Basic Concepts.
Naïve Bayes.
Performance Metrics.

Radu Ionescu, Prof. PhD.

raducu.ionescu@gmail.com

Faculty of Mathematics and Computer Science
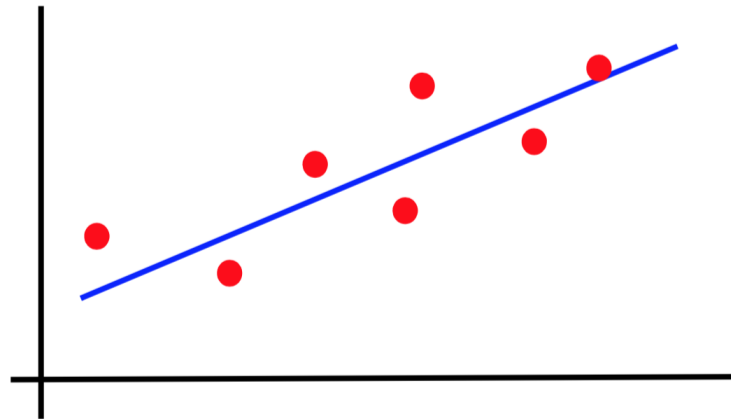
University of Bucharest

Learning paradigms

- Standard learning paradigms:

  ➤ Supervised learning

  ➤ Unsupervised learning

  ➤ Semi-supervised learning

  ➤ Reinforcement learning

- Non-standard paradigms:

  ➤ Active learning

  ➤ Transfer learning

  ➤ Transductive learning

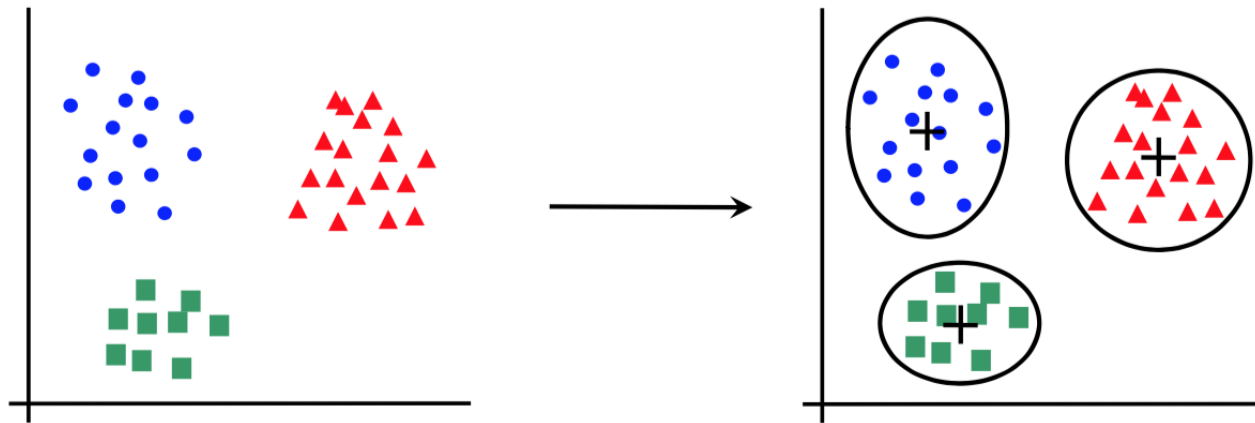# Canonical forms of supervised learning problems
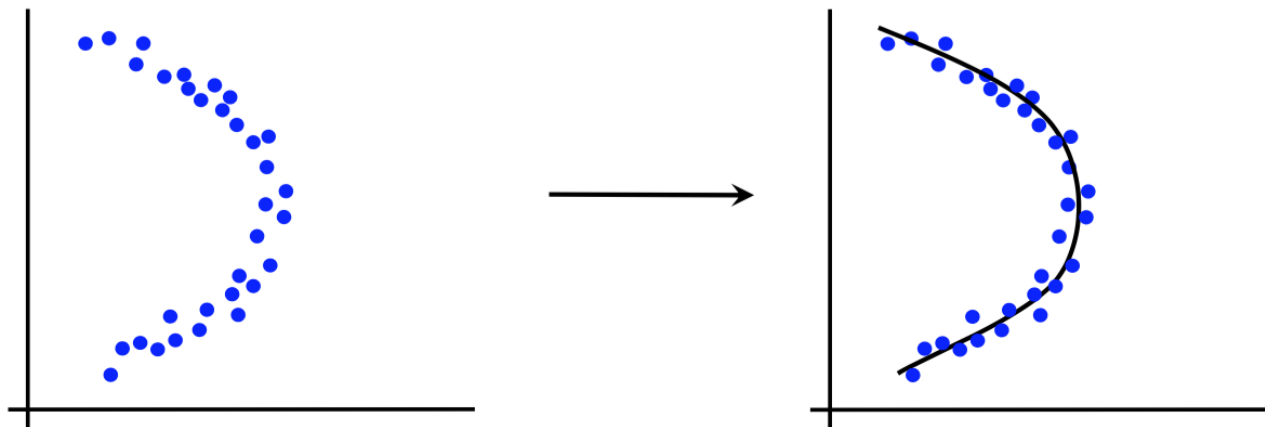
- Classification
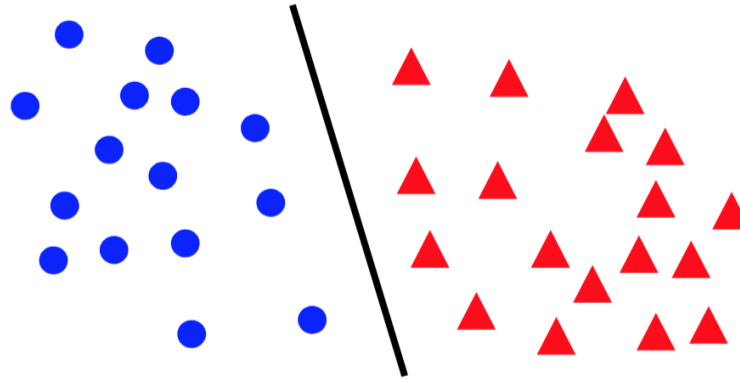
- Regression
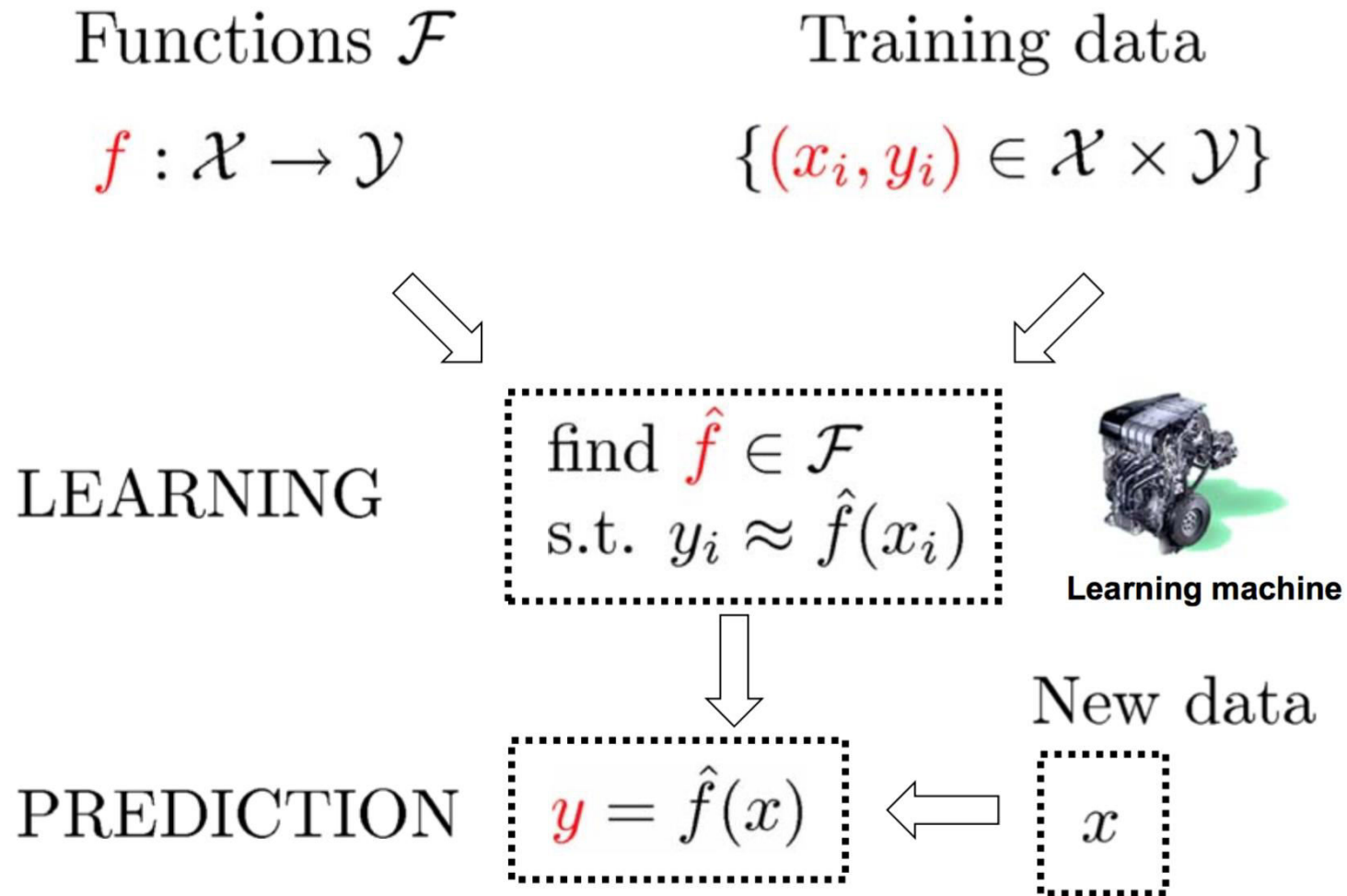
Canonical forms of unsupervised learning problems

- Clustering

- Dimensionality Reduction

The supervised learning paradigm

Functions $\mathcal{F}$

$$f : \mathcal{X} \to \mathcal{Y}$$

Training data

$$\{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}$$

LEARNING

find $\hat{f} \in \mathcal{F}$
s.t. $y_i \approx \hat{f}(x_i)$

Learning machine

PREDICTION $\quad y = \hat{f}(x)$

New data

x

Basic Steps of Supervised Learning

- **Set up** a supervised learning problem
- **Data Collection**

Start with training data for which we know the correct outcome (provided by a teacher or oracle)

- **Representation**

Choose how to represent the data

- **Modeling**

Choose a hypothesis class: $H = \{g: X \to Y\}$

- **Learning / Estimation**

Find best hypothesis you can in the chosen class

- **Model Selection**

Try different models and pick the best one

- If results are good, then stop

Else refine one or more of the above

Classification into Banana or Furbish

- **Training data**

➢ Banana language:
baboi, bananonina, bello, hana, stupa

➢ Furbish:
doo, dah, toh, yoo, dah-boo, ee-tay

- **Test data**

➢ gelato

- What is the language?
- Why?
- Learning is hard without establishing the hypothesis class!

Training versus Testing

- What do we want?
- ➤ Good performance (low loss) on training data?
- ➤ No, good performance on unseen test data!

- Training Data:
- ➤ $\{(x_1,y_1), (x_2,y_2), \ldots, (x_N,y_N)\}$
- ➤ Given to us for learning the mapping function f

- Test Data:
- ➤ $\{x_1, x_2, \ldots, x_M\}$
- ➤ Used to see if we have learnt anything 🙈

Loss / Error Functions

- How do we measure performance?

