
Topic 3. Classification

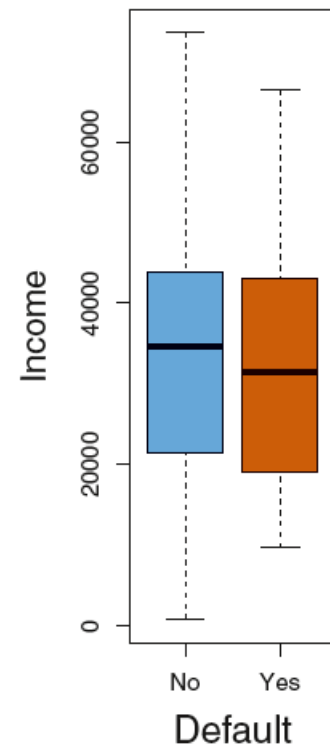
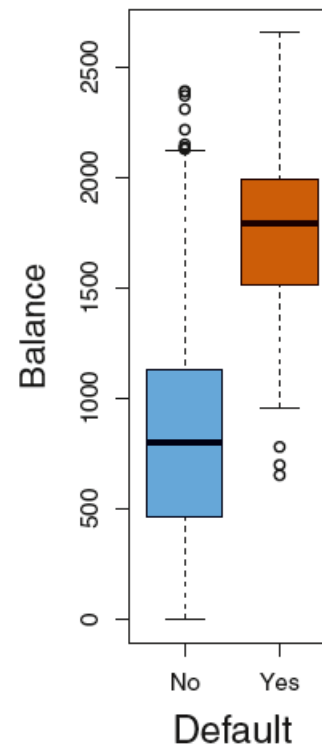
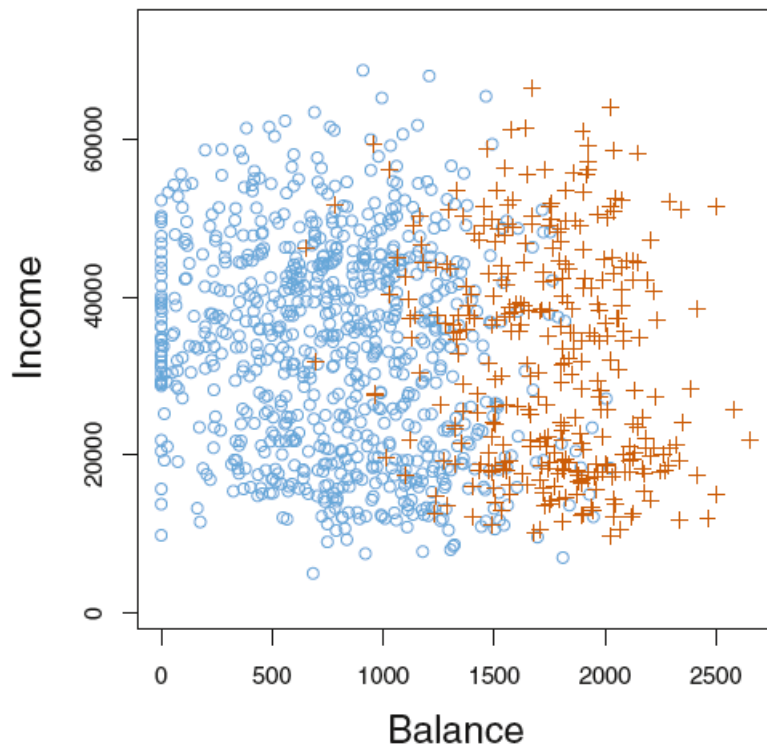
Overview

- **Classification:** predicting a qualitative response
- A classification technique is called a *classifier*.
- **Popular classifiers**
 - Logistic regression
 - Linear discriminant analysis (LDA) & QDA
 - K-Nearest Neighbors (KNN)
- **Performance assessment and comparison**

Logistic Regression

Motivating Example

- Predicting whether an individual will default on his/her credit card payment
- **Default** data set: $Y = \text{default}$ (yes/no), $X_1 = \text{balance}$, $X_2 = \text{income}$



Why Not Linear Regression?

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

- Linear regression model using the binary response

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon$$

- Problem: the prediction ($\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1$) can take any value between negative and positive infinity. How do we interpret values greater than 1? Or values between 0 and 1?

Why Not Linear Regression?

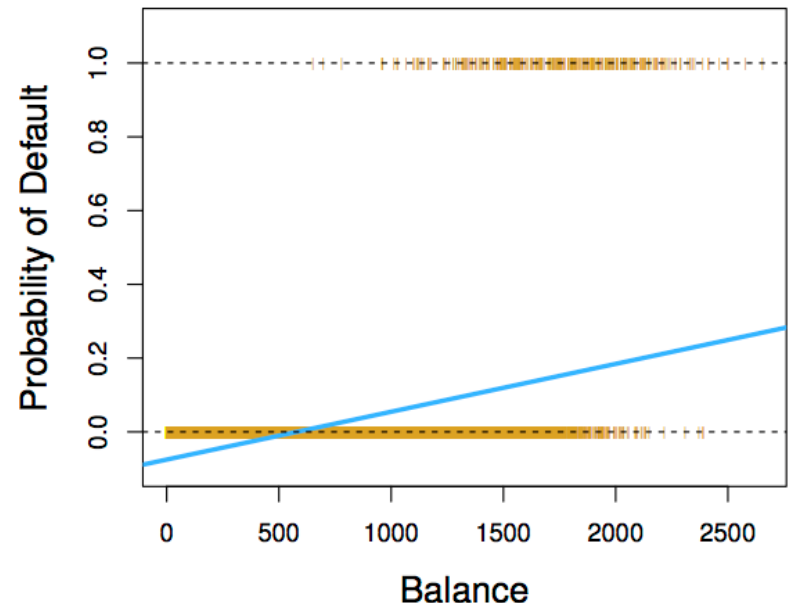
$$Y = \begin{cases} 0 & \text{default} = \text{No} \\ 1 & \text{default} = \text{Yes} \end{cases}$$

- Linear regression model using the probability as response

$$P(Y = 1) = \beta_0 + \beta_1 X_1$$

*the probability
of default*

- Problem: now the values between 0 and 1 makes sense. But it is still difficult to interpret negative values and values greater than 1.

