

# Classification

- **Qualitative** variables take values in an **unordered** set  $\mathcal{C}$ , such as:  
 $\text{eye color} \in \{\text{brown}, \text{blue}, \text{green}\}$   
 $\text{email} \in \{\text{spam}, \text{ham}\}.$
- Given a feature vector  $X$  and a **qualitative response**  $Y$  taking values in the set  $\mathcal{C}$ , the classification task is to build a function  $C(X)$  that takes as input the feature vector  $X$  and predicts its value for  $Y$ ; i.e.  $C(X) \in \mathcal{C}$ .
- Often we are more interested in **estimating the probabilities** that  $X$  belongs to each **category in  $\mathcal{C}$** .



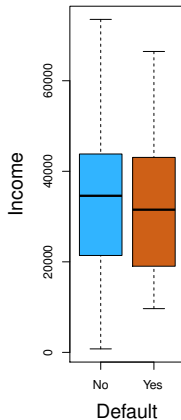
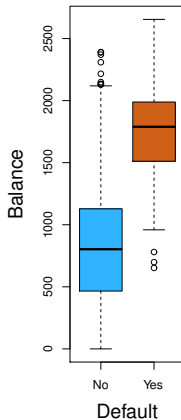
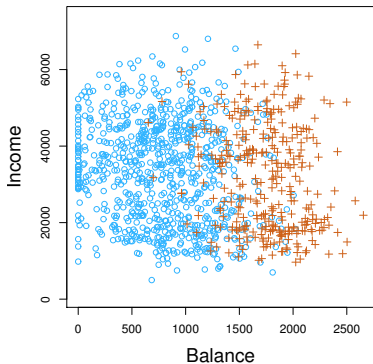
# Classification

- Qualitative variables take values in an unordered set  $\mathcal{C}$ , such as:  
 $\text{eye color} \in \{\text{brown}, \text{blue}, \text{green}\}$   
 $\text{email} \in \{\text{spam}, \text{ham}\}.$
- Given a feature vector  $X$  and a qualitative response  $Y$  taking values in the set  $\mathcal{C}$ , the classification task is to build a function  $C(X)$  that takes as input the feature vector  $X$  and predicts its value for  $Y$ ; i.e.  $C(X) \in \mathcal{C}$ .
- Often we are more interested in estimating the *probabilities* that  $X$  belongs to each category in  $\mathcal{C}$ .

For example, it is more valuable to have an estimate of the probability that an insurance claim is fraudulent, than a classification fraudulent or not.



# Example: Credit Card Default



## Can we use Linear Regression?

Suppose for the **Default** classification task that we code

$$Y = \begin{cases} 0 & \text{if No} \\ 1 & \text{if Yes.} \end{cases}$$

Can we simply perform a linear regression of  $Y$  on  $X$  and classify as **Yes** if  $\hat{Y} > 0.5$ ?

## Can we use Linear Regression?

Suppose for the **Default** classification task that we code

$$Y = \begin{cases} 0 & \text{if No} \\ 1 & \text{if Yes.} \end{cases}$$

Can we simply perform a linear regression of  $Y$  on  $X$  and classify as **Yes** if  $\hat{Y} > 0.5$ ?

- In this case of a binary outcome, linear regression does a good job as a classifier, and is equivalent to *linear discriminant analysis* which we discuss later.
- Since in the population  $E(Y|X = x) = \Pr(Y = 1|X = x)$ , we might think that regression is perfect for this task.

## Can we use Linear Regression?

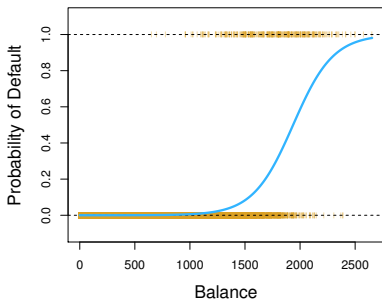
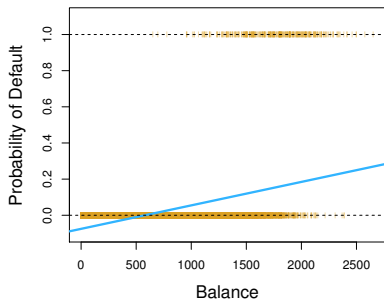
Suppose for the **Default** classification task that we code

$$Y = \begin{cases} 0 & \text{if No} \\ 1 & \text{if Yes.} \end{cases}$$

Can we simply perform a linear regression of  $Y$  on  $X$  and classify as **Yes** if  $\hat{Y} > 0.5$ ?

- In this case of a **binary outcome**, **linear regression** does a **good** job as a classifier, and is **equivalent** to **linear discriminant analysis** which we discuss later.
- Since in the population  $E(Y|X = x) = \Pr(Y = 1|X = x)$ , we might think that regression is perfect for this task.
- However, **linear regression** might produce **probabilities less than zero** or **bigger than one**. **Logistic regression** is more appropriate.

## Linear versus Logistic Regression



The orange marks indicate the response  $Y$ , either 0 or 1. Linear regression does not estimate  $\Pr(Y = 1|X)$  well. **Logistic** regression seems well suited to the task.

## Linear Regression continued

Now suppose we have a response variable with three possible values. A patient presents at the emergency room, and we must classify them according to their symptoms.

$$Y = \begin{cases} 1 & \text{if stroke;} \\ 2 & \text{if drug overdose;} \\ 3 & \text{if epileptic seizure.} \end{cases}$$

This coding suggests an ordering, and in fact implies that the difference between **stroke** and **drug overdose** is the same as between **drug overdose** and **epileptic seizure**.