- Regression:
➤ Mean Squared Error (MSE)
➤ Mean Absolute Error (MAE)

- Classification:
➤ Misclassification rate
➤ Weighted misclassification rate via a cost matrix

➤ Binary classification:
True Positive, False Positive, True Negative, False Negative

➤ Multi-class classification:
Confusion Matrix
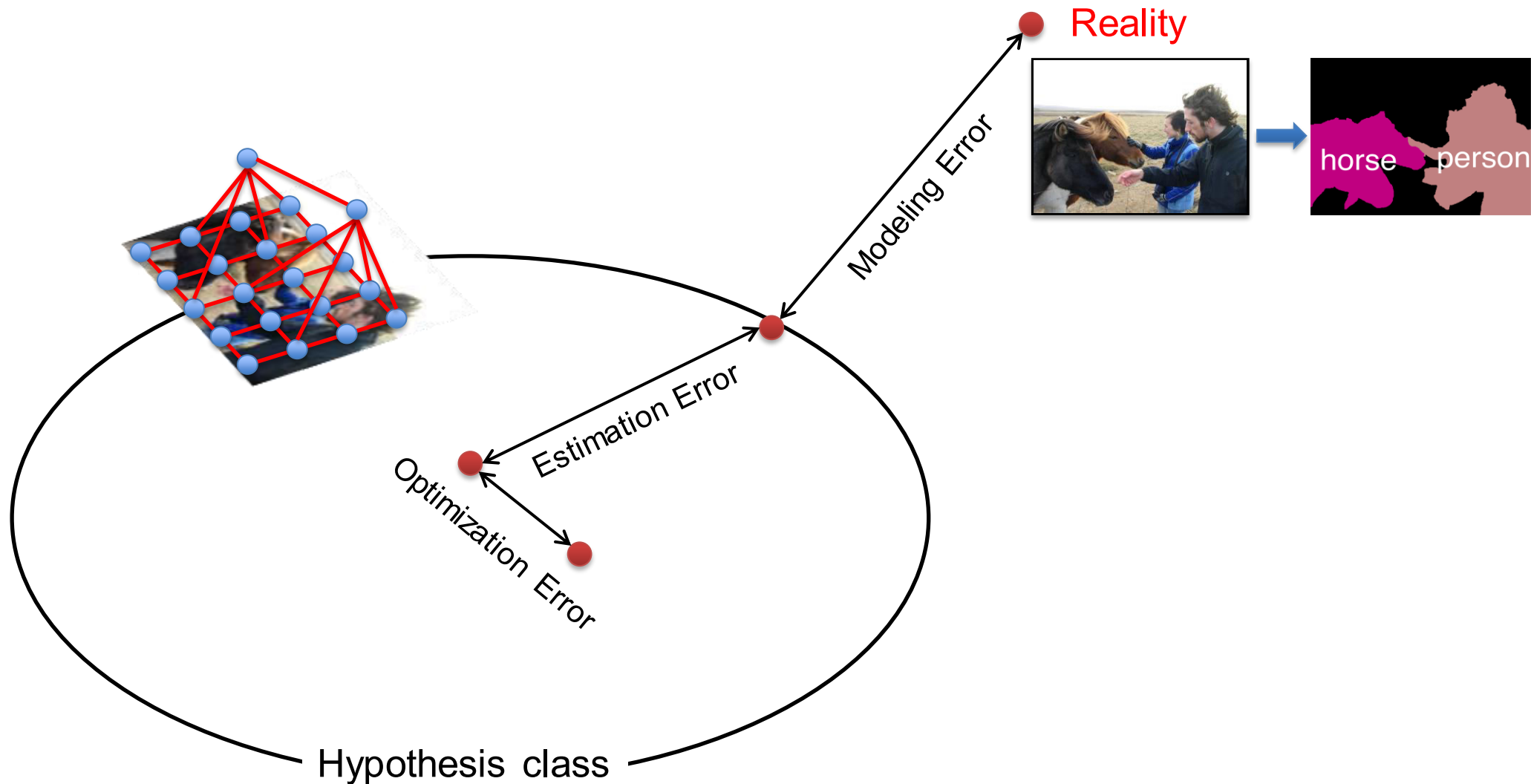
Errors

- Generalization error:

$$\mathcal{E}(h) = \int_{X \times Y} V(h(x), y)\rho(x, y)dxdy$$

- The joint probability $\rho(x, y)$ is usually unknown
- Hence, we compute the empirical error:

$$E(h) = \frac{1}{n}\sum_{i=1}^{n} V(h(x_i), y_i)$$

- Do we estimate the empirical error on the training set or on the test set?
- ➢ Reporting training error (instead of test) is CHEATING!

Error Decomposition

Reality

Modeling Error

Estimation Error

Optimization Error

Hypothesis class

horse   person

Error Decomposition

- Approximation/Modeling Error
  - ➤ We approximate the reality with a model (hypothesis class)

- Estimation Error
  - ➤ We try to learn a model with finite data

- Optimization Error
  - ➤ We could/did not optimize to completion

- Bayes Error (more on this later)
  - ➤ Reality just sucks!

Bias-Variance Trade-off

- Bias
- ➢ Systematic error that comes from the inability of the model to learn the true relationship between features and labels (underfitting)
- ➢ Can be corrected by increasing the model's complexity

- Variance
- ➢ Random error that comes from sensitivity to small fluctuations in the data, because the algorithm modeled noise in the training data (overfitting)
- ➢ Can be corrected by adding more training samples or by decreasing the model's complexity

Bias-Variance Trade-off

Bias-Variance Trade-off
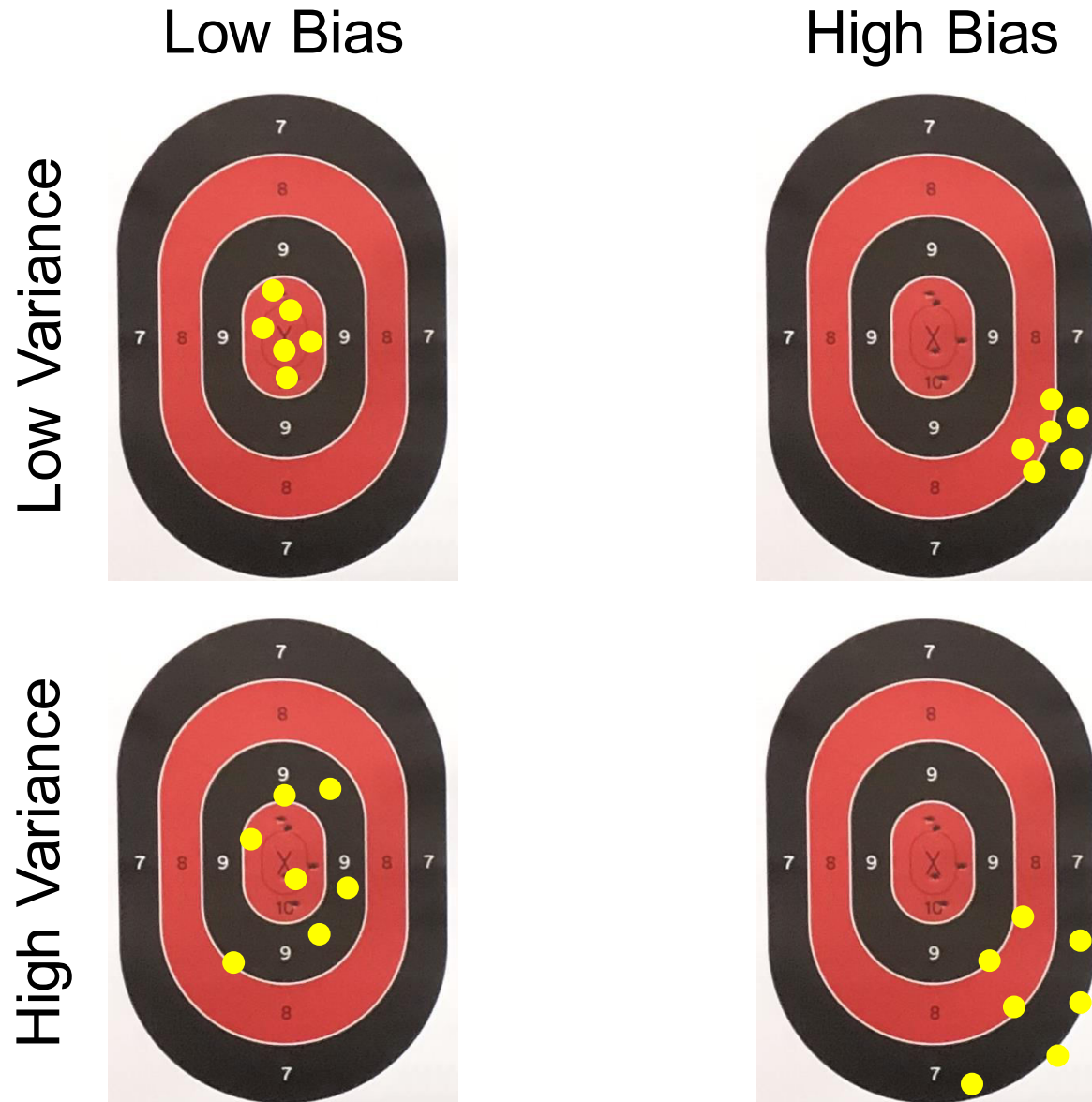
Underfitting versus overfitting

- Perhaps the most important problem of learning?

- All about improving generalization capacity
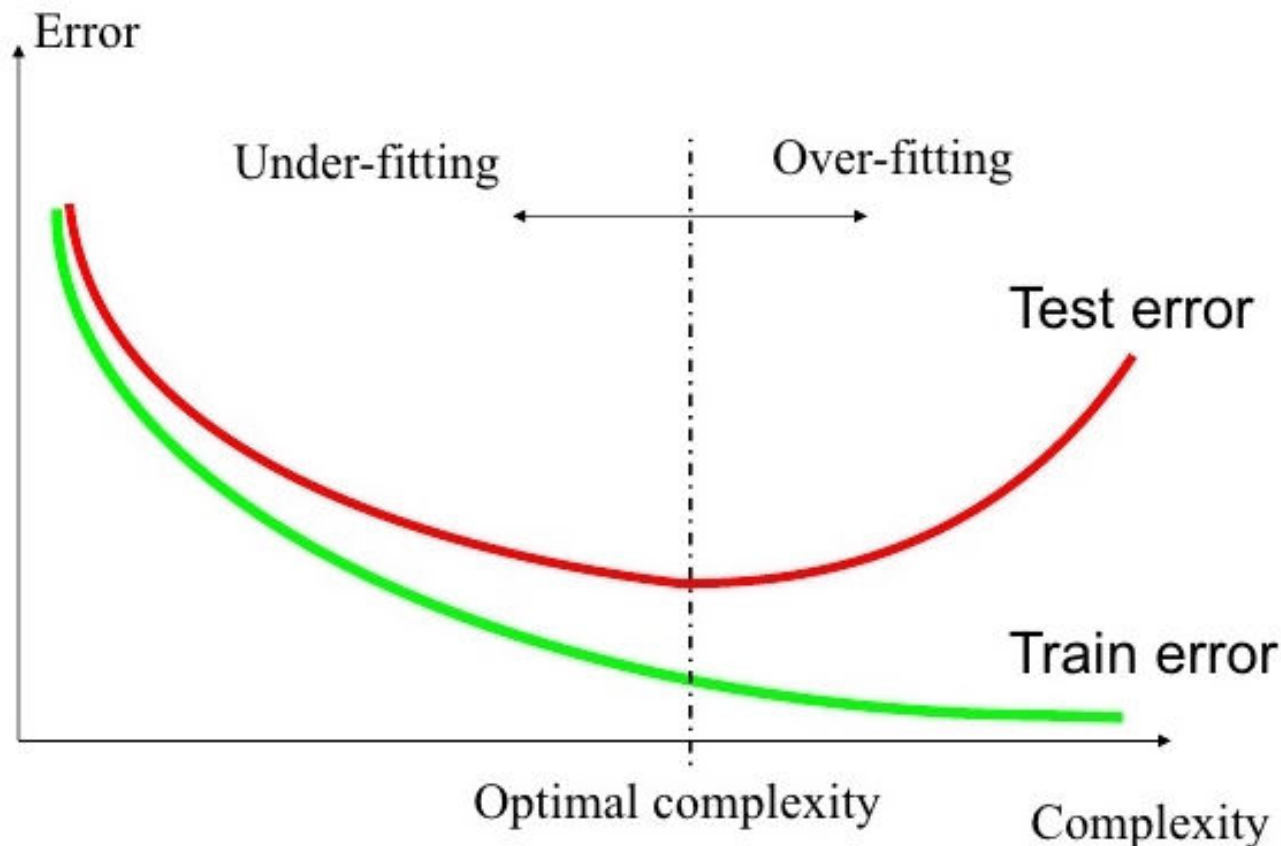
Underfitting versus overfitting

- Example 1: regression task
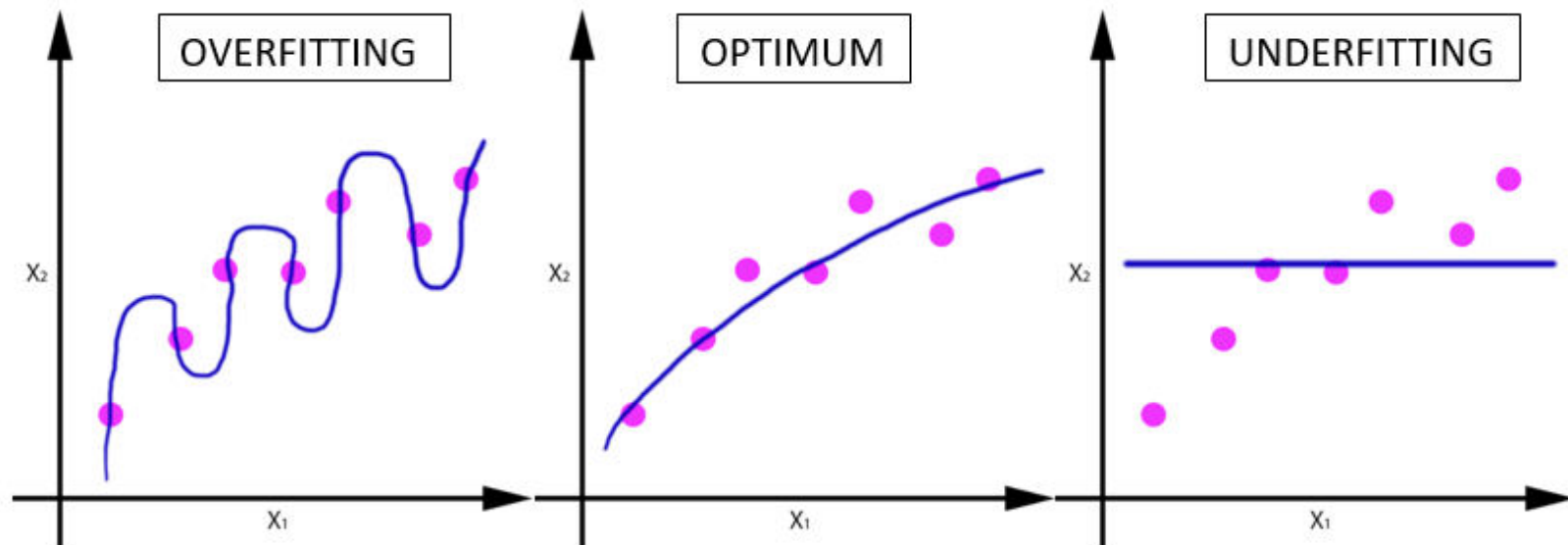
Underfitting versus overfitting