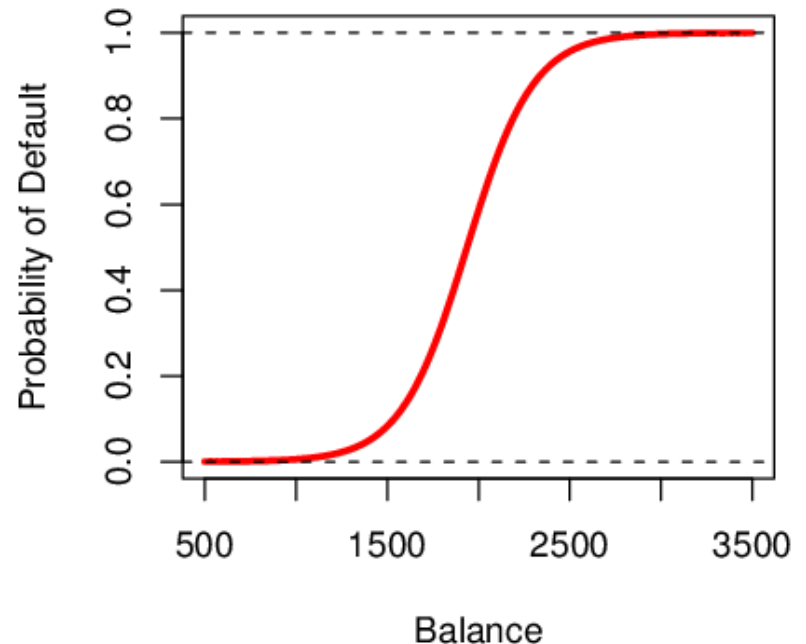


Solution: Logistic Function

$$P(Y = 1) = \beta_0 + \beta_1 X_1$$

- Left side: $[0,1]$ Right side: $(-\infty, \infty)$
- **Question:** is there a transformation of the right side such that it has the same range as the left side?
- The logistic function

$$P(Y = 1) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$



Logistic Regression

- Logistic regression is very similar to linear regression

logistic regression:
$$\log \left(\frac{P(Y = 1)}{1 - P(Y = 1)} \right) = \beta_0 + \beta_1 X$$

linear regression:
$$Y \approx \beta_0 + \beta_1 X$$

- In linear regression, β_1 represents the average change in Y for one-unit increase in X . However, this simple interpretation does not work for logistic regression because we are predicting the probability $P(Y)$, not the response Y .

Interpreting β_1

$$\log \left(\frac{P(Y = 1)}{1 - P(Y = 1)} \right) = \beta_0 + \beta_1 X \leftrightarrow P(Y = 1) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

- If $\beta_1 = 0$, this means that there is no relationship between the response and the predictor(s).
- If $\beta_1 > 0$, this means that when X gets larger so does the probability of default.
- If $\beta_1 < 0$, this means that when X gets larger, the probability of default gets smaller.
- How much bigger or smaller depends on value of the slope.

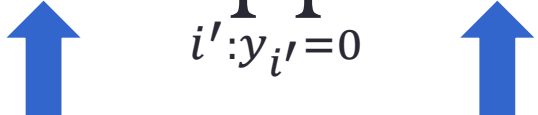
Estimating Coefficients

- Find estimates of the parameters β_0, β_1 based on training data

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

- Maximum likelihood method

Likelihood function

$$l(\beta_0, \beta_1) = \prod_{i: y_i=1} P(y_i = 1) \prod_{i': y_{i'}=0} 1 - P(y_{i'} = 1)$$


$$P(y_i = 1) = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}}$$

$$P(y_{i'} = 0) = \frac{1}{1 + e^{\beta_0 + \beta_1 x_{i'}}}$$

- The estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ are chosen to maximize this function.

Results of the Default Example

	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

- Predicting default using balance
- Use a z test instead of t test (as used in linear regression) to see whether β_0 and β_1 are significantly different from zero
- Here the p-value for balance is very small, and $\hat{\beta}_1$ is positive. That means if the balance increases, then the probability of default will increase as well.

Making Predictions

- Suppose an individual has an average balance of \$1000. What is their probability of default?

$$P(Y = 1) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}} = \frac{e^{-10.6513 + 0.0055 \times 1000}}{1 + e^{-10.6513 + 0.0055 \times 1000}} = 0.00576$$

- The predicted probability of default for an individual with a balance of \$1000 is less than 1%.
- For a balance of \$2000, the probability is much higher, and equals to 0.586 (58.6%).

Qualitative Predictors

- We can predict if an individual default by checking if she is a student or not. Thus we can use a qualitative variable “student” coded as a dummy variable (student=1, non-student=0).

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

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

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

- The estimate is positive, which indicates that students tend to have higher default probabilities than non-students.

Multiple Logistic Regression

- Predicting a binary response using multiple predictors

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

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

Default Example

- Predict **default** using **balance**, **income**, and **student** (qualitative)

	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

- The p-values of **balance** and **student** are very small, indicating that they are associated with the probability of default.
- The coefficient for **student** is negative, indicating that students are less likely to default than non-students.

Predictions

- A **student** with a credit card balance of \$1,500 and an income of \$40,000 has an estimated probability of default

$$P(Y = 1) = \frac{e^{-10.869 + 0.00574 \times 1500 + 0.003 \times 40 - 0.6468 \times 1}}{1 + e^{-10.869 + 0.00574 \times 1500 + 0.003 \times 40 - 0.6468 \times 1}} = 0.058$$

- A **non-student** with the same balance and income has an estimated probability of default

$$P(Y = 1) = \frac{e^{-10.869 + 0.00574 \times 1,500 + 0.003 \times 40 - 0.6468 \times 0}}{1 + e^{-10.869 + 0.00574 \times 1,500 + 0.003 \times 40 - 0.6468 \times 0}} = 0.105$$

An Apparent Paradox!

- Predicting **default** using only **student**

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

Conclusion: students tend to have higher probability of default than non-students.

- Predicting **default** using **balance**, **income**, and **student**

	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

Conclusion: students tend to have lower probability of default than non-students.

Interpretation

- Students tend to have higher **balance**. Higher **balance** tends to have higher probability of default. So, if not consider **balance**, students tend to have higher probability of default.

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

Interpretation

- However, given the **balance** (i.e., after adjusting for the effect of **balance**), students tend to have lower probability of default.

	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

Useful Inference for Credit Card Company

- **Question:** to whom should they offer credit card?
- A student is riskier than a non-student if no information about the student's credit card balance is available
- However, that student is less risky than a non-student with the same credit card balance!

