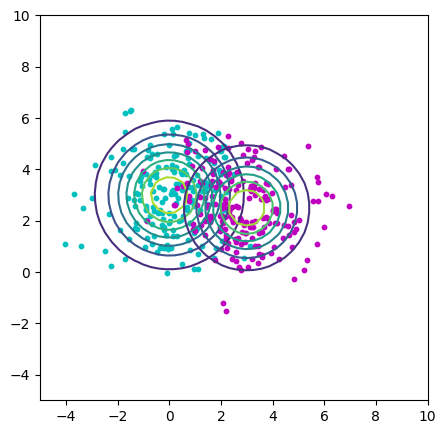
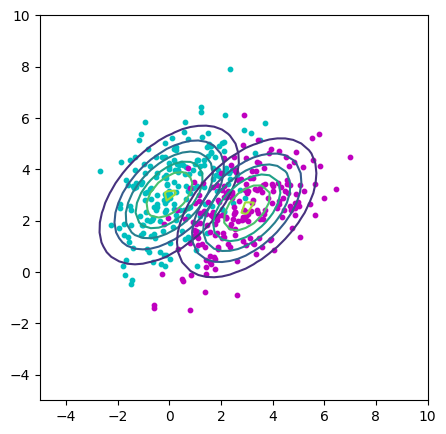
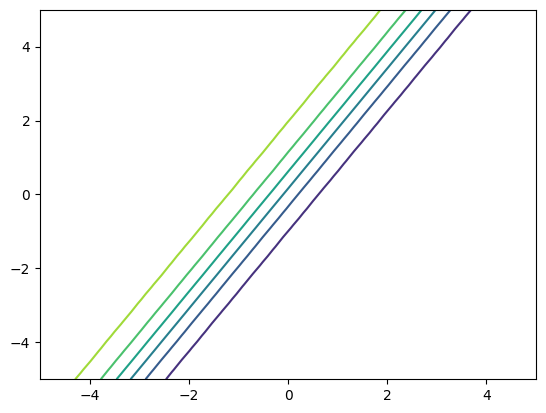
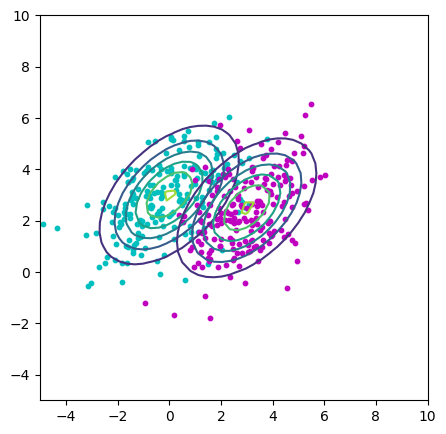
CO544 E/19/226

