# CLASSIFICATION METHODS

Chapter 04 (part 02)

# LINEAR DISCRIMINANT ANALYSIS (LDA) &

# QUADRATIC DISCRIMINANT ANALYSIS (QDA)

# Outline

- Overview of LDA
- Why not Logistic Regression?
- Estimating Bayes' Classifier
- LDA formulation
- Alternative LDA formulation
- 2-class performance measures
- Overview of QDA
- Comparison between LDA and QDA

# Linear Discriminant Analysis

- Goal:  Classify observations
  - Will a consumer buy a product or not?
  - Will a customer be satisfied or not?
  - Which candidate will a voter vote for?
- LDA Key intuition:
  - Represent each class as a simple distribution with parameters
  - Predict the class of a new observation by which class distribution has the highest probability at that observation's feature values

# Assumptions of LDA

- The observations are an unbiased random sample ($i.i.d.$) of the population
- Each predictor variable is normally distributed
- All classes share common (co)variance parameters

# Why not Logistic Regression?

- Logistic regression parameter values are unstable when the classes are well separated

  - **Work on in-class Problem #1**

- In the case where n is small, and the distribution of predictors $X$ is approximately normal, then LDA is more stable than Logistic Regression

- LDA is also more popular than logistic regression when there are more than two response classes

# Bayes' Classifier

- Bayes' classifier is the golden standard. Unfortunately, it is usually not determinable unless we are using synthetic data from a known distribution
- **Concept check: What is the property associated with data points along the Bayes Classification Boundary?**

- So far, we have *estimated* Bayes classifier with two methods:
  - KNN classifier
  - Logistic Regression

# Estimating Bayes' Classifier

- With Logistic Regression we modeled the probability of $Y$ being from the $k^{\text{th}}$ class as

$$p(X) = \Pr(Y = k \mid X = x) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

- However, Bayes' Theorem states for a $K$-class problem,

$$p_k(X) = \Pr(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

$\pi_k$ : Prior Probability of coming from class $k$

$f_k(x)$ : Unknown density function for $x$ given that $x$ is an observation from class $k$ (we can choose this function depending on our model)

Idea: Model classes using distributions, then use Bayes Theorem to make classification decisions

# Bayes requires estimating $\pi_k$ and $f_k(x)$

- We need to estimate $\pi_k$ and $f_k(x)$ to compute $p(x)$

$$p_k(X) = \Pr(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

- The most common model for $f_k(x)$ is the Normal Density

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left( -\frac{1}{2\sigma_k^2}(x-\mu_k)^2 \right)$$

- Using the normal density, we only need to estimate three parameters to compute $p(x)$:

$\mu_k$ $\sigma_k^2$ $\pi_k$

# Use Training Data set for Estimation

- The mean $\hat{\mu}_k$ could be estimated by the feature-wise average of all training observations from the $k^{\text{th}}$ class.
- The variance $\hat{\sigma}$ could be estimated as the weighted average of variances of all $K$ classes. In LDA we make the assumption that the variances for each class are equal. $\sigma_k^2 = \sigma^2$
- Estimate, $\hat{\pi}_k$ as the proportion of the training observations that belong to the $k^{\text{th}}$ class.

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i \qquad \hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