Multinomial Logistic Regression

- **Question:** How to extend logistic regression to response variables with more than two classes?
- Multinomial Logistic Regression
- Two ways:
 - Select a single class as baseline
 - Treat all classes symmetrically

Linear Discriminant Analysis (LDA)

Assumptions of LDA

- Each predictor variable is normally distributed.
- If there are more than one predictor, the predictors follow a multivariate normal distribution.

Why Not Logistic Regression?

- In the case where n is small and the distribution of predictors X is approximately normal in each of the classes, LDA is more stable than Logistic regression.
- LDA is more popular when the response has more than two classes (Logistic regression is usually used when there are only two classes).

The Bayes Classifier

➤ Bayes' theorem

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

- K : the total number of response classes
- π_k : the overall or prior probability of the k th class
- $f_k(x)$: the density function of observations from the k th class

- **Rule of classification:** Given an observation $X = x$, calculate $P(Y = 1|X = x), P(Y = 2|X = x), \dots, P(Y = K|X = x)$. Assign this observation to the class with the largest probability.

Example

- Default data set, $Y = \text{default}(\text{No/Yes})$, $X = \text{balance}$
- The number of classes $K = 2$

$$\begin{aligned} P(Y = 0|X = x) &= P(\text{default} = \text{No}|\text{balance} = x) \\ &= \frac{\pi_0 f_0(x)}{\pi_0 f_0(x) + \pi_1 f_1(x)} \end{aligned}$$

$$\begin{aligned} P(Y = 1|X = x) &= P(\text{default} = \text{Yes}|\text{balance} = x) \\ &= \frac{\pi_1 f_1(x)}{\pi_0 f_0(x) + \pi_1 f_1(x)} \end{aligned}$$

- Given $\text{balance} = \$1000$, if we find $P(Y = 0|X = 1000) = 0.7$, $P(Y = 1|X = 1000) = 0.3$, the predicted class is $Y = 0$, i.e., no default.

Gold Standard for Classification

- In theory, the Bayes classifier has the best performance in classification, so we would always like to use it.
- However, for real data, we do not know the distribution of predictor(s) in each class, so computing the Bayes classifier is impossible.
- The Bayes classifier serves as an unattainable gold standard against which to compare other classifiers.

Idea of LDA

- Assume in each class the predictor follows a normal distribution

Predictor in class k : $X \sim N(\mu_k, \sigma_k^2)$

- Assume those normal distributions have equal variance

$$\sigma_1^2 = \sigma_2^2 = \dots = \sigma_K^2 = \sigma^2$$

- So the density function in class k is

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

Idea of LDA

- Plugging the normal density function into the Bayes classifier

$$\boxed{\log P(Y = k | X = x)} = \boxed{\frac{\mu_k}{\sigma^2}} \cdot x + \boxed{\log(\pi_k) - \frac{\mu_k^2}{2\sigma^2}}$$

$\delta_k(x) = a \cdot x + b$ **Discriminant function**

- **Rule of classification:** Given an observation $X = x$, calculate $\delta_1(x), \delta_2(x), \dots, \delta_K(x)$. Assign this observation to the class with the largest δ .

Estimates Used in LDA

- To calculate $\delta_k(x)$, we need to find estimates for the prior probabilities π_k and parameters μ_k, σ^2 of the normal distribution

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

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

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

- μ_k is estimated by the average of the training data from the k th class.
- σ^2 is estimated by the weighted average of the sample variances for the K classes.
- π_k is estimated by the proportion of the training data that belong to the k th class.

LDA for Binary Response

- A binary response Y , single predictor
- Discriminant function for class 1

$$\delta_1(x) = \frac{\mu_1}{\sigma^2} \cdot x + \left[\log(\pi_1) - \frac{\mu_1^2}{2\sigma^2} \right]$$

- Discriminant function for class 2

$$\delta_2(x) = \frac{\mu_2}{\sigma^2} \cdot x + \left[\log(\pi_2) - \frac{\mu_2^2}{2\sigma^2} \right]$$

- Decision boundary (assume $\pi_1 = \pi_2 = 0.5$)

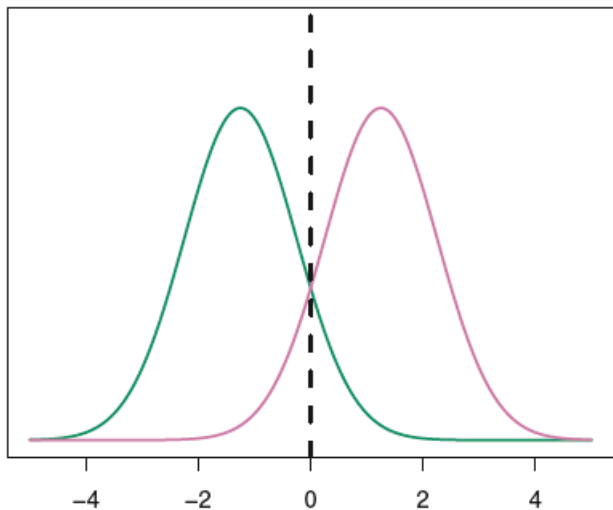
$$\delta_1(x_0) = \delta_2(x_0) \rightarrow x_0 = \frac{\mu_1 + \mu_2}{2}$$

- Rule of classification: assume $\mu_2 > \mu_1$,

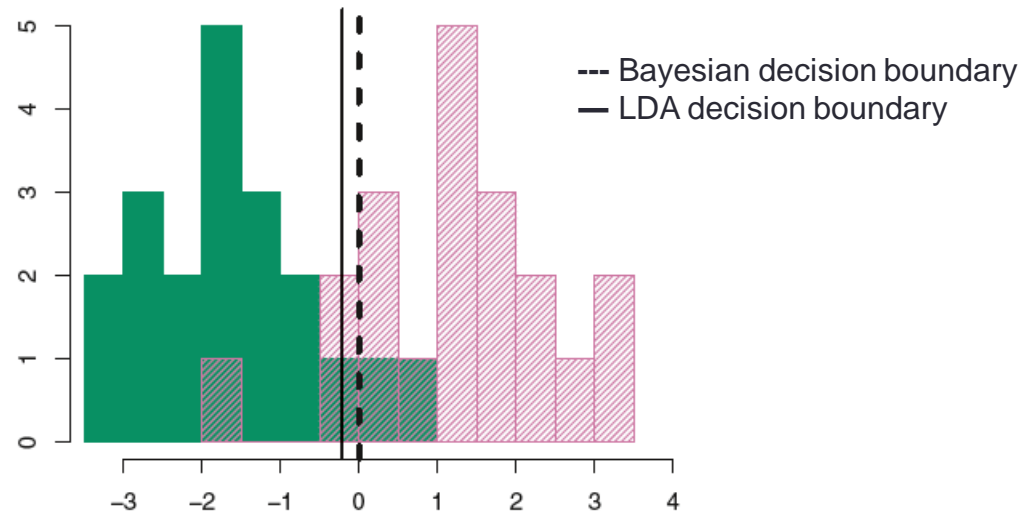
$$y(x) = \begin{cases} \text{class 2, if } x > x_0 \\ \text{class 1, if } x < x_0 \end{cases}$$

