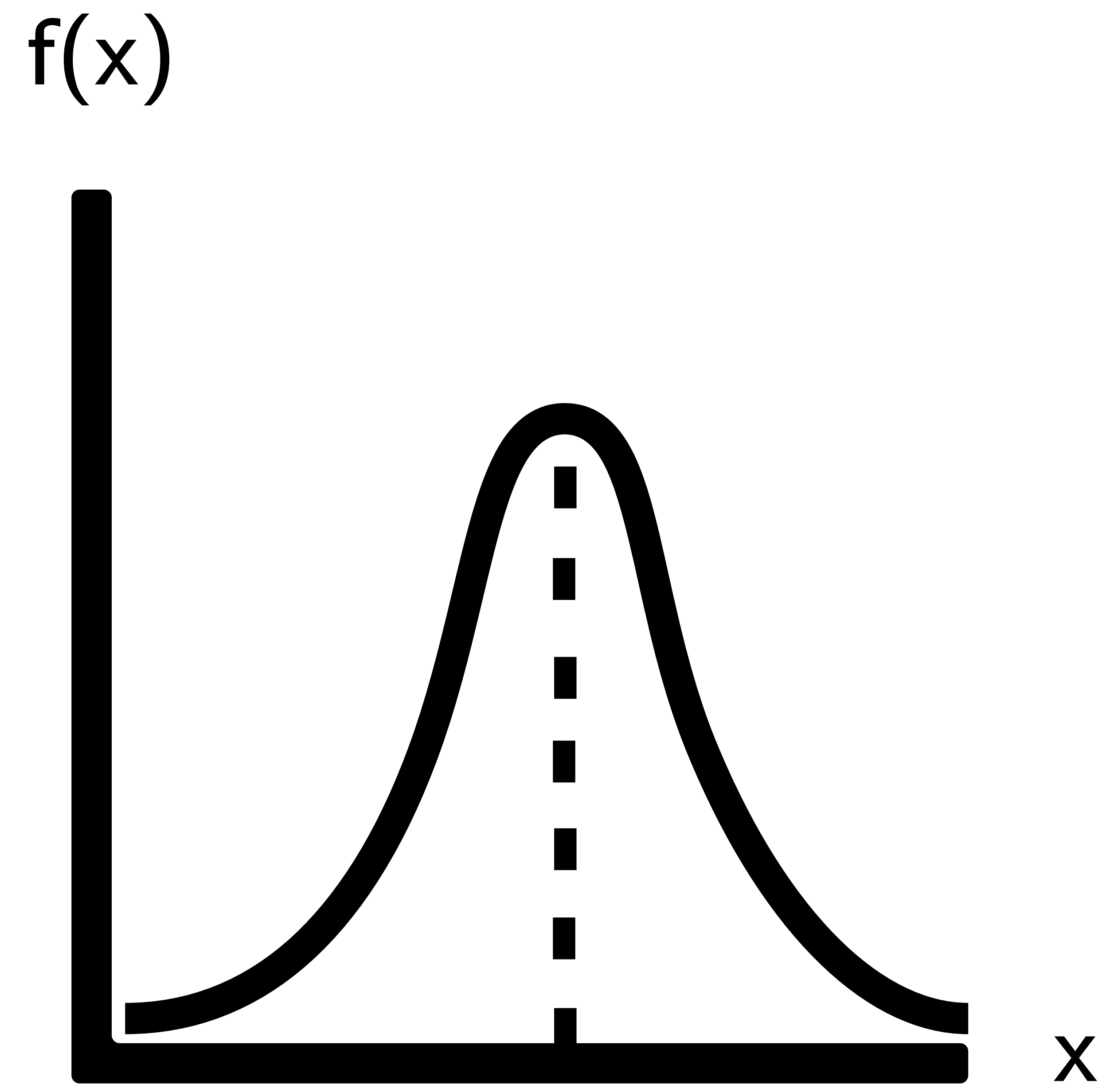
Lecture 5: Classification Problems

Francisco Rosales, PhD.
Maastricht University | Machine Learning for Public Policy

February 14th 2023

Probability Distribution Functions

Definition (Gaussian Random Variable): is a real valued random variable with probability distribution function given by:



$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

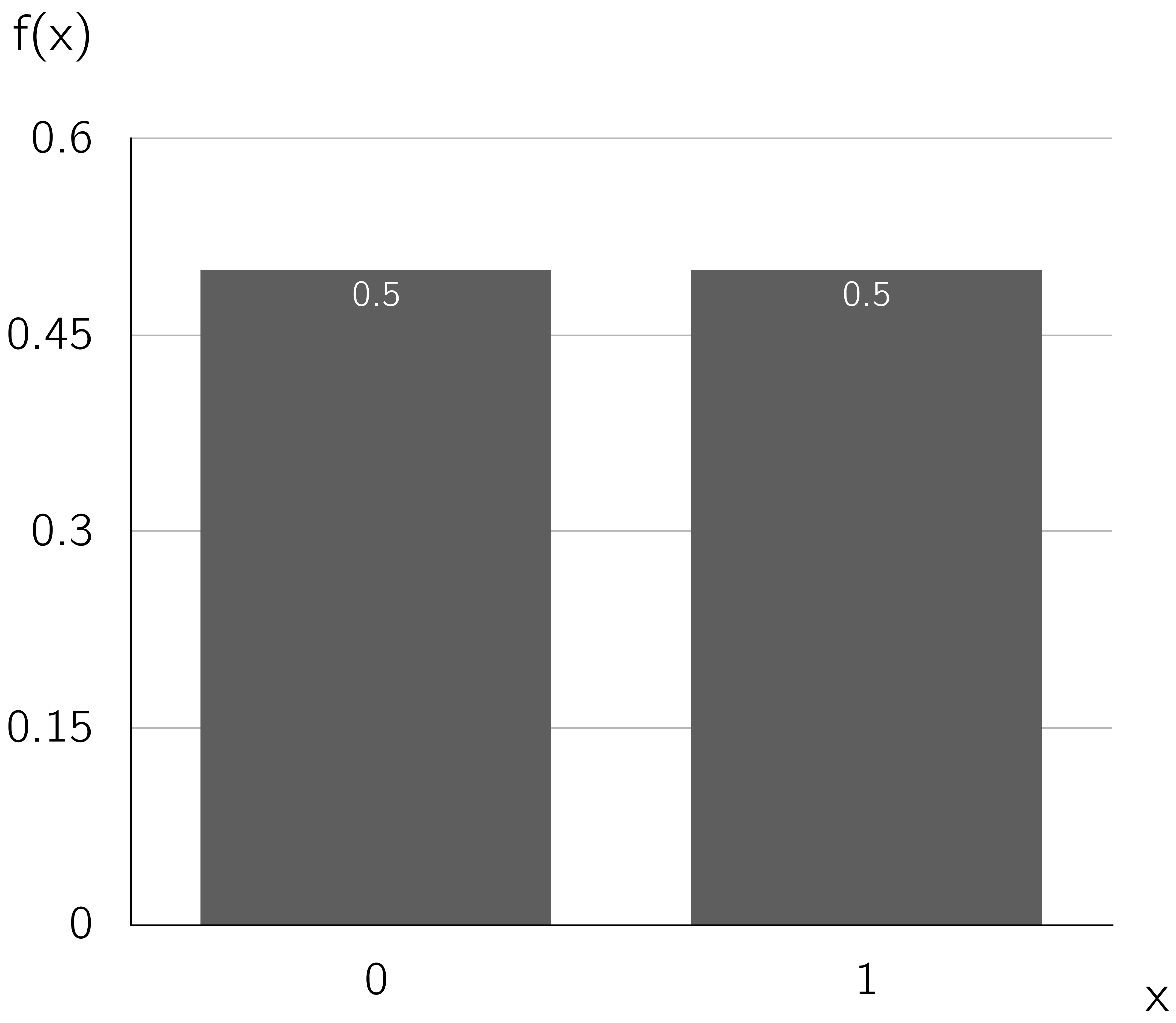
$$\mathbb{E}[x] = \mu \qquad Var[x] = \sigma^2$$

Definition (Bernoulli Random Variable): can only take two values: success (1) and failure (0). The probability of success is “p”, and the probability of failure “1-p”. Formally we write $x \sim \text{Bernoulli}(p)$

Its probability function is given by

$$f(x) = \begin{cases} 1 - p & x = 0 \\ p & x = 1 \end{cases}$$

Bernoulli Random Variable



Bernoulli Random Variable

Expected value

$$\mathbb{E}[x] = \sum_{x \in \{1,0\}} x f(x)$$

$$= 1 \times p + 0 \times (1 - p) = p$$

Variance

$$\mathrm{Var}[x] = \sum_{x \in \{1,0\}} (x - \mathbb{E}[x])^2 f(x)$$

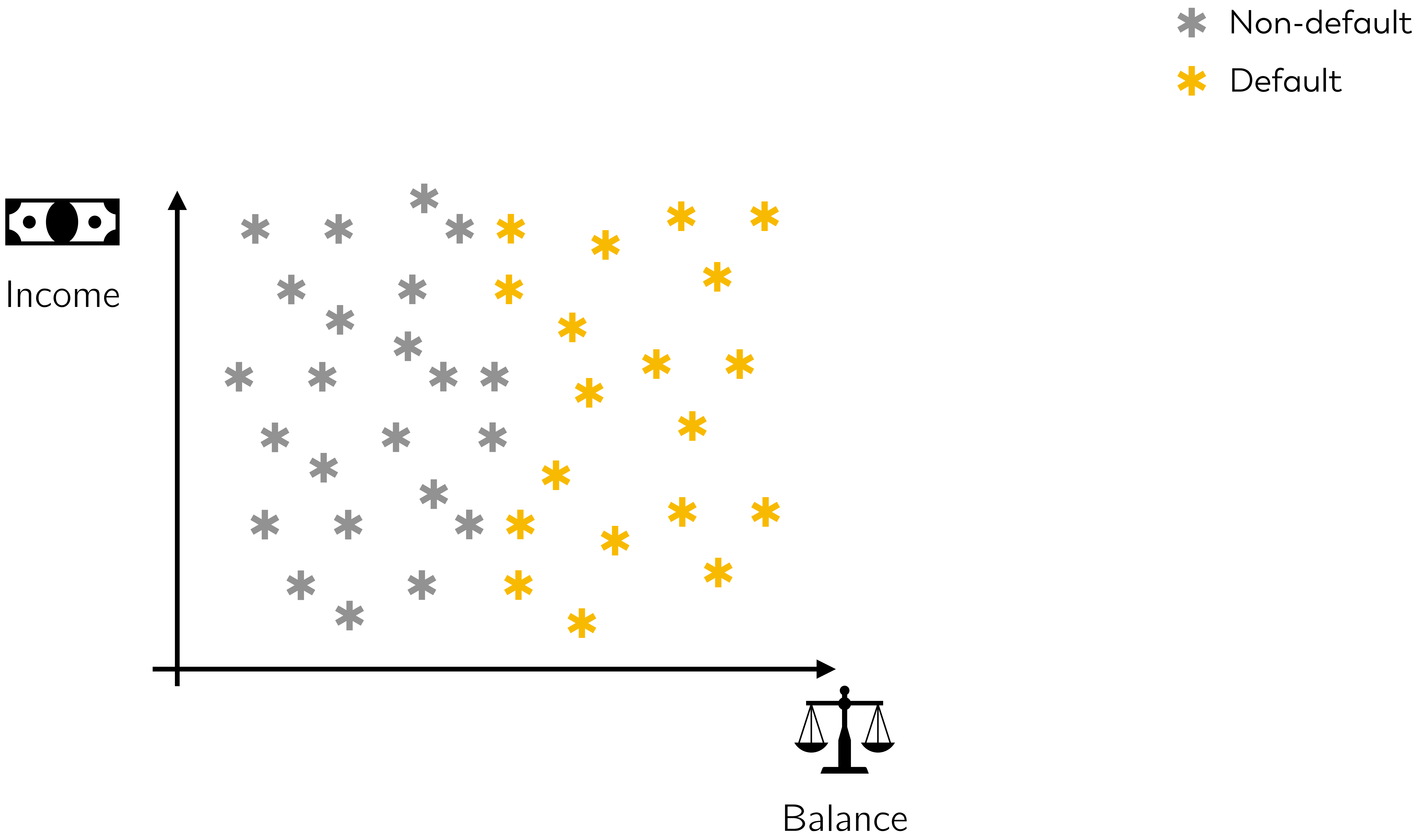
$$= (1 - p)^2 \times p + (0 - p)^2 \times (1 - p) = p \times (1 - p)$$

