Introduction to Machine Learning

Rocío Alaiz Rodríguez

Full Professor

University of León (Spain)

rocio.alaiz@unileon.es

Overview

- Machine Learning
  - Concept
  - Application fields
  - Supervised, Unsupervised & Reinforcement learning
- Approaching a problem of learning from examples
- Building Machine Learning Models
- Supervised Learning models
  - K-NN
  - Naïve Bayes
  - Neural Networks
- Evaluating classifier performance

MACHINE LEARNING:
CONCEPT
and
APPLICATION FIELDS

Artificial Intelligence

REGINNA 4.0

Using AI, scientists find a drug that could combat drug-resistant infections

How Google Maps uses machine learning to predict bus traffic delays in real time

A.I. is transforming the job interview—and everything after

“AI-based ‘digital bridge’ enables paraplegic patient to walk”

Will AI cause or solve more problems?

universidad de león

Artificial Intelligence

Federal Oversight

White House unveils efforts to guide federal research of AI

Artificial intelligence prominent in automotive market

Deepfakes: faces created by AI now look more real than genuine photos

What is Intelligence?

REGINNA 4.0

Intelligence

...the ability to **reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly and learn from experience.**

Artificial Intelligence

Artificial Intelligence (AI)

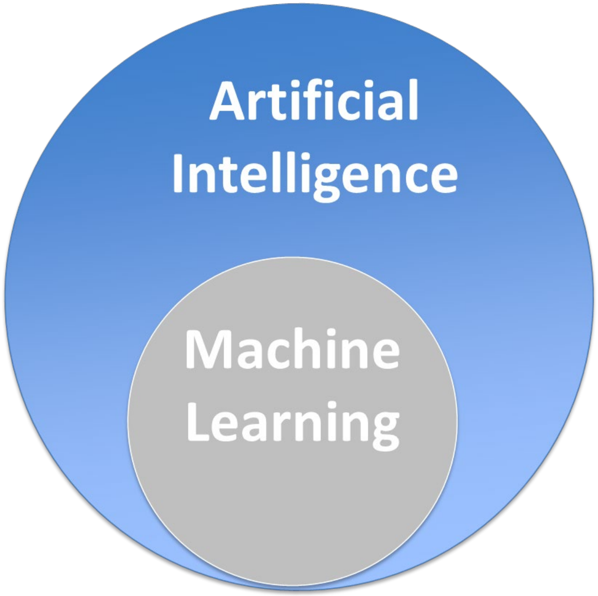
AI is the ability of a machine or computer system to emulate aspects of human intelligence:

- Reasoning
- Learning
- Intelligent behaviour and thought
- Capable of analyzing the environment and performing action

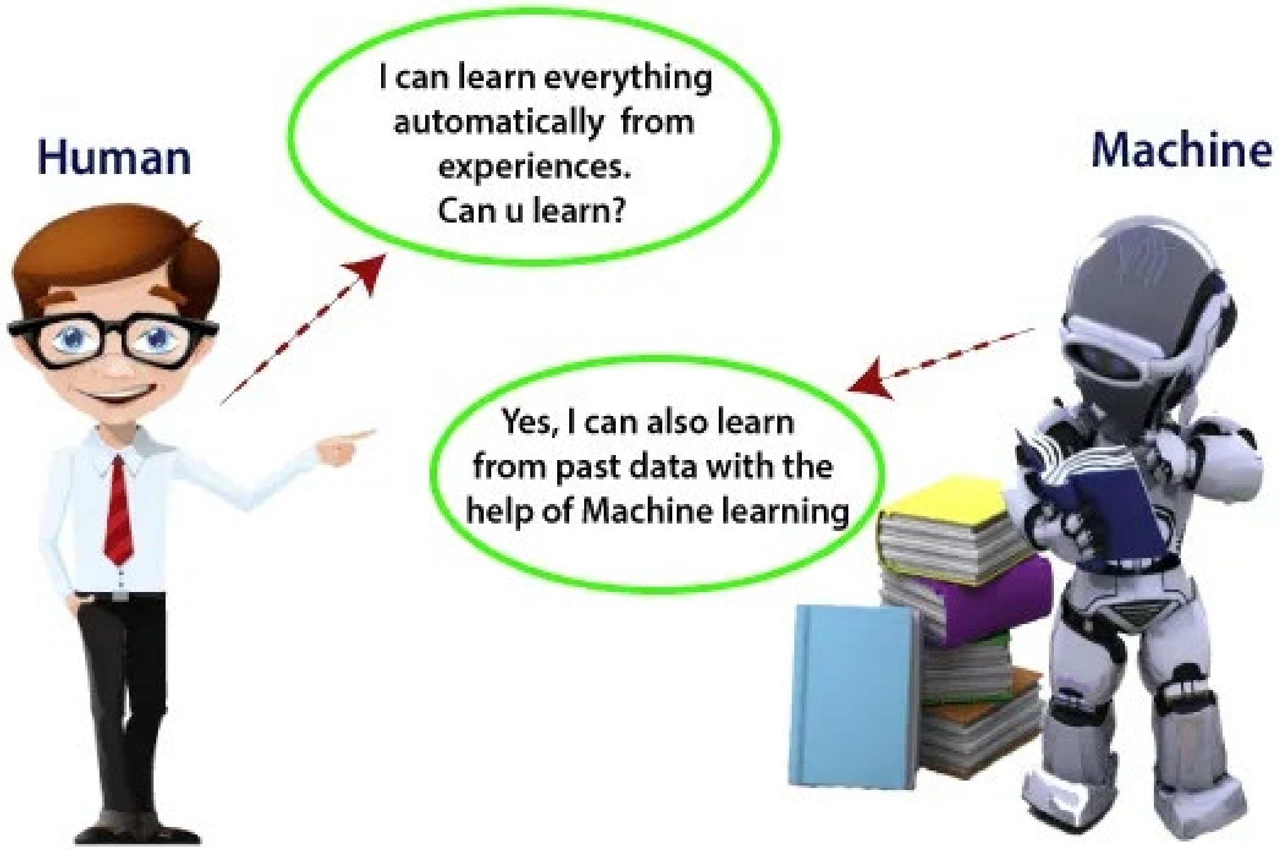
Machine Learning

Artificial Intelligence: The general picture

- *The effort to automate intellectual tasks normally performed by humans*.
- AI is a general field that includes Machine Learning and Deep Learning, **but also other approaches that do not involve any learning at all**



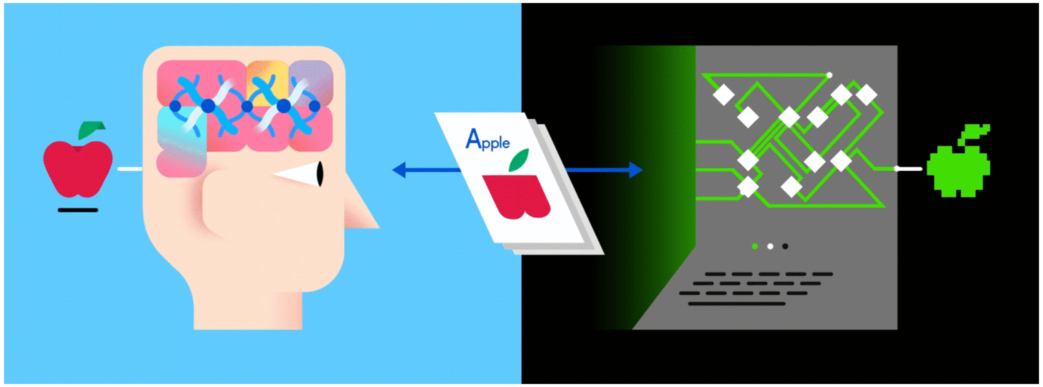
Machine Learning



Machine Learning

REGINNA 4.0

Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can **learn** from data.



Source: GumGum

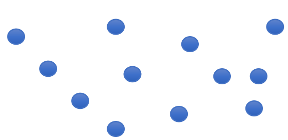
Machine Learning

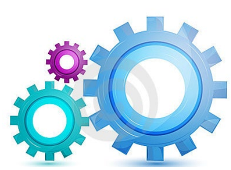
Machine Learning (Aprendizaje Automático) a branch of artificial intelligence, concerns the construction and study of systems that can **learn** from data.

Learning from experience

Learning from experience → data

Data





Machine Learning Techniques

To predict future events
Infer the causes of events

Machine Learning: Ingredients

- Technology allows us to have, store and process **large amounts of data (information society)**
- **High commercial and industrial interest** in the development of techniques for extracting knowledge from data, for finding patterns. Particularly in problems:
  - where algorithms do not exist
  - not well defined
  - informally proposed
- Great progress in the **development of algorithms and models by researchers**. Development of tools:
  - Classification (Assignment to a predefined category)
  - Regression (Estimation of a numerical value)

Typical tasks

- Why data-driven learning (machine learning)?
- Ability to mimic humans and replace them in monotonous tasks that require intelligence

  Handwriting recognition

- Develop systems that can automatically adapt to individual users

  Personalised news feeds, e-mail filters

- Extracting knowledge from large databases

  Shopping basket analysis

Typical applications

- Mundane - Ordinary Tasks (humans learn ordinary tasks since their birth but we can't describe how we do them).
  Character recognition

