Pattern Classification and Recognition
ECE 681

Spring 2019
Homework #6: Linear Discriminant, Logistic Regression, and DANN

Due: 5:00 PM, Thursday, April 4, 2019
Grace Period Concludes: 11:30 PM, Tuesday, April 9, 2019

This homework assignment is worth **330 points**.
Each problem is worth some multiple of 10 points, and will be scored on the below letter scale.
The letter grades B through D may be modified by + (+3%) and A through D may be modified by a - (-3%).

A+ = 100%: Exceeds expectations, and no issues identified
A = 95%: Meets expectations, and (perhaps) minor/subtle issues
B = 85%: Issues that need to be addressed
C = 75%: Significant issues that must be addressed
D = 65%: Major issues, but with noticeable perceived effort
F = 50%: Major issues, and insufficient perceived effort
Z = 30%: Minimal perceived effort
N = 0%: Missing, or no (or virtually no) perceived effort

Your homework is not considered submitted until both components (**one self-contained pdf file** and your code) have been submitted. Please do not include a print-out of your code in the pdf file.

You should strive to submit this assignment by the due date/time. The grace period is intended to afford an opportunity to, for example, work through technical glitches that may arise when submitting, update your submission if you realize after submission that you submitted the wrong file or would prefer to answer a question differently, and provide some flexibility to manage your workload as it ebbs and flows during the semester (so you do not need to ask for an extension if circumstances arise that make it difficult for you to submit this assignment by the due date/time).

Comparing Linear Discriminant and Logistic Discriminant (and Bayes)

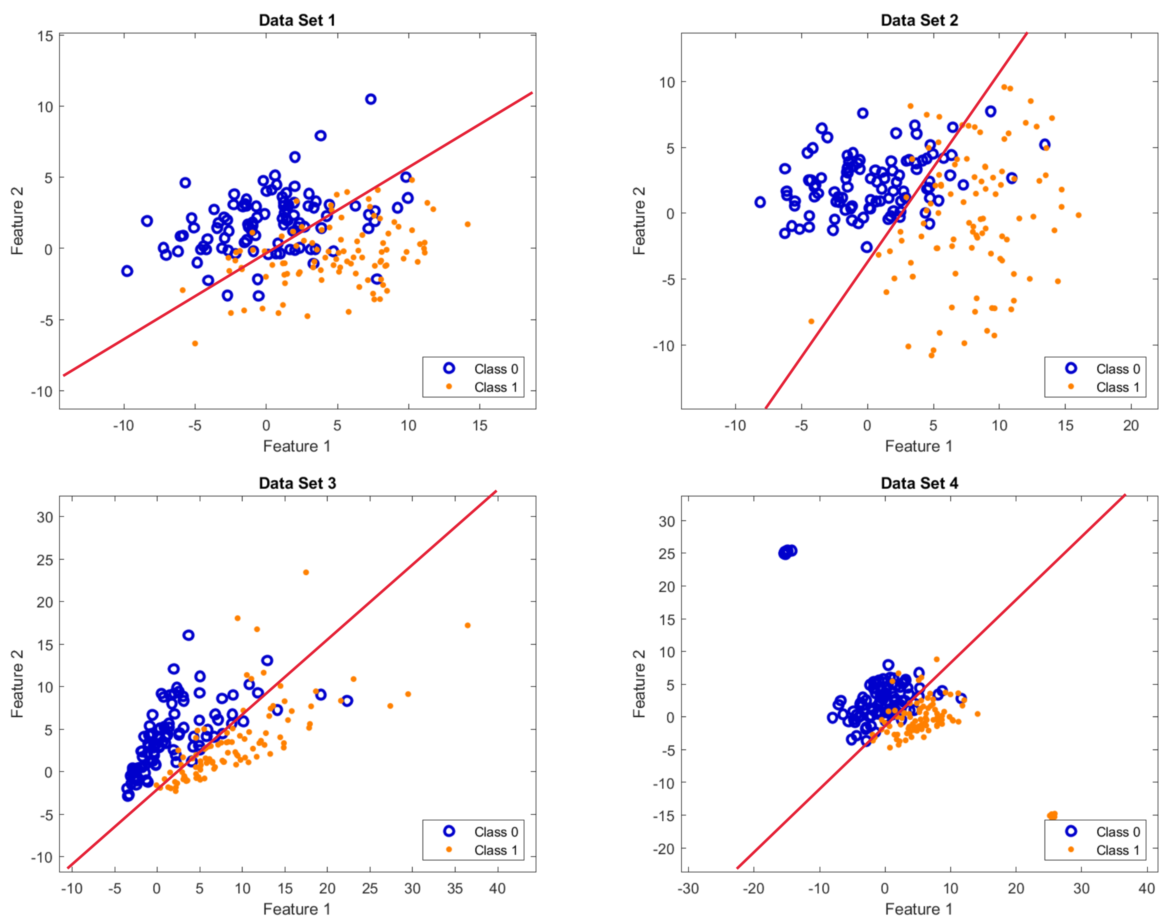
Linear Discriminant and Logistic Discriminant both assume a linear boundary separates the two classes; they differ in the assumptions they make to arrive at the resulting linear boundary and in how the coefficients for that boundary are computed. Here, you are going to explore how linear discriminant and logistic discriminant behave when operating on data that meet their underlying assumptions to different degrees. We will compare these linear classifiers to the Bayes classifier, to evaluate the implications of the linear boundary assumption.

Make sure you are able to apply a linear discriminant classifier.

Make sure you are able to apply a logistic discriminant classifier.

Regardless of whether you choose to write your own functions or leverage functions that may be available through Matlab or Python packages or libraries, you are responsible for understanding how the function(s) you are using work so you can effectively apply them to suit your needs and correctly interpret the results they provide.

