CS5691: Pattern Recognition and Machine Learning
Assignment #2

Topics: LDA, GMM, DBSCAN **Deadline:** 28 April 2023, 11:55 PM

Teammate 1: (Aditya Mahesh Patil) (60% contribution) **Roll number:** CS20B004
Teammate 2: (Chathur Bommineni) (40% contribution) **Roll number:** CS20B018

- **For any doubts regarding questions 1 and 2**, you can mail cs22s013@smail.iitm.ac.in and cs21s043@smail.iitm.ac.in
- **For any doubts regarding question 3**, you can mail cs21d015@smail.iitm.ac.in and cs22s015@smail.iitm.ac.in
- Please refer to the **Additional Resources** tab on the Course webpage for basic programming instructions.
- This assignment has to be completed in teams of 2. Collaborations outside the team are strictly prohibited.
- Any kind of plagiarism will be dealt with severely. These include copying text or code from any online sources. These will lead to disciplinary actions according to institute guidelines. Acknowledge any and every resource used.
- Be precise with your explanations. Unnecessary verbosity will be penalized.
- Check the Moodle discussion forums regularly for updates regarding the assignment.
- You should submit a zip file titled **'rollnumber1_rollnumber2.zip'** on Moodle where rollnumber1 and rollnumber2 are your institute roll numbers. Your assignment will **NOT** be graded if it does not contain all of the following:
  1. Type your solutions in the provided LaTeX template file and title this file as **'Report.pdf'**. **State your respective contributions in terms of percentage at the beginning of the report clearly.** Also, embed the result figures in your LaTeX solutions.
  2. Clearly name your source code for all the programs in **individual Google Colab files**. Please submit your code only as Google Colab file (.ipynb format). Also, embed the result figures in your Colab code files.
- We highly recommend using `Python 3.6+` and standard libraries like `NumPy, Matplotlib, Pandas, Seaborn`. Please use `Python 3.6+` as the only standard programming language to code your assignments. Please note: the TAs will only be able to assist you with doubts related to Python.
- You are expected to code all algorithms from scratch. **You cannot use standard inbuilt libraries for algorithms until and unless asked explicitly**.
- **Any graph that you plot is unacceptable for grading unless it labels the x-axis and y-axis clearly.**

- Please note that the TAs will **only** clarify doubts regarding problem statements. The TAs won't discuss any prospective solution or verify your solution or give hints.
- Please refer to the CS5691 PRML course handout for the late penalty instruction guidelines.

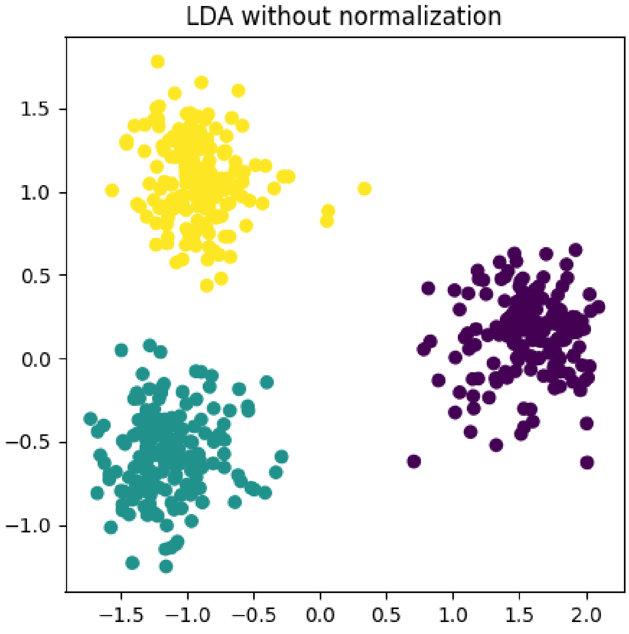
---

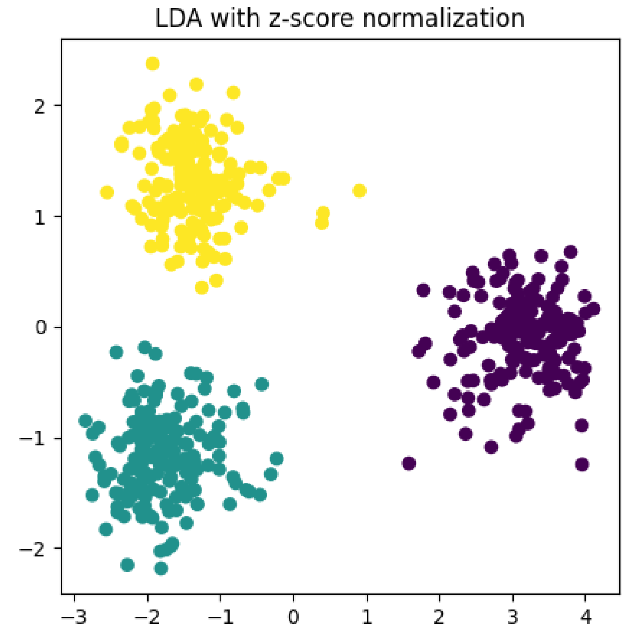
1. **[Linear Discriminant Analysis (LDA), Principal Component Analysis (PCA)**] You will implement dimensionality reduction techniques (LDA, PCA) as part of this question for the dataset1 provided here.

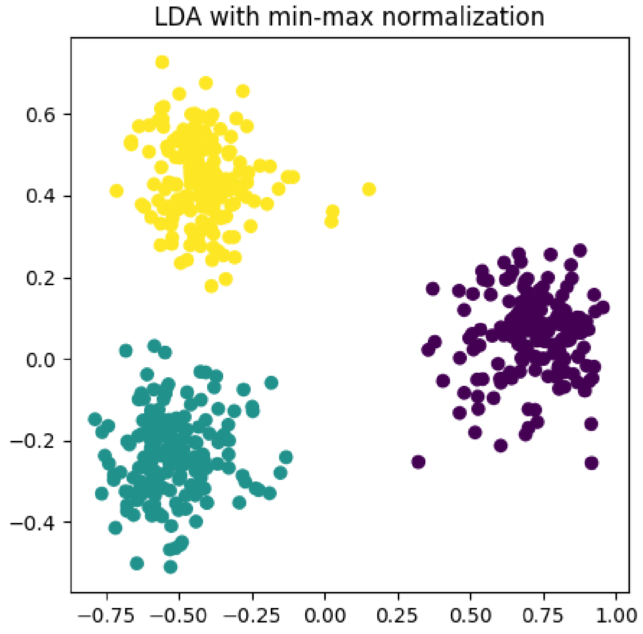
   Note that you have to implement **LDA from scratch** without using any predefined libraries (i.e. sklearn, scipy) . However, you can use **predefined libraries to implement PCA.**

   (a) (2 marks) Use Linear Discriminant analysis (LDA) to convert dataset1 into the two-dimensional dataset and then visualize the obtained dataset. Also, perform an analysis on how results will change if we perform normalization (i.e., zero mean, unit variance normalization) on the initial dataset before applying LDA.

Solution:





