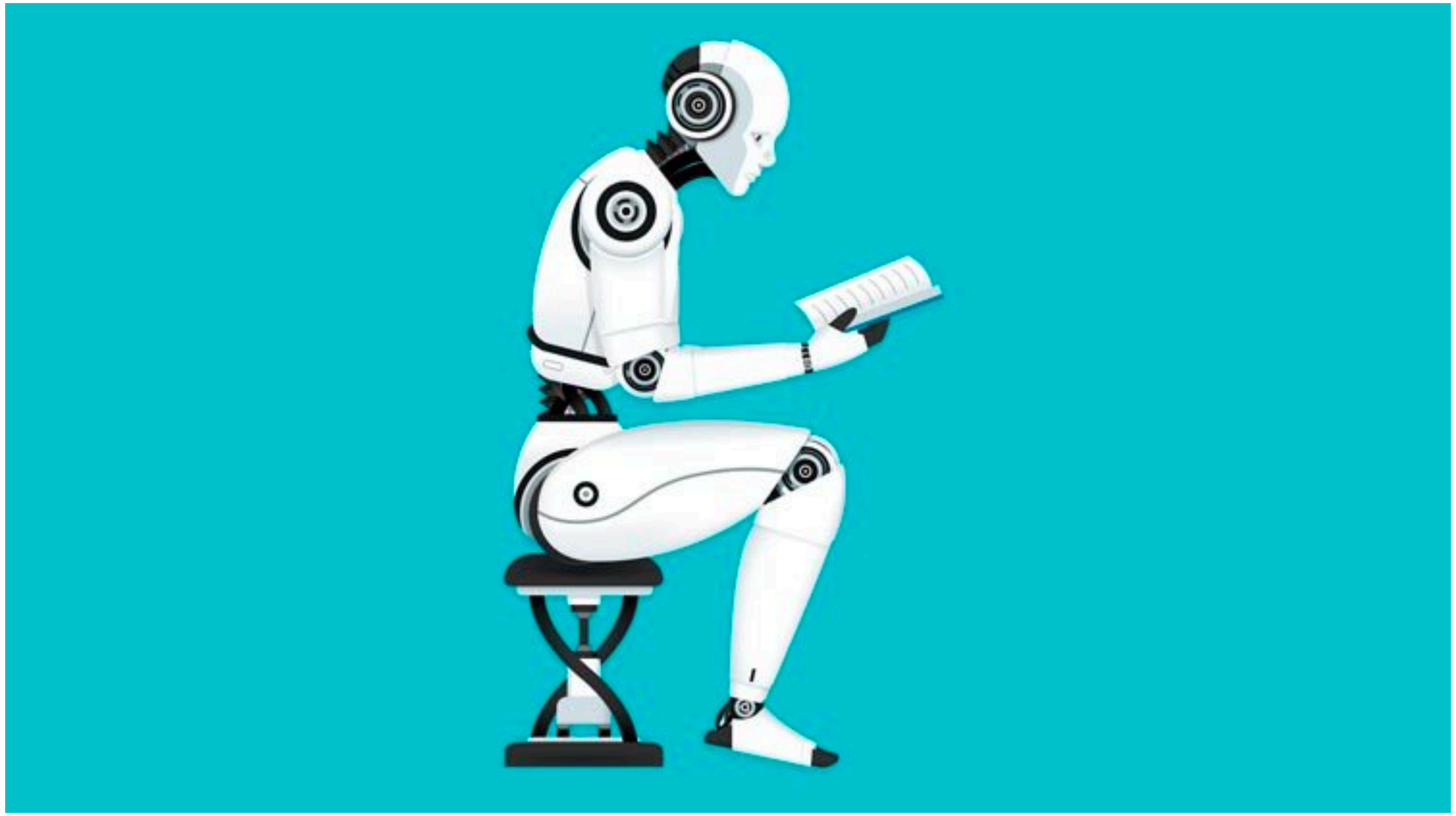
UNIVERSITAT DE BARCELONA



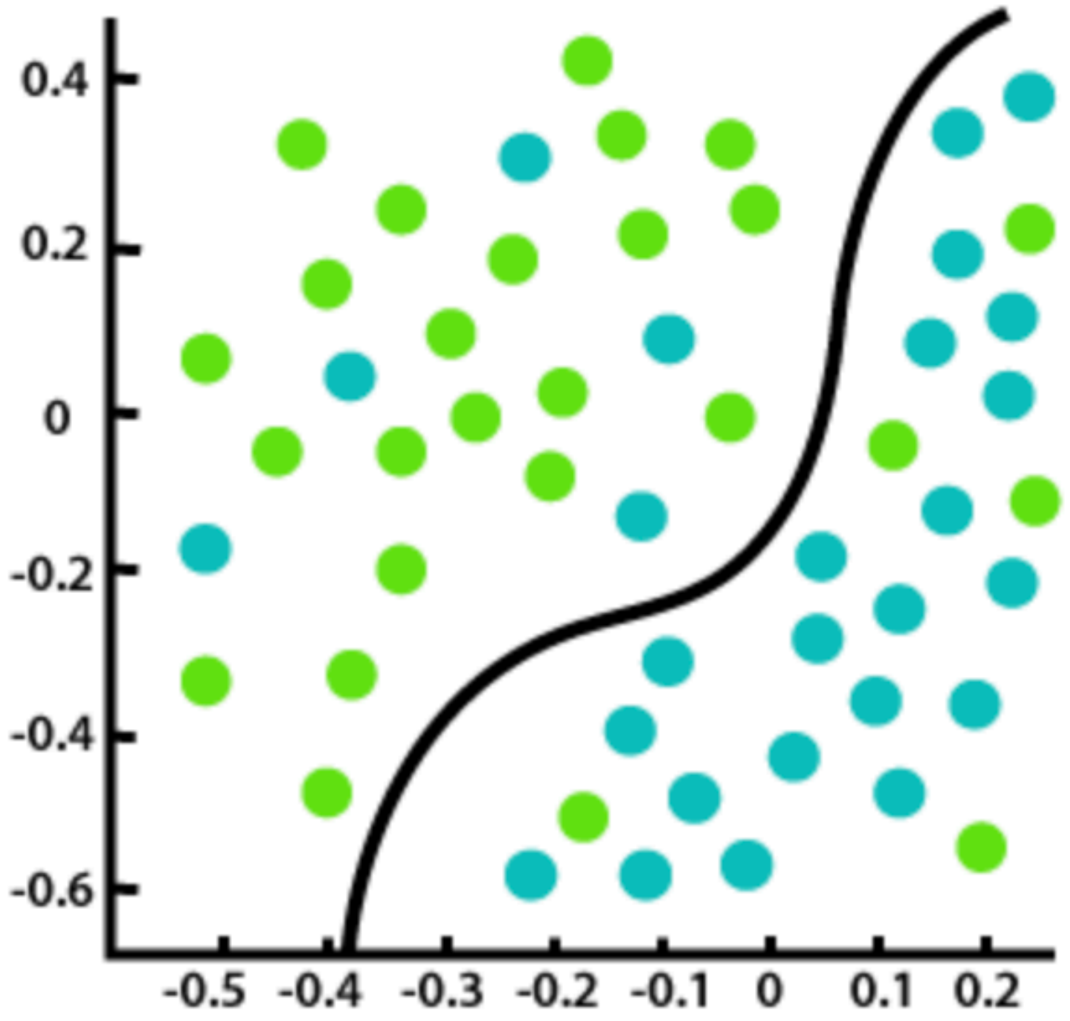
Classification

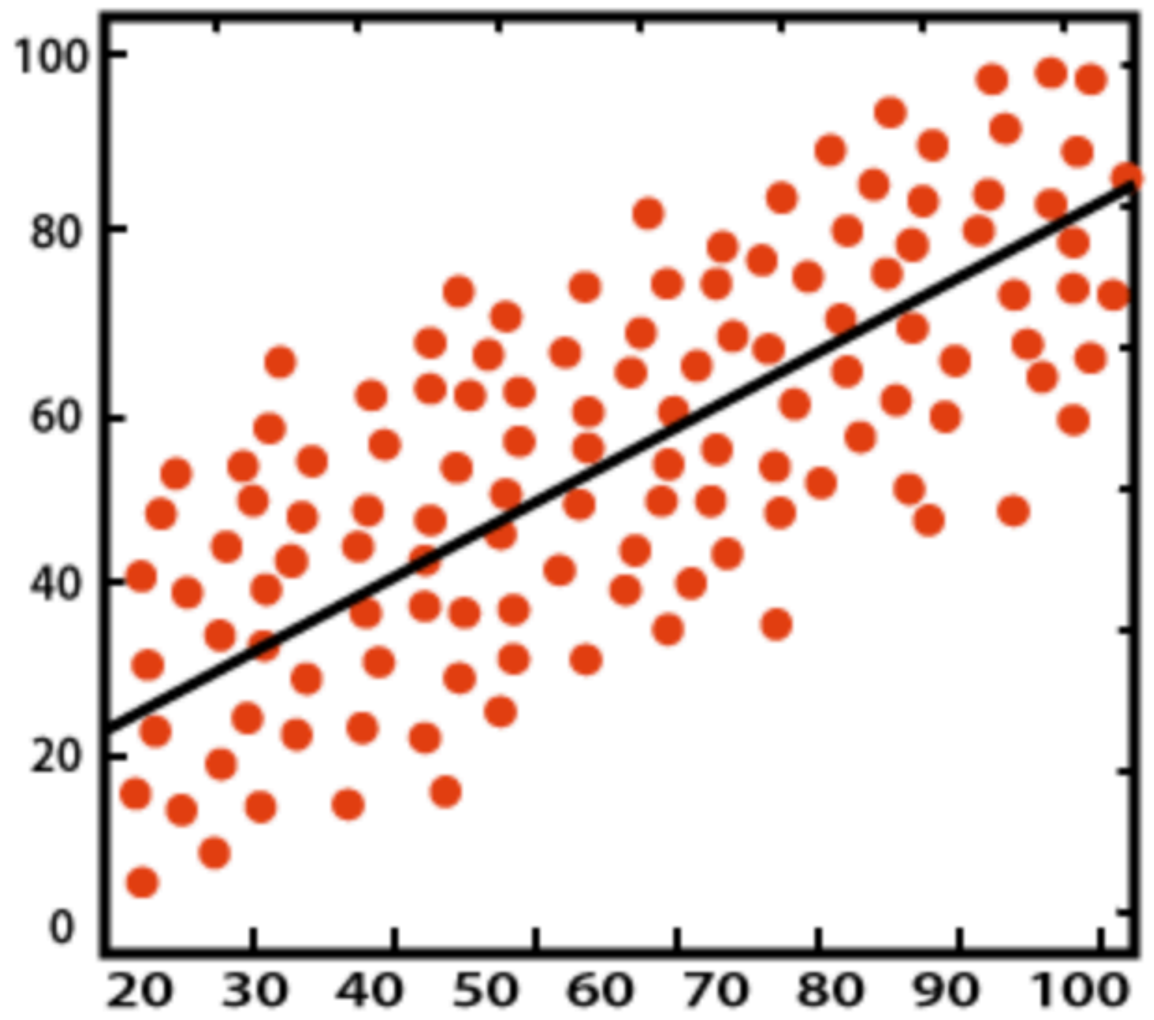
Machine Learning | Enginyeria Informàtica

Santi Seguí | **2022-2023**

Classification

What happens when the response is qualitative (also named categorical)?





Types of Classification Tasks

Binary Classification

Spam
Ham

MultiClass Classification

Cat
Dog
Horse
Person
Fish
Bird
...

Multilabel Classification

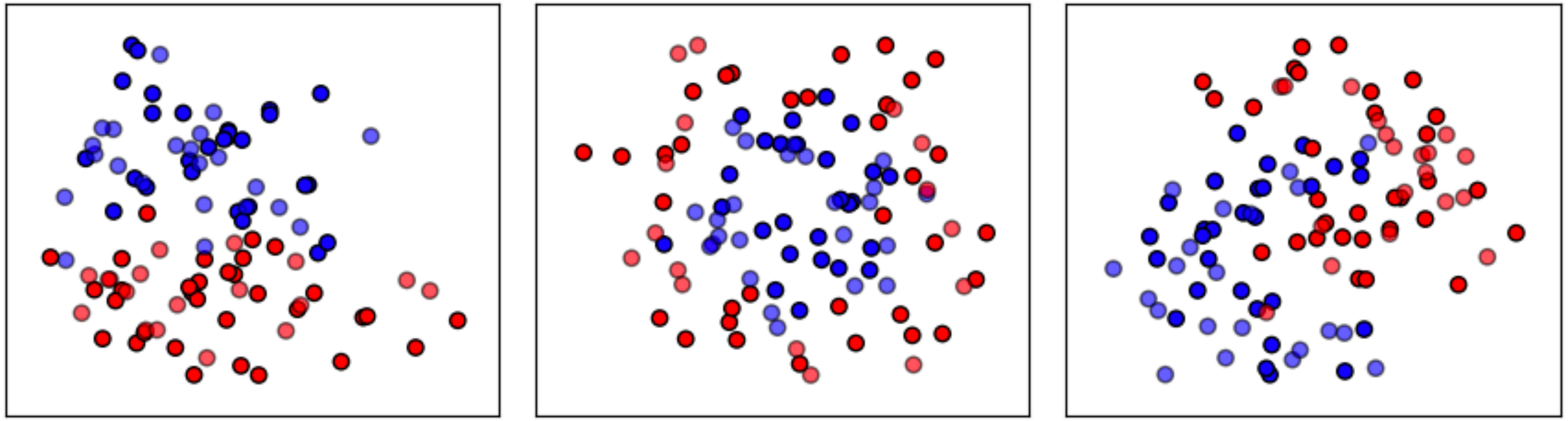
Cat
Dog
Horse
Person
Fish
Bird
...

Binary Classification

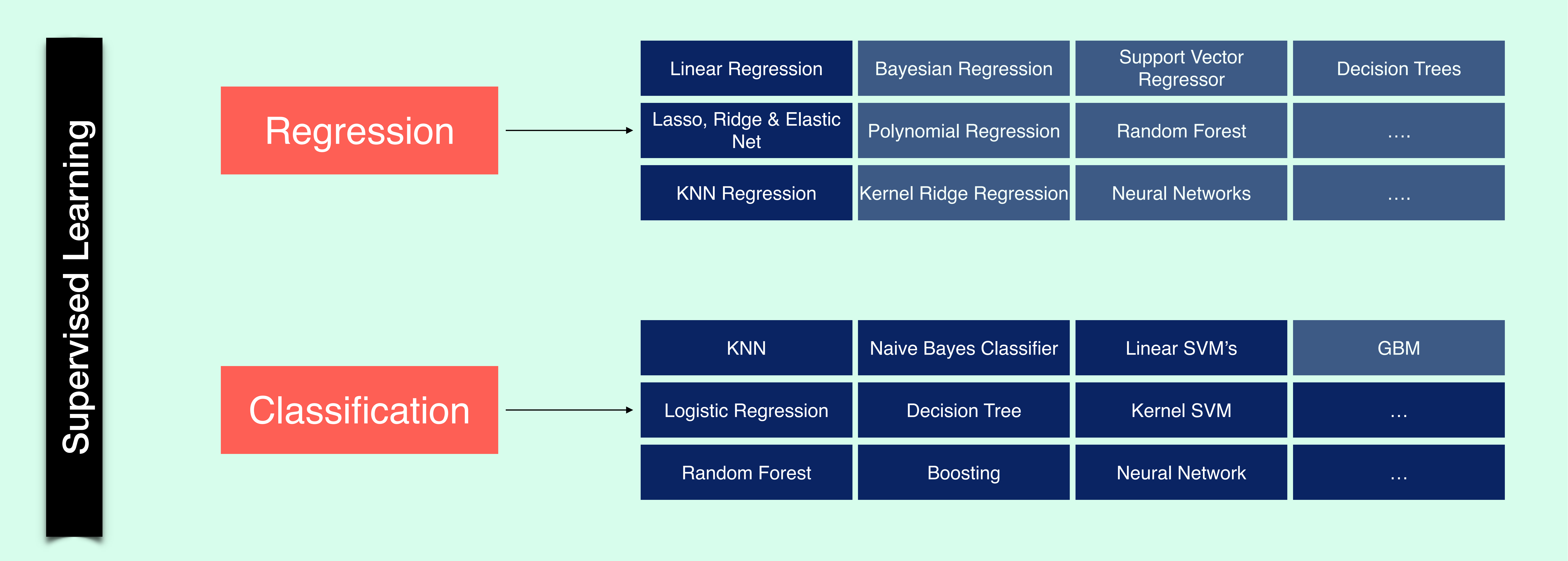
- Refers to those classification tasks that have two class labels.
- Examples include:
  - Email spam detection (spam or not).
  - Churn prediction (churn or not).
  - Conversion prediction (buy or not).
- Typically, binary classification task involve one class that is the normal state and another class is is the abnormal state.
- For example "***not spam***" is the **normal** state and "***spam***" is the **abnormal** state. Another example is "***cancer not detected***" is the **normal** state of a task that involves a medical test and "***cancer detected***" is the **abnormal** state.
  - Usually, the class for the **normal** state is assigned the class **label 0** and the class with the **abnormal** state is assigned the **class label 1**.