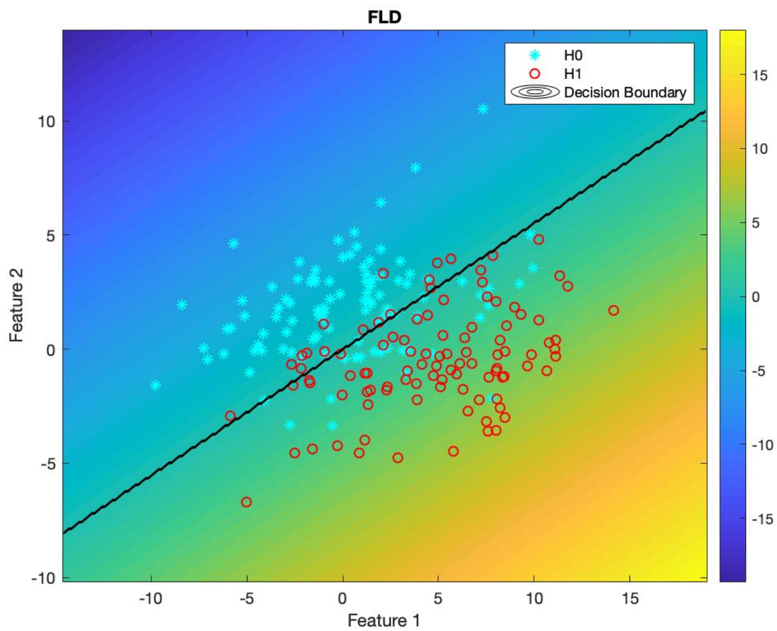
The following questions concern four data sets provided as a `csv` files that contain the data `dataSet1.csv`, `dataSet2.csv`, `dataSet3.csv`, and `dataSet4.csv`. Each csv file is organized such that each row contains the true class (either 0 or 1), followed by the associated (2-dimensional) feature vector. When you visualize the data sets, you should see this:



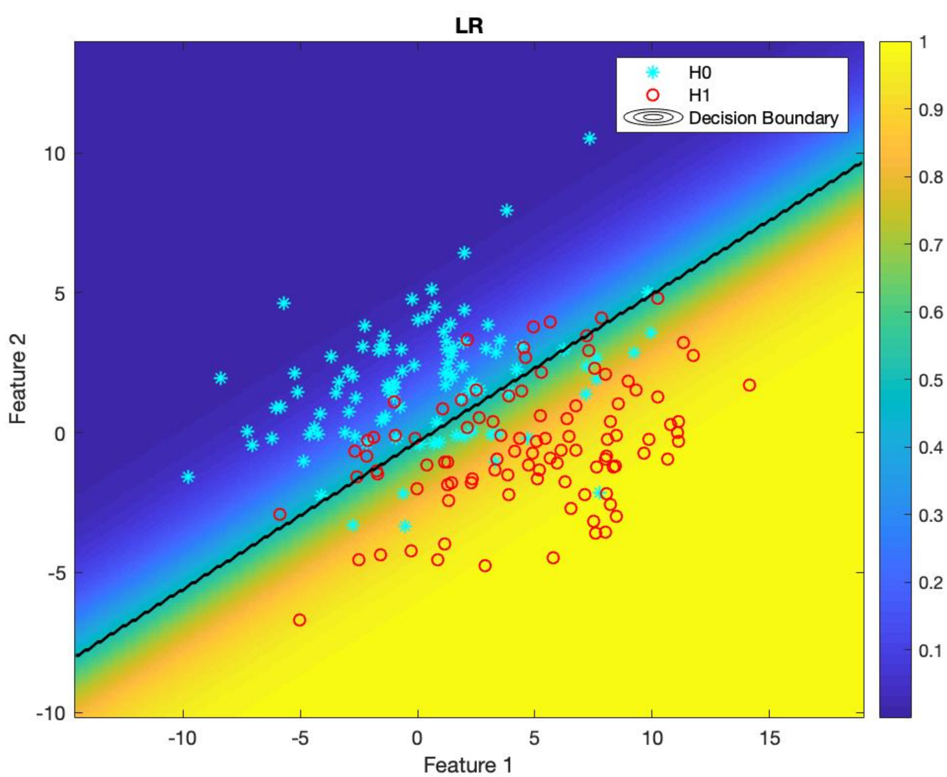
(10) 1. From visual inspection of these datasets (figures on previous page), qualitatively sketch on the provided figures what you would consider to be a "good" linear decision boundary (assuming the goal is maxP_{cd} (or minP_e).

(60) 2. Data set 1 is consistent with the assumptions underlying LDA – the data is Gaussian with means for the two classes that are distinct, and identical covariances.

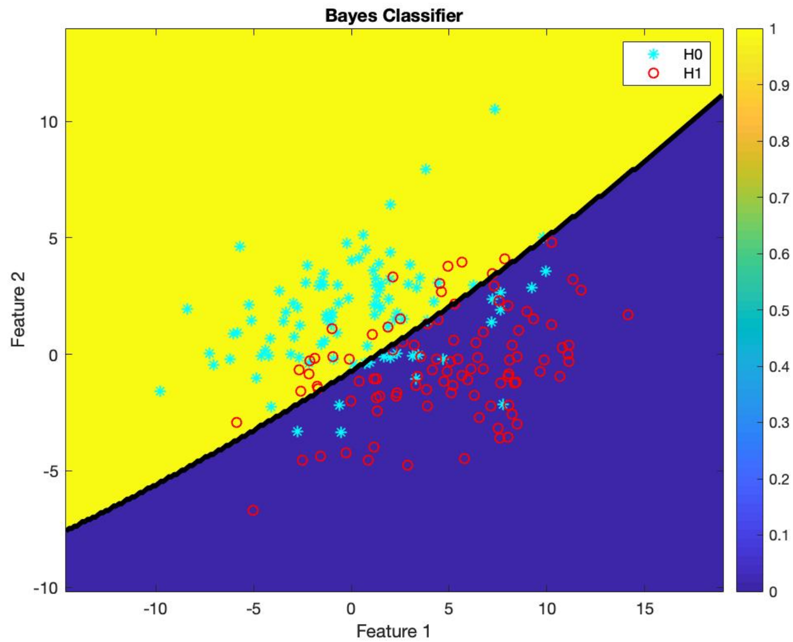
(a) Apply the linear discriminant to dataset 1, and plot the decision statistic surface with both the training data and the decision boundary corresponding to $\lambda(x) = 0$ superimposed on top.