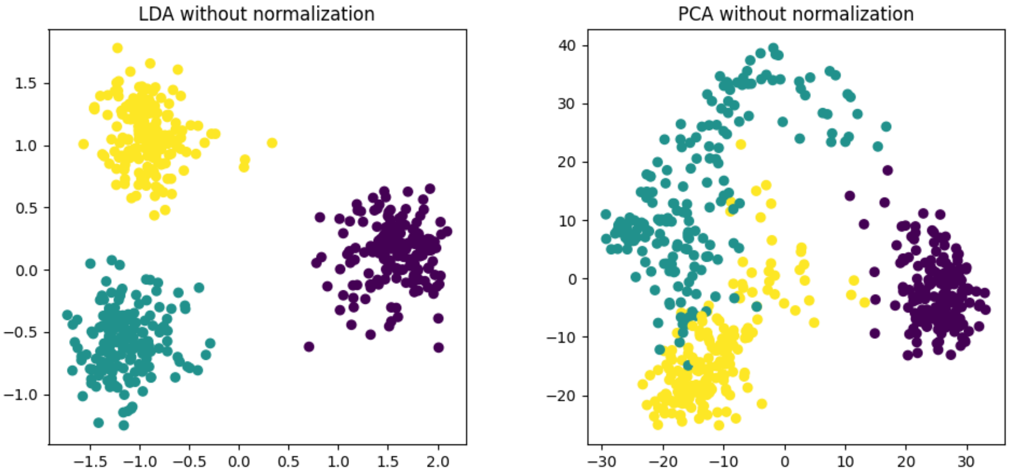


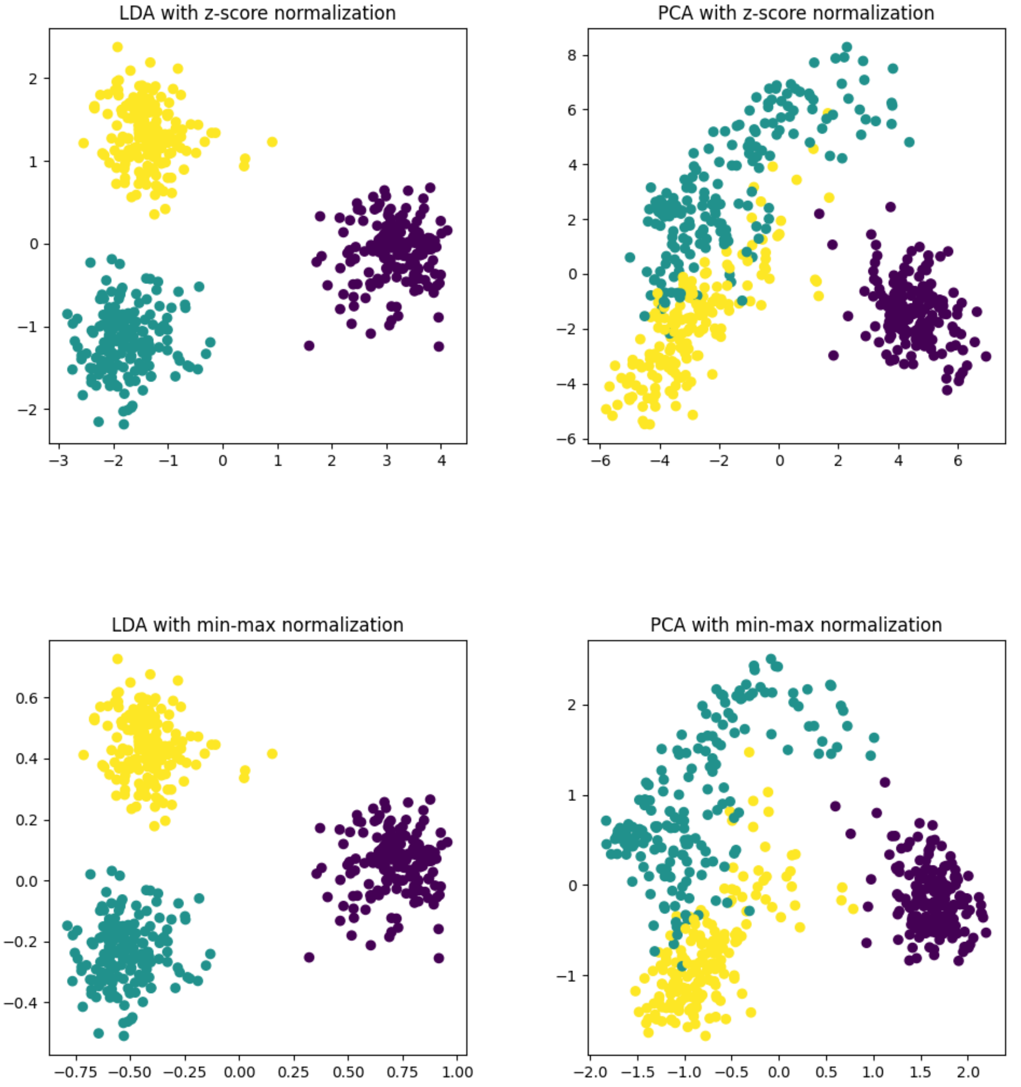
It can be observed that the the distribution of points is same but the data is scaled.

For LDA, the results should not differ if the data has been standardized or not. This is because LDA always internally centers the variables to extract the discriminants. When we apply normalization, we are scaling the axes, but this does not actually change the distribution of data. The shape/position of the decision boundaries between classes may change(since the data is being scaled) but the actual data points will still belong the same classes. Standardization does not effect the final results because LDA decomposes the ratio of between-to-within covariances and not the magnitute of the covariances itself.

(b) (1.5 marks) Use PCA to convert dataset1 into two-dimensional data and then visualize the obtained dataset. Now, compare and contrast the visualizations of the final datasets obtained using LDA and PCA.

Solution:





LDA with z-score normalization

PCA with z-score normalization

LDA with min-max normalization

PCA with min-max normalization

- It is clear from the visualisation of the resultant 2D data that PCA performs better than LDA at the task of separating labeled clusters of data from each other. This is due to the fact that PCA doesn't use labels to separate the data.

- LDA tries to find the feature subspace that maximises the separability between the groups.

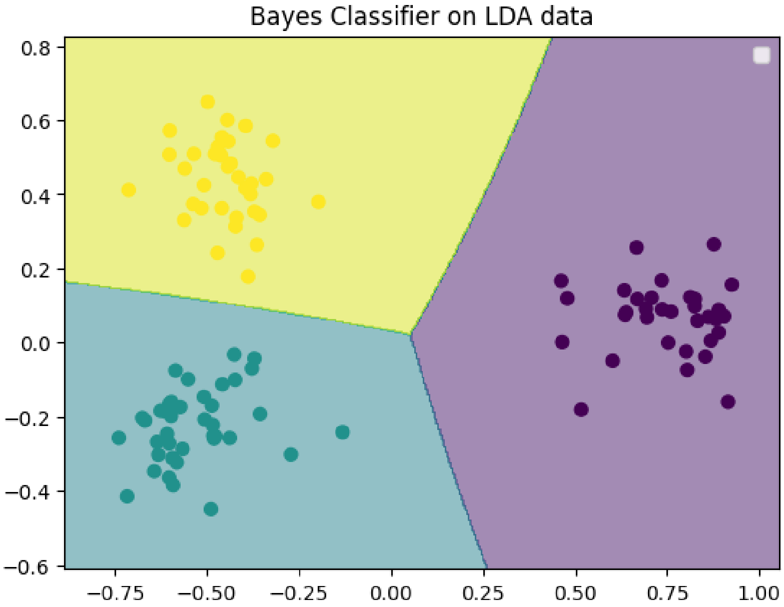
- PCA tries to find the direction of maximum variance in the data.

- Also PCA results are dependant on whether the data was normalized or not. This is due to the fact that PCA is a variance maximizing exercise. This is quite different from LDA which is a separability maximising exercise.

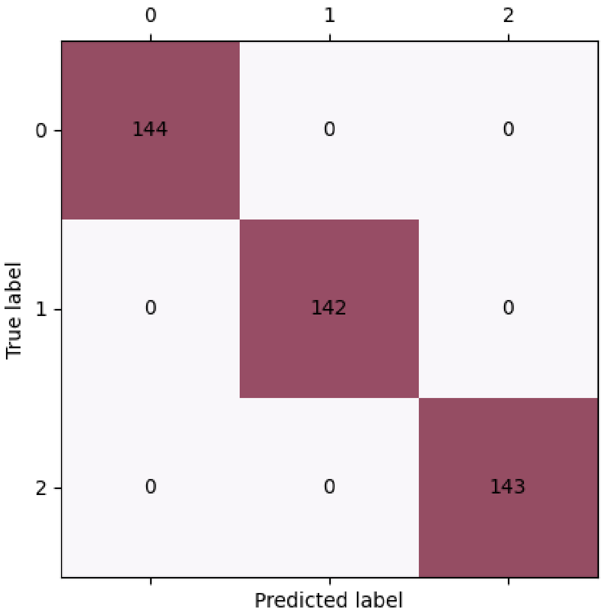
(c) (1.5 marks) Randomly shuffle and split the obtained dataset from part (a) into a training set (80%) and testing set (20%). Now build the Bayes classifier using the training set and report the following:

- Accuracy on both train and test data.
- Plot of the test data along with your classification boundary.
- confusion matrices on both train and test data.