## Linear Regression continued

Now suppose we have a response variable with **three** possible values. A patient presents at the emergency room, and we must classify them according to their symptoms.

$$Y = \begin{cases} 1 & \text{if stroke;} \\ 2 & \text{if drug overdose;} \\ 3 & \text{if epileptic seizure.} \end{cases}$$

This coding suggests an ordering, and in fact implies that the **difference** between **stroke** and **drug overdose** is the **same** as between **drug overdose** and **epileptic seizure**.

Linear regression is **not appropriate** here.

***Multiclass Logistic Regression*** or ***Discriminant Analysis*** are more appropriate.

# Logistic Regression

Let's write  $p(X) = \Pr(Y = 1|X)$  for short and consider using **balance** to predict **default**. Logistic regression uses the form

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}.$$

( $e \approx 2.71828$  is a mathematical constant [Euler's number.] )

It is easy to see that no matter what values  $\beta_0$ ,  $\beta_1$  or  $X$  take,  $p(X)$  will have values between 0 and 1.

# Logistic Regression

Let's write  $p(X) = \Pr(Y = 1|X)$  for short and consider using **balance** to **predict default**. Logistic regression uses the form

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}.$$

( $e \approx 2.71828$  is a mathematical constant [Euler's number.] )

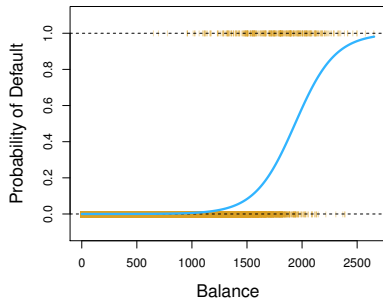
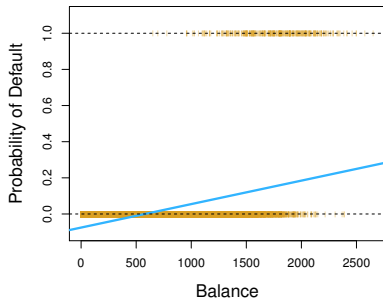
It is easy to see that **no matter** what values  $\beta_0$ ,  $\beta_1$  or  $X$  take,  **$p(X)$**  will have values **between 0 and 1**.

A bit of rearrangement gives

$$\log \left( \frac{p(X)}{1 - p(X)} \right) = \beta_0 + \beta_1 X.$$

This **monotone** transformation is called the **log odds** or **logit** transformation of  $p(X)$ .

# Linear versus Logistic Regression



Logistic regression ensures that our estimate for  $p(X)$  lies between 0 and 1.

# Maximum Likelihood

We use maximum likelihood to estimate the parameters.

$$\ell(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)).$$

This *likelihood* gives the probability of the observed zeros and ones in the data. We pick  $\beta_0$  and  $\beta_1$  to maximize the likelihood of the observed data.

## Maximum Likelihood

We use maximum likelihood to estimate the parameters.

$$\ell(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)).$$

This *likelihood* gives the probability of the observed zeros and ones in the data. We pick  $\beta_0$  and  $\beta_1$  to maximize the likelihood of the observed data.

Most statistical packages can fit linear logistic regression models by maximum likelihood. In R we use the `glm` function.

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.6513	0.3612	-29.5	< 0.0001
balance	0.0055	0.0002	24.9	< 0.0001

## Making Predictions

What is our estimated probability of **default** for someone with a balance of \$1000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}} = \frac{e^{-10.6513 + 0.0055 \times 1000}}{1 + e^{-10.6513 + 0.0055 \times 1000}} = 0.006$$

## Making Predictions

What is our **estimated probability of default** for someone with a balance of \$1000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}} = \frac{e^{-10.6513 + 0.0055 \times 1000}}{1 + e^{-10.6513 + 0.0055 \times 1000}} = 0.006$$

With a balance of \$2000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}} = \frac{e^{-10.6513 + 0.0055 \times 2000}}{1 + e^{-10.6513 + 0.0055 \times 2000}} = 0.586$$



Lets do it again, using **student** as the predictor.

	Coefficient	Std. Error	Z-statistic	P-value
<b>Intercept</b>	-3.5041	0.0707	-49.55	< 0.0001
<b>student[Yes]</b>	0.4049	0.1150	3.52	0.0004

Lets do it again, using **student** as the predictor.

	Coefficient	Std. Error	Z-statistic	P-value
<b>Intercept</b>	-3.5041	0.0707	-49.55	< 0.0001
<b>student[Yes]</b>	0.4049	0.1150	3.52	0.0004

$$\widehat{\Pr}(\text{default}=\text{Yes}|\text{student}=\text{Yes}) = \frac{e^{-3.5041+0.4049 \times 1}}{1 + e^{-3.5041+0.4049 \times 1}} = 0.0431,$$

$$\widehat{\Pr}(\text{default}=\text{Yes}|\text{student}=\text{No}) = \frac{e^{-3.5041+0.4049 \times 0}}{1 + e^{-3.5041+0.4049 \times 0}} = 0.0292.$$

## Logistic regression with several variables

$$\log \left( \frac{p(X)}{1 - p(X)} \right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

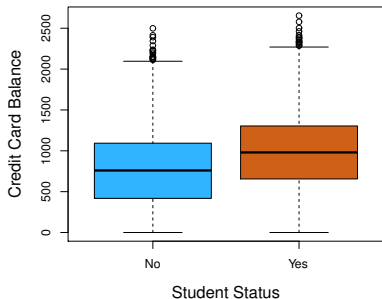
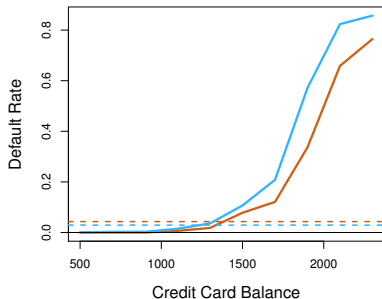
$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}$$


	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student [Yes]	-0.6468	0.2362	-2.74	0.0062

Why is coefficient for **student** negative, while it was positive before?



