COM S 474/574 Spring 2020
Introduction to Machine Learning

Homework 3

1 Directions:

- **Due: Thursday March 12, 2020 at 10pm.** Late submissions will be accepted for 24 hours after that time, with a 15% penalty.
- Upload the homework to Canvas as a pdf file. Answers to problems 1-4 can be handwritten, but writing must be neat and the scan should be high-quality image. Other responses should be typed or computer generated.
- Any non-administrative questions must be asked in office hours or (if a brief response is sufficient) Piazza.

2 Problems

Problem 1. [5 points] Ch. 4 #7 "Suppose that we wish ..."

Suppose that we wish to predict whether a given stock will issue a dividend this year ("Yes" or "No") based on X, last year's percent profit. We examine a large number of companies and discover that the mean value of X for companies that issued a dividend was ¯ X = 10, while the mean for those that didn't was ¯ X = 0. In addition, the variance of X for these two sets of companies was ˆ σ2 = 36. Finally, 80% of companies issued dividends. Assuming that X follows a normal distribution, predict the probability that a company will issue a dividend this year given that its percentage profit was X = 4 last year.

(0.8*(1/sqrt(2*pi*6))*e*(-(1/72)*(4-10)^2))/((0.8*(1/sqrt(2*pi*6))*e*(-(1/72)*(4-10)^2))+ (0.2*(1/sqrt(2*pi*6))*e*(-(1/72)*(4-0)^2))) = 0.9

Problem 2. [20 points] Suppose you are predicting a feature y that can take on three values $y \in \{+1,+2,+3\}$ and you can predict y using two features x_1 and x_2. You decide to try LDA. Suppose that the covariance matrices for x_1 and x_2 are the identity matrix,

$$\Sigma_{y=+1} = \Sigma_{y=+2} = \Sigma_{y=+3} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

the priors are equal

$$\pi_{y=+1} = \pi_{y=+2} = \pi_{y=+3} = \frac{1}{3},$$

and the mean vectors are

$$\mu_{y=+1} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \qquad \mu_{y=+2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \mu_{y=+3} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

(a). Find the equation for the LDA boundary between $y = +1$ and $y = +2$.

$-2X_1-2X_2=0$

$X_2=-X_1$

(b). Find the equation for the LDA boundary between $y = +1$ and $y = +3$.

$0X_1-2X_2=0$

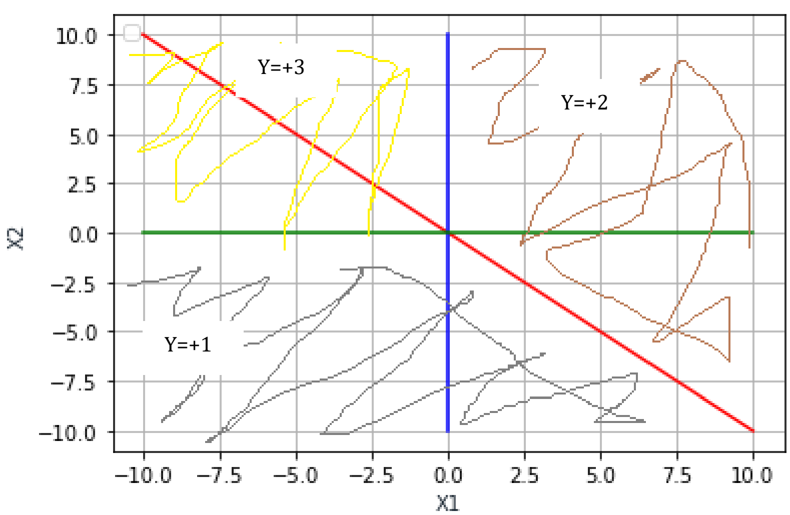
$X_2=0$

(c). Find the equation for the LDA boundary between $y = +2$ and $y = +3$.

