Classification I

Statistical Decision Theory (from last lecture)

- $X \in R^p$ – random input vector and $Y \in R$ – random output variable have joint distribution $\Pr(X, Y)$
- $f(X)$ – a function we are looking for to use to predict Y
- $L(Y, f(X))$ – a loss function used to penalize for the error of prediction
- Common choice is the squared loss function

$$L\big(Y, f(X)\big) = (Y - f(X))^2$$

- We choose f according to the expected (squared) prediction error (EPE)

$$EPE(f) = E(Y - f(X))^2 = \int [y - f(x)]^2 \Pr(dx, dy)$$

- Then applying the iterated expectation formula (conditioning on X)

$$EPE(f) = E_X E_{Y|X}([Y - f(X)]^2 | X)$$

which is can be minimized w.r.t. f pointwise

$$f(x) = \operatorname{argmin}_c E_{Y|X}([Y - c]^2 | X = x)$$

- The solution is the conditional expectation also called the regression function

$$f(x) = E(Y|X = x)$$

- How do we modify the above when the output is a categorical variable G?

- Denote by $\hat{G}$ an estimate of G and by $\mathcal{G}$ the set of possible values or classes. Assume that there are J such classes.

- We are going to use a different loss function L - a $J \times J$ matrix $\boldsymbol{L}$ with elements $L(j, l) \geq 0$ representing the penalty for wrongly classifying an observation belonging to class $\mathcal{G}_j$ as $\mathcal{G}_l$. Then $L(j, j) = 0$ (zero on the diagonal) in case of a correct classification

- Most often the 0-1 loss function is used: $L(j, l) = 1$ if $j \neq l$

- The EPE (expected prediction error)

$$EPE = E[L(G, \hat{G}(X))]$$

- Using joint distribution $\Pr(G, X)$ and conditioning we can write EPE as

$$EPE = E_X \sum_{j=1}^{J} L[\mathcal{G}_j, \hat{G}(X)] \Pr(\mathcal{G}_j|X)$$

which is minimized pointwise:

$$\hat{G}(x) = \ argmin_{g\in\mathcal{G}} \sum_{j=1}^{J} L[\mathcal{G}_j, g] \Pr(\mathcal{G}_j|X = x)$$

With the 0-1 loss function and $g = \mathcal{G}_k$ for some k, the above sum is

$$\sum_{j=1}^{J} L[\mathcal{G}_j, \mathcal{G}_k] \Pr(\mathcal{G}_j|X = x) = \sum_{j=1, j\neq k}^{J} L[\mathcal{G}_j, \mathcal{G}_k] \Pr(\mathcal{G}_j|X = x) + L[\mathcal{G}_k, \mathcal{G}_k] \Pr(\mathcal{G}_k|X = x)$$

$$= \sum_{j=1, j\neq k}^{J} 1 \times \Pr(\mathcal{G}_j|X = x) + 0 \times \Pr(\mathcal{G}_k|X = x) = 1 - \Pr(\mathcal{G}_k|X = x)$$

Then the optimal choice for the class g given x is determined from:

$$\hat{G}(x) = argmin_{g \in \mathcal{G}}[1 - \Pr(g|X = x)] = argmax_{g \in \mathcal{G}} \Pr(g|X = x)$$

$$\Longrightarrow \hat{G}(x) = \mathcal{G}_j \text{ if } \Pr\left(\mathcal{G}_j \middle| X = x\right) = max_{g \in \mathcal{G}} \Pr(g|X = x)$$

- This simply means that we classify to the most probable class, using the conditional distribution $\Pr(G|X)$. This is called the Bayes classifier and its error rate is called the Bayes rate.

