1 Introduction

This project investigates algorithms in unsupervised learning: PCA (Principal Component Analysis), K-means clustering and GMMs (Gaussian Mixture Models).

1.1 Classification problems

(Note: these problem descriptions are taken from the descriptions in Assignment 1).

1.1.1 Pima Indians Diabetes Dataset

This classification problem involves predicting whether or not a person has diabetes based on certain features: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age. There are 768 images in the dataset.

This classification problem is interesting since it is not necessary clear whether someone has diabetes given those limited features, so there is not necessarily a clearly separable boundary. Running different algorithms in this problem should give some insight on how to find a proper boundary for such a dataset.

1.1.2 Predicting whether mushrooms are edible

The objective of this classification problem is to predict whether or not a mushroom is edible or poisonous, based on a set of given features: cap-shape, cap-surface, cap-color, bruises, odor, gill-attachment, gill-spacing, gill-size, gill-color, stalk-shape, stalk-root, stalk-surface-above-ring, stalk-surface-below-ring, stalk-color-above-ring, stalk-color-below-ring, veil-type, veil-color, ring-number, ring-type, spore-print-color, population, habitat The dataset contains 8,124 data points in total.

1.2 Experiments

First, two datasets are chosen. In this experiment, the datasets are the Mushrooms dataset from Kaggle and the Pima Indians Diabetes dataset from Kaggle.

Then, two clustering algorithms (K-means clustering and GMMs via expectation maximization) are run.

Then, the datasets are reduced into fewer dimensions using Principal Component Analysis. The clustering algorithms are re-performed on the reduced dataset.

Then, the original results of the PCA algorithm on the Pima Indians Diabetes dataset are taken, and the neural network is trained on this reduced dataset. To evaluate the accuracy of the neural network, both the training and testing datasets are projected with the dimensionality function that had been generated before.

Finally, the training dataset of the Pima Indians Diabetes dataset is clustered using the k-Means and GMM algorithms and reduced to fewer dimensions using the two algorithms (as will be explained in the last section). Then, with the clustering algorithms used on the training set, the test set is also reduced to those same features. A neural network is then trained on the training datasets produced by the two clustering algorithms.

2 Results and Analysis

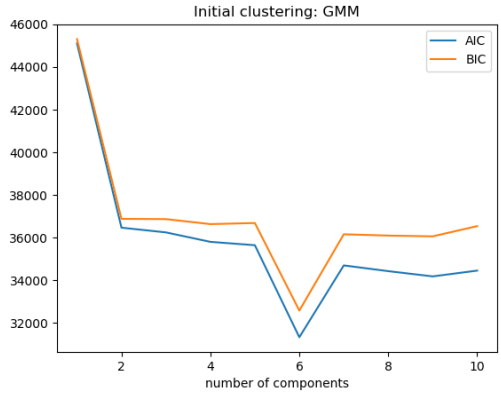
2.1 Initial clustering

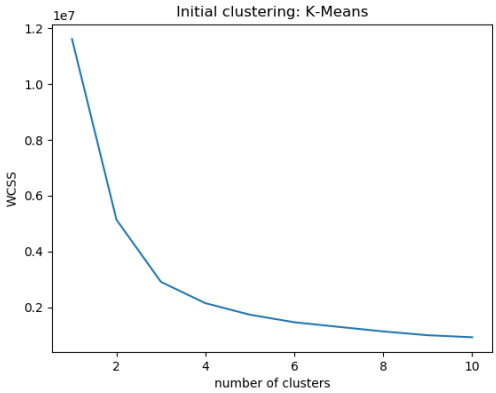
For each of the two datasets, the GMM and K-means clustering were both run on the training set. To determine the optimal number of cluster, some form of a metric for cost was plotted over the number of clusters. For K-means, this metric was WCSS (within-cluster sums of squares), while metric for GMMs was AIC(Alkaline Information Criterion) and BIC(Bayesian information criterion).

In order to have a reasonably low cost while not having too many clusters, it was useful to find a part of the graph that was shaped like an elbow to optimize for those two quantities as described in [4].