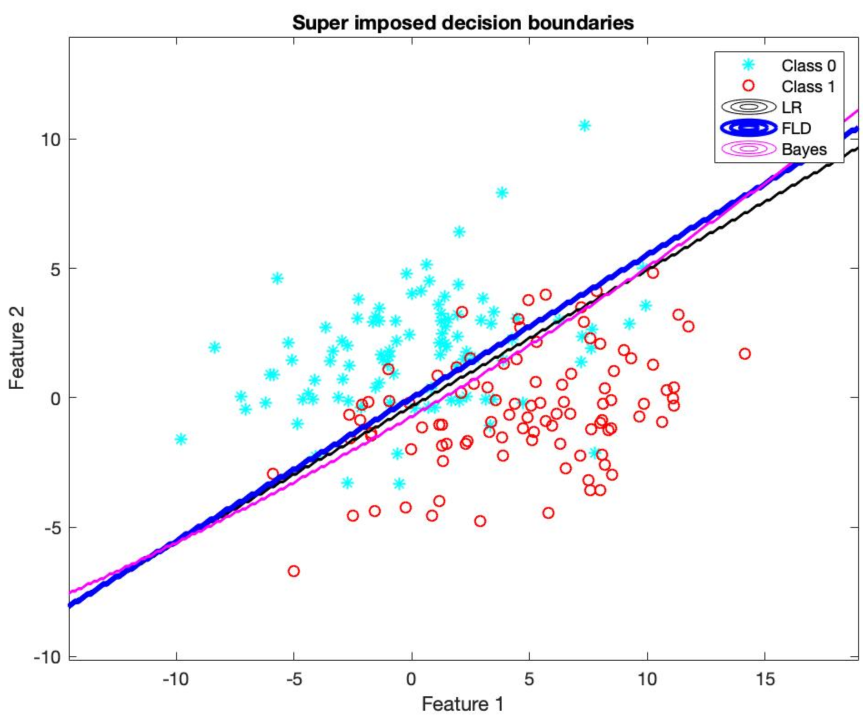
(b) Apply the logistic discriminant to dataset 1, and plot the decision statistic surface with both the training data and the decision boundary corresponding to $\lambda(x) = 0.5$ superimposed on top.



(c) Apply a Bayes Classifier to dataset 1, assuming the features may be dependent and the covariance matrices for the two classes are distinct (*i.e.*, estimate full covariance matrices for both class 0 and class 1), and plot the decision statistic surface for the ln-likelihood ratio with both the training data and the decision boundary under the assumptions of equal class priors and symmetric costs ($\ln\lambda(x) = 0$) superimposed on top.



(d) Compare the three decision boundaries (linear discriminant, logistic discriminant, and Bayes) by visualizing the data (replicating the figure provided for dataset 1 at the beginning of this section), and superimposing all three decision boundaries on top of the data.

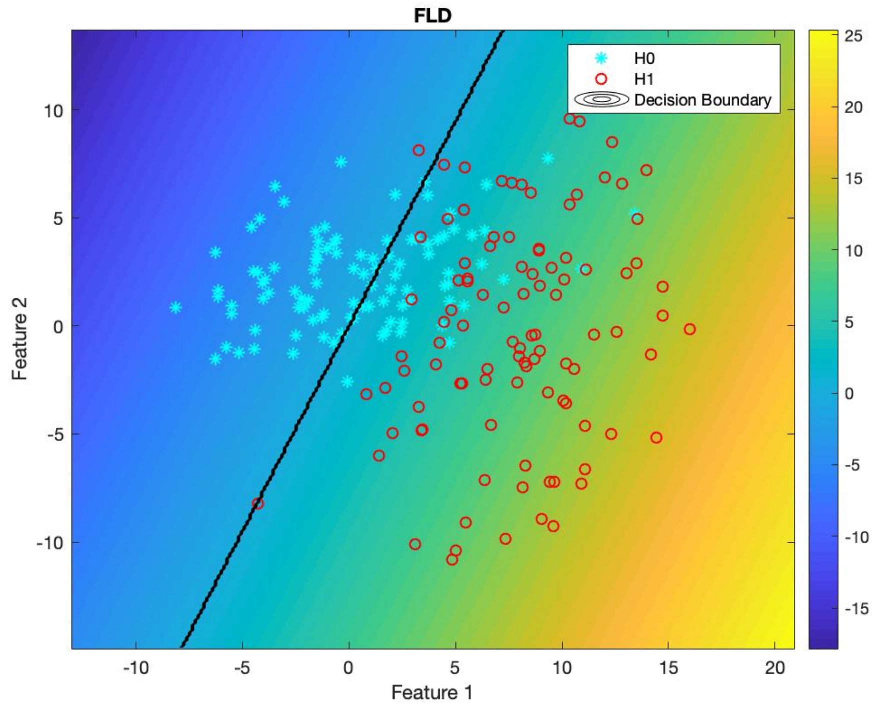


(e) How do the classifier decision boundaries compare to the decision boundary you sketched as a result of visual inspection of data set 1? Explain why the boundaries produced by these three classifiers are similar, or different from, the decision boundary you sketched.

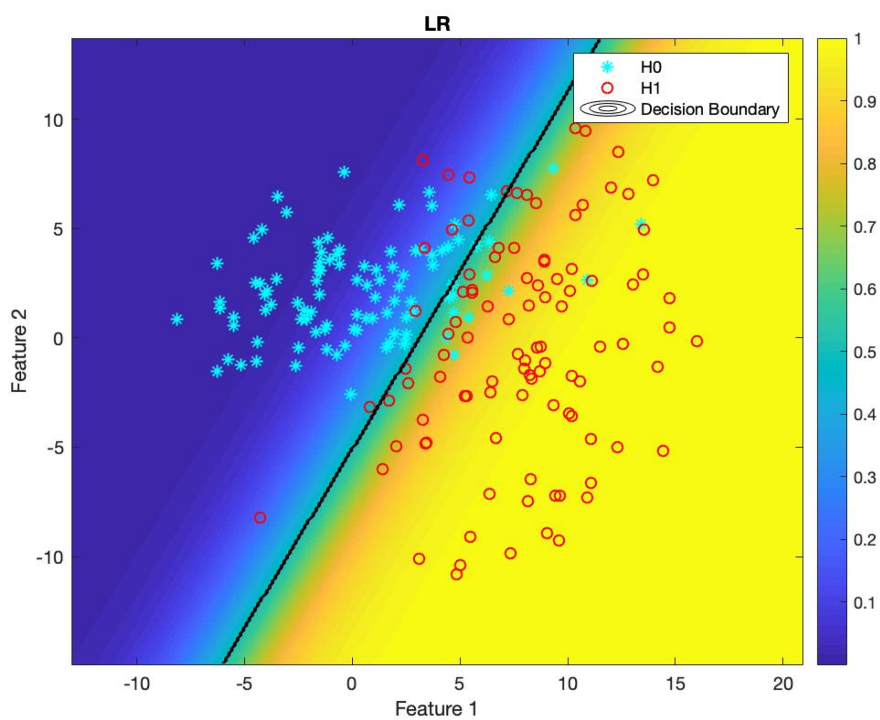
These decision boundaries produced between all three classifiers are very similar.
FLD and LR both assume a linear decision boundary, which was produced as expected.
FLD assumes Gaussian data with different means and identical covariances. Because this assumption held true with this data, the FLD produced a solid decision boundary, paralleling the Bayes and the LR.