- This links directly to the k-nearest neighbor method, since the “majority vote" approach taken in implementations approximates the max probability

Next, we will consider the following methods for classification

- K-nearest neighbor (KNN)
- Logistic Regression
  - Why Not Linear Regression?
  - Simple Logistic Regression
    - Logistic Function
    - Interpreting the coefficients
    - Making Predictions
    - Adding Qualitative Predictors
  - Multiple Logistic Regression
- Linear and Quadratic Discriminant Analysis (LDA and QDA); KNN

KNN method

- The KNN method directly implements the above recipe. Assume for simplicity that we have a 2-class problem and use a binary variable Y to dummy-code G: $Y = 1$ if $G = \mathcal{G}_1$ (and $Y = 0$ otherwise).

- At each point x_0 we predict the outcome using the squared error loss

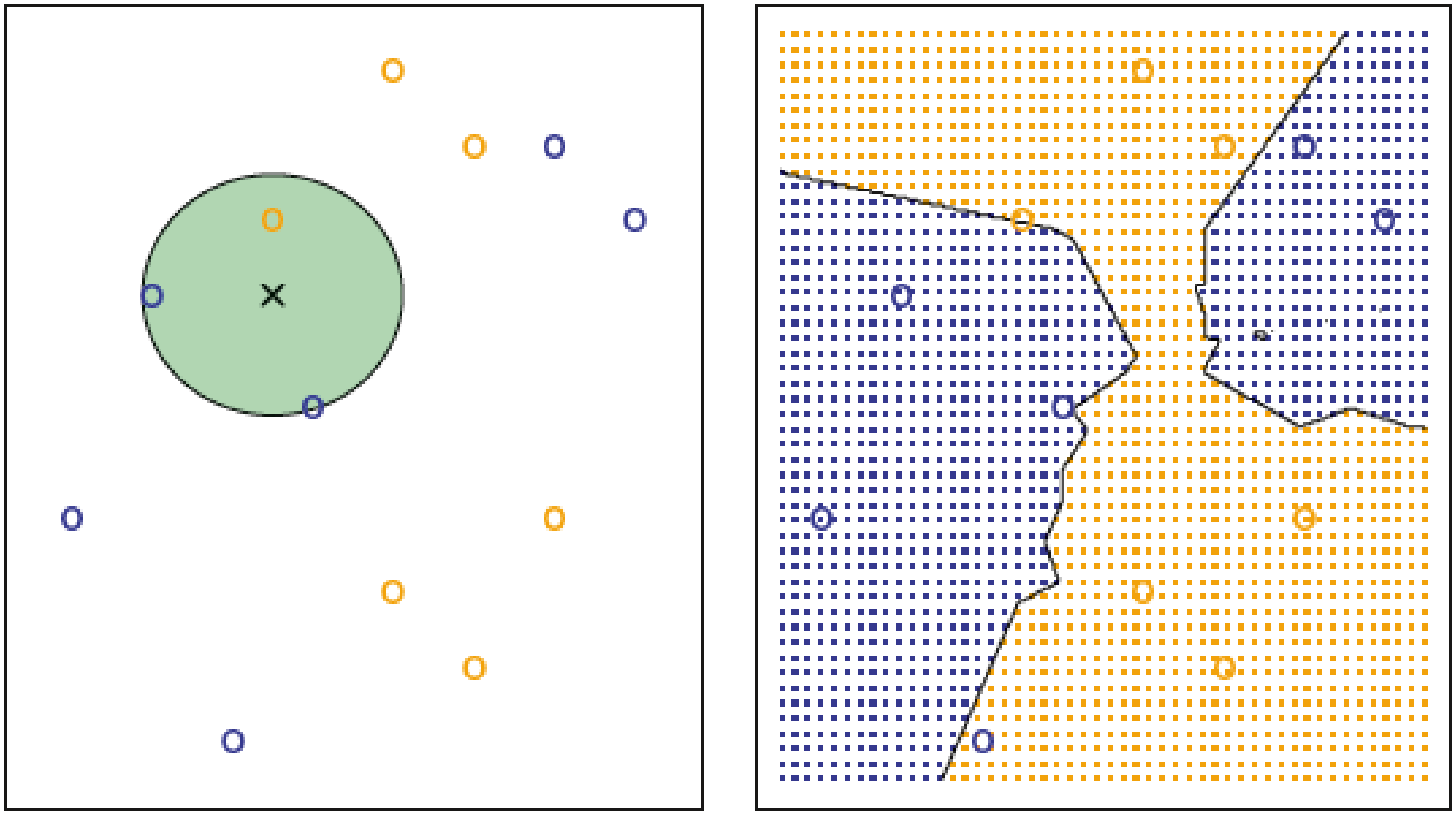
$$\hat{f}(x_0) = E(Y|X = x_0) = 0\times \Pr(Y = 0|\, X = x_0) + 1\times \Pr(Y = 1|\, X = x_0)$$