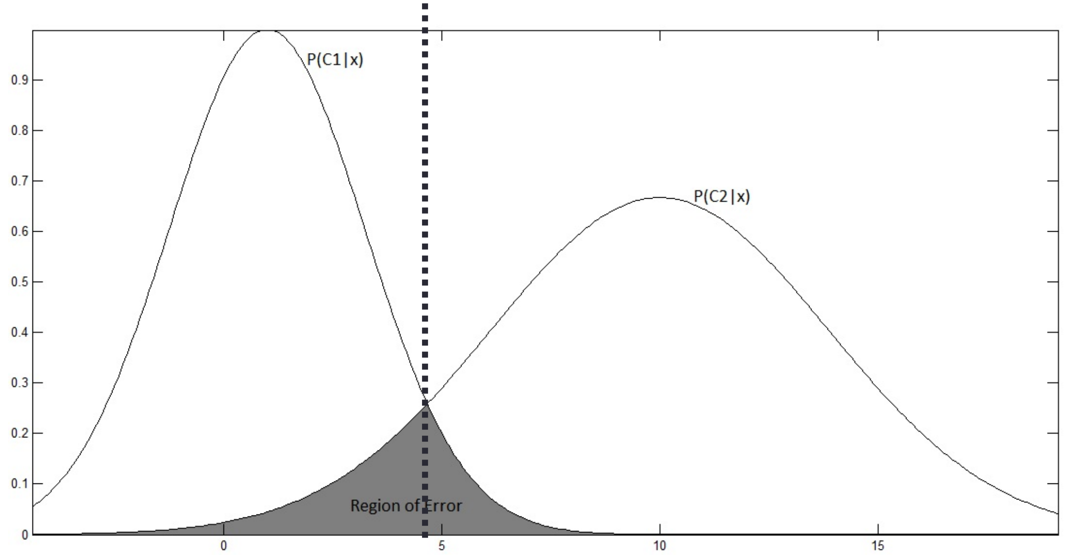
$$\hat{\pi}_k = \frac{n_k}{n}$$

# Bayes Example with One Predictor (p =1)

- Suppose we have only one predictor ($p$ = 1)
- Two normal density function $f_1(x)$ and $f_2(x)$, represent two distinct classes
- The two density functions overlap, so there is some uncertainty about the class to which an observation with an unknown class belongs
- The dashed vertical line represents Bayes' decision boundary

# More complex example with one predictor (2-class, single feature)

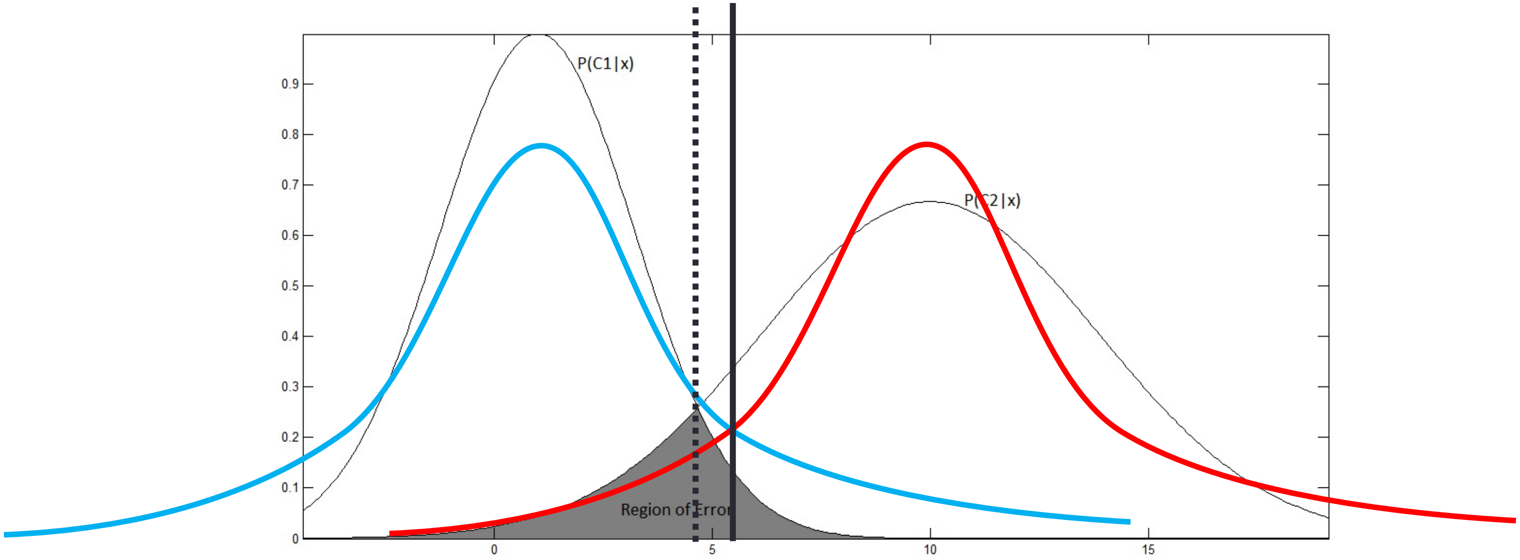
- A discriminator is established at the point of equal probability…
- With a true Bayes classifier, this discriminator is not necessarily exactly between the class means