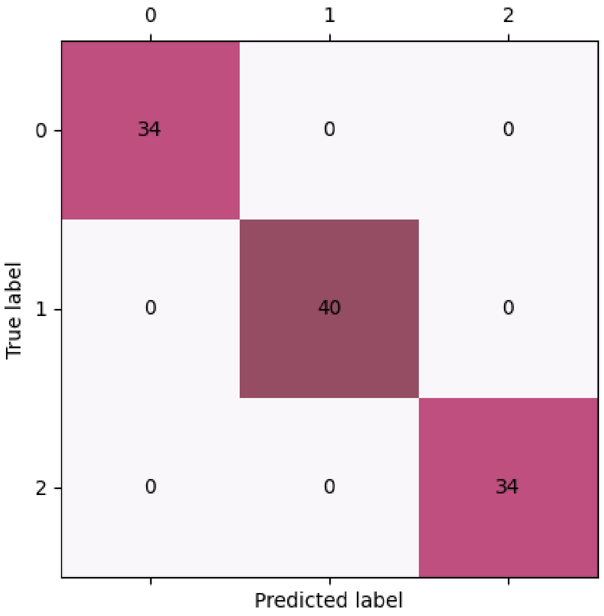
Solution:
Accuracy on Train Data = 100 % Accuracy on Test Data = 100 %



Confusion matrix on Train Data:



Confusion matrix on Test Data:



2. **[DBSCAN]** In this Question, you are supposed to implement **DBSCAN algorithm from**

scratch on dataset2 provided here and dataset3 provided here. You also need to compare and contrast your observations from above with K-Means applied on both datasets. **However, you can use predefined libraries to implement K-means.**

(a) (1 mark) Visualize the data in dataset2. Then, find a suitable **range of values for epsilon** (a hyperparameter in DBSCAN algorithm) by using the 'Elbow Curve' of Datapoints plotted between K-Distance vs Epsilon. For simplicity, take only integer values for epsilon. **You can use predefined libraries to implement K-distance.**

Solution:

- Visualization of dataset2.

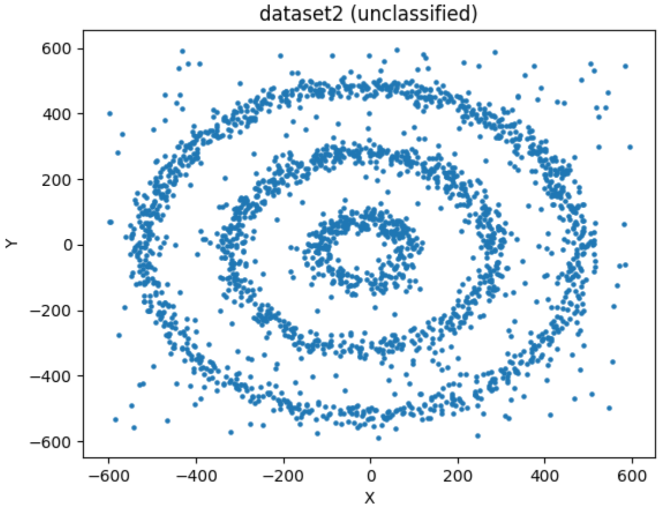


Figure 1: dataset2

- Elbow Curve

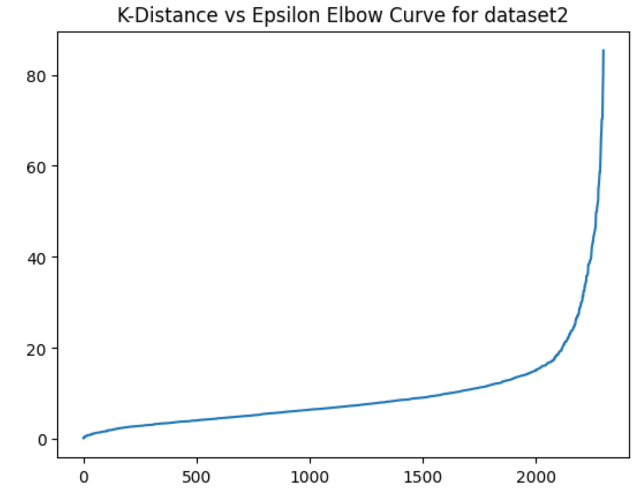


Figure 2: K-distance vs Epsilon

(b) (2 marks) Implement DBSCAN with the above suitable range of values of epsilon and detect the optimal value of epsilon, which gives the best clustering visually on the dataset. Show a visualization of the clusters formed for the best value of epsilon.

Solution:

- DBSCAN on dataset2 at e = 20

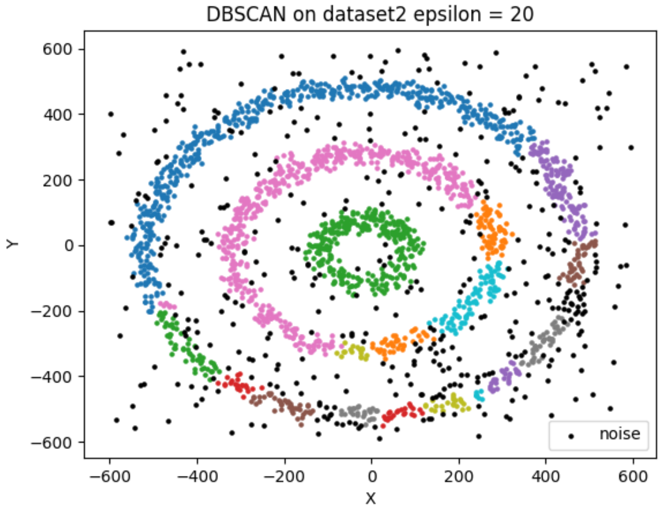


Figure 3: eps = 20

- DBSCAN on dataset2 at e = 24

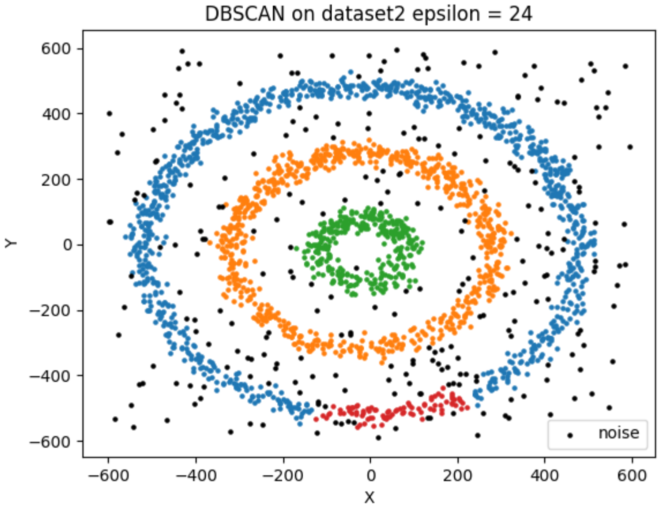


Figure 4: eps = 24

- DBSCAN on dataset2 at e = 28

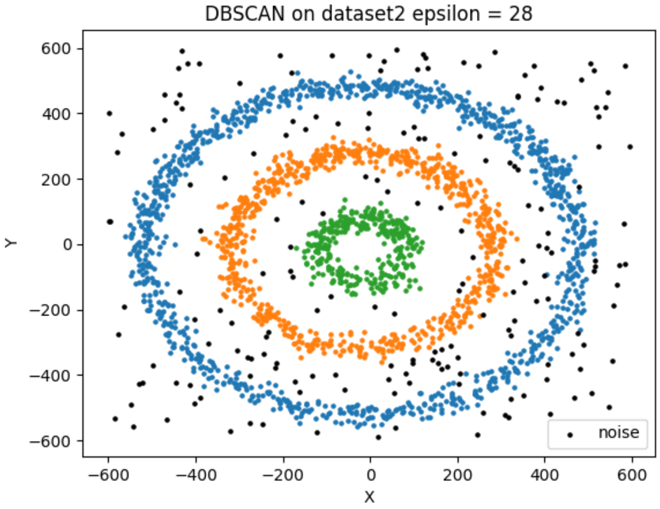


Figure 5: eps = 28

- DBSCAN on dataset2 at e = 32

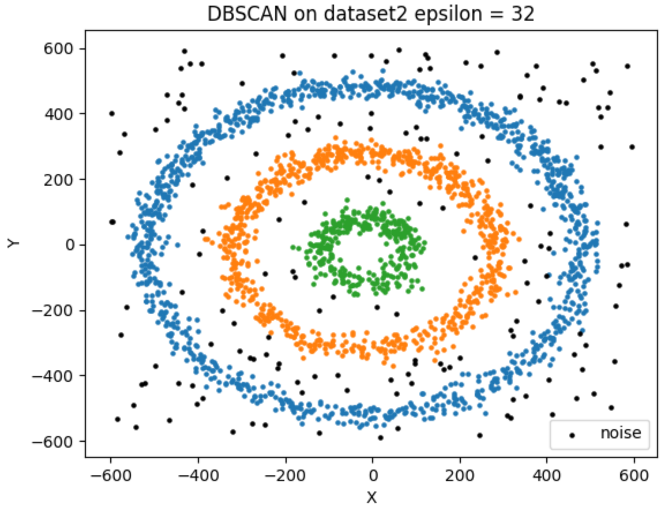


Figure 6: eps = 32

- DBSCAN on dataset2 at e = 36

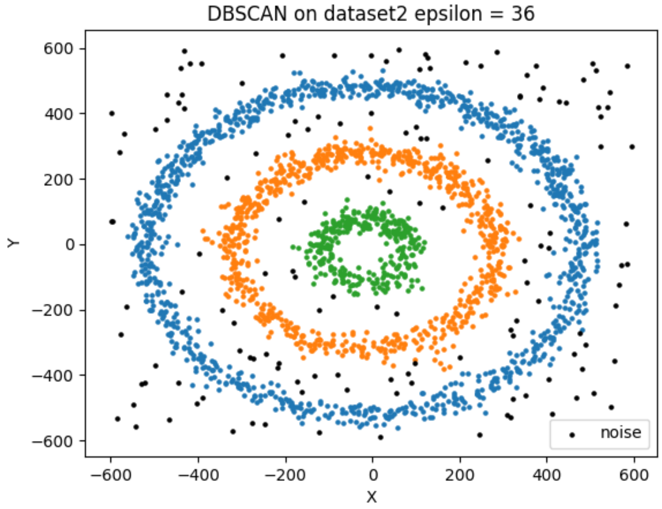


Figure 7: eps = 36

- DBSCAN on dataset2 at e = 40

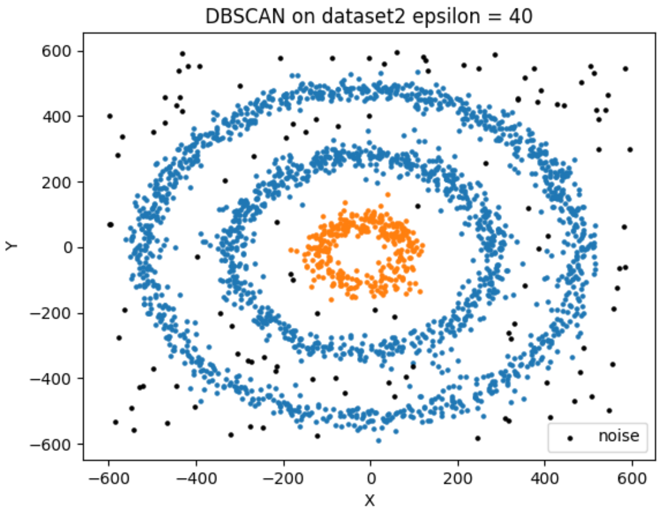


Figure 8: eps = 40

From above we can visually see that we get optimal clustering at around epsilon = 28 with 3 clusters. For smaller values, the clusters get fragmented into lots of smaller clusters. For bigger values, the clusters get combined into bigger clusters.

(c) (1.5 marks) Implement K-Means and use it on dataset2 with value of K (number of clusters) set to the optimum number of clusters that you get from (b) above. Suggest various techniques to improve the clustering by KMeans in this case.

Solution:
K-means on dataset2 at K = 3

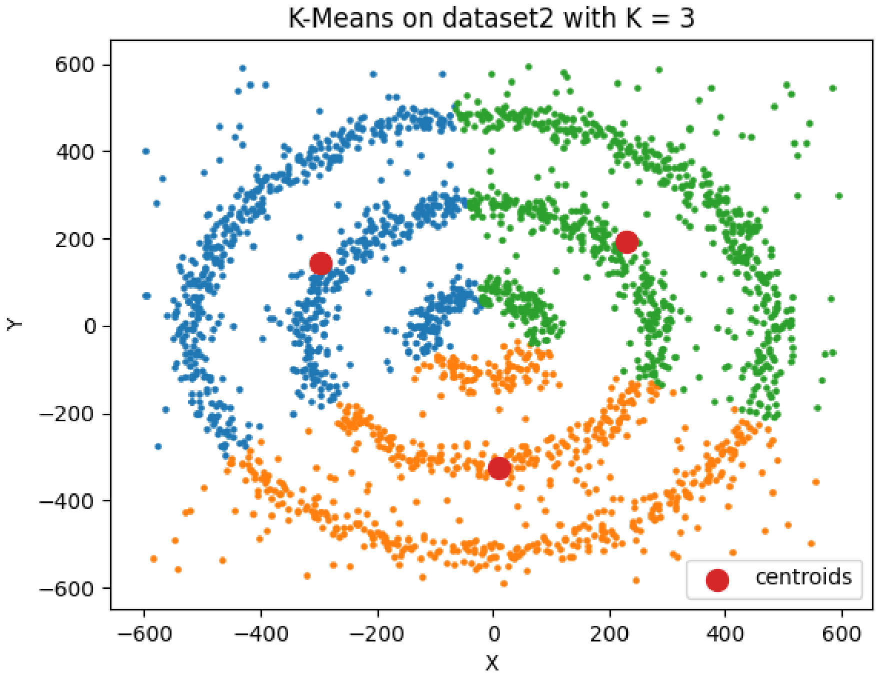


Figure 9: K = 3

- K-Means can be improved by selecting a different number of clusters. But this is not relevant in this case as we already found the optimal number of clusters from dbscan.
- Another method is to use a different distance metric. For this assignment, the distance metric we are using is the Euclidean Distance between the data points. But it is possible that a different metric might give much better results.

(d) (1.5 marks) Show a visualization of the data in dataset3. Use your implementation of DBSCAN with `minPts=15` on dataset3. Plot 'Elbow curve' to get an optimal range of values for `eps`. Detect the optimal value of epsilon which gives the best clustering visually on the dataset. Show a visualization of the clusters formed for the best value of epsilon.

Solution:

- Visualization of dataset3.