(60) 3. Data set 2 is not consistent with the assumptions underlying LDA – the data is Gaussian with means for the two classes that are distinct, but the covariancesare also distinct.

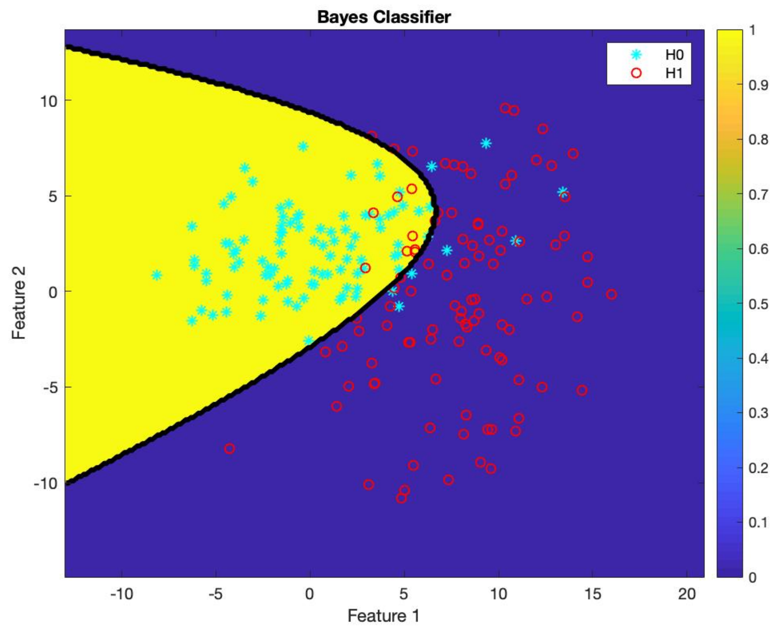
(a) Apply the linear discriminant to dataset 2, and plot the decision statistic surface with both the training data and the decision boundary corresponding to $\lambda(x) = 0$ superimposed on top.



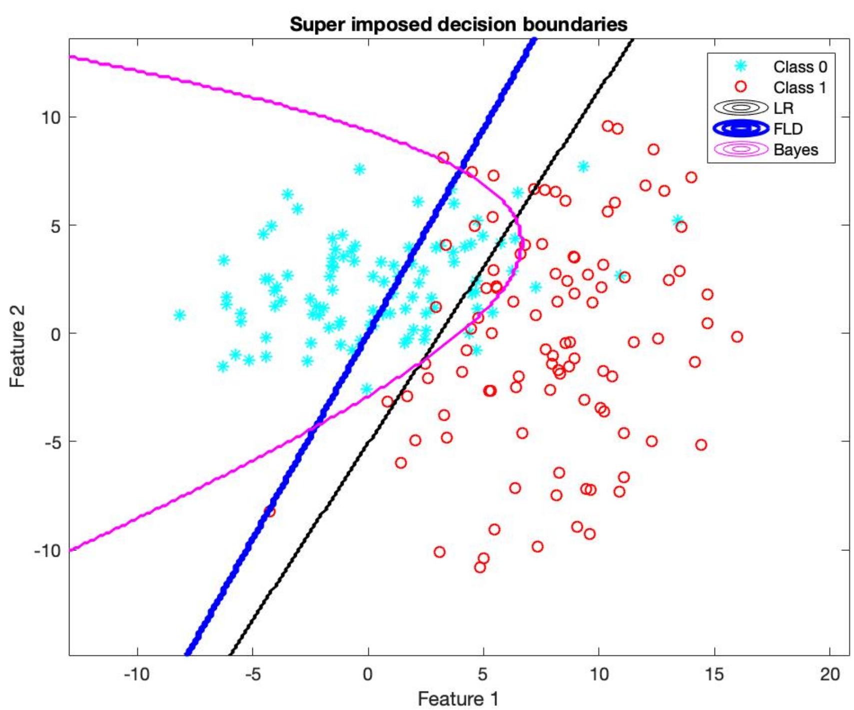
(b) Apply the logistic discriminant to dataset 2, and plot the decision statistic surface with both the training data and the decision boundary corresponding to $\lambda(x) = 0.5$ superimposed on top.



(c) Apply a Bayes Classifier to dataset 2, assuming the features may be dependent and the covariance matrices for the two classes are distinct (*i.e.*, estimate full covariance matrices for both class 0 and class 1), and plot the decision statistic surface for the ln-likelihood ratio with both the training data and the decision boundary under the assumptions of equal class priors and symmetric costs ($\ln\lambda(x) = 0$) superimposed on top.



(d) Compare the three decision boundaries (linear discriminant, logistic discriminant, and Bayes) by visualizing the data (replicating the figure provided for dataset 2 at the beginning of this section), and superimposing all three decision boundaries on top of the data.