Machine Learning and Data Mining K.G.M. Madhushanka

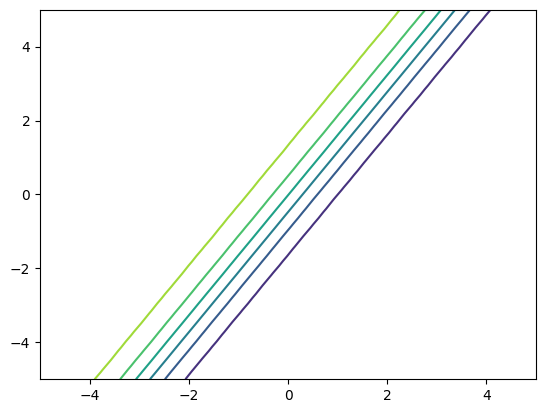
Lab 03

**1. Class Boundaries and Posterior Probabilities**

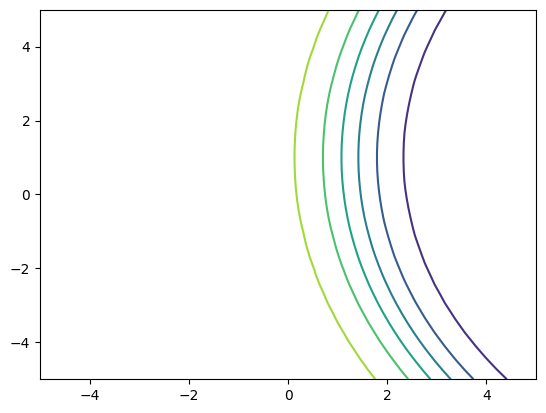
* ,



* ,



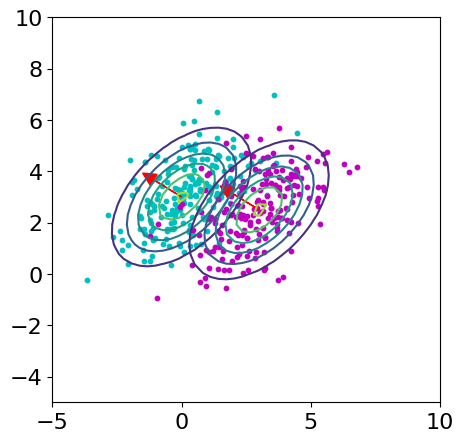
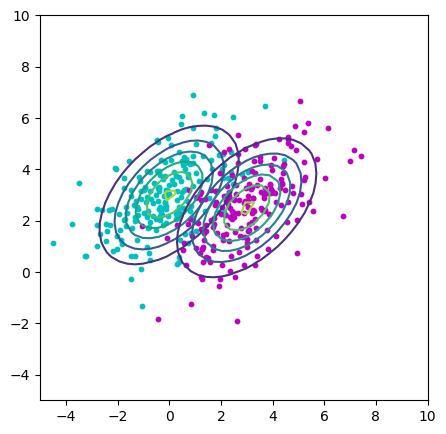
* ,

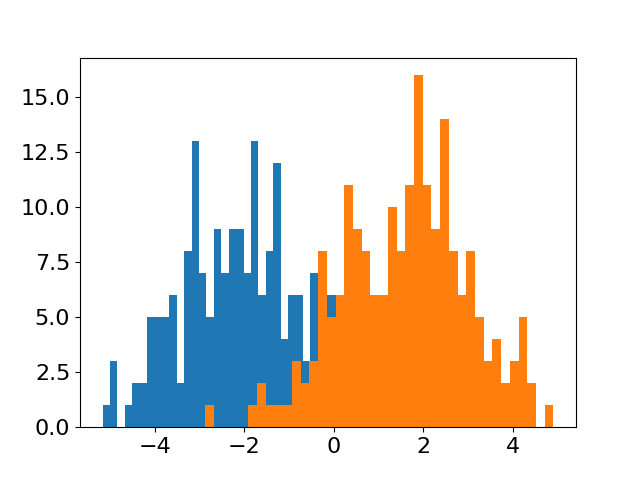
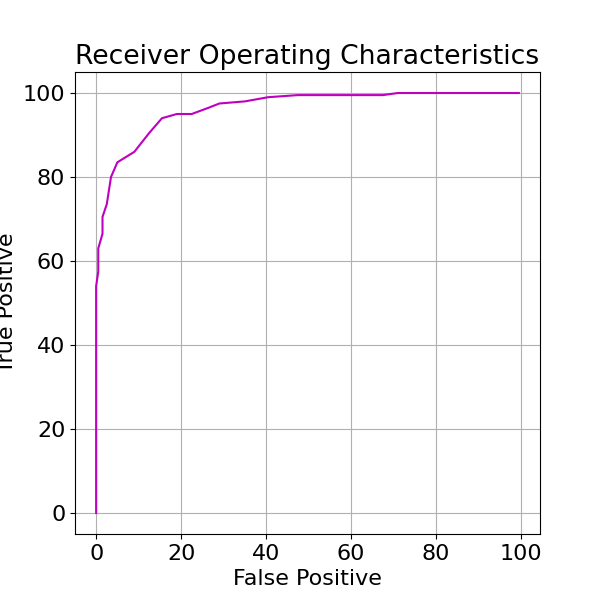


The plot illustrates how two categories are distinguished based on how probable they are to be in one category versus the other. The curved line on the plot that outlines the likelihood of being in a specific category is called the "posterior probability contour," and it shows the decision boundary. This boundary is where the chance of being in either category is the same. If the plot aligns with our mathematical calculations, we should see the decision boundary positioned between the two categories, where the probabilities of being in either category are nearly equal.