- Expert Tasks
  Quality control in manufacturing

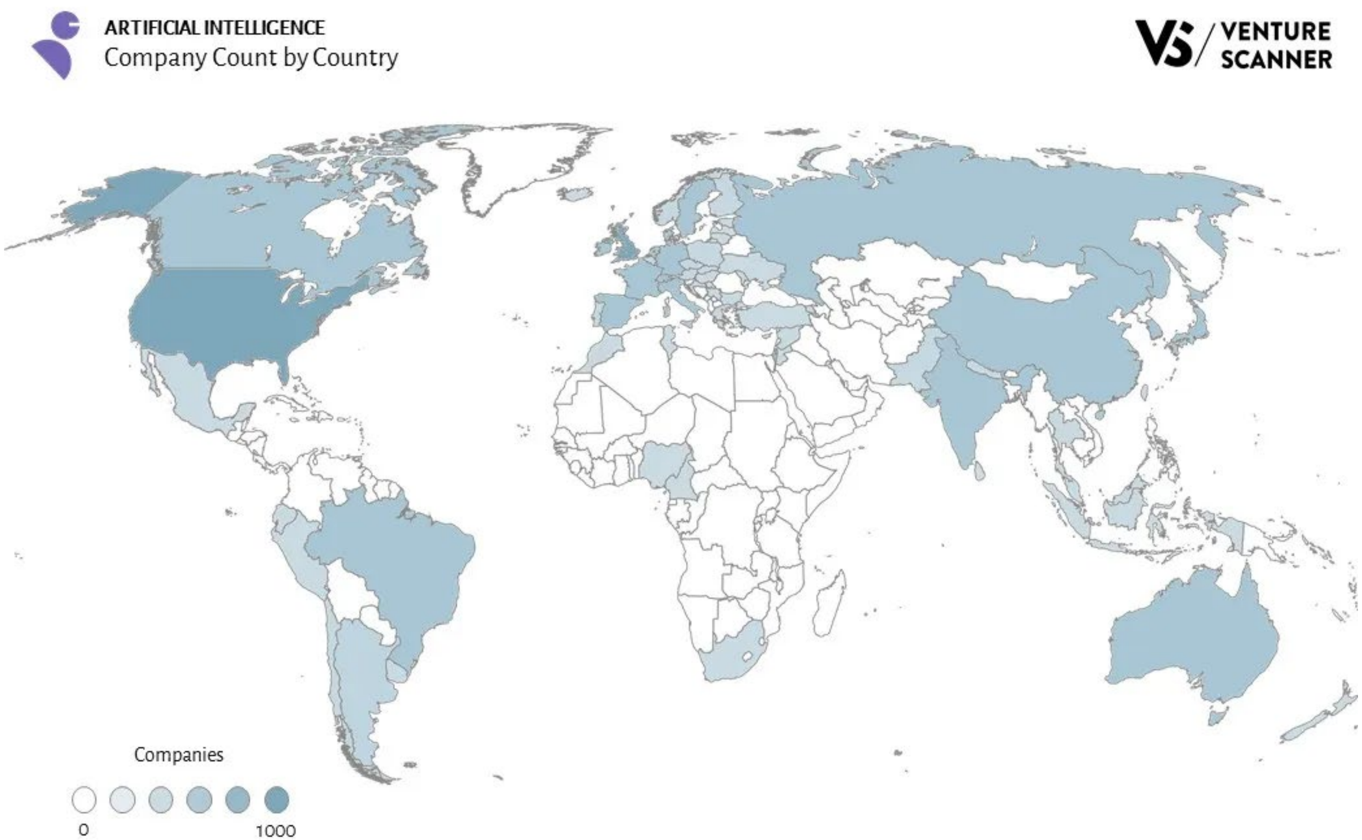
- Problems where there are no human experts
  Importance of certain genes in disease risk

- Situations where each user has his or her own target function
  Newspapers with personalised news, personalised advertising

- Problems where the volume of data makes it impossible for humans to perform any analysis
  Techniques capable of finding relationships within large DBs

Machine Learning Today

We see it practically every day, across every industry. From healthcare to agriculture, entertainment to transportation, AI applications are shaping our present and redefining our future.

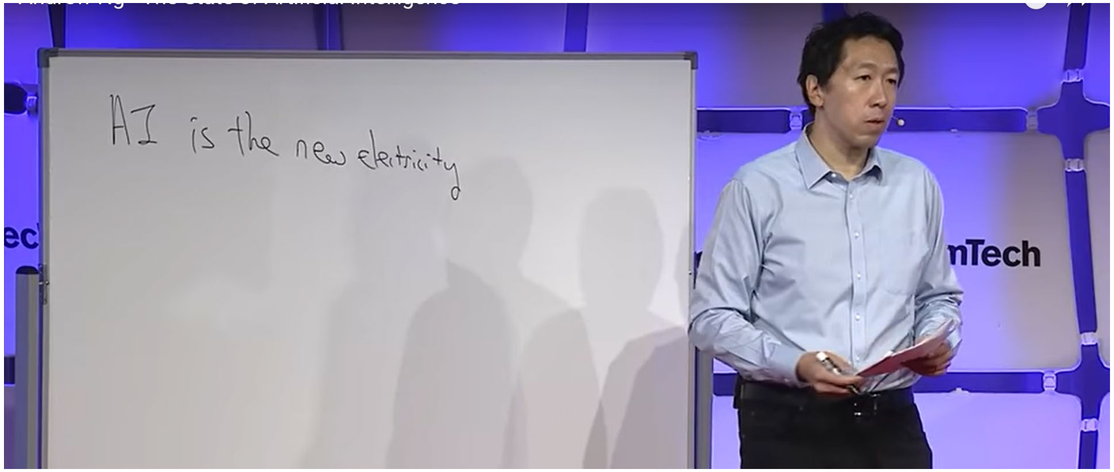


AI is the new Electricity [Andrew Ng]

REGINNA 4.0

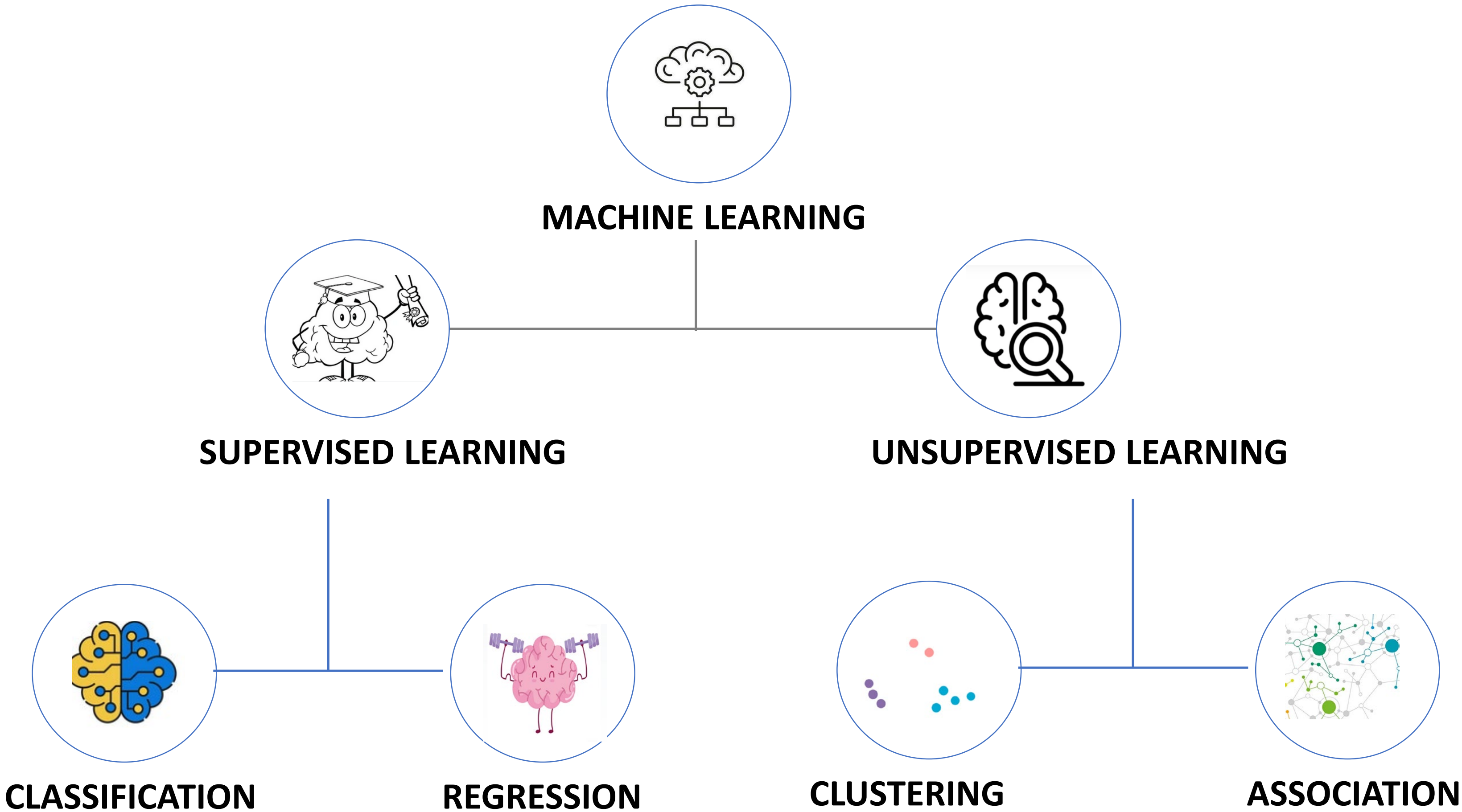
Dr. Andrew Ng:

- Globally recognized leader in AI
- He is Founder of DeepLearning.AI, Founder & CEO of Landing AI, General Partner at AI Fund, Chairman and Co-Founder of Coursera
- Professor at Stanford University
- Pioneer in machine learning and online education