$2X_1+0X_2=0$

$X_1=0$

(d). Make a plot with x_1 along horizontal axis, x_2 along vertical axis and draw each of the LDA boundaries you found. For each region, write the class label (e.g. "+2") that would be chosen for a new sample that appeared in that region.



(e). How would your result differ if you used QDA? (keeping all the parameters above the same)

The QDA decision boundaries would be quadratic instead of liner like the Bayes decision boundary and LDA decision boundary. So the QDA decision boundary is inferior, because it would suffer from higher variance.

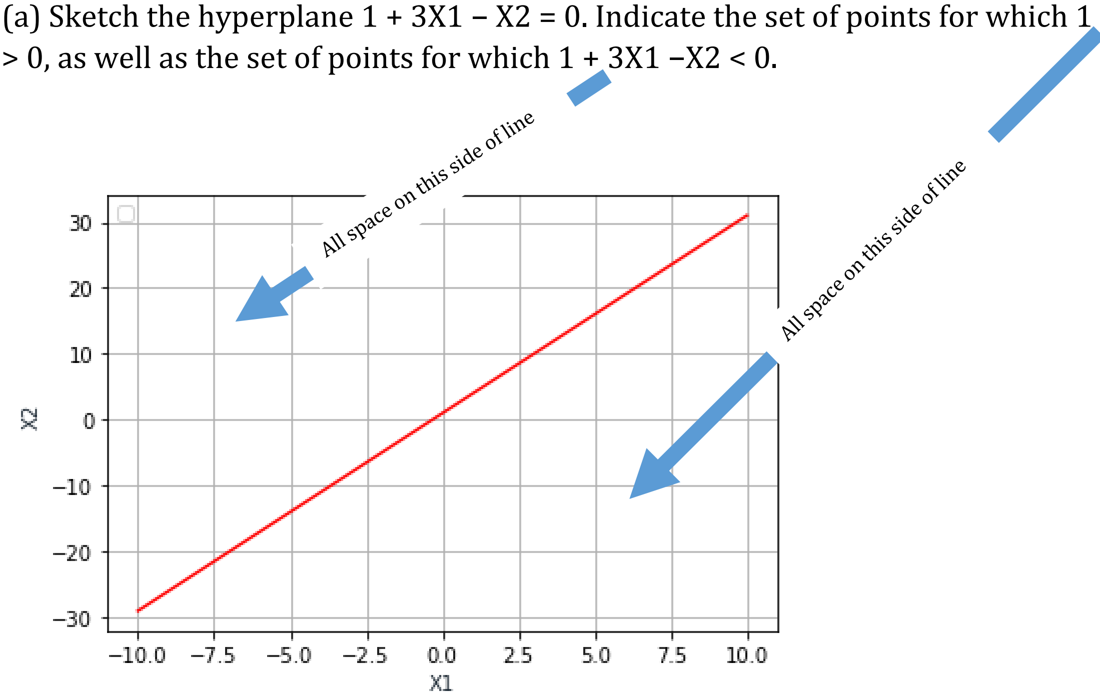
(f). In class, we separately discussed LDA and naive Bayes assumptions. If you wanted to use both for the above problem, explain what would change (if anything).

The testing error for the LDA boundary would be slightly higher than the error for the bayes boundary. Also, the lines would not be the same. The LDA method approximates the Bayes classifier by plugging in estimates for πk, μk, andσ2 unlike in the naïve bay method.