$$= \Pr(Y = 1|\, X = x_0) = \text{ Ave } (y_i|x_i \in N_K(x_0)) = \frac{1}{K} \sum_{i:x_i \in N_K(x_0)} I(y_i = 1)$$

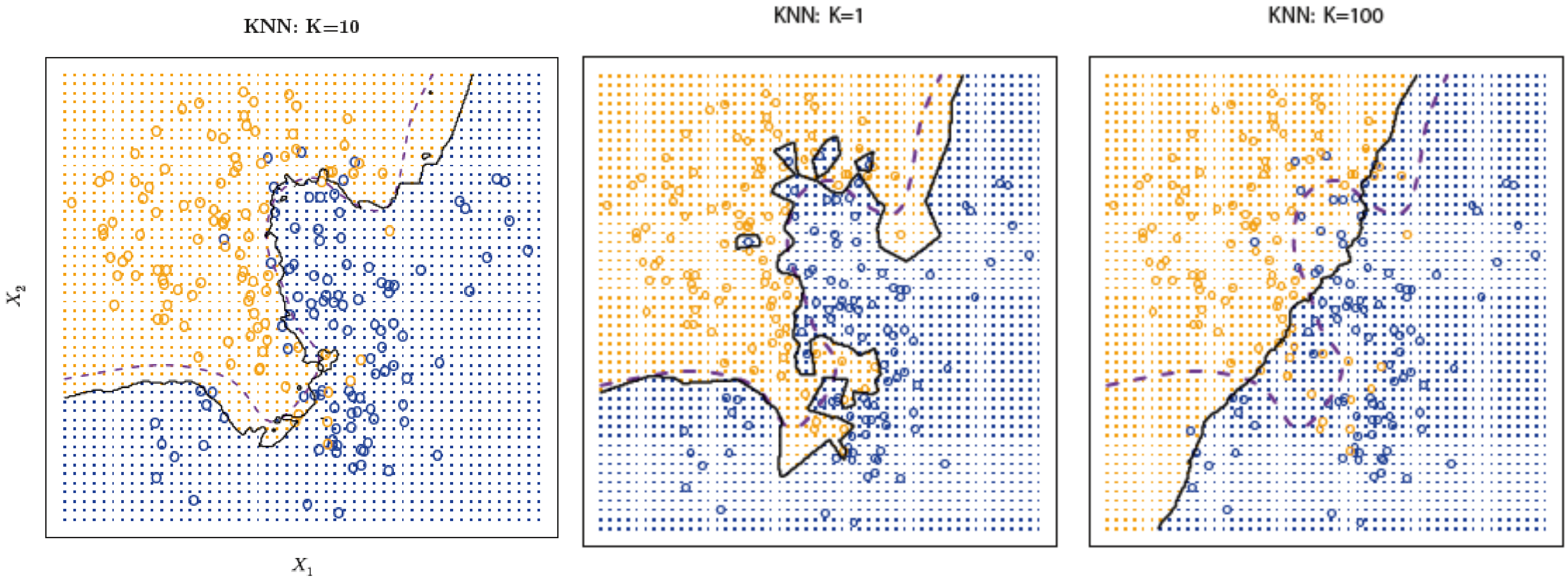
- This is the fraction of the points in $N_K(x_0)$ whose response equals 1. Then if $\hat{f}(x_0) > 0.5$ (majority) we predict class 1 for the ith observation.