- Example 2: classification task
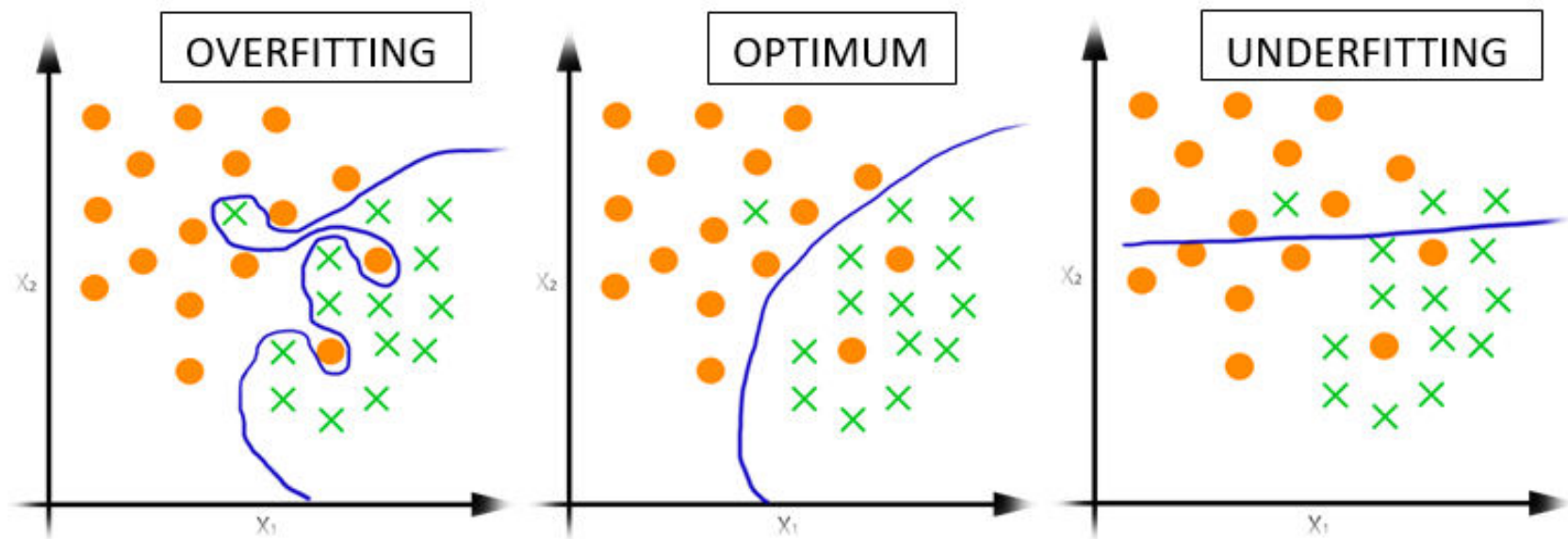
Procedural View

- **Training Stage:**
  - ➢ Raw data → x
    (feature extraction)
  - ➢ Training data {(x, y)} → f
    (learning)

- **Testing Stage:**
  - ➢ Raw data → x
    (feature extraction)
  - ➢ Test data x → f(x)
    (apply function, evaluate error)

Statistical Estimation View

- Probabilities to the rescue:
  - ➢ x and y are random variables
  - ➢ $D = (x_1,y_1), (x_2,y_2), \ldots, (x_N,y_N) \sim P(X,Y)$

- We suppose that data is IID (Independent Identically Distributed):
  - ➢ Both training and test data sampled IID from $P(X,Y)$
  - ➢ Learn on training set
  - ➢ Have some hope of generalizing to test set

Important Concepts

- ## Model capacity
  - ➤ Measure how large hypothesis class H is?
  - ➤ Are all functions allowed?

- ## Overfitting
  - ➤ f works well on training data
  - ➤ Works poorly on test data

- ## Generalization capacity
  - ➤ The ability to achieve low error on new test data

Guarantees

- 20 years of research in Learning Theory oversimplified…

- If we have:
  - ➢ enough training data D
  - ➢ and H is not too complex
- then probably we can generalize to unseen test data

Probabilities (recap)

- A is non-deterministic event:
  - Can think of A as a boolean-valued variable
  - Example: A = "Simona Halep will win Roland Garros"

- What does P(A) mean?

- Statistical View:

$$\lim_{N \to \infty} \frac{\#(A = true)}{N}$$

  - Limiting frequency of a repeating non-deterministic event

- Bayesian View:
  - P(A) is your "belief" about A

- Market Design View:
  - P(A) tells you how much you would bet when choosing A

Axioms of Probability (recap)

- $0 \leq P(A) \leq 1$
- $P(\emptyset) = 0$
- $P(\forall) = 1$
- $P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$

A or B

A and B

A

B
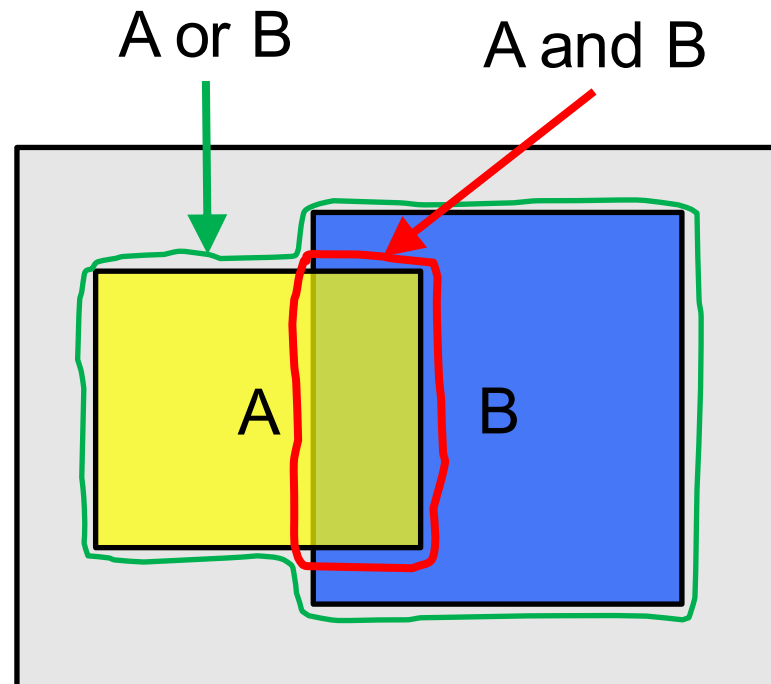
Conditional Probabilities (recap)

$$P(Y = y \mid X = x)$$

- What do you believe about $Y = y$, if I tell you $X = x$?

- P(Simona Halep will win Roland Garros 2020)?

- What if I tell you:
  - In 2018, Simona Halep won Roland Garros
  - Simona Halep lost two Roland Garros finals
  - Simona is on 6th place in the WTA ranking
  - In 2019, Ashleigh Barty (1st place in the WTA ranking) won Roland Garros

Conditional Probabilities (recap)

- P(A | B) = In worlds where B is true, fraction where A is true

- Example:
➢ H: "Have a headache"
➢ F: "Have the flu"

- $P(H) = \frac{1}{10}$

- $P(F) = \frac{1}{40}$

- $P(H \mid F) = \frac{1}{2}$

- Headaches are rare and flu is even more rare, but if you have the flu, there is a 50-50 chance you will have a headache
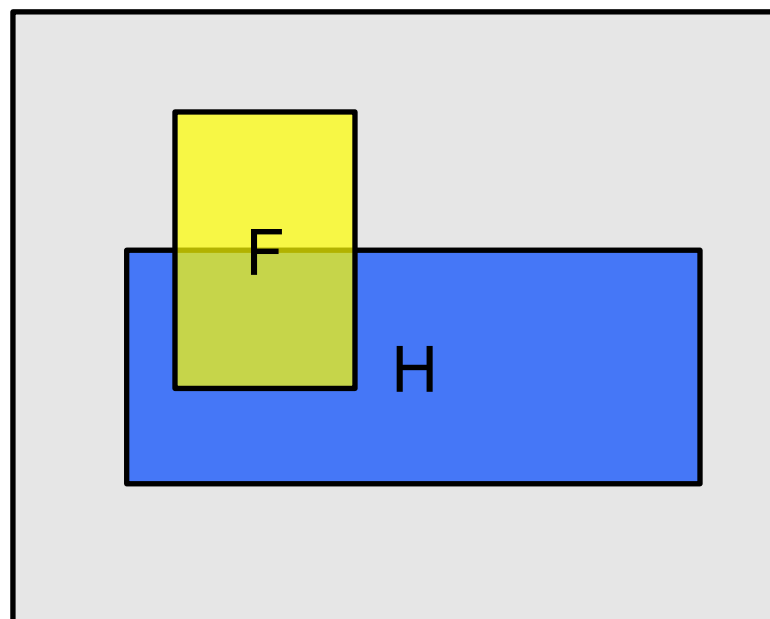
Bayes Rule

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} = \frac{P(A \mid B)\, P(B)}{P(A)}$$

- Thomas Bayes "An Essay towards solving a Problem in the Doctrine of Chances" Royal Society, 1763.
- Easy to grasp, if you think of areas:
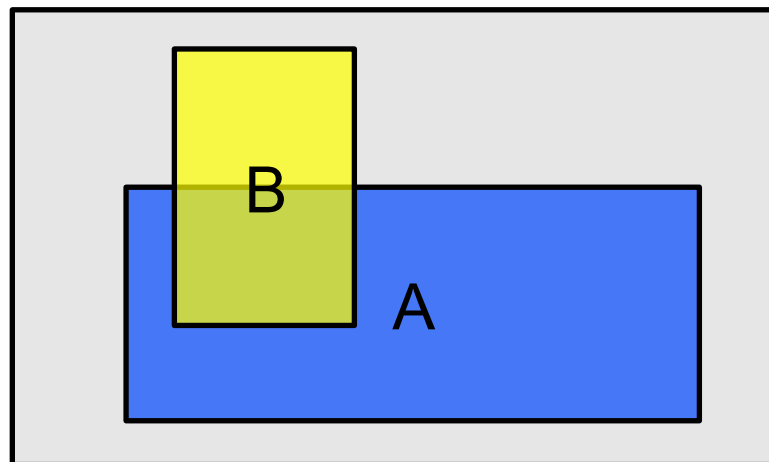
Bayes Rule

Concepts:

- Likelihood
  ➢ How much does a certain hypothesis explain the data?

- Prior
  ➢ What do you believe before seeing any data?

- Posterior
  ➢ What do we believe after seeing the data?

Monty Hall Problem
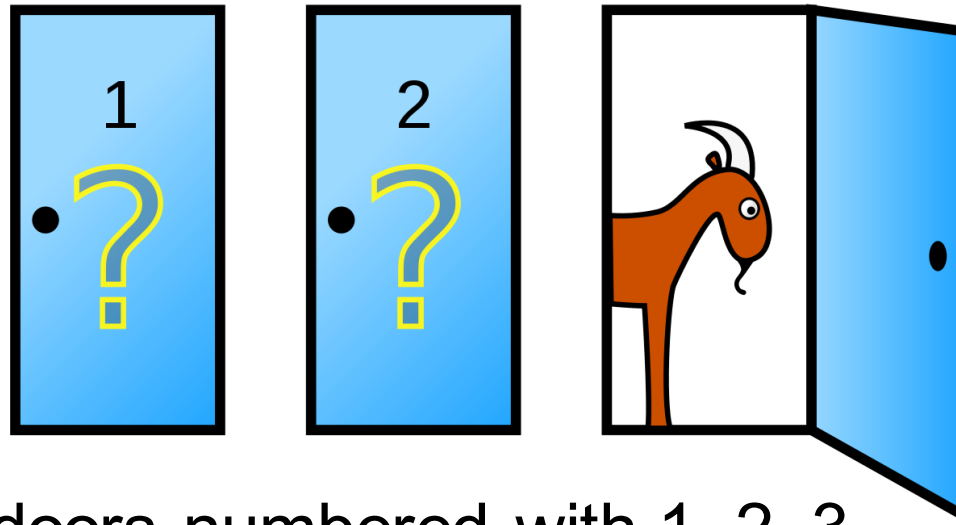
- There are 3 doors numbered with 1, 2, 3.
- A large prize (a car) is hidden behind one of the doors. The other doors each have a goat behind.
- We have to pick a door.
- Suppose that we opt for door 1. Then, the host opens door 3, revealing the goat behind. We are offered the option to change our pick. What should we do next?