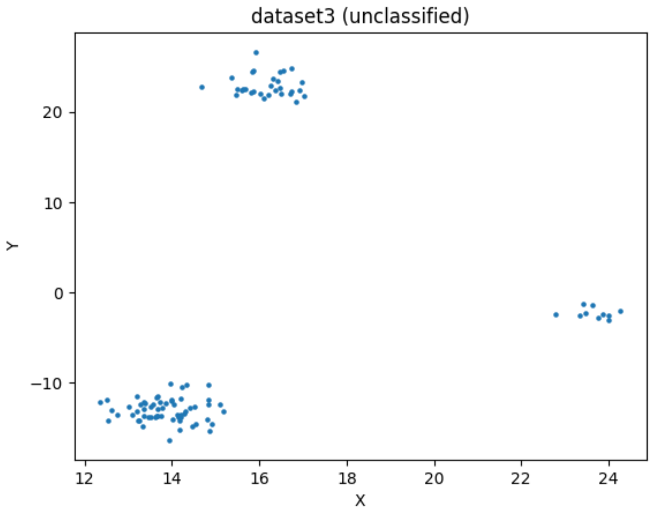


Figure 10: dataset3

- Elbow Curve

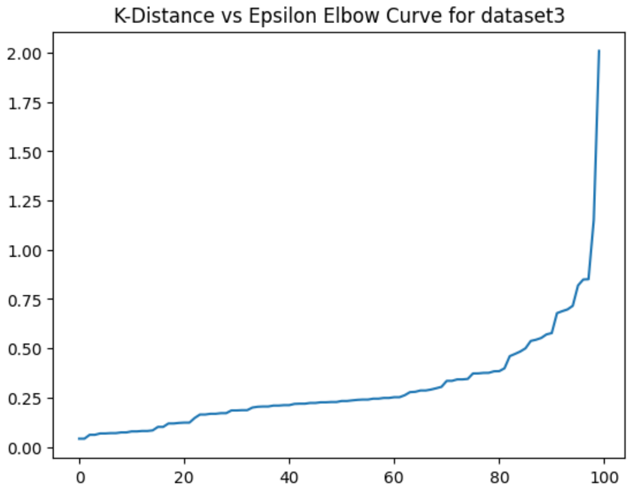


Figure 11: K-distance vs Epsilon

- DBSCAN on dataset2 at e = 1.0

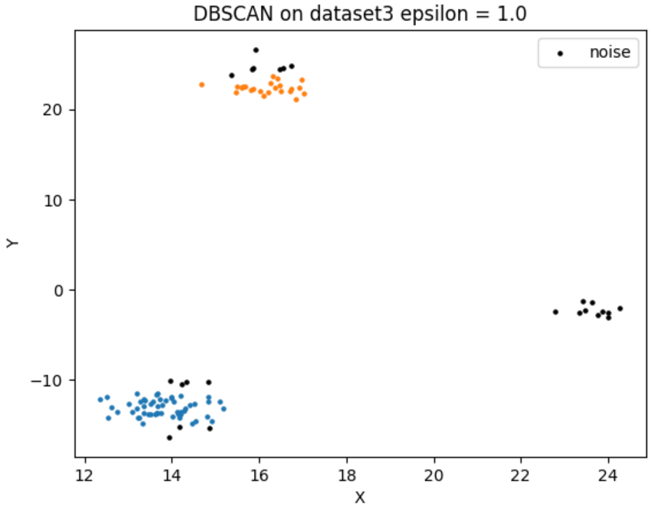


Figure 12: eps = 1.0

- DBSCAN on dataset2 at e = 1.5

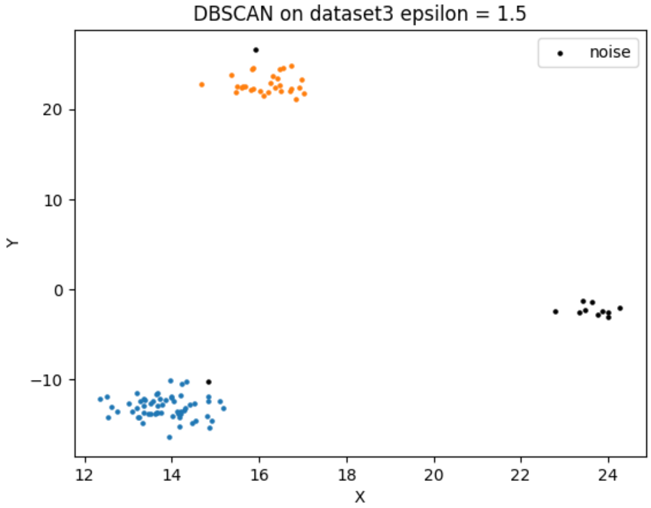


Figure 13: eps = 1.5

- DBSCAN on dataset2 at e = 2.0

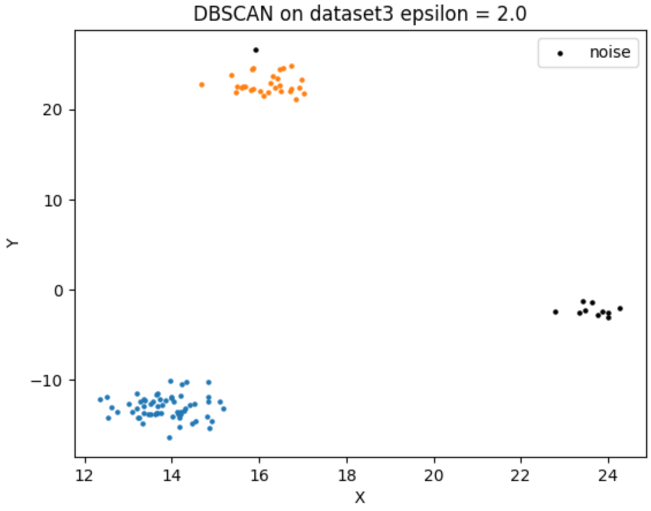


Figure 14: eps = 2.0

- DBSCAN on dataset2 at e = 2.5

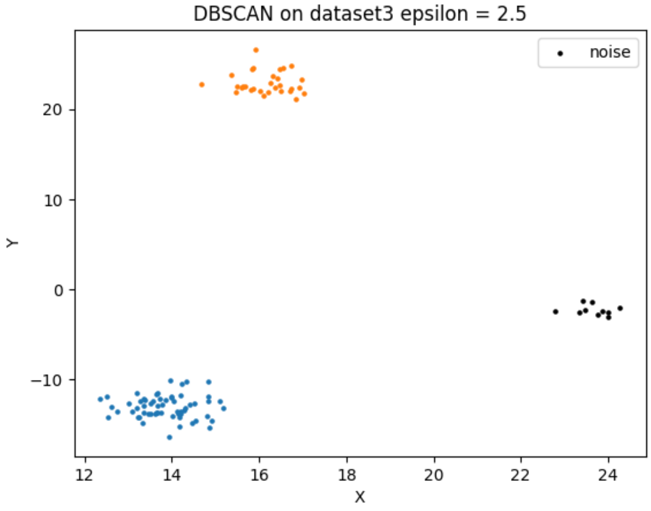


Figure 15: eps = 2.5

- DBSCAN on dataset2 at e = 3.0

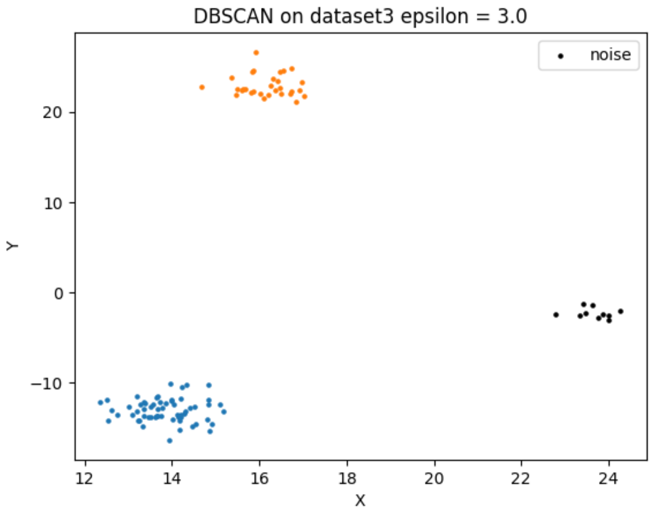


Figure 16: eps = 3.0

The optimal value for best clustering is around epsilon = 2.5.

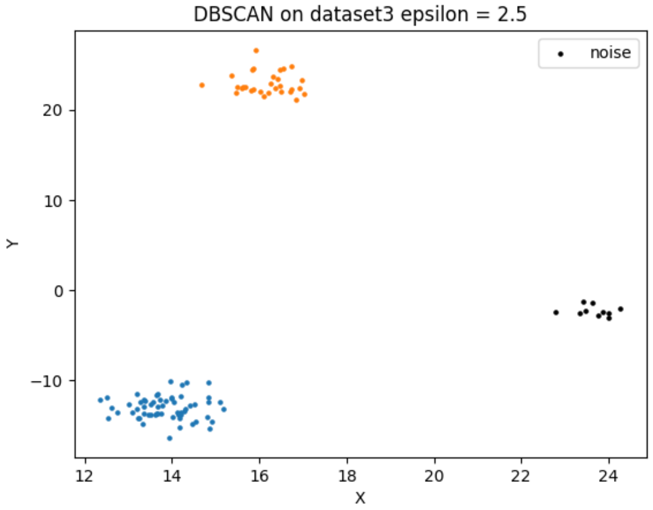


Figure 17: best clustering at eps = 2.5

(e) (1 mark) Now perform KMeans with K=3. Write your observations for obtained results in (d) and (e). Did we give you bad initialization values?

Solution:
K-means on dataset3 at K = 3

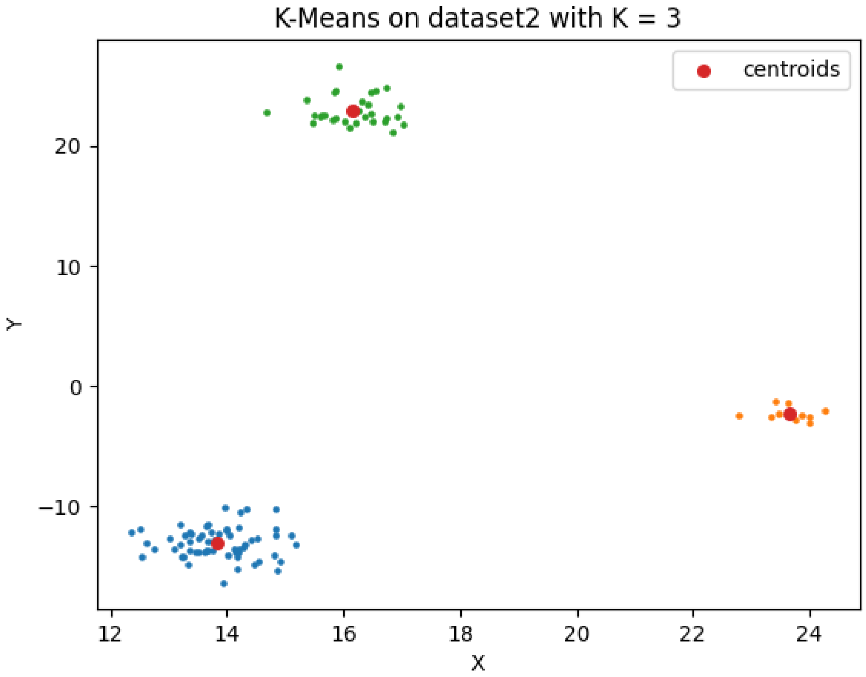


Figure 18: K = 3

Observations:

- Initialization given for K-means (K=3) is good as it provides optimal clusters.
- Initialization given for DBSCAN (minPts=15) is bad as we can see that the third cluster is not recognized and all those points get labelled as noise. This is because minPts = 15, but there are less than 15 points in that cluster. So no core point is recognized inside it and hence the cluster is not formed.