- This way of classification approximates the optimal Bayes classifier

Example how KNN works for $p = 2$ and $K = 3$



The black lines are the decision boundaries.



In purple dashed line – the Bayesian decision boundary, in **black** - the KNN decision boundary for different K.

$K = 10$ seems to be the best of choice amongst $K = 1, 10$ and 100.

Logistic Regression

Let's review Logistic regression using a data set

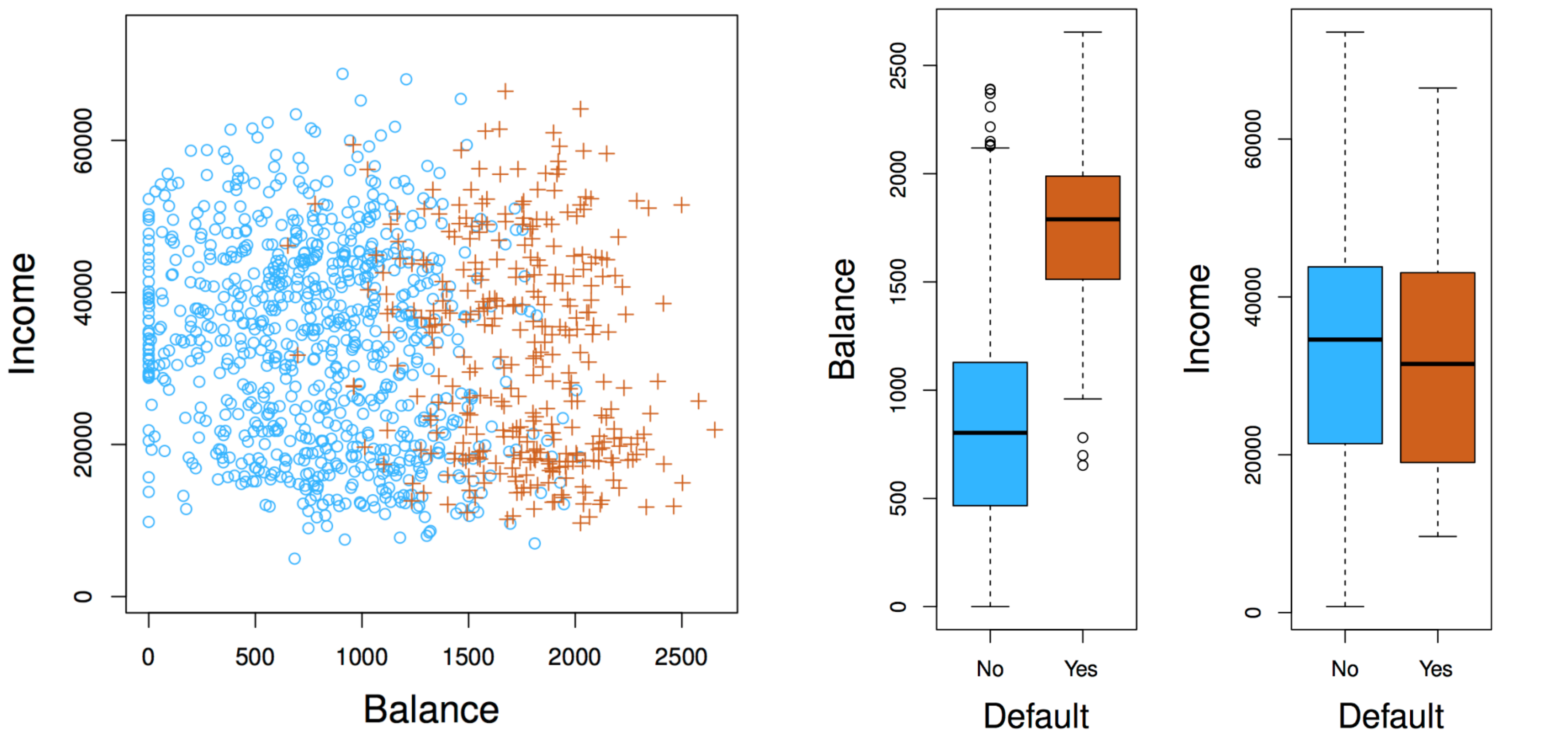
Example. The data set `Default` is simulated

- We would like to be able to predict customers that are likely to default
- Possible X variables are
    - Monthly credit card `balance` (X_1)
    - Annual `income` (X_2)
    - The Y variable (Default) is categorical: `Yes` or `No`

How do we check the relationship between Y and X?