(a) We keep our initial choice (door 1);
(b) We switch to door 2;
(c) There is no difference.

Monty Hall Problem

- $H = i$ is the hypothesis "the prize is behind door i". A priori, all 3 doors are equally likely to hide the prize:

$$P(H = 1) = P(H = 2) = P(H = 3) = \frac{1}{3}$$

- We chose door 1.
- If the prize is behind door 1, the host is indifferent and can choose between doors 2 or 3 with equal probability:

$$P(U = 2 \mid H = 1) = \frac{1}{2}, P(U = 3 \mid H = 1) = \frac{1}{2}$$

- If the prize is behind door 2 (or 3, respectively), the host must choose door 3 (or 2, respectively):

$$P(U = 2 \mid H = 2) = 0, P(U = 3 \mid H = 2) = 1$$

$$P(U = 2 \mid H = 3) = 1, P(U = 3 \mid H = 3) = 0$$

- The host opens door 3 (U=3), revealing the goat. The observation is U=3. Is the prize behind door 1 or 2?

Monty Hall Problem

$$P(H = 1) = P(H = 2) = P(H = 3) = \frac{1}{3}$$

$$P(U = 2 \mid H = 1) = \frac{1}{2}, \, P(U = 3 \mid H = 1) = \frac{1}{2}$$

$$P(U = 2 \mid H = 2) = 0, \, P(U = 3 \mid H = 2) = 1$$

$$P(U = 2 \mid H = 3) = 1, \, P(U = 3 \mid H = 3) = 0$$

- We apply the Bayes rule:

$$P(H = 1 \mid U = 3) = \frac{P(U = 3 \mid H = 1) \, P(H = 1)}{P(U = 3)} = \frac{\frac{1}{2} \cdot \frac{1}{3}}{\frac{1}{2}} = \frac{1}{3}$$

$$P(H = 2 \mid U = 3) = \frac{P(U = 3 \mid H = 2) \, P(H = 2)}{P(U = 3)} = \frac{1 \cdot \frac{1}{3}}{\frac{1}{2}} = \frac{2}{3}$$

Optimal classifier

- **Learn**: $h: X \to Y$

  ➢ **X** – features

  ➢ Y – target classes

- Suppose we know P(Y|**X**) exactly, how should we classify the data?

  ➢ We apply the Bayes classifier:

$$y^* = h^*(x) = \underset{y}{\operatorname{argmax}} P(Y = y \mid X = x)$$

- **Why?**

Optimal classifier

- **Theorem:** The Bayes classifier h_{Bayes} is optimal!

  - This is:
  $$error_{true}(h_{Bayes}) \leq error_{true}(h), \forall h$$

- The **Bayes error** is the smallest possible error:

$$error_{Bayes} = 1 - \sum_{y \neq y^*} \int_{x \in H_i} P(y \mid x) P(x) \, dx$$

Optimal classifier

- How hard is it to learn the optimal classifier?
➢ What about categorical data?

- How do we represent these? How many parameters?
➢ Class-Prior P(Y):

Suppose Y is composed of k classes

➢ Likelihood P($\mathbf{X}$ | Y):

Suppose $\mathbf{X}$ is composed of n binary features

Complex model → High variance with limited data!

Independence to the rescue

- Two variables are independent iif their joint factors:
$$P(x, y) = P(x)\, P(y)$$

P(x,y)

$=$ **P(y)** $\times$ **P(x)**
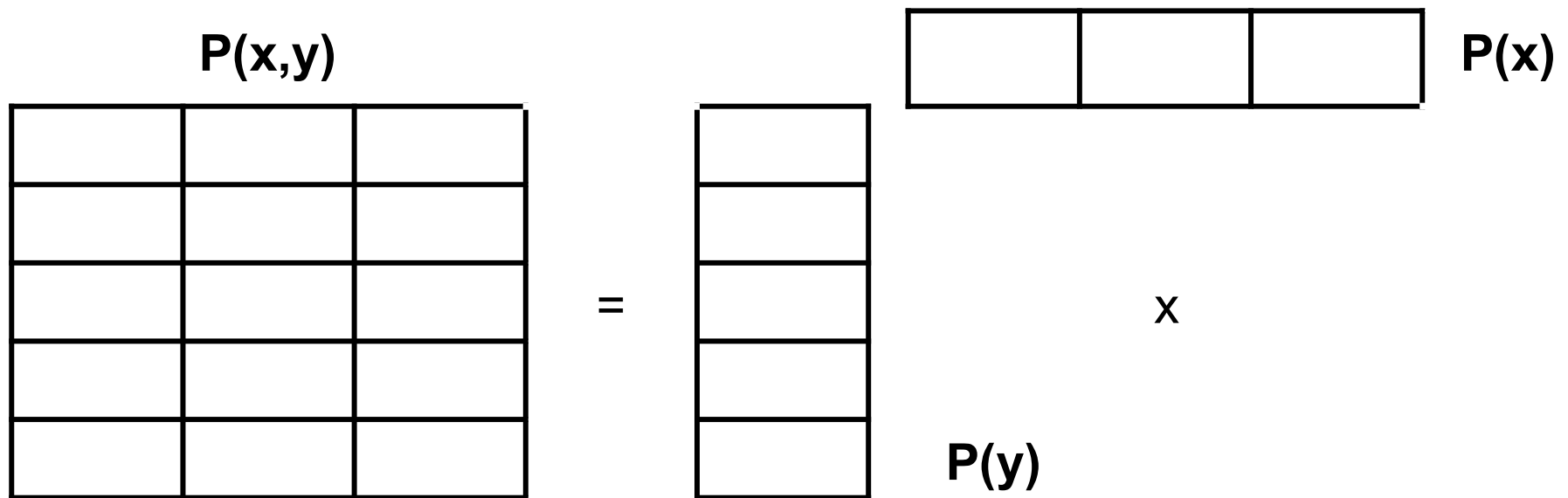
- Two variables are conditionally independent if, given a third variable, we have:
$$P(x, y \mid z) = P(x \mid z)\, P(y \mid z)$$

Naïve Bayes assumption

- Naïve Bayes assumption:
  - Features are independent given class:
    $$P(X_1, X_2 \mid Y) = P(X_1 \mid Y)P(X_2 \mid Y)$$

  - More generally:
    $$P(X_1 \dots X_n \mid Y) = \prod_i P(X_i \mid Y)$$

- How many parameters now?
  - Suppose **X** is composed of n binary features
  - Reduced from 2^n to $2 \cdot n$

Naïve Bayes classifier

- Given:
  - ➢ Class-Prior P(Y)
  - ➢ *n* conditionally independent features **X** given the class Y
  - ➢ For each X_i, we have likelihood $P(X_i \mid Y)$

- Naïve Bayes decision rule:

$$h_{NB}(x) = \underset{y}{\operatorname{argmax}} \, P(y) \, P(x_1, \dots, x_n \mid y)$$

$$h_{NB}(x) = \underset{y}{\operatorname{argmax}} \, P(y) \prod_i P(x_i \mid y)$$

- In practice we use sum of logs! Why?
- If assumption holds, NB is the optimal classifier!

Estimating parameters of NB

- We apply Maximum Likelihood Estimation (MLE)

➢ Given the training data set, we compute the number of samples for which A=a and B=b:

count(A=a, B=b)

- MLE for NB is simply:

➢ Estimation of Class-Prior: $P(Y = y) = $ ...

➢ Estimation of likelihood: $P(X_i = x_i \mid Y = y) = $ ...

In-class demo

- Variables:

Y = {does not play football, plays football}
X_1 = {does not watch sports on TV, watches sports on TV}
X_2 = {girl, boy}

- Estimate:

$P(Y = 0), P(Y = 1)$
$P(X_1 = 0 \mid Y = 0), P(X_1 = 1 \mid Y = 0)$
$P(X_2 = 0 \mid Y = 0), P(X_2 = 1 \mid Y = 0)$
$P(X_1 = 1 \mid Y = 1), P(X_1 = 1 \mid Y = 1)$
$P(X_2 = 1 \mid Y = 1), P(X_2 = 1 \mid Y = 1)$

- Prediction:

$$\operatorname*{argmax}_{y} P(Y = y) \, P(X_1 = x_1 \mid Y = y) P(X_2 = x_2 \mid Y = y)$$

Violating the NB assumption

- Usually, features are not conditionally independent:

$$P(X_1 \ldots X_n \mid Y) \neq \prod_i P(X_i \mid Y)$$

- Probabilities P(Y|**X**) often biased towards 0 or 1

- Nonetheless, NB is a very popular classifier
- ➢ Often performs well, even when assumption is violated

Underfitting versus overfitting

- All about improving generalization capacity
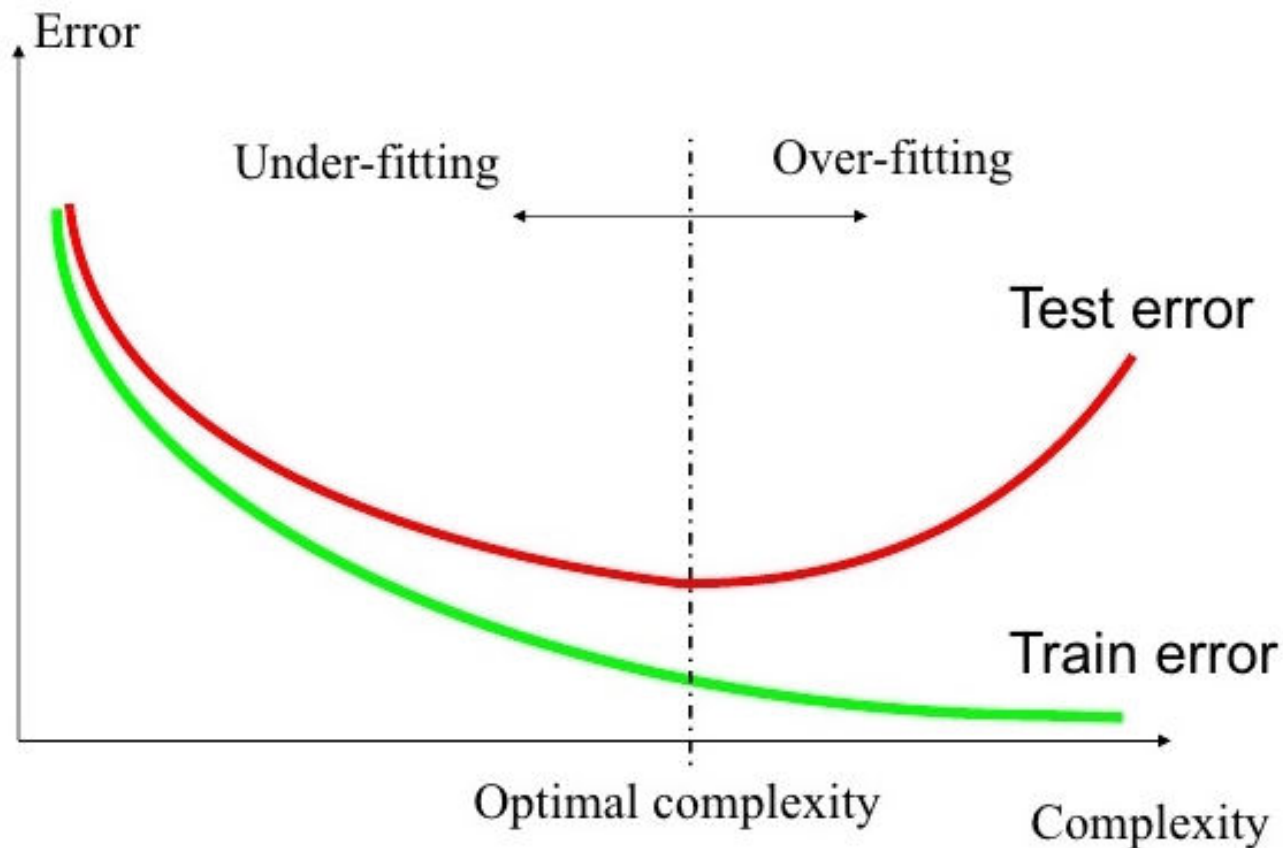
Splitting the data into training, validation and test

- In order to build a model with good generalization capacity, we have to test it on unseen data

- First approach (works well when we have enough data):

➢ 50% samples for training

➢ 25% samples for validation

➢ 25% samples for test

(percentages can vary)

Why not just split the data into training and test?

- Repeatedly using the same split when trying different hyperparameters can "wear out" the test set:

  ➤ We are overfitting in hyperparameter space!