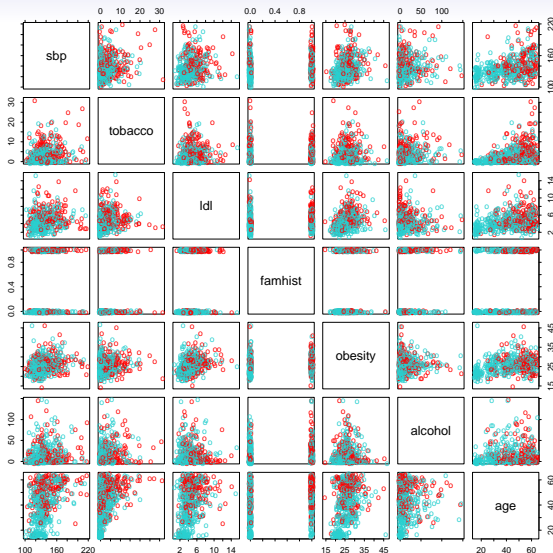
# Confounding



- Students tend to have higher balances than non-students, so their marginal default rate is higher than for non-students. 
- But for each level of balance, students default less than non-students.
- Multiple logistic regression can tease this out.

## Example: South African Heart Disease

- 160 cases of MI (myocardial infarction) and 302 controls (all male in age range 15-64), from Western Cape, South Africa in early 80s.
- Overall prevalence very high in this region: 5.1%.
- Measurements on **seven predictors** (risk factors), shown in scatterplot matrix.
- Goal is to **identify relative strengths and directions** of risk factors.
- This was part of an intervention study aimed at educating the public on healthier diets.



Scatterplot matrix of the *South African Heart Disease* data. The response is color coded — The cases (MI) are red, the controls turquoise. **famhist** is a binary variable, with 1 indicating family history of MI.

```
> heartfit<-glm(chd~.,data=heart,family=binomial)
> summary(heartfit)
```

Call:

```
glm(formula = chd ~ ., family = binomial, data = heart)
```


Coefficients:

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	-4.1295997	0.9641558	-4.283	1.84e-05	***
sbp	0.0057607	0.0056326	1.023	0.30643	
tobacco	0.0795256	0.0262150	3.034	0.00242	**
ldl	0.1847793	0.0574115	3.219	0.00129	**
famhistPresent	0.9391855	0.2248691	4.177	2.96e-05	***
obesity	-0.0345434	0.0291053	-1.187	0.23529	
alcohol	0.0006065	0.0044550	0.136	0.89171	
age	0.0425412	0.0101749	4.181	2.90e-05	***

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 596.11 on 461 degrees of freedom  
Residual deviance: 483.17 on 454 degrees of freedom  
AIC: 499.17

## Case-control sampling and logistic regression

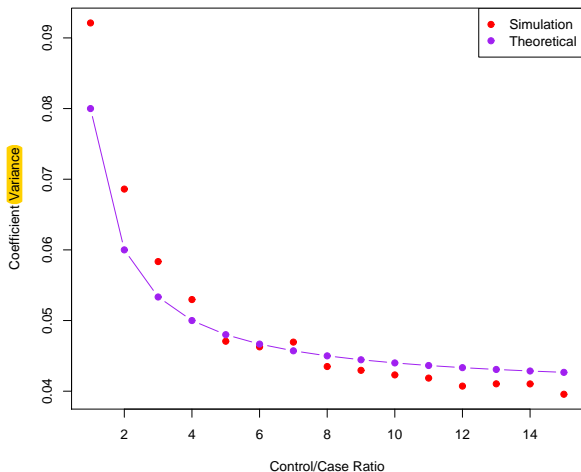
- In South African data, there are 160 cases, 302 controls —  $\tilde{\pi} = 0.35$  are cases. Yet the prevalence of MI in this region is  $\pi = 0.05$ .
- With case-control samples, we can estimate the regression parameters  $\beta_j$  accurately (if our model is correct); the constant term  $\beta_0$  is incorrect. 
- We can correct the estimated intercept by a simple transformation

$$\hat{\beta}_0^* = \hat{\beta}_0 + \log \frac{\pi}{1 - \pi} - \log \frac{\tilde{\pi}}{1 - \tilde{\pi}}$$

- Often cases are rare and we take them all; up to five times that number of controls is sufficient. See next frame



## Diminishing returns in unbalanced binary data



Sampling more controls than cases reduces the variance of the parameter estimates. But after a ratio of about 5 to 1 the variance reduction flattens out.

## Logistic regression with more than two classes

So far we have discussed logistic regression with two classes. It is easily generalized to more than two classes. One version (used in the R package **glmnet**) has the symmetric form

$$\Pr(Y = k|X) = \frac{e^{\beta_{0k} + \beta_{1k}X_1 + \dots + \beta_{pk}X_p}}{\sum_{\ell=1}^K e^{\beta_{0\ell} + \beta_{1\ell}X_1 + \dots + \beta_{p\ell}X_p}}$$

Here there is a linear function for *each* class.

## Logistic regression with more than two classes

So far we have discussed logistic regression with two classes. It is easily generalized to more than two classes. One version (used in the R package `glmnet`) has the symmetric form

$$\Pr(Y = k|X) = \frac{e^{\beta_{0k} + \beta_{1k}X_1 + \dots + \beta_{pk}X_p}}{\sum_{\ell=1}^K e^{\beta_{0\ell} + \beta_{1\ell}X_1 + \dots + \beta_{p\ell}X_p}}$$

Here there is a linear function for *each* class.


(The *mathier* students will recognize that some cancellation is possible, and only  $K - 1$  linear functions are needed as in 2-class logistic regression.)