Let's plot it first.

The `Default` dataset. Here, the defaulted cases are plotted in orange



- Why not use Linear Regression?

- For simplicity, consider only 1 predictor (`balance`) and code Y to be 1 for default and 0 otherwise

- The regression line $Y = \beta_0 + \beta_1 X$ fit by Least Squares will produce the estimate $\hat{\beta}_0 + \hat{\beta}_1 X$ which can be shown to be an approximation of the probability of default

$$p(X) = \Pr(Y = 1|X)$$

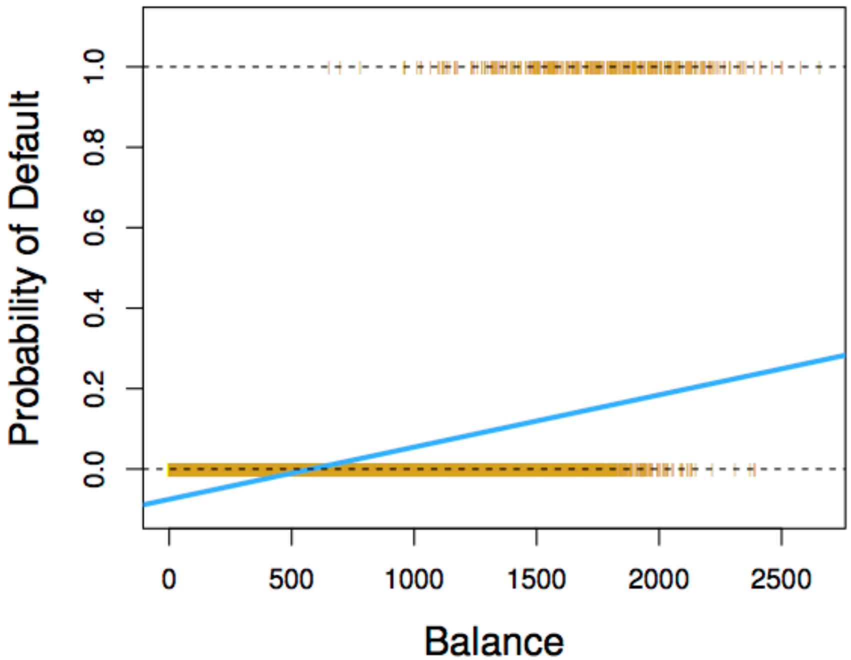
- So essentially we are fitting the linear regression

$$p(X) = \beta_0 + \beta_1 X$$

- The problem with this approach is that the left hand side is a probability which must be in $[0, 1]$ but the right hand side can be any value between negative and positive infinity