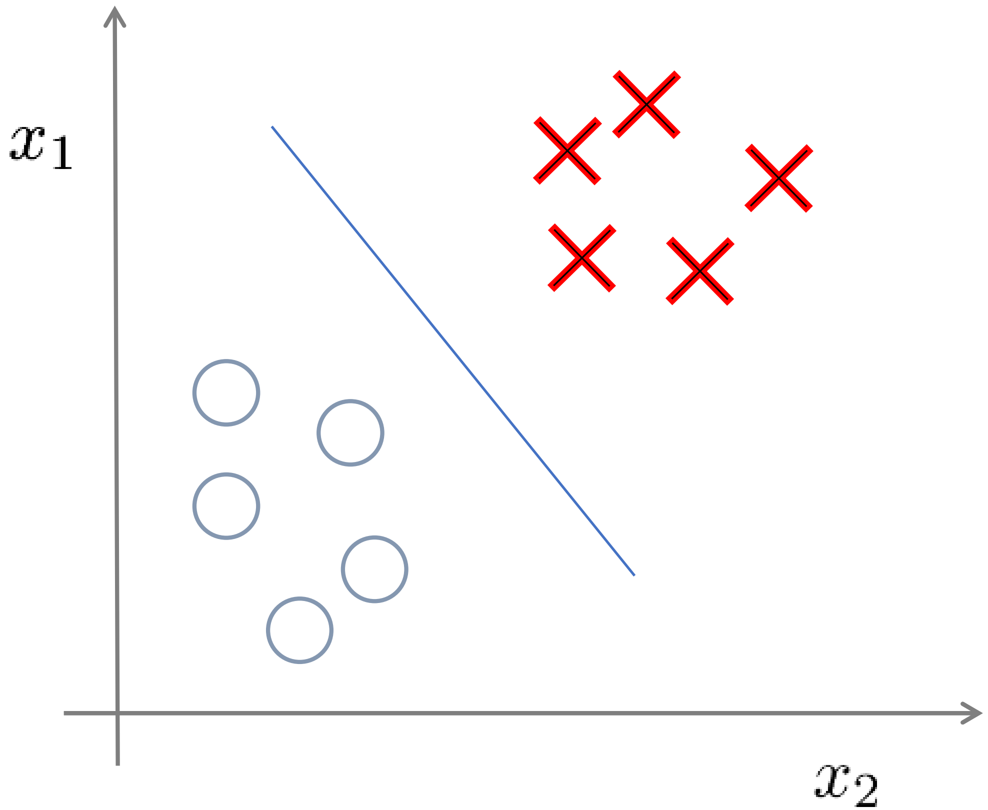


universidad de león

Machine Learning: Big tasks



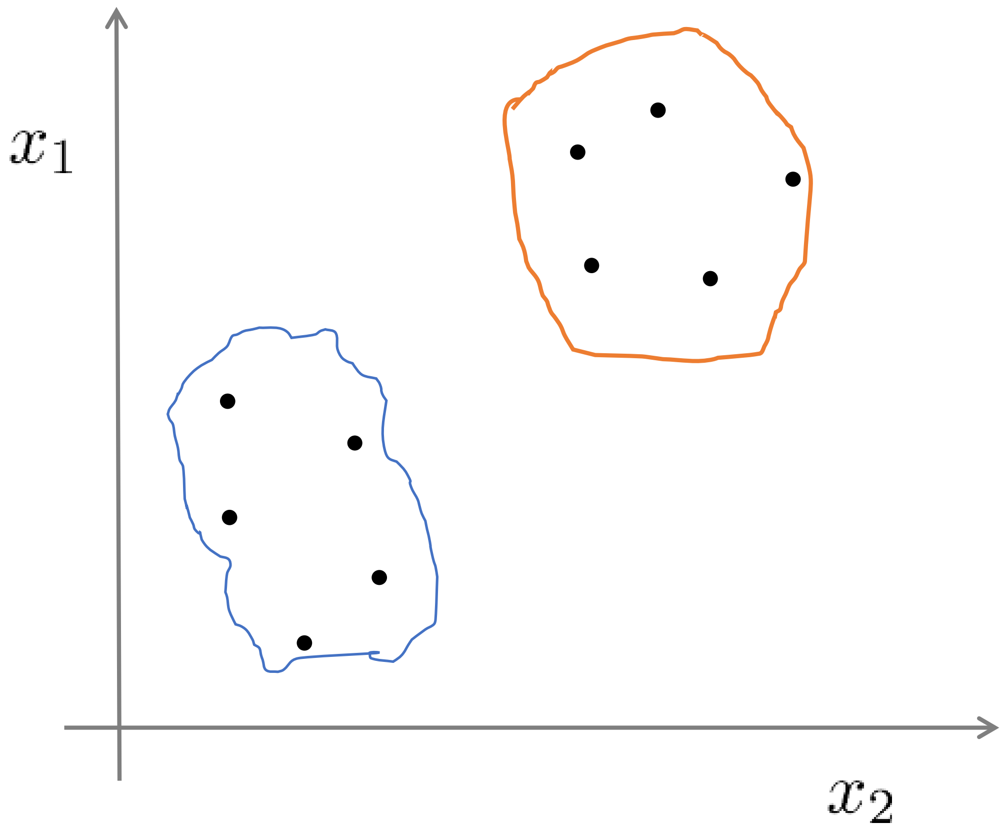
Supervised learning



WITH labels

Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(m)}, y^{(m)})\}$

Unsupervised learning



Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)}\}$ WITHOUT labels

Supervised learning

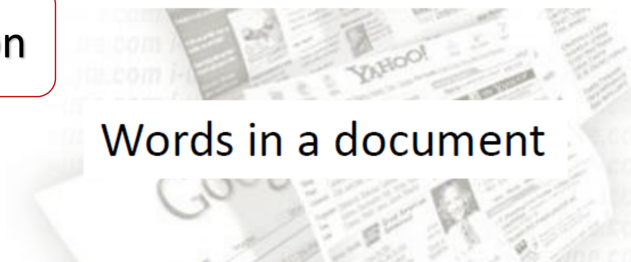
REGINNA 4.0

We have a set of labelled data (examples + labels)

Feature Space $\mathcal{X}$

Label Space $\mathcal{Y}$

Classification

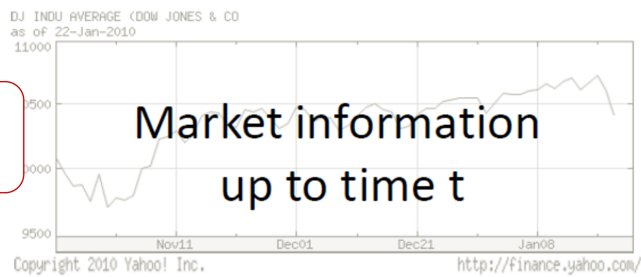


Words in a document

“Sports”
“News”
“Science”
…

Discrete labels (categories)

Regression



Market information up to time t

Share Price
“$ 24.50”

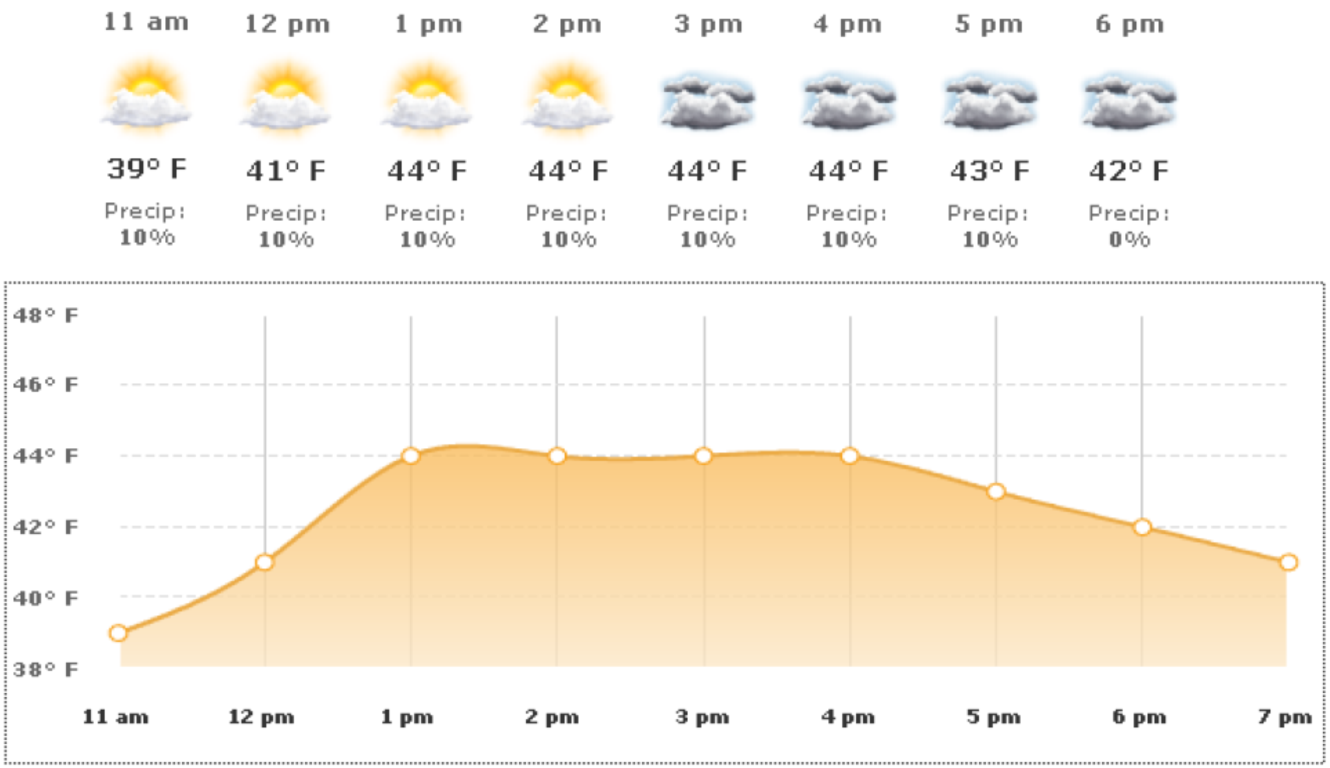
Continuos numerical values

Task: Given $X \in \mathcal{X}$, predict $Y \in \mathcal{Y}$.

universidad de león

Supervised learning problems

Regression or Classification?



Temperature/Weather Forecast

Supervised learning problems

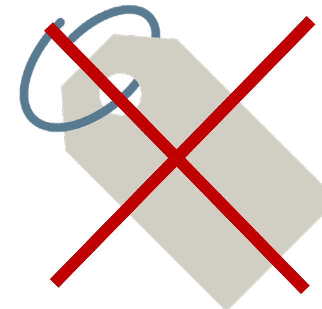
Regression or Classification?