**2. Fisher LDA and ROC Curve**

,

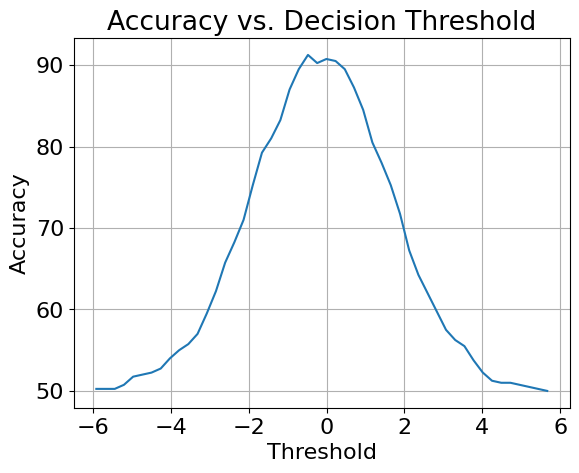
1.,2. 3.

4. 5.

6.

Fisher discriminant direction: [ 1.08333337 -0.66666669]

AUC for Fisher discriminant direction: 0.9691875 %

 7.

Threshold Accuracy

-1.4205894899154021 81.0

-1.1844690963724185 83.25

-0.948348702829434 87.0

-0.7122283092864503 89.5

-0.4761079157434658 91.25

-0.23998752220048214 90.25

-0.0038671286574976094 90.75

0.23225326488548603 90.5

0.46837365842847056 89.5

0.7044940519714542 87.25

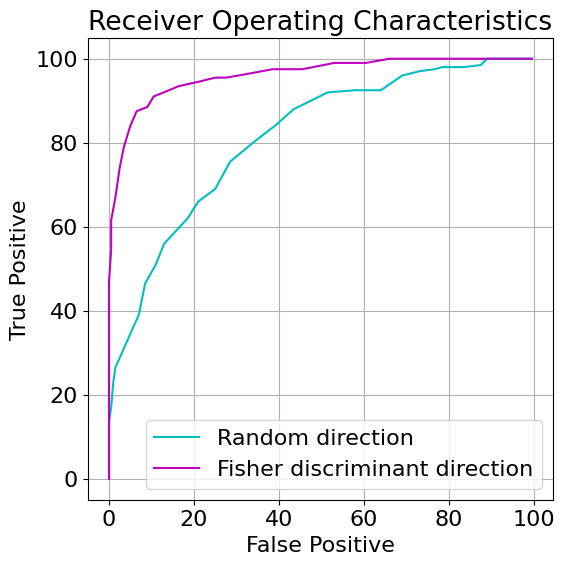
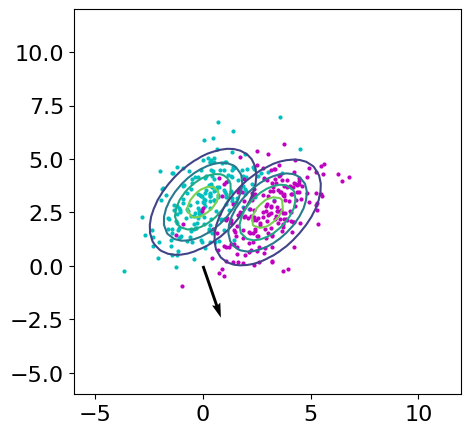
0.9406144455144387 84.5

1.1767348390574224 80.5

Highest accuracy (90.75%) can be gained by using -0.0038671 threshold value.