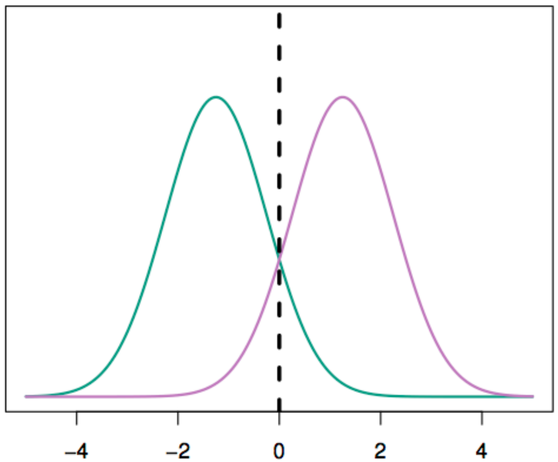

https://ranabasheer.wordpress.com/tag/bayes/

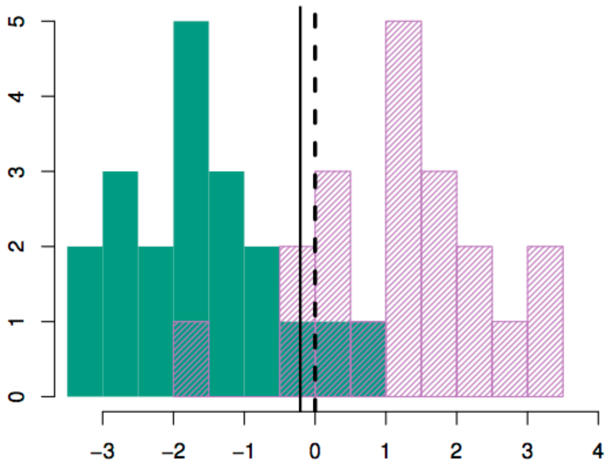
# LDA intuition (2-class, single feature)

- LDA assumes that the observations in each class are Gaussian with the *same variance* but *different* means
- Model each class using
  - sample mean
  - (average) sample variance of each class
- Bayes classifier & LDA not necessarily equal



- Differences between Bayes and LDA performance are also due to sampling issues which are used to estimate class means and variances
  - 20 observations were drawn from each of the two classes
  - The dashed vertical line is the Bayes' decision boundary
  - The solid vertical line is the LDA decision boundary
    - Bayes' error rate: 10.6%
    - LDA error rate: 11.1%



# Apply LDA

- LDA assumes that each class has a normal distribution with one mean per class but the same variance for every class $\hat{\mu}_k$ $\hat{\sigma}$
- The key variables are estimated from the training data

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i: y_i = k} x_i \qquad \hat{\pi}_k = \frac{n_k}{n} \qquad \hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i: y_i = k} (x_i - \hat{\mu}_k)^2$$