(f) (1 mark) Based on all your learnings from this question, state the relative pros and cons of KMeans vs DBSCAN.

Solution:
K-Means

- Pros:
  - K-means clustering is easy to understand, implement and interpret, compared to DBScan.
  - It is also fairly efficient computationally, and can quickly produce results for even large datasets.

- Cons:
  - It requires the number of clusters to be known before the computation.
  - It cannot identify noise points in the data.
  - It can be very sensitive to the intial conditions.

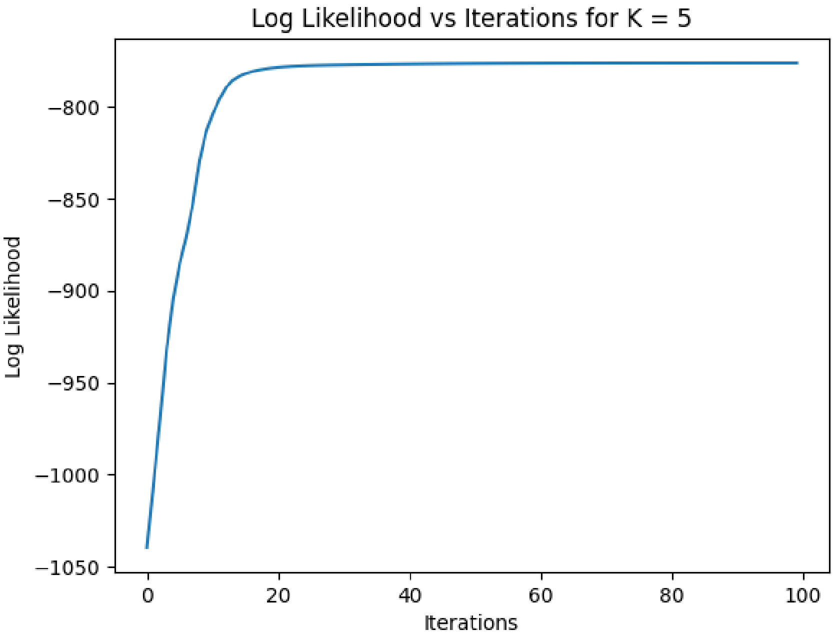
DBSCAN:

- Pros:
  - It can automatically determine the number of clusters in the dataset.
  - It can handle irregularly shaped/non-convex clusters also.
  - It is effective at identifying noise points in the dataset.
- Cons:
  - It is not very efficient computationally and may not work well for large datasets.
  - Its performance can vary wildly depending upon the choice of the hyper parameters(minPts, epsilon).

3. **[GMM]** In this question, you are supposed to implement the Expectation-Maximization algorithm for Gaussian mixture models on the given dataset4. The data can be found here.

   (a) (3 marks) Implement EM for GMM and plot the log-likelihood as a function of iterations.

   Solution: Plot of log-likelihood as a function of iterations:



We used a random value from K = 2 to 10 (5 in this case) as our initial model.

(b) (2 marks) Run EM for different numbers of Gaussians (k)(Try 2,3,4,5,6). Plot figures that can help in visualization and also log likelihood as a function of iteration for different values of k. Report the observations.

Solution: We have compiled results for K in range 2 to 10.

