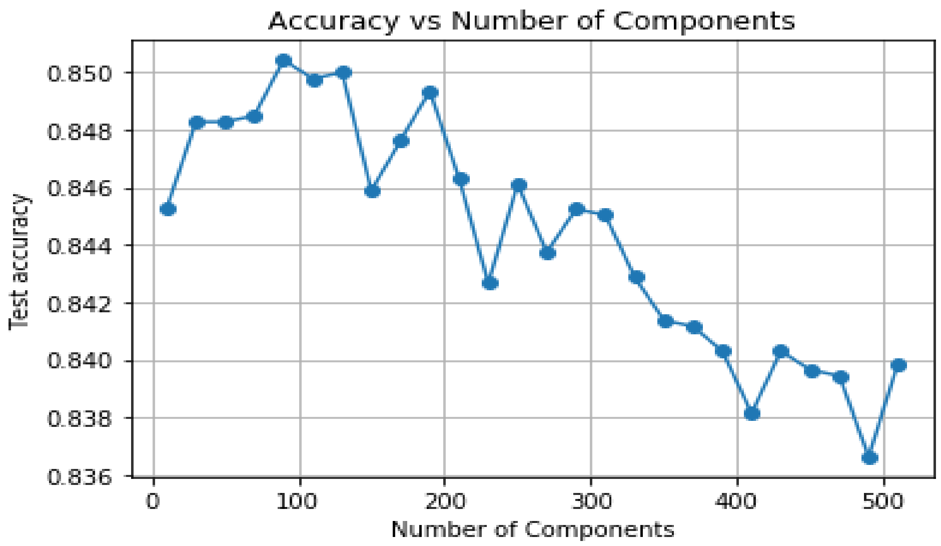


```
Model Evaluation Metrics Using Untouched Test Dataset
*****************************************************
Model Training Time (s):   17.73265
Model Prediction Time (s): 0.00651

Accuracy:  0.8450
*****************************************************
```

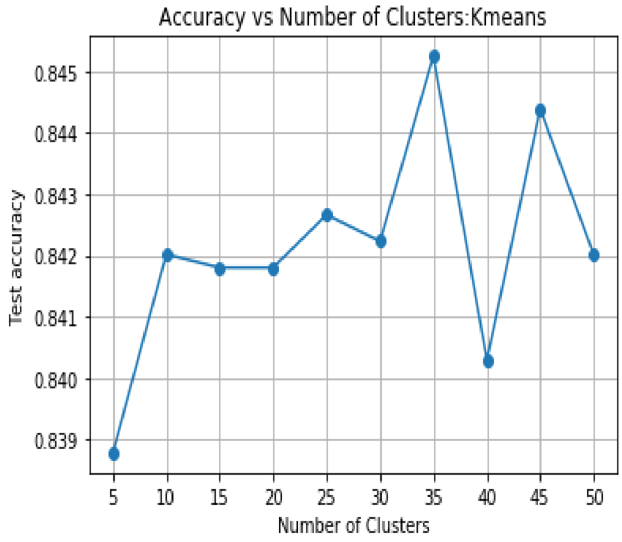
PCA Transformed Data

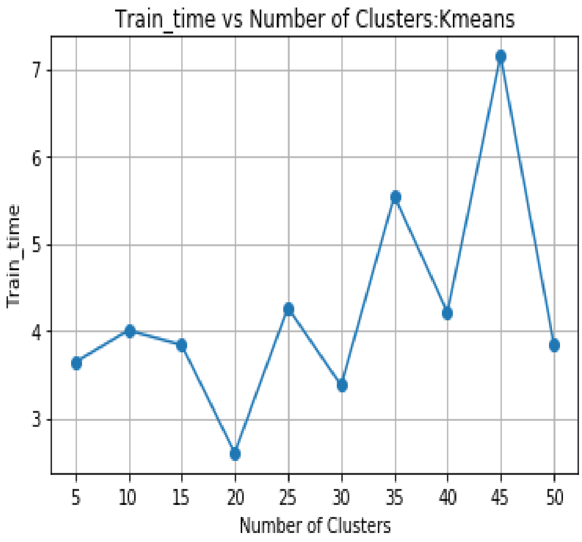
I applied PCA on training set and after learning a projection solely from the training set and projecting both the training and test sets, test accuracy was iteratively computed for component values in range of [10,505] in interval of 10. As evident from the accuracy vs number of components graph below, highest accuracy (0.8504) is achieved at **n_components=90** which is better performance than on the original dataset before PCA. It indicates that as seen above lot of features in the dataset are redundant or simply insignificant (curse of dimensionality) since better neural network performance has been achieved by using just 1/5th components. Also, model training time is way (3.53 s) lesser. This dataset clearly shows the benefit of dimensionality reduction technique like PCA.