- This is demonstrated with the `Default` data on the right

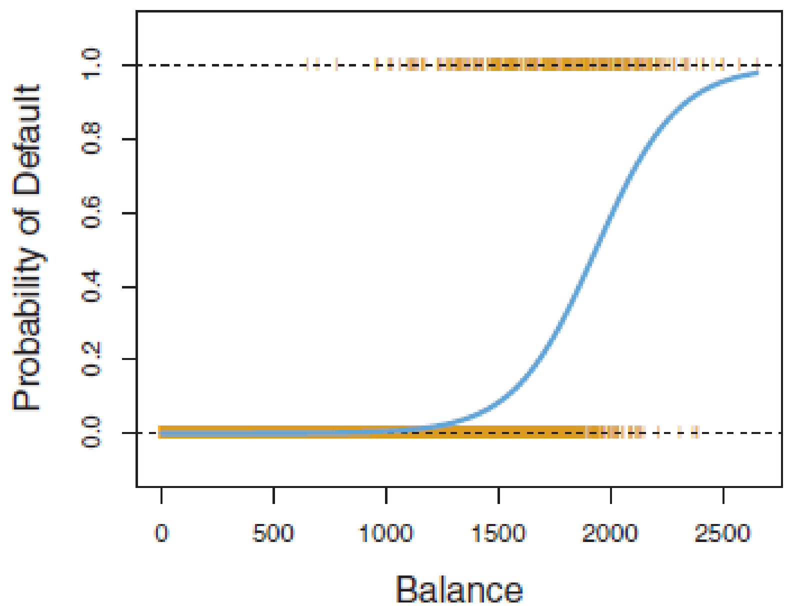
- The linear regression line (in blue) shows that for very low balances we predict a negative probability, and for high balances we predict a probability above 1!



- So, we are going to use the Logistic function

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

- It has the correct properties – the probabilities are between 0 and 1
- The Logistic function has the typical sigmoid shape shown in the logistic regression fit to the `Default` data below



- Another useful representation of the above is the odds ratio

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

and the log-odds ratio (or logit) which is linear in X

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

These have useful interpretation in medicine or betting

- For example, if $p(X) = 0.75$ then $o(X) \equiv p(x)(1 - p(X)) = 0.75/0.25 = 3$. In other words, a success is three times as likely as a failure, and we expect about three successes for every one failure. In terms of payoffs for bets, this situation is called 3-1 on bet and would pay only $\$1$ for every $\$3$ bet.

- On the other side if $p = 0.25$ (expect one success for every three failures) then $o = 0.25/0.75 = 1/3$. In this situation (3-1 against bet), the payoff is $\$3$ for every $\$1$ bet. This is because riskier bets payoff more if winning. It is clear also that

$$p(X) = o(X)/(1 + o(X)).$$

- The coefficients are estimated using the Maximum Likelihood Principle. The estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ maximize the likelihood function

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

  Thus, the parameter estimation provides an estimate for the probability of default which is close to 1 for individuals who defaulted and close to 0 for those who didn't

- Similar questions about significance, population values etc. as in linear regression, arise here too

- Before discussing that let's look at coefficient interpretation

Interpreting the coefficients

- Interpreting what β_1 means is not very easy with logistic regression, simply because we are predicting $Pr(Y)$ and not Y.
- If $\beta_1 = 0$, this means that there is no relationship between Y and X
- If $\beta_1 > 0 \implies$ when X gets larger so does the probability that $Y = 1$
- If $\beta_1 < 0 \implies$ when X gets larger, the probability that $Y = 1$ gets smaller
- Another way to say it is that a unit increase in X increases the log-odds of success by β_1 or increases the odds of success by a factor of $exp(\beta_1)$

Are the coefficients significant?

- We still want to perform a hypothesis test to see whether we can be sure that are β_0 and β_1 significantly different from zero

- We use a z-test instead of a t-test, but of course that doesn't change the way we interpret the p-value

- Here the p-value for `balance` is very small, and β_1 is positive, so we are sure that if the balance increase, then the probability of default will increase as well.

	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

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

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+\ e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.651+0.005\times 1000}}{1+\ e^{-10.651+0.005\times 1000}} = 0.0058$$

- 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 in Logistic Regression

- We can predict if some individual defaults by checking if she is a student or not. Thus, we can use a qualitative variable `Student` coded as (Student = 1, Non-student =0)

- β_1>0: This indicates students tend to have higher default probabilities than non-students

	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

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

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

Multiple Logistic Regression

We can fit multiple logistic regression just like regular regression. We will have

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

and the probability for $Y = 1$ will be

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}}{1+ e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}}$$

Multiple Logistic Regression (`Default` Data)

- Predict Default using:
  - Balance (quantitative)
  - Income (quantitative)
  - 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