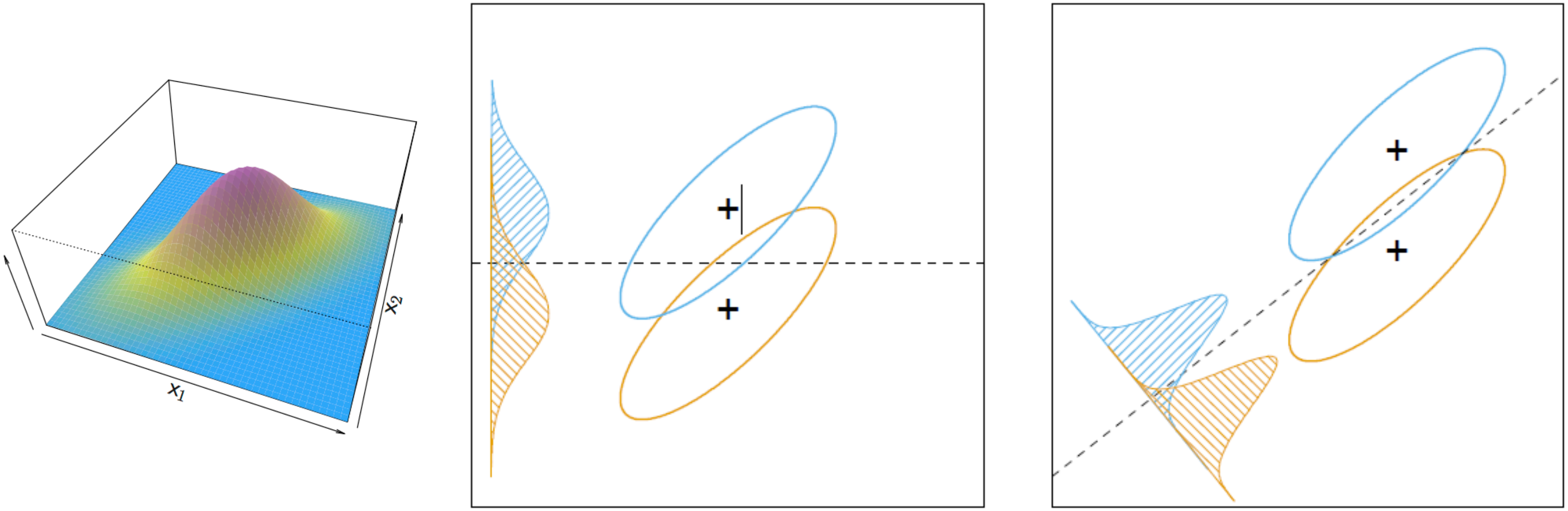
$$f_k(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left( -\frac{1}{2\sigma^2} (x - \mu_k)^2 \right)$$

- Bayes' theorem is used to compute $p_k$ and the observation is assigned to the class with the maximum probability among all $K$ probabilities

$$p_k(X) = \Pr(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

# LDA intuition (more than 1 feature)

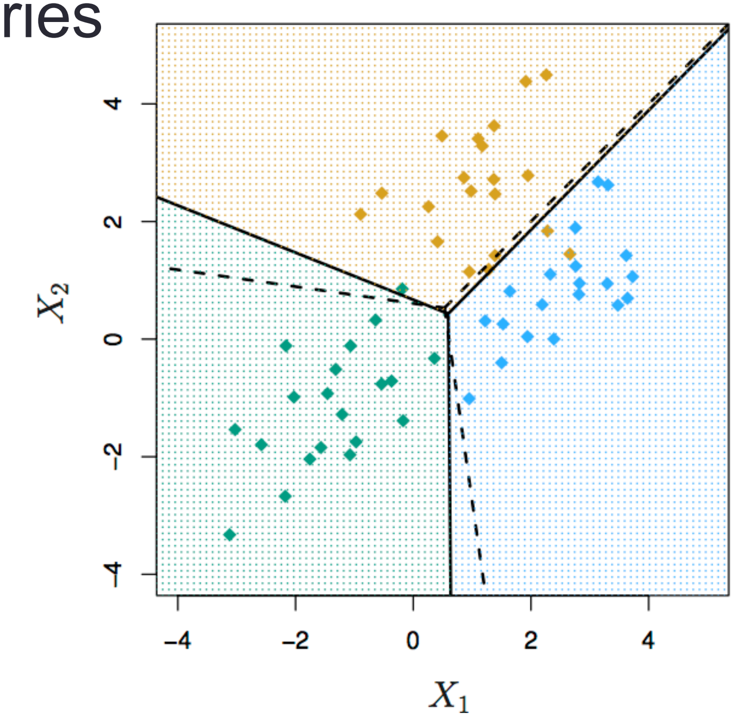
- If X is multidimensional ($p > 1$), we use exactly the same approach except the density function $f(x)$ is modeled using the multivariate normal density
- Need to find the direction for which a projection into fewer dimensions yields the most information for discrimination of the LDA (Bayes-like) classifier using Covariance




Elements of Statistical Learning – Figure 4.9

# Multiclass LDA

- Three classes & Two predictors (p =2)
- 20 observations were generated from each class
- The dashed lines are Bayes' boundaries
- The solid lines are LDA boundaries



