Assignment No. 6
EECS 690
Introduction to Machine Learning
Due: 11:59 PM, Thursday, April 22, 2021
Submit deliverables in a single zip file to BlackBoard
Name of the zip file: FirstnameLastname_Assignment6
Name of the Assignment folder within the zip file: FirstnameLastname_Assignment6

Deliverables:

1. Copy of Rubric6.docx with your name and ID filled out (do not submit a PDF)
2. Python source code.
3. Screen print showing the successful execution of your Python code. (Copy and paste the output from the Python console screen to a Word document and PDF it).
4. Answer to Part 1, Question 1.
5. Answer to Part 2, Question 2a.
6. Answer to Part 2, Question 2b.

Assignment:

- For both parts use the entire iris data set. We don't need to do training and test sets because this is Unsupervised ML. In both parts you will cluster the data, to see if it clusters into 3 classes or not. Then, you will use the predict() function with the clusters to see how well the k-means and GMM clustered the data. Maybe you will find that there should be more or less than 3 species of iris, based on the data Fisher collected.
- Use the scikit-learn libraries I referenced in class with the default parameters unless otherwise specified below.
- Print out labels between the outputs below so it is clear what you are displaying.

Part 1: k-Means Clustering

- Run the k-means algorithm for k = 1 through 20 and plot the reconstruction error vs. k. You will need to figure out how to plot something in Python. The tutorial we used for the first assignment might help.
- Find the "elbow" of the curve manually. We will call that the elbow_k. (10 points extra credit, if you can find a way to find it algorithmically).
- Now use the predict() method and the clusters for k=elbow_k to classify the entire iris data set.
- Print out the confusion matrix and accuracy.
    - When you examine the knee of the curve for k-means, you may find an elbow_k different than 3. That is perfectly fine. There is no right way to determine k.
    - If you selected elbow_k=3, then you have to keep track of which cluster each class went to when you calculate the Accuracy Score. The easiest way to do that is to see what the majority class is in each cluster. Note: Look at https://stackoverflow.com/questions/45114760/how-to-plot-the-confusion-similarity-matrix-of-a-k-mean-algorithm. You can use this to match the k-mean labels (or GMM prediction) and the truth labels such that the number of true-positive predictions is maximized, essentially

rearranging the columns so that the sum of diagonal entries is maximized. From this you can calculate accuracy score by (sum of diagonal entries)/(sum of all entries).
    - If you selected elbow_k not equal to 3, then you cannot calculate an Accuracy Score. There are similar measures, but we didn't go over them in class, so I don't expect you to calculate them. Instead, print out a message something like this: "Cannot calculate Accuracy Score because the number of classes is not the same as the number of clusters".
    - You CAN, however, print out the Confusion Matrix, if you selected elbow_k!=3 using the built-in Confusion Matrix function. The top 3 rows represent one of the iris species. Columns of the top 3 rows show how the 3 classes were distributed among the clusters, where each column represents a cluster. The bottom rows should be all zeros.
- Now use the predict() method and the clusters for k=3 to classify the entire iris data set.
- Print out the confusion matrix and accuracy.
    - Once again, you have to keep track of which cluster each class went to when you calculate the Accuracy Score. The easiest way to do that is to see what the majority class is in each cluster.
- Question 1: According to your results (i.e., elbow_k), are there 3 species of iris represented in the iris data set?

Part 2: Gaussian Mixture Models (GMM)

- Run the GMM algorithm for k = 1 through 20 and plot the AIC vs. k, where k is the number of components (n_components). Use the aic() method to obtain the AIC. Remember to use "diag" as the covariance_type parameter, not the default or your AIC curve won't look right.
- Find the "elbow" of the curve. We will call that the aic_elbow_k.
- Now run the GMM algorithm for k = 1 through 20 and plot the BIC vs. k, where k is the number of components (n_components). Use the bic() method to obtain the BIC.
- Find the "elbow" of the curve. We will call that the bic_elbow_k.
- Now use the predict() method and the components for k=aic_elbow_k to classify the entire iris data set.
- Print out the confusion matrix and accuracy.
    - When you examine the knee of the curve for AIC, you may find an aic_elbow_k different than 3. That is perfectly fine. There is no right way to determine k.
    - If you selected aic_elbow_k=3, then you have to keep track of which cluster each class went to when you calculate the Accuracy Score. The easiest way to do that is to see what the majority class is in each cluster.
    - If you selected aic_elbow_k not equal to 3, then you cannot calculate an Accuracy Score. There are similar measures, but we didn't go over them in class, so I don't expect you to calculate them. Instead, print out a message something like this: "Cannot calculate Accuracy Score because the number of classes is not the same as the number of clusters".

    - You CAN, however, print out the Confusion Matrix, if you selected aic_elbow_k!=3 using the built-in Confusion Matrix function. The top 3 rows represent one of the iris species. Columns of the top 3 rows show how the 3 classes were distributed among the clusters, where each column represents a cluster. The bottom rows should be all zeros.
- Now use the predict() method and the components for k=bic_elbow_k to classify the entire iris data set.
- Print out the confusion matrix and accuracy.
    - When you examine the knee of the curve for BIC, you may find a bic_elbow_k different than 3. That is perfectly fine. There is no right way to determine k.
    - If you selected bic_elbow_k=3, then you have to keep track of which cluster each class went to when you calculate the Accuracy Score. The easiest way to do that is to see what the majority class is in each cluster.
    - If you selected bic_elbow_k not equal to 3, then you cannot calculate an Accuracy Score. There are similar measures, but we didn't go over them in class, so I don't expect you to calculate them. Instead, print out a message something like this: "Cannot calculate Accuracy Score because the number of classes is not the same as the number of clusters".
    - You CAN, however, print out the Confusion Matrix, if you selected bic_elbow_k!=3 using the built-in Confusion Matrix function. The top 3 rows represent one of the iris species. Columns of the top 3 rows show how the 3 classes were distributed among the clusters, where each column represents a cluster. The bottom rows should be all zeros.
- Now use the predict() method and the components for k=3 to classify the entire iris data set.
- Print out the confusion matrix and accuracy.
    - Once again, you have to keep track of which cluster each class went to when you calculate the Accuracy Score. The easiest way to do that is to see what the majority class is in each cluster.
- Question 2a: According to your AIC results (i.e., aic_elbow_k), are there 3 species of iris represented in the iris data set?
- Question 2b: According to your BIC results (i.e., bic_elbow_k), are there 3 species of iris represented in the iris data set?

Remember:

- Your Programming Assignments are individual-effort.
- You can brainstorm with other students and help them work through problems in their programs, but everyone should have their own unique assignment programs.