Classification

Which is the optimal boundary for these datasets?



Supervised Learning



Linear Regression?

Credit Card Default Data

Description

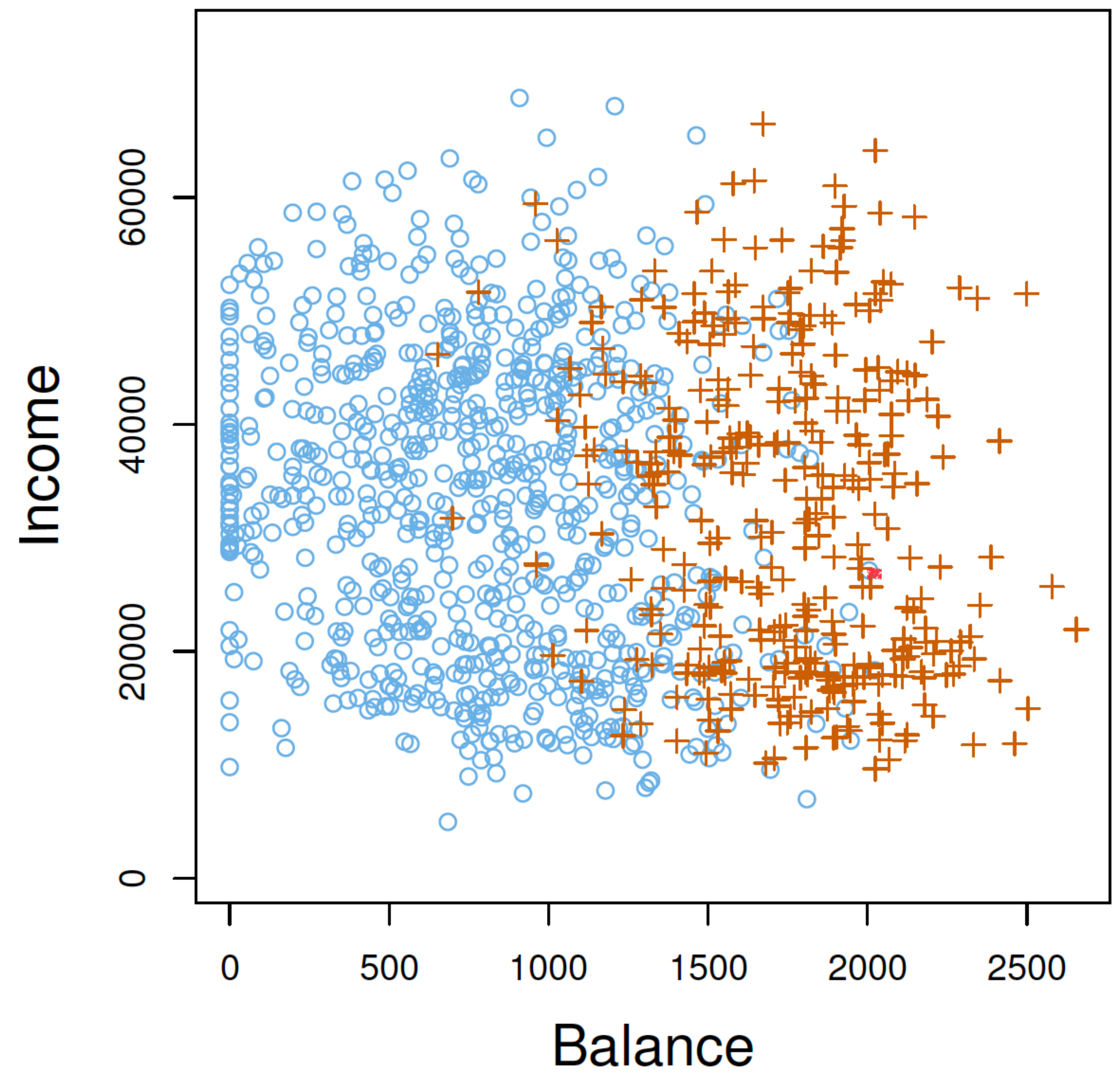
A simulated data set containing information on ten thousand customers. The aim here is to predict which customers will default on their credit card debt.

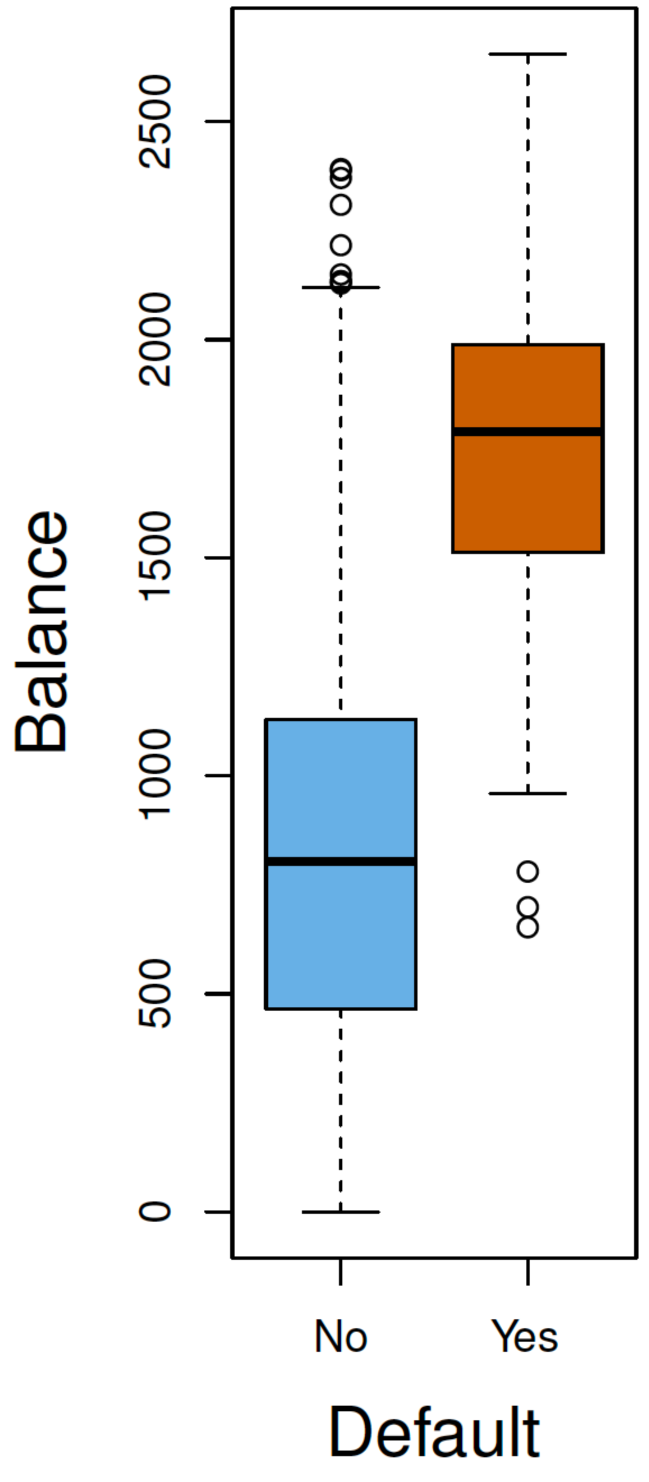
Format

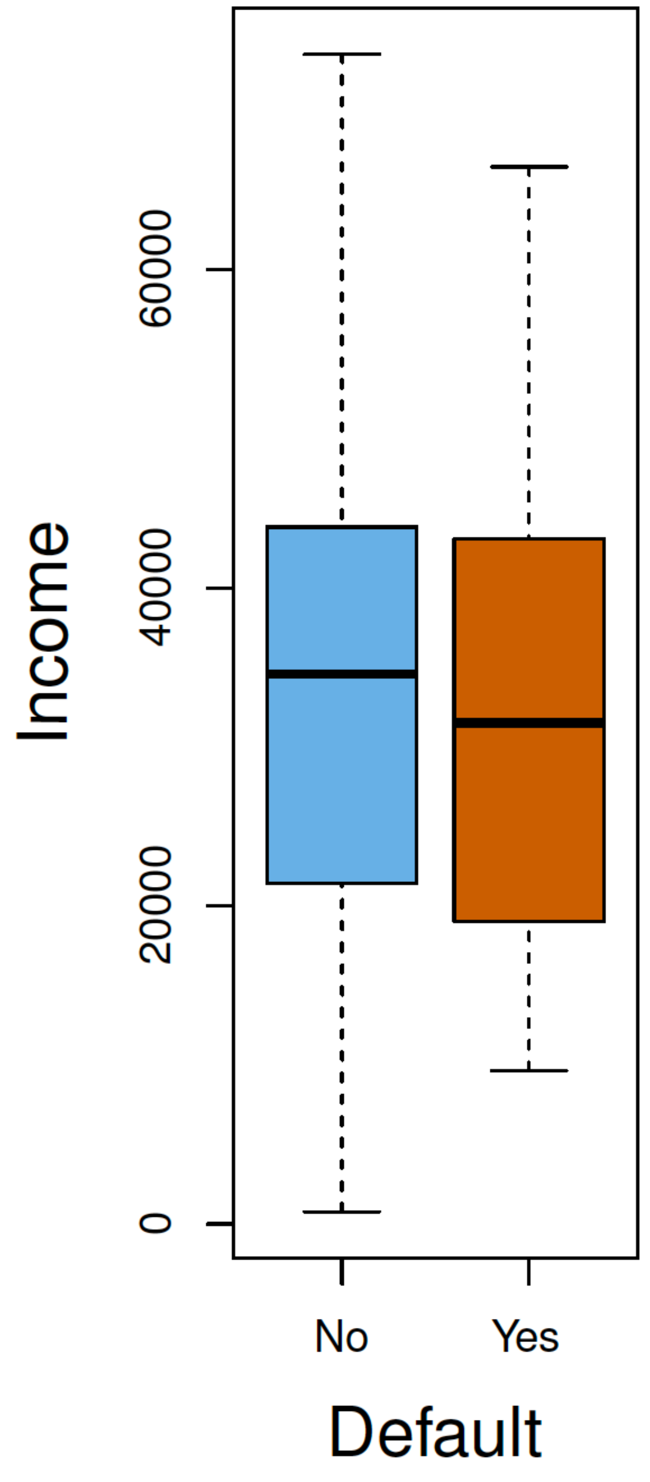
A data frame with 10000 observations on the following 4 variables.

- **default :** A factor with levels No and Yes indicating whether the customer defaulted on their deb
- **student**: A factor with levels No and Yes indicating whether the customer is a student
- **balance:** The average balance that the customer has remaining on their credit card after making their monthly payment
- **income:** Income of customer

Source: Simulated data







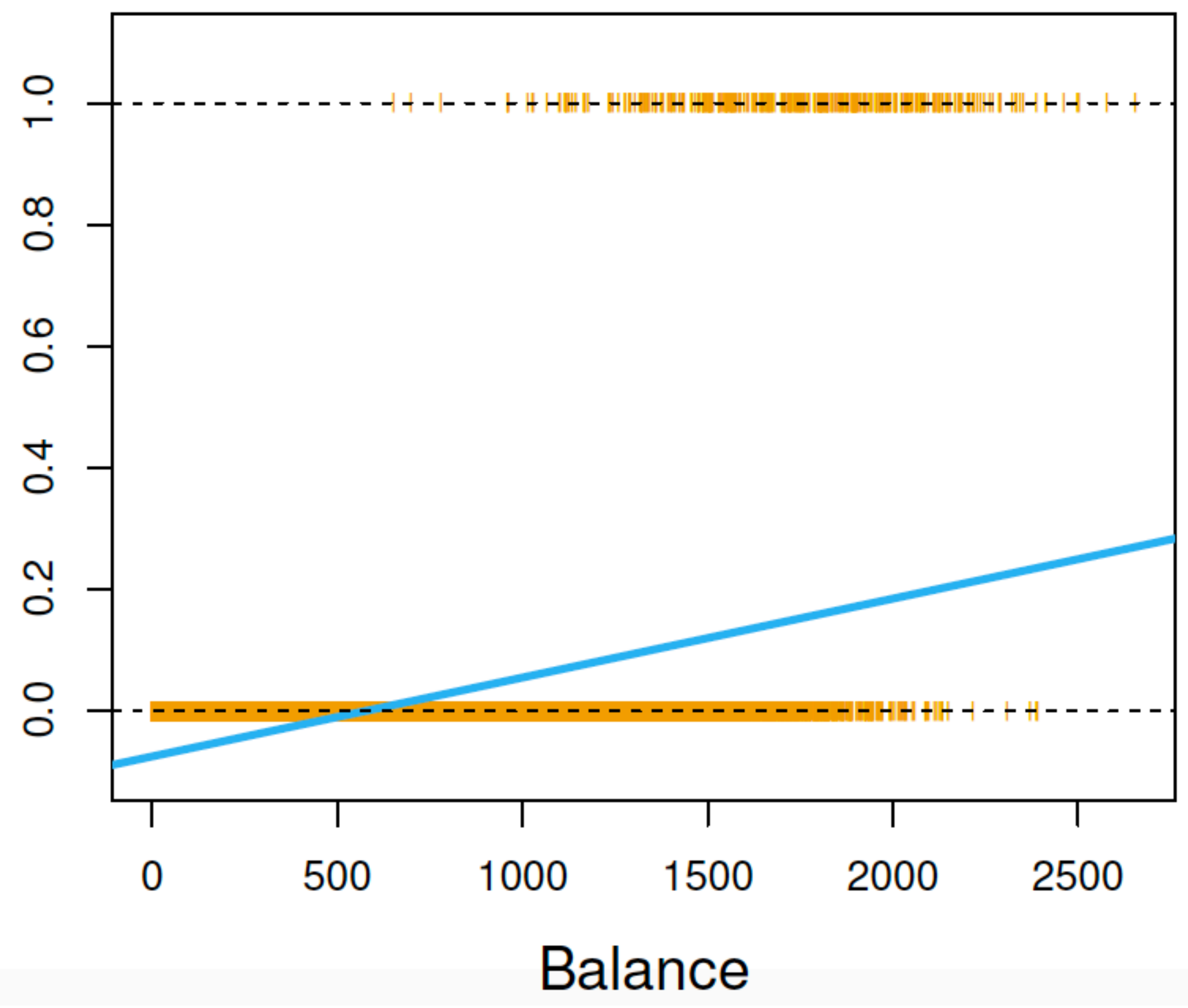
Can we use Linear Regression?

- Suppose for the Default classification task that we code as:

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

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

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



Can we use Linear Regression?

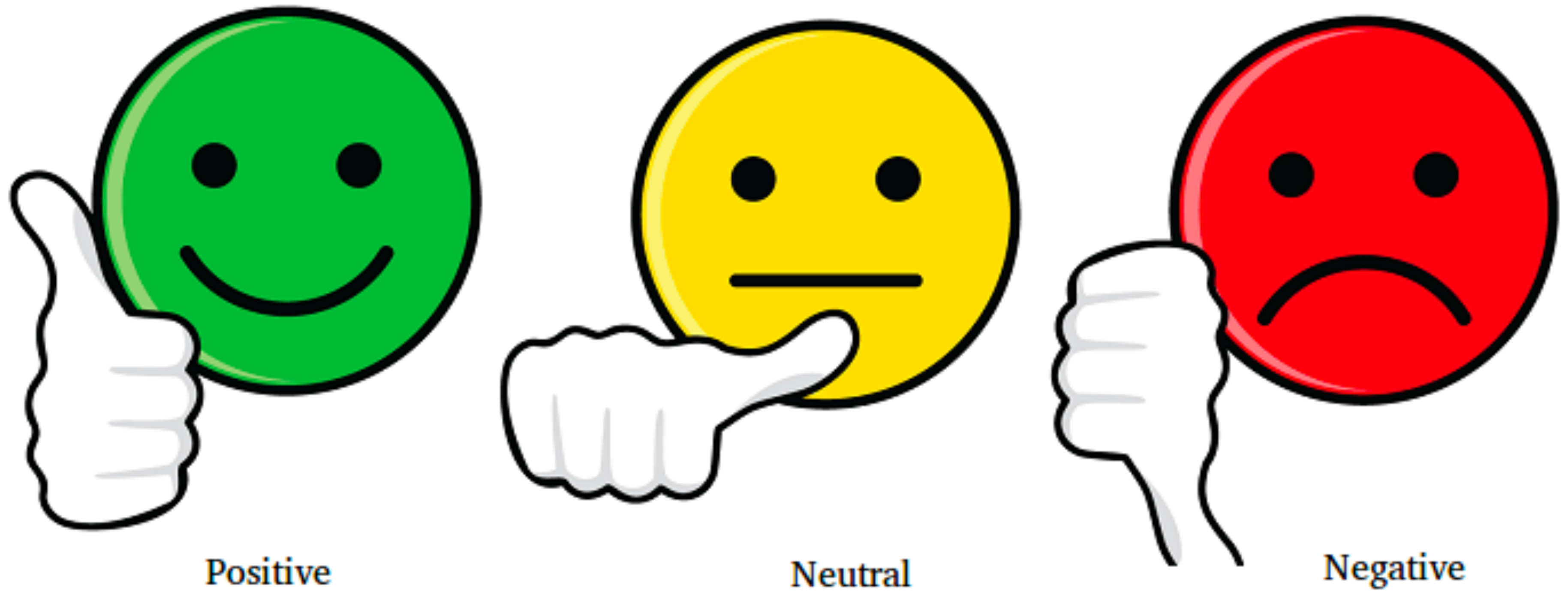
- Suppose for the Default classification task that we code as:

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

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

MultiClass?