## Logistic regression with more than two classes

So far we have discussed logistic regression with two classes. It is easily generalized to **more than two classes**. One version (used in the R package **glmnet**) has the symmetric form

$$\Pr(Y = k|X) = \frac{e^{\beta_{0k} + \beta_{1k}X_1 + \dots + \beta_{pk}X_p}}{\sum_{\ell=1}^K e^{\beta_{0\ell} + \beta_{1\ell}X_1 + \dots + \beta_{p\ell}X_p}}$$

Here there is a linear function for *each* class.

(The *mathier* students will recognize that some cancellation is possible, and only  **$K - 1$  linear functions** are **needed** as in 2-class logistic regression.) 

**Multiclass logistic regression** is also referred to as ***multinomial regression***.

# Discriminant Analysis

Here the approach is to model the distribution of  $X$  in each of the classes separately, and then use *Bayes theorem* to flip things around and obtain  $\Pr(Y|X)$ .

When we use normal (Gaussian) distributions for each class, this leads to linear or quadratic discriminant analysis.

However, this approach is quite general, and other distributions can be used as well. We will focus on normal distributions.

## Bayes theorem for classification

Thomas Bayes was a famous mathematician whose name represents a big subfield of statistical and probabilistic modeling. Here we focus on a simple result, known as Bayes theorem:

$$\Pr(Y = k|X = x) = \frac{\Pr(X = x|Y = k) \cdot \Pr(Y = k)}{\Pr(X = x)}$$

## Bayes theorem for classification

Thomas Bayes was a famous mathematician whose name represents a big subfield of statistical and probabilistic modeling. Here we focus on a simple result, known as Bayes theorem:

$$\Pr(Y = k|X = x) = \frac{\Pr(X = x|Y = k) \cdot \Pr(Y = k)}{\Pr(X = x)}$$

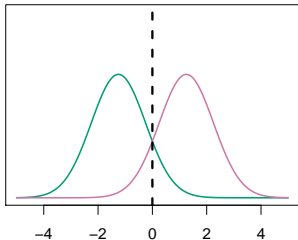
One writes this slightly differently **for discriminant analysis**:

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^K \pi_l f_l(x)}, \quad \text{where}$$

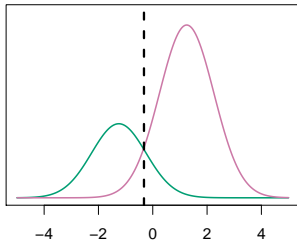
- $f_k(x) = \Pr(X = x|Y = k)$  is the **density** for  $X$  in class  $k$ . Here we will use **normal densities** for these, separately in each class.
- $\pi_k = \Pr(Y = k)$  is the marginal or **prior probability** for class  $k$ .

## Classify to the highest density

$$\pi_1=.5, \pi_2=.5$$



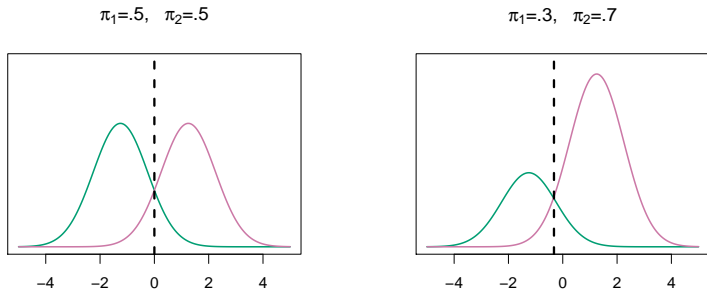
$$\pi_1=.3, \pi_2=.7$$



We classify a new point according to which density is highest.



## Classify to the highest density



We classify a new point according to which density is highest.

When the priors are different, we take them into account as well, and compare  $\pi_k f_k(x)$ . On the right, we favor the pink class — the decision boundary has shifted to the left.



## Why discriminant analysis?

- When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable. Linear discriminant analysis does not suffer from this problem.
- If  $n$  is small and the distribution of the predictors  $X$  is approximately normal in each of the classes, the linear discriminant model is again more stable than the logistic regression model.
- Linear discriminant analysis is popular when we have more than two response classes, because it also provides low-dimensional views of the data.

## Linear Discriminant Analysis when $p = 1$

The Gaussian density has the form

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

Here  $\mu_k$  is the mean, and  $\sigma_k^2$  the variance (in class  $k$ ). We will assume that all the  $\sigma_k = \sigma$  are the same.

## Linear Discriminant Analysis when $p = 1$

The Gaussian density has the form

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

Here  $\mu_k$  is the mean, and  $\sigma_k^2$  the variance (in class  $k$ ). We will assume that all the  $\sigma_k = \sigma$  are the same.

Plugging this into Bayes formula, we get a rather complex expression for  $p_k(x) = \Pr(Y = k|X = x)$ :

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2}}{\sum_{l=1}^K \pi_l \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu_l}{\sigma}\right)^2}}$$

Happily, there are simplifications and cancellations.

## Discriminant functions

To classify at the value  $X = x$ , we need to see which of the  $p_k(x)$  is largest. Taking logs, and discarding terms that do not depend on  $k$ , we see that this is equivalent to assigning  $x$  to the class with the largest *discriminant score*:

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

Note that  $\delta_k(x)$  is a *linear* function of  $x$ .

## Discriminant functions

To classify at the value  $X = x$ , we need to see which of the  $p_k(x)$  is largest. Taking logs, and discarding terms that do not depend on  $k$ , we see that this is equivalent to assigning  $x$  to the class with the largest *discriminant score*:

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

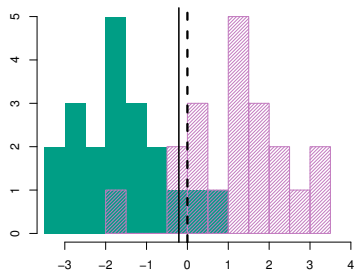
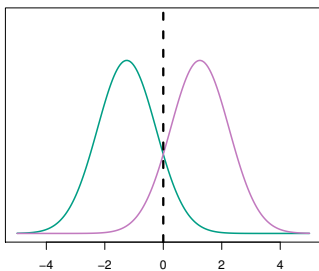
Note that  $\delta_k(x)$  is a *linear* function of  $x$ .