Logit Model

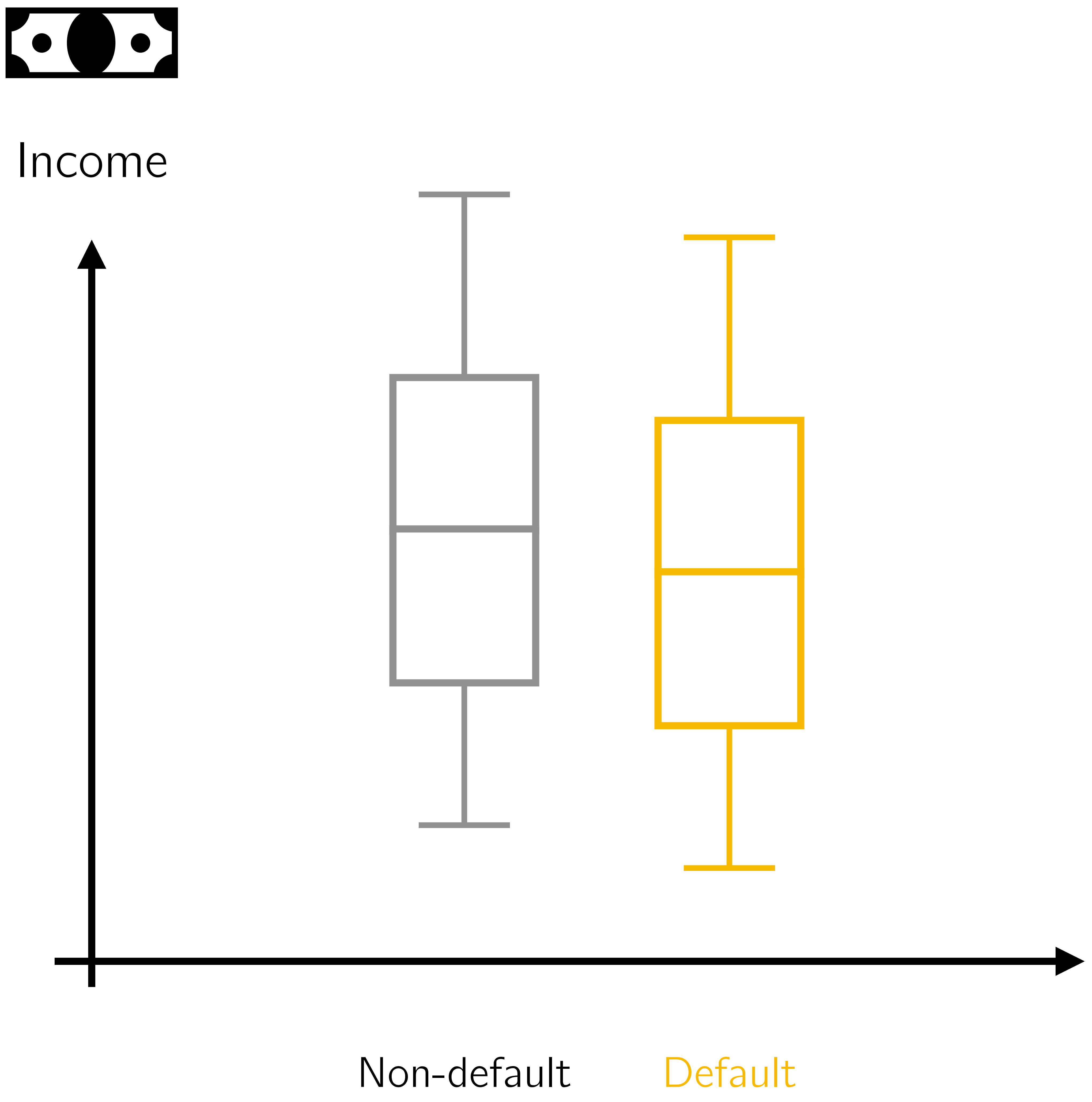
Example (Credit card Payment): Consider the problem of classifying customers in "default" and "non-default" using their annual income (`income`) and their monthly credit card balance (balance) as predictors.

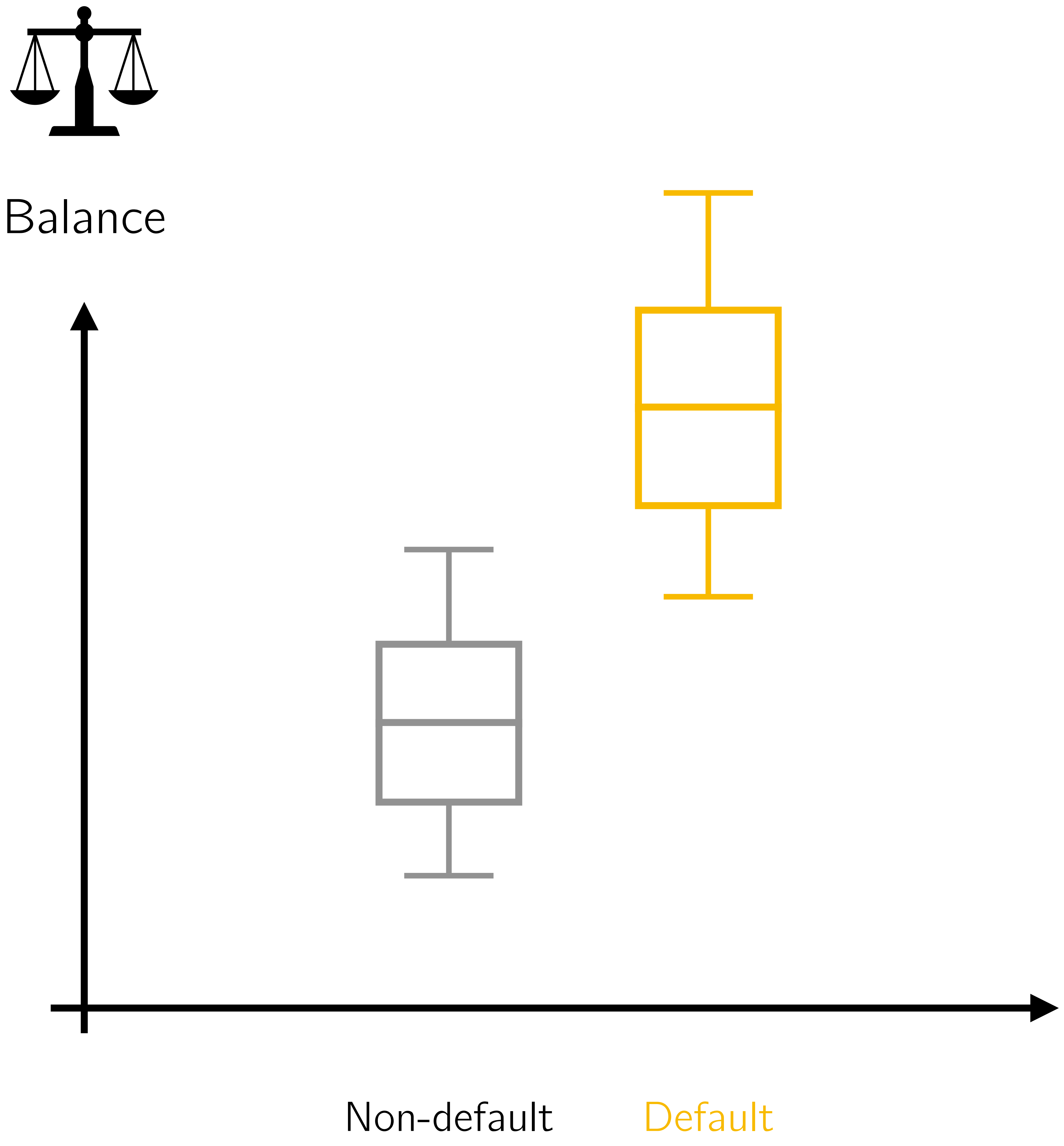
Note that:

This problem is very similar to the regression problem, with the only difference that in this case the variable to be predicted is categorical.


Non-default
Default
Income
Balance

Classification Logit





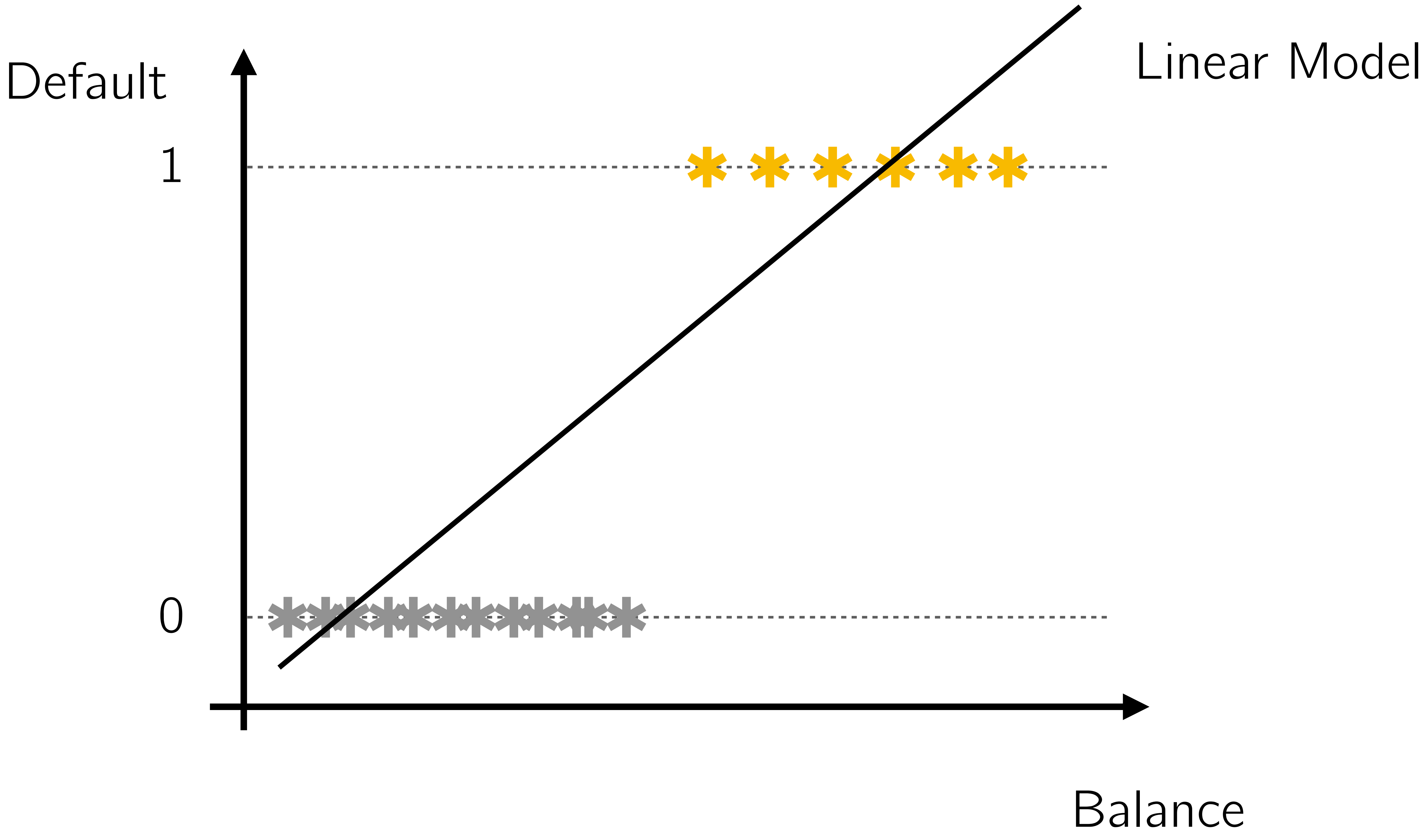
Question: if you had to choose between "balance" and "income" to explain probability of default, which variable would you select?

Lets focus on the univariate case (the extension to multivariate is easy). In our example consider "balance" as the only predictor variable and "default" as the response.

