Assignment No. 6
EECS 658
Introduction to Machine Learning
Due: 11:59 PM, Thursday, November 14, 2024
Submit deliverables in a single zip file to Canvas
Files in other formats (e.g., .tar) will not be graded
Name of the zip file: FirstnameLastname_Assignment6 (with your first and last name)
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). It should show:
    a. Part 1:
        i. Plot of reconstruction error vs. k
        ii. Confusion matrix and accuracy (if appropriate) for k = elbow
        iii. Confusion matrix and accuracy for k = 3
    b. Part 2:
        i. Plot of AIC vs. k
        ii. Plot of BIC vs. k
        iii. Confusion matrix and accuracy (if appropriate) for k = aic_elbow_k
        iv. Confusion matrix and accuracy (if appropriate) for k = bic_elbow_k
    c. Part 3:
        i. U-Matrix for grid sizes: 3x3, 7x7, 15x15, 25x25
        ii. Quantization error for grid sizes: 3x3, 7x7, 15x15, 25x25
        iii. Graph of quantization error vs grid sizes for grid sizes: 3x3, 7x7, 15x15, 25x25.
4. Answer to Part 1, Question 1.
5. Answer to Part 2, Question 2a.
6. Answer to Part 2, Question 2b.
7. Answer to Part 3, Question 3a.
8. Answer to Part 3, Question 3b.
9. Answer to Part 3, Question 3c.

Assignment:

- For all parts use the entire iris data set. We don't need to do training and test sets because this is Unsupervised ML. In all 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, GMM, and SOM 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. (Hint: Look at PlottingCode.py in Assignment 6 module).
- Find the "elbow" of the curve manually. We will call that the elbow_k.
- 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.

- 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?

Part 3: Self Organizing Map (SOM)

- Implement a Self-Organizing Map (SOM) for the iris data set.
- Setup Instructions
    - ○ Use the implementation from https://github.com/JustGlowing/minisom
    - ○ For guidance on how to set up your SOM, follow the example for the Iris dataset available at: https://github.com/JustGlowing/minisom/blob/master/examples/BasicUsage.ipynb
- Load the iris dataset
- Normalize the features to ensure they are on a similar scale. Normalize means to scale the features values (in each column) to between 0 and 1 using the following formula:

$$f(x) = (x - x_{min}) / (x_{max} - x_{min})$$

- Where
    - ○ x = unnormalized feature (e.g., sepal-length = 5.1)
    - ○ x_{min} = minimum feature value (e.g., sepal-length min = 4.3)
    - ○ x_{max} = maximum feature value (e.g., sepal-length max = 7.9)
    - ○ f(5.1) = (5.1 – 4.3) / (7.9 – 4.3) = 0.22
- Train the SOM
    - ○ Initialize the SOM with four different grid sizes: 3x3, 7x7, 15x15, 25x25
    - ○ Train each SOM using the normalized features.
- Use matplotlib to visualize the U-Matrix (distance map) and plot the response of each SOM with markers for each species.
- Print the Quantization error for each grid size
- Plot a graph for quantization error vs grid sizes
- Question 3a; Use the quantization error vs grid sizes graph to identify the 'elbow' for grid size using the quantization error in the same way you found the elbow in k-means (reconstruction error) and GMM (AIC/BIC), what grid size would you select based on this elbow?
- Question 3b: From the results above, conclude how does the grid size affect SOM’s performance.
- Question 3c: Which grid size between 7x7 and 25x25 would be a perfect fit for iris data set and why?

Rubric for Program Comments		
Exceeds Expectations (90-100%)	**Meets Expectations (80-89%)**	**Unsatisfactory (0-79%)**
Software is adequately commented with prologue comments, comments summarizing major blocks of code, and comments on every line.	Prologue comments are present but missing some items or some major blocks of code are not commented or there are inadequate comments on each line.	Prologue comments are missing all together or there are no comments on major blocks of code or there are very few comments on each line.

Adequate Prologue Comments:

- Name of program contained in the file (e.g., EECS 658 Assignment 1)
- Brief description of the program, e.g.,
  - Check versions of Python & create ML "Hello World!" program
- Inputs (e.g., none, for a function, it would be the parameters passed to it)
- Output, e.g.,
  - Prints out the versions of Python, scipy, numpy, pandas, and sklearn
  - Prints out "Hello World!"
  - Prints out the overall accuracy of the classifier.
  - Prints out the confusion matrix.
  - Prints out the P, R, and F1 score for each of the 3 varieties of iris.
- All collaborators
- Other sources for the code ChatGPT, stackOverflow, etc.
- Author's full name
- Creation date: The date you first create the file, i.e., the date you write this comment

Adequate comments summarizing major blocks of code and comments on every line:

- Provide comments that explain what each line of code is doing.
- You may comment each line of code (e.g., using //) and/or provide a multi-line comment (e.g., using /* and */) that explains what a group of lines does.
- Multi-line comments should be detailed enough that it is clear what each line of code is doing.
- Each block of code must indicate whether you authored the code, you obtained it from one of the sources listed in the prolog, or one of your collaborators authored the code, or if it was a combination of all of these.

Collaboration and other sources for code:

- When you collaborate with other students or use other sources for the code (e.g., ChatGPT, stackOverflow):
  - Your comments must be significantly different from your collaborators.
  - More scrutiny will be applied to grading your comments in particular explaining the code "in your own words", not the source's comments (e.g., ChatGPT's comments).

- Failure to identify collaborators or other sources of code will not only result in a 0 on the assignment but will be considered an act of Academic Misconduct.
- Students who violate conduct policies will be subject to severe penalties, up through and including dismissal from the School of Engineering.