A Simple Example

- Normal density functions $f_1(x)$ and $f_2(x)$ from the two classes
- The two density functions overlap, so there is some uncertainty to classify an observation with unknown class.
- 20 observations were drawn from each of the two classes.
- LDA performs pretty well in prediction: LDA error rate=11.1% vs. Bayes error rate=10.6%



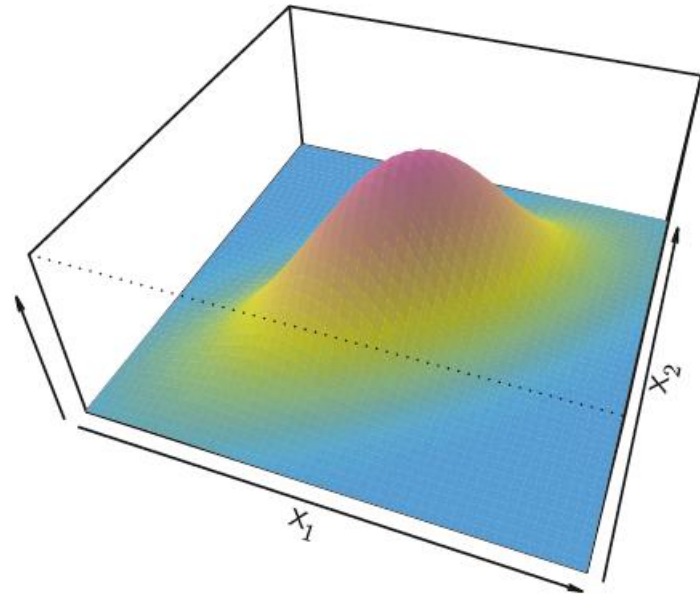
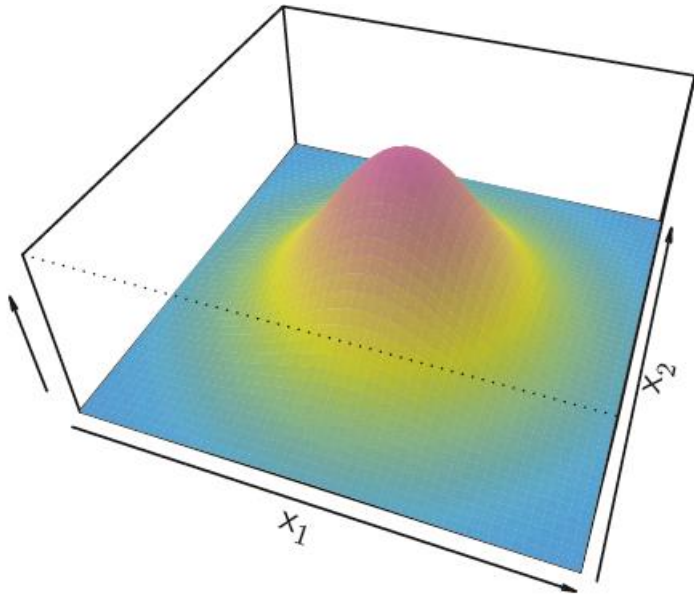
Left: normal density functions



Right: histograms of observations

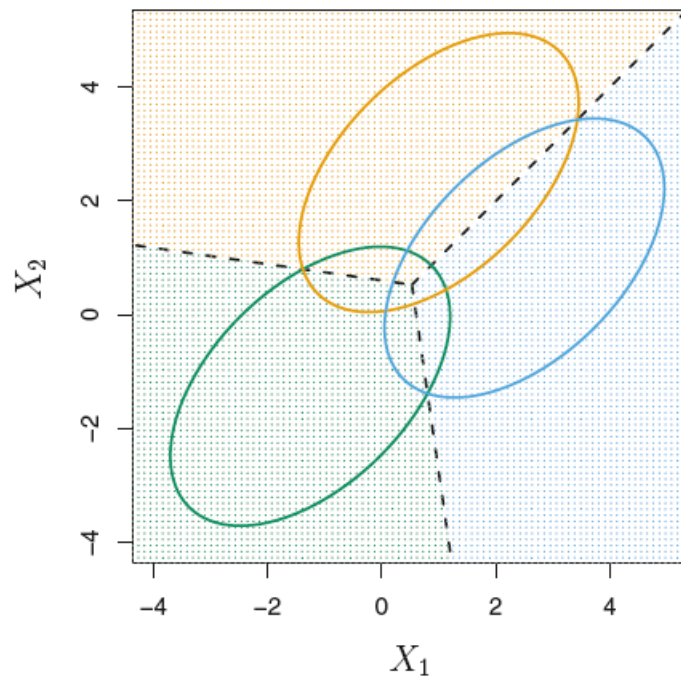
Cases with Multiple Predictors ($p > 1$)

- When there are multiple predictors (i.e., $p > 1$), we use exactly the same approach except that the density function of the predictors is modeled as a multivariate normal density.

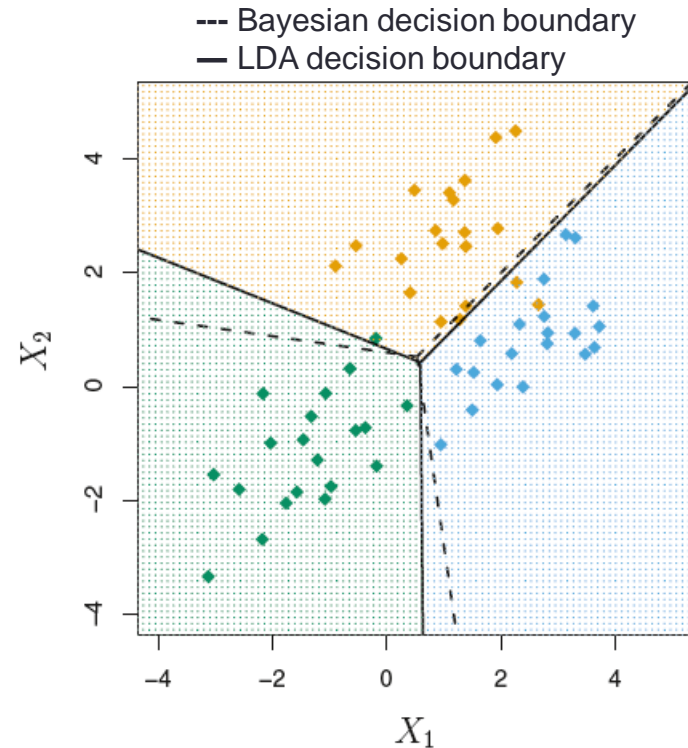


An Example with Two Predictors ($p = 2$)