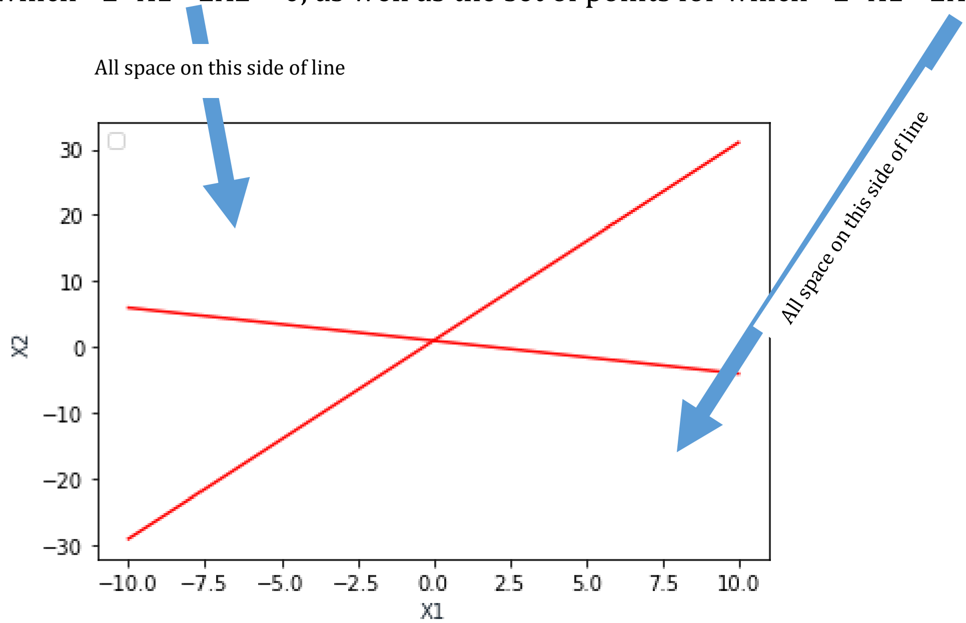
Problem 3. [10 points] Ch. 9 #1 "This problem involves hyperplanes ..."

1. This problem involves hyperplanes in two dimensions.

(a) Sketch the hyperplane 1 + 3X1 − X2 = 0. Indicate the set of points for which 1 + 3X1 −X2 > 0, as well as the set of points for which 1 + 3X1 −X2 < 0.



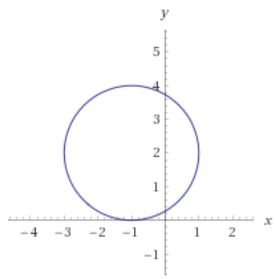
(b) On the same plot, sketch the hyperplane −2+X1 +2X2 = 0. Indicate the set of points for which −2+X1 +2X2 > 0, as well as the set of points for which −2+X1 +2X2 < 0.



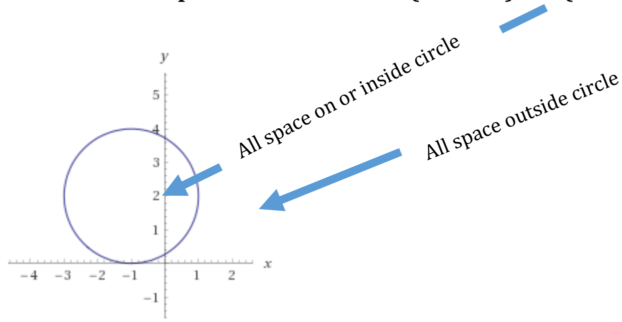
Problem 4. [15 points] Ch. 9 #2 “We have seen that in ...”

We have seen that in p = 2 dimensions, a linear decision boundary takes the form β0+β1X1+β2X2 = 0. We now investigate a non-linear decision boundary.

(a) Sketch the curve: $(1 + X1)^2 + (2-X2)^2 = 4$



(b) On your sketch, indicate the set of points for which $(1 + X1)^2 + (2−X2)^2 > 4$, as well as the set of points for which $(1 + X1)^2 + (2−X2)^2 \le 4$.



(c) Suppose that a classifier assigns an observation to the blue class if (1 + X1)2 + (2−X2)2 > 4, and to the red class otherwise. To what class is the observation
(0,0): Blue
(−1,1): Red
(2,2): Blue
(3,8): Blue

(d) Argue that while the decision boundary in (c) is not linear in terms of X1 and X2, it is linear in terms of X1, X2 1, X2, and X2 2.

Notes for problem 5:

- This problem will use a 2d mesh to make plots. A Jupyter notebook HW3.ipynb is included on canvas which includes code for some of the plots. You can copy that code for your submission. An html is also included so you can copy the code from a browser.