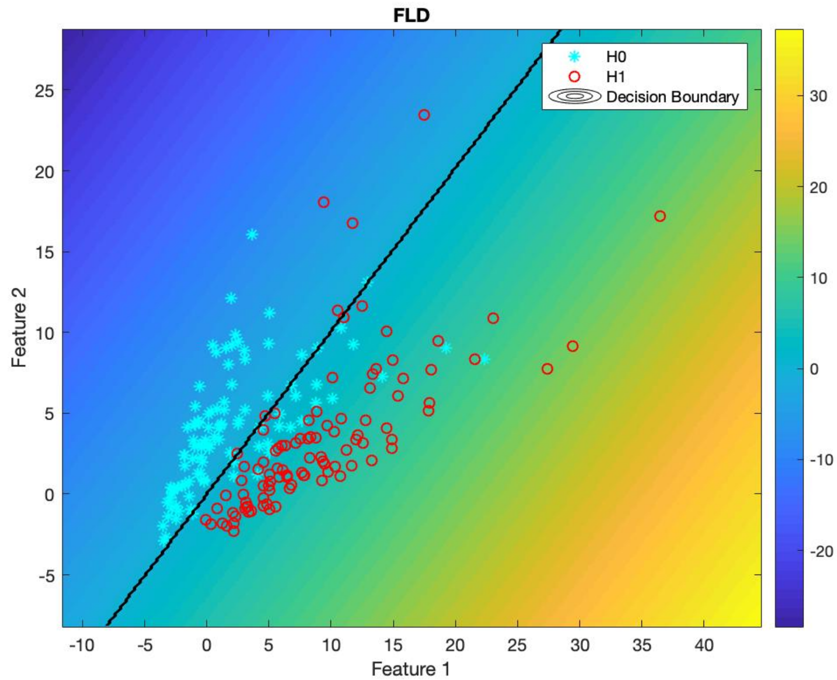


(e) How do the classifier decision boundaries compare to the decision boundary you sketched as a result of visual inspection of data set 2? Explain why the boundaries produced by these three classifiers are similar, or different from, the decision boundary you sketched.

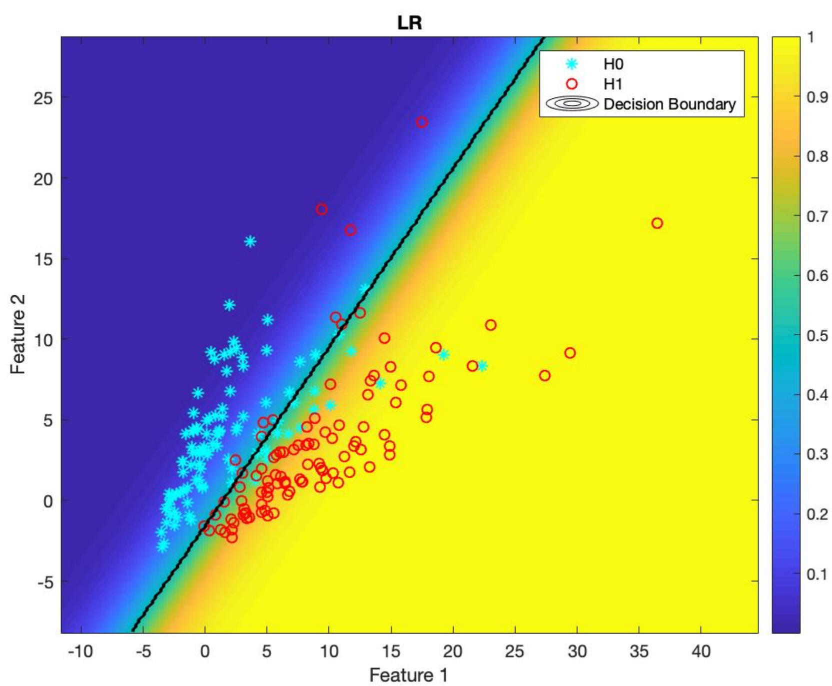
Now in this case, the classifiers produce very different decision boundaries. For FLD, the data no longer matches the underlying assumption that the covariances are the same. In this case, the covariances are distinct. Therefore, FLD performs poorly.
LR and Bayes still perform well because they do not hold this underlying assumption that the covariances are the same.

(60) 4. Data set 3 is not consistent with the assumptions underlying LDA – the data is not Gaussian, but rather is log-normal.

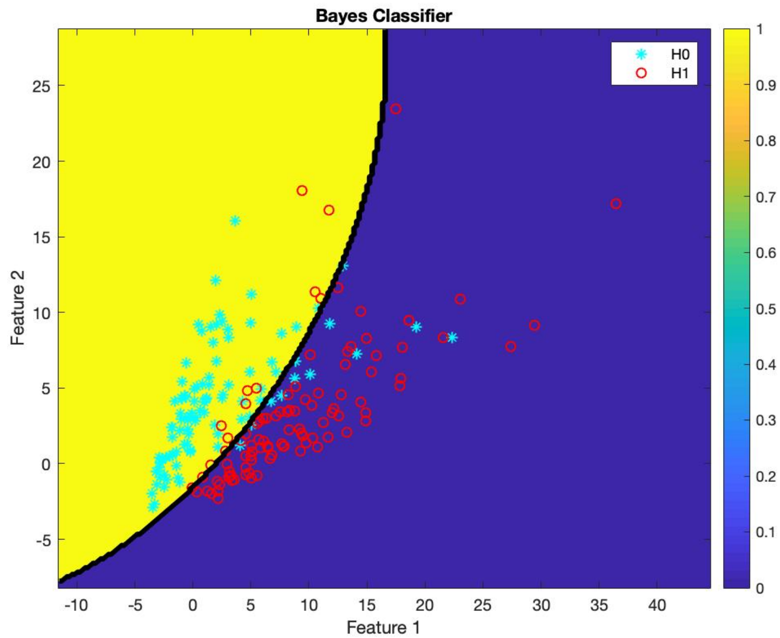
(a) Apply the linear discriminant to dataset 3, and plot the decision statistic surface with both the training data and the decision boundary corresponding to $\lambda(x) = 0$ superimposed on top.



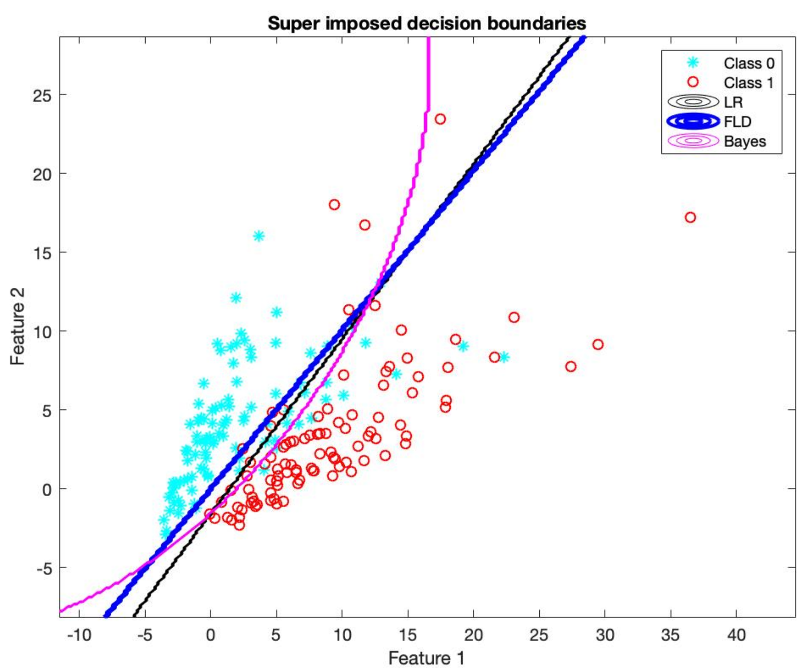
(b) Apply the logistic discriminant to dataset 3, and plot the decision statistic surface with both the training data and the decision boundary corresponding to $\lambda(x) = 0.5$ superimposed on top.