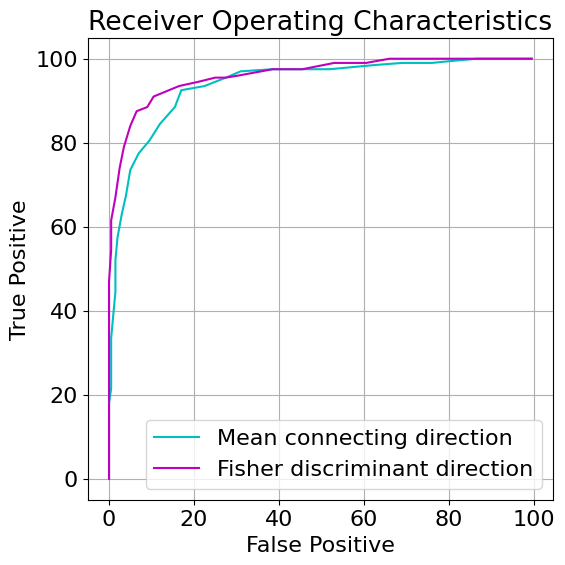
8.

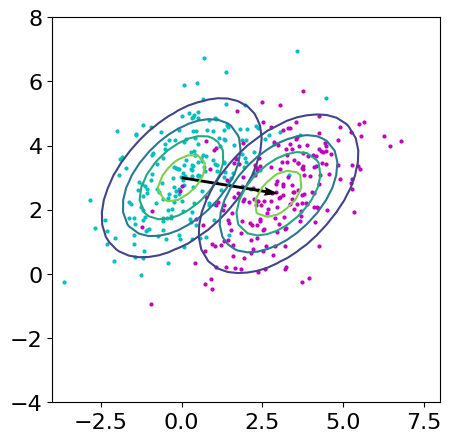
ROC curve (on the same scale) for A random direction (instead of the Fisher discriminant direction)



Random direction : [0.16627595694764397, -0.4839280865164908]

AUC for random direction: 0.808525 %

ROC curve (on the same scale) for Projections onto the direction connecting the means of the two classes.



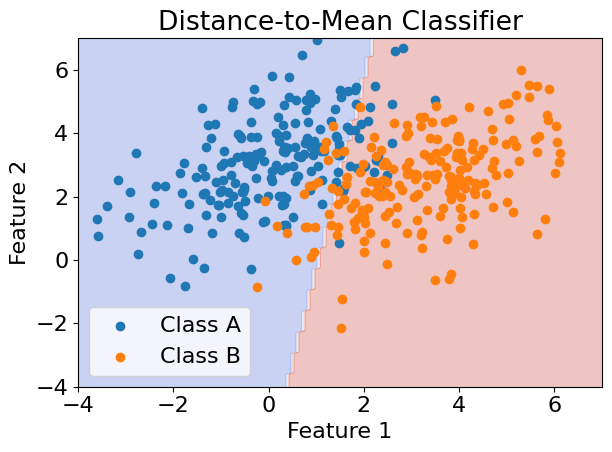
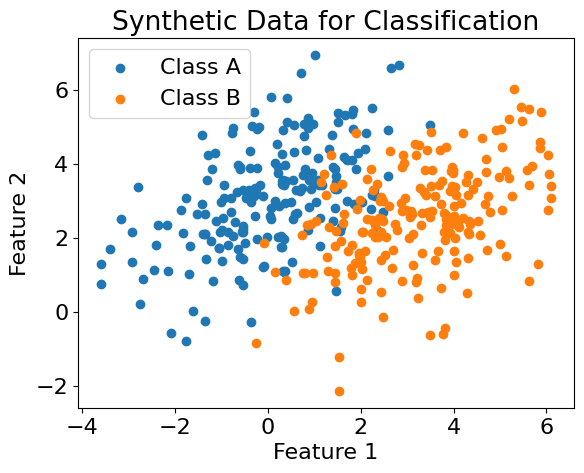
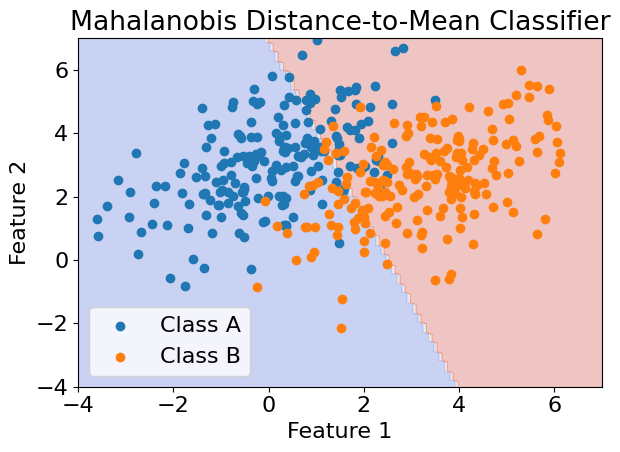
Mean connecting direction : [3.0, -0.5]