Here are the plots generated:

2.1.1 Initial clustering: Pima Indians Diabetes dataset

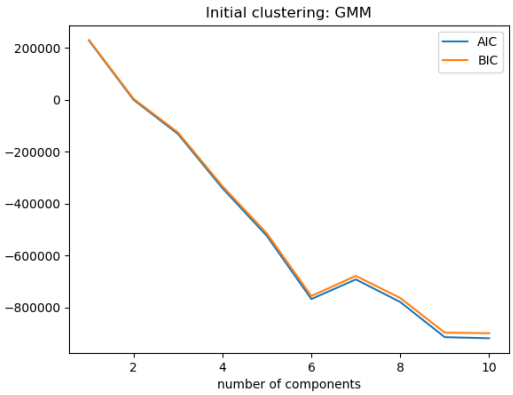


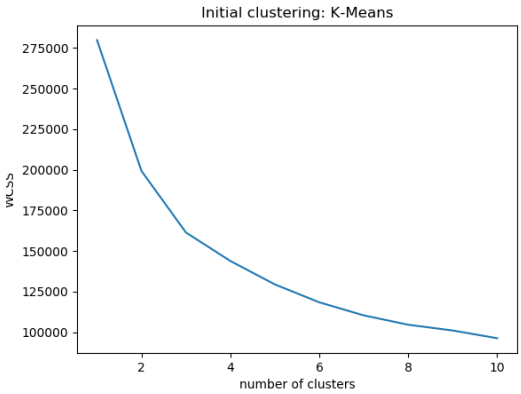


In this dataset, a clear "elbow" can be seen at two clusters for the GMM, and there is a slight "elbow" at three clusters for the K-means.

The gmm graph has an unintuitive dip at k = 6. However, this could have just been due to randomness; there is no striking behavior seen in the final experiment in the plot for the neural network when k = 6.

2.1.2 Initial clustering: Mushrooms dataset





These two graphs look quite different. The plot for Gaussian Mixture Models has a much sharper elbow, implying that this model does much better when given a higher number of clusters compared to a low number, unlike k-Means, which would run in the least cost manner when running it with

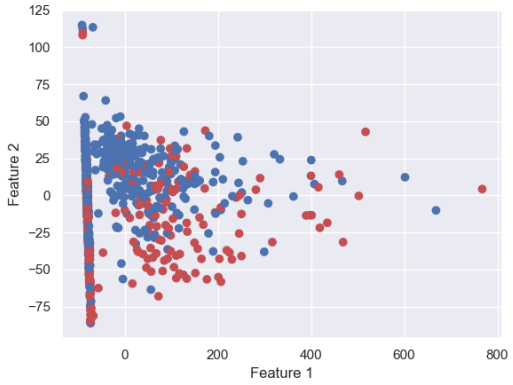
a smaller number of components.

A reasonable explanation for this disparity is that K-means is sensitive to outliers and would put them in a separate cluster by [3] and thus would need more clusters to minimize its cost.

2.2 PCA

PCA was run on a couple of the datasets. To make the datasets easy to visualize, the PCA was run such that the data was to be split into two components. Here are the results of the PCA that was performed:

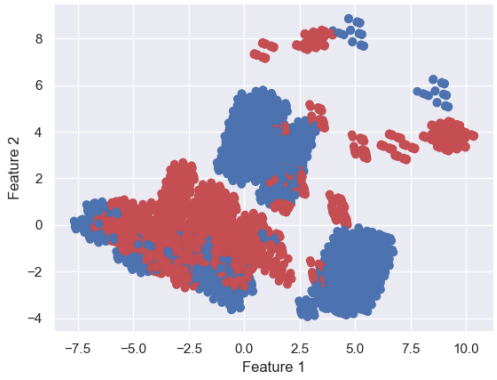
2.2.1 PCA: Pima Indians Diabetes dataset



When considering the visualization of the data reduced to two dimensions, it can clearly be seen that there are two different regions, the vertical region in the left hand side and the larger region in the right hand side.

The regions in this data do not align with their labels; for each region, one half consists of cases of diabetes, and the other half consists of cases without diabetes.