(c) Apply a Bayes Classifier to dataset 3, assuming the features may be dependent and the covariance matrices for the two classes are distinct (*i.e.*, estimate full covariance matrices for both class 0 and class 1), and plot the decision statistic surface for the ln-likelihood ratio with both the training data and the decision boundary under the assumptions of equal class priors and symmetric costs ($\ln\lambda(x) = 0$) superimposed on top.



(d) Compare the three decision boundaries (linear discriminant, logistic discriminant, and Bayes) by visualizing the data (replicating the figure provided for dataset 3 at the beginning of this section), and superimposing all three decision boundaries on top of the data.

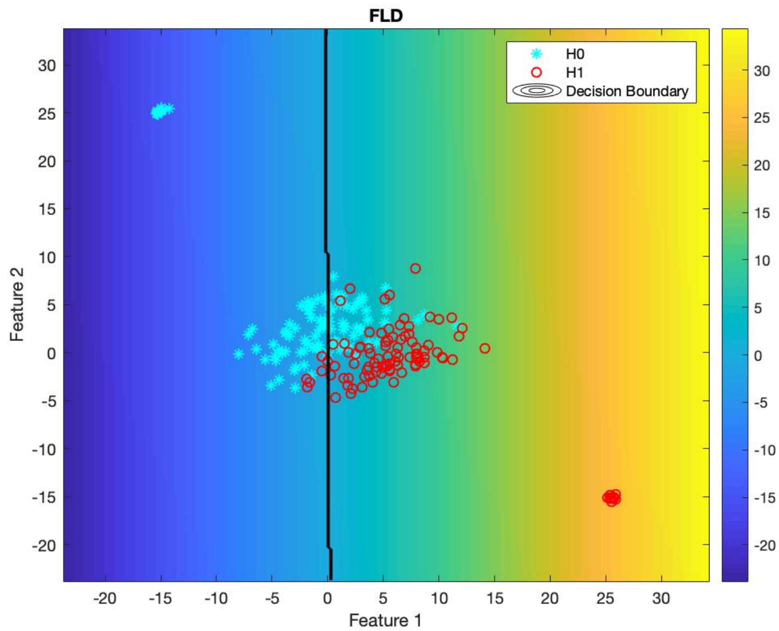


(e) How do the classifier decision boundaries compare to the decision boundary you sketched as a result of visual inspection of data set 3? Explain why the boundaries produced by these three classifiers are similar, or different from, the decision boundary you sketched.

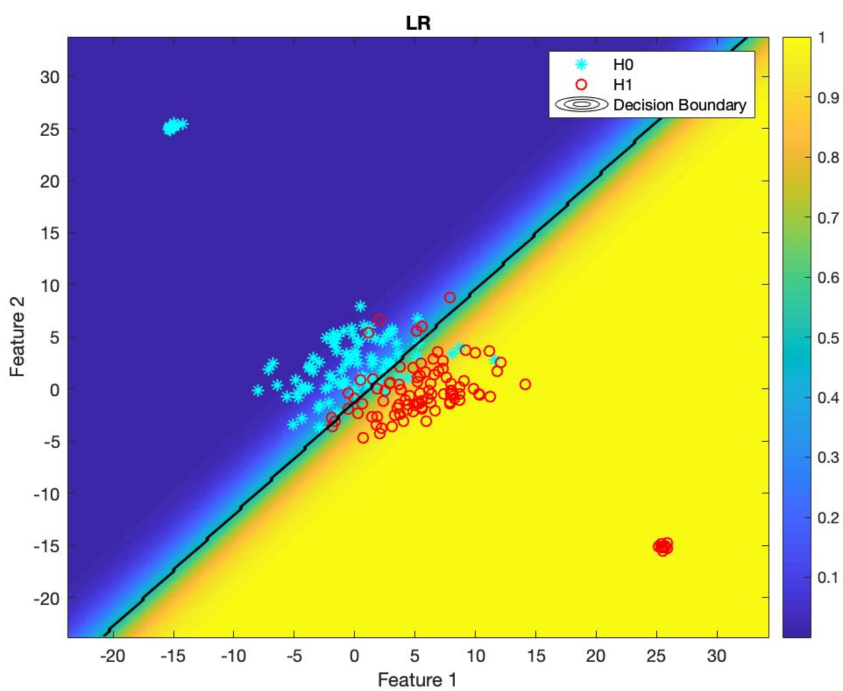
The assumption behind FLD is that the data is Gaussian, however here the data is log-normal.
Therefore, the FLD classifier performs worse in this case than the Gaussian data case, however it doesn't perform horribly.
The LR classifier performs well because it does not rely on the underlying data distribution assumptions.
The Bayes classifier does have the underlying Gaussian assumption, however because it has more flexibility to be nonlinear, it still performs adequately well in separating the classes.

(60) 5. Data set 4 is not consistent with the assumptions underlying LDA – the data is not Gaussian, but rather has outliers.