AUC for random direction: 0.934075

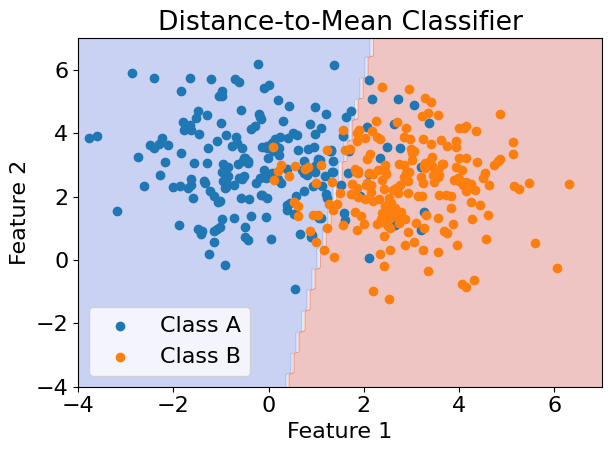
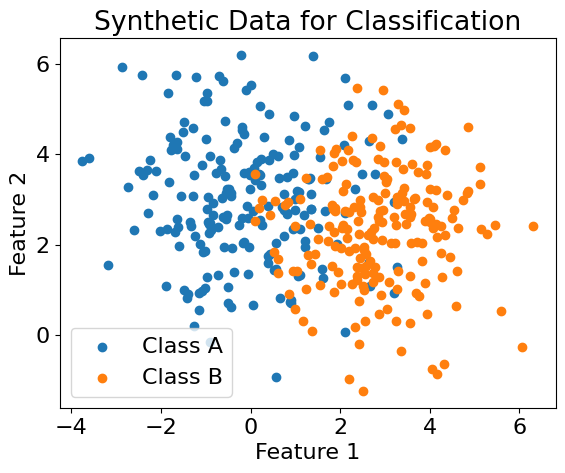
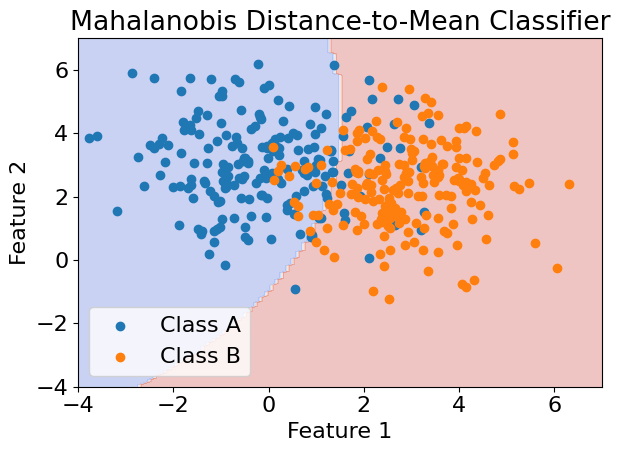
**3. Mahalanobis Distance**

*Distance to Mean Classifier:* This classifier calculates the Euclidean distance from each point to the mean of each class. The point is then classified based on which class mean it is closest to in terms of Euclidean distance. This method assumes that all dimensions are equally important and does not take into account the covariance of the data.