If there are  $K = 2$  classes and  $\pi_1 = \pi_2 = 0.5$ , then one can see that the *decision boundary* is at

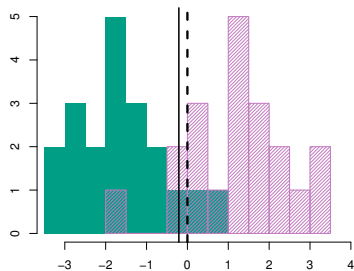
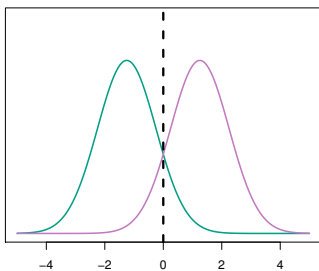


$$x = \frac{\mu_1 + \mu_2}{2}.$$

(See if you can show this)



Example with  $\mu_1 = -1.5$ ,  $\mu_2 = 1.5$ ,  $\pi_1 = \pi_2 = 0.5$ , and  $\sigma^2 = 1$ .



Example with  $\mu_1 = -1.5$ ,  $\mu_2 = 1.5$ ,  $\pi_1 = \pi_2 = 0.5$ , and  $\sigma^2 = 1$ .

Typically we don't know these parameters; we just have the training data. In that case we simply estimate the parameters and plug them into the rule.



## Estimating the parameters

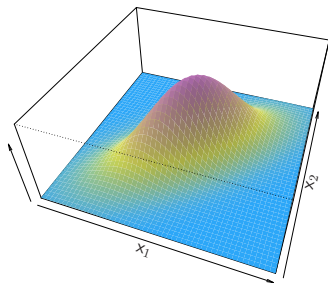
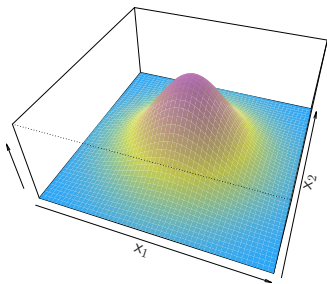
$$\hat{\pi}_k = \frac{n_k}{n}$$

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i: y_i=k} x_i$$

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

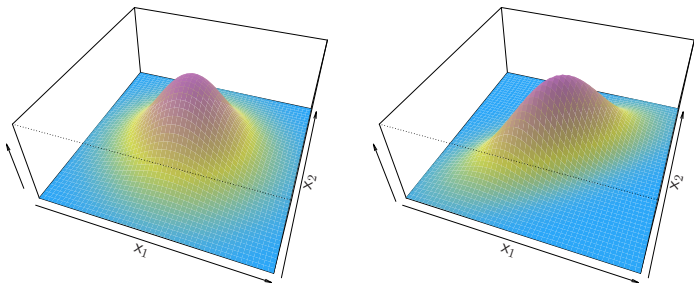
where  $\hat{\sigma}_k^2 = \frac{1}{n_k-1} \sum_{i: y_i=k} (x_i - \hat{\mu}_k)^2$  is the usual formula for the estimated variance in the  $k$ th class.

## Linear Discriminant Analysis when $p > 1$



$$\text{Density: } f(x) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

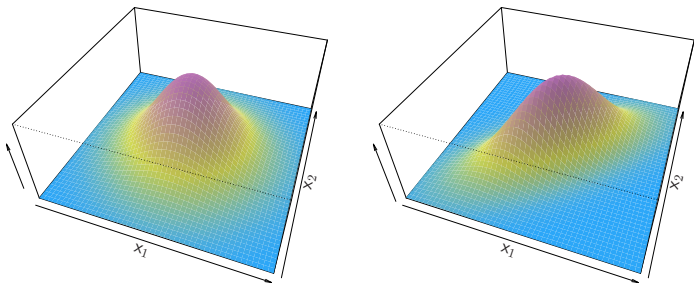
## Linear Discriminant Analysis when $p > 1$



$$\text{Density: } f(x) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

$$\text{Discriminant function: } \delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$$

## Linear Discriminant Analysis when $p > 1$



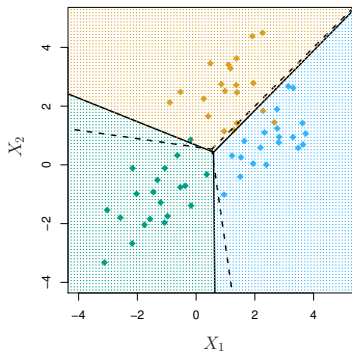
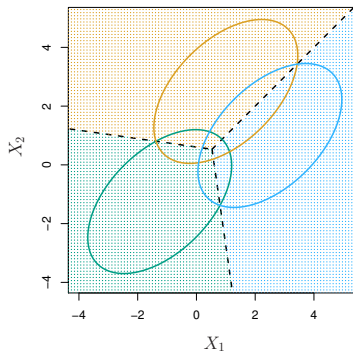
$$\text{Density: } f(x) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

**Discriminant function:**  $\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k$

Despite its complex form,

$\delta_k(x) = c_{k0} + c_{k1}x_1 + c_{k2}x_2 + \dots + c_{kp}x_p$  — a **linear** function.

## Illustration: $p = 2$ and $K = 3$ classes



Here  $\pi_1 = \pi_2 = \pi_3 = 1/3$ .