- We obtain a better error estimate by tuning the hyperparameters on a (different) validation set

Training, validation, test
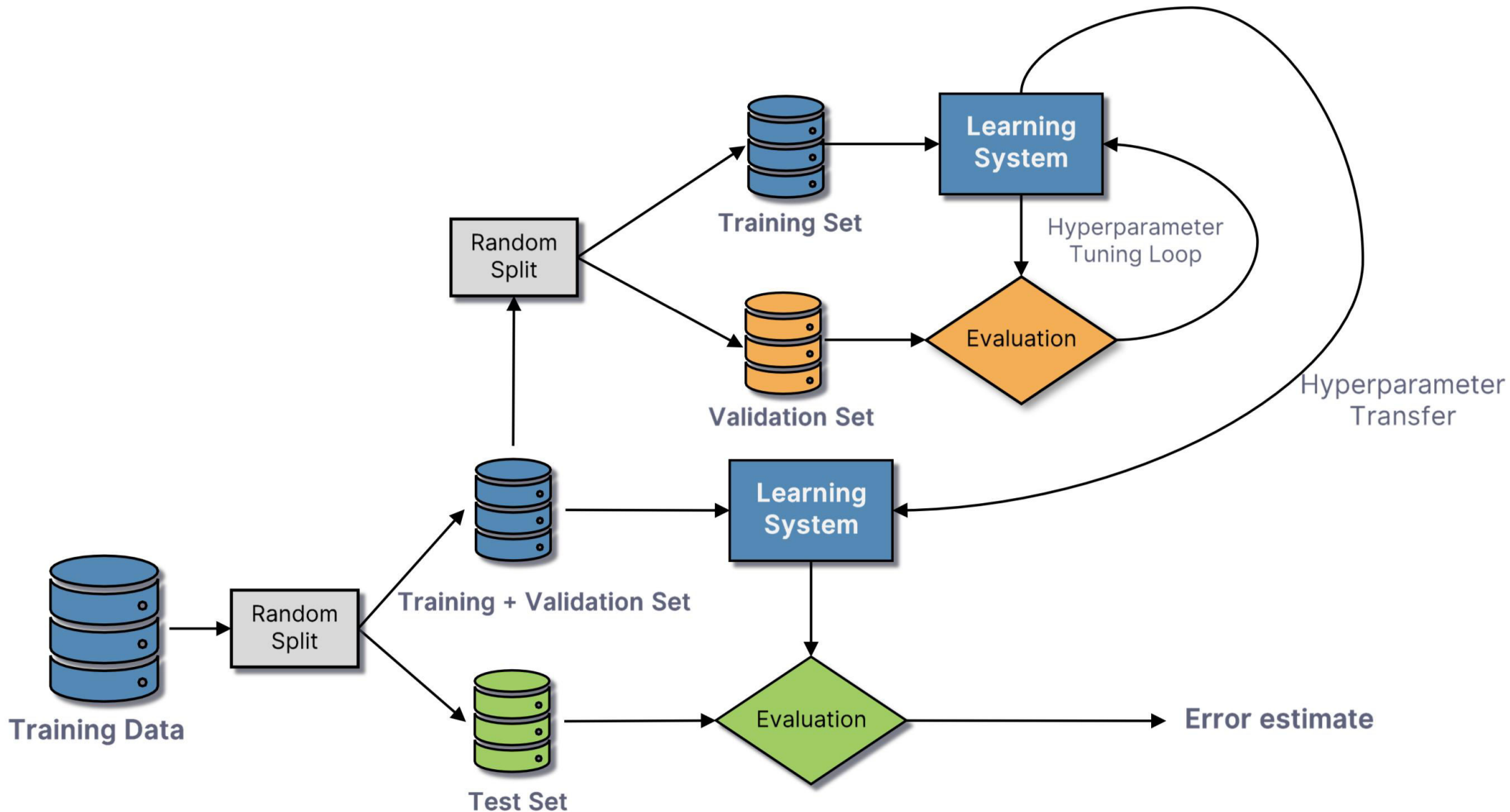
Cross-validation

- Another approach (works well when we have limited data):
- ➢ Split the data into k equal parts (folds)
- ➢ Train on k-1 folds and test on the left out fold
- ➢ Repeat for k times
- ➢ Average the results

- When the number of folds is equal to the number of samples:
- ➢ Leave-one-out cross-validation
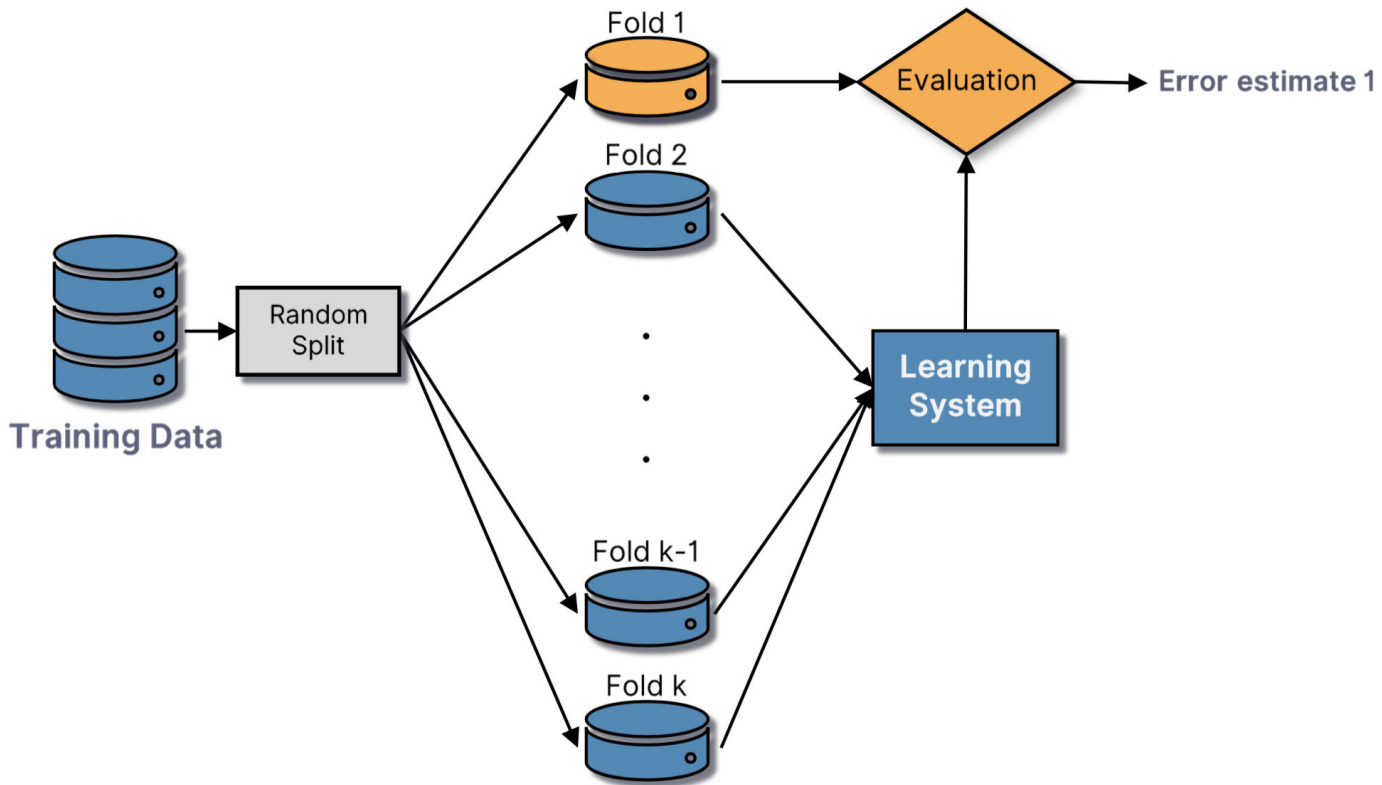
Cross-validation
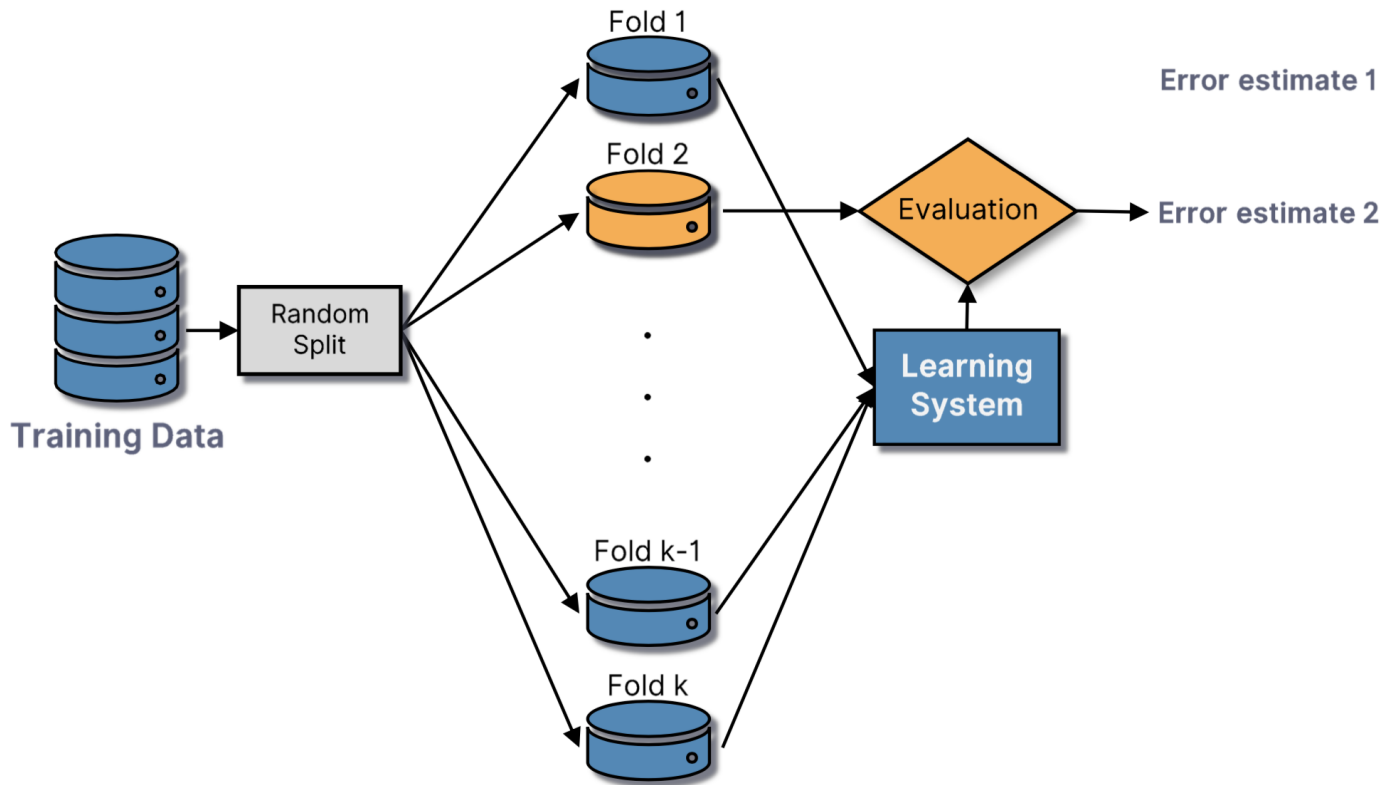
Cross-validation
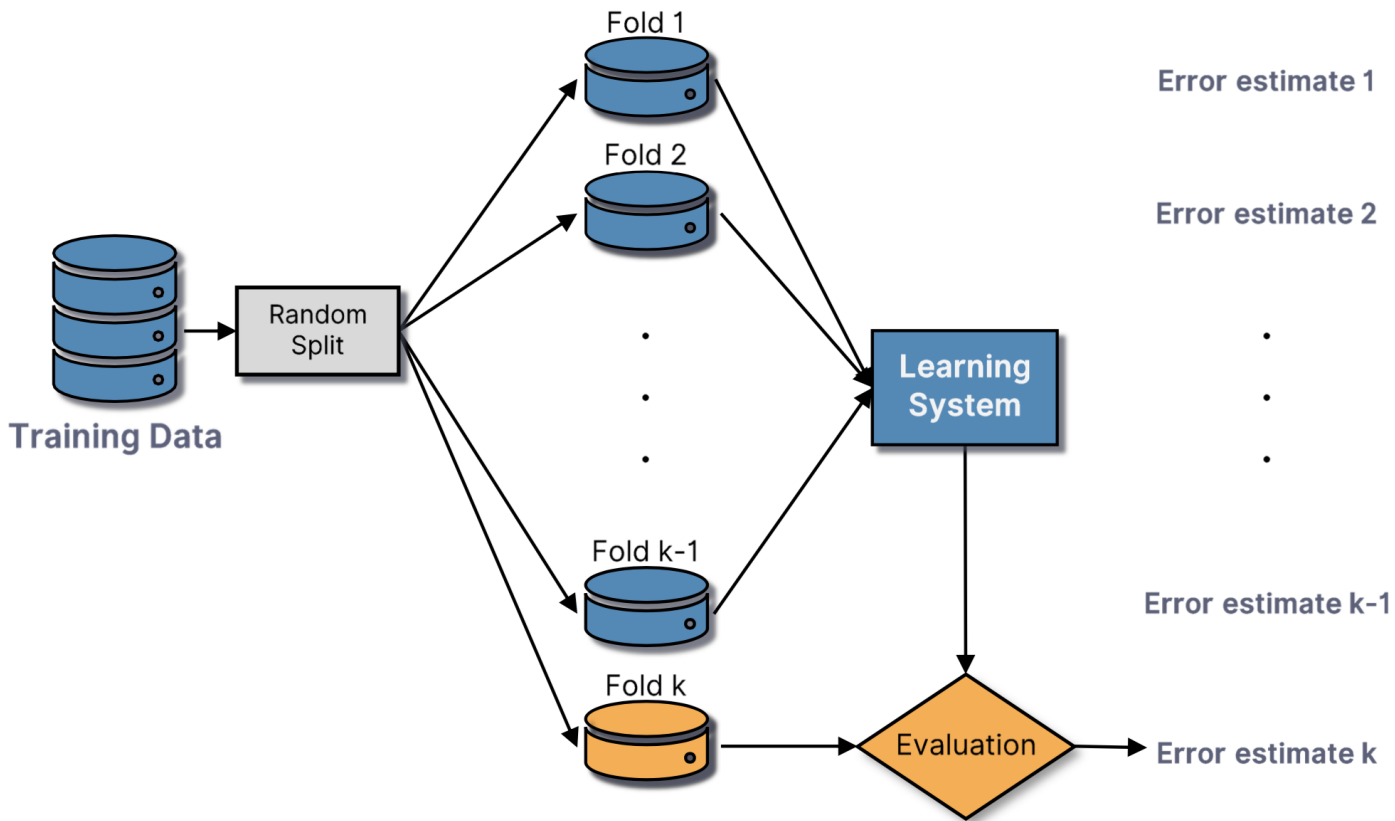
Cross-validation