2.2.2 PCA: Mushrooms dataset

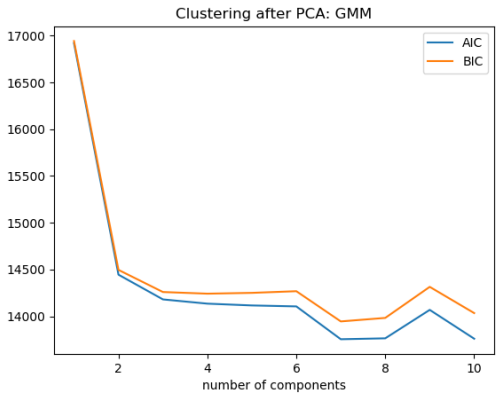


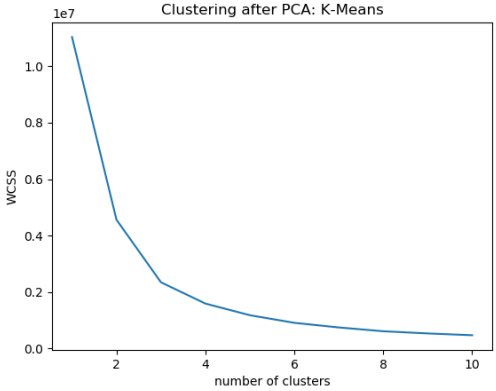
In this two-dimensional projection of the Mushrooms dataset, we can see clear candidates for clusters. There are two blue regions. Also, there is an overlap of blue and red in the bottom left corner. In addition to those main components, there are some outlying red and blue distributed around the upper right portion of the graph.

2.3 Clustering after PCA

The two datasets were clustered with the two algorithms after the PCA dimensionality reduction algorithms were run on them.

2.3.1 Clustering after PCA: Pima Indians Diabetes dataset



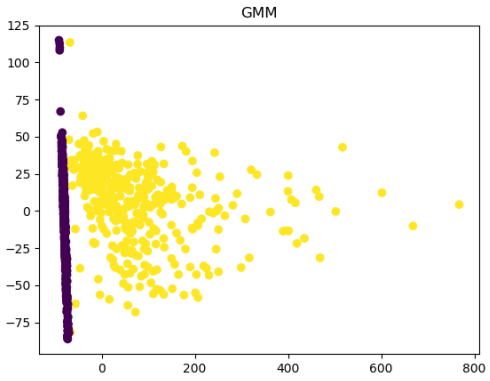


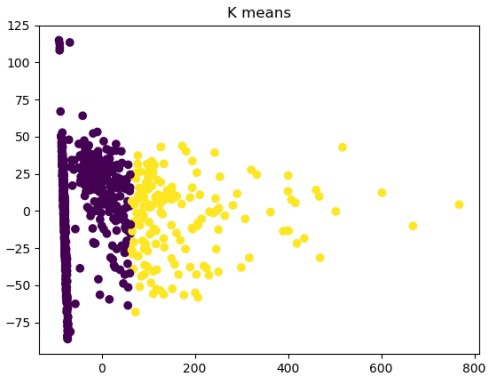
This plot still has a slight dip at k = 6, but it is much smaller than the dip that was created before PCA. Still, there is not any clear information that can be obtained from this dip.

Aside from the dip, the graphs seem similar to the ones created before the PCA algorithm had been run (an elbow at 2 clusters for the GMM and at 3 clusters for the PCA).

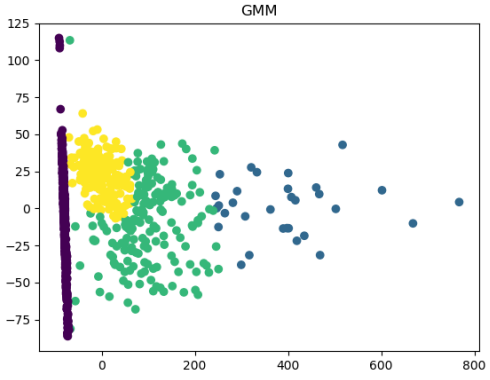
Here are images of the clustering algorithms on the 2-dimensional data.