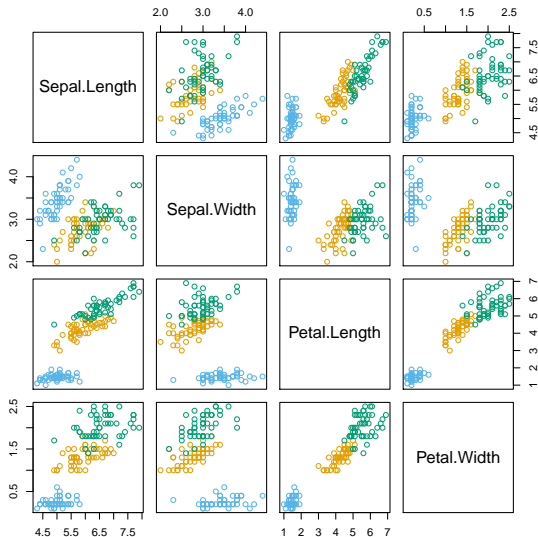
The **dashed lines** are known as the *Bayes decision boundaries*. Were they known, they would yield the fewest misclassification errors, among all possible classifiers.

# Fisher's Iris Data

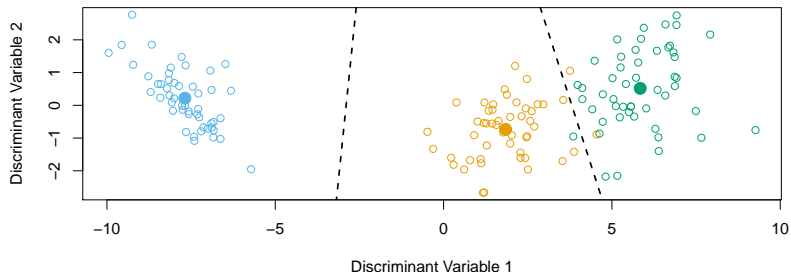
4 variables  
3 species  
50 samples/class

- Setosa
- Versicolor
- Virginica

LDA classifies all  
but 3 of the 150  
training samples  
correctly.



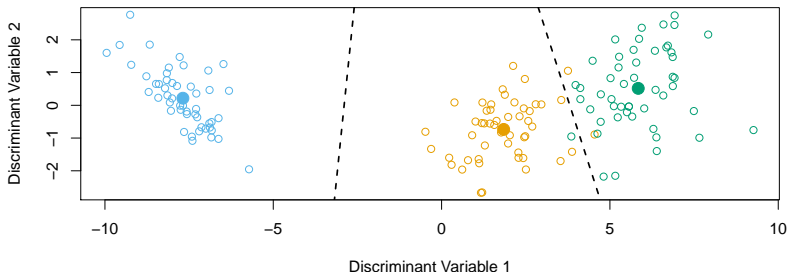
## Fisher's Discriminant Plot



When there are  $K$  classes, linear discriminant analysis can be viewed exactly in a  $K - 1$  dimensional plot.

Why?

## Fisher's Discriminant Plot

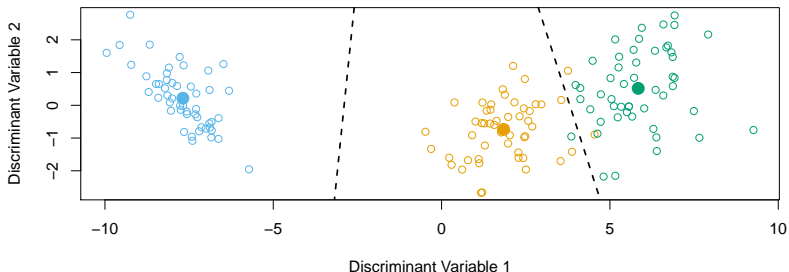


When there are  $K$  classes, linear discriminant analysis can be viewed exactly in a  $K - 1$  dimensional plot.

Why? Because it essentially classifies to the closest centroid, and they span a  $K - 1$  dimensional plane.



## Fisher's Discriminant Plot



When there are  $K$  classes, linear discriminant analysis can be viewed exactly in a  $K - 1$  dimensional plot.

Why? Because it essentially classifies to the closest centroid, and they span a  $K - 1$  dimensional plane.

Even when  $K > 3$ , we can find the “best” 2-dimensional plane for visualizing the discriminant rule.

## From $\delta_k(x)$ to probabilities

Once we have estimates  $\hat{\delta}_k(x)$ , we can turn these into estimates for class probabilities:

$$\widehat{\Pr}(Y = k|X = x) = \frac{e^{\hat{\delta}_k(x)}}{\sum_{l=1}^K e^{\hat{\delta}_l(x)}}.$$

So classifying to the largest  $\hat{\delta}_k(x)$  amounts to classifying to the class for which  $\widehat{\Pr}(Y = k|X = x)$  is largest.

## From $\delta_k(x)$ to probabilities

Once we have estimates  $\hat{\delta}_k(x)$ , we can turn these into estimates for class probabilities:

$$\widehat{\Pr}(Y = k|X = x) = \frac{e^{\hat{\delta}_k(x)}}{\sum_{l=1}^K e^{\hat{\delta}_l(x)}}.$$

So classifying to the largest  $\hat{\delta}_k(x)$  amounts to classifying to the class for which  $\widehat{\Pr}(Y = k|X = x)$  is largest.



When  $K = 2$ , we classify to class 2 if  $\widehat{\Pr}(Y = 2|X = x) \geq 0.5$ , else to class 1.

## LDA on Credit Data

		<i>True Default Status</i>		
		No	Yes	Total
<i>Predicted Default Status</i>	No	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

$(23 + 252)/10000$  errors — a 2.75% misclassification rate!

Some caveats:

- This is *training* error, and we may be overfitting.