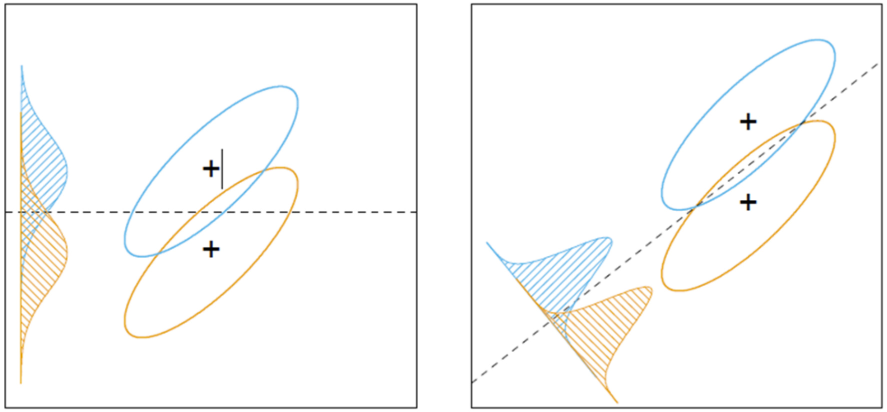
# Alternative LDA formulation

- LDA involves the determination of linear equation (just like linear regression) that will predict which **class** the case belongs to.

$$D = w_0 + w_1 X_1 + w_2 X_2 + \ldots + w_i X_i$$

  - $D$: discriminant hyperplane
  - $w$: discriminant coefficients
  - $X$: variable
  - $w_0$: constant (default = 0)

# Alternative LDA formulation



- Goal: discriminate between the different categories
- Choose the $w$'s in a way to
  <u>*maximize the distance between the means*</u>
  of different categories
- Features which help classify observations tend to have large $w$'s (weight)