Cross-validation
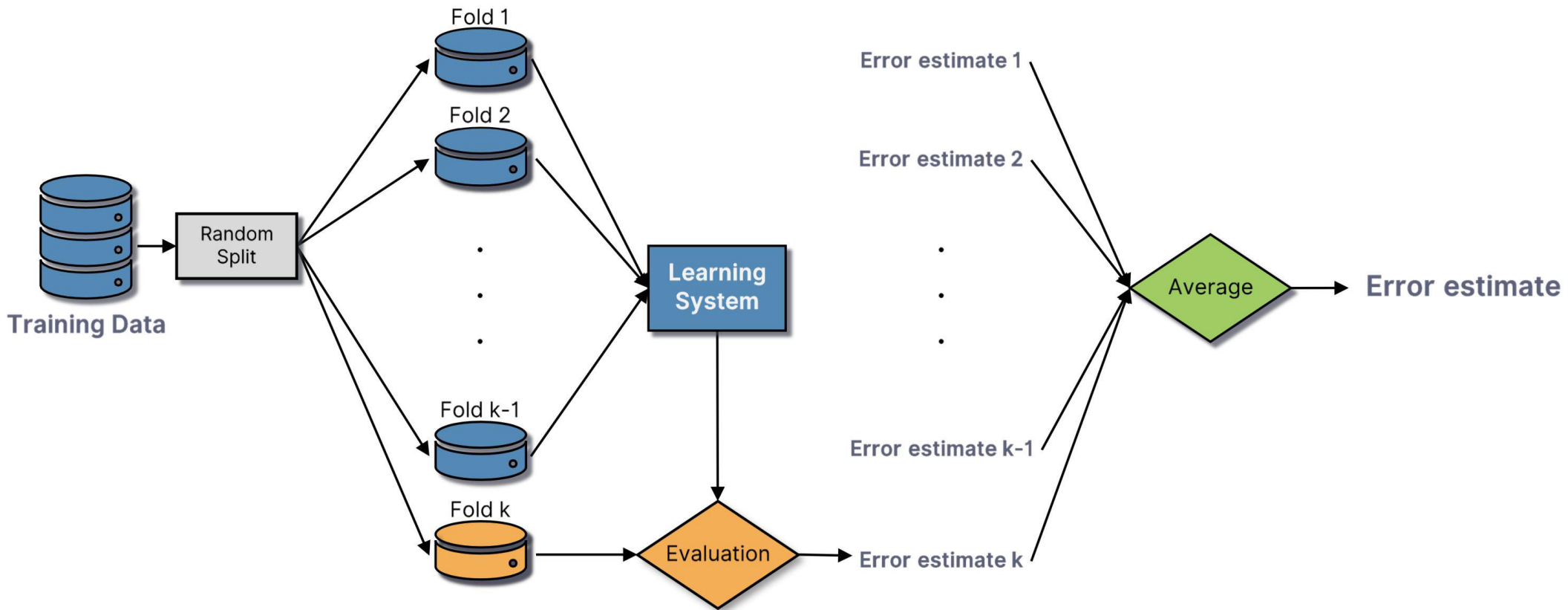
Improving generalization

- Early Stopping

➤ Stop the learning process when we notice that the validation error starts to increase

- Regularization

➤ Add a penalty term to the loss function to reduce model complexity, by imposing smoothing restrictions or a limit on the weight vector norm:

$$\min_{f} \sum_{i=1}^{n} V(f(\hat{x}_i), \hat{y}_i) + \lambda R(f)$$

Performance Evaluation

How do we evaluate an ML model?

- We measure the accuracy / error on test data:

- Accuracy: 4 correct out of 6 = 66.67%

- Error: 2 wrong out of 6 = 33.33%

How do we evaluate an ML model?

- We build the confusion matrix
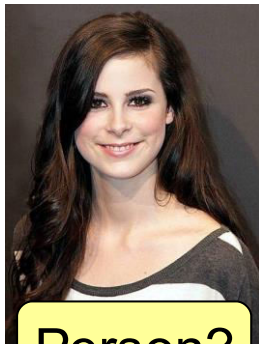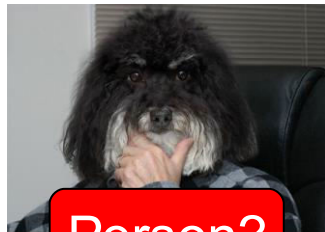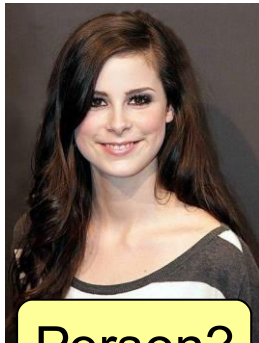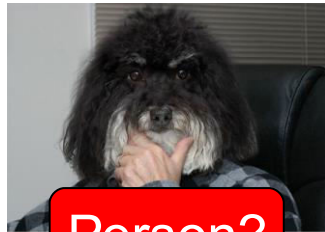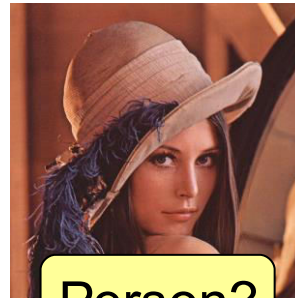

Car?   Person?   Person?   Person?   Dog?   Dog?
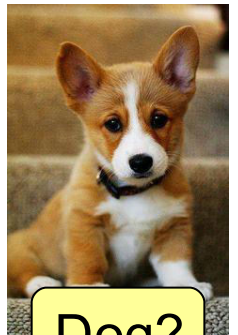
- Accuracy: the sum of the elements from the main diagonal divided by the sum of non-zero components (4/6)

- Error: the sum of the elements outside the main diagonal divided by the sum of non-zero components (2/6)

Predicted Actual	Car	Dog	Person
Car	1	1	0
Dog	0	1	1
Person	0	0	2

How do we evaluate an ML model?
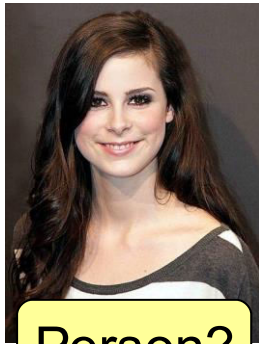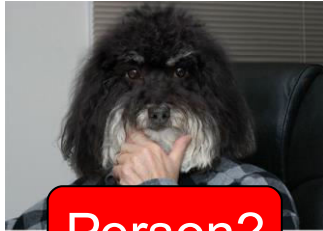
- Confusion matrix in the binary case
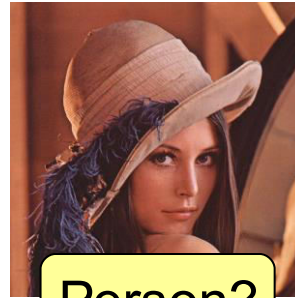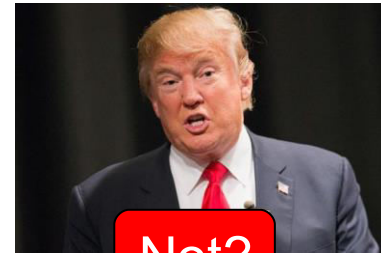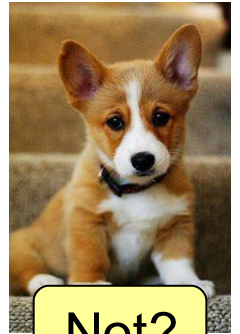
	Predicted YES	Predicted NO
Actual YES	True Positive	False Negative
Actual NO	False Positive	True Negative

How do we evaluate an ML model?
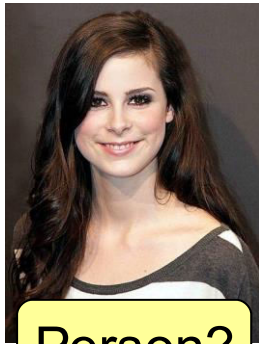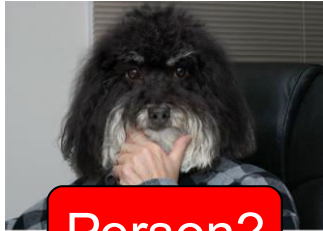
- Confusion matrix in the binary case
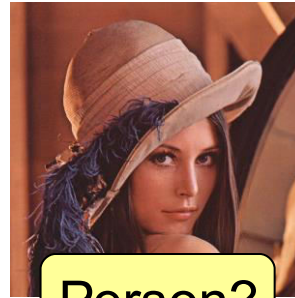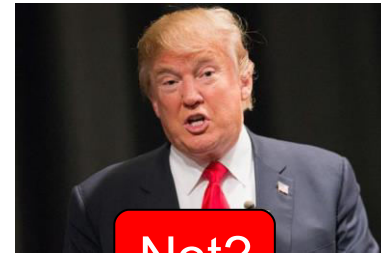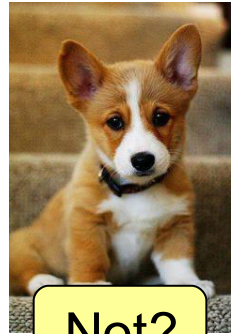
	Predicted YES	Predicted NO
Actual YES	2	False Negative
Actual NO	False Positive	True Negative
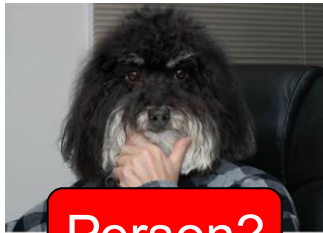
How do we evaluate an ML model?
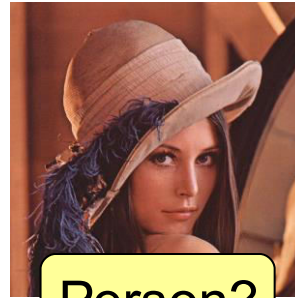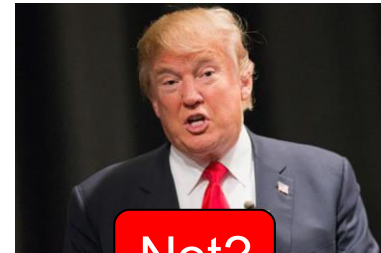
- Confusion matrix in the binary case

	Predicted YES	Predicted NO
Actual YES	2	1
Actual NO	False Positive	True Negative

How do we evaluate an ML model?