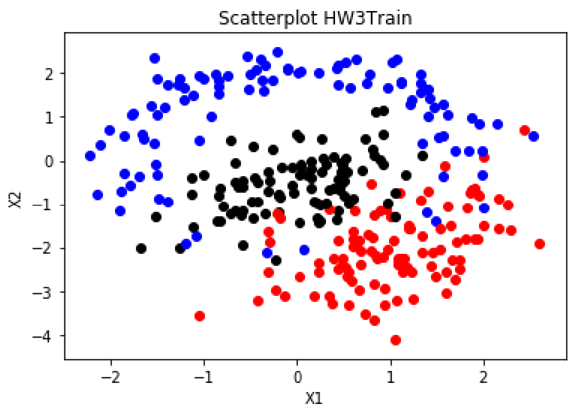
If you are using another language (Matlab, R, Java, etc.) and are not sure how to make similar plots, contact the teaching staff.

- Some of the fitting or predicting may run slow on old computers. If you find that running any steps takes more than several seconds, contact the teaching staff. We may suggest increasing the mesh step size (so fewer points are used) if prediction is slow or setting max iter to a lower value if fitting SVMs is slow. • When fitting SVMs, you may get convergence warnings for some input parameter values. This is expected. The .ipynb on canvas includes code to suppress those warnings.

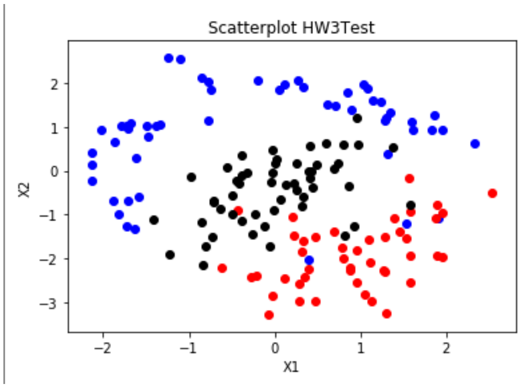
Problem 5. [50 points]

A. Download the data sets HW3train.csv and HW3test.csv from Canvas. In both files, the first column is the feature *y* we want to predict. *y* can take on three values. The next two columns are the continuous-valued features x_1 and x_2.

(1). Make a scatter-plot of the data-set HW3train with x_1 values on the horizontal axis and x_2 values on the vertical axis.



(2). Make another scatter-plot for HW3test data.

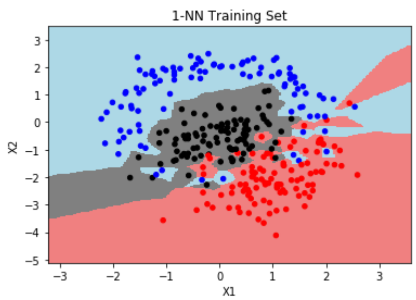


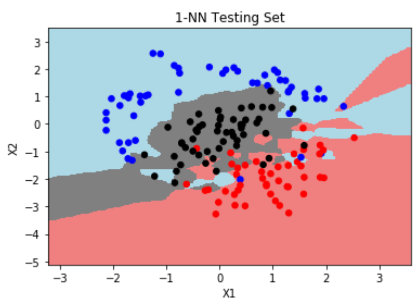
B. K-nn:

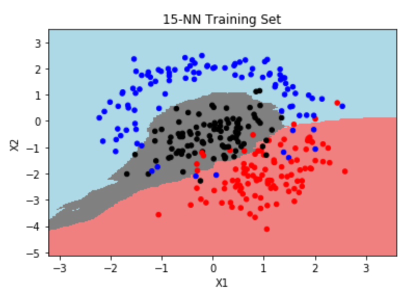
(1). For $k \in \{1,5,15\}$, make plots of the decision regions (e.g. what a new sample would be predicted) with the training points, similar to the $k = 3$ example in the posted HW3.ipynb on canvas.