$$D = w_0 + w_1 X_1 + w_2 X_2 + \ldots + w_i X_i$$

# Alternative LDA computation (2 class, 1 feature)

- Select the discriminator line D such that

$$D = w_1 X_1 + w_0$$

where

$$w_1 = \frac{\mu_{c1} - \mu_{c0}}{\sigma}$$

$w_0$ is default 0, or selectable to maximize traning performance

Thus, select class $\{0,1\}$ according to: $w_1 X_1 > w_0$

# Alternative LDA computation (2 class, multi-feature)

- Select the discriminator line D such that

$$D = w^T X + w_0$$

where

$$w = \frac{\mu_{c1} - \mu_{c0}}{\Sigma^{-1}}$$

$\Sigma^{-1}$ is the inverse (shared) covariance matrix of the classes

$w_0$ is default 0, or selectable to maximize traning performance

Thus, select class $\{0,1\}$ according to: $w^T X > w_0$

# Alternative LDA in practice

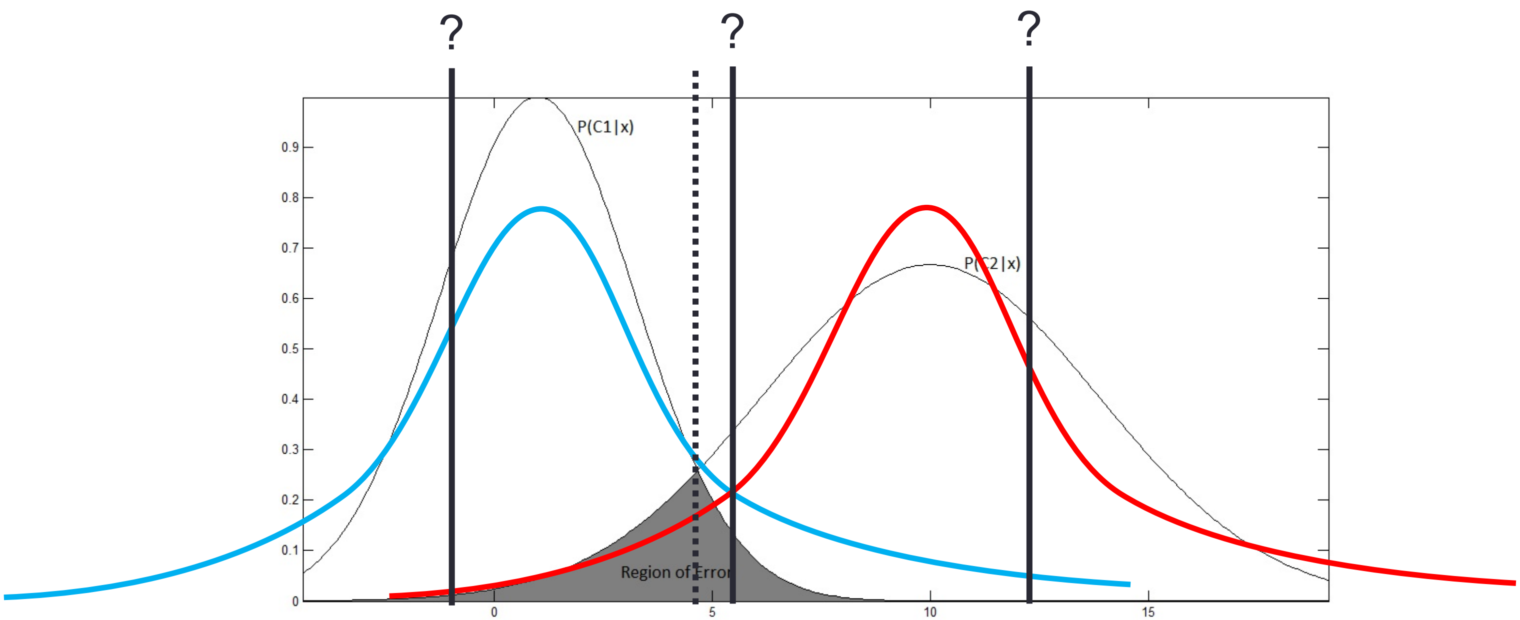
- In practice LDA is often combined with a feature reduction technique to reduce the effective dimensionality of the space
- When using LDA packages, select a smaller “components” parameter to enact dimensionality reduction
- `sklearn.discriminant_analysis.LinearDiscriminantAnalysis(solver='svd', shrinkage=None, priors=None, n_components=None, store_covariance=False, tol=0.0001)[source]`
  - `n_components : int, optional`
  - `Number of components (< n_classes - 1) for dimensionality reduction.`
- Further details in Elements of Statistical Learning

# 2-class Performance Measures

- Altering the decision boundary
- Confusion Matrix
- ROC

# Altering the decision boundary

- Sometimes the (approximate) Bayes decision boundary may not be adequate for the business case
- **Audience participation: Give an example of this and explain why**



# Classifier performance on Default Data (at $p(y|x)$ > 0.5 as Threshold for Default)

| | | *True Default Status* | | |
|---|---|---|---|---|
| | | No | Yes | Total |
| *Predicted* | No | 9644 | 252 | 9896 |
| *Default Status* | Yes | 23 | 81 | 104 |
| | Total | 9667 | 333 | 10000 |

- LDA makes 252+ 23 = 275 mistakes on 10000 predictions (2.75% misclassification error rate)
- But LDA miss-predicts 252/333 = 75.5% of defaulters
- We shouldn't use 0.5 as threshold for predicting default if this will cost the bank a lot of money

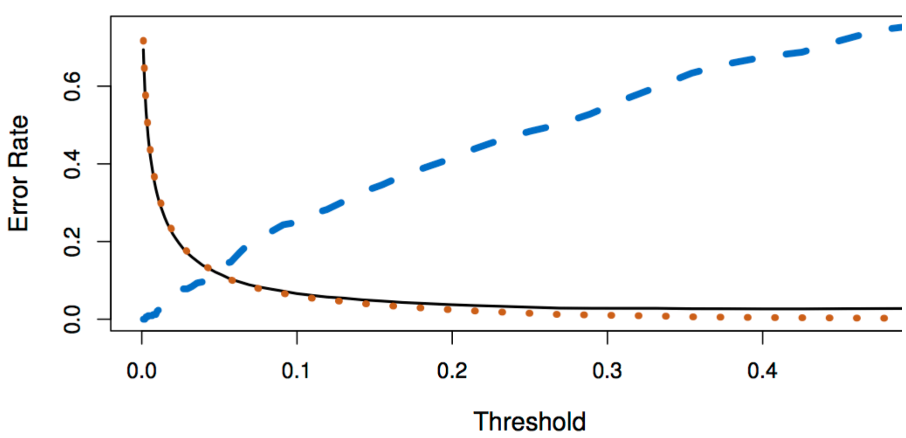
# Use $p(y|x)$ > 0.2 as Threshold for Default?