- Bivariate normal density functions $f_1(\mathbf{x})$, $f_2(\mathbf{x})$ and $f_3(\mathbf{x})$ from three classes



Left: Ellipses that contain 95% of the probability



Right: 20 observations generated from each class

Quadratic Discriminant Analysis (QDA)

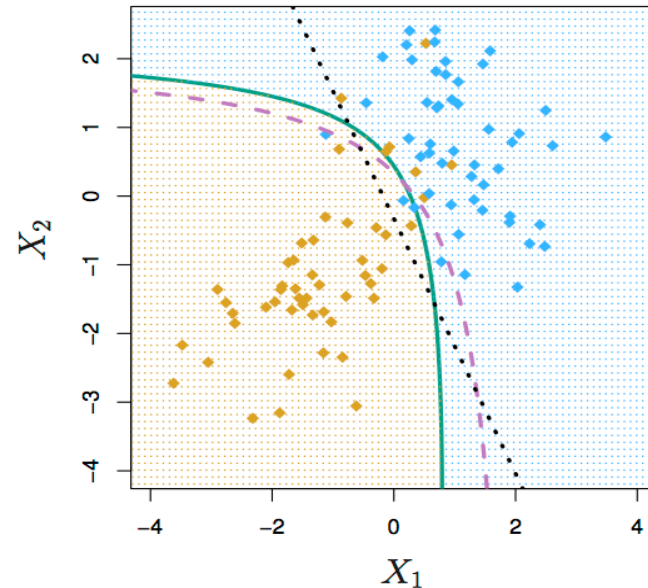
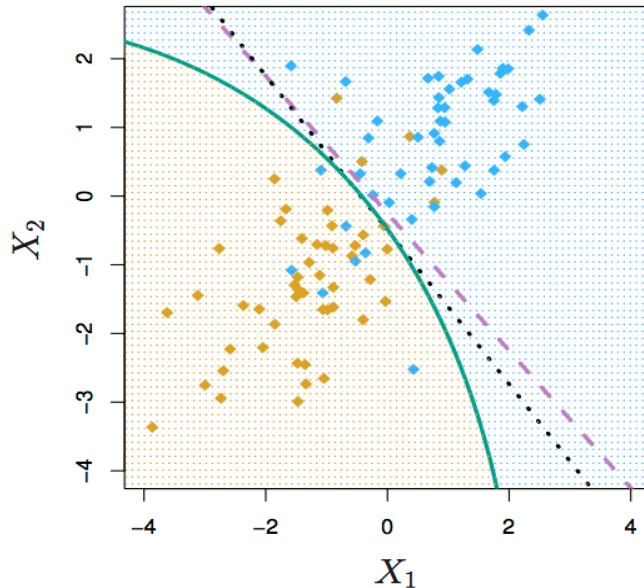
- LDA assumes that all the classes have the same variance (or variance-covariance matrix in the case of multiple predictors).
- LDA may perform poorly if this assumption is far from truth.
- QDA works identically as LDA except that it estimates separate variance (or variance-covariance matrix in the case of multiple predictors) for each class.
- The discriminant function of QDA takes a quadratic form.

Which Is Better?

- Since QDA allows for different variances among classes, the resulting boundaries become quadratic.
- Which approach is better: LDA or QDA?
 - QDA is more flexible than LDA.
 - QDA works best when the variances are very different between classes and we have enough observations to accurately estimate the variances.
 - LDA works best when the variances are similar among classes or we don't have enough data to accurately estimate the variances.

Comparing LDA and QDA

- Two simulated examples with binary response
 - Left: variances of the two classes are equal (LDA is better)
 - Right: variances of the two classes are not equal (QDA is better)



- **Black** dotted: LDA boundary
- **Purple** dashed: Bayes' boundary
- **Green** solid: QDA boundary

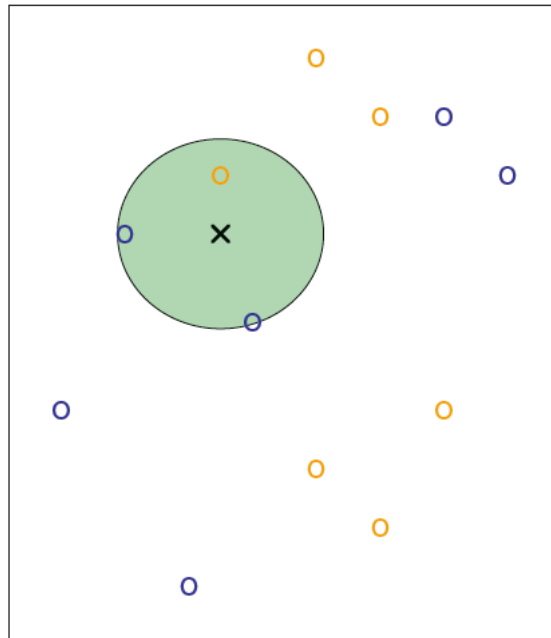
K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) Classifier

- Given a test observation $X = x_0$, KNN works in the following steps to find its predicted class:
- **Step 1:** identify the K points in the training data that are closest to x_0 , i.e., the “ K nearest neighbors”.
- **Step 2:** calculate the fraction of observations belonging to each class among the K points.
- **Step 3:** assign this observation to the class with the largest fraction.