Sentiment Classification on twitter



Can we use Linear Regression?

- Suppose that we are trying to predict medical conditions of a patient in emergency room based on their symptoms. Imagine there is three possible diagnosis: **stroke**, **drug overdose** and **epileptic seizure**.

- We could consider encoding these values as a quantitative responses variable, Y, as follows:

$$Y = \begin{cases} 1 \textbf{ if stroke} \\ 2 \textbf{ if drug overdose} \\ 3 \textbf{ if epileptic seizure} \end{cases}$$

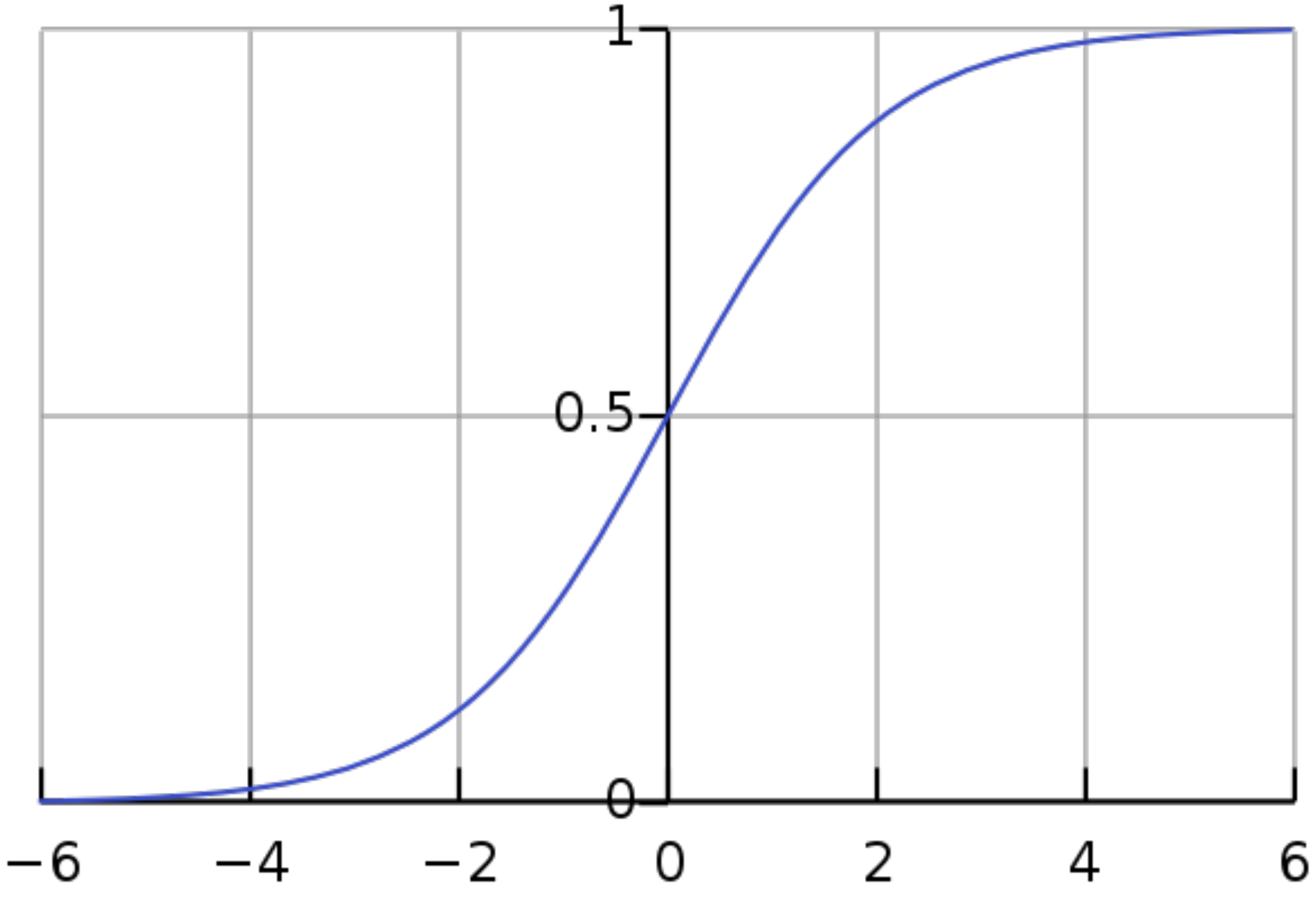
Any problem?

Logistic Regression

Logistic Regression

- A solution for classification is logistic regression. Instead of fitting a straight line or hyperplane, the logistic regression model uses the logistic function to squeeze the **output** of a linear equation between **0 and 1**. The logistic function is defined as:

  - $f(x) = \dfrac{1}{1+e^{-x}} = \dfrac{e^x}{e^x+1}$



Standard Logistic sigmoid Function

Logistic Regression

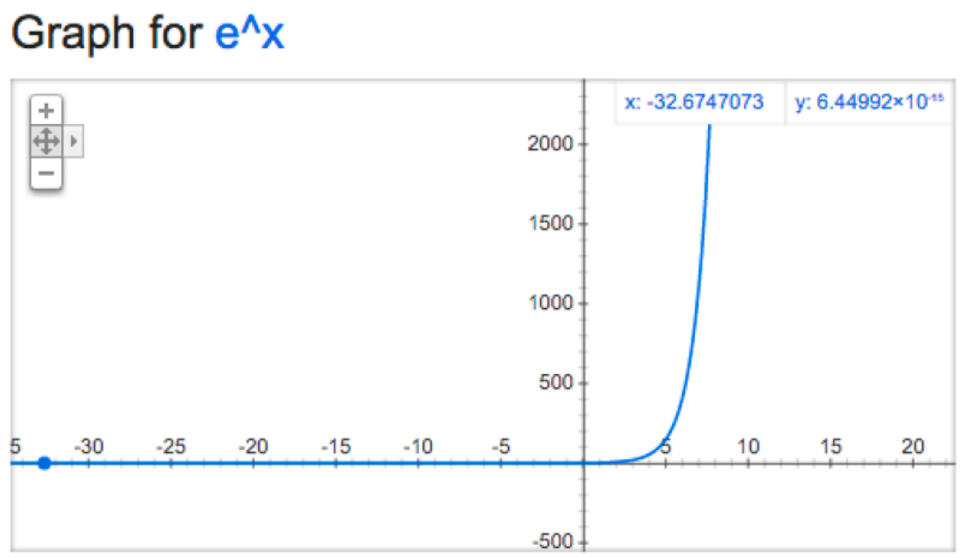
- How should we model the relationship between $p(X) = Pr(Y = 1 | X)$ and X using a function that gives outputs between 0 and 1 for all values of X?

- Logistic regression uses the form:

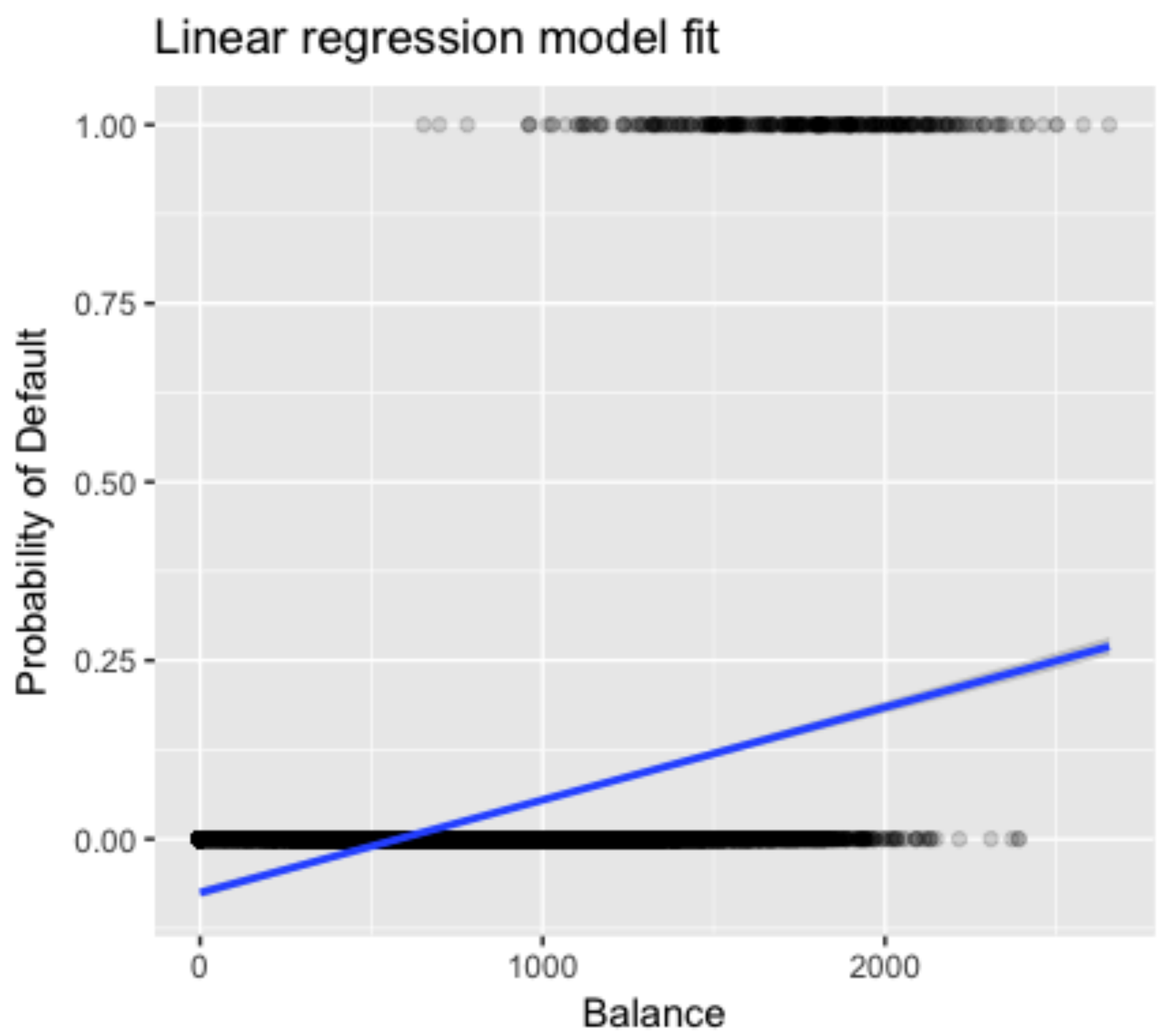
$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

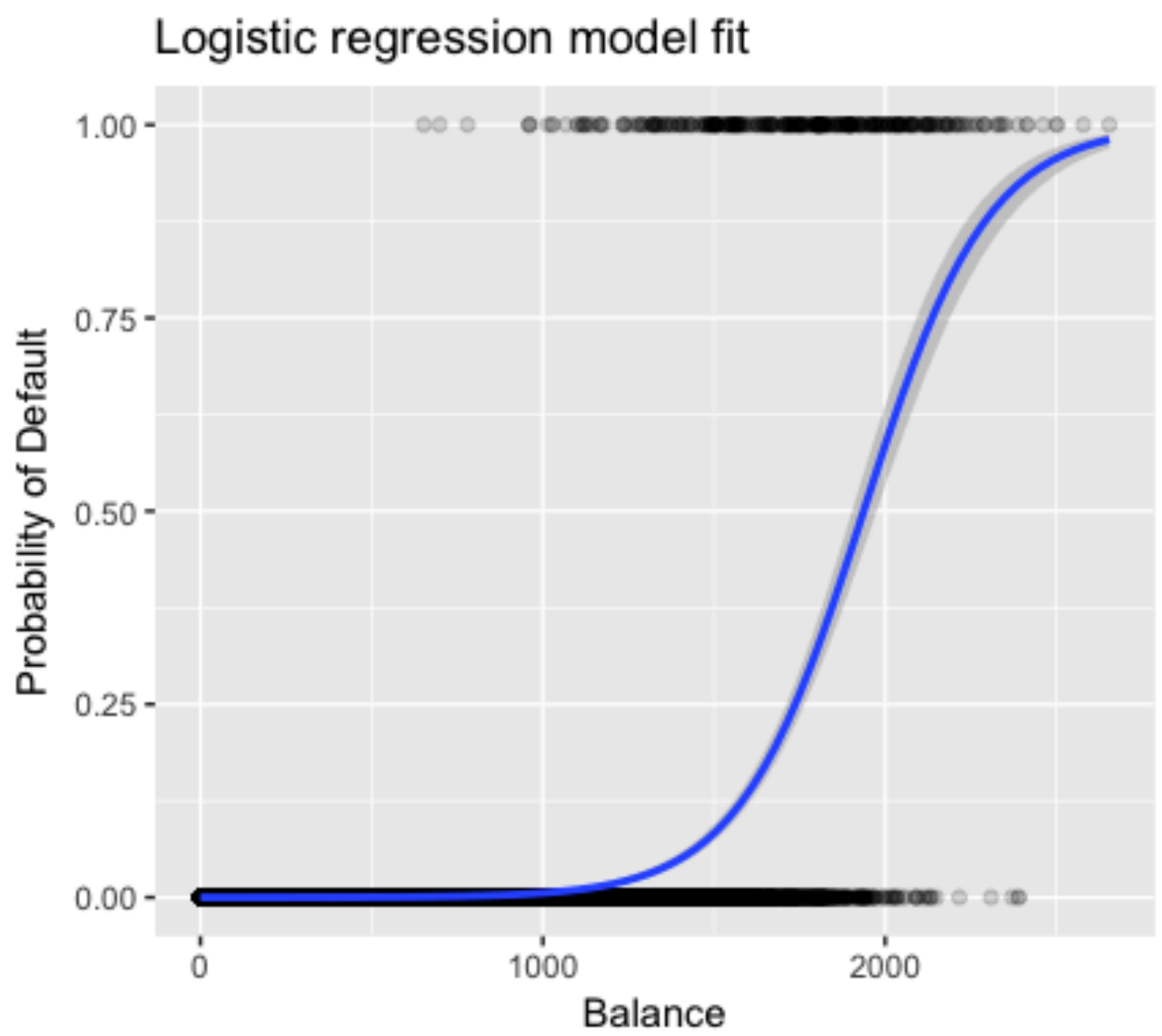
($e = 2.71828$ is a mathematical constant [Euler's number.])

- Remember the e^x function is:



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

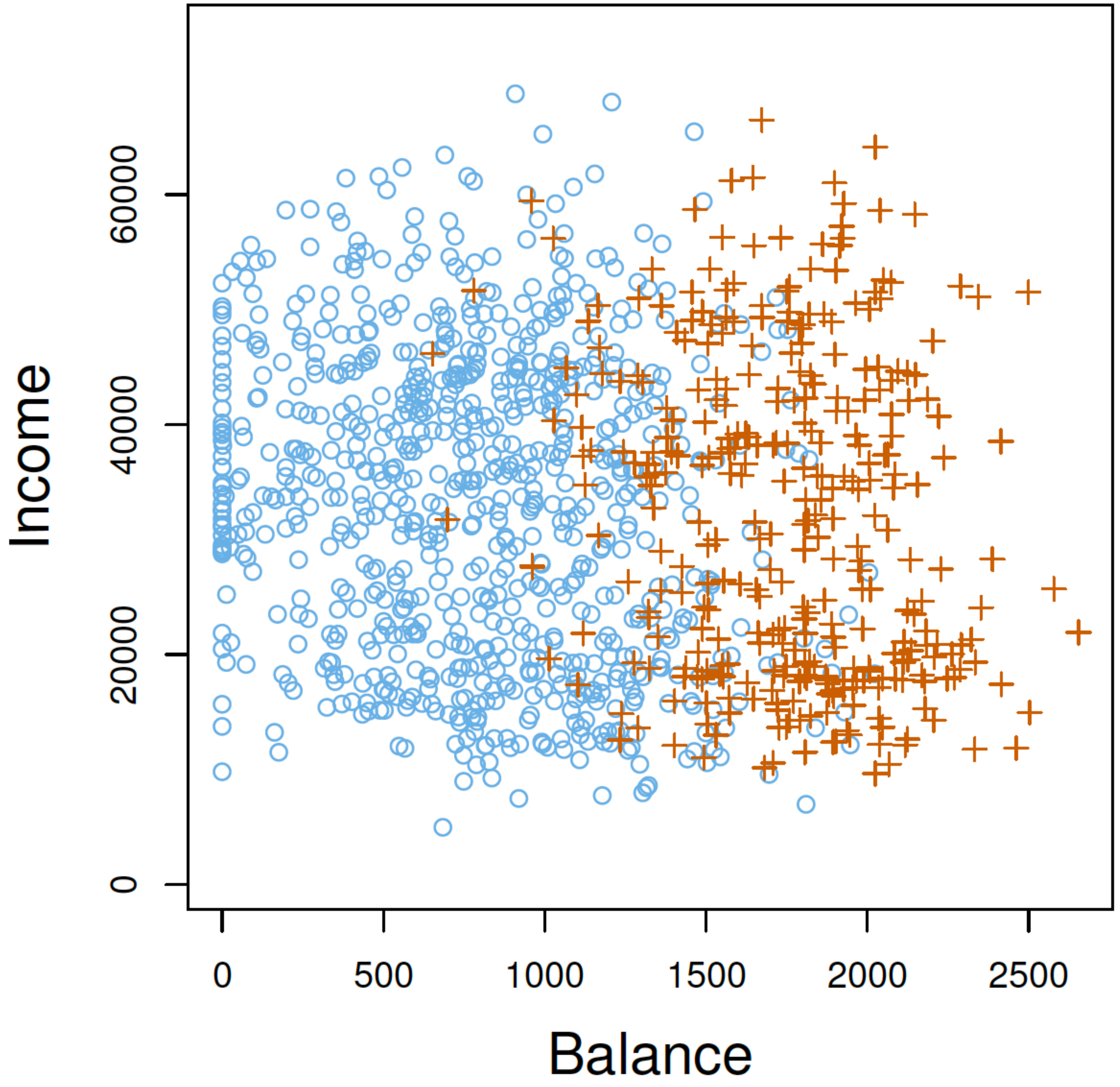




Linear regression does not estimate $Pr(Y = 1 | X)$ well. Logistic regression models the probability the Y belongs to a particular category.