K-means Transformed Data

Similar to PCA, I applied k-means on training set and after learning a projection solely from the training set and projecting both the training and test sets, test accuracy was iteratively computed for k (number of clusters) in range of [5,50] in interval of 5. Here, distance of data point from each cluster center has been used as features for transformation.

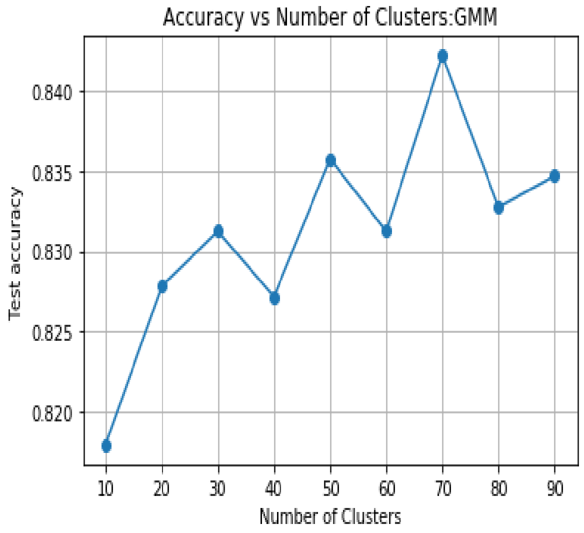


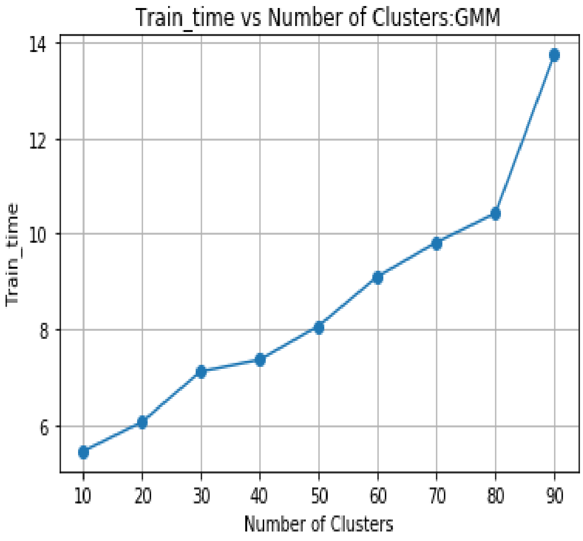


As evident from the accuracy vs number of clusters graph above, highest accuracy (0.8452) is achieved at **k=35** which is same performance as non-reduced dataset but very little worse than PCA. Also, the training time is (5.54 s) is less as expected than on original dataset due to dimensionality reduction. It again shows evidence of curse of dimensionality in the original gene expression dataset.

GMM Transformed Data

In case of GMM, the probability of belonging to each cluster has been used as features for transformation. Test accuracy was iteratively computed for n_components (number of clusters) in range of [10,90] in interval of 10.





As evident from the accuracy vs number of clusters graph for GMM above, highest accuracy (0.8422) is achieved at **n_components=70** which is very slightly less compared to non-reduced original data. Also, the training time is (9.81 s) is less as expected than on original dataset due to dimensionality reduction.

Comparison Table

Above results for neural network performance have been tabulated below:

Neural Network	**Test Accuracy**	**Training Time (sec)**
Original Dataset	0.8450	17.73
PCA transformed	0.8504	3.53
K-means transformed	0.8452	5.54
GMM transformed	0.8422	9.81

7. Conclusion:

This assignment gave us a lot of insight in analyzing unsupervised algorithms which helps a lot in making sense of unstructured data. We also applied various dimensionality reduction techniques to reduce our feature space. This is a very important step considering the curse of dimensionality!. It helps us to speed up any machine learning pipeline as well as reducing noise of our data for better predictions.

References

1. Gene Expression Dataset: https://www.kaggle.com/c/gene-expression-prediction
2. Phishing Dataset: https://www.kaggle.com/akashkr/phishing-website-dataset
3. Scikit-Learn Library: Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.