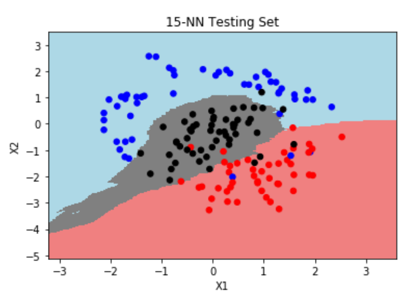
In Python, you can call

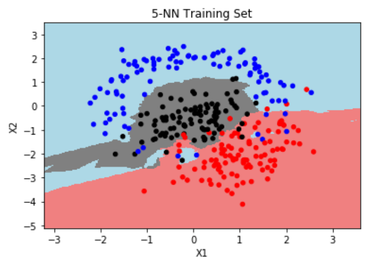
```
clf=KNeighborsClassifier(k,weights='uniform')
```

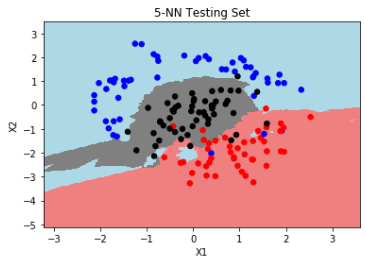








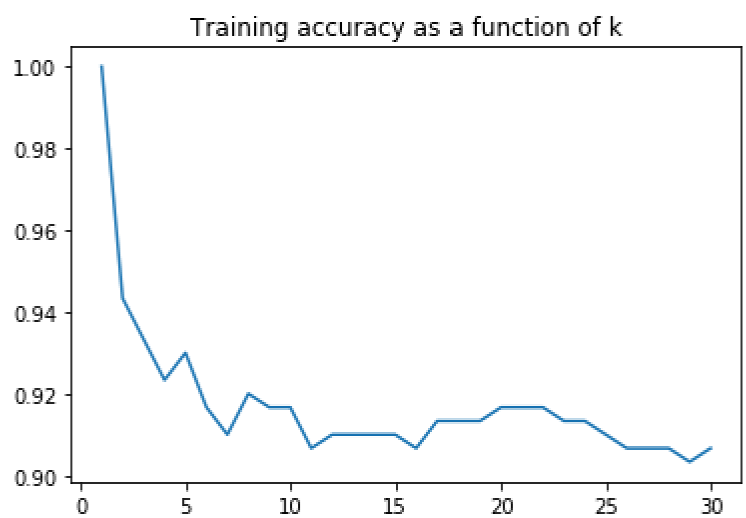




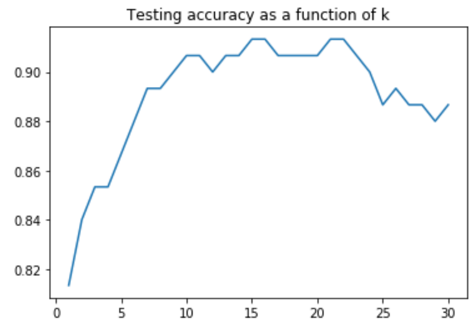
(2). Briefly comment (2-3 sentences) on the similarities and differences of the regions in those plots.

The regions in all plots classify most of the point's accurately. As K-NN increases, the testing data gets classified better. The regions change shape as K-NN increases.

(3). Make a plot of training accuracy as a function of k, for $k \in \{1,2,3,...,30\}$.



(4). Make a plot of testing accuracy as a function of k, for $k \in \{1,2,3,...,30\}$.