- Confusion matrix in the binary case
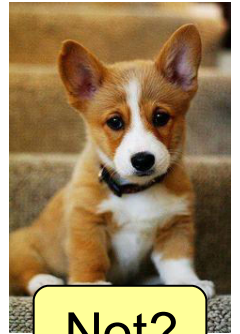
	Predicted YES	Predicted NO
Actual YES	2	1
Actual NO	1	True Negative

How do we evaluate an ML model?
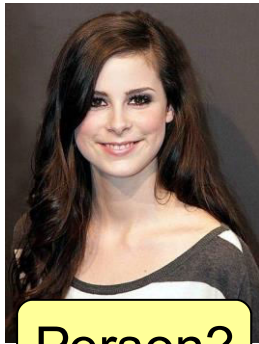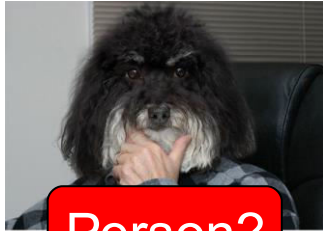
- Confusion matrix in the binary case
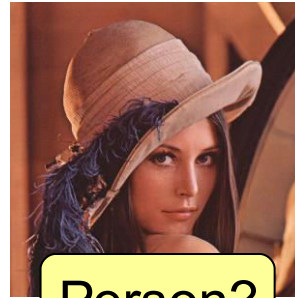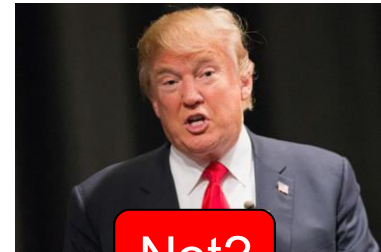
	Predicted YES	Predicted NO
Actual YES	2	1
Actual NO	1	2

How do we evaluate an ML model?

- We compute Precision and Recall
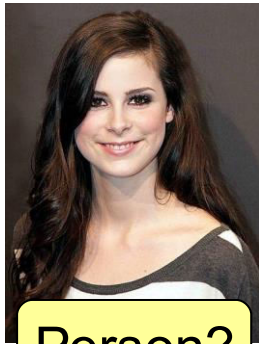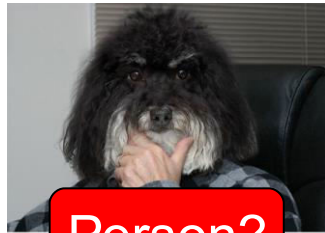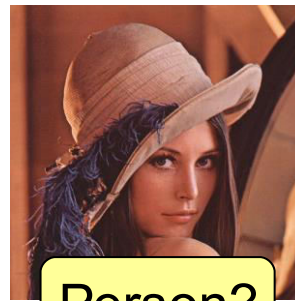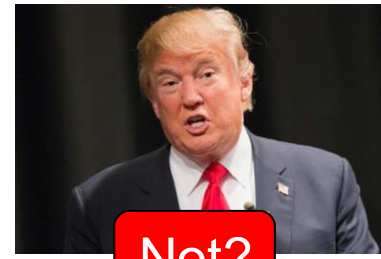
- Precision = TP / (TP + FP)

  = 66.67%

- Recall = TP / (TP + FN)

  = 66.67%

	Predicted YES	Predicted NO
Actual YES	2	1
Actual NO	1	2

How do we evaluate an ML model?

- We build the Precision-Recall curve
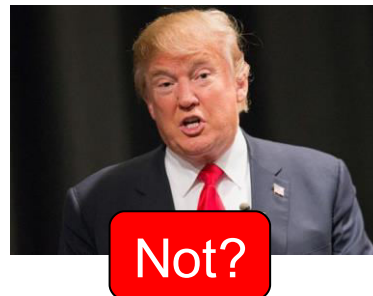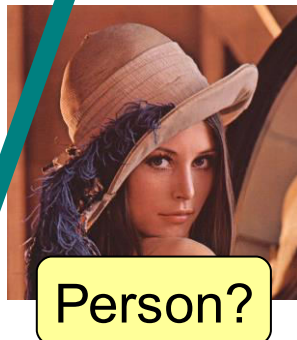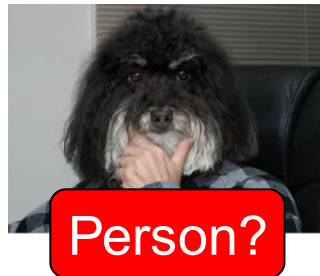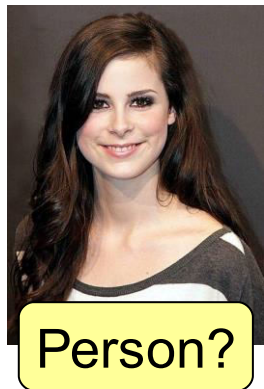
How do we evaluate an ML model?

- Precision-Recall curve
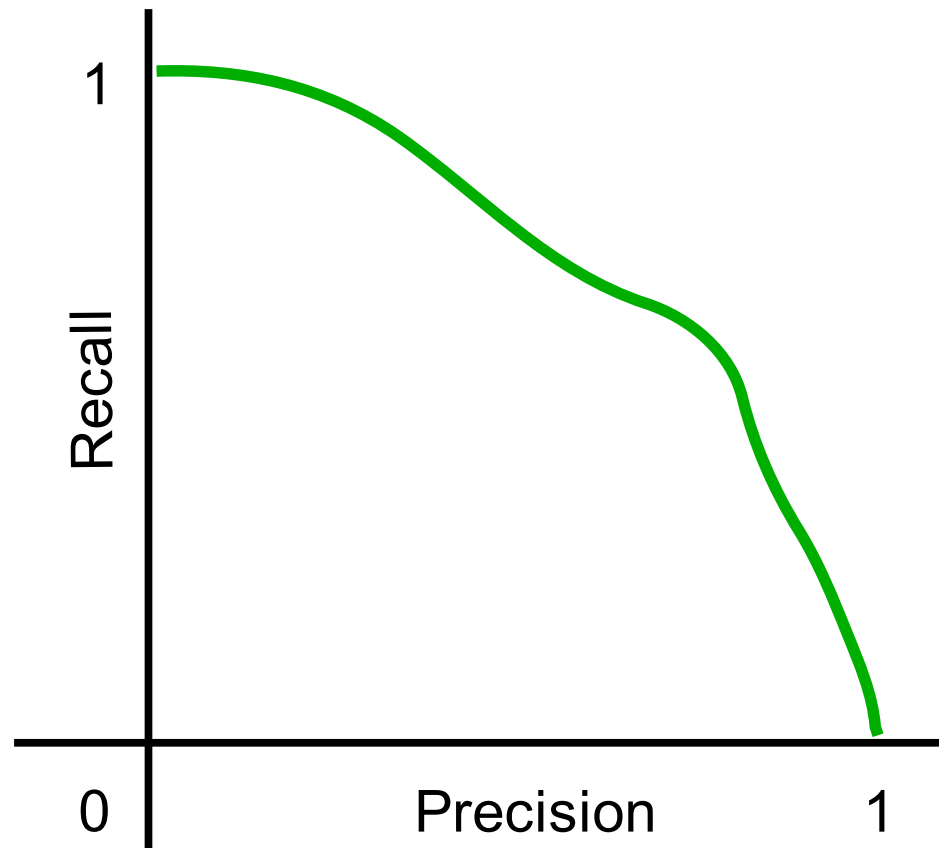
How do we evaluate an ML model?

- We compute the Average Precision
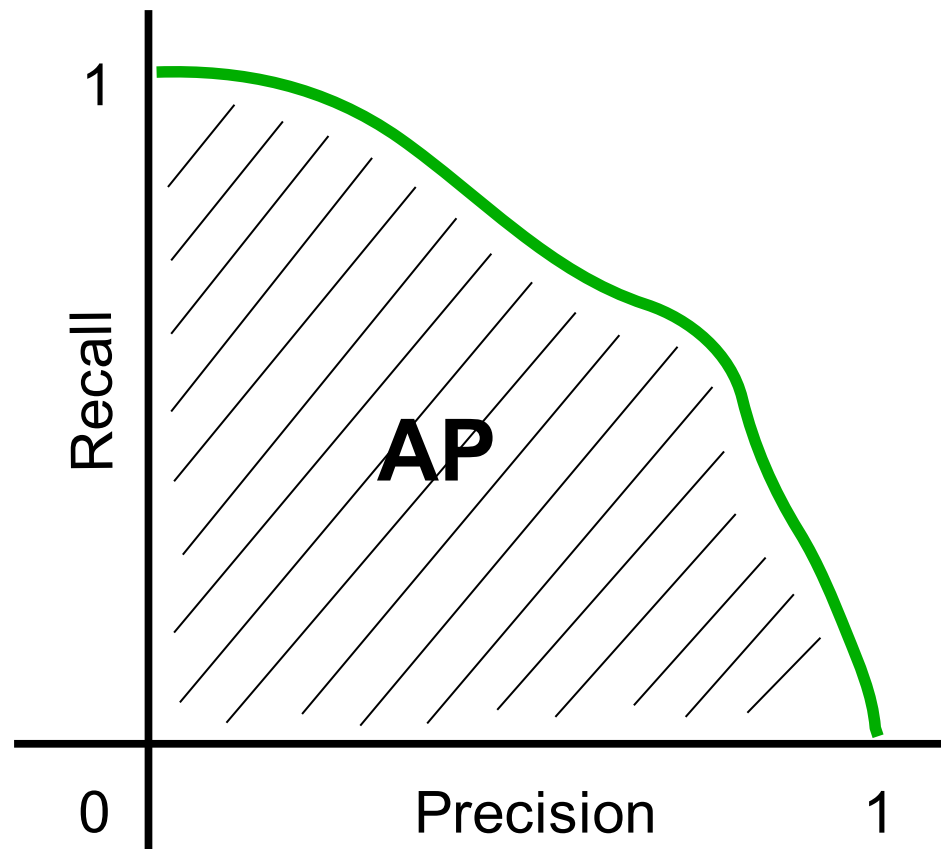
How do we evaluate an ML model?

- We compute the TP Rate and the FP Rate
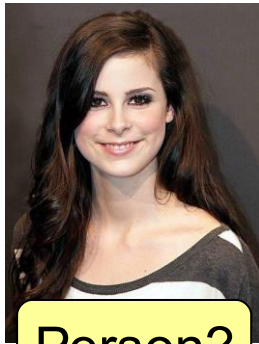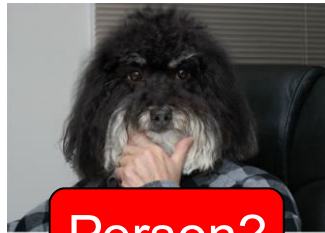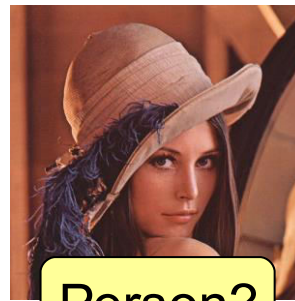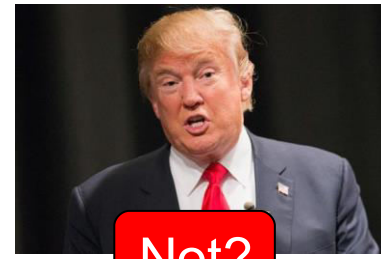
- TPR = TP / (TP + FP)

   = 66.67%

- FPR = FP / (FP + TN)

   = 33.33%

	Predicted YES	Predicted NO
Actual YES	2	1
Actual NO	1	2

How do we evaluate an ML model?

- ROC (Receiver Operating Characteristic) Curve
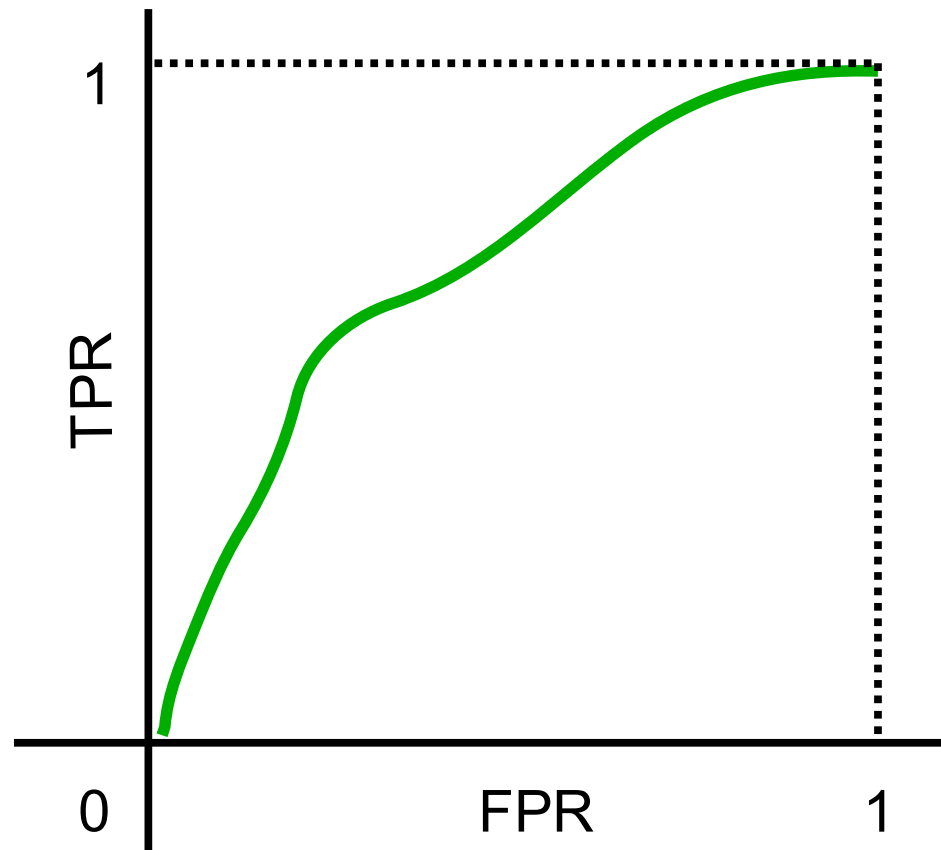
How do we evaluate an ML model?