Input (x)	Output
Home features (#bedrooms, size,…)	Price
Advertisement, user info	Click on ad? (Yes/No)
Image	Object (1,2,…,1000)
Age, sex, cholesterol, #Cigarettes, blood sugar, family history	Heart disease (True/False)
Employee's atributes(seniority, income, department, distance from home,..)	How long until an employee looks for another job
Age, level of education,area, job title,..	Income

Unsupervised learning

REGINNA 4.0

We have an unlabelled dataset (examples)
Also known as "learning without a teacher"



Feature Space $\mathcal{X}$

Words in a document

Word distribution
(Probability of a word)

Task: Given $X \in \mathcal{X}$, learn $f(X)$.

Unsupervised learning problems

Cluster similar items, for example, images

Goldberger et al.

Unsupervised learning problems

REGINNA 4.0

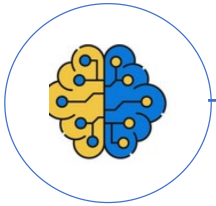
Cluster similar items, for example, customers

universidad de león

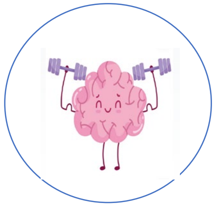
Types of machine learning

SUPERVISED LEARNING

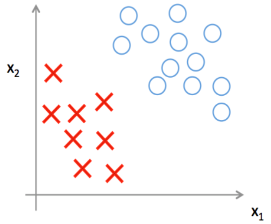
UNSUPERVISED LEARNING



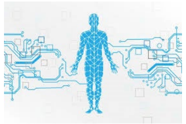
CLASSIFICATION

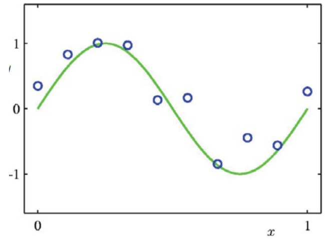


REGRESSION

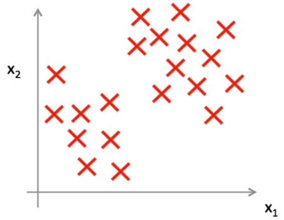


Data with labels (finite set of symbolic classes)





Data with labels (numerical label)



Data

No labels

universidad de león

APPROACHING A PROBLEM OF LEARNING FROM EXAMPLES

Hands-on an illustrative problem

What do we need ?

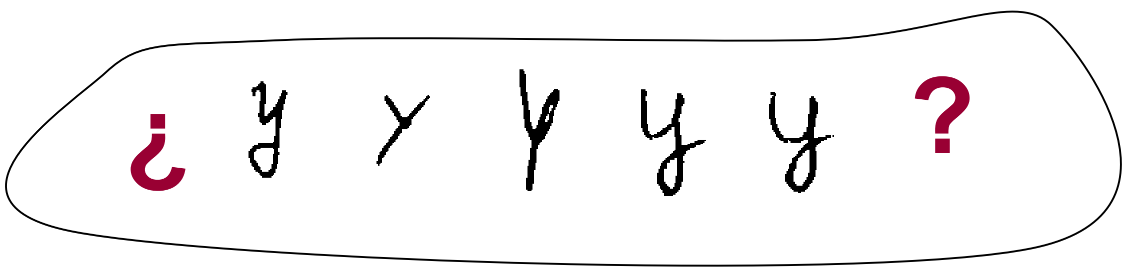
- A system that verifies that the person is who he/she claims to be to enter the security room.

¿user?

universidad de león

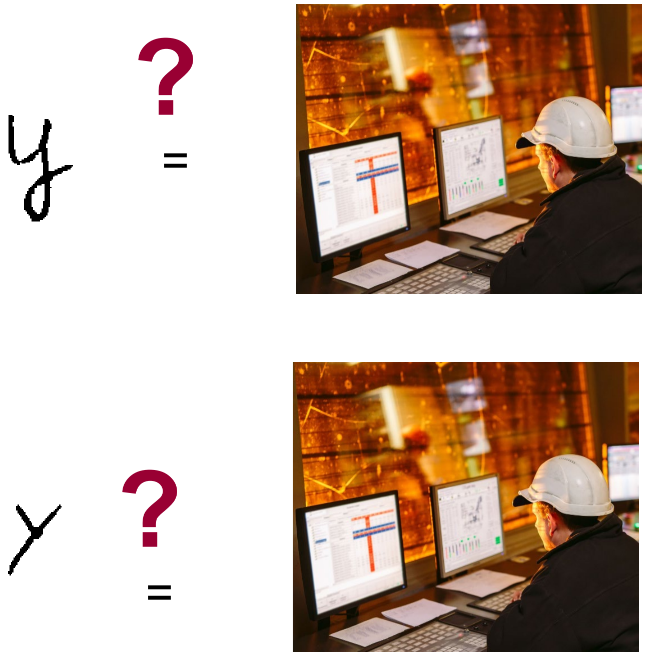
Identity verification with biometric features

Handwriting



Classification

- It verifies that the person is who he/she claims to be.
- It is based on a pattern recognition system.