*Mahalanobis Distance Classifier:* This classifier calculates the Mahalanobis distance from each point to the mean of each class. The Mahalanobis distance takes into account the covariance of the data and is not affected by different scales in the data. The point is then classified based on which class mean it is closest to in terms of the Mahalanobis distance.

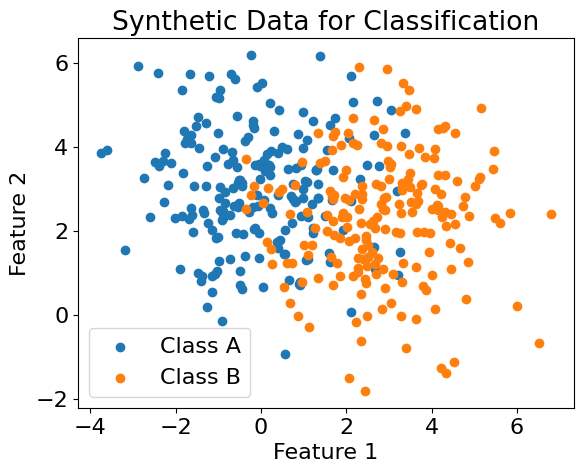
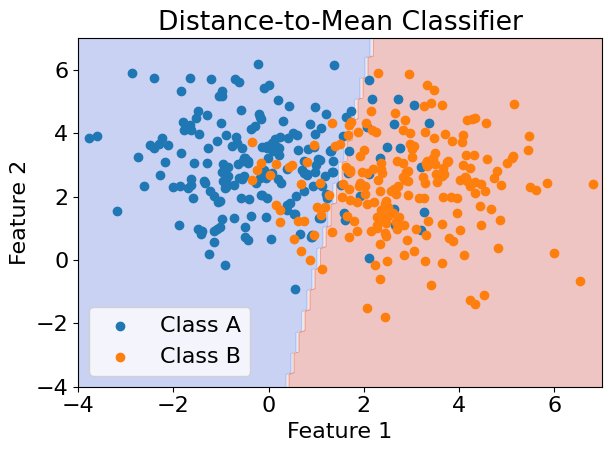
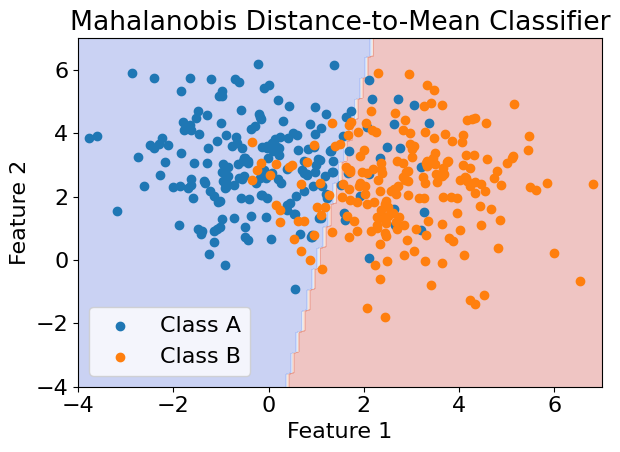
 a. ,

In this case, the covariance matrices C1 and C2 are the same, but the means m1 and m2 are different. The distance\_to\_mean\_classifier does not take into account the covariance, and so it draws a straight-line decision boundary with a positive gradient. The mahalanobis\_distance\_classifier, however, does take into account the covariance, and so it draws a decision boundary with a negative gradient.

 b. ,

The distance\_to\_mean\_classifier uses Euclidean distance, which does not take into account the covariance of the data. This results in a straight-line decision boundary.

On the other hand, the mahalanobis\_distance\_classifier uses Mahalanobis distance, which does take into account the covariance of the data. This results in a decision boundary that can be a curve, not just a straight line

 c. ,

If the covariance is the same for both classes, the Mahalanobis distance reduces to the Euclidean distance (scaled by the variance if the variance is not 1).

***~This shows the advantage of the Mahalanobis distance in capturing the shape and orientation of the data, which can lead to better classification performance in some cases.~***