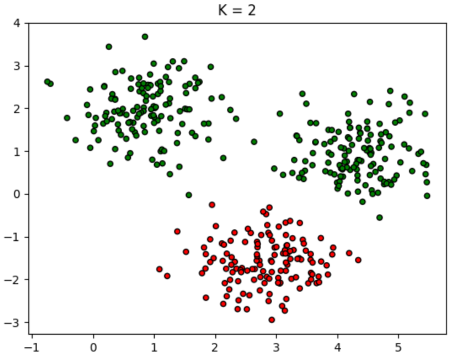


Figure 19: Scatter Plot

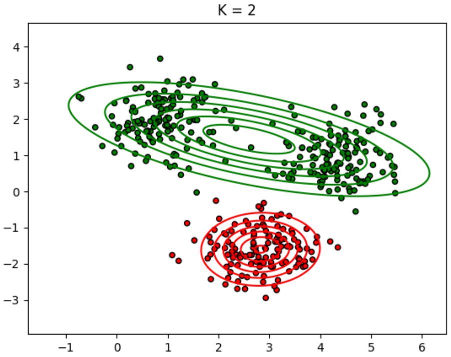


Figure 20: Scatter Plot with Underlying Gaussians

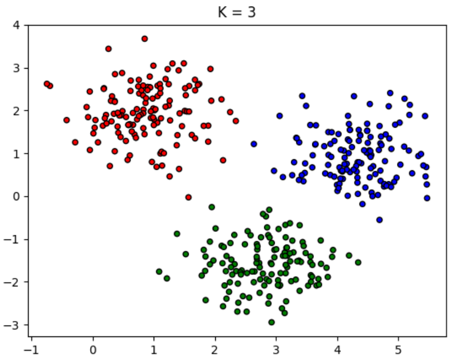


Figure 21: Scatter Plot

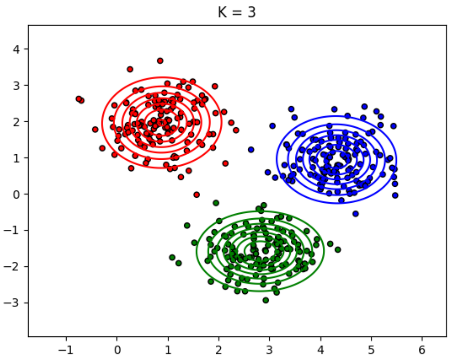


Figure 22: Scatter Plot with Underlying Gaussians

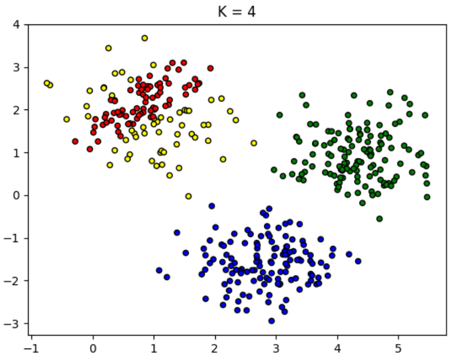


Figure 23: Scatter Plot

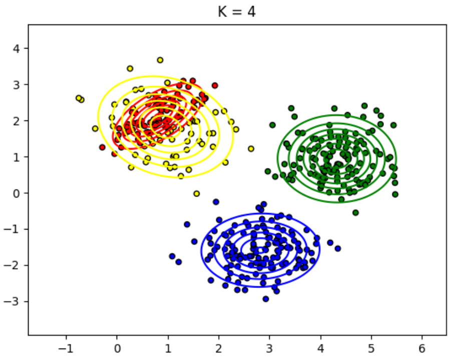


Figure 24: Scatter Plot with Underlying Gaussians

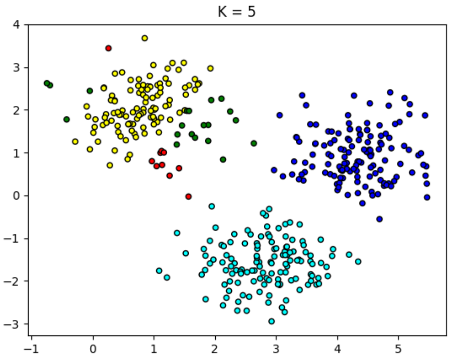


Figure 25: Scatter Plot

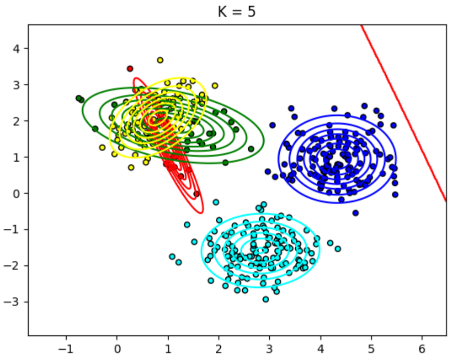


Figure 26: Scatter Plot with Underlying Gaussians

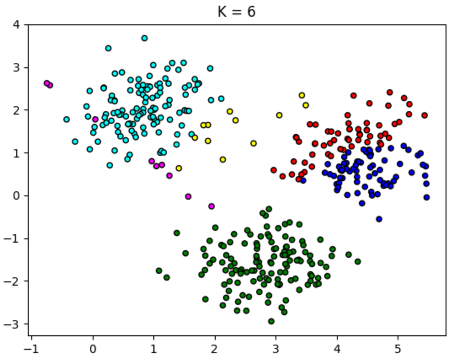


Figure 27: Scatter Plot

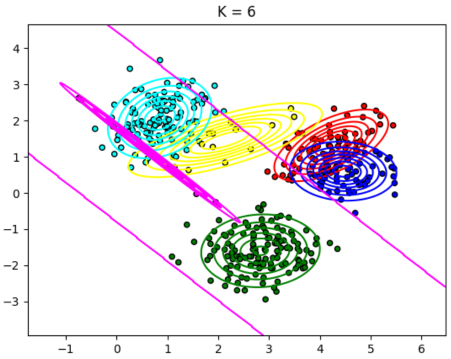


Figure 28: Scatter Plot with Underlying Gaussians

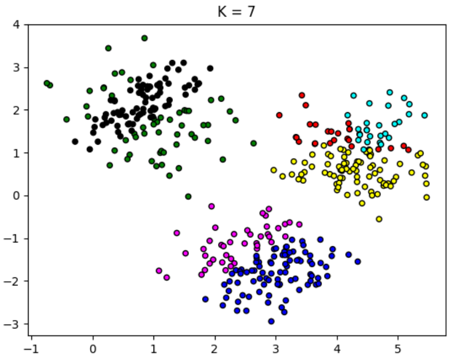


Figure 29: Scatter Plot

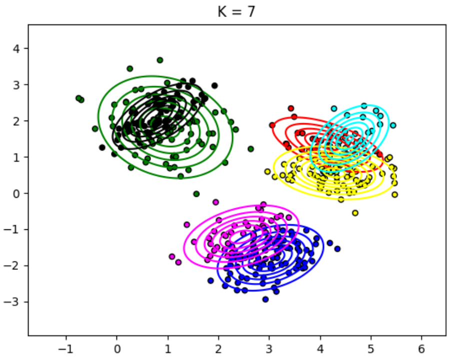


Figure 30: Scatter Plot with Underlying Gaussians

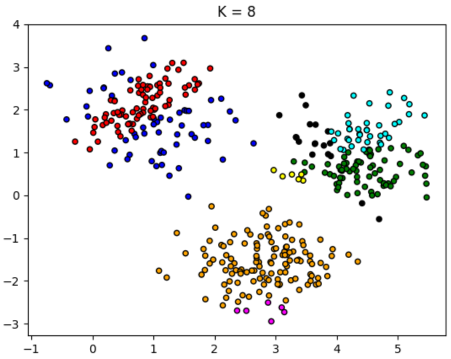


Figure 31: Scatter Plot

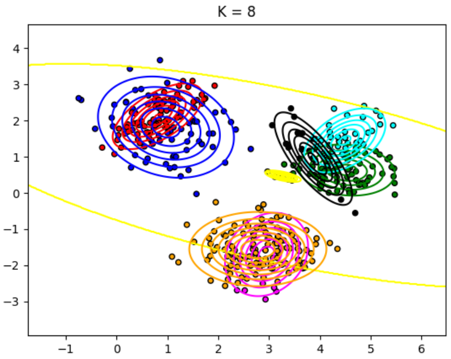


Figure 32: Scatter Plot with Underlying Gaussians

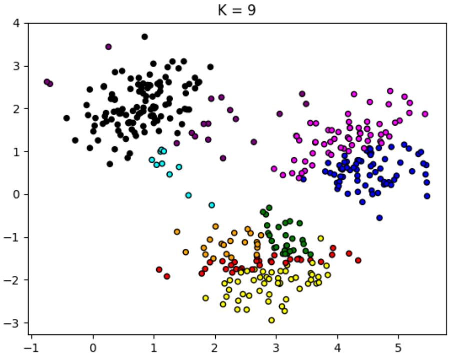


Figure 33: Scatter Plot

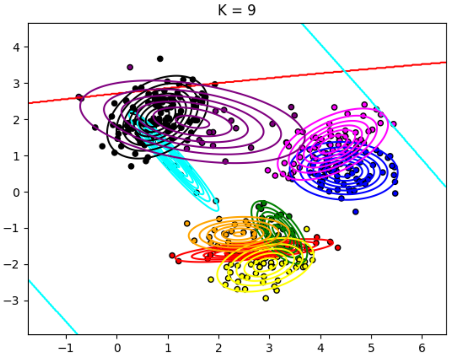


Figure 34: Scatter Plot with Underlying Gaussians

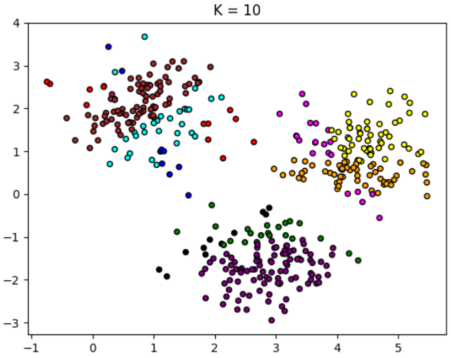


Figure 35: Scatter Plot

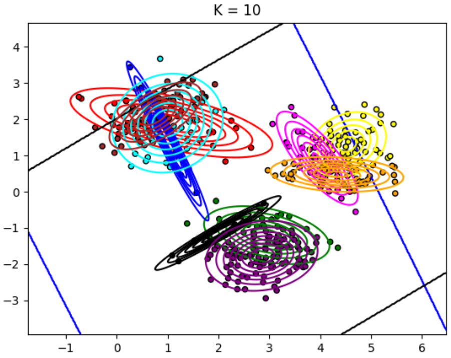


Figure 36: Scatter Plot with Underlying Gaussians

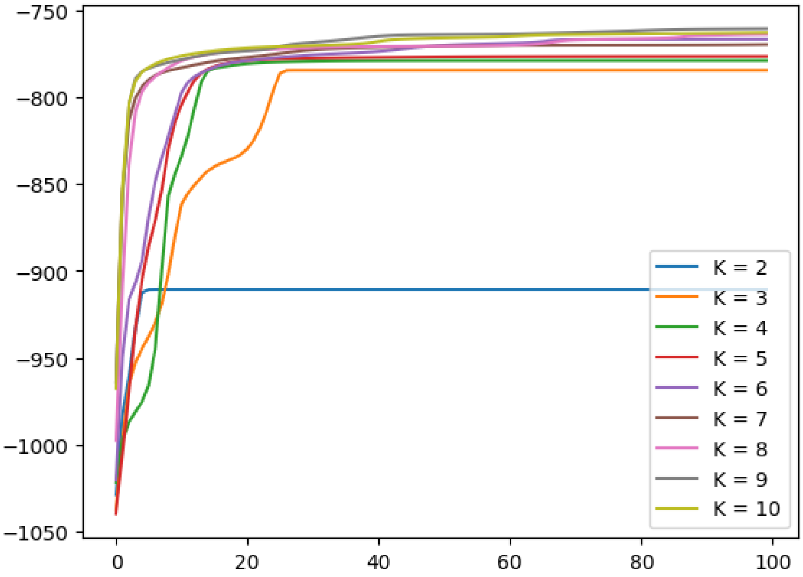


Figure 37: K vs Log-Likelihood

Observations:

- From visual inference, it is found that K=3 best fits the given data. For values larger than 3 the Gaussians over-fit the data while for K=2 underfitting is observed. But to have a mathematical proof of this, we would have to use metrics to find the optimal K.
- Also it is likely that just by going with the log-likelihood scores for finding the optimal K would not be correct as overfitting may occur for higher Ks.