Basic steps to create the system

- Dataset
- Training the model
- Model test
- Model evaluation

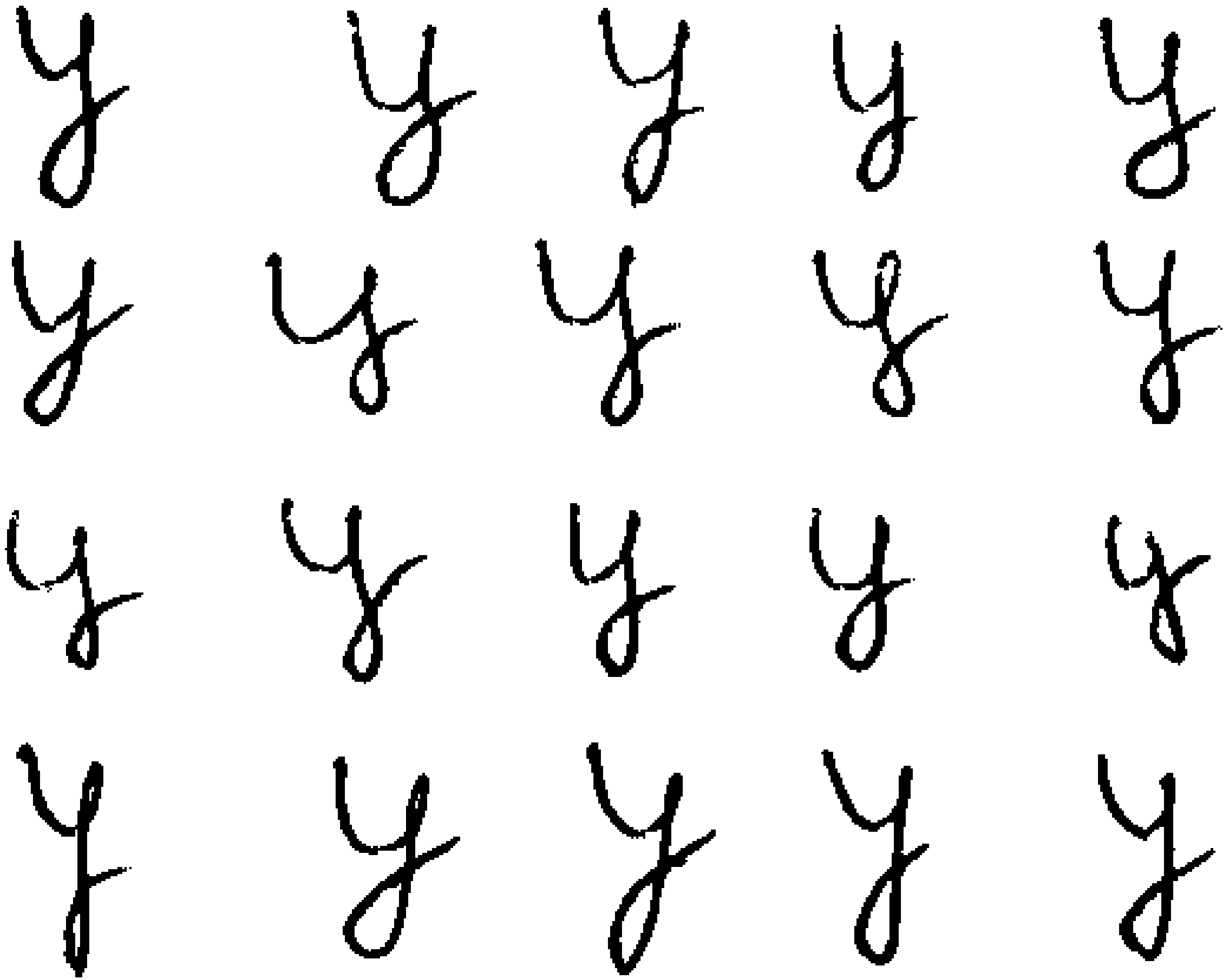
Basic steps to create the system

- Dataset
- Training the model
- Model test
- Model evaluation

$d = 1$

User X

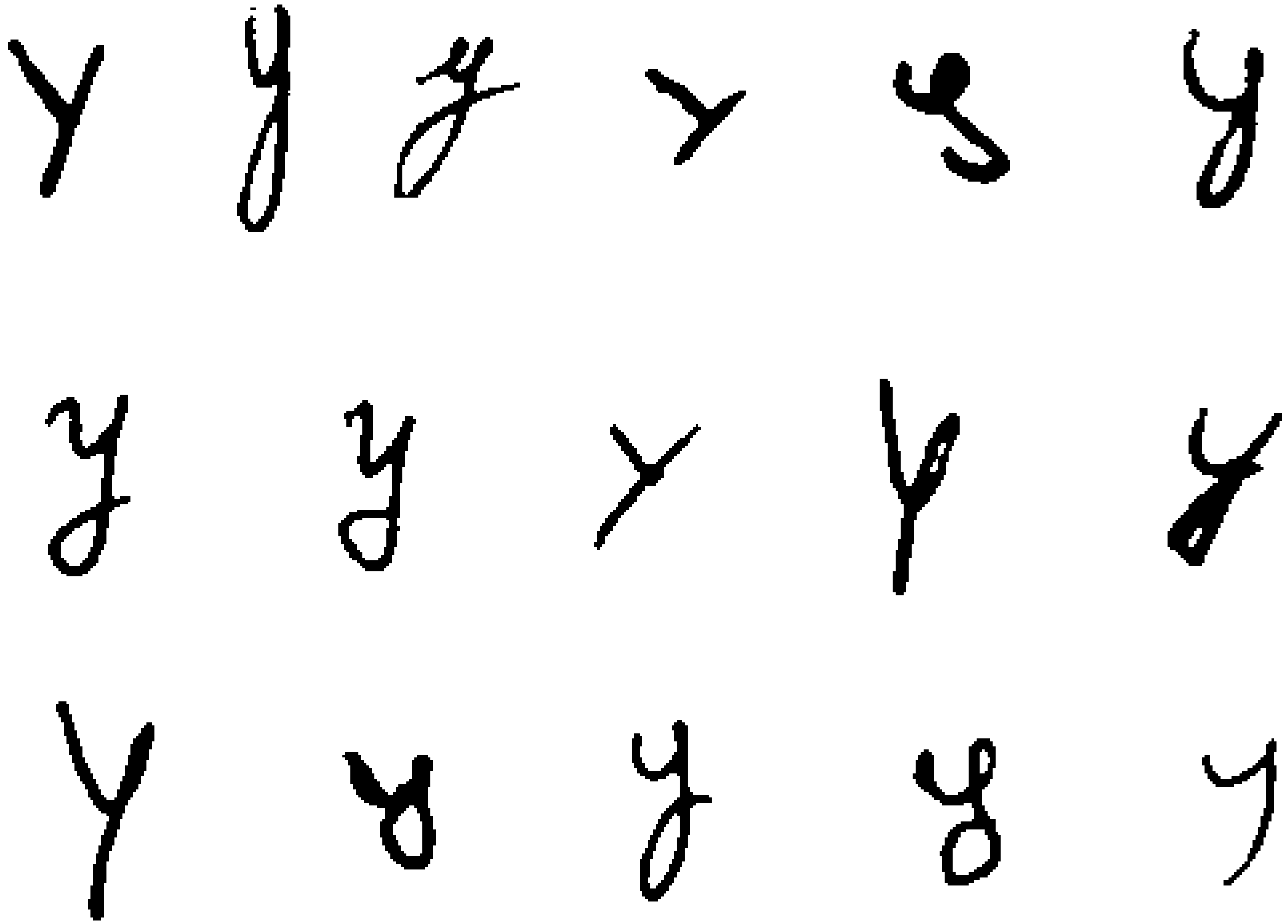
Handwriting “y”



$d = 0$

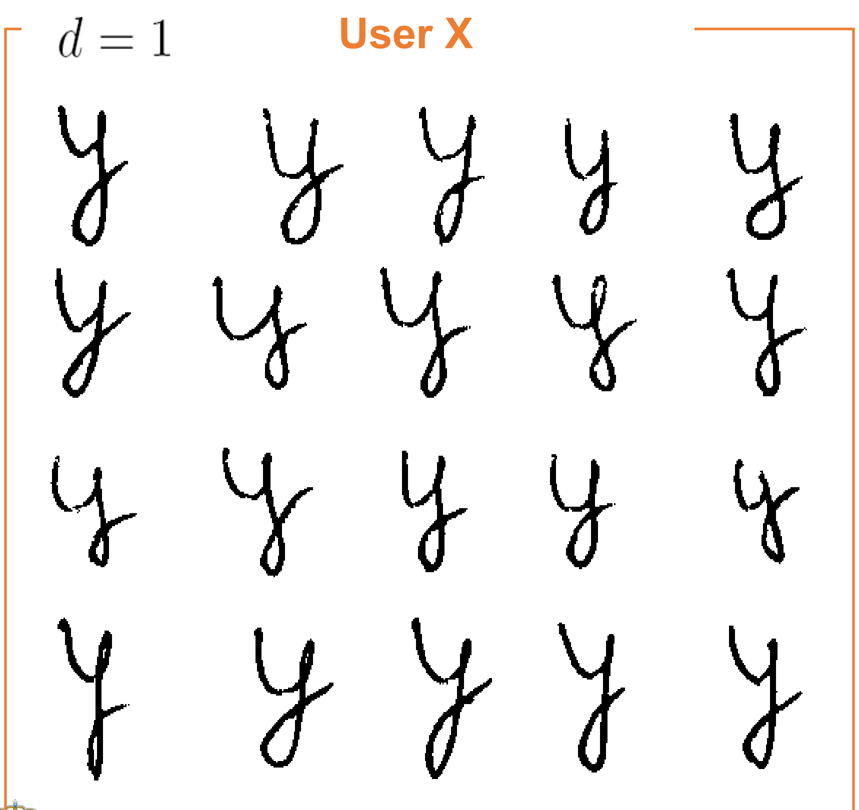
Other users

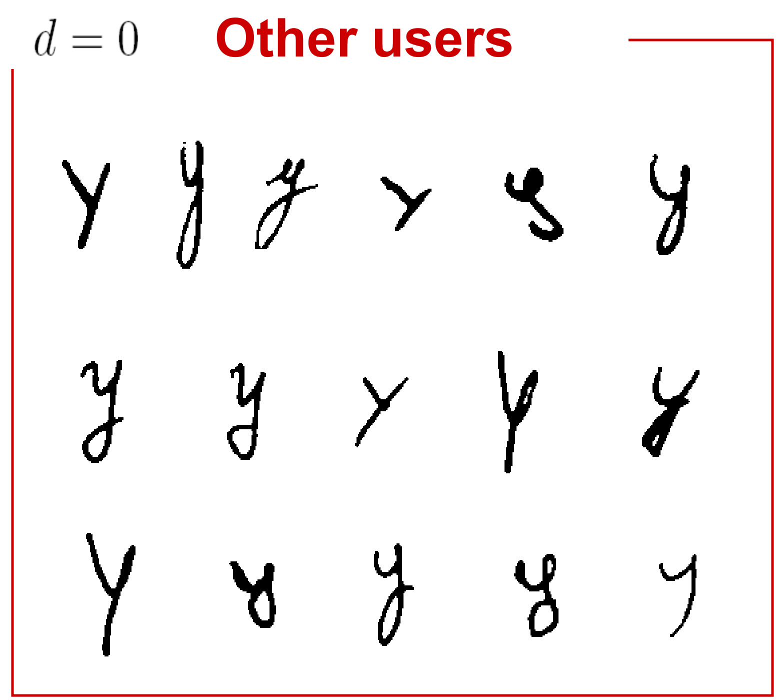
Handwriting “y”



universidad de león

Available dataset





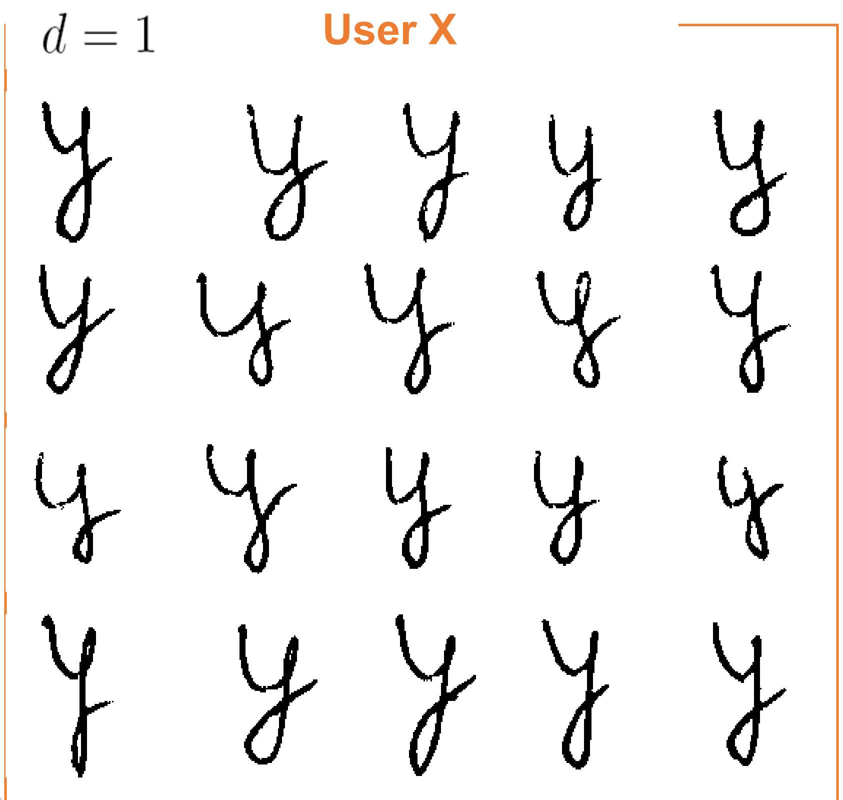
Basic steps to create the system

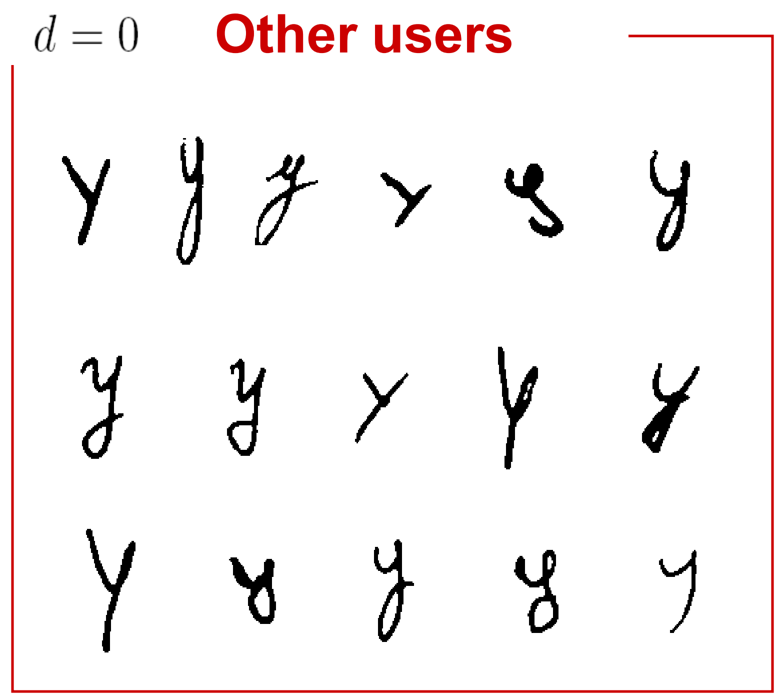
- Dataset
- Training the model
- Model test
- Model evaluation

What can **YOU** do?

TRY TO LEARN

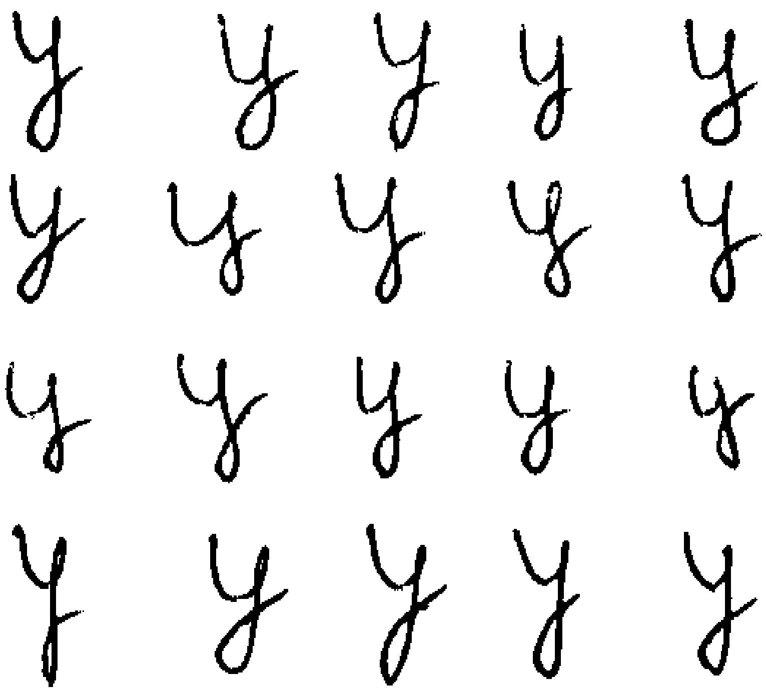
Model training



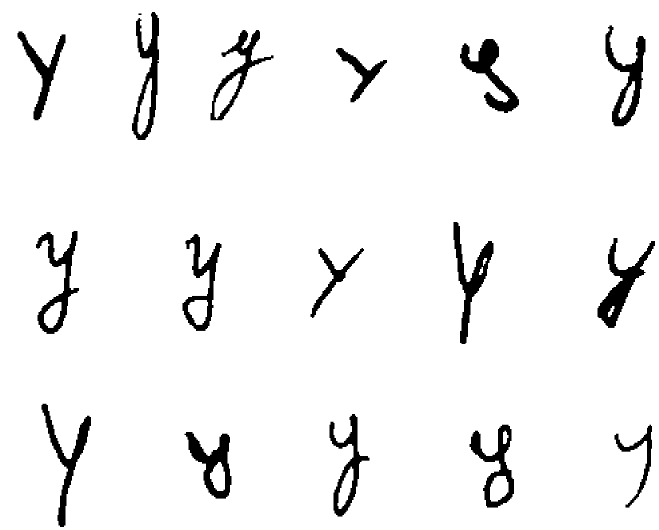


Model training

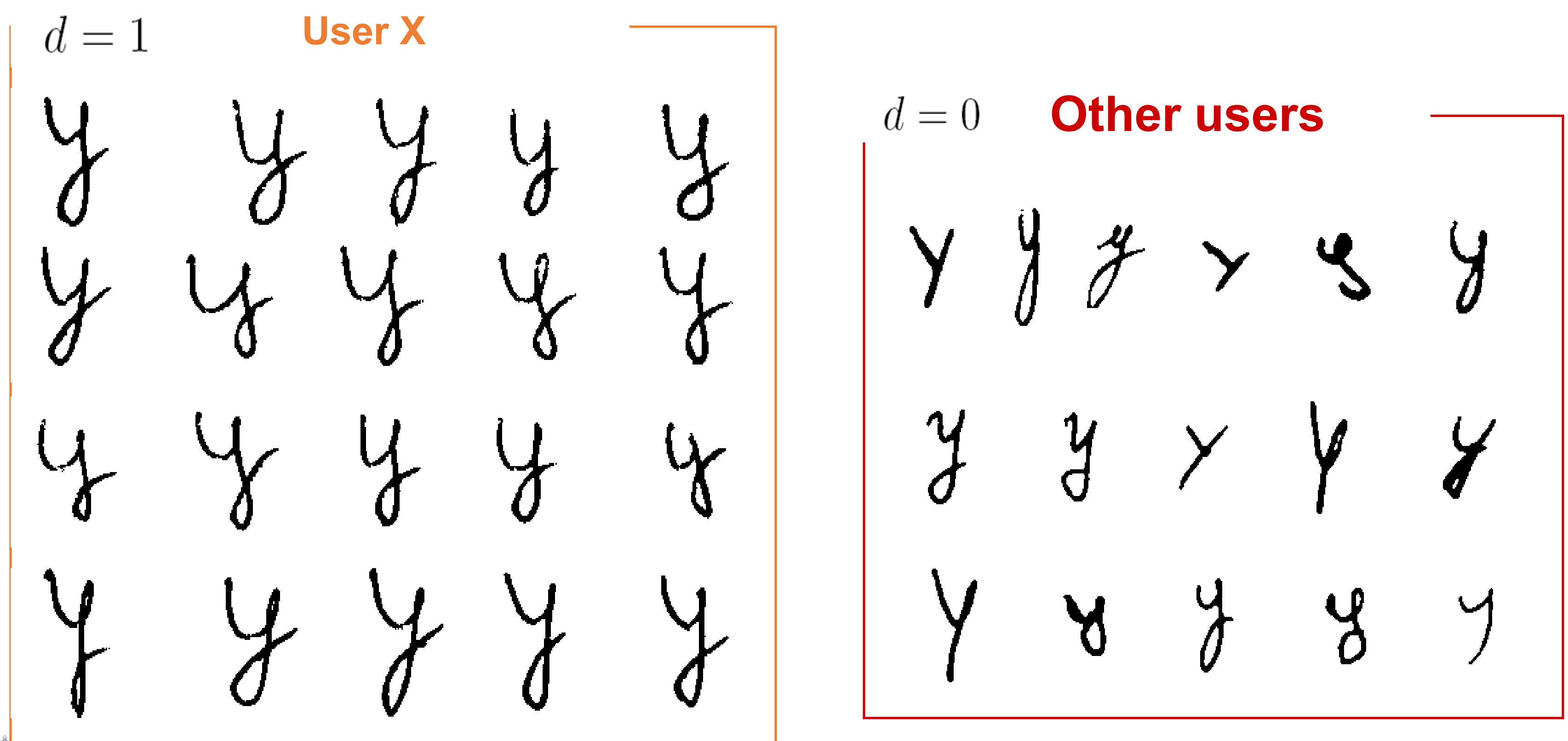
$d = 1$ **User X**



$d = 0$ **Other users**



Available dataset



Basic steps to create the system

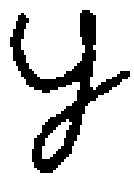
- Dataset
- Training the model
- Model test
- Model evaluation

$$\widehat{d} = 1$$

y

$\boxed{\text{User X} \quad d = 1}$

$$\widehat{d} = 0$$



$\boxed{\begin{array}{c}\text{User X} \\ d = 1\end{array}}$

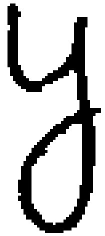
False rejection

Basic steps to create the system

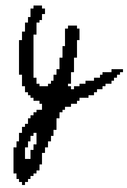
- Dataset
- Training the model
- Model test
- Model evaluation

TEST

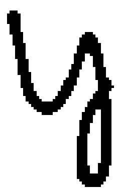
¿ $\widehat{d} = 0$ **or** $\widehat{d} = 1$?



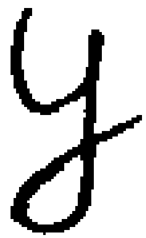
$d =$ $\widehat{d} =$



$d =$ $\widehat{d} =$

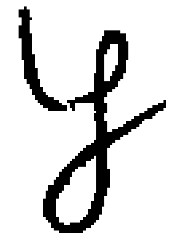


$d =$ $\widehat{d} =$

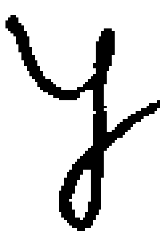


$d =$ $\widehat{d} =$

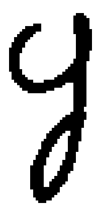
$d =$ $\widehat{d} =$



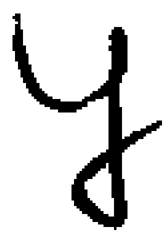
$d =$ $\widehat{d} =$



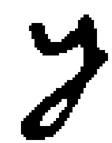
$d =$ $\widehat{d} =$



$d =$ $\widehat{d} =$



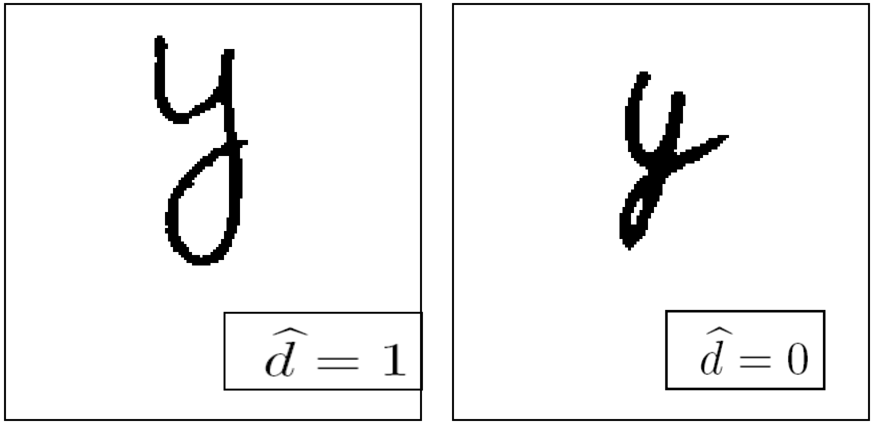
$d =$ $\widehat{d} =$

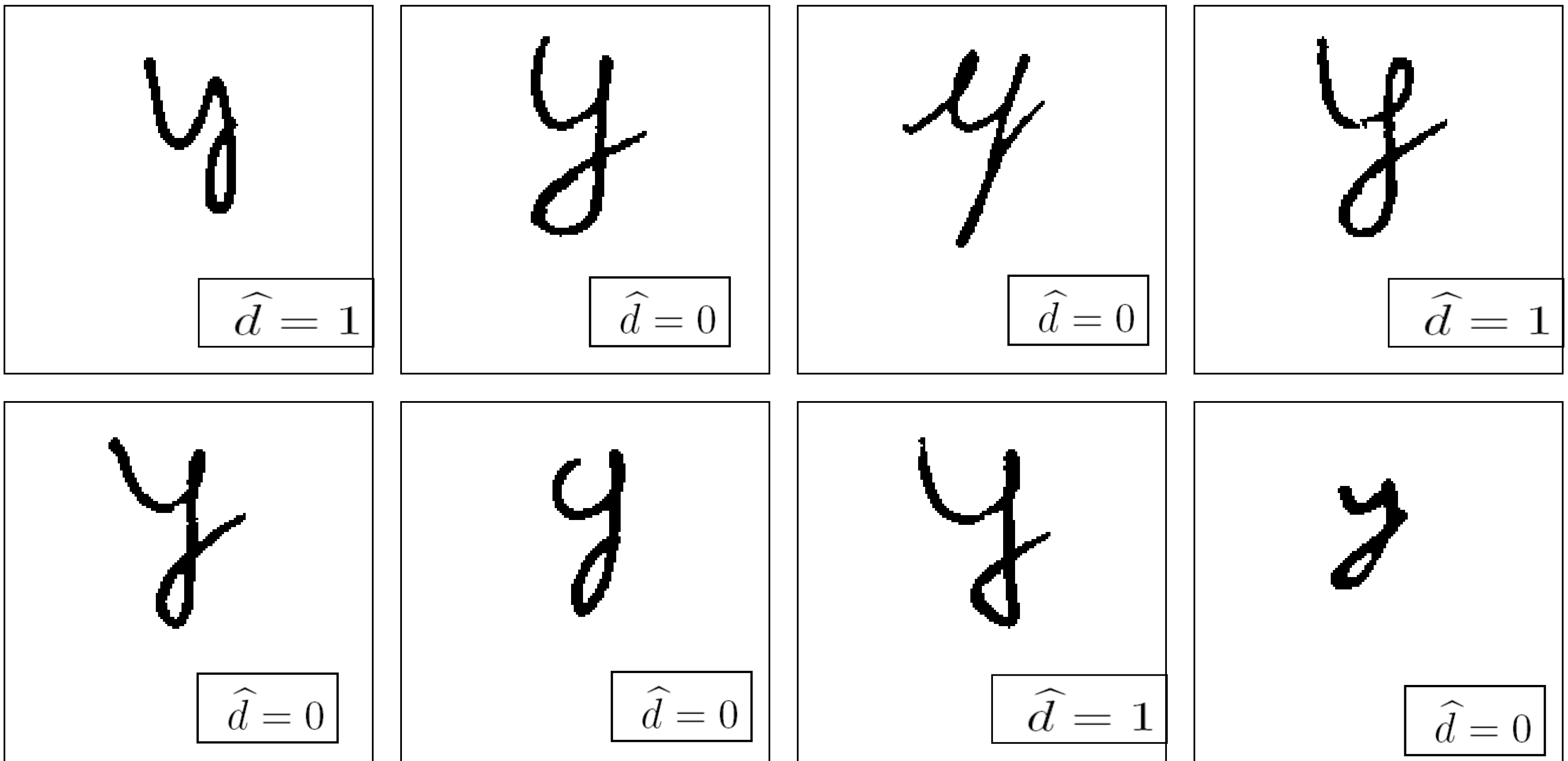


$d =$ $\widehat{d} =$

TEST

¿ $\widehat{d} = 0$ or $\widehat{d} = 1$?

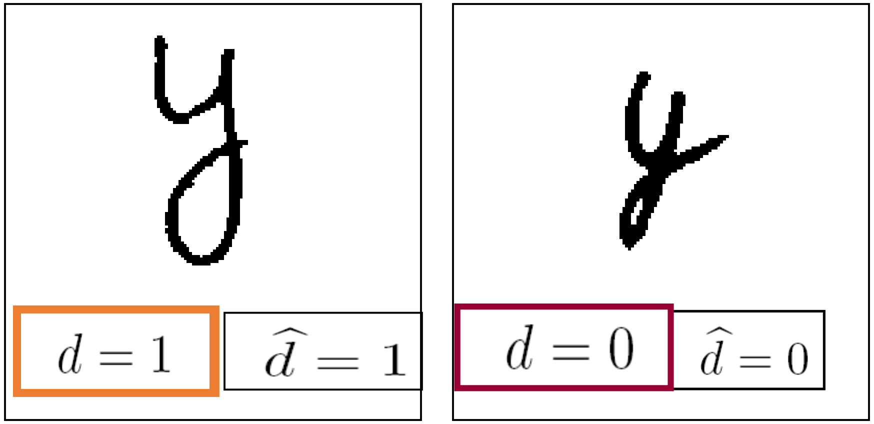


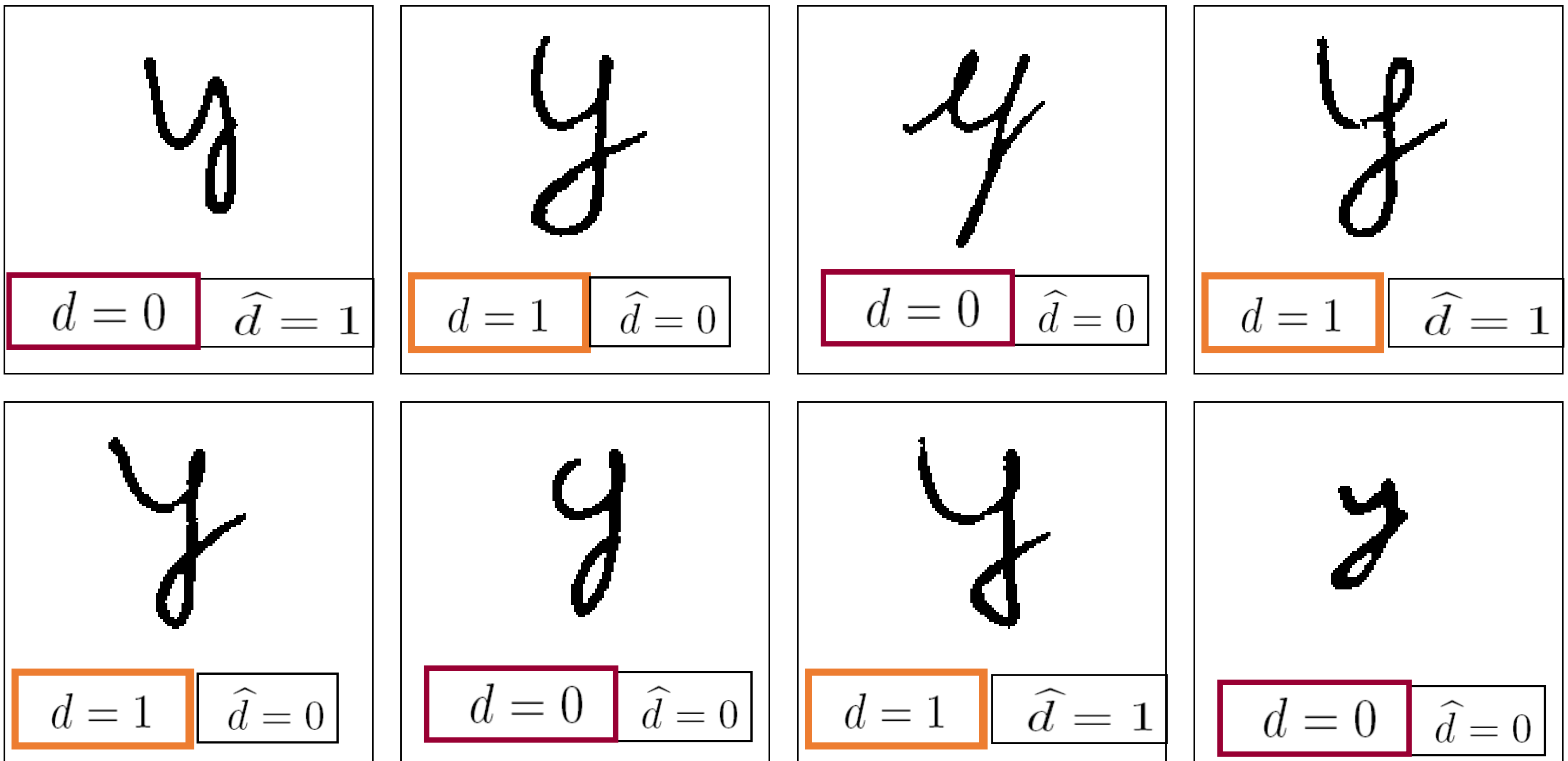


TEST

REGINNA 4

¿ $\widehat{d} = 0$ or $\widehat{d} = 1$?





universidad de león

Evaluation metrics

Error rate?

False Rejection Rate?

False Aceptance Rate?

SUPERVISED LEARNING MODELS

Supervised Learning Models

REGINNA 4.0

K-NN	Naïve Bayes	SVM
Decision Trees	Random Forest	Neural Networks

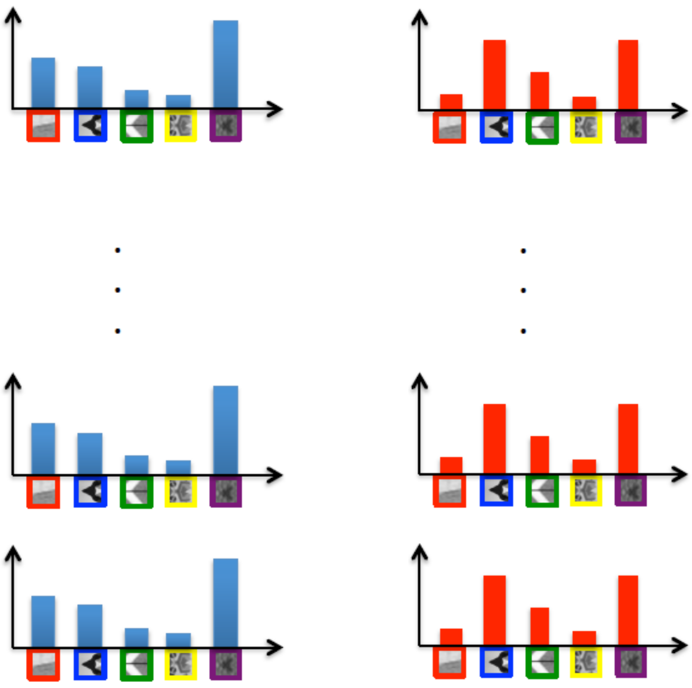
universidad de león

SUPERVISED LEARNING:
K-NEAREST NEIGHBOURS (K-NN)

K-nearest Neighbours (kNN)

kNN Assumes that the samples from the same class are **close** in the feature space

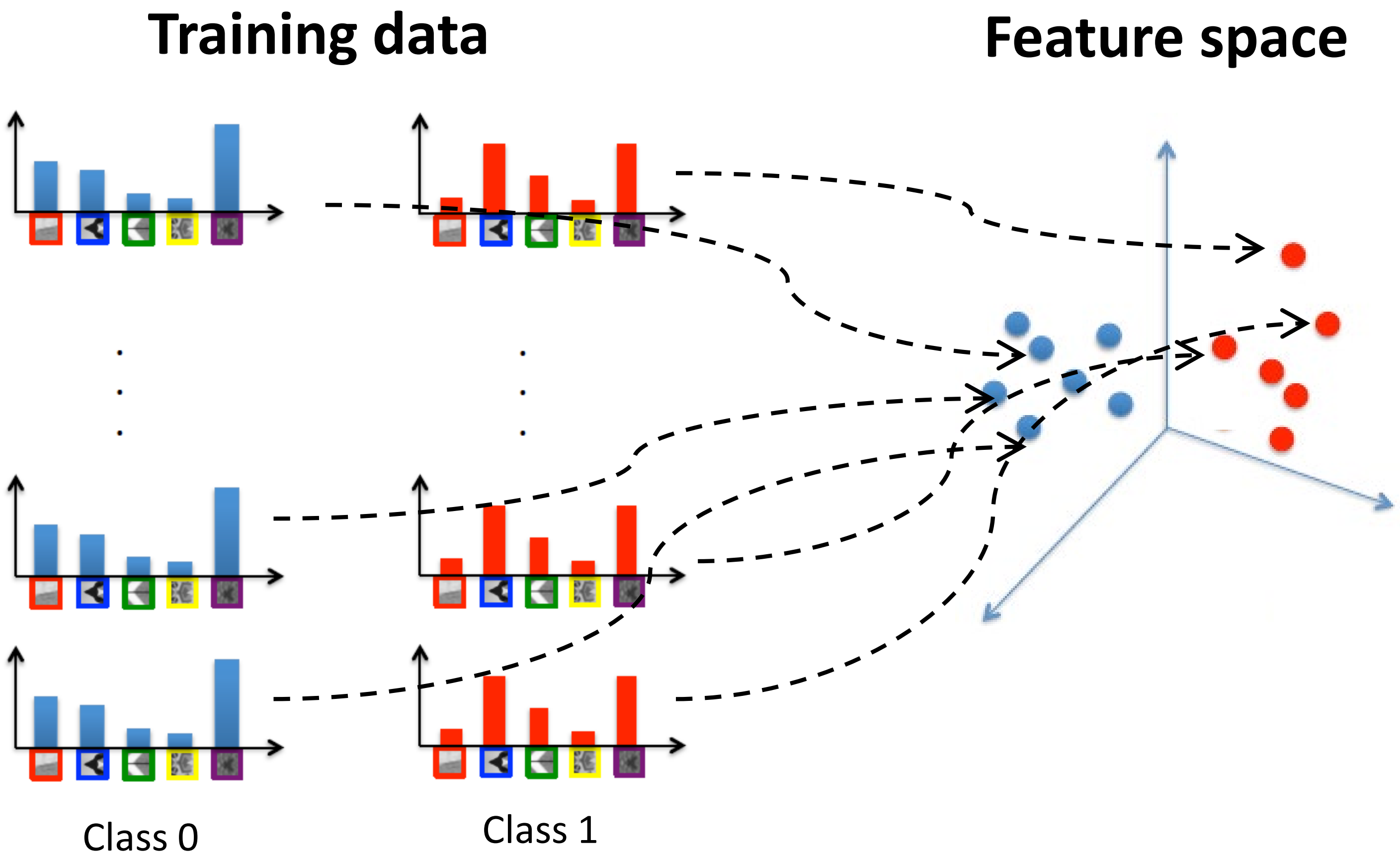
Use a set of **training data** to adjust the model



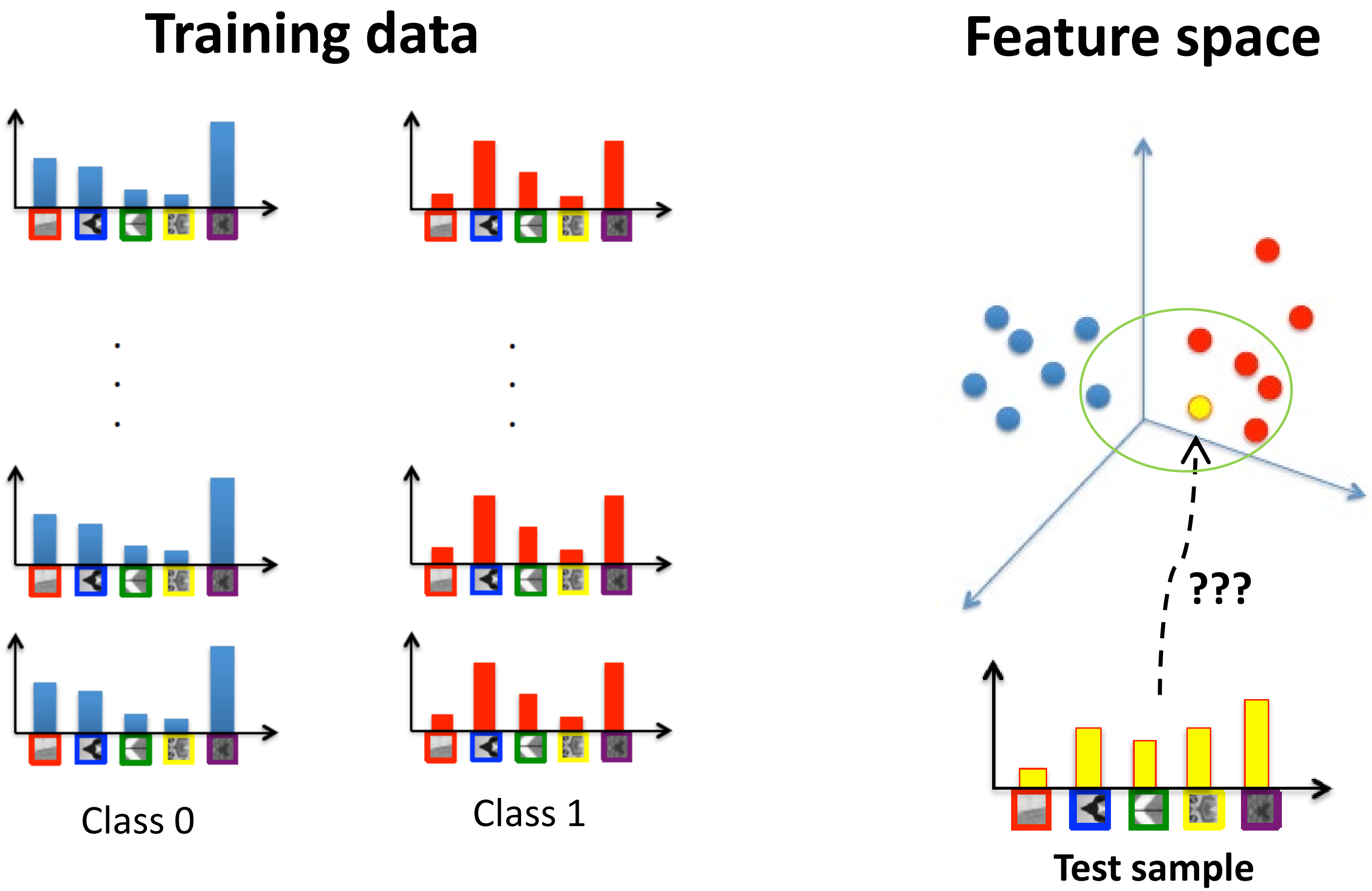
Class 0

Class 1