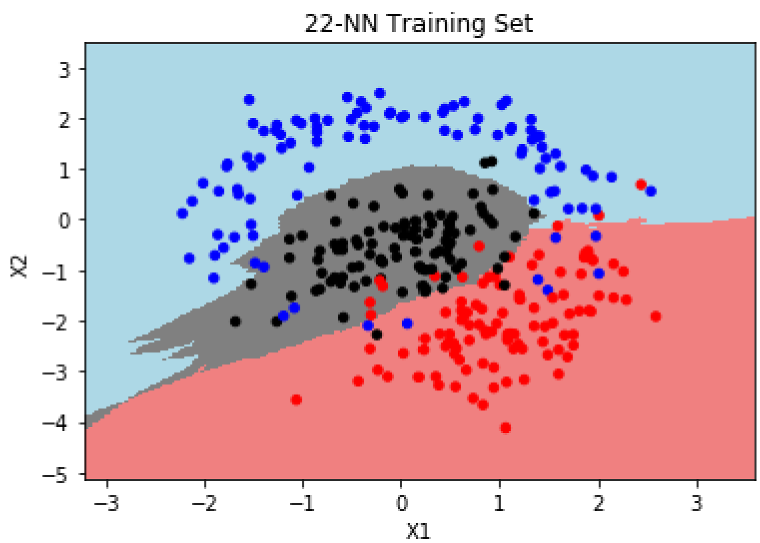
(5). Report the best *k*, along with its associated training and testing accuracies.

K=22

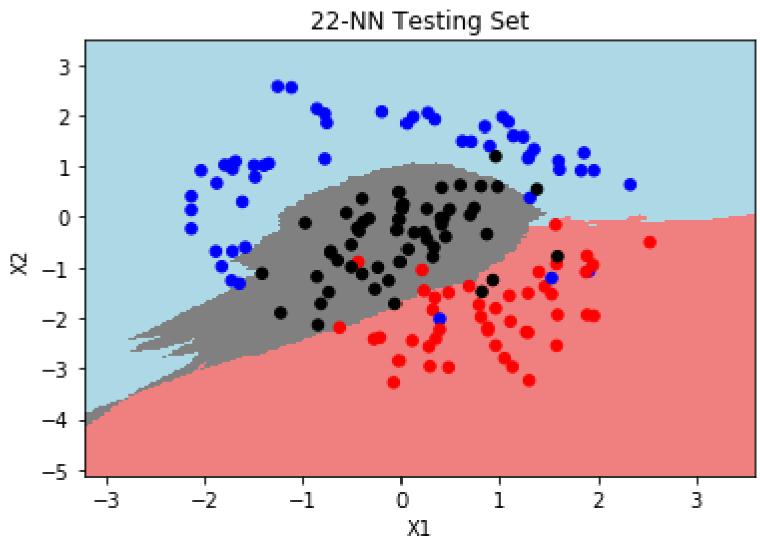
Testing accuracy: 0.9133333333333333

Training accuracy: 0.9166666666666666

(6). For that best *k*, make a plot with both the decision regions and the training points.



(7). For that best *k*, make a plot with both the decision regions and the testing points.



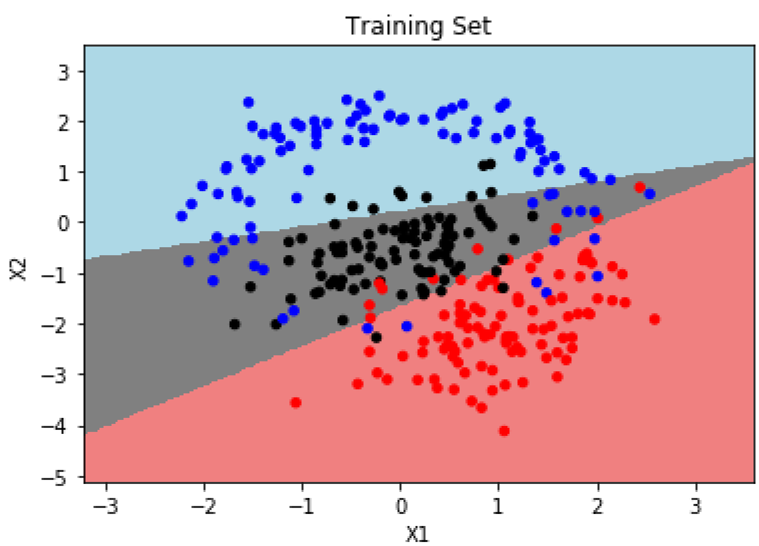
C. LDA:

(1). Report the training and testing accuracies for LDA.