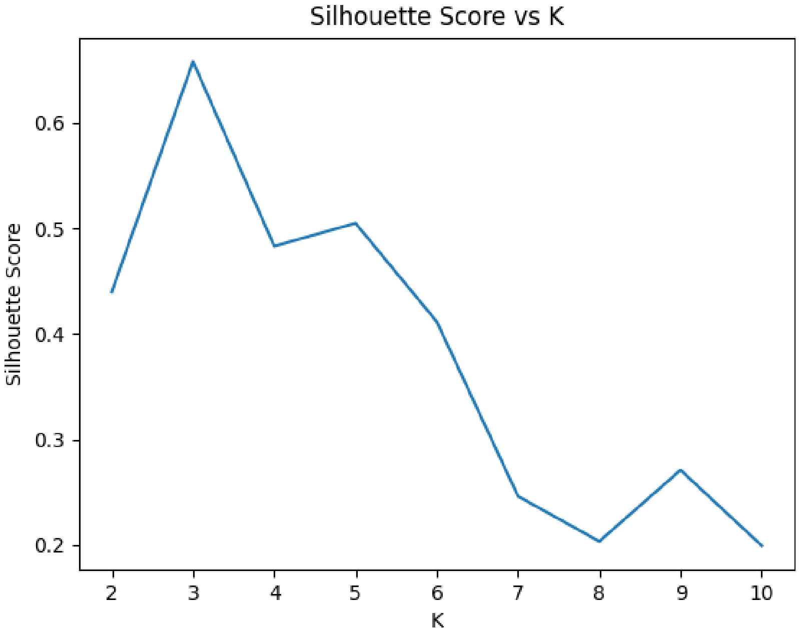
(c) (2 marks) Find the optimal k. There are several metrics like Silhouette score, Distance between GMMs, and Bayesian information criterion (BIC), or even you can use log-likelihood from the last question to infer. Give a clear explanation for your decision. Note: **You can use third-party libraries - sklearn or any other only in this subsection.**

Solution:

- Silhouette score:

  The Silhouette score is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette score for a sample is (b - a) / max(a, b). To clarify, b is the distance between a sample and the nearest cluster that the sample is not a part of. Hence, higher the Silhouette score, better the model with that number of clusters. Note that

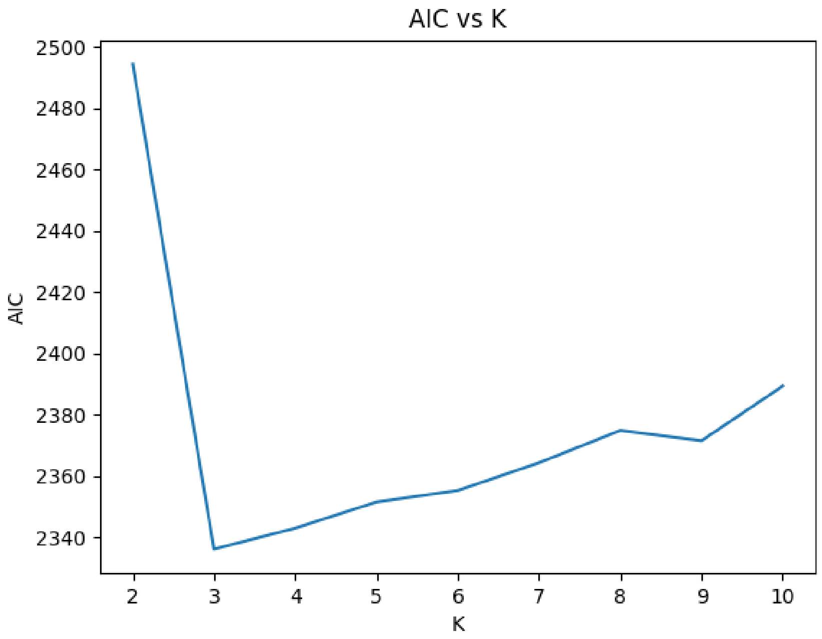
Silhouette score is only defined if : $2 \leq$ no. of labels $\leq$ no. of samples - 1. Higher the



From the graph it can be inferred that K = 3 is the optimal value according to Silhouette score. Silhouette score is a good metric as it tries to maximise the nearest-cluster distance and minimise the intra-cluster distance.

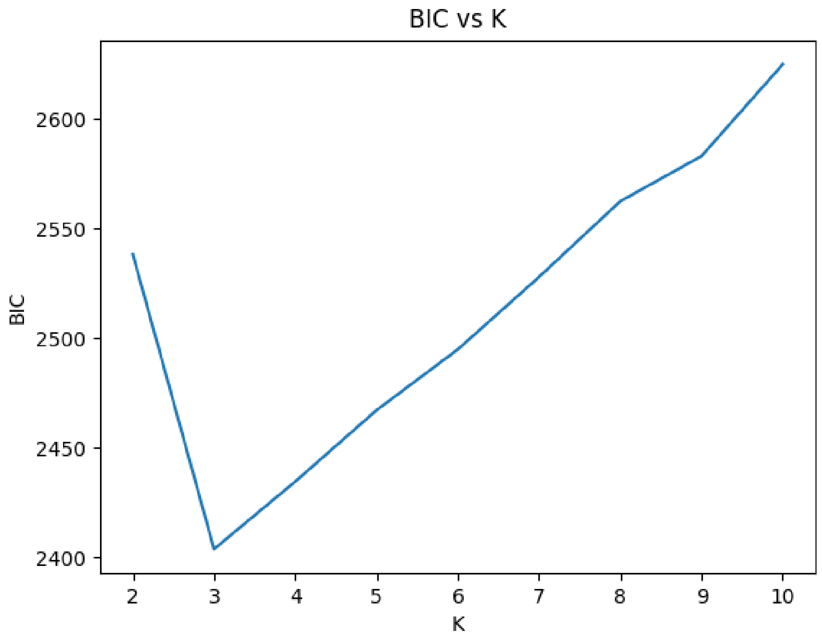
- Akaike information criterion (AIC):

  The Akaike information criterion (AIC) is an estimator of prediction error and thereby relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Thus, AIC provides a means for model selection. Note, lower the AIC score, better the model.



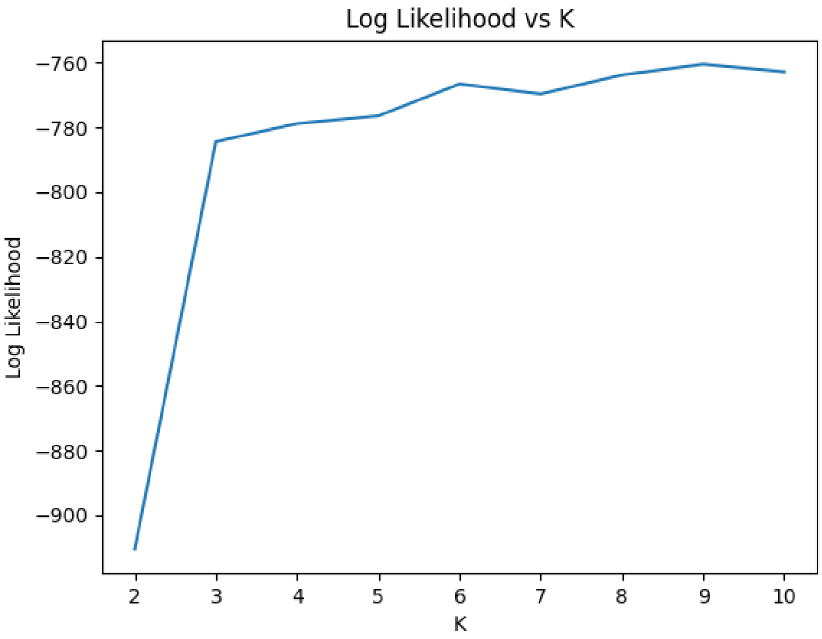
From the graph it can be inferred that K = 3 is the optimal value according to AIC. Note that AIC determines if a specific model is better compared to other models by using the number of parameters used and the maximum likelihood.

- Bayesian information criterion (BIC): The Bayesian information criterion (BIC) or Schwarz information criterion (also SIC, SBC, SBIC) is a criterion for model selection among a finite set of models; models with lower BIC are generally preferred.



From the graph it can be inferred that K = 3 is the optimal value according to BIC. BIC uses, the maximum likelihood , number of data points and number of parameters to judge a model.

- Maximum Log Likelihood: Likelihood determines the probability of the current parameters of a model to fit certain data. Maximum log likelihood thus denotes how well the model has fit to the current data.



As can be inferred from the previous two criteria (AIC, BIC), the models tend to overfit for high values of K and this is clearly visible from the Log Likelihood plots. Thus this isn't the ideal criteria to determine the optimal value of K.

Thus after going through multiple metrics, it is clear that $\mathbf{K = 3}$ is the optimal value to fit the given data.