Note that:

The response variable is a random variable that takes values 0 (no-default) and 1 (default). Thus, a prediction of 0.7 or 1.4 makes no practical sense.

Classification Logit



$f(x_i) = \beta_0 + \beta_1 x_i$

Linear Model

$$y_i \sim \mathcal{N}(f(x_i), \sigma_\epsilon^2)$$

$$\mathbb{E}[y_i] = f(x_i) = \beta_0 + \beta_1 x_i$$

$$\hat{f}(x_i) = b_0 + b_1 x_i$$

Minimizes mean squared error

$$b_0, b_1$$

$f(x_i) = \beta_0 + \beta_1 x_i$

Logit Model

$$y_i \sim \text{Bernoulli}(p_i)$$

$$\mathbb{E}[y_i] = p_i = g(f(x_i)), \quad g(z) = \frac{e^z}{1 + e^z}$$

$$\hat{p}_i(x_i) = \frac{e^{b_0 + b_1 x_i}}{1 + e^{b_0 + b_1 x_i}}$$

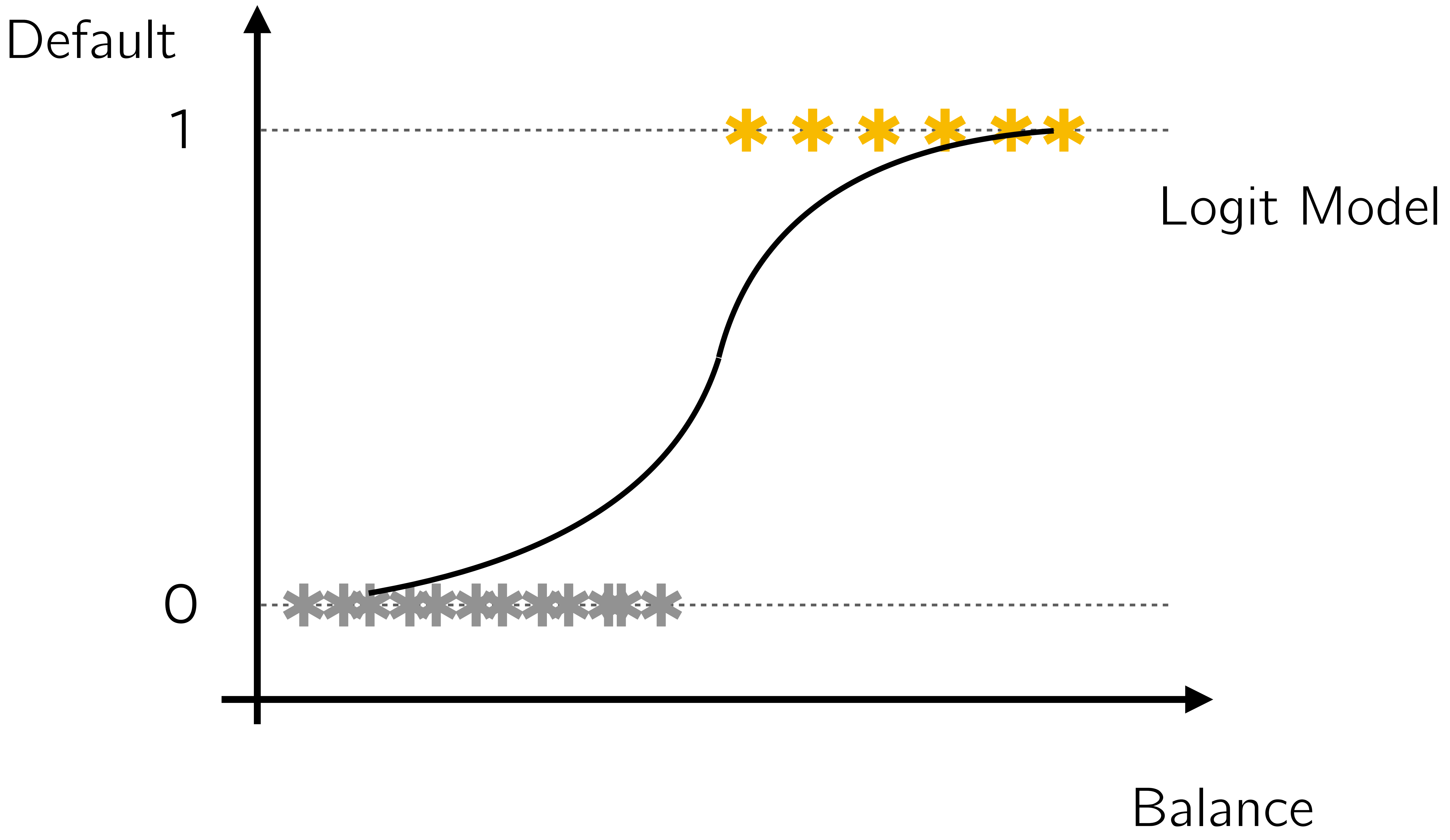
Maximizes log-likelihood

$$b_0, b_1$$

Note that:

There are many ways to bound the values of f(x_i) in [0,1]. The **logit model** get its name from function “h” (logit function), h(z) = log(z/(1-z)), which is the inverse of function “g”.

Classification Logit



Question: Suppose you know that the best model for our example is b_0 = - 10 and b_1 = 1. Fill in the following table. How would you make a default/non-default prediction?

Default (y)	Balance (x)	f(x)	Default probability g(x)	Prediction (y hat)
0	5			
1	10			
1	20			

Can compute the default probability, but need some kind of rule to make the prediction. Say, if g(x) is greater or equal to 0.5, we predict “default” for that observation.

Default (y)	Balance (x)	f(x)	Default probability g(x)	Prediction (y hat)
0	5	-5	0.0067	0
1	10	0	0.5	1
1	20	10	0.9999	1

How are the b's selected?

- Each y_i is 0 or 1, that is, y_i is Bernoulli(p_i).
- We can ask ourselves what is the probability that y_i = 1?
- Since y_i es Bernoulli, this probability is p_i. If y_i was 0, this probability would be 1 - p_i.

Question: complete the table

Default (y)	Balance (x)	f(x)	Default probability g(x)	Probability of observing y
0	5	-5	0.0067	
1	10	0	0.5	
1	20	10	0.9999	

The probability of observing each y_i is given by

Default (y)	Balance (x)	f(x)	Default probability g(x)	Probability of observing y
0	5	-5	0.0067	0.9933
1	10	0	0.5	0.5
1	20	10	0.9999	0.9999

Question: What is the probability of observing y_1=0, y_2=1 e y_3=1 in a sample? Consider that each observation is independent from each other. Can you find a pair of values b_0 y b_1 that allow us to increase this probability?

Default (y)	Balance (x)	f(x)	Default probability g(x)	Probability of observing y
0	5	-5	0.0067	0.9933
1	10	0	0.5	0.5
1	20	10	0.9999	0.9999

The probability of observing y_1 = 0, y_2 = 1 e y_3 = 1 is: 0.9933 x 0.5 x 0.9999 = 0.4966.

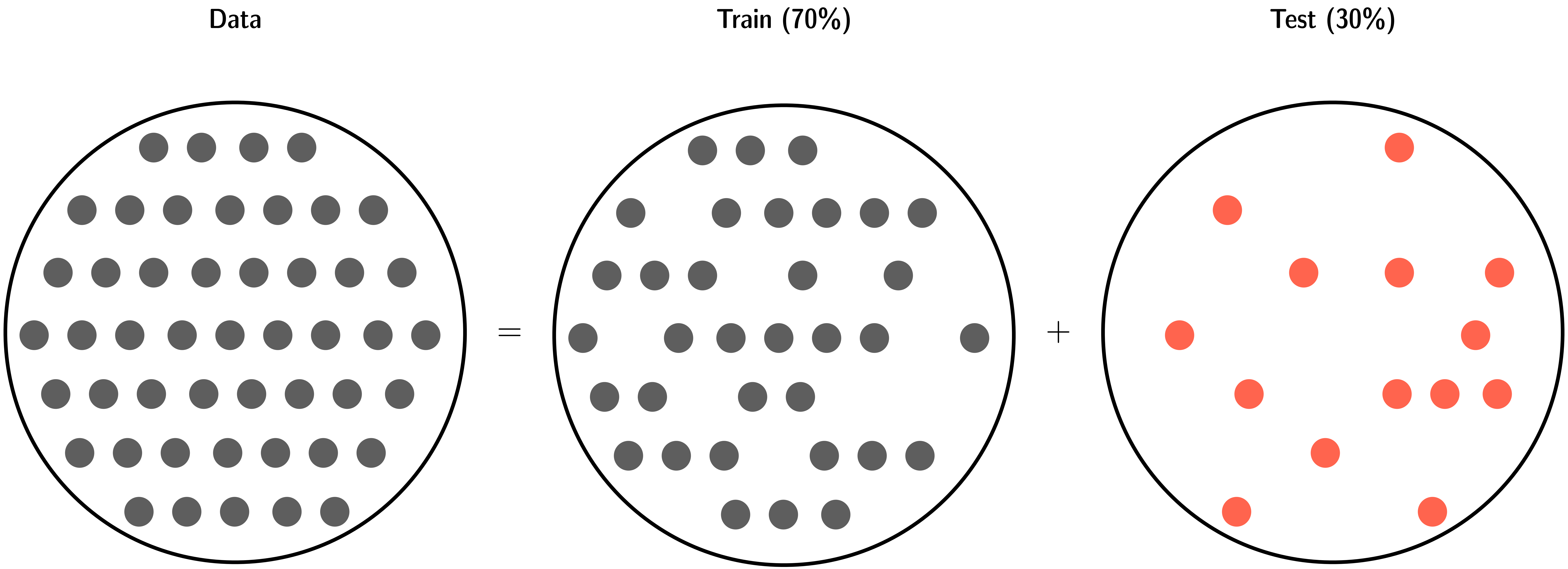
This probability is known as **likelihood**. In fact, this value can be increased to 0.9179 selecting, for example, b_0 = - 10, b_1 = 1.5. For numerical reasons is better to optimize for the **log-likelihood.**

Note that:

- Estimator b_1 is not the marginal contribution of X over p(X), but the marginal contribution of X over the “log-odds”, i.e. log(p_i/(1-p-i)).
- An important performance measure in classification models is the **error rate**, defined as: number of misclassified cases/ total number of cases.

Performance Metrics

Train-Test Split



Note that:

The training data is used to "fit" the models (find the b's). The test data is used to evaluate the performance of the model in practice.

True Condition

Predicted Condition	P	N
P	True P	False P (type I error)
N	False N (type II error)	True N

Performance Metrics

True Condition

Predicted Condition	P	N
P	TP	FP
N	FN	TN

$$\text{Error Rate} = \frac{FP + FN}{FP + FN + TP + TN}$$

Performance Metrics

True Condition

Predicted Condition	P	N
P	TP	FP
N	FN	TN

$$\text{Accuracy} = \frac{TP + TN}{FP + FN + TP + TN}$$

Maximizing accuracy is equivalent to minimizing error rate.

True Condition

Predicted Condition	P	N
P	TP	FP
N	FN	TN

$$\boxed{\text{Precision} = \frac{TP}{TP + FP}}$$

Maximizing precision is equivalent to minimizing FP.