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

Which will be the best boundary?



Logistic Regression

- How should we model the relationship between $p(X) = Pr(Y = 1 | X)$ and X using a function that gives outputs between 0 and 1 for all values of X?
  - Logistic regression uses the form:

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

  - A bit of rearrangement we find that:

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

  - and taking the logarithm in both sides we arrive at:

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

  - Which is called the log odds or logit transformation of p(X).

Suposa que hem recol·lectat dades d'un grup d'estudiants d'aprenentatge automàtic i mineria de dades amb les següents variables:

X_1 : hores d'estudi.

X_2 : nota d'accés a la universitat

Y: Aprovat (1) / Suspès (0)

Hem après una regressió logística i els coeficients obtinguts han estat els següents: $B_0 = -1.3$, $\beta_1 = 0.1$, $\beta_2 = 1$.

Recordeu que el model de la regressió logística té la següent formulació:

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

(a) [1 punt] Estima la probabilitat que té l'estudiant si ha dedicat 60 hores i la nota d'accés a la universitat ha estat un 10.

(b) [1 punt] Quantes hores hauria d'estudiar un estudiant per tenir més d'un 50% de possibilitats d'aprovar l'assignatura d'estadística?

Maximum Likelihood

- As in linear regression, the coefficients β_0 and β_1 are unknown, and must be estimated using the training data.

- To fit the model we use a method called **maximum likelihood**. The basic intuition behind maximum likelihood to fit a logistic regression model as follows: we seek estimates β_0 and β_1 such that the predicted probabilities $\hat{p}(x_i)$ of default for each individual, corresponds as closely as possible to the individual's observed default status.

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

- This **likelihood** gives the probability of the observed zeros and ones in the data. We pick β_0 and β_1 to maximize the likelihood of the observed data.

Maximum Likelihood

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

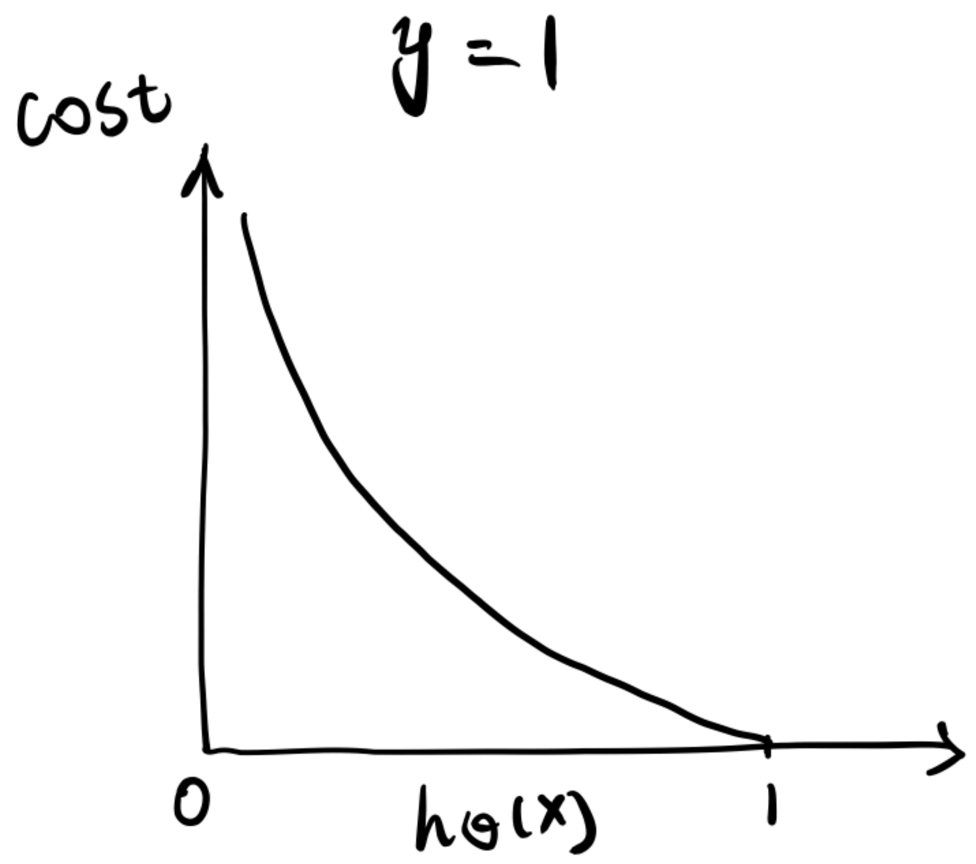
Remember:

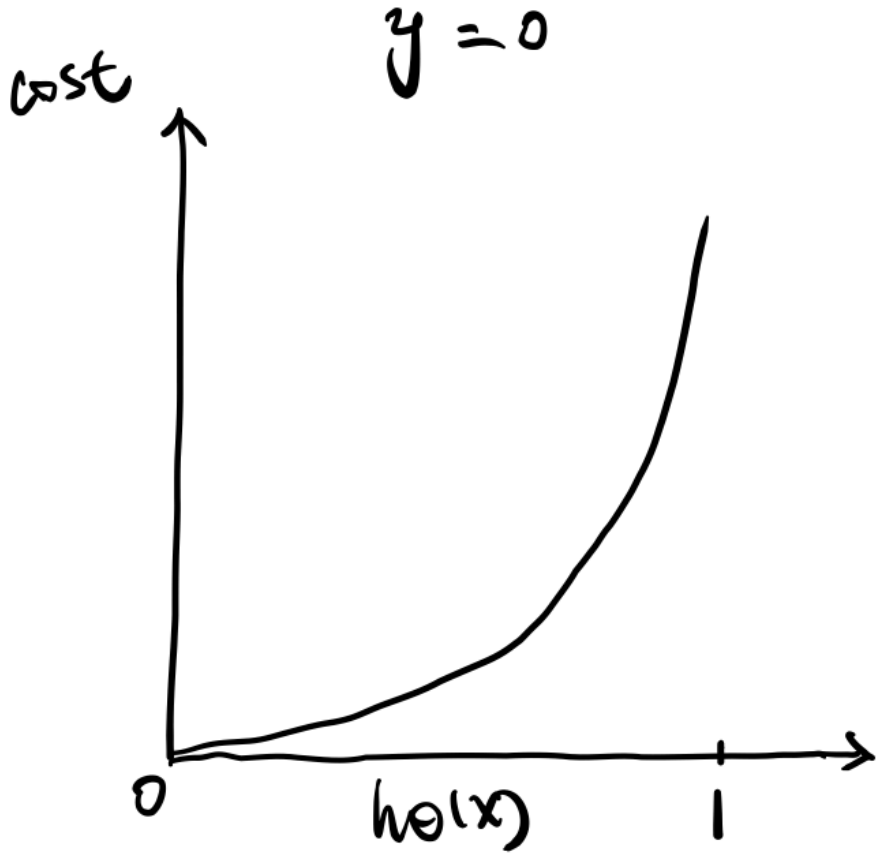
$$\log(\prod AB) = log(\sum A) + log(\sum B)$$

Cost of a single example

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{if } y = 1 \\ -log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

$$Cost(h_\theta(x), y) = -ylog(h_\theta(x)) - (1 - y)log(h_\theta(x))$$





$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

$$J(\theta) = \frac{1}{m}[\sum_{i=1}^{m} -y^{(i)} log(h_\theta(x^{(i)})) + (1 - y^{(i)})log(1 - h_\theta(x^{(i)}))]$$

$$m = number\ of\ samples$$

Optimization

- Gradient Descent or any of its variant can be used to optimize the parameters θ of the model.

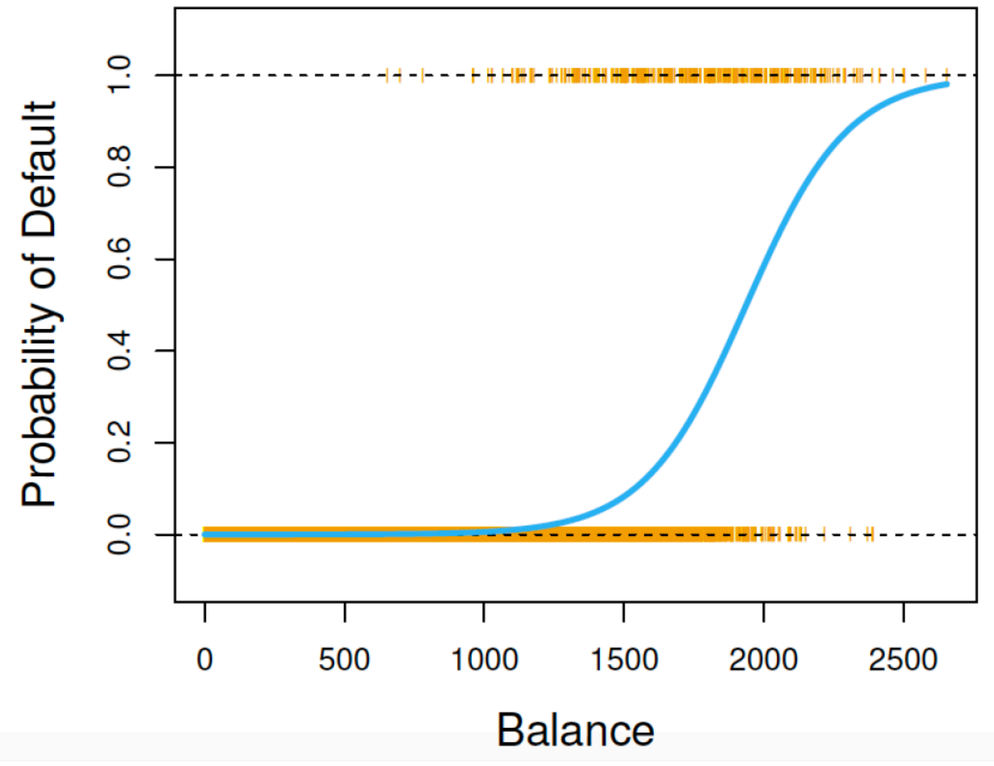
- We need to compute the partial derivative with regard to j^{th} model parameter θ_j:

    - $\frac{\partial}{\partial\theta_j}J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(\sigma(\theta^T x^{(i)} - y^{(i)})x_j^{(i)})$

Maximum Likelihood

- 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



Maximum Likelihood

	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

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

$$\hat{p}(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} = \frac{e^{-10.6513+0.0055x1000}}{1+\epsilon^{-10.6513+0.0055x1000}} = 0.006$$

- What is our estimated probability of default for someone with a balance of 2000$?

$$\hat{p}(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} = \frac{e^{-10.6513+0.0055x2000}}{1+\epsilon^{-10.6513+0.0055x2000}} = 0.586$$

- Let's do it again, using student as the predictor.

	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

$$\widehat{Pr}\,(default = Yes\,|\,student = Yes) = \frac{e^{-3.5041+0.4049x1}}{1+e^{-3.5041+0.4049x1}} = 0.0431$$

$$\widehat{Pr}\,(default = Yes\,|\,student = No) = \frac{e^{-3.5041+0.4049x0}}{1+e^{-3.5041+0.4049x0}} = 0.0292$$

Logistic regression with several variables

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

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

- Lets do it again, using **balance**, **income** and **student** as the predictors

	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

Logistic regression with several variables

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

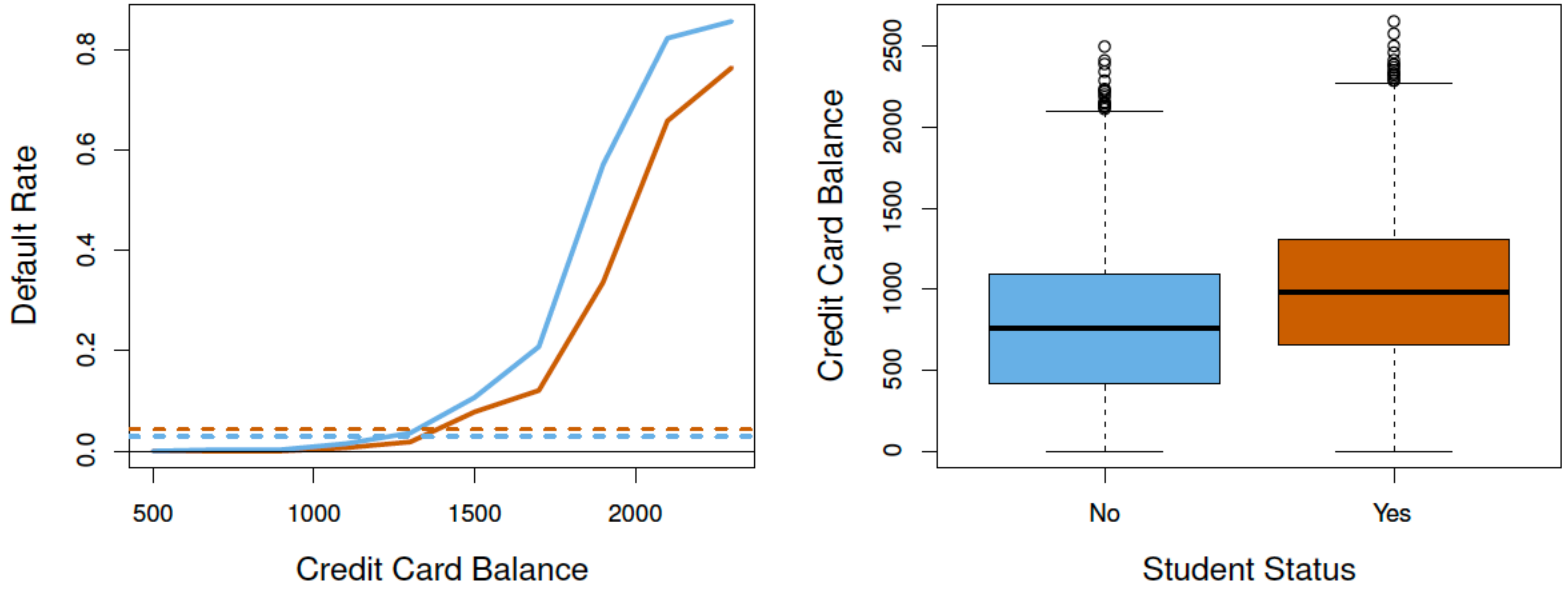
$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\ldots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\ldots+\beta_p X_p}}$$

- Lets do it again, using **balance**, **income** and **student** as the predictors