Predictions

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

$$\hat{p}(X) = \frac{e^{-10.869+0.00574\times 1500+0.003\times 40-0.6468\times 1}}{1+e^{-10.869+0.00574\times 1500+0.003\times 40-0.6468\times 1}} = 0.058.$$

Comparing to the regression on the single `Student` variable…

We get a strange result

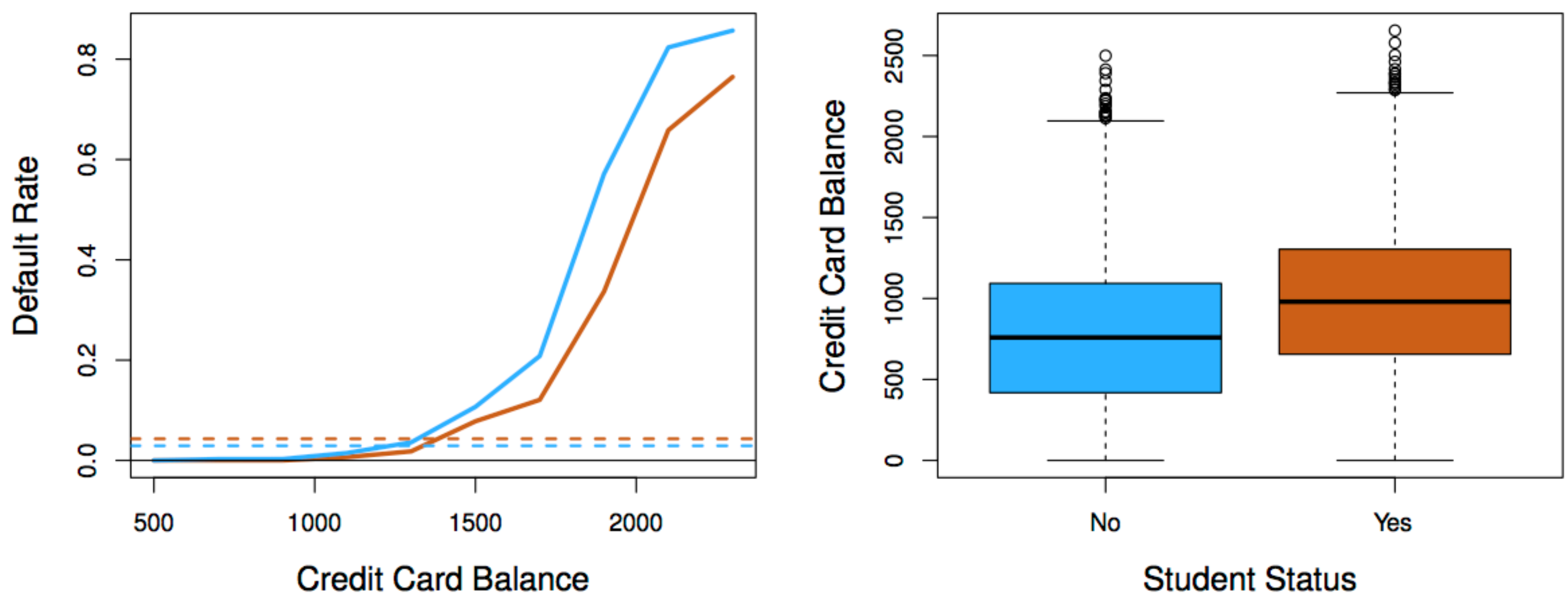
	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

Positive

	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

Negative

Students (orange) vs. Non-Students (blue)



Who should we offer credit to?

- A student is risker than non-students if no information about the credit card balance is available
- However, a student is less risky than a non-student with the same credit card balance!

Logistic Regression for more than 2 Response Classes

- We sometimes wish to classify a response variable that has more than two classes. For example, we could have three categories of medical condition in the emergency room: stroke, drug overdose, epileptic seizure

- In this setting, we wish to model all 3 probabilities

$Pr(Y = \text{stroke}|X)$,
$Pr(Y = \text{drug overdose}|X)$,
$Pr(Y = \text{epileptic seizure}|X) = 1 - Pr(Y = \text{stroke}|X) - Pr(Y = \text{drug overdose}|X)$.