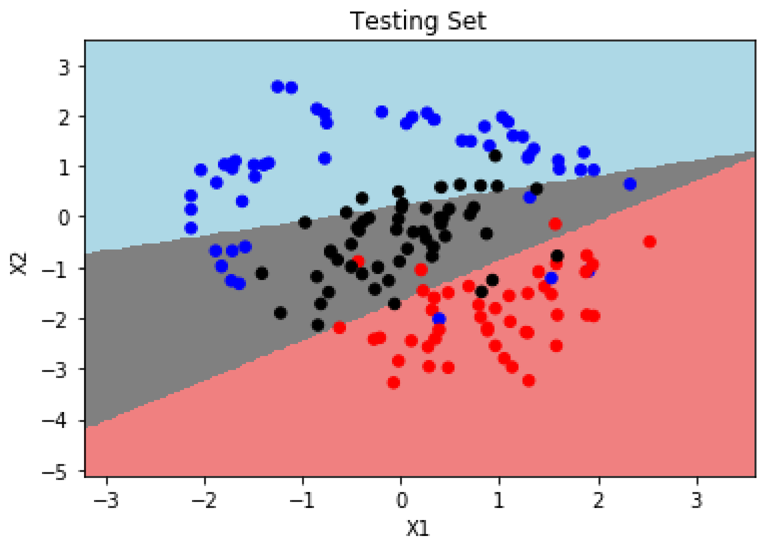
In Python, you can call

clf=LinearDiscriminantAnalysis(solver='svd',shrinkage=None,priors= None)

(2). Make a plot with both the decision regions and the training points.



(3). Make a plot with both the decision regions and the testing points.



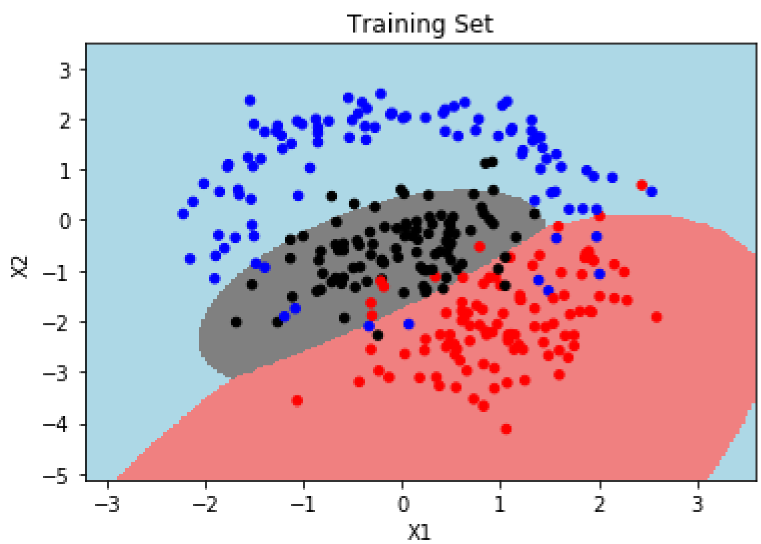
D. QDA:

(1). Report the training and testing accuracies for QDA.