True Condition

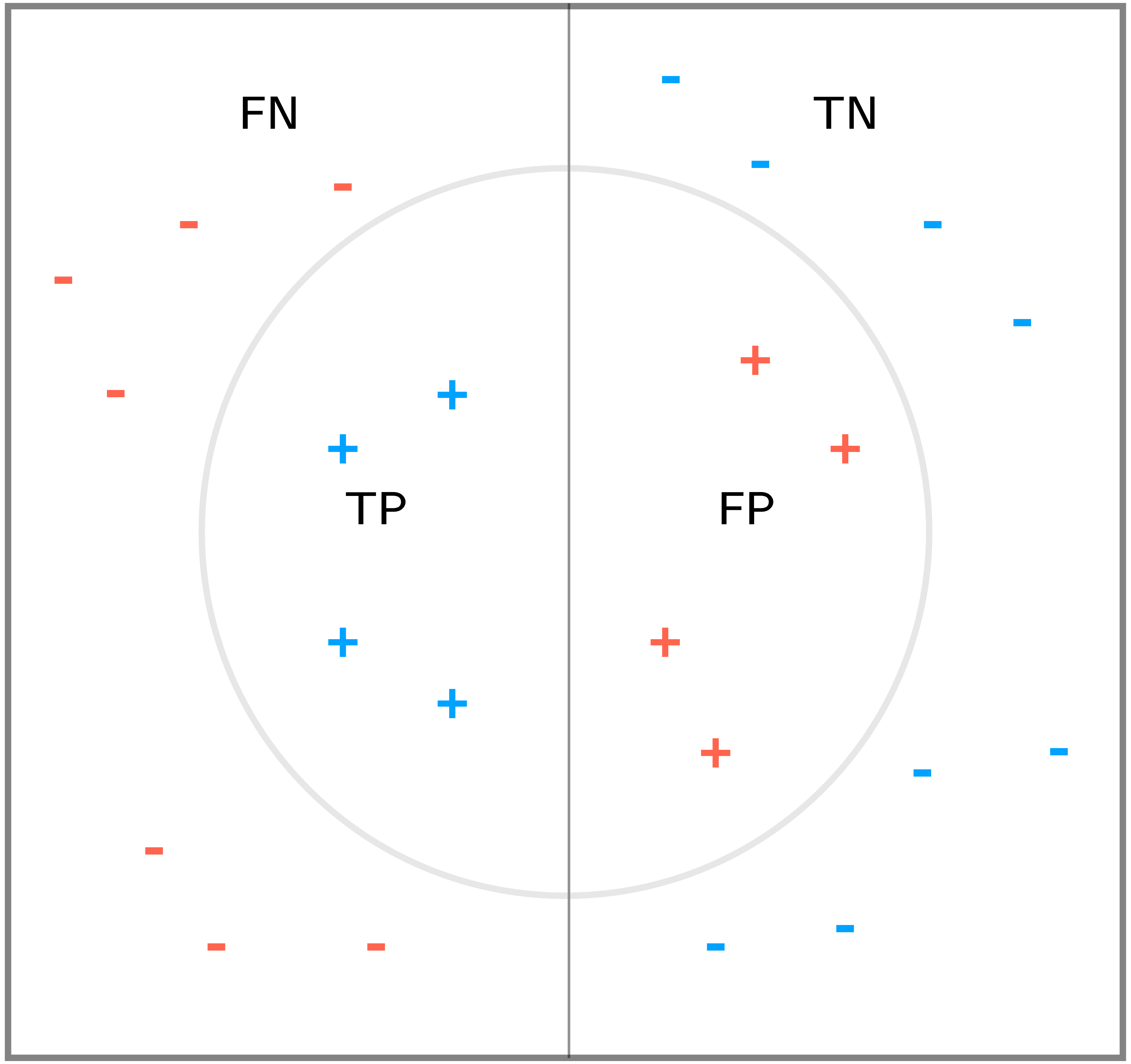
Predicted Condition

	P	N
P	TP	FP
N	FN	TN

$$\boxed{\text{Recall} = \frac{TP}{TP + FN}}$$

Maximizing recall is equivalent to minimizing FN.

Performance Metrics



Recall =

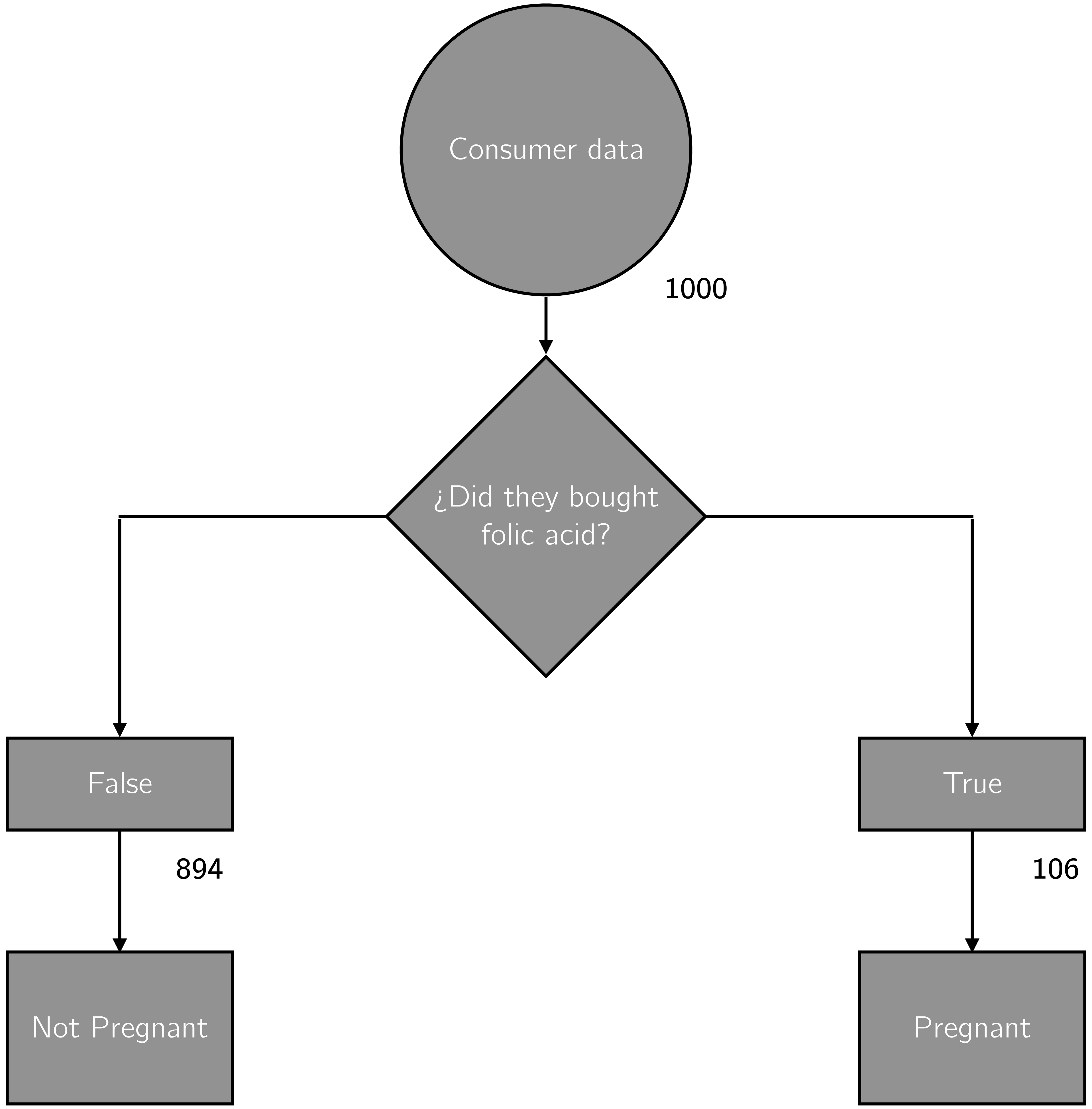
Precision =

Tree-based Methods

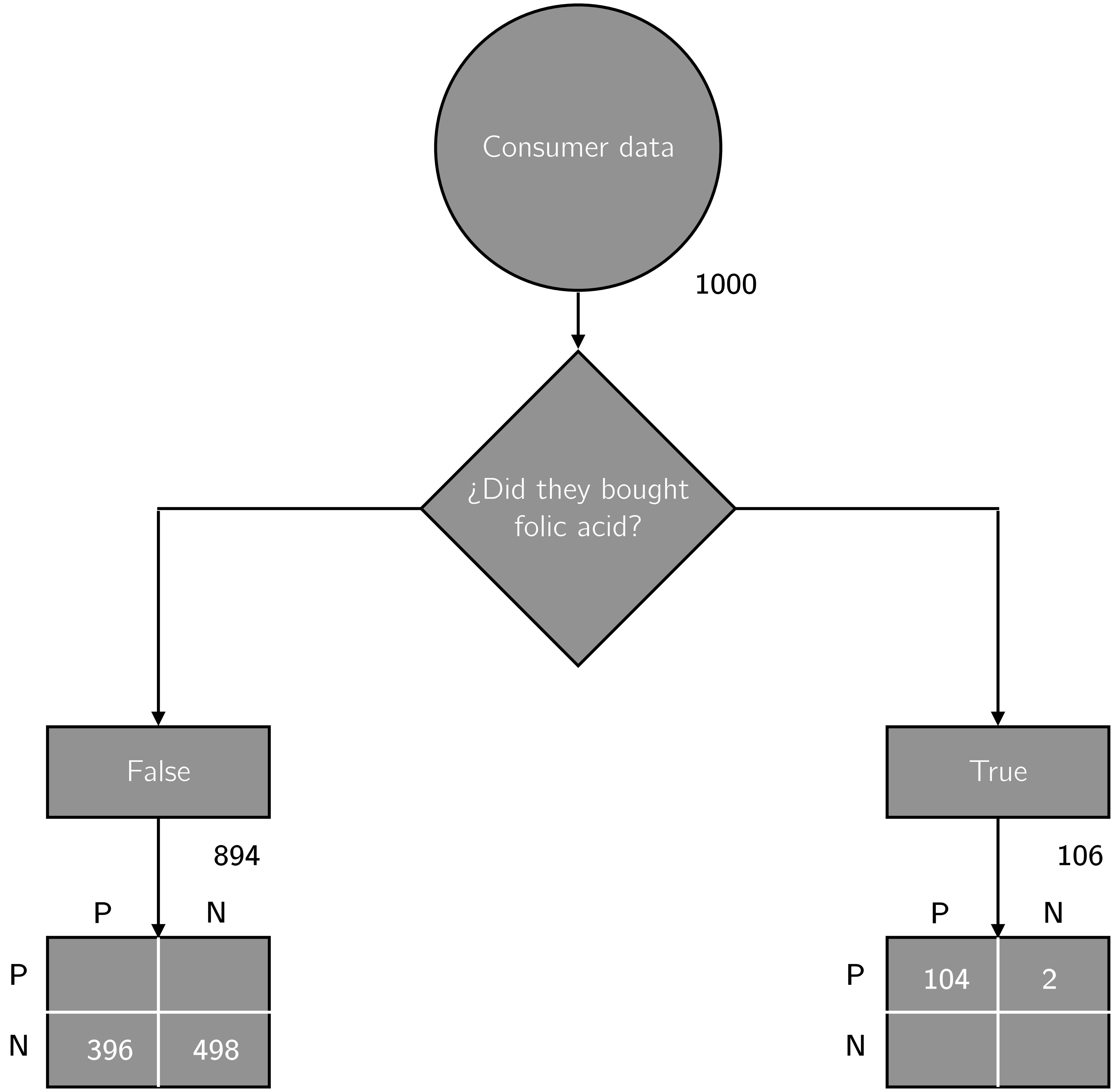
Question: Consider the following ranking problems, and indicate which performance metric you find most appropriate:

(a) Pregnant classifier (pregnant = 1, non-pregnant = 0)
(b) Covid classifier (covid = 1, non-covid = 0)
(c) Image (chihuahua =1, muffin = 0)