- The two-class logistic regression model has multiple-class extensions, but in practice they tend not to be used all that often

- One of the reasons is that the LDA we'll discuss next, is popular for multiple-class classification

Part1: Appendix

- The Receiver Operator Characteristic (ROC) curve and the associated Area Under the Curve (AUC) is a widely used method to judge the predictive power of a classification model. Another measure is the miss-classification error.

- Given that the model (called also classifier) produces a probability p of the observation to be from the "positive" $(= 1)$ of two possible classes $(0/1)$, a prediction $(0$ or $1)$ can be given for each cut-off value for p

- For the Default data (training) in the R code, the classification (output) variable is the `Default` (Yes=1 for default, No=0 otherwise)

- We can build a logistic regression model on the train portion of the data and then produce the ROC curve and 2 confusion matrices (contingency tables). In the first one we predict class 1 for an observation if $p > 0.5$ and in the second table, we predict class 1 if $p > 0.2$

$p > 0.5$

		Predicted class 1	Predicted class 0	
Actual class	1	100	233	333
	0	42	9625	9667
				10000

$p > 0.2$

		Predicted class 1	Predicted class 0	
Actual class	1	199	134	333
	0	263	9404	9667
				10000

- From the first table the (misclassification) error is [(42+233)]/10000 = 2.75% and for the second cut-off, the error rate is: 397/10000 = 3.97%

- As is evident from this example, the cut-off value influences critically the prediction accuracy

- The error is a direct function of the correct classifications counts – the numbers on a blue background, and called True Positive (TP) and True Negative (TN)

- The most frequently used definitions are summarized below

		Predicted class		
		1	0	
Actual class	1	True Positive	False Negative	P
	0	False Positive	True Negative	N

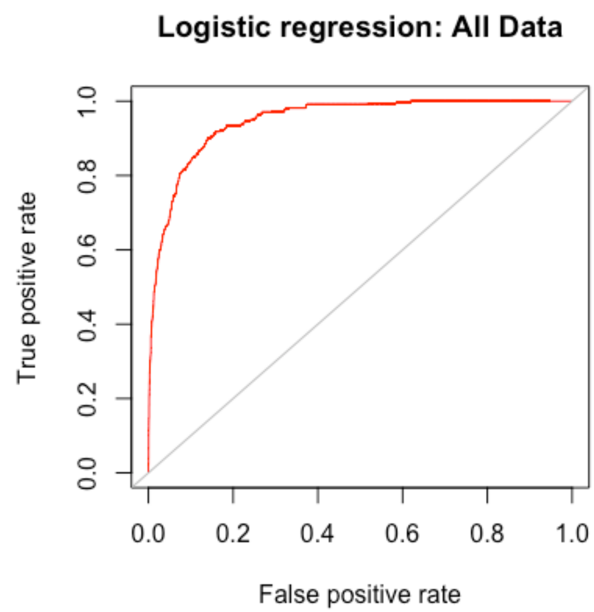
P = Total "1" (positive) cases
N = Total "0" (negative)

TPR = TP/P = Recall = Sensitivity
FPR = FP/N = 1-Specificity
Accuracy = (TP+TN)/(P+N)
Precision = TP/(TP+FP)

- The dependence of the Sensitivity (TPR) and Specificity (1-FPR) on the probability cut-off are best depicted through the ROC curve

- The area under the curve (AUC), called also c-statistic, is between 0 and 1 and best reflects how good the model is. The closer to 1 it is the better the model is. Values in the range 0.75-0.95 are considered good to excellent.

- For random decision on the class of an observation ("flipping a coin") AUC=0.5 and the ROC curve for it is the diagonal plotted below
- The ROC curve for the model (train data) is plotted in red and the corresponding AUC=0.9479



For more:
http://www.hpl.hp.com/techreports/2003/HPL-2003-4.pdf
https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Reading: **ISLR**: Chapter 4.1-4.3 for Part 1 (today)