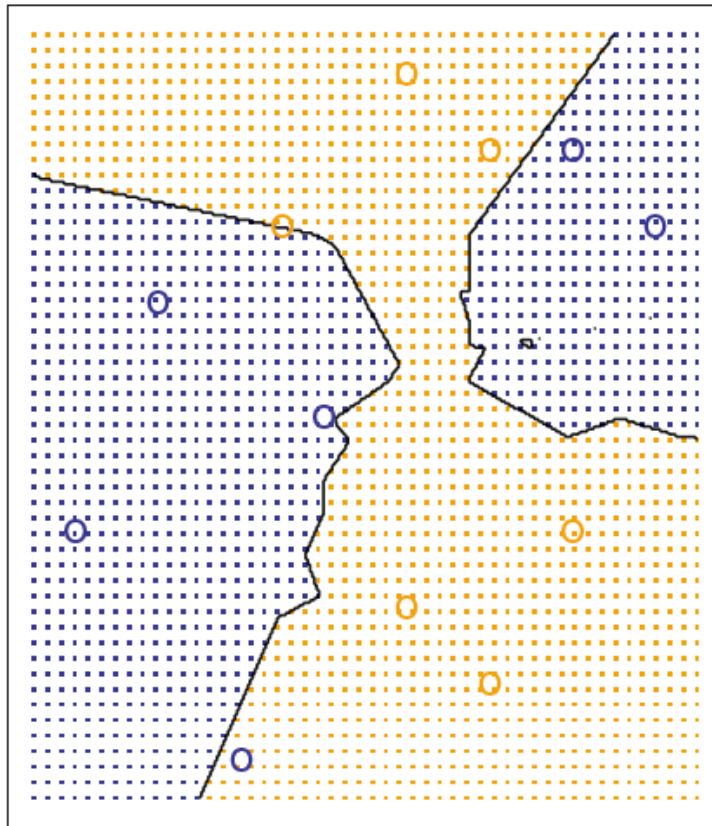
A Simple Example

- Training data set includes 6 blue points and 6 orange points.
- KNN method ($K = 3$):
 - (1) find the 3 nearest neighbors of the test point
 - (2) among the 3 points, 2/3 belong to blue class, 1/3 orange class.
 - (3) the test point is assigned to blue class.



KNN Decision Boundary

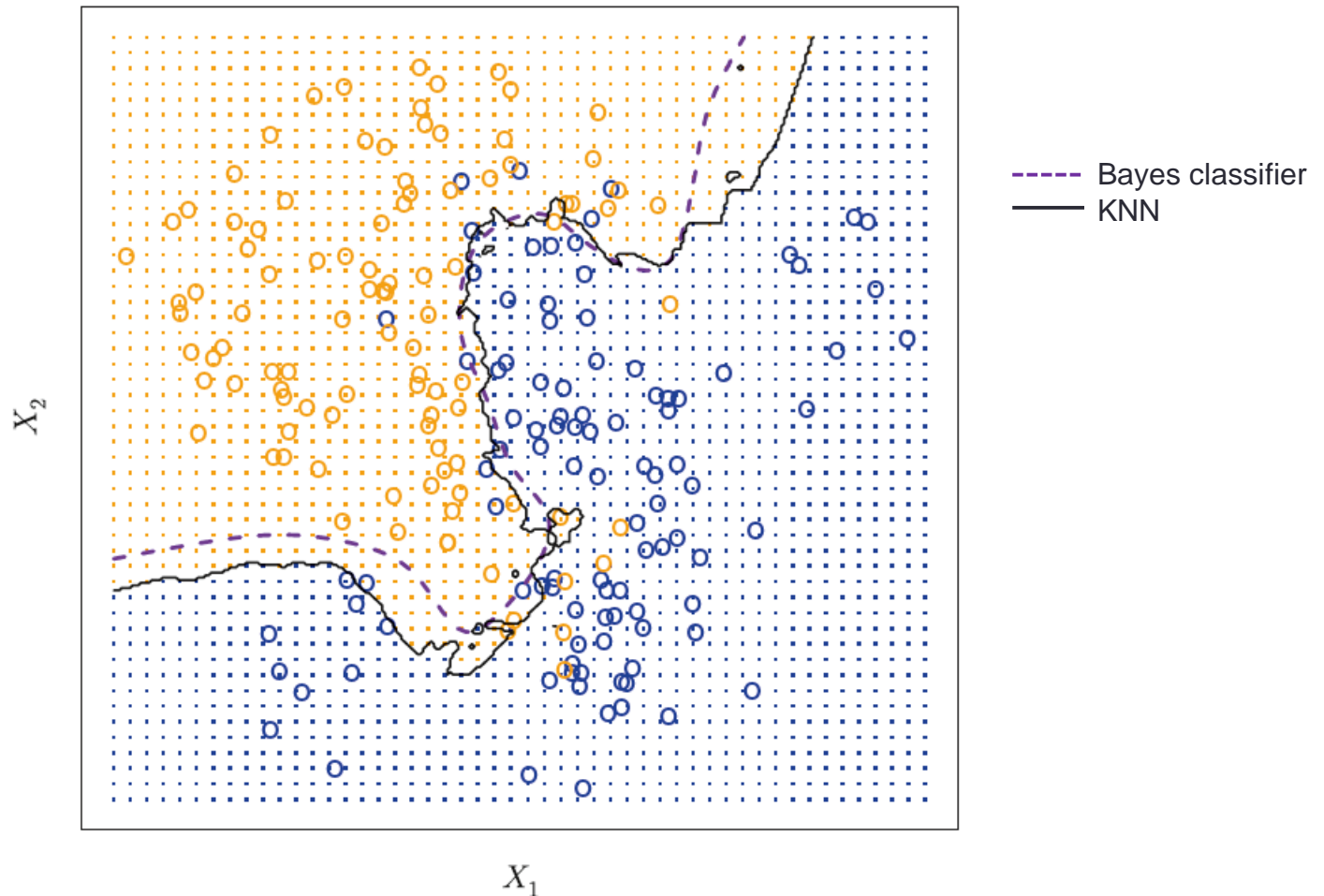
- **Blue** grid: region of the blue class
- **Orange** grid: region of the orange class



KNN Can Work Pretty Well!

- Though KNN is a very simple approach, it can often perform surprisingly well, close to the Bayes classifier.