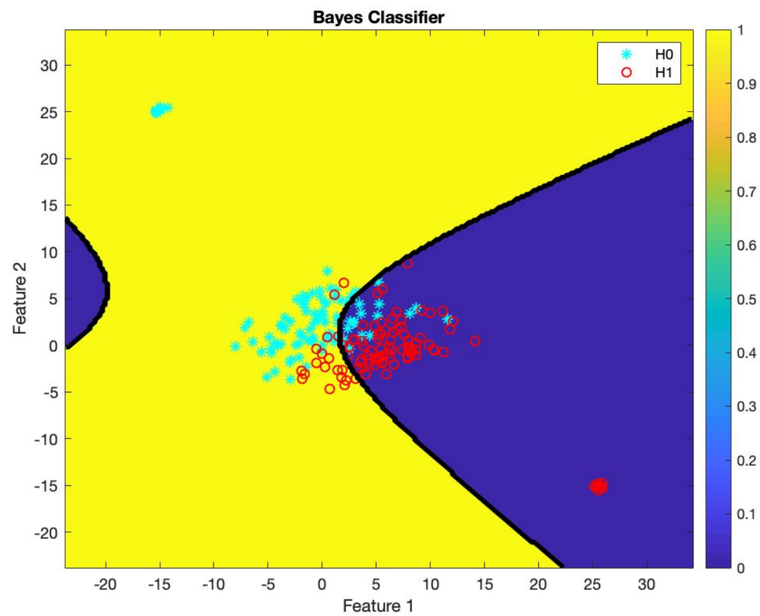
(a) Apply the linear discriminant to dataset 4, and plot the decision statistic surface with both the training data and the decision boundary corresponding to $\lambda(x) = 0$ superimposed on top.



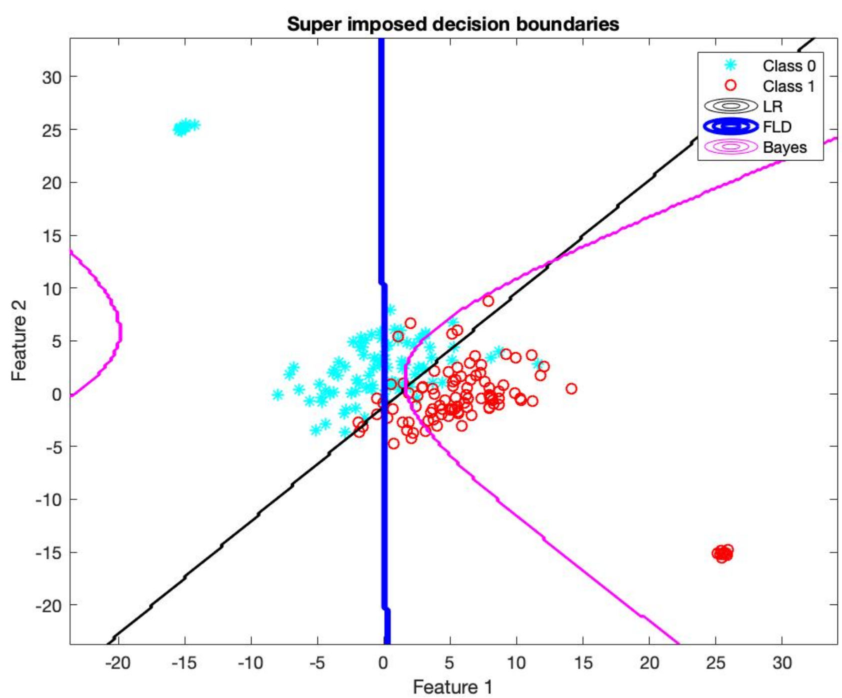
(b) Apply the logistic discriminant to dataset 4, and plot the decision statistic surface with both the training data and the decision boundary corresponding to $\lambda(x) = 0.5$ superimposed on top.



(c) Apply a Bayes Classifier to dataset 4, assuming the features may be dependent and the covariance matrices for the two classes are distinct (*i.e.*, estimate full covariance matrices for both class 0 and class 1), and plot the decision statistic surface for the ln-likelihood ratio with both the training data and the decision boundary under the assumptions of equal class priors and symmetric costs ($\ln \lambda(x) = 0$) superimposed on top.



(d) Compare the three decision boundaries (linear discriminant, logistic discriminant, and Bayes) by visualizing the data (replicating the figure provided for dataset 4 at the beginning of this section), and superimposing all three decision boundaries on top of the data.



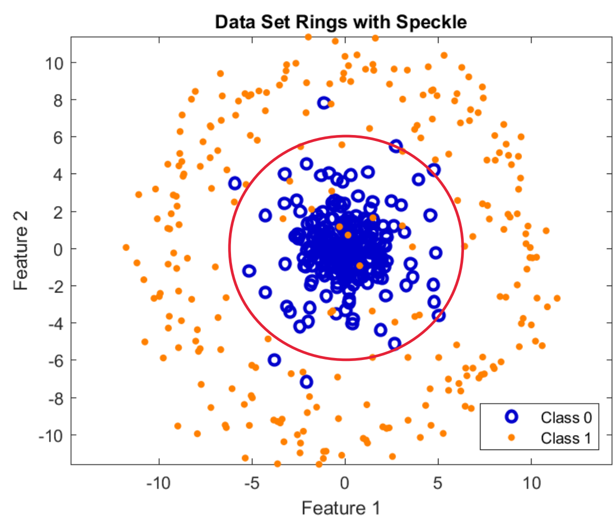
(e) How do the classifier decision boundaries compare to the decision boundary you sketched as a result of visual inspection of data set 4? Explain why the boundaries produced by these three classifiers are similar, or different from, the decision boundary you sketched.

LR works the best here because it does not assume the data is Gaussian. In the case of Bayes and FLD, they do rely on the Gaussian data assumption and therefore outliers greatly skew the covariance matrices, producing poor performance. Clearly, LR works much better in the presence of outliers, due to not having an underlying Gaussian assumption.

Exploring Discriminative Adaptive Nearest Neighbors (DANN)

The Discriminative Adaptive Nearest Neighbors (DANN) classifier[1] is a variant of nearest neighbor classification in which the local neighborhood from which the K nearest neighbors are taken is permitted to adapt (scale and rotate) to reflect the data distribution within the local neighborhood surrouding the test point rather than remain spherical in which all dimensions are considered equally.