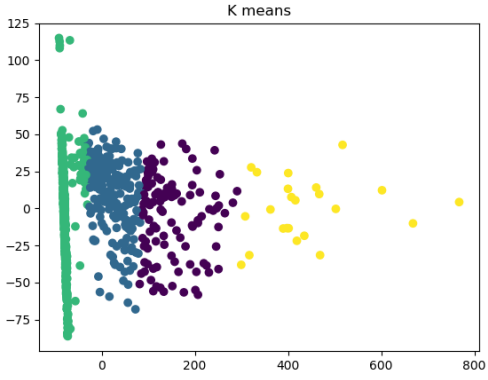
2 clusters:





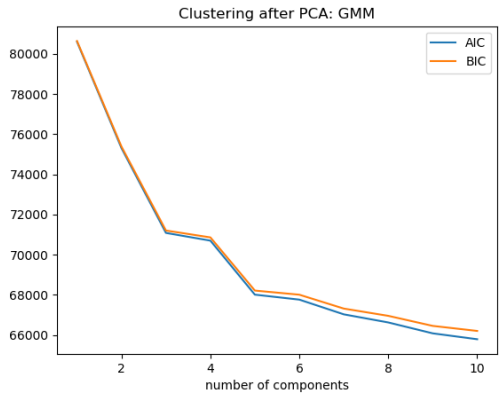
4 clusters:

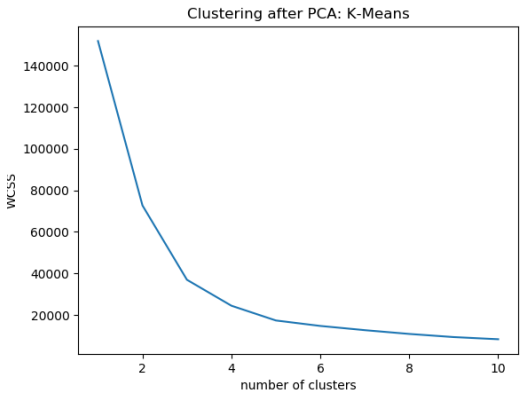




When the value of K (the number of clusters) is increased, the GMM model seems to separate the data in a way such that it corresponds to the training labels (for example, for four clusters on the GMM, the yellow cluster corresponds to blue points in the scatter plot in 2.2.1, while the green cluster corresponds to red points).

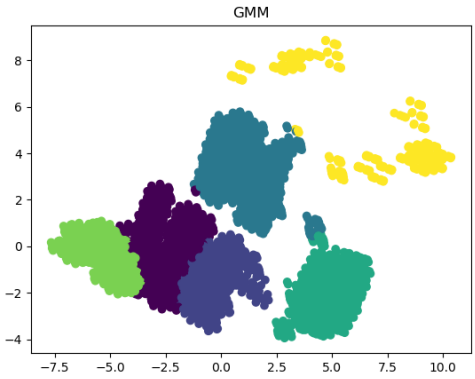
2.3.2 Clustering after PCA: Mushrooms dataset

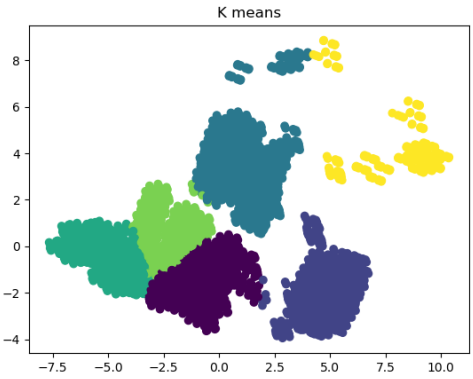




The above two graphs seem quite similar to each other. This is unlike how they were before PCA, where the GMM had an elbow at a much larger number of clusters compared to the GMM.

Here are images of the clustering algorithms on the 2-dimensional data.





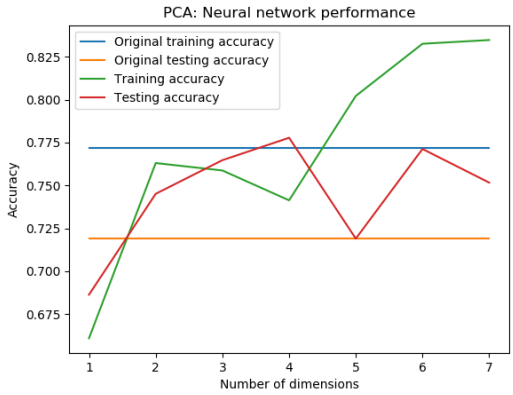
The clusters in the above two images are very similar (with an exception of the way the outlying points were clustered, but that is not very significant). A possible reason for this could be that the Mushrooms dataset had discrete labels, so there are not two many combinations of features that can occur in the input, leading to clusters that are relatively easy to learn.

2.4 Training a neural network after PCA

The Pima Indians dataset was split such that 60 percent of it was training data and 20 percent was testing data.