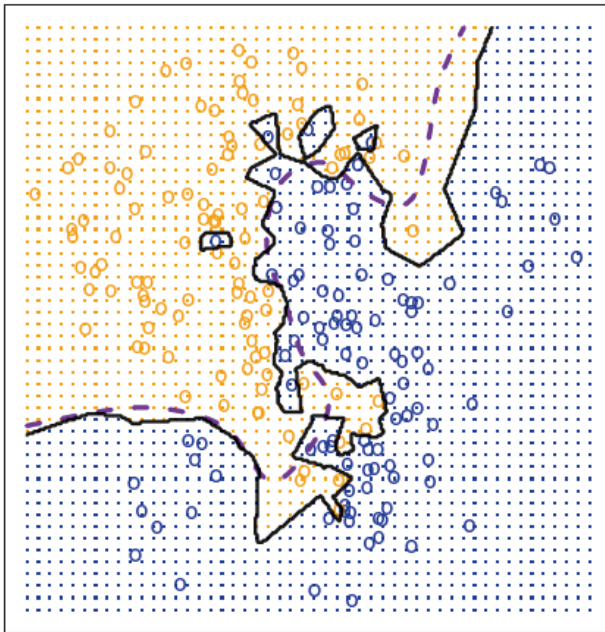
KNN: K=10



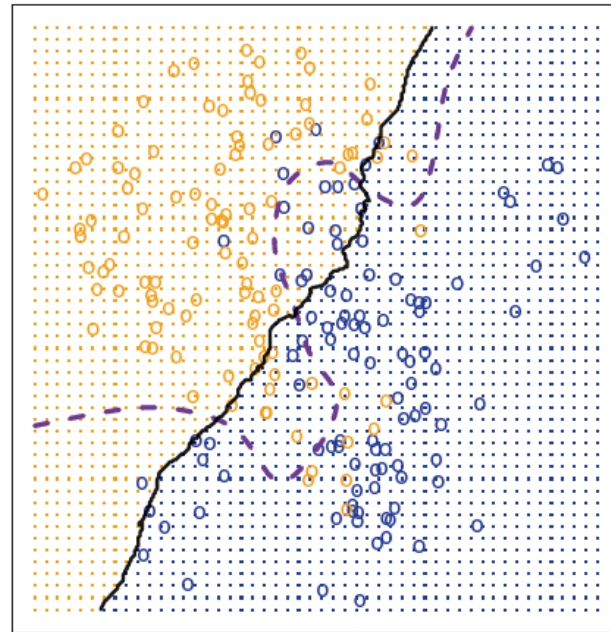
Choice of K

- $K = 1$: decision boundary is flexible
- As K grows, the boundary becomes less flexible and gets close to linear.
- $1/K$ represents the level of flexibility.

KNN: $K=1$

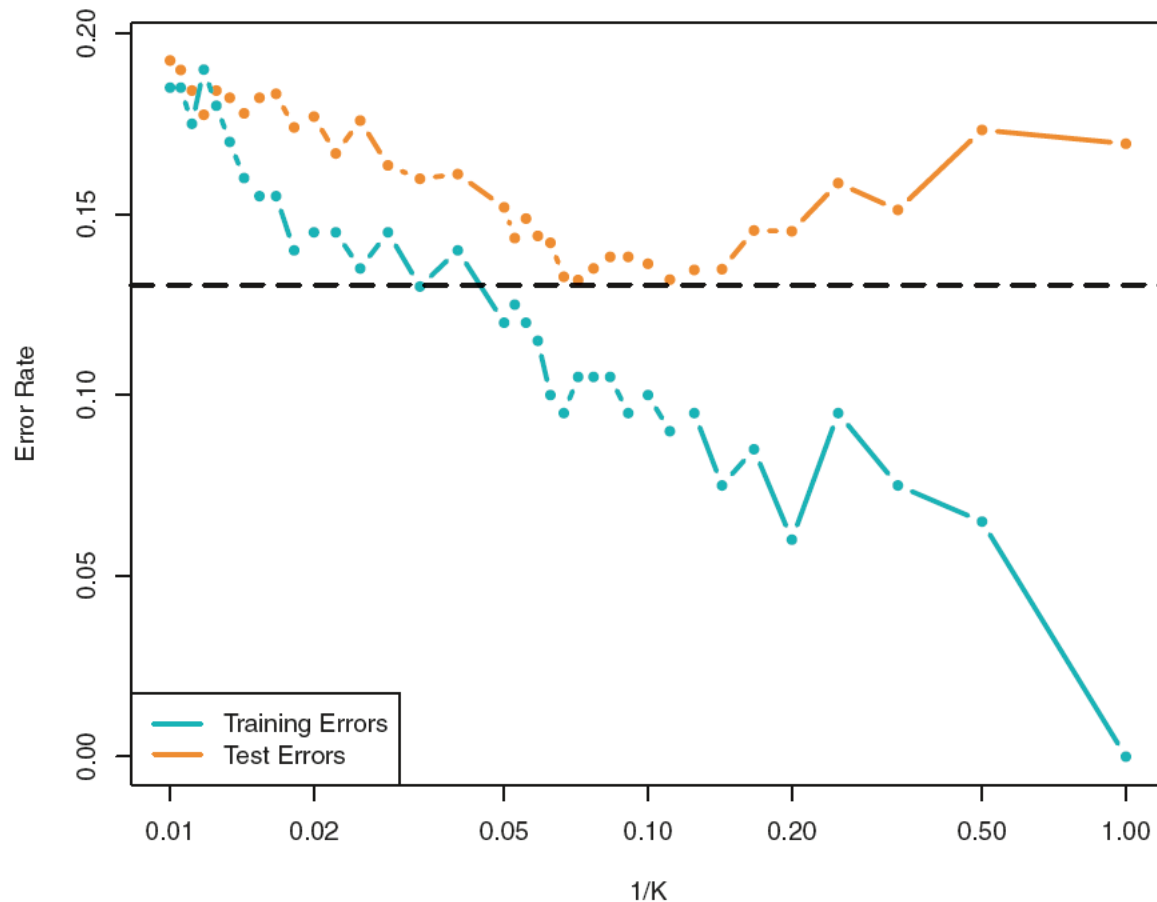


KNN: $K=100$



Training/Test Error vs. Flexibility

- Optimal choice to minimize test error rate: $K = 10$



Performance Assessment and Comparison

Classifiers

- Logistic regression (linear boundary)
- LDA (linear boundary)
- QDA (nonlinear boundary)
- KNN (highly nonlinear boundary)

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 normality assumption holds, otherwise logistic regression can outperform LDA.