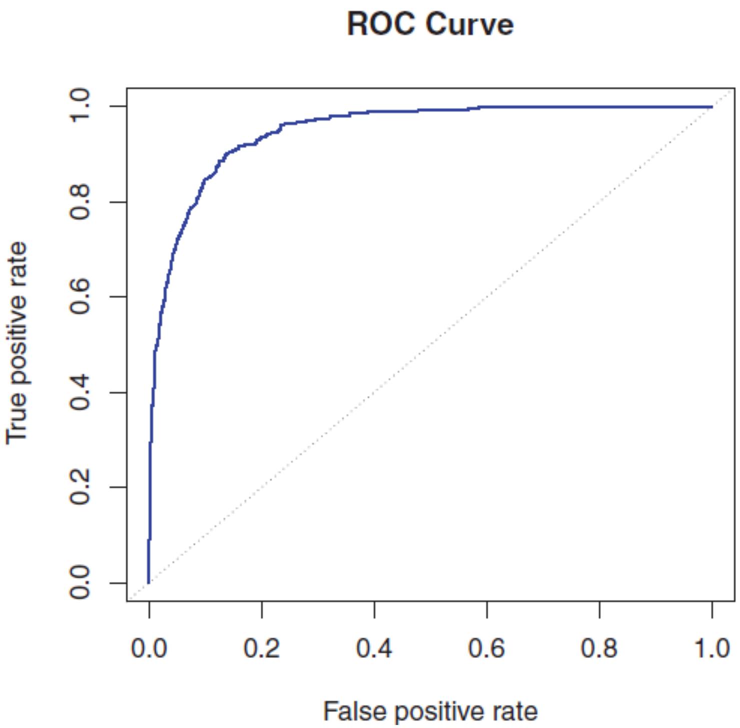
- Now the total number of mistakes is 235+138 = 373 (3.73% misclassification error rate)
- But we only miss-predicted 138/333 = 41.4% of defaulters

| | | *True Default Status* | | |
|---|---|---|---|---|
| | | No | Yes | Total |
| *Predicted* | No | 9432 | 138 | 9570 |
| *Default Status* | Yes | 235 | 195 | 430 |
| | Total | 9667 | 333 | 10000 |

# Threshold Values, Error Rates, ROC

- Black solid: overall error rate
- Blue dashed: Fraction of defaulters missed
- Orange dotted: non defaulters incorrectly classified



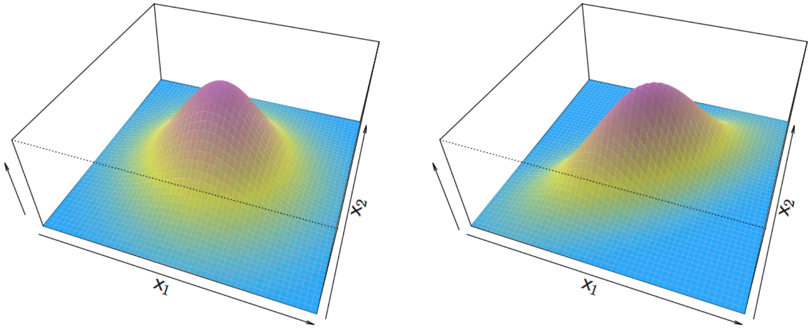



# In Class Work – Classifier Performance and ROC

- Work on problem 2 now

# Quadratic Discriminant Analysis (QDA)

- LDA assumed that every class has the same variance/ covariance
- However, LDA may perform poorly if this assumption is far from true
- QDA works identically as LDA except that it estimates separate variances/ covariance for each class