## LDA on Credit Data

		<i>True Default Status</i>		
		No	Yes	Total
<i>Predicted Default Status</i>	No	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

$(23 + 252)/10000$  errors — a 2.75% misclassification rate!

Some caveats:

- This is *training* error, and we may be overfitting. Not a big concern here since  $n = 10000$  and  $p = 4$ !

## LDA on Credit Data

		<i>True Default Status</i>		
		No	Yes	Total
<i>Predicted Default Status</i>	No	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

$(23 + 252)/10000$  errors — a 2.75% misclassification rate!

Some caveats:

- This is *training* error, and we may be overfitting. Not a big concern here since  $n = 10000$  and  $p = 4$ !
- If we classified to the prior — always to class **No** in this case — we would make  $333/10000$  errors, or only 3.33%.

## LDA on Credit Data

		<i>True Default Status</i>		
		No	Yes	Total
<i>Predicted Default Status</i>	No	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

$(23 + 252)/10000$  errors — a 2.75% misclassification rate!

Some caveats:

- This is *training error*, and we may be overfitting. Not a big concern here since  $n = 10000$  and  $p = 4$ !



- If we *classified to the prior* — *always to class No* in this case — we would make  $333/10000$  errors, or only 3.33%.
- Of the true **No**'s, we make  $23/9667 = 0.2\%$  errors; of the true **Yes**'s, we make  $252/333 = 75.7\%$  errors!

## Types of errors



**False positive rate:** The fraction of negative examples that are classified as positive — 0.2% in example.

**False negative rate:** The fraction of positive examples that are classified as negative — 75.7% in example.

We produced this table by classifying to class **Yes** if

$$\widehat{\Pr}(\text{Default} = \text{Yes} | \text{Balance}, \text{Student}) \geq 0.5$$

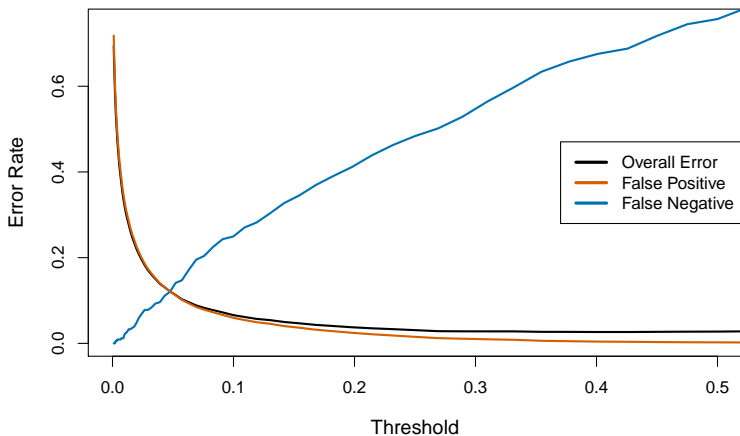
We can change the two error rates by changing the threshold from 0.5 to some other value in  $[0, 1]$ :

$$\widehat{\Pr}(\text{Default} = \text{Yes} | \text{Balance}, \text{Student}) \geq \text{threshold},$$

and vary *threshold*.

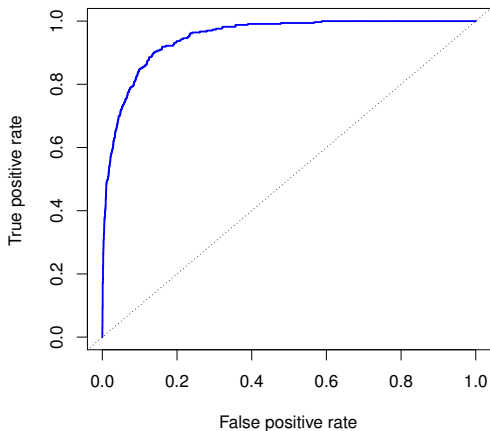


## Varying the *threshold*



In order to **reduce the false negative** rate, we may want to **reduce the threshold to 0.1** or less.

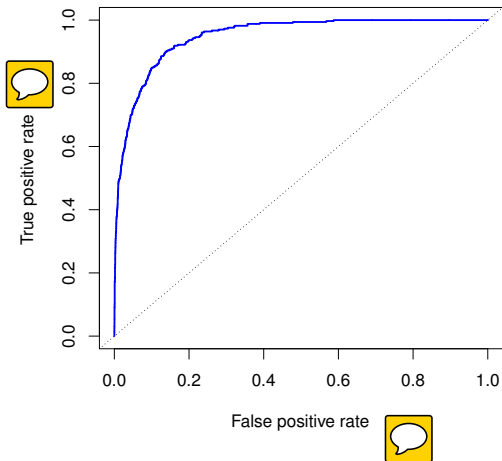
ROC Curve



The *ROC plot* displays both simultaneously.



## ROC Curve



The *ROC plot* displays both simultaneously.

Sometimes we use the **AUC** or *area under the curve* to summarize the overall performance. Higher **AUC** is good.

## Other forms of Discriminant Analysis

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

When  $f_k(x)$  are Gaussian densities, with the same covariance matrix  $\Sigma$  in each class, this leads to linear discriminant analysis. By altering the forms for  $f_k(x)$ , we get different classifiers.

- With Gaussians but different  $\Sigma_k$  in each class, we get *quadratic discriminant analysis*.

## Other forms of Discriminant Analysis

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

When  $f_k(x)$  are Gaussian densities, with the same covariance matrix  $\Sigma$  in each class, this leads to linear discriminant analysis. By altering the forms for  $f_k(x)$ , we get different classifiers.

- With Gaussians but different  $\Sigma_k$  in each class, we get *quadratic discriminant analysis*.
- With  $f_k(x) = \prod_{j=1}^p f_{jk}(x_j)$  (conditional independence model) in each class we get *naive Bayes*. For Gaussian this means the  $\Sigma_k$  are diagonal.