Classification Trees



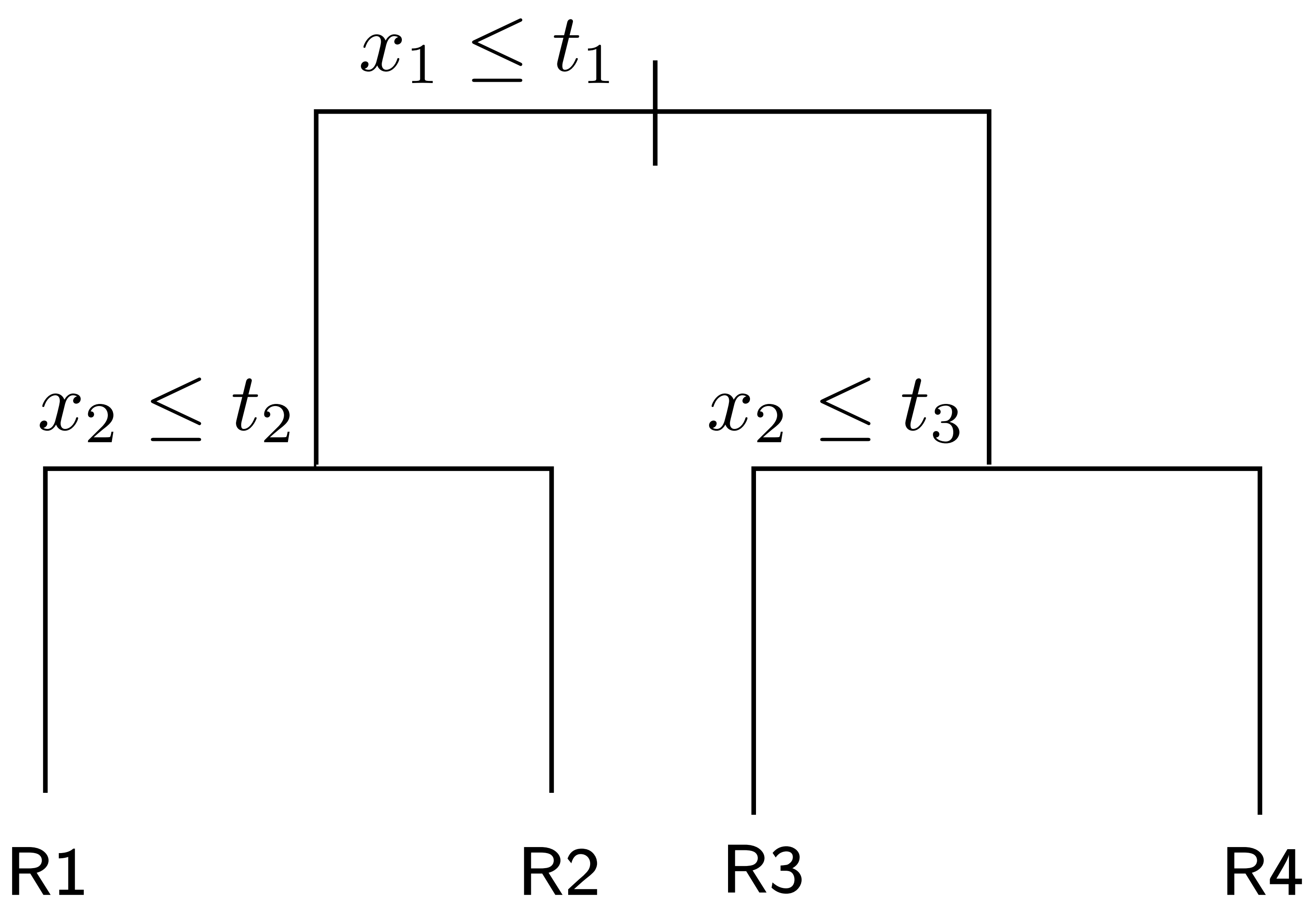
Classification Trees

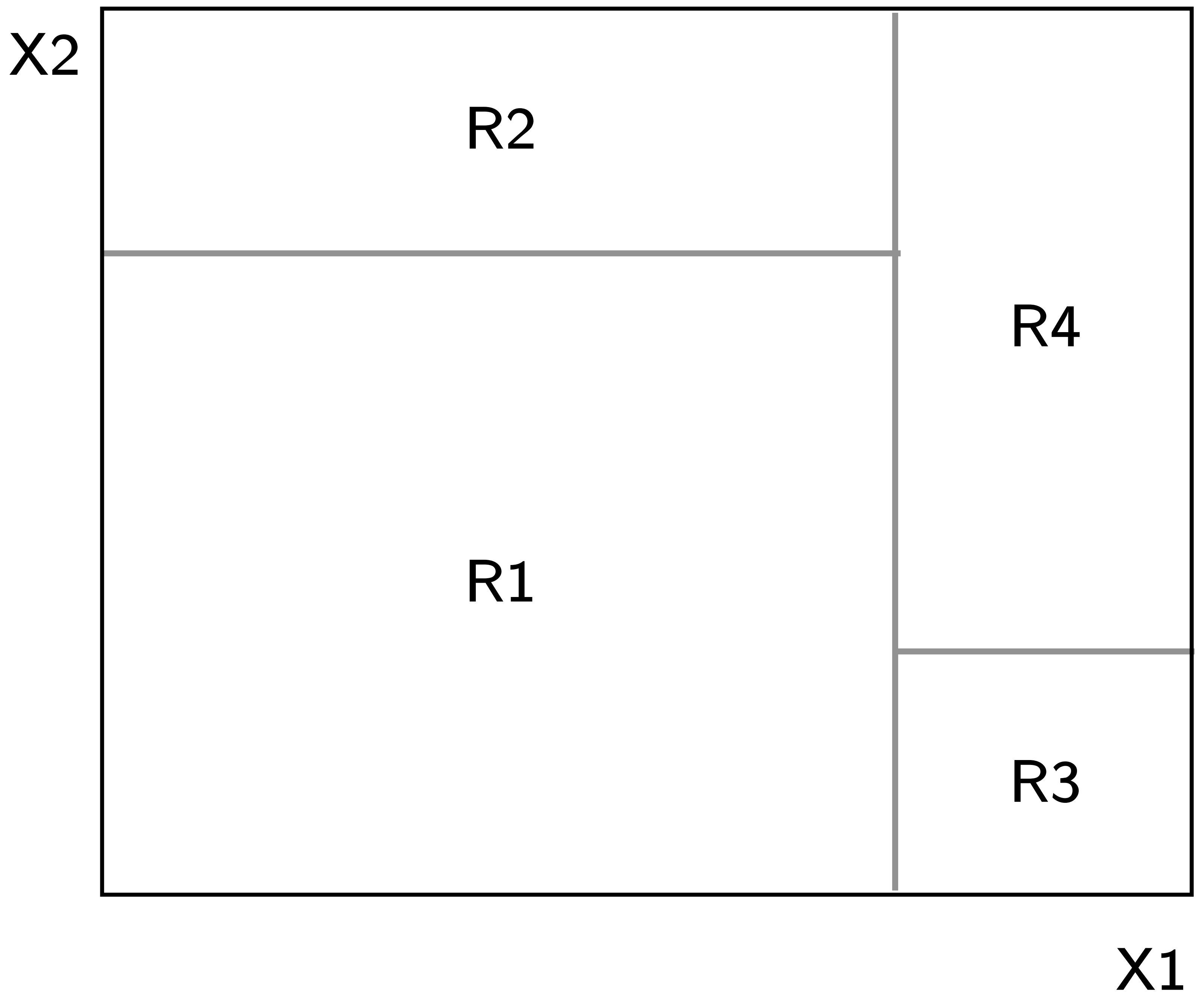


		True Condition	
		P	N
Predicted Condition	P	104	2
	N	396	498

Classification Trees

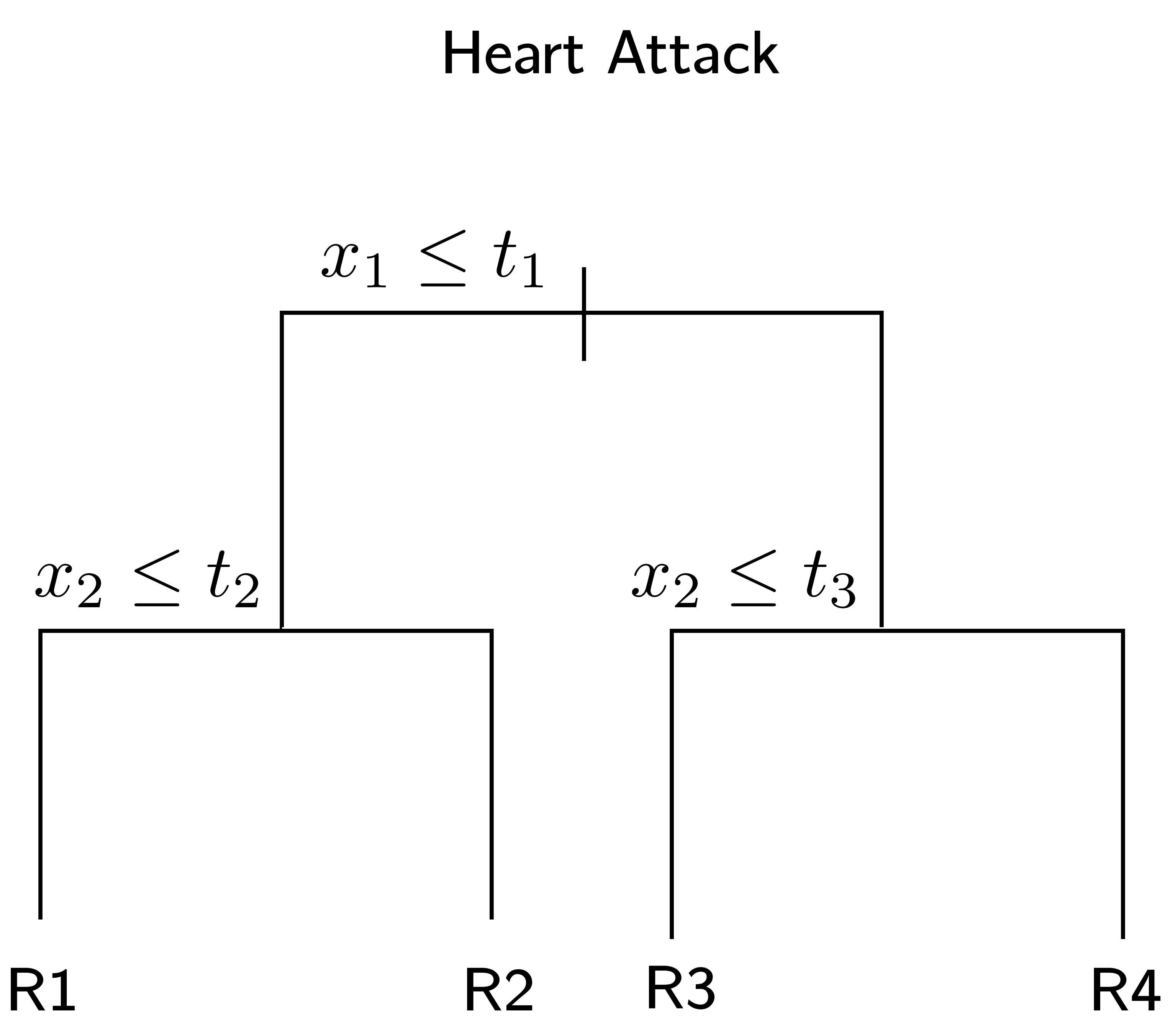
Heart Attack

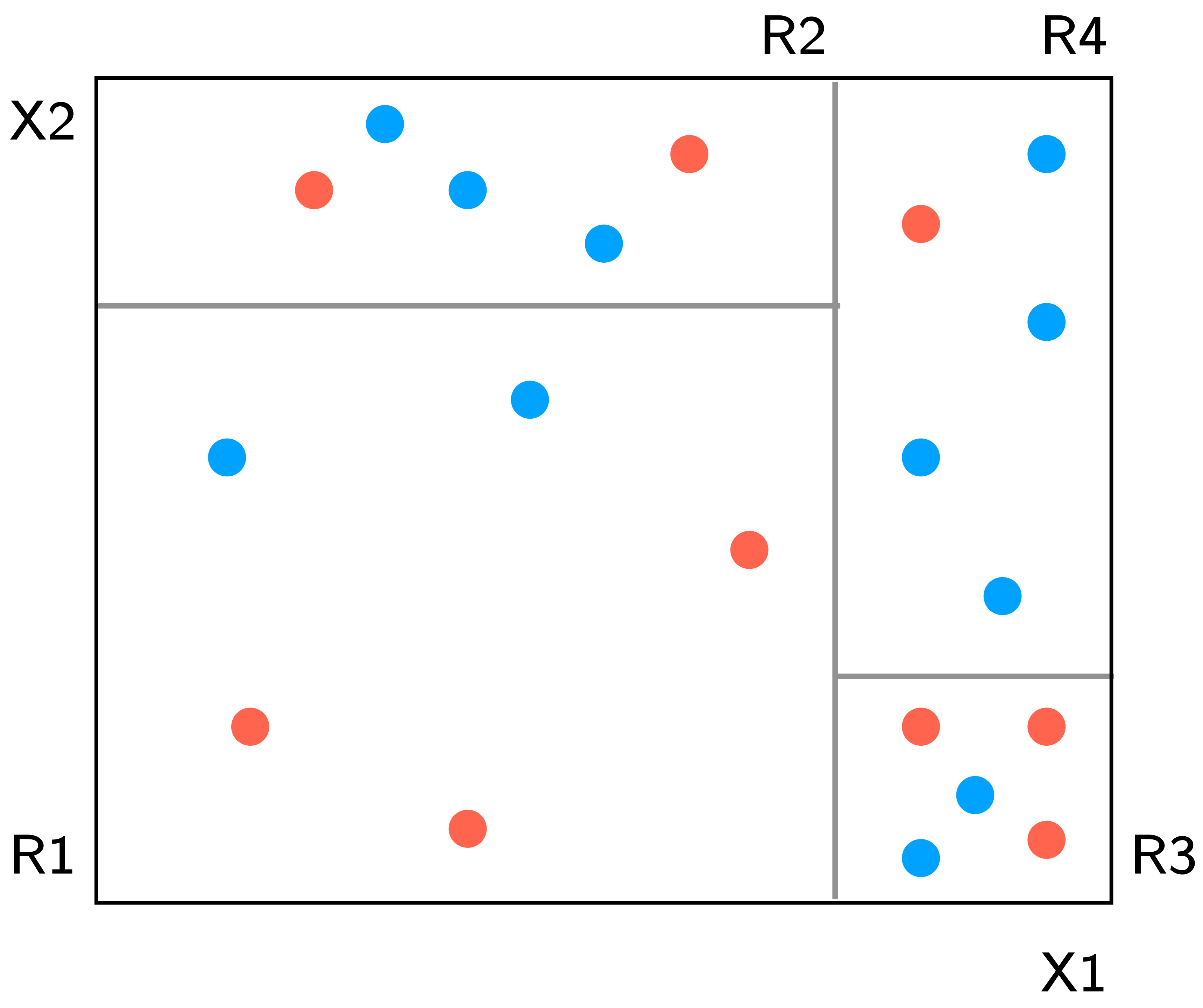