- We compute the AUC (area under the ROC curve)
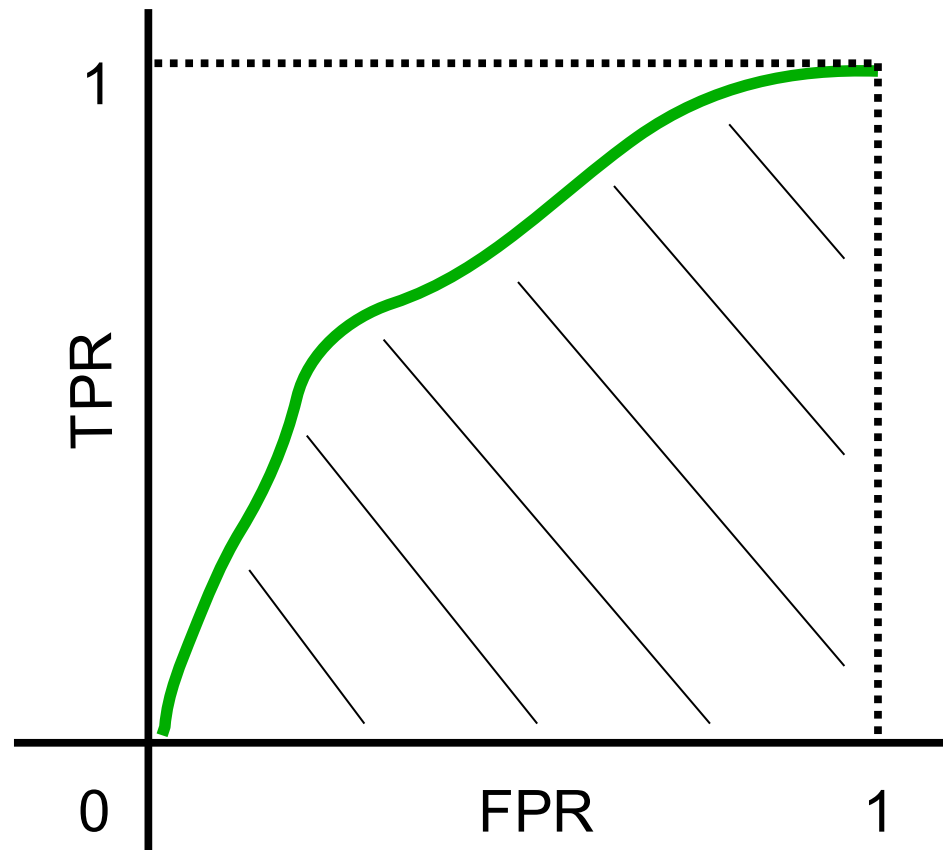
How do we evaluate an ML model?

- We compute the F_β score

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

- The F_1 score is the most commonly-used in practice:

➢ Gives equal importance to precision and recall

How do we evaluate an object detector?

- Intersection over Union (Jaccard index)
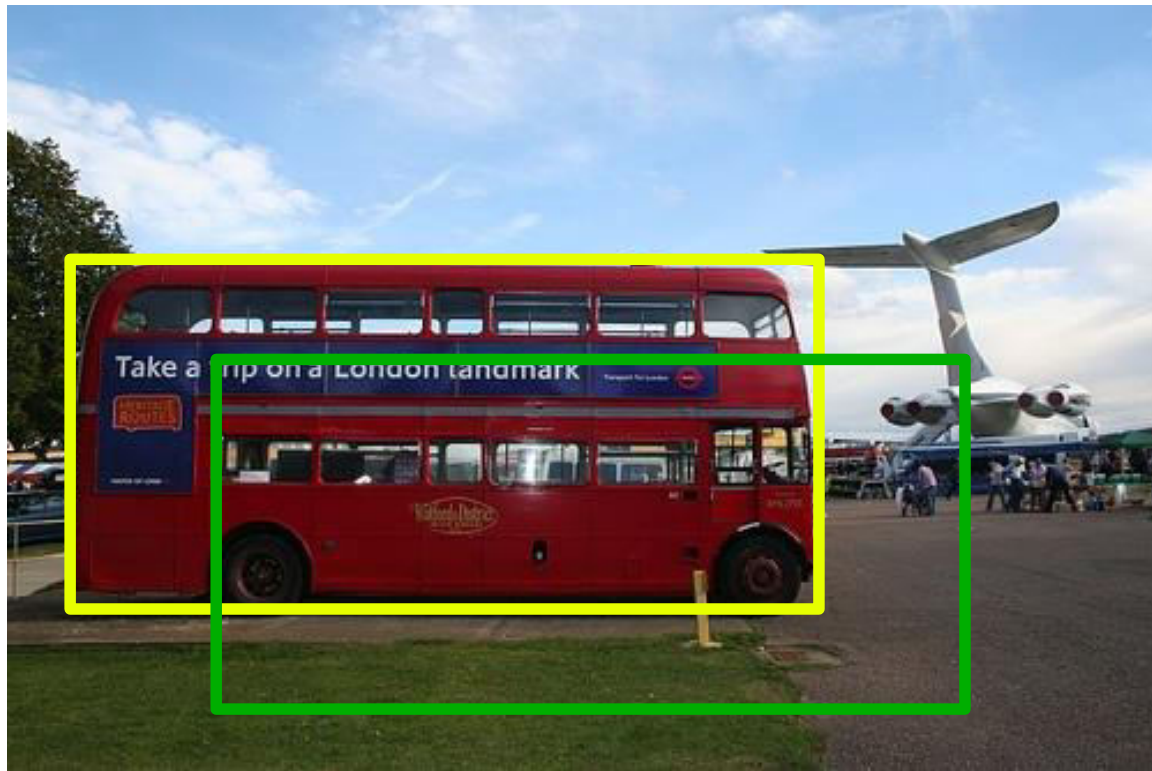
How do we evaluate an object detector?

- Intersection over Union (Jaccard index)

- Correct detection if IoU(A,B) > 0.5

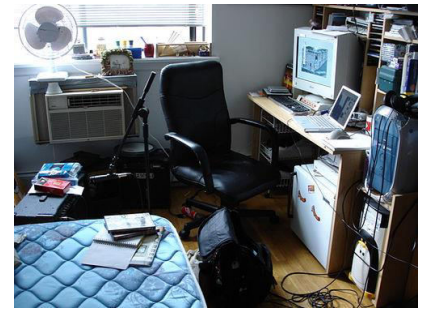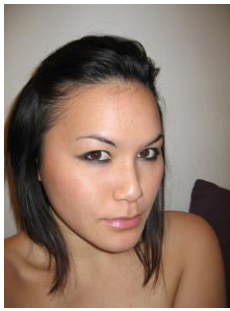# How do we evaluate a regression model?

- Mean squared error (MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2$$
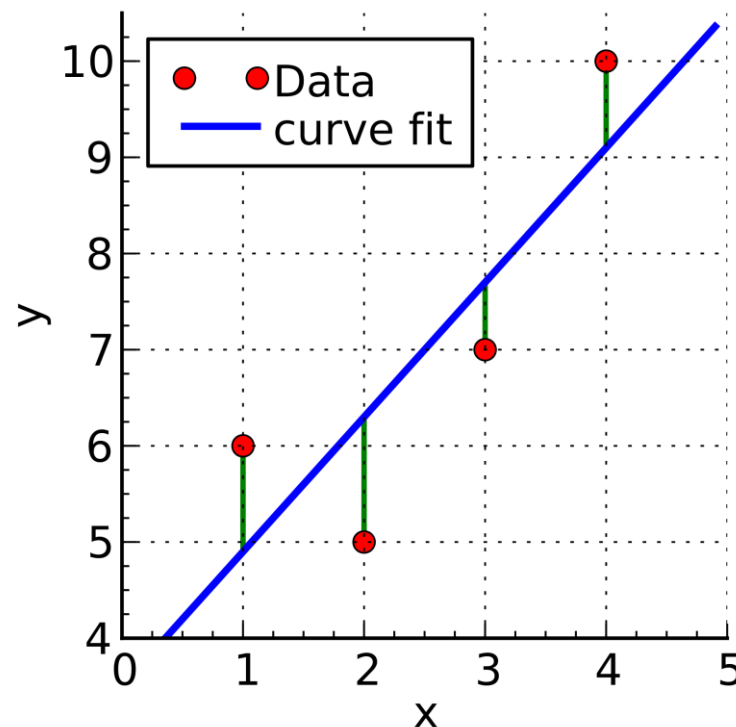
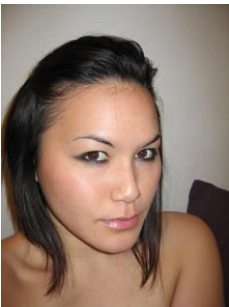# How do we evaluate a regression model?

- Order of difficulty according to humans

- Order of difficulty predicted by system

How do we evaluate a regression model?

- Kendall Tau correlation:

$$\tau_a = \frac{P - Q}{\frac{n(n-1)}{2}}$$

- Ordinal measure based on counting concordant (P) and discordant (Q) pairs:

$$P = |\{(i,j) : 1 \le i < j \le n, (x_i - x_j)(y_i - y_j) > 0\}|$$

$$Q = |\{(i,j) : 1 \le i < j \le n, (x_i - x_j)(y_i - y_j) < 0\}|$$

How do we evaluate a regression model?
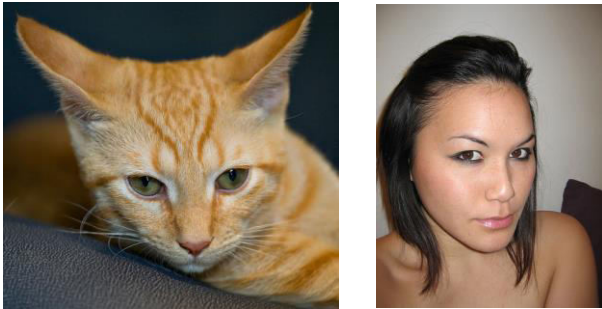
- Order of difficulty according to humans

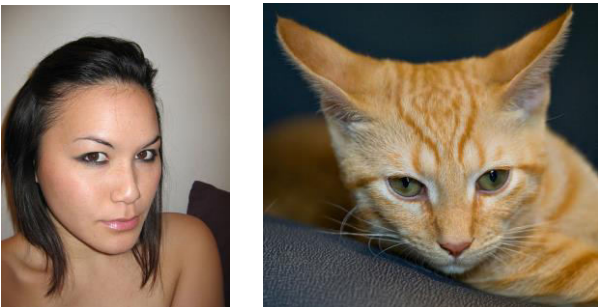- Concordant with order predicted by the system

How do we evaluate a regression model?
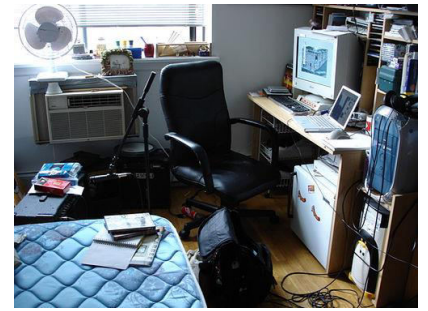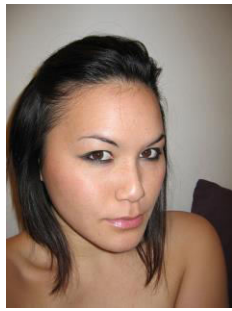
- Order of difficulty according to humans
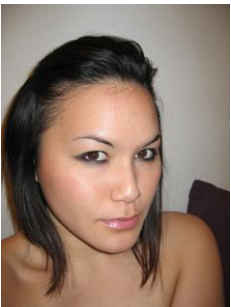

- Discordant with order predicted by the system

How do we evaluate a regression model?
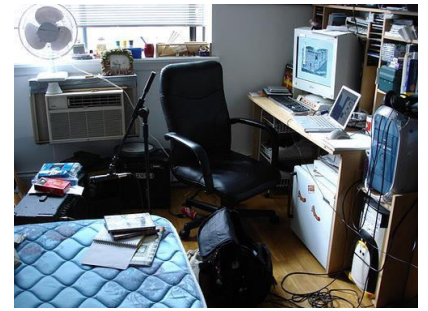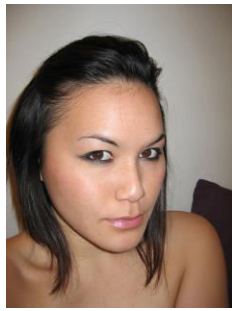
- What is the value of Kendall Tau correlation?

- P = ?, Q = ?

How do we evaluate a regression model?

- What is the value of Kendall Tau correlation?

- P = 8, Q = 2, Kendall Tau = (8-2) / 10 = 0.6