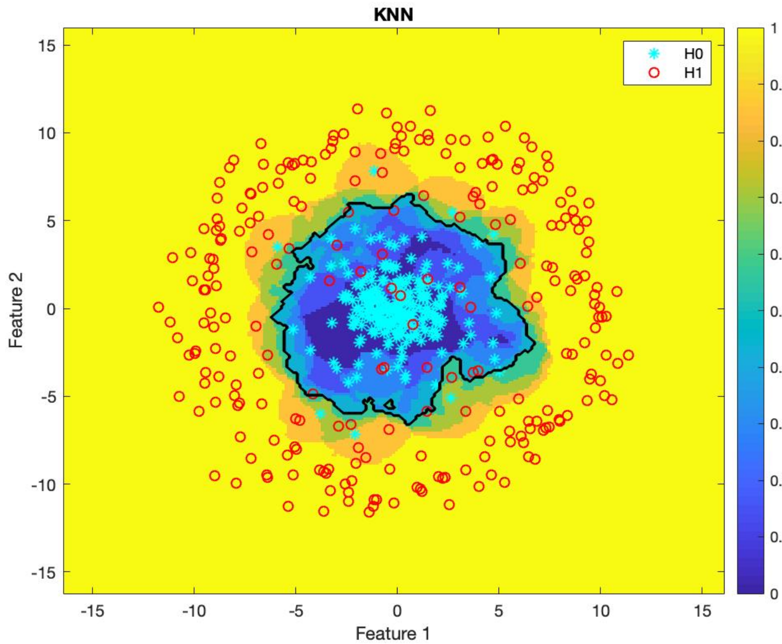
The following questions concerns the `dataSetRingsWithSpeckle.csv` data set provided as a `csv` file. This csv file is organized such that each row contains the true class (either 0 or 1) followed by the associated (2-dimensional) feature vector. When you visualize the data set, you should see this:



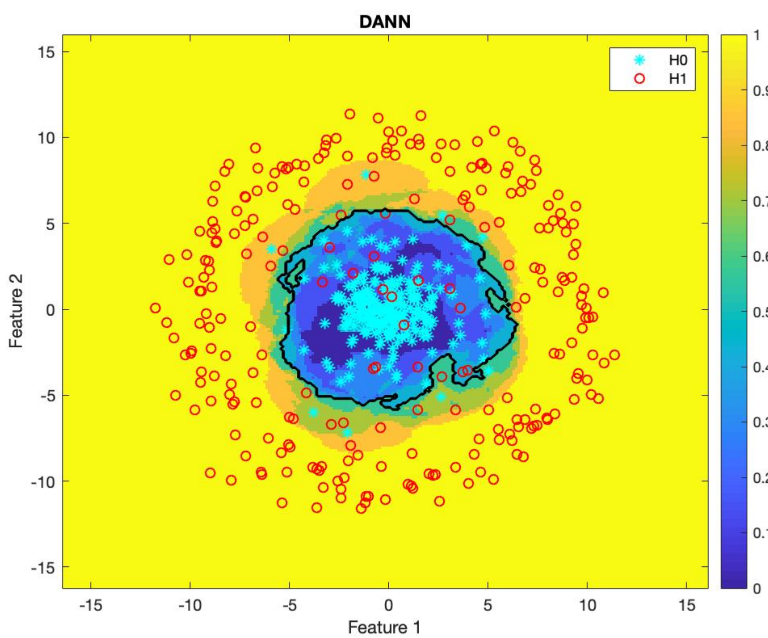
(10) 6. From visual inspection of this dataset , qualitatively sketch on the provided figure what you would consider to be a "good" decision boundary (assuming the goal is maxP_{cd} (or minP_e), taking into consideration the bias-variance trade-off (*i.e.*, sketch a boundary that is no more complicated than it needs to be).

[1] See Hastier, *et al.*, p 477.

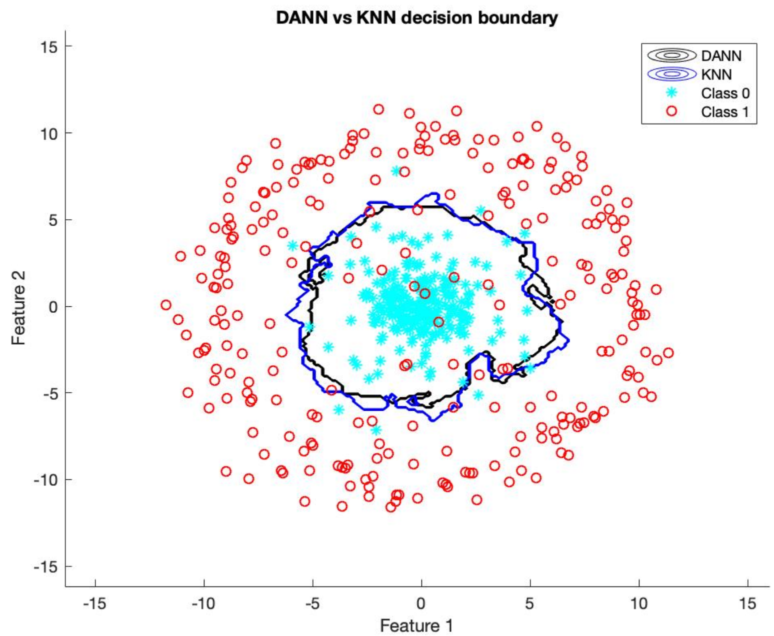
(70) 7. (a) Apply a KNN classifier with $K = 7$ to this data, and plot the decision statistic surface with both the training data and the decision boundary corresponding to $\lambda(x) = 0.5$ superimposed on top.



(b) Apply DANN classifier with $K = 7$ and $N = 100$ (N is the number of neighbors used to find the local discriminant) to this data, and plot the decision statistic surface with both the training data and the decision boundary corresponding to $\lambda(x) = 0.5$ superimposed on top.



(c) Compare both decision boundaries (KNN and DANN) by visualizing the data (replicating the figure provided the beginning of this section), and superimposing both decision boundaries on top of the data.



(d) How do the classifier decision boundaries compare to the decision boundary you sketched as a result of visual inspection of the data?

The decision boundaries pictured above are more specific to the data. My decision boundary was a simple circle without the added complexity that the DANN and KNN produced above. Other than that, the general circled region is fairly similar. DANN and KNN produce very similar decision boundaries to one another.

(e) What are your impressions as to when DANN may be able to provide advantages over KNN?

For certain data distributions, DANN can be beneficial over KNN. KNN assumes a circular shape when identifying the K nearest neighbors. Sometimes, the data distribution may give us helpful insight into assuming a different shape such that the classification produces better results.

(f) What disadvantages, if any, do you see in DANN compared to KNN?

Higher computational complexity, especially in the case when the DANN ends up providing the same result as KNN anyway (in the case that sigma is equal to the identity matrix).