x_1: Body fat percentage
x_2: Average steps per week

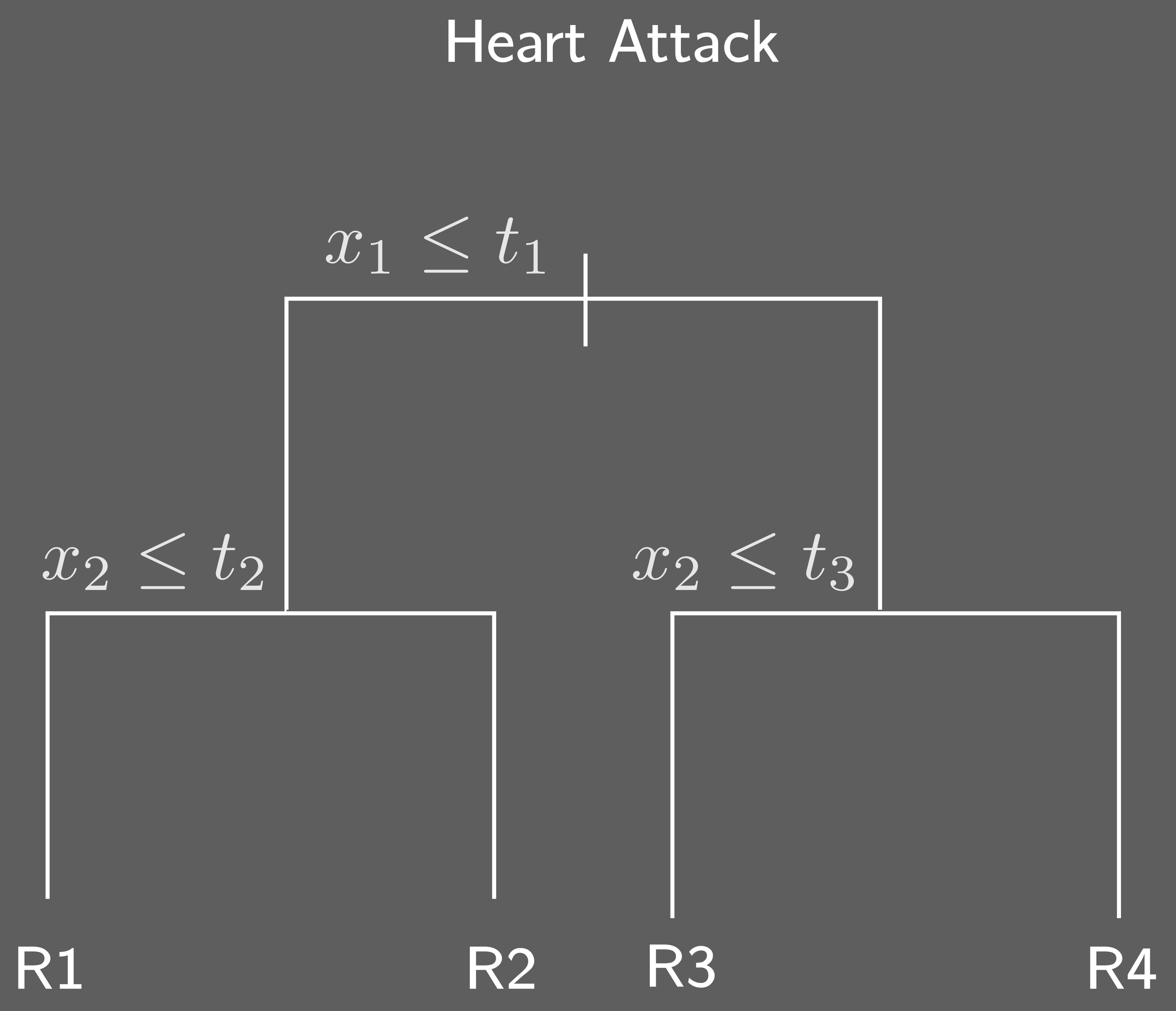
Classification Trees

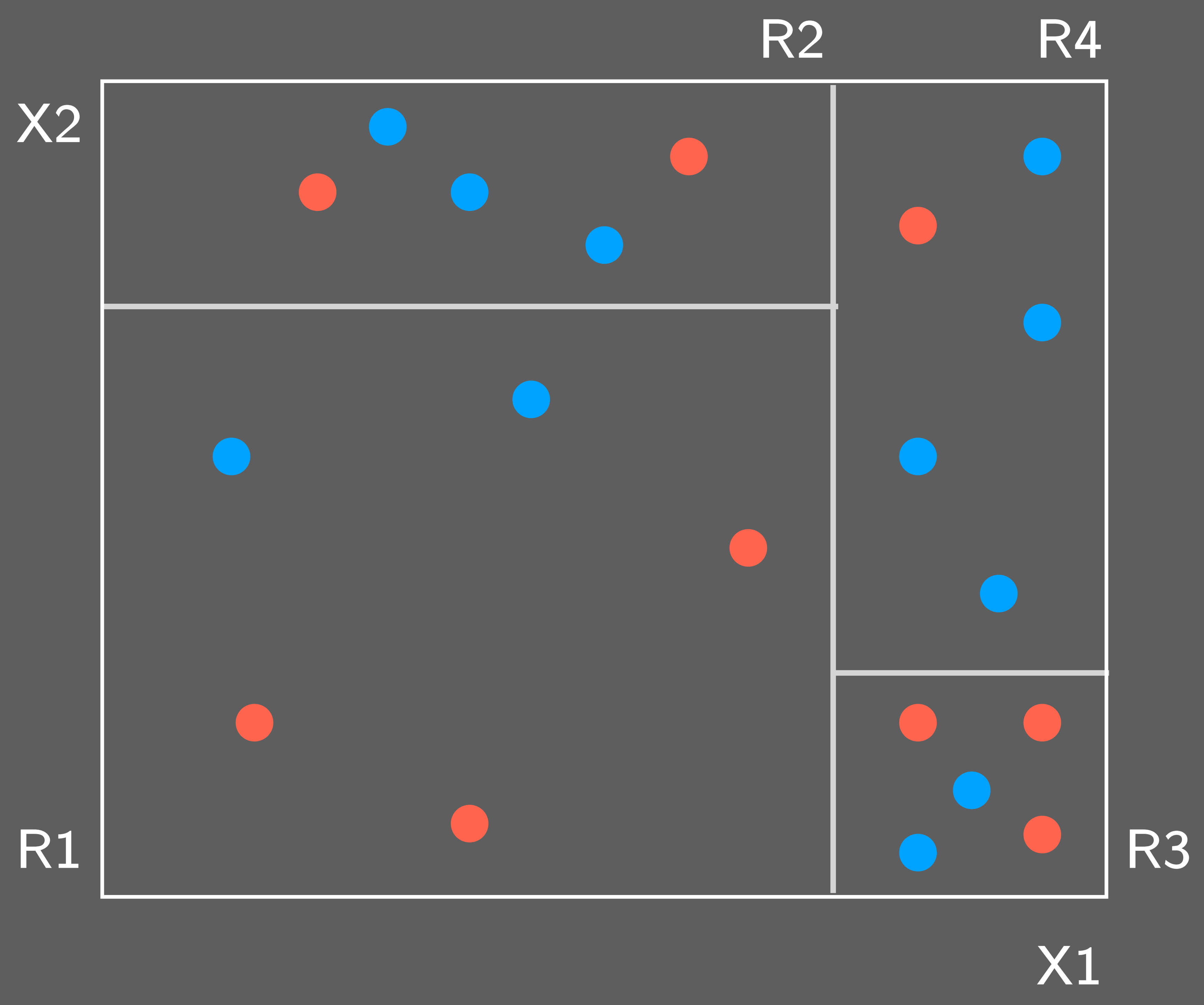




x_1: Body fat percentage
x_2: Average steps per week

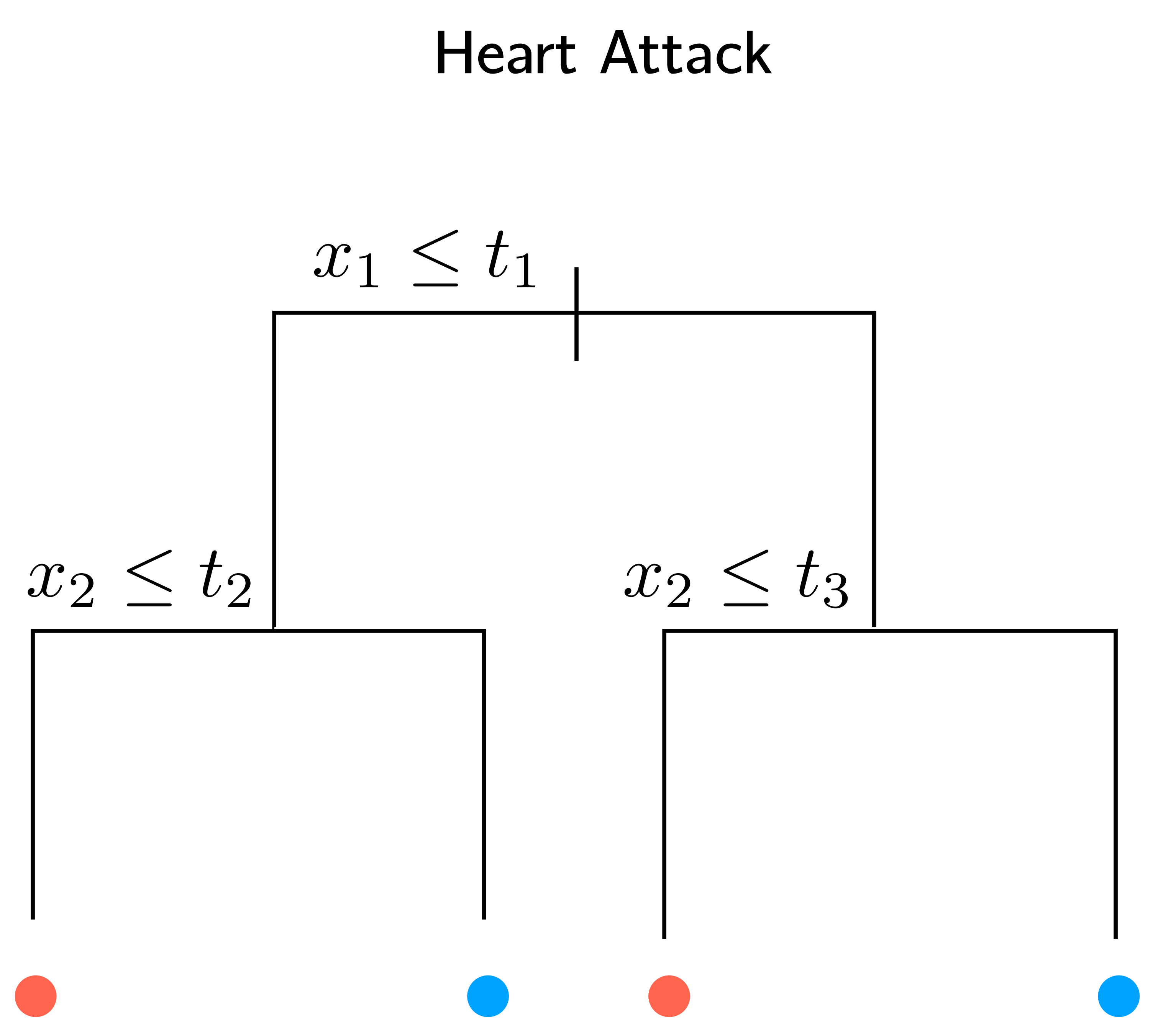
Question: What is your best prediction for each region? What is your criteria?