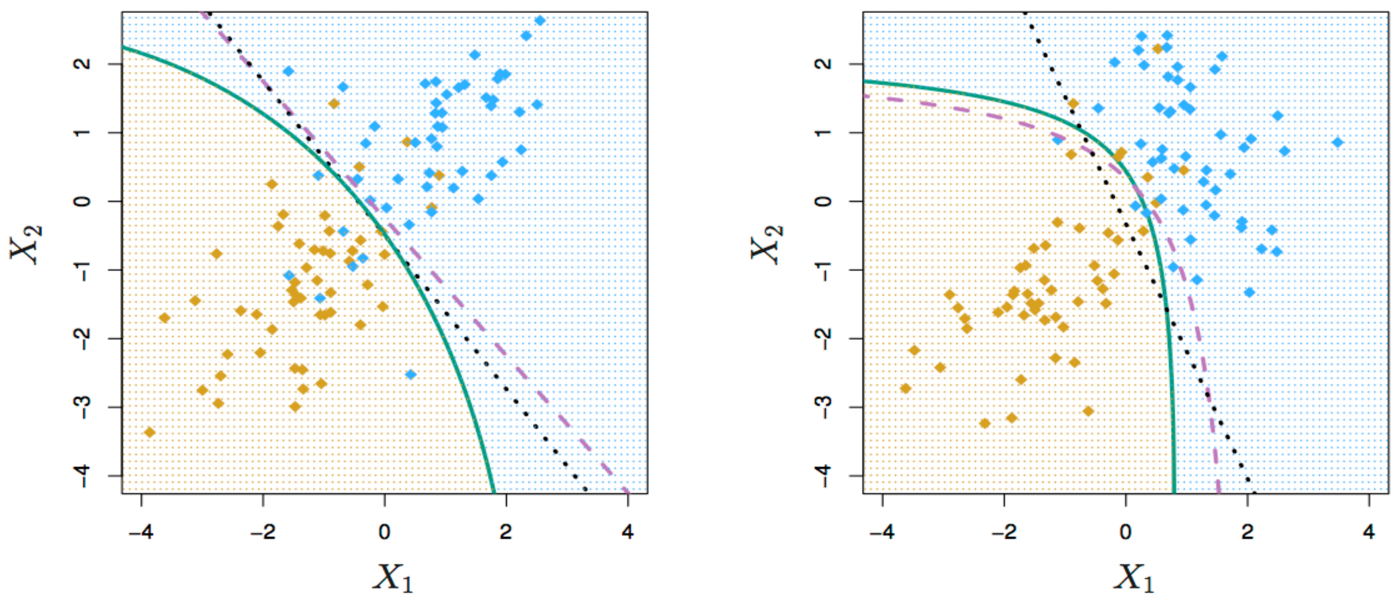
# Which is better? LDA or QDA?

- Since QDA allows for different variances among classes, the resulting boundaries become quadratic

- Which approach is better: LDA or QDA?
  - QDA may work better when the variances are very different between classes and we have enough observations to accurately estimate the variances
  - LDA may work better when the variances are similar among classes or we don't have enough data to accurately estimate the true differences in the variances

# Comparing LDA to QDA

- Black dotted: LDA boundary
- Purple dashed: Bayes' boundary
- Green solid: QDA boundary
- Left: variances of the classes are equal (LDA is better fit)
- Right: variances of the classes are not equal (QDA is better fit)



# Comparison of Classification Methods

- KNN (Chapter 2)
- Logistic Regression (Chapter 4)
- LDA (Chapter 4)
- QDA (Chapter 4)

# Logistic Regression vs. LDA

- Similarity: Both Logistic Regression and LDA produce linear boundaries
- Difference: LDA assumes that the observations are drawn from the normal distribution with common variance in each class, while logistic regression does not have this assumption.
  - LDA would do better than Logistic Regression if the assumption of normality holds,
  - otherwise logistic regression can outperform LDA

# KNN vs. (LDA and Logistic Regression)

- KNN takes a completely different approach
- KNN is completely non-parametric: No assumptions are made about the shape of the decision boundary
- <u>Advantage of KNN</u>: We can expect KNN to dominate both LDA and Logistic Regression when the decision boundary is highly non-linear
- <u>Disadvantage of KNN:</u> KNN does not tell us which predictors are important (no table of coefficients)

# QDA vs. (LDA, Logistic Regression, and KNN)

- QDA is a higher variance parametric model which offers a compromise in performance between non-parametric KNN method and linear methods such as LDA and logistic regression

- If the true decision boundary is:
  - Linear: LDA and Logistic outperforms
  - Moderately Non-linear: QDA outperforms
  - More complicated: KNN is superior