Cross-validation was not necessary, since the objective here is not to produce the best neural network possible, but to rather observe and analyze the similarities in the neural network performance based on how the dimensions are reduced. Cross-validation adds an overcomplication which detracts from the main purpose of these experiments.

Here is a plot of the accuracy of the neural network (training and testing) with respect to the number of principal components generated in the PCA algorithm:



As the number of dimensions increase, the training error obviously increases (since we have more information about the data). From dimensions greater than or equal to four, the neural network performs at least as well as the original neural network.

However, for four dimensions and onwards, the testing accuracy plateaus (and even has a big 5-percent dip from 4 dimensions to 5 dimensions). This may be due to a more complex input size, which makes it more difficult for the neural network to generalize.

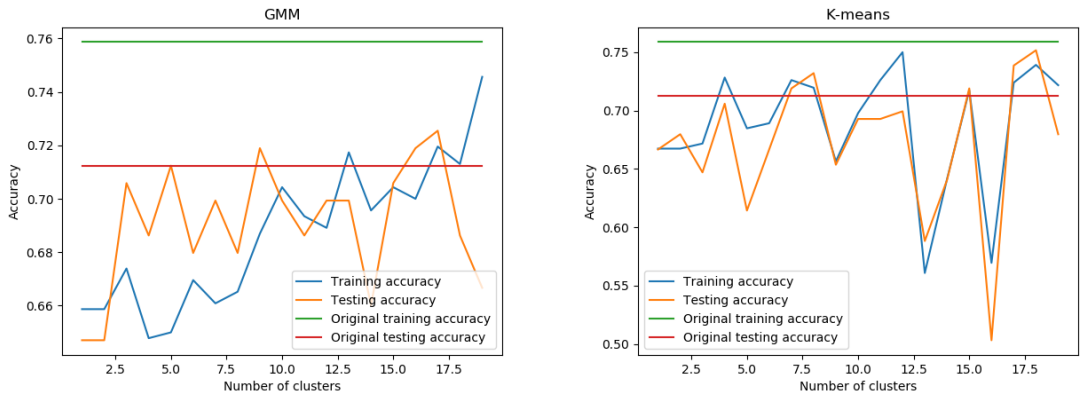
2.5 Clustering for dimensionality reduction

The Pima Indians diabetes was split into a training and testing set (60 percent training and 20 percent testing). By the same reasoning as in the previous experiment, cross validation was not necessary here.

Then, the GMM and the k-Means algorithm were each used to reduce the dimensions of the data for the Pima Indian Diabetes dataset.

In these specific experiements, the k-Means algorithm reduces the data to the cluster-distance space (sklearn's `transform` function), while the GMM reduces each datapoint to the probabilities of the datapoint being in each cluster (sklearn's `predict_proba` function).

For each number of clusters, a neural network was trained to produce a mapping from the reduced data to labels. Here are the results for the training and testing accuracies:



It makes sense that the training and testing accuracy for two clusters is only slightly better than guessing (with percentages in the sixties), as inspecting the figure in section 2.2.1 can give us that each region is likely to have points of either label, thus providing very little information on what the label could be based on the probabilities of solely two clusters.

For the GMM, we see a gradual increase in training accuracy as more and more clusters are added. The lower training accuracy for fewer clusters can be attributed to the loss of information when using fewer components.

However, this trend in the training accuracy is not seen in the plot for the K-means algorithm. In fact, when K = 13, the training accuracy is 56 percent which is only slightly better than guessing a class for this binary classification problem.

A possible reason for this discrepancy in the training accuracy trend was previously mentioned in section 2.3.1; the GMM is better at learning the difference between the labels when more clusters are added (as seen in the yellow and green cluster in the image for four clusters).

As opposed to the previous neural network experiment with PCA, this experiment does not have as high of a training and testing accuracy. This is probably due to the loss in information when reducing it to clusters (especially for k-Means, where the clusters do not give that much information about the labels).

3 Sources

[1] https://www.kaggle.com/uciml/pima-indians-diabetes-database

[2] https://www.kaggle.com/uciml/mushroom-classification

[3] https://people.eecs.berkeley.edu/ jordan/courses/294-fall09/lectures/clustering/slides.pdf

[4] https://www.linkedin.com/pulse/finding-optimal-number-clusters-k-means-through-elbow-asanka-perera/