KNN vs. (Logistic regression and LDA)

- KNN is completely *non-parametric*: No assumptions are made about the shape of the decision boundary!
- **Advantage of KNN**: We can expect KNN to dominate both LDA and logistic regression when the decision boundary is highly non-linear
- **Disadvantage of KNN**: KNN does not tell us which predictors are important (no table of coefficients)

QDA vs. (Logistic regression, LDA, KNN)

- QDA is a compromise between non-parametric KNN method and the linear LDA and logistic regression

Choosing A Classifier

- If the true decision boundary is:
 - **Linear**: LDA and logistic regression outperform
 - **Moderately non-linear**: QDA outperforms
 - **More complicated**: KNN is superior

Assessing Performance of Classification

➤ **Performance measures for all classifiers**

Accuracy

Error rate

Sensitivity

Specificity

➤ **Performance measures for logistic regression/LDA/QDA**

Receiver operating characteristics (ROC)

Area under curve (AUC)

Confusion Matrix

➤ Example: **Default** data set (training)

		<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

$$\text{Accuracy} = \frac{(\quad)}{(\quad)}$$

$$\text{Error rate} = \frac{(\quad)}{(\quad)}$$

$$\text{Sensitivity} = \frac{(\quad)}{(\quad)}$$

$$\text{Specificity} = \frac{(\quad)}{(\quad)}$$

Performance Measures

$$\text{True positive rate} = \frac{TP}{P}$$

$$\text{False positive rate} = \frac{FP}{N}$$

$$\text{Accuracy} = \frac{(TN + TP)}{N + P}$$

$$\text{Error rate} = \frac{(FN + FP)}{N + P}$$

$$\text{Sensitivity} = \frac{TP}{P}$$

$$\text{Specificity} = \frac{TN}{N}$$

		<i>Predicted class</i>		
		– or Null	+ or Non-null	Total
<i>True class</i>	– or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
Total		N*	P*	

$$\text{Error rate} = 1 - \text{Accuracy}$$

$$\text{Sensitivity} = \text{True positive rate}$$

$$\text{Specificity} = 1 - \text{False positive rate}$$

Interpretation

		<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

- **Accuracy**: fraction of people that are correctly classified (97.25%).
- **(Training) Error rate**: fraction of people that are incorrectly classified (2.75%)
- **Sensitivity (true positive rate)**: fraction of defaulters that are correctly identified (24.3%)
- **Specificity**: fraction of non-defaulters that are correctly identified as non-defaulters (99.76%)
- **False positive rate** ($1 - \text{Specificity}$): fraction of non-defaulters that are incorrectly classified as defaulters (0.24%)

Threshold in Class Prediction

- Recall that logistic regression and LDA/QDA produce a probability estimate for each observation, and then a threshold is used to determine its predicted class.
- Usually 0.5 is used as threshold.
- If we use a different value for the threshold, the performance of classification will be different.

Example

- Use LDA for the **Default** data set
- Threshold = 0.5: the training error rate is 2.75%, but the sensitivity is only 24.3%.
- Threshold = 0.2: the training error rate is 3.73%, while the sensitivity increases to 58.6%.

Threshold = 0.5

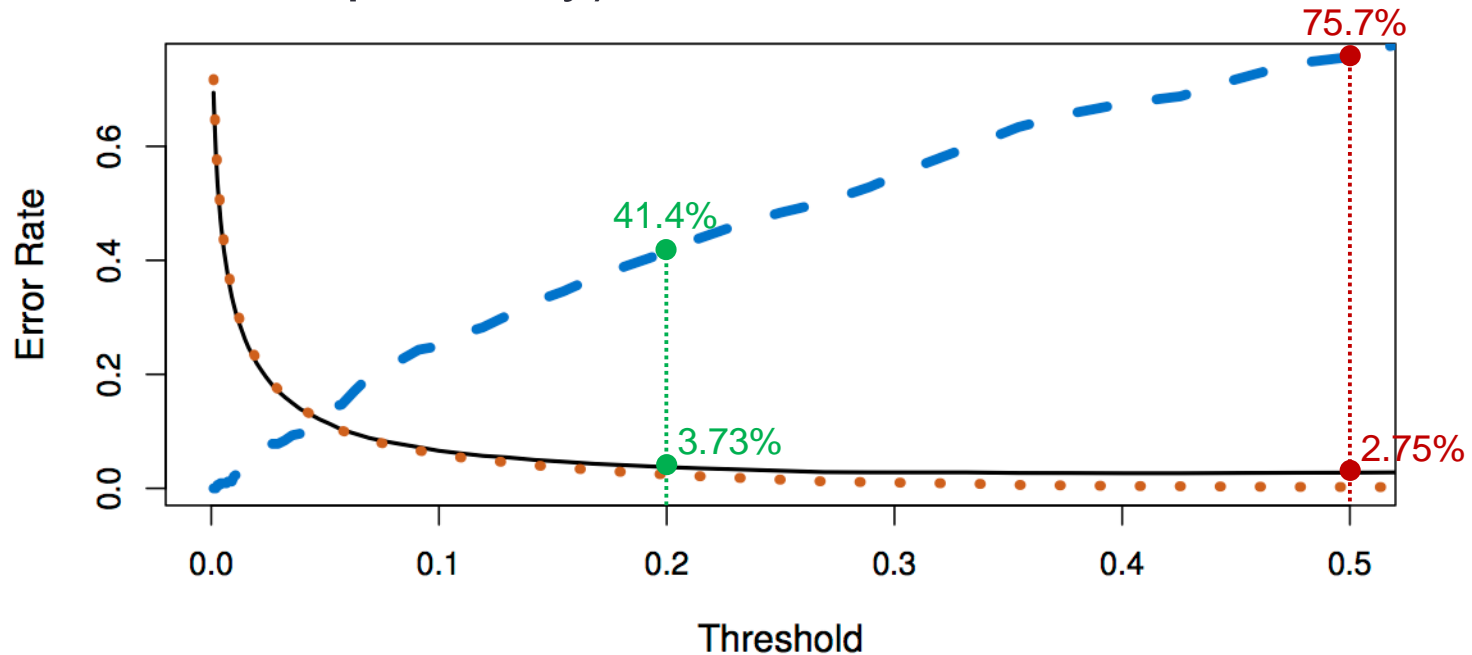
		<i>True Default Status</i>		
		No	Yes	Total
<i>Predicted Default Status</i>	No	9644	252	9896
	Yes	23	81	104
Total		9667	333	10000

Threshold = 0.2

		<i>True Default Status</i>		
		No	Yes	Total
<i>Predicted Default Status</i>	No	9432	138	9570
	Yes	235	195	430
Total		9667	333	10000

Threshold Values vs. Error Rates

- **Black solid:** overall error rate
- **Blue dashed:** fraction of defaulters missed ($1 - \text{Sensitivity}$)
- **Orange dotted:** non-defaulters incorrectly classified (False positive rate, $1 - \text{Specificity}$)



- Decide the threshold based on domain knowledge, such as detailed information about the cost associated with default.

Receiver Operating Characteristics (ROC)

- **Ideal**: top left corner.
- **Diagonal**: “no information” classifier or random guessing
- **Overall performance**: area under the curve (AUC)