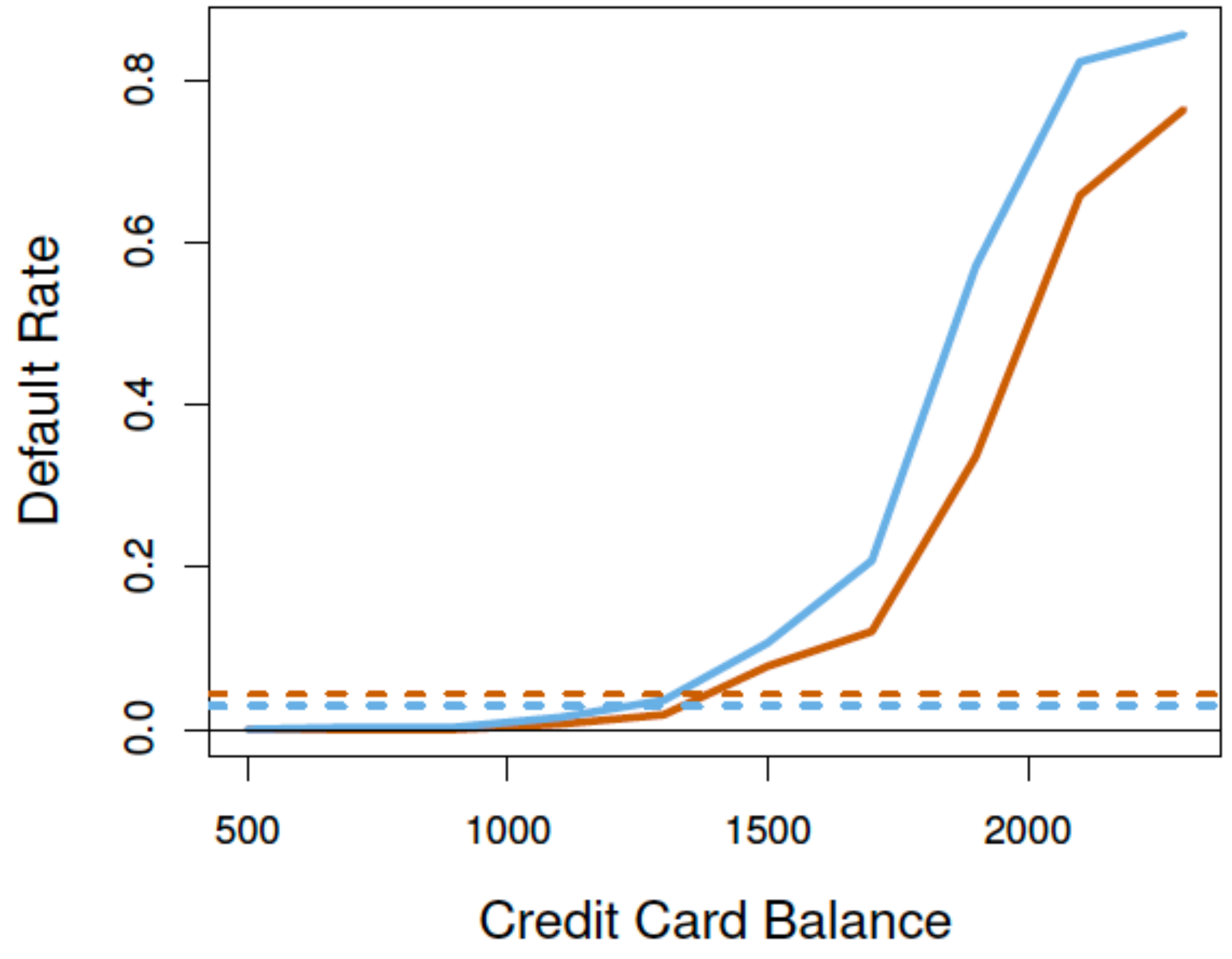
	Coefficient	Std. Error	Z-statistic	p-value
Intercept	-10.8690	0.4923	-22.08	<0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	-0.6468	0.2362	-2.74	0.0062

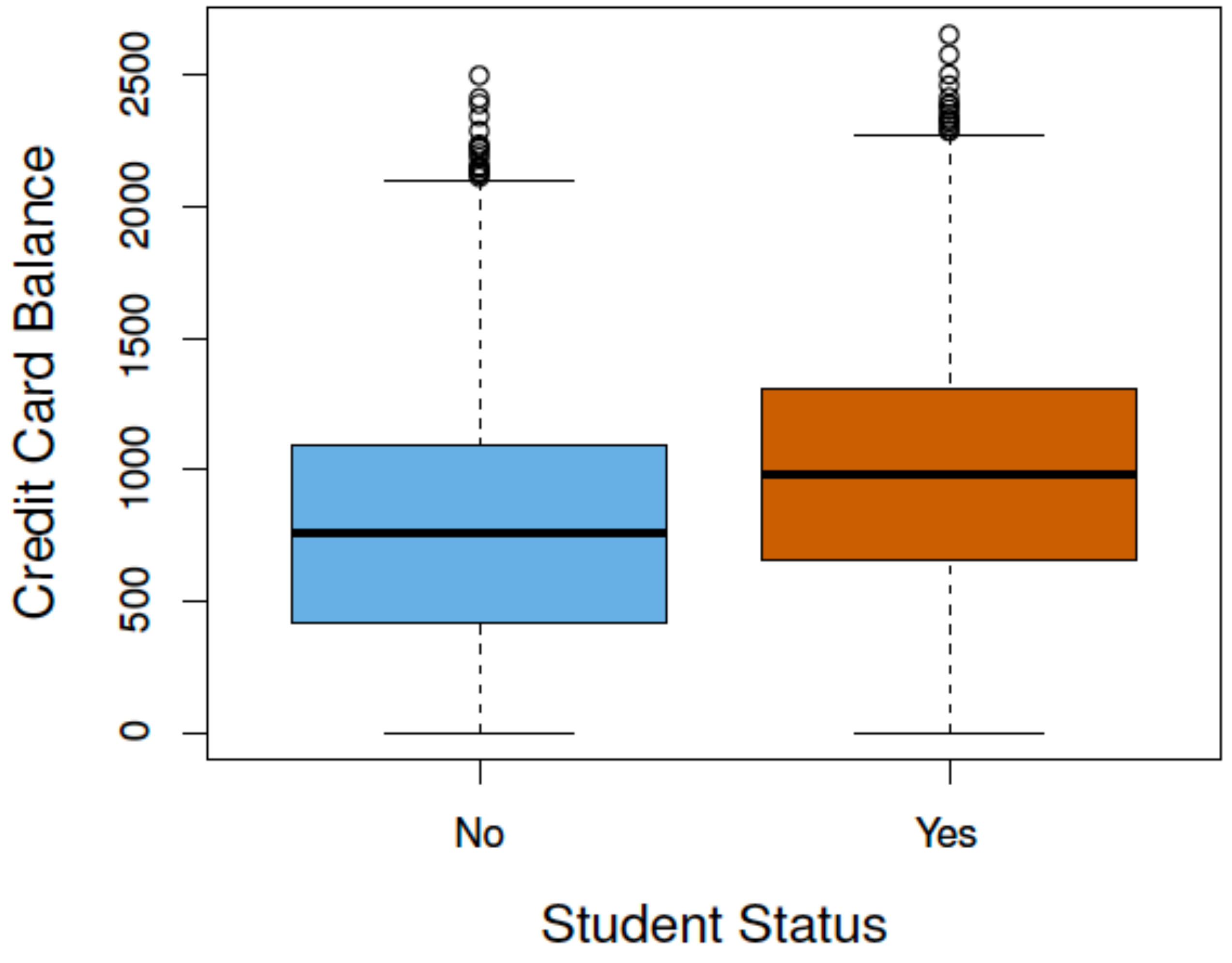
Why is coefficient for **student** negative, while it was positive before?

Confounding



Confounding

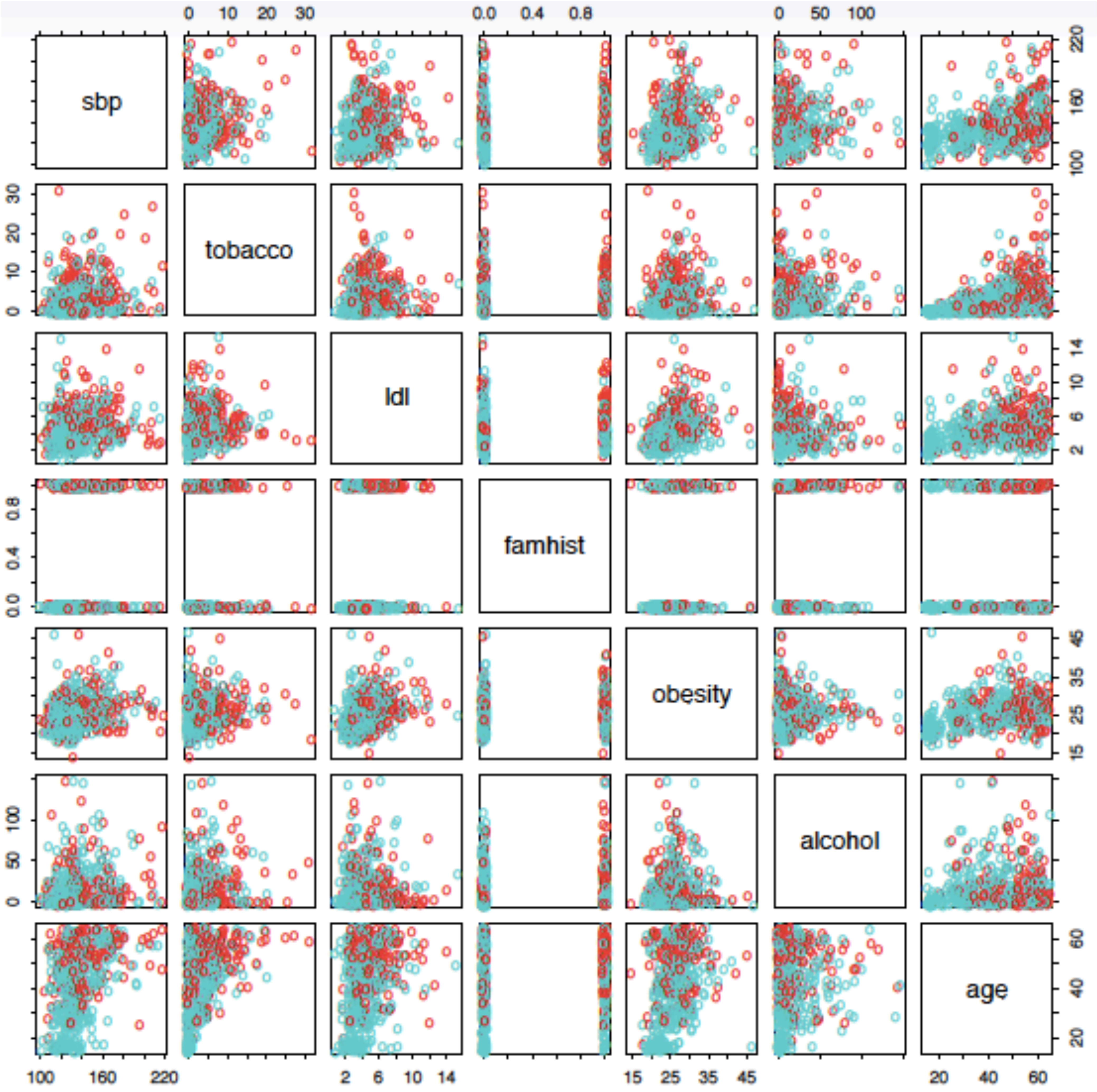




- Students tend to have higher balances than non-students, so their marginal default rate is higher than for non-students.
- But for each level of balance, students default less than non-students.
- **Student is less risky than a non-student with the same credit balance!**

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.



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
```

Which is the probability of having a heart attack?

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 β_j accurately (if our model is correct); the constant term β_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.

How to evaluate a Classifier

Binary Classification

Classification on Credit Data

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

- (23 + 252) / 10000 errors - 2.75% misclassification rate! Some points to consider:
- 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

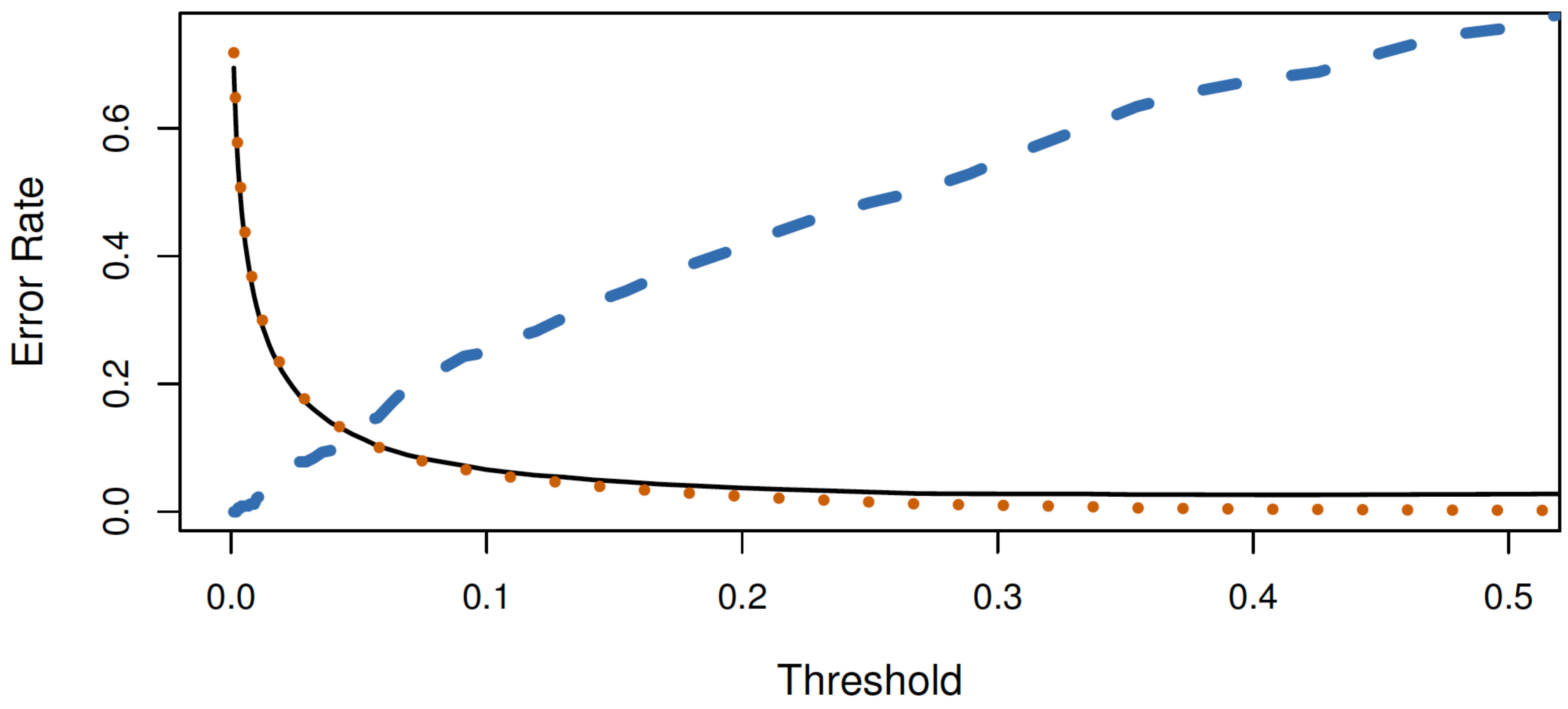
$$Pr(Default = Yes \mid Balance, Student) \geq 0{,}5$$

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

$$Pr(Default = Yes \mid Balance, Student) \geq threshold$$

and vary threshold

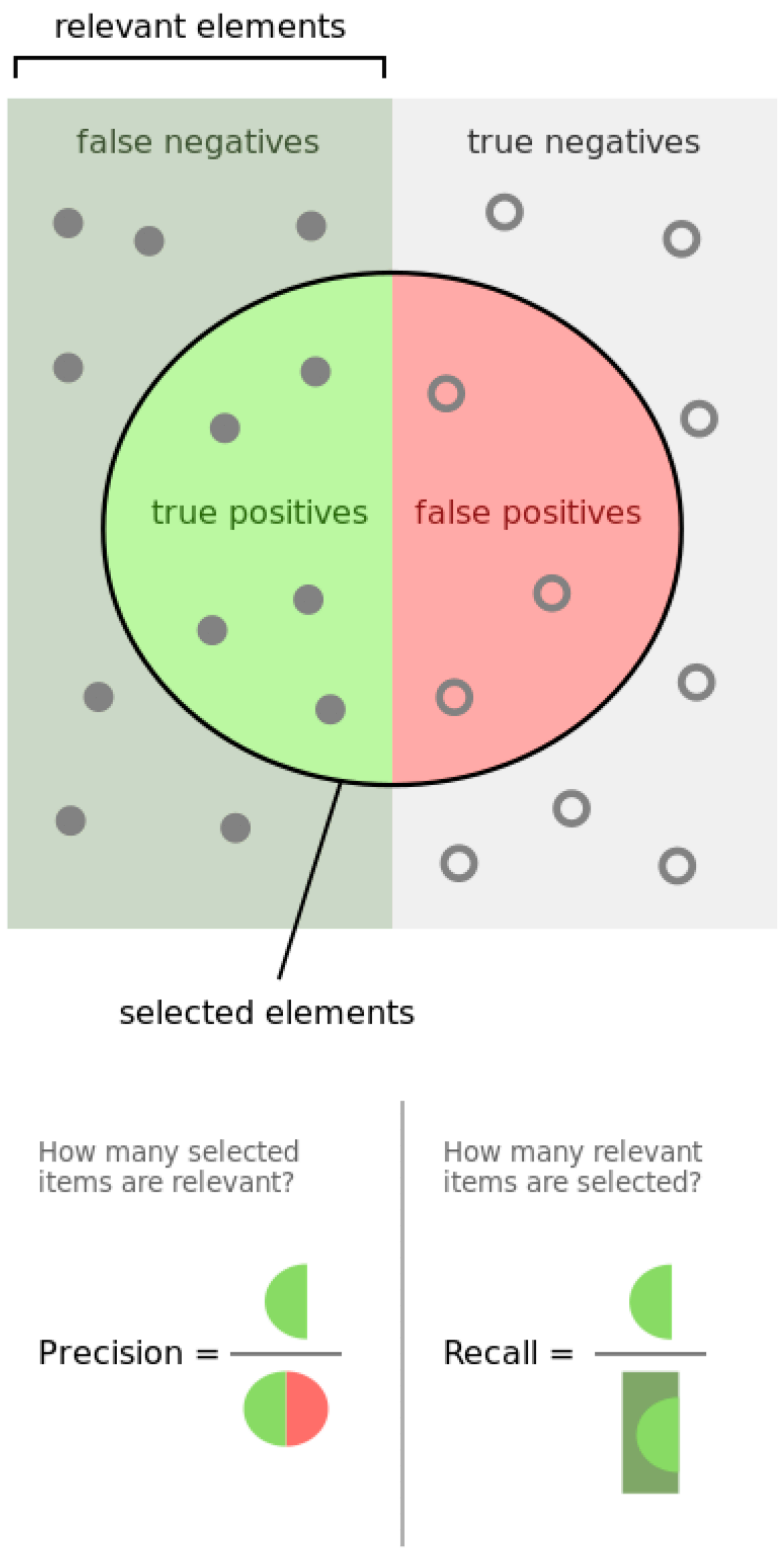
Varying the threshold



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

Classification Error

- **FP** = Negative considered Positive, also considered Type I Error
- **FN** = Positive considered Negative, also know as Type II error
- **What is best, FP or FN?**
  - **It depends on the problem.**
    - A FP may lead a subject to undergo an unnecessary treatment
    - A FN may not receive an intervention when one would have been beneficial


relevant elements
false negatives
true negatives
true positives
false positives
selected elements
How many selected items are relevant?
How many relevant items are selected?
Precision =
Recall =

Example: We want to train a model that for the diagnosis of Colon Cancer

Training:

Positive Cases: 10

Negative Cases: 90

Test:

Positive Cases: 10

Negative Cases: 90

Example: We want to train a model that for the diagnosis of Colon Cancer

Training:

Positive Cases: 10

Negative Cases: 90

Test:

Positive Cases: 10

Negative Cases: 90

Train Model

Evaluate Model

— — — — — Result — — — — —

Positive Cases: **7 classified as Positive**

3 classified as Negative

Negative Cases **81 classified as Negative**

9 classified as Positive

Evaluation Measures

Accuracy or Classification Error

Confusion Matrix

Precision

Recall/Sensitivity

Specificity

F1-Score

ROC Curve

AUC

Accuracy = (TP + TN) / N

(7 + 81) / 100 = 0.88

Evaluation Measures

Accuracy or Classification Error

Confusion Matrix

Precision

Recall/Sensitivity

Specificity

F1-Score

ROC Curve

AUC

		Predicted	
		Positive	Negative
Real	Positive	TP	FN
	Negative	FP	TN

		Predicted	
		Positive	Negative
Real	Positive	7	3
	Negative	9	81

Evaluation Measures

Accuracy or Classification Error

Confusion Matrix

Precision

Recall/Sensitivity

Specificity

F1-Score

ROC Curve

AUC

Precision = TP / (TP + FP)

$$\frac{7}{7+9} = \frac{7}{16} = 0.4375$$

Evaluation Measures

Accuracy or Classification Error

Confusion Matrix

Precision

Recall/Sensitivity

Specificity

F1-Score

ROC Curve

AUC

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{P}$$

$$\frac{7}{7+3} = \frac{7}{10} = 0.7$$

Evaluation Measures

Accuracy or Classification Error

Confusion Matrix

Precision

Recall/Sensitivity

Specificity

F1-Score

ROC Curve

AUC

$$Specificity = \frac{TN}{TN + FP} = \frac{TN}{N}$$

$$\frac{81}{81 + 9} = \frac{81}{90} = 0.9$$

Evaluation Measures

Accuracy or Classification Error

Confusion Matrix

Precision

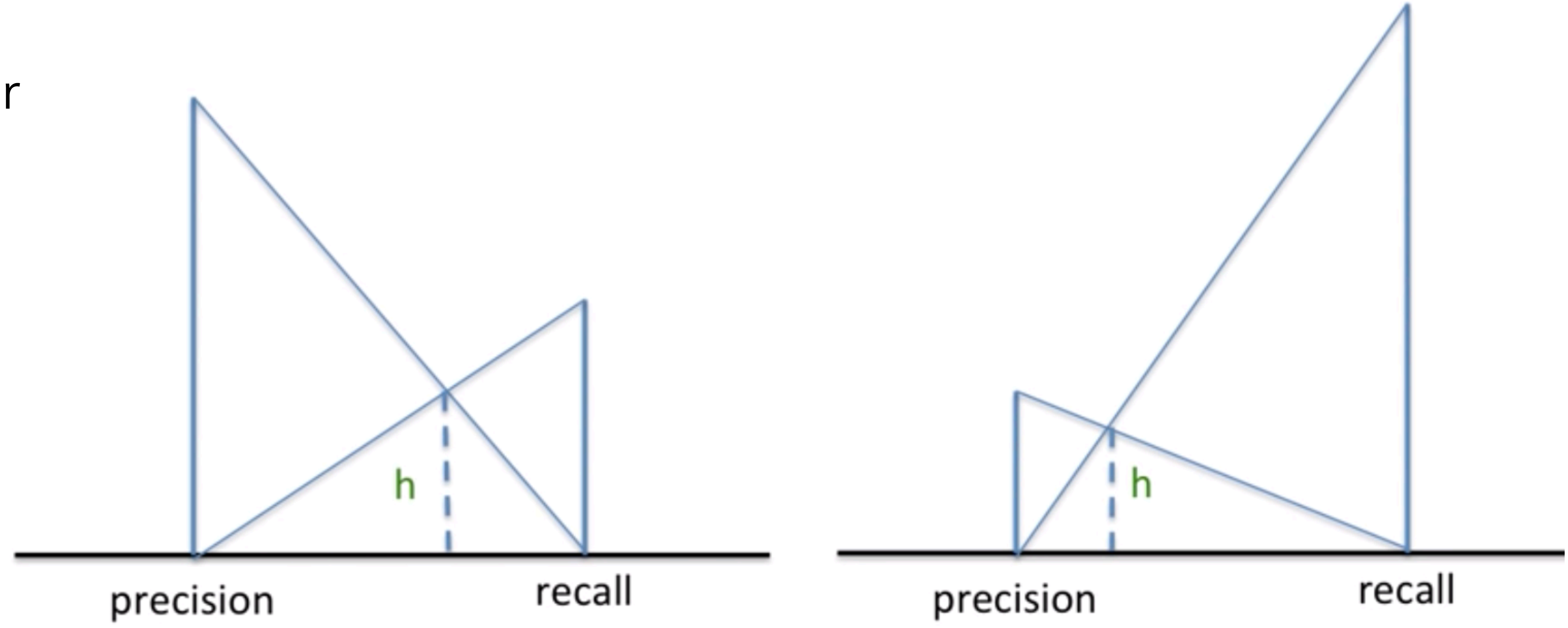
Recall/Sensitivity

Specificity

F1-Score

ROC Curve

AUC



h is half the harmonic mean

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Evaluation Measures

Accuracy or Classification Error

Confusion Matrix

Precision

Recall/Sensitivity

Specificity

F1-Score

ROC Curve

AUC

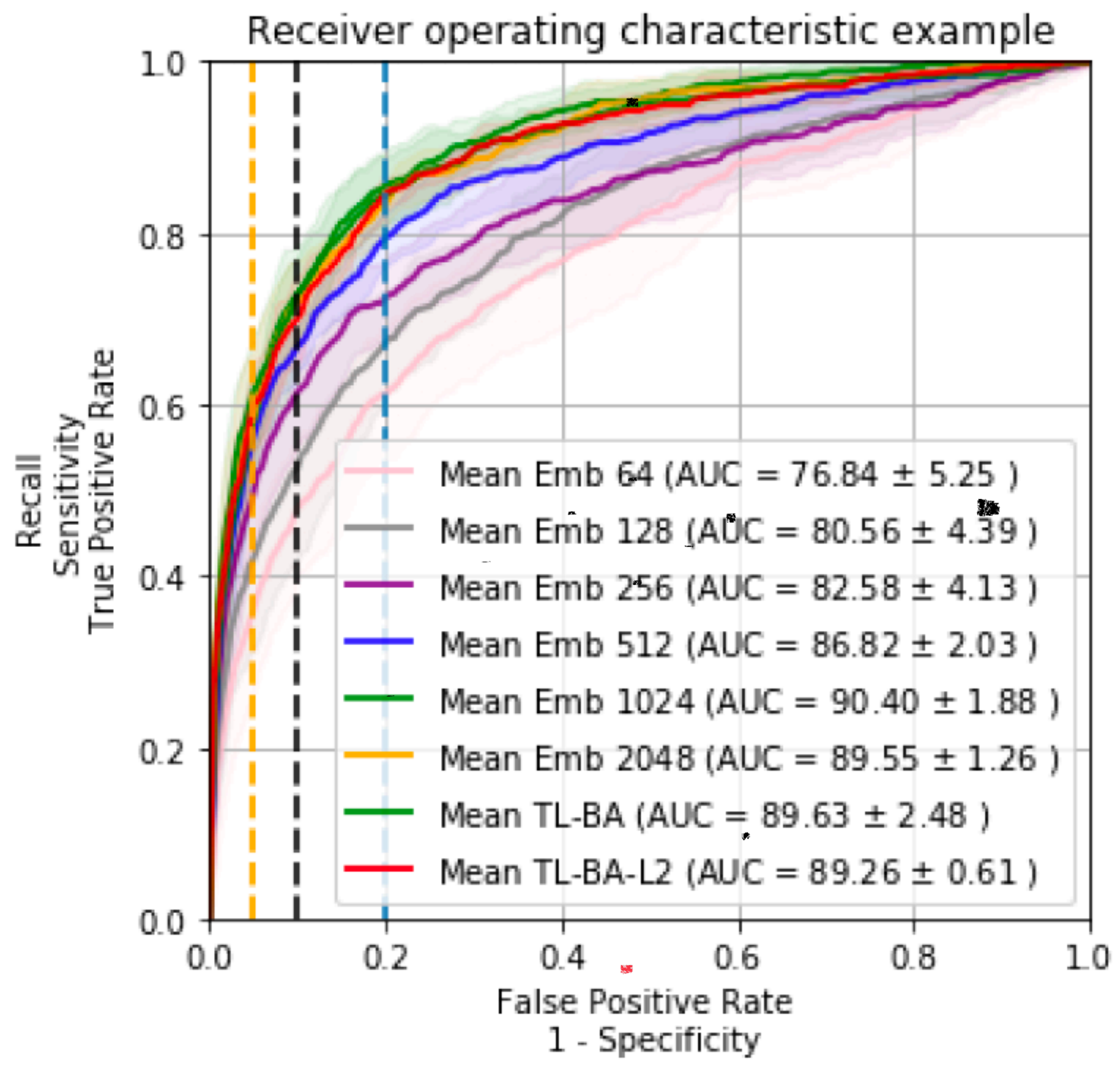
$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

$$2 \cdot \frac{0.4375 \cdot 0.7}{0.4375 + 0.7} = 0.538$$

ROC-Curve & AUC

- There is, usually, a trade-off between *sensitivity* and *specificity*: this is controlled by the classification threshold θ, and can be displayed graphically through the receiver operating curve.
- The *area under the ROC* curve (AUC) provides a measure of the performance of the classification model over all values θ.
- The AUC is independent on the amount of positive and negative cases.

ROC-Curve & AUC



Which is the best?

MODEL A

—————Result—————

Positive Cases:
- **7 classified as Positive**
- 3 classified as Negative

Negative Cases
- **81 classified as Negative**
- 9 classified as Positive

MODEL B

—————Result—————

Positive Cases:
- **9 classified as Positive**
- 1 classified as Negative

Negative Cases
- **78 classified as Negative**
- 12 classified as Positive

Which is the best?

MODEL A

MODEL B

	Model A	Model B
Accuracy	**0.88**	0.87
Precision	**0.43**	0.42
Recall/Sensitivity	0.70	**0.9**
Specificity	**0.90**	0.86
F1-Score	0.53	**0.58**

MultiClass Classification

Definitions

- **Multi-Class:** classification task with more than two classes such that the input is to be classified into one, and only one of these classes. Example: classify a set of images of fruits into any one of these categories – apples, bananas, and oranges.

- **Multi-labels:** classifying a sample into a set of target labels. Example: tagging a blog into one or more topics like technology, religion, politics etc. Labels are isolated and their relations are not considered important.

- **Hierarchical:** each category can be grouped together with similar categories, creating meta-classes, which in turn can be grouped again until we reach the root level (set containing all data). Examples include text classification & species classification.

Example: We want to train a model that predict the type of Colon Cancer (A;B;C and D):

MODEL A

	Predicted			
	Type A	Type B	Type C	Type D
Type A	10	0	0	0
Type B	0	5	3	2
Type C	0	1	8	1
Type D	0	1	0	9

MODEL B

	Predicted			
	Type A	Type B	Type C	Type D
Type A	8	2	0	0
Type B	1	7	3	2
Type C	0	1	9	1
Type D	2	3	0	5

Example: We want to train a model that predict the type of Colon Cancer (A;B;C and D):

MODEL A

	Predicted			
	Type A	Type B	Type C	Type D
Type A	10	0	0	0
Type B	0	5	3	2
Type C	0	1	8	1
Type D	0	1	0	9

Accuracy = (10+5+8+9)/40 = 0.8

MODEL B

	Predicted			
	Type A	Type B	Type C	Type D
Type A	8	2	0	0
Type B	1	7	3	2
Type C	0	1	9	1
Type D	2	3	0	5

Accuracy = (8+7+9+5)/40 = 0.725

Example: We want to train a model that predict the type of Colon Cancer (A;B;C and D): with imabalanced dataset

MODEL A

	Predicted			
	Type A	Type B	Type C	Type D
Type A	100	80	10	10
Type B	0	9	0	1
Type C	0	1	8	1
Type D	0	1	0	9
Precision	100/100	9/91	8/18	9/21

MODEL B

	Predicted			
	Type A	Type B	Type C	Type D
Type A	198	2	0	0
Type B	7	1	0	2
Type C	0	8	1	1
Type D	2	3	4	1
Precision	198/207	1/14	1/5	1/4

Example: We want to train a model that predict the type of Colon Cancer (A;B;C and D): with imabalanced dataset

MODEL A

	Predicted			
	Type A	Type B	Type C	Type D
Type A	100	80	10	10
Type B	0	9	0	1
Type C	0	1	8	1
Type D	0	1	0	9
Precision	100/100	9/91	8/18	9/21

Average Accuracy = (0.5+0.9+0.8+0.9)/4 = 0.775

Average Precisions = 0.492

MODEL B

	Predicted			
	Type A	Type B	Type C	Type D
Type A	198	2	0	0
Type B	7	1	0	2
Type C	0	8	1	1
Type D	2	3	4	1
Precision	198/207	1/14	1/5	1/4

Average Accuracy = (0.99+0.1+0.1+0.1)/4 = 0.3225

Table 3
Measures for multi-class classification based on a generalization of the measures of Table 1 for many classes C_i: tp_i are true positive for C_i, and fp_i – false positive, fn_i – false negative, and tn_i – true negative counts respectively. μ and M indices represent micro- and macro-averaging.

Measure	Formula	Evaluation focus
Average Accuracy	$\frac{\sum_{i=1}^{l} \frac{tp_i+tn_i}{tp_i+fn_i+fp_i+tn_i}}{l}$	The average per-class effectiveness of a classifier
Error Rate	$\frac{\sum_{i=1}^{l} \frac{fp_i+fn_i}{tp_i+fn_i+fp_i+tn_i}}{l}$	The average per-class classification error
$Precision_\mu$	$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i+fp_i)}$	Agreement of the data class labels with those of a classifiers if calculated from sums of per-text decisions
$Recall_\mu$	$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i+fn_i)}$	Effectiveness of a classifier to identify class labels if calculated from sums of per-text decisions
$Fscore_\mu$	$\frac{(\beta^2+1)Precision_\mu Recall_\mu}{\beta^2 Precision_\mu+Recall_\mu}$	Relations between data's positive labels and those given by a classifier based on sums of per-text decisions
$Precision_M$	$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i+fp_i}}{l}$	An average per-class agreement of the data class labels with those of a classifiers
$Recall_M$	$\frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i+fn_i}}{l}$	An average per-class effectiveness of a classifier to identify class labels
$Fscore_M$	$\frac{(\beta^2+1)Precision_M Recall_M}{\beta^2 Precision_M+Recall_M}$	Relations between data's positive labels and those given by a classifier based on a per-class average

https://www.sciencedirect.com/science/article/pii/S0306457309000259

- In Micro-average method, you sum up the individual true positives, false positives, and false negatives of the system for different sets and then apply them to get the statistics.

- In Macro-average, you take the average of the precision and recall of the system on different sets

- **Macro-average** is preferable if there is a class imbalance problem.

Linear Discriminant Analysis

Linear 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 these data distribution are assumed to be normal, it turns out that the model is very similar to logistic regression.

- Although, this approach is quite general, and other distributions can be used as well, we will focus on normal distributions.

Linear Discriminant Analysis

- 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:

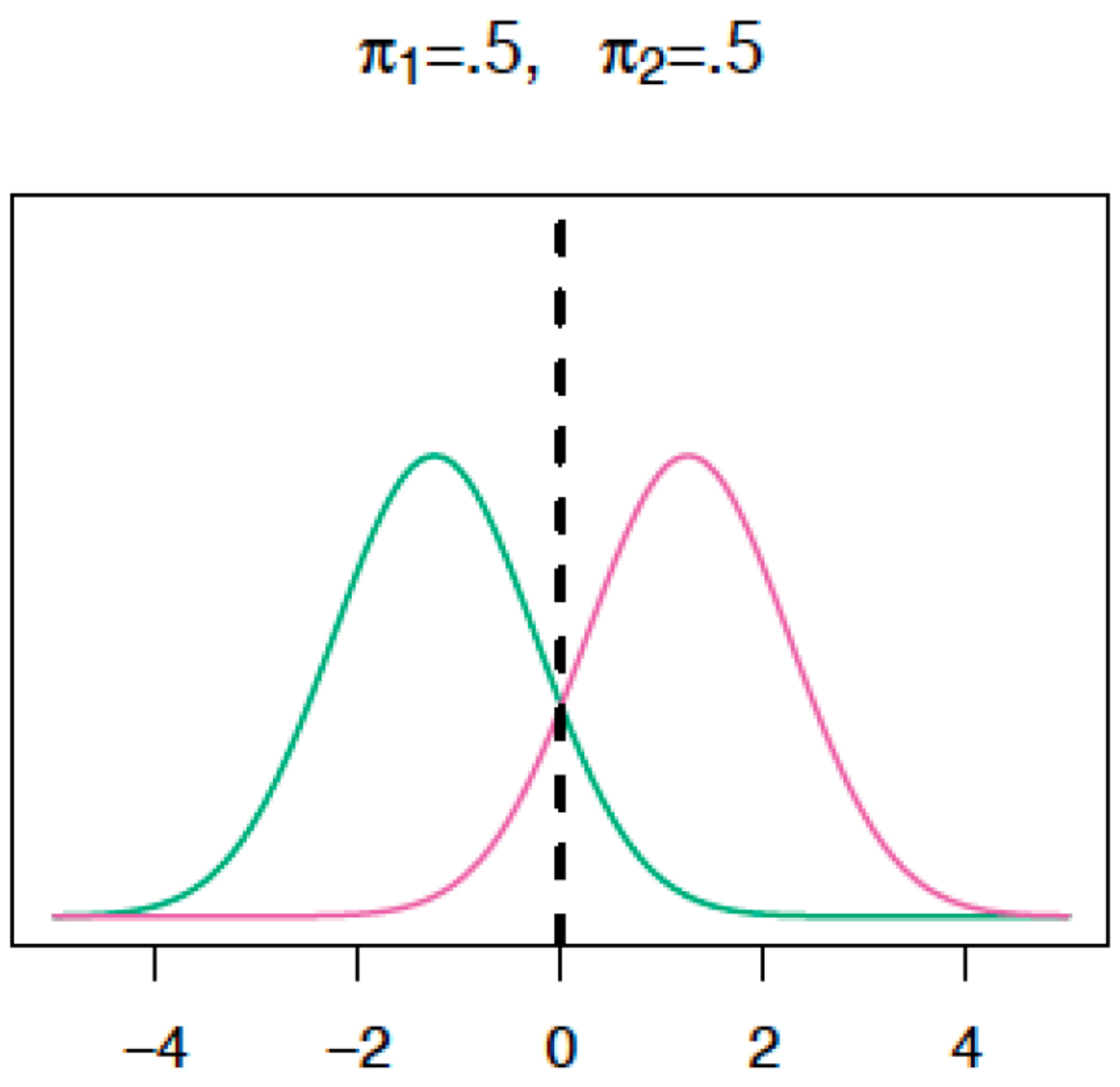
$$Pr(Y = k \,|\, X = x) = \frac{Pr(X = x \,|\, Y = k) \cdot Pr(Y = k)}{Pr(X = x)}$$

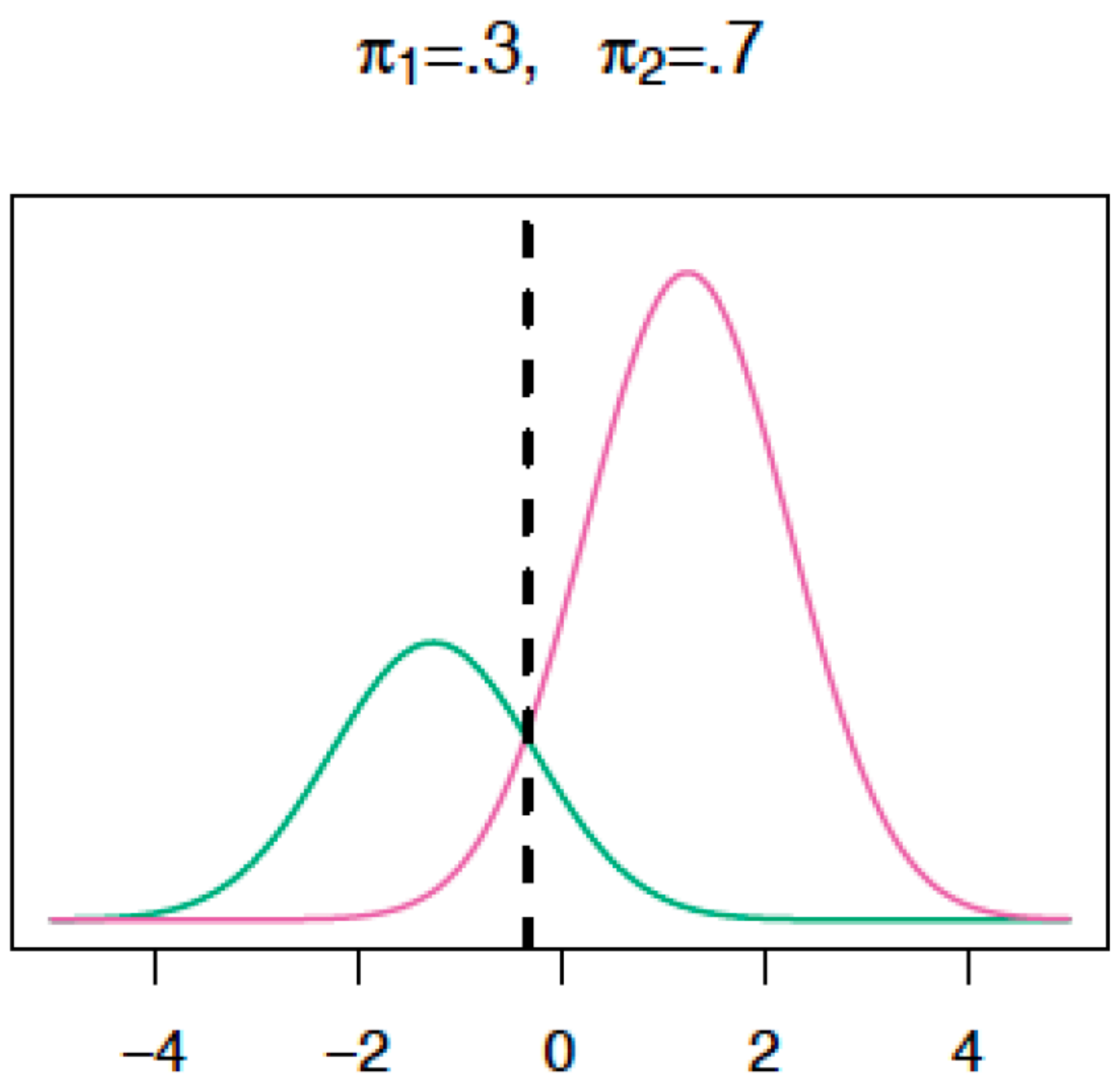
- One writes this slightly differently for discriminant analysis:

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

- where,
  - $f_k(x) = Pr(X = x \,|\, Y = k)$ is **the density function** 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.

Linear Discriminant Analysis





- 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

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

- Here μ_k is the mean, and σ_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$p_k(x) = Pr(Y = k | X = x)$:

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

- Happily, there are simplifications and cancellations.

Discriminate functions

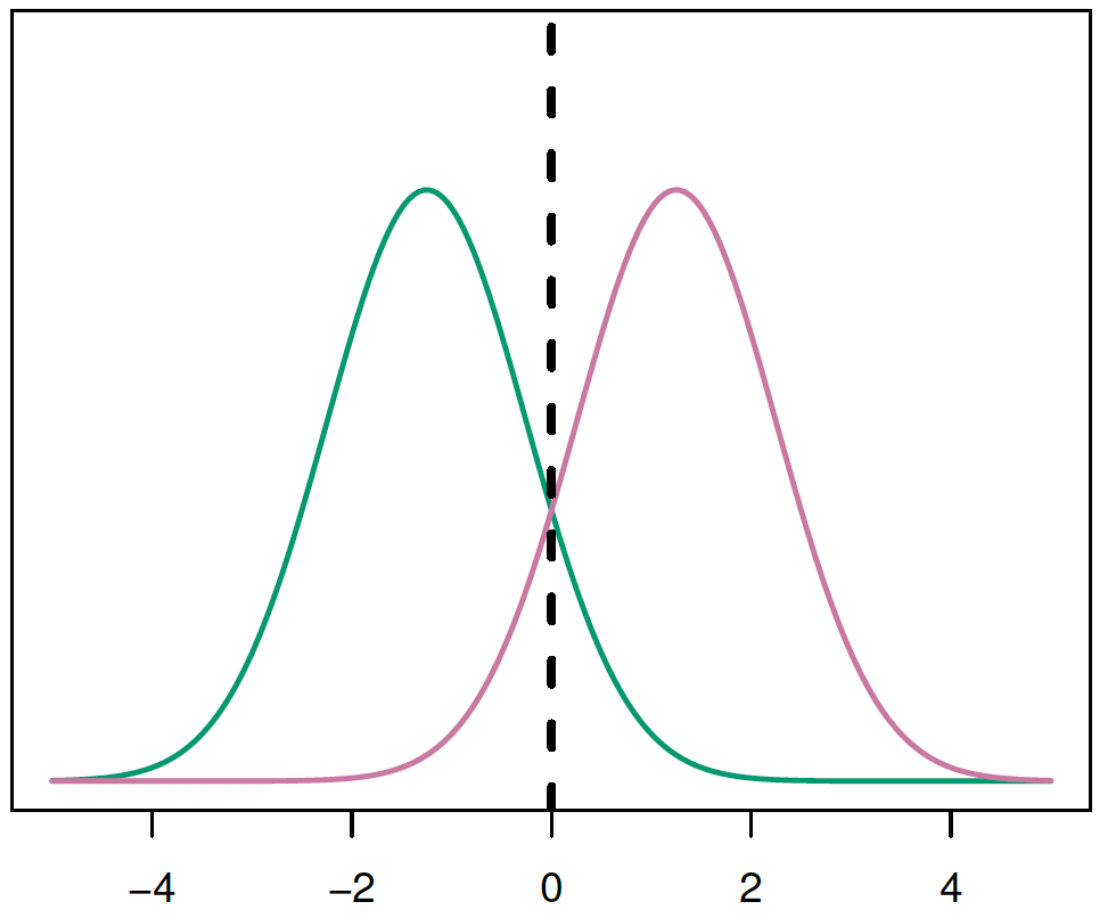
- To classify at the value $X = x$, we need to see which of the $p_k(x)$ is largest. Taking logs, and discarding terms that do not depend on k , we see that this is equivalent to assigning x to the class with the largest discriminant score:

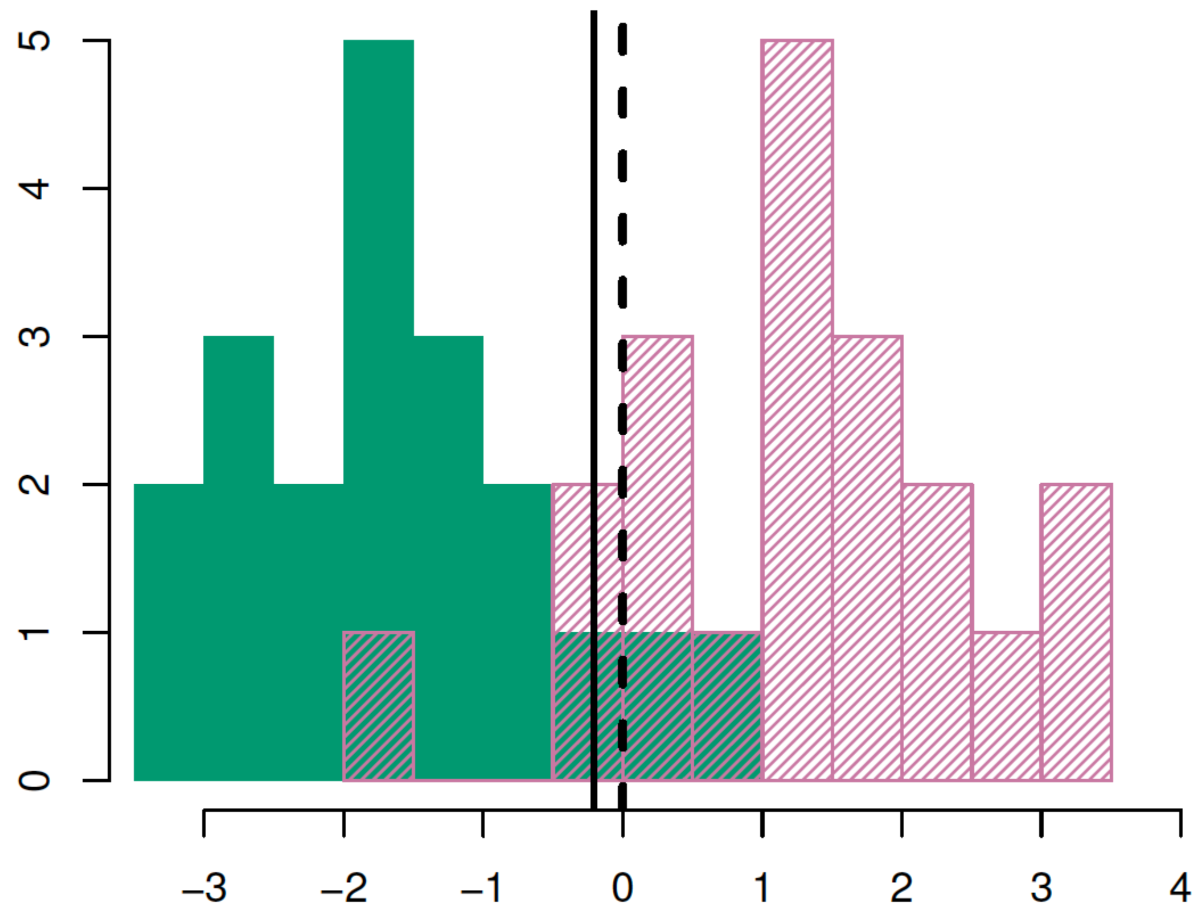
$$\lambda_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + log(\pi_k)$$

- Note that $\lambda_k(x)$ 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

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





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.

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

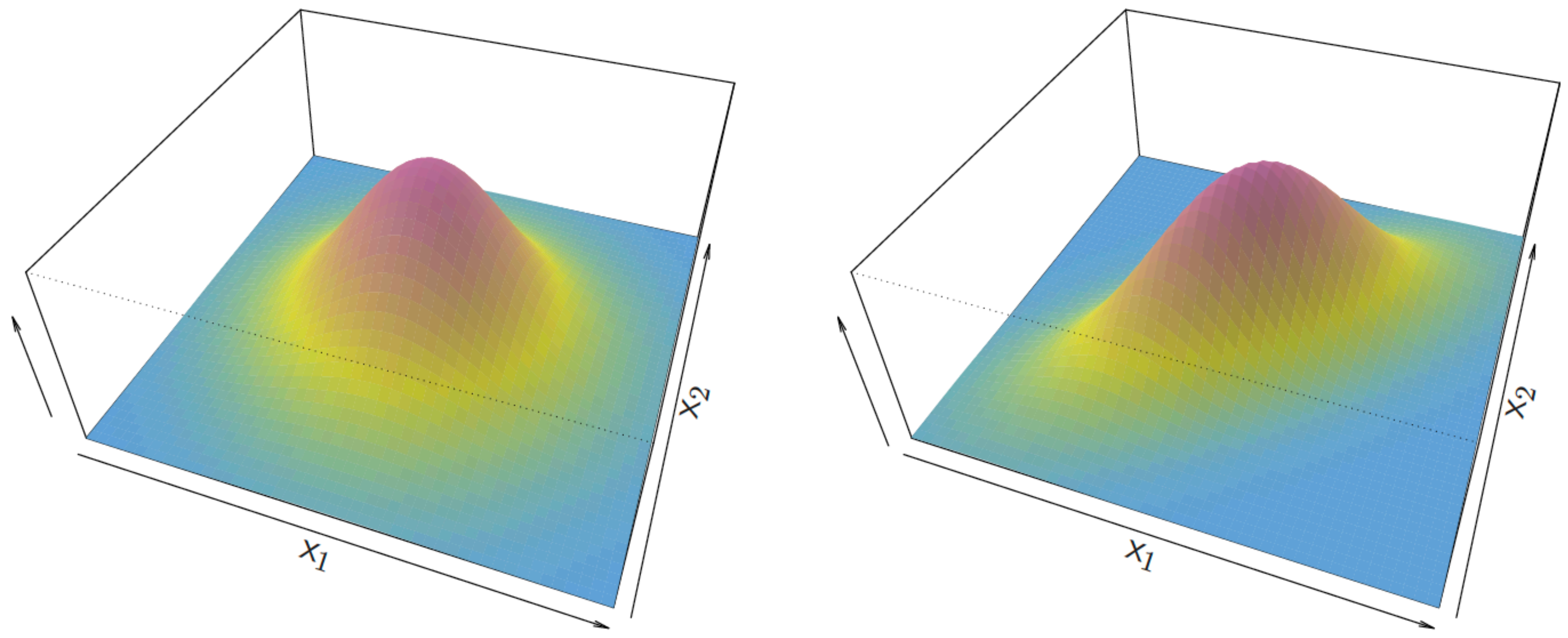
$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

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

Where

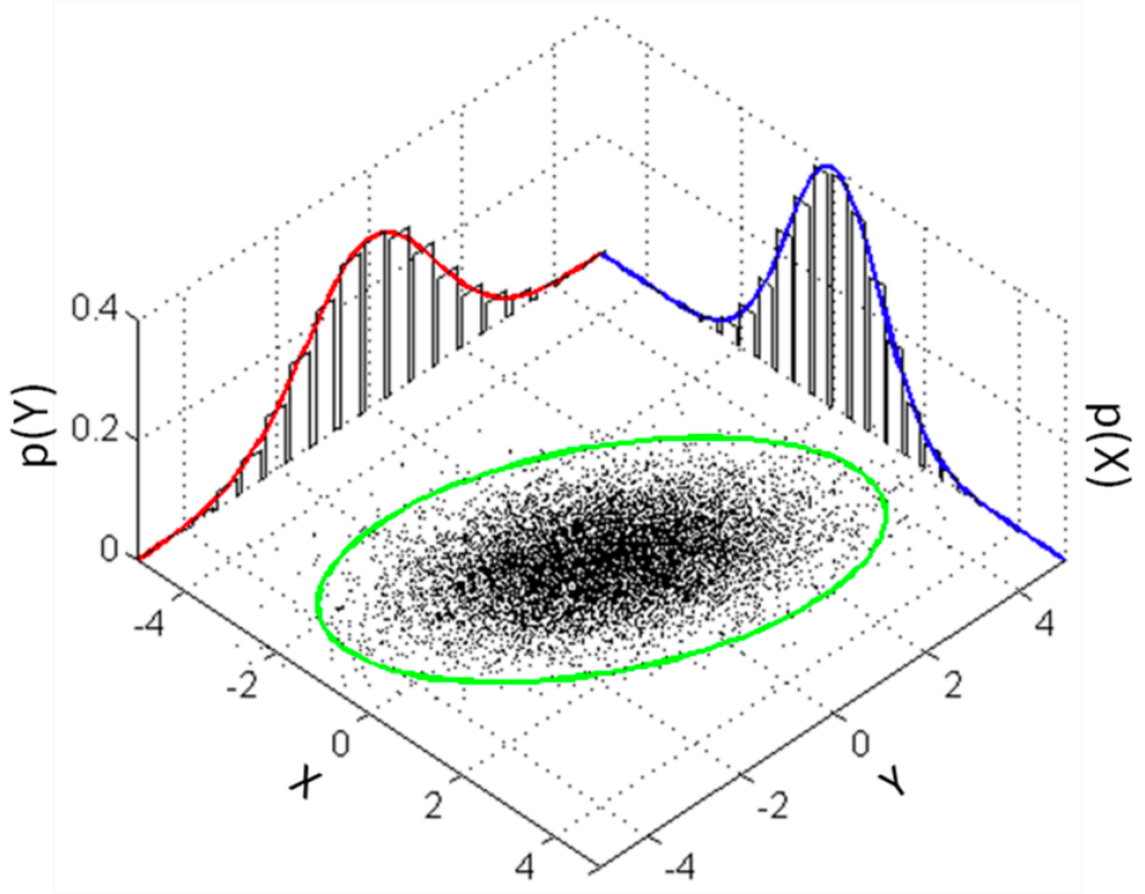
$$\hat{\sigma}_k^2 = \frac{1}{n_k - 1} \sum_{:y_i=k} (x_i - \hat{\mu}_k)^2$$

is the usual formula for the estimated variance in the k th class

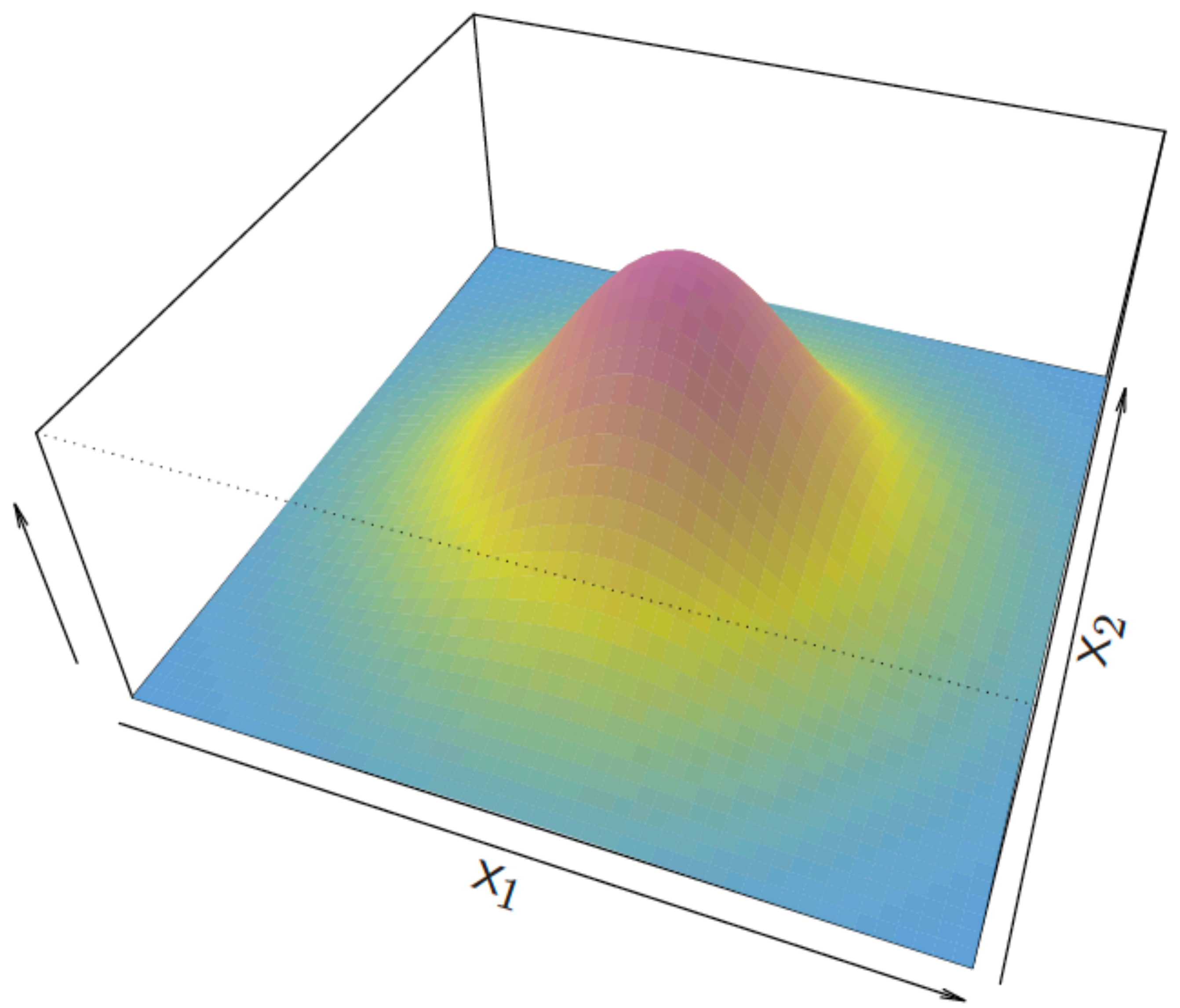


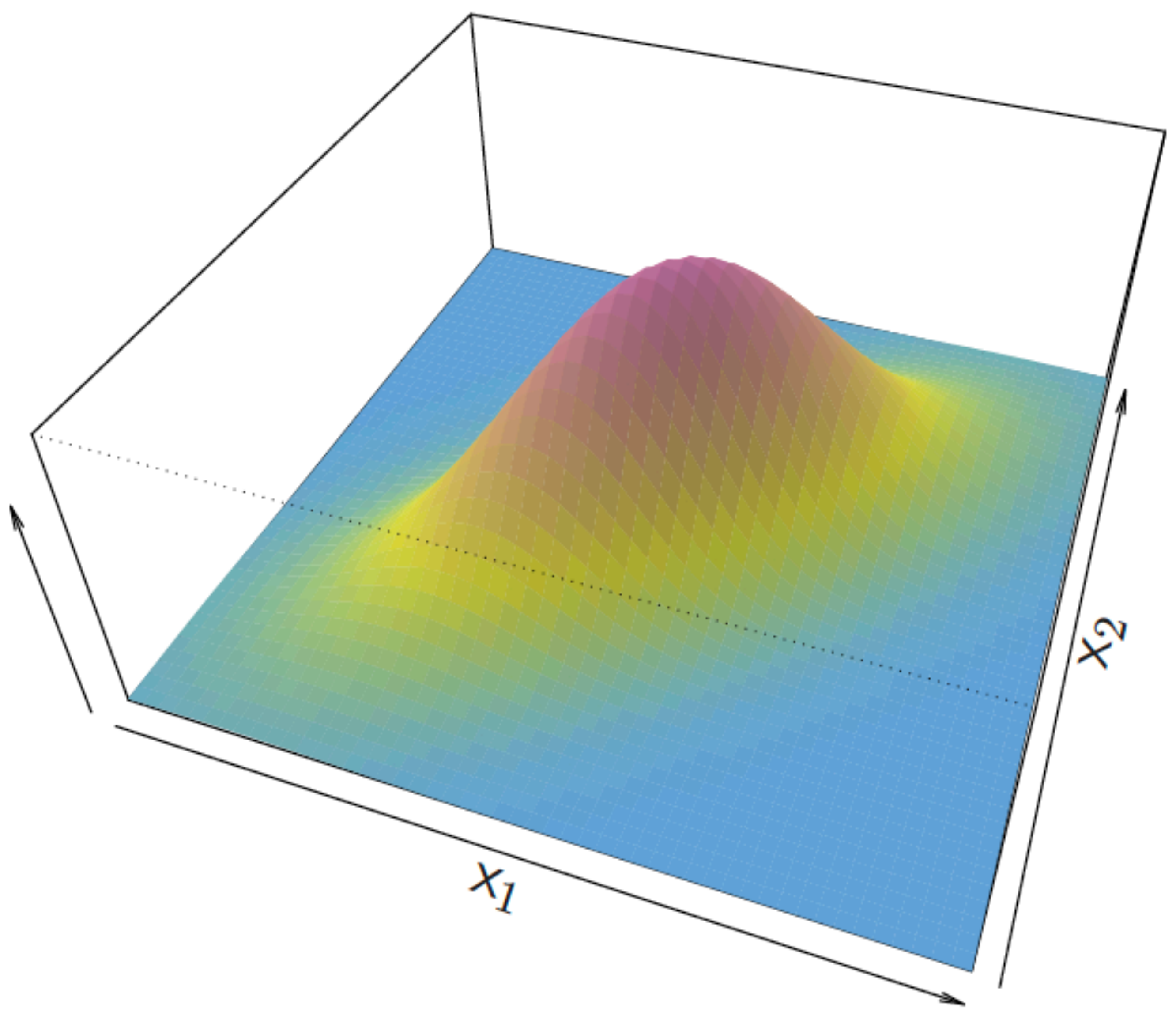
To extend the LDA classifier to the case of multiple predictors we will assume that $X = (X_1, X_2, \ldots, X_p)$ is drawn from a multivariate Gaussian distribution, with a class-specific mean vector and a common covariance matrix Σ.

It is defined as $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.



Many sample points from a multivariate normal distribution with $\boldsymbol{\mu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 3/5 \\ 3/5 & 2 \end{bmatrix}$, shown along with the 3-sigma ellipse, the two marginal distributions, and the two 1-d histograms.



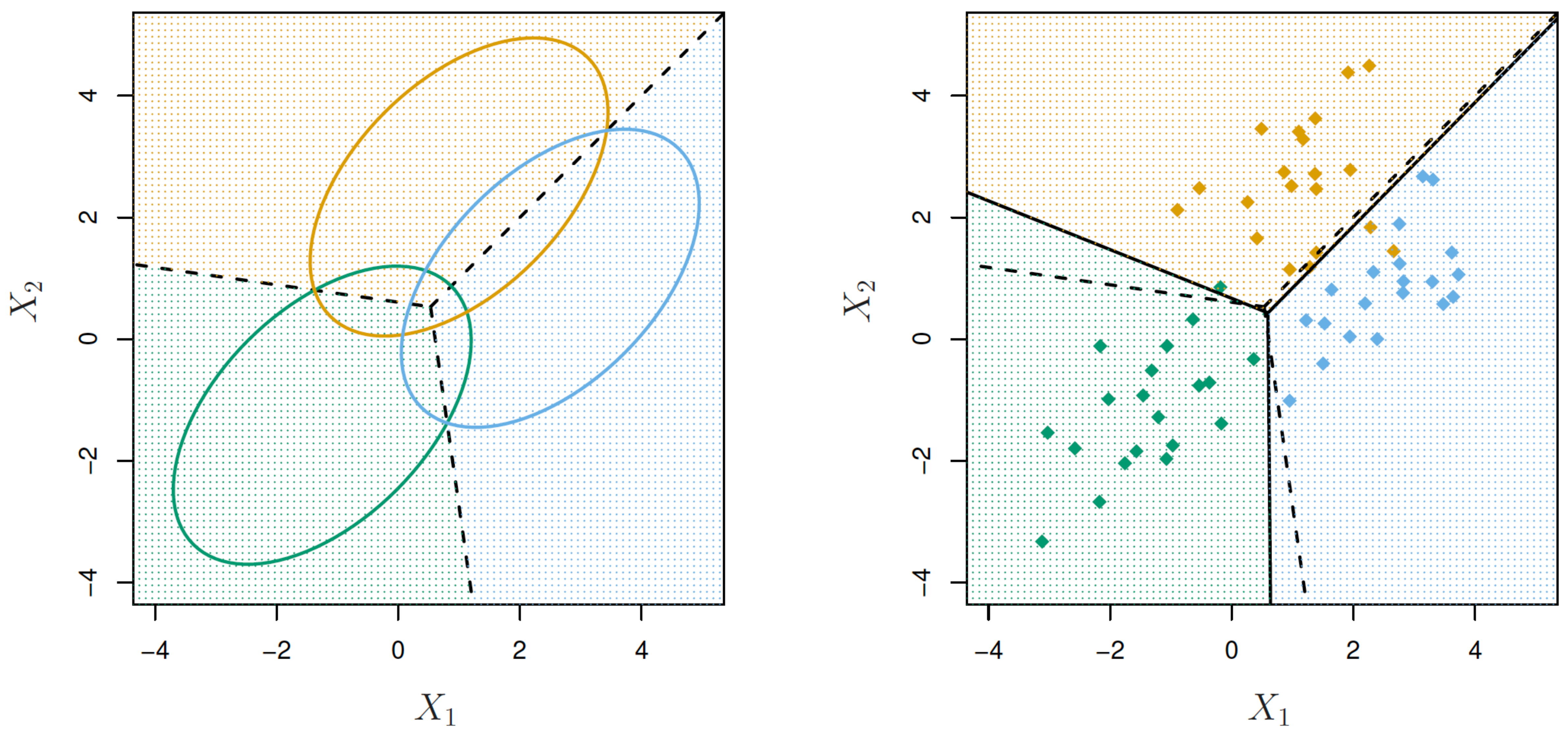


Density function: $f(x) = \dfrac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$

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

Despite its complex form, $\delta_k(x) = c_{k0} + c_{k1}x_1 + c_{k2}x_2 + \ldots c_{kp}x_p$ **is a linear combinations**

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. Where they known, they would yield the fewest misclassification errors, among all possible classifiers.