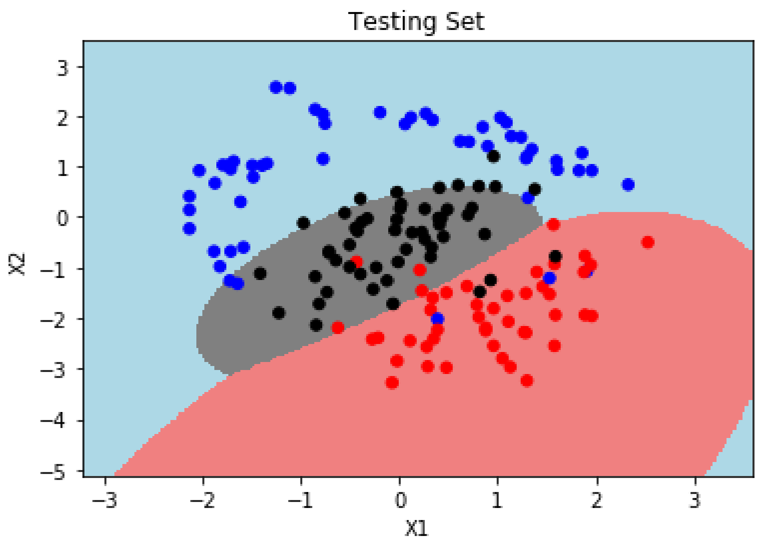
In Python, you can call

clf=QuadraticDiscriminantAnalysis(priors=None,reg_param=0.0)

(2). Make a plot with both the decision regions and the training points.



(3). Make a plot with both the decision regions and the testing points.



E. In about 3-5 sentences, discuss some of the differences between the decision regions you plotted for for kNN, LDA, and QDA. Comments should include their shapes, how smooth the boundaries are, whether the region (for any given y value) are connected or disjoint. LDA has liner boundaries while QDA and kNN do not. The QDA boundaries are smooth while kNN is not smooth. QDA has the smallest regions.

F. SVM with polynomial kernel: For degrees $p \in \{1,2,3,4\}$, fit SVMs with a polynomial kernel of degree p by searching over what input parameters give the best performance. Specifically,

- Create an array

  ```
  Cvals=np.logspace(-4,2,25,base=10)
  ```

  that contain different C parameters for how much error will be tolerated compared to minimizing the boundary coefficients.

- Set gamma=1.0
- Set coef0=1.0
- Set max_iter=1000 This will cause the solver to terminate early when for some C values that do not work well.
- Call

  ```
  clf=SVC(C=c,kernel='poly',degree=degree,gamma=gamma,coef0=1.0,
          shrinking=True,probability=False,max_iter=max_iter)
  ```

  for each of the C values, keeping track of which is best for that degree p.

For each degree $p \in \{1,2,3,4\}$,

(1). Report the best C value

(2). Report the training and testing accuracies using that C value.

(3). Make a plot of its associated decision regions with the training points.

(4). Make a plot of its associated decision regions with the testing points.

G. SVM with the radial basis function (RBF) kernel: Find the best parameters C and kernel width parameter gamma for the RBF kernel. Specifically,

- Use the same Cvals array as for the polynomial kernel
- Create an array

  gamma_vals=np.logspace(-2,2,25,base=10)

  that contain different kernel width parameters.
- Set max_iter=1000 This will cause the solver to terminate early when for some gamma values that do not work well.
- Call

  clf=SVC(C=c,kernel='rbf',gamma=gamma,shrinking=True,probability=False,max_iter=max_iter)

  for each of the gamma and C values, keeping track of which pair is best.

(1). Report the best pair of gamma and the C values.

(2). Report the training and testing accuracies using that pair.

(3). Make a plot of its associated decision regions with the training points.

(4). Make a plot of its associated decision regions with the testing points.

H. In about 4-6 sentences, comment on the performance and decision regions of SVM using the different polynomial kernels and the RBF kernel. Also compare/contrast the regions to those you saw for kNN, LDA, and QDA.

SVM has non liner boundaries and performs differently based on type type of kernel. The SVM has similar boundaries to QDA and kNN in that they are non-liner as in the LDA case. QDA had an testing accuracy of 0.87 LDA had 0.81 and kNN had 0.91. QDA had an training accuracy of 0.90 LDA had 0.83 and kNN had 0.92.

I. Suppose that the data was randomly generated. Based on the results you obtained, which method do you think best captures the shapes of the data you see? In 1-3 sentences, explain why. I believe I would choose kNN. This is because it has the highest testing accuracy and it isn't restricted to being quadratic(QDA) or liner(LDA).