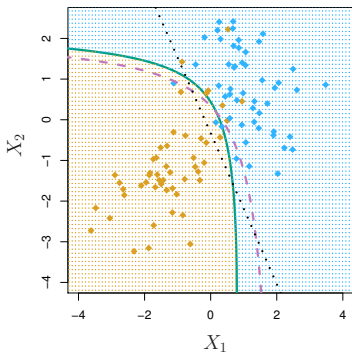
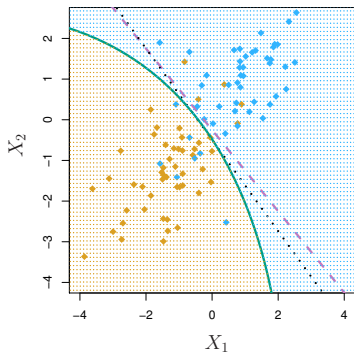
## Other forms of Discriminant Analysis

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

When  $f_k(x)$  are Gaussian densities, with the same covariance matrix  $\Sigma$  in each class, this leads to linear discriminant analysis. By altering the forms for  $f_k(x)$ , we get different classifiers.

- With Gaussians but different  $\Sigma_k$  in each class, we get *quadratic discriminant analysis*.
- With  $f_k(x) = \prod_{j=1}^p f_{jk}(x_j)$  (conditional independence model) in each class we get *naive Bayes*. For Gaussian this means the  $\Sigma_k$  are diagonal.
- Many other forms, by proposing specific density models for  $f_k(x)$ , including nonparametric approaches.

# Quadratic Discriminant Analysis



$$\delta_k(x) = -\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k) + \log \pi_k$$

Because the  $\Sigma_k$  are different, the quadratic terms matter.

# Naive Bayes

Assumes features are independent in each class.

Useful when  $p$  is large, and so multivariate methods like QDA and even LDA break down.

- Gaussian naive Bayes assumes each  $\Sigma_k$  is diagonal:

$$\delta_k(x) \propto \log \left[ \pi_k \prod_{j=1}^p f_{kj}(x_j) \right] = -\frac{1}{2} \sum_{j=1}^p \frac{(x_j - \mu_{kj})^2}{\sigma_{kj}^2} + \log \pi_k$$

- can use for *mixed* feature vectors (qualitative and quantitative). If  $X_j$  is qualitative, replace  $f_{kj}(x_j)$  with probability mass function (histogram) over discrete categories.

Despite strong assumptions, naive Bayes often produces good classification results.



## Logistic Regression versus LDA

For a two-class problem, one can show that for LDA

$$\log \left( \frac{p_1(x)}{1 - p_1(x)} \right) = \log \left( \frac{p_1(x)}{p_2(x)} \right) = c_0 + c_1 x_1 + \dots + c_p x_p$$

So it has the same form as logistic regression.

The difference is in how the parameters are estimated.

## Logistic Regression versus LDA

For a two-class problem, one can show that for LDA

$$\log \left( \frac{p_1(x)}{1 - p_1(x)} \right) = \log \left( \frac{p_1(x)}{p_2(x)} \right) = c_0 + c_1 x_1 + \dots + c_p x_p$$

So it has the same form as logistic regression.

The difference is in how the parameters are estimated.

- Logistic regression uses the conditional likelihood based on  $\Pr(Y|X)$  (known as *discriminative learning*).

## Logistic Regression versus LDA

For a two-class problem, one can show that for LDA

$$\log \left( \frac{p_1(x)}{1 - p_1(x)} \right) = \log \left( \frac{p_1(x)}{p_2(x)} \right) = c_0 + c_1 x_1 + \dots + c_p x_p$$

So it has the same form as logistic regression.

The difference is in how the parameters are estimated.

- Logistic regression uses the conditional likelihood based on  $\Pr(Y|X)$  (known as *discriminative learning*).
- LDA uses the full likelihood based on  $\Pr(X, Y)$  (known as *generative learning*).

## Logistic Regression versus LDA

For a two-class problem, one can show that for LDA

$$\log \left( \frac{p_1(x)}{1 - p_1(x)} \right) = \log \left( \frac{p_1(x)}{p_2(x)} \right) = c_0 + c_1 x_1 + \dots + c_p x_p$$

So it has the same form as logistic regression.

The difference is in how the parameters are estimated.

- Logistic regression uses the conditional likelihood based on  $\Pr(Y|X)$  (known as *discriminative learning*).
- LDA uses the full likelihood based on  $\Pr(X, Y)$  (known as *generative learning*).
- Despite these differences, in practice the results are often very similar.

## Logistic Regression versus LDA

For a two-class problem, one can show that for LDA

$$\log \left( \frac{p_1(x)}{1 - p_1(x)} \right) = \log \left( \frac{p_1(x)}{p_2(x)} \right) = c_0 + c_1 x_1 + \dots + c_p x_p$$

So it has the same form as logistic regression.

The difference is in how the parameters are estimated.

- Logistic regression uses the conditional likelihood based on  $\Pr(Y|X)$  (known as *discriminative learning*).
- LDA uses the full likelihood based on  $\Pr(X, Y)$  (known as *generative learning*).
- Despite these differences, in practice the results are often very similar.

Footnote: logistic regression can also fit quadratic boundaries like QDA, by explicitly including quadratic terms in the model.

# Summary

- Logistic regression is very popular for classification, especially when  $K = 2$ .
- LDA is useful when  $n$  is small, or the classes are well separated, and Gaussian assumptions are reasonable. Also when  $K > 2$ .
- Naive Bayes is useful when  $p$  is very large.
- See Section 4.5 for some comparisons of logistic regression, LDA and KNN.