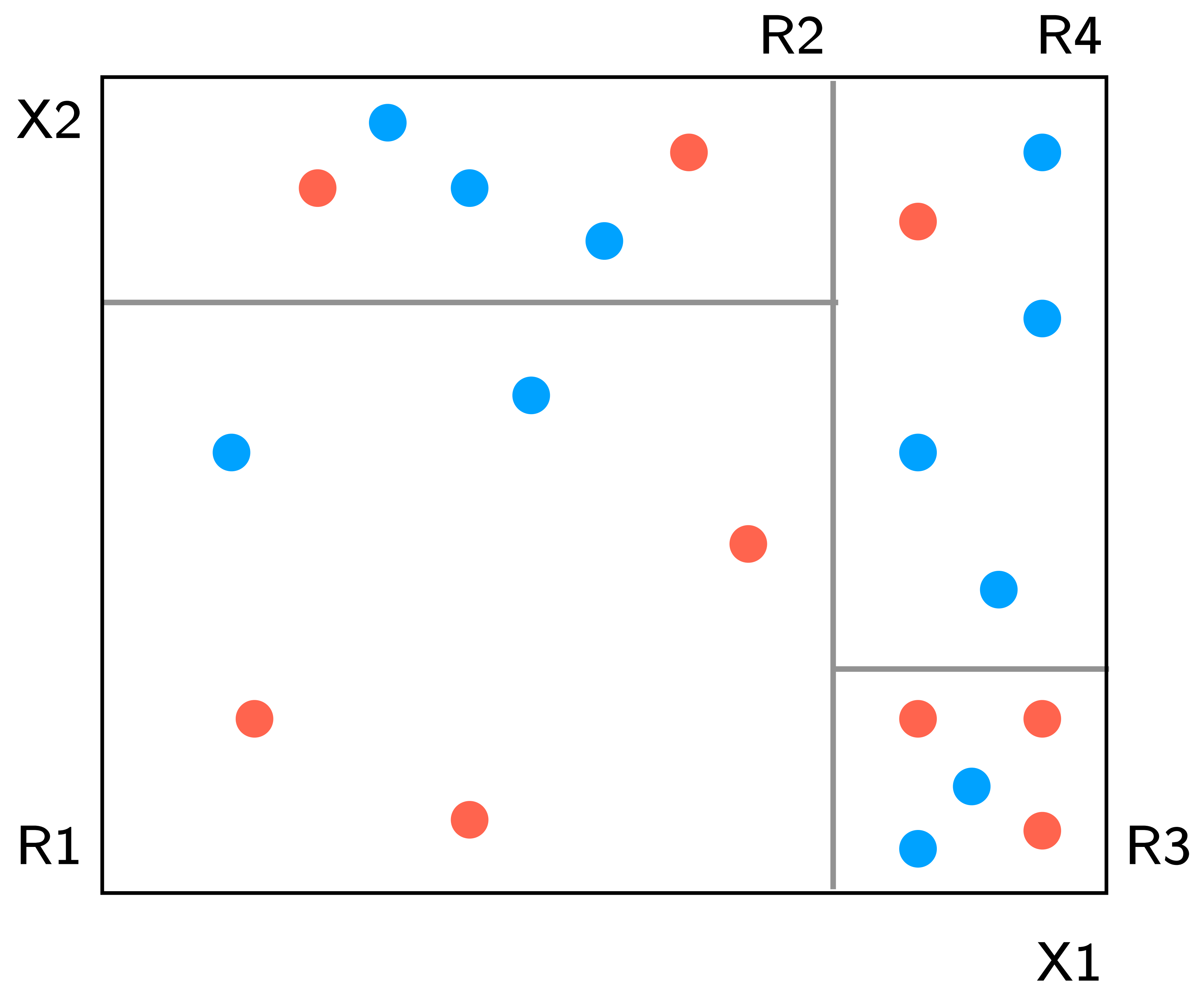




x_1: Body fat percentage
x_2: Average steps per week

Classification Trees

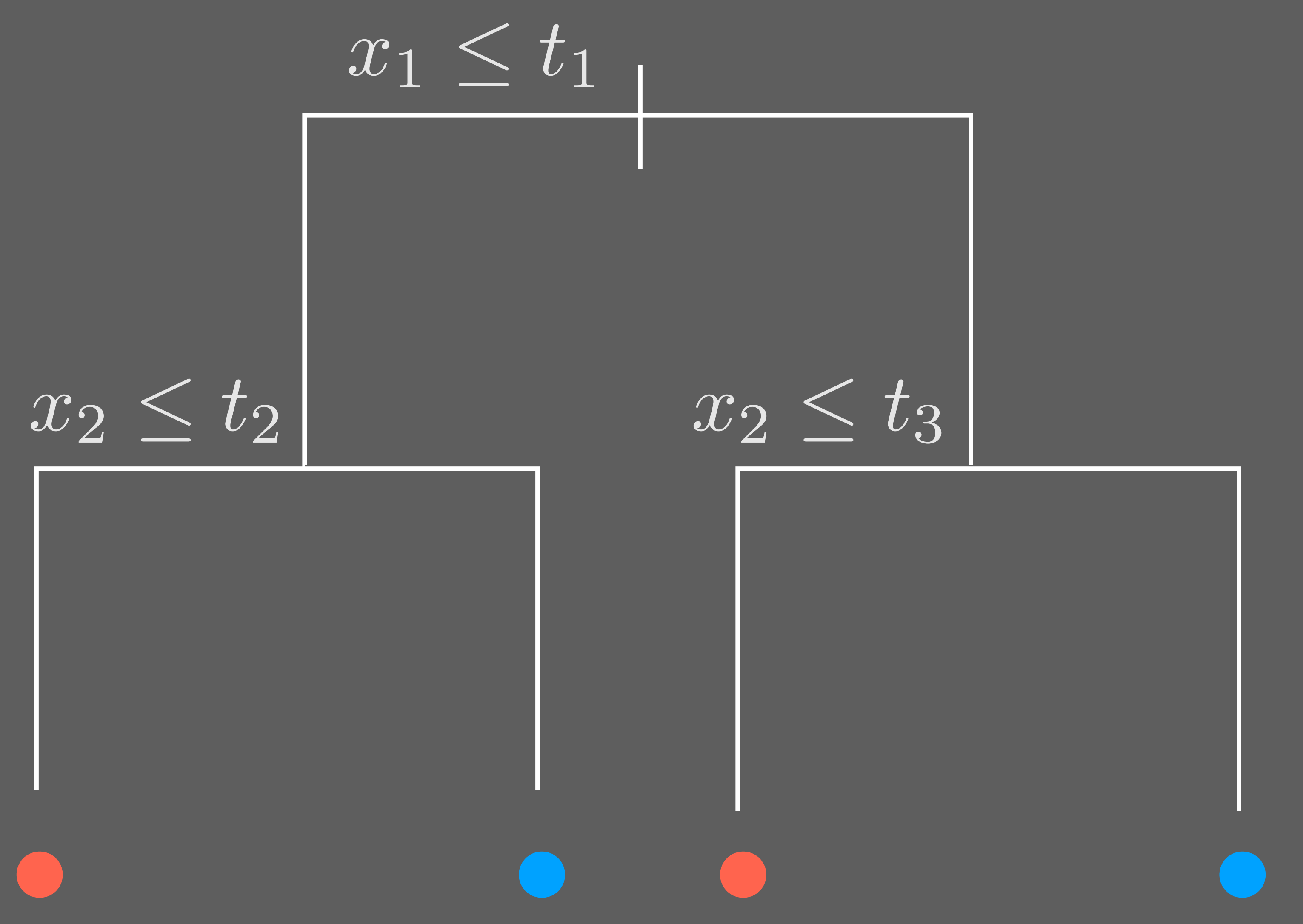


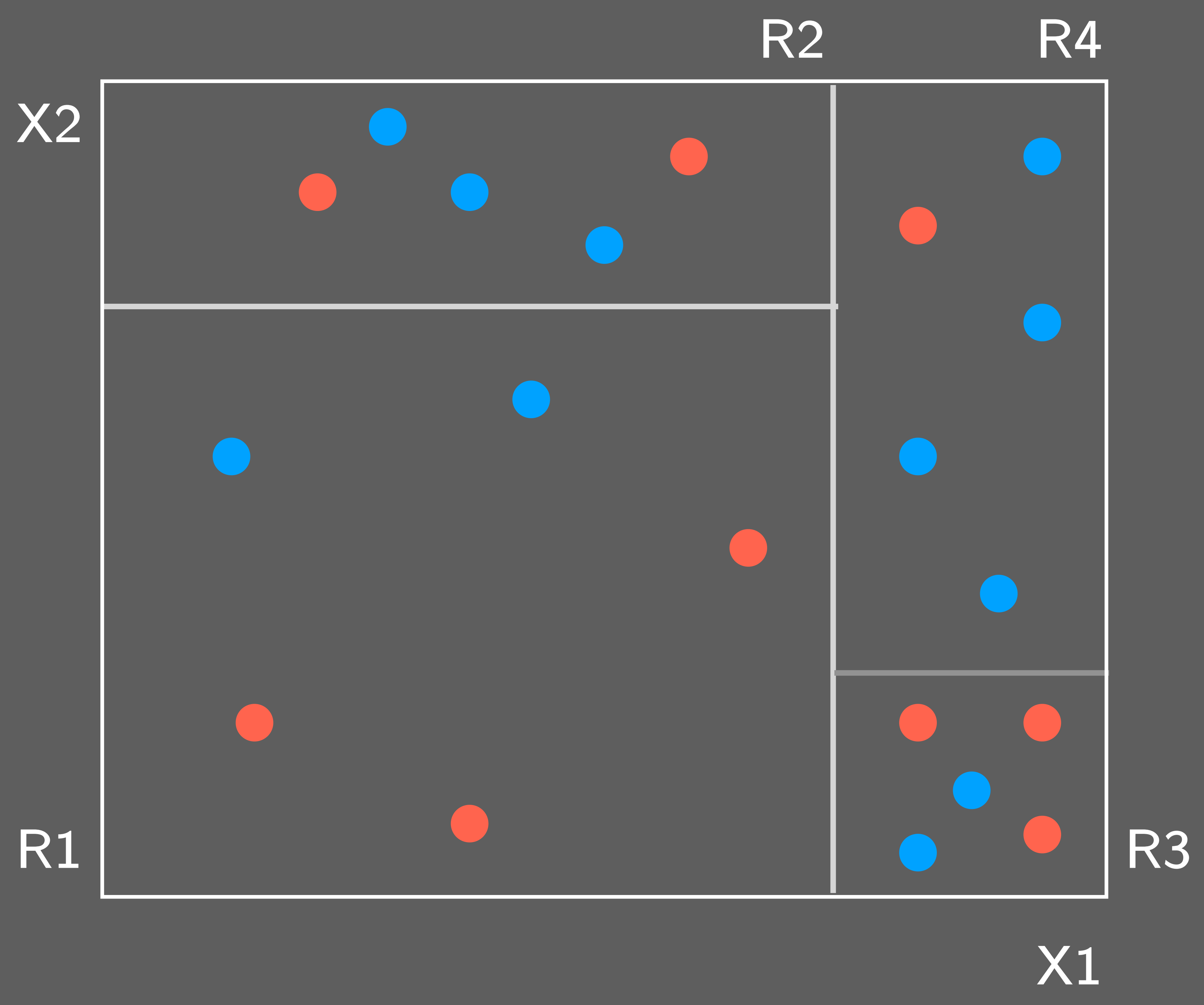


x_1: Body fat percentage
x_2: Average steps per week

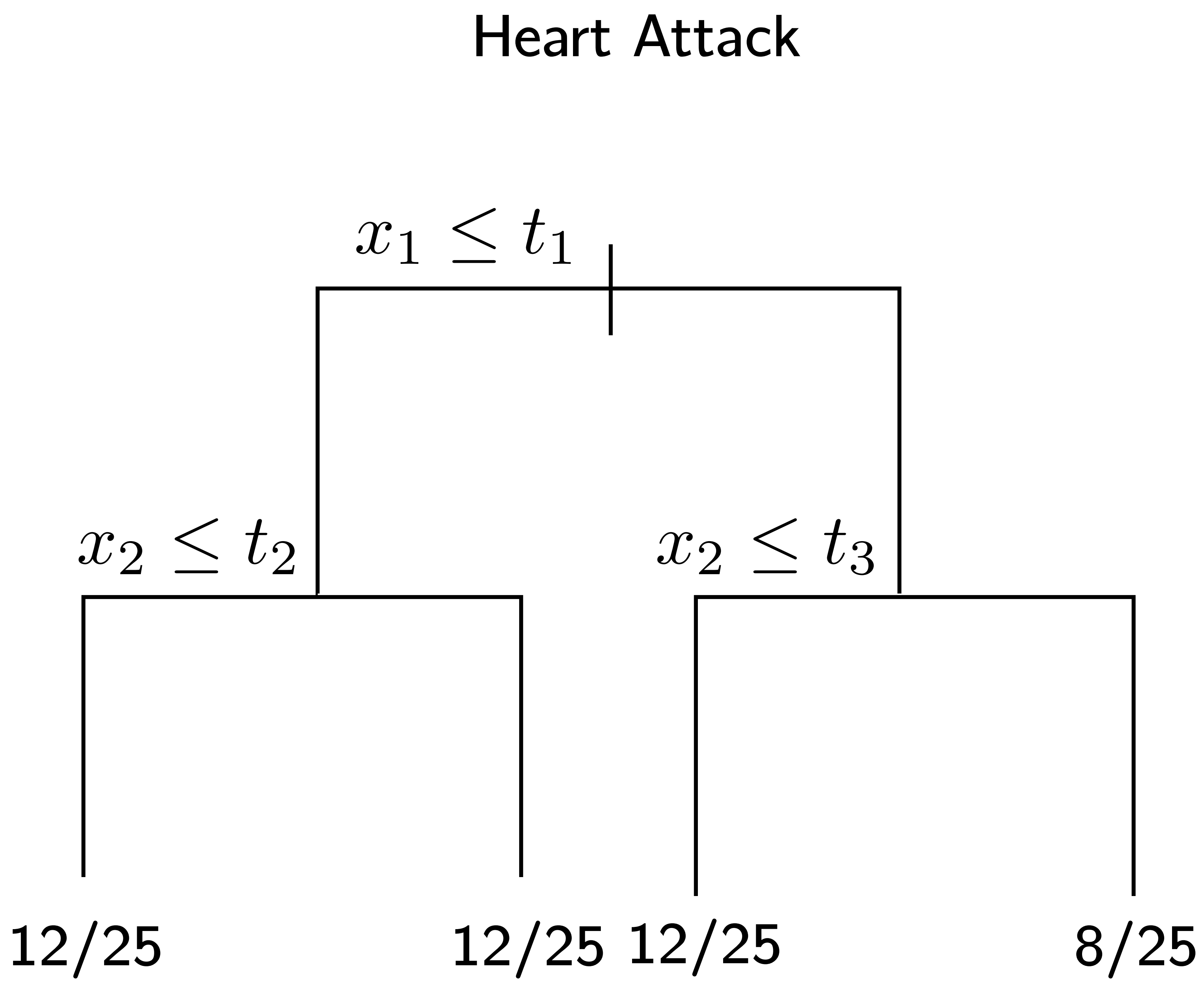
Question: Compute the impurity for each leaf as 2*p_i*(1-p_i), and compute the total impurity as the sum of the impurities for each leaf weighted by its number of elements.

Heart Attack





x_1: Body fat percentage
x_2: Average steps per week



IMPURITY = 8.8

x_1: Body fat percentage
x_2: Average steps per week

Tree Impurity

$$n_1 \times \mathrm{IMP}_1 + n_2 \times \mathrm{IMP}_2$$

$$\mathrm{IMP}_1 = 2p_i(1 - p_i), \quad i : x_i \in R_1$$

$$\mathrm{IMP}_2 = 2p_i(1 - p_i), \quad i : x_i \in R_2$$

Note that:

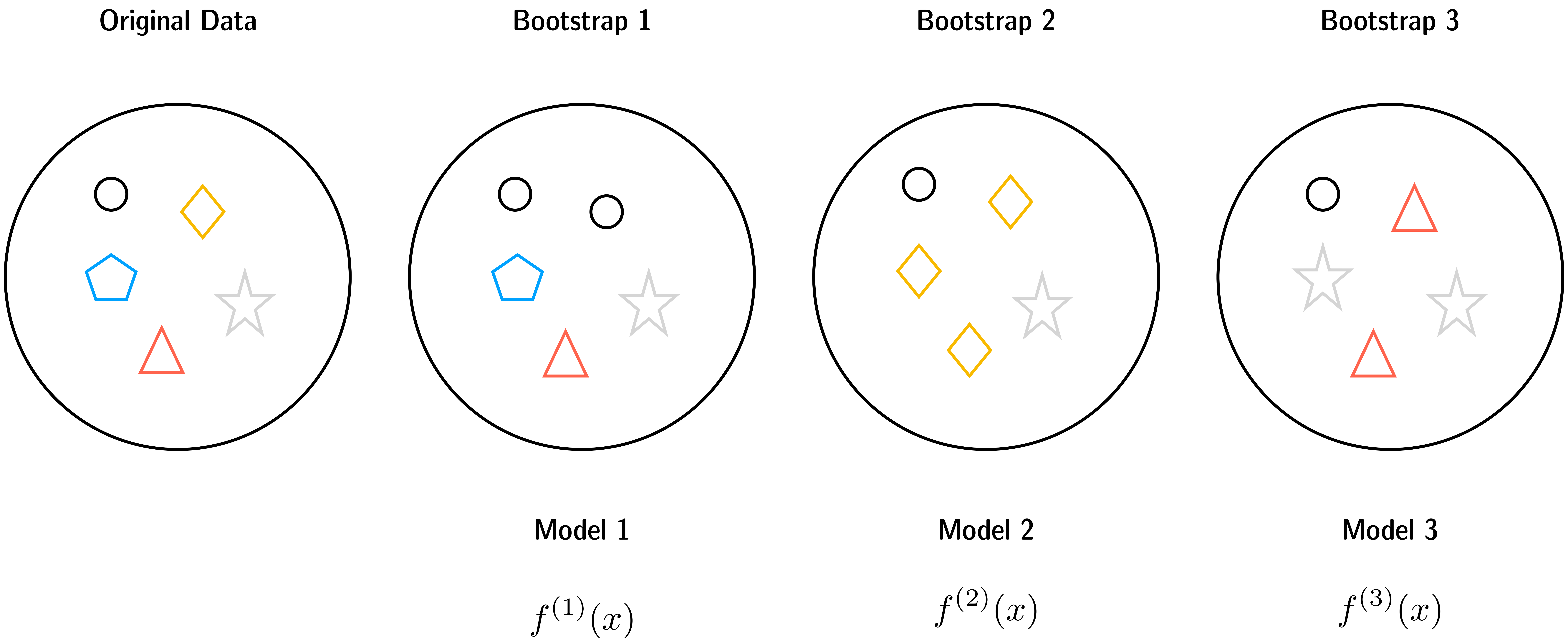
- We only discussed classification trees, but there are regression trees, and they are just as easy to apply. Of course the performance metrics are different.
- Trees with many regions are said to be “highly complex”. It is common to use pruning algorithms, which minimize certain cost function and penalize for tree complexity.
- Trees are easy to interpret/explain, but not very precise. In many cases a logit model can outperform a tree model.
- Trees are sometimes called “weak learners”, and are commonly utilized in ensemble models.

Ensemble Models

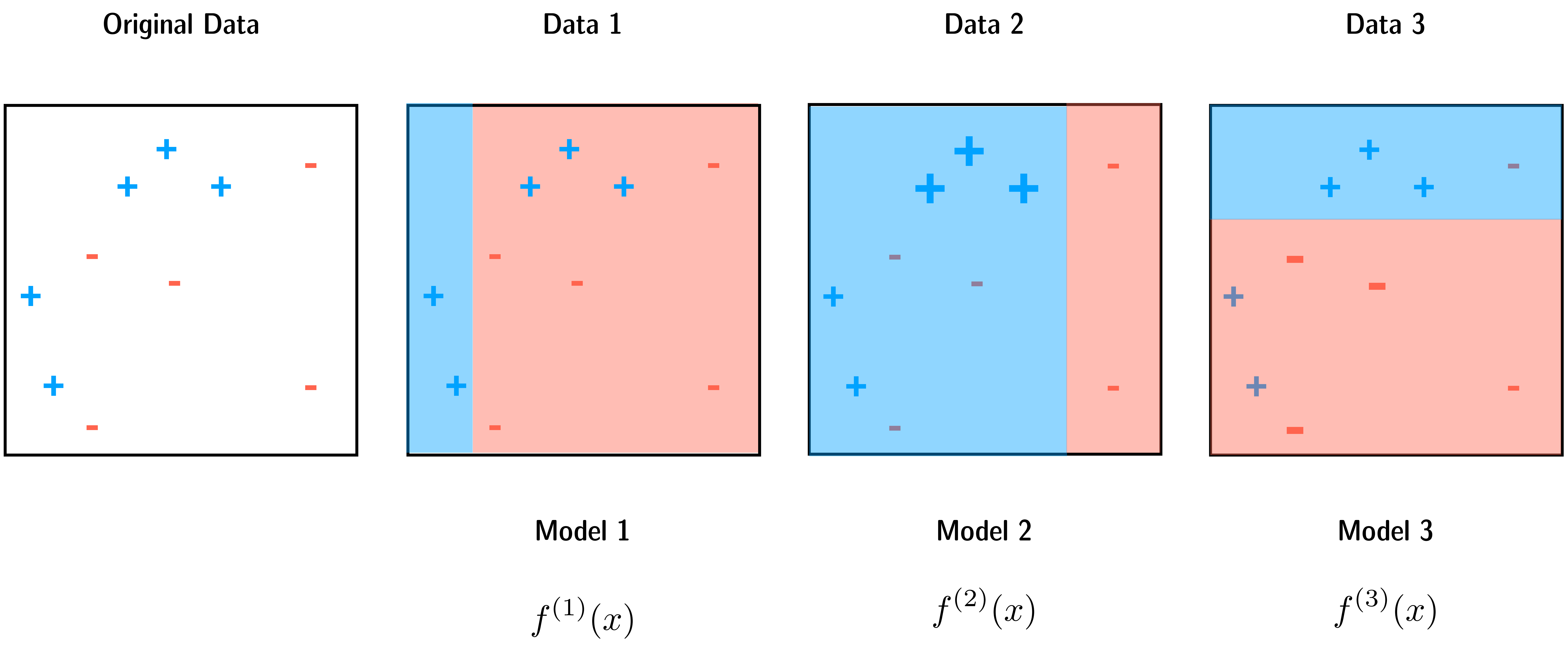
Weak "+" Weak ... "+" ... Weak = Strong

$f^{(1)}(x)$ $f^{(2)}(x)$ $f^{(K)}(x)$ $f^*(x)$

Ensemble Models: Bagging



Ensemble Models: Boosting



Classification of Many Models

Voting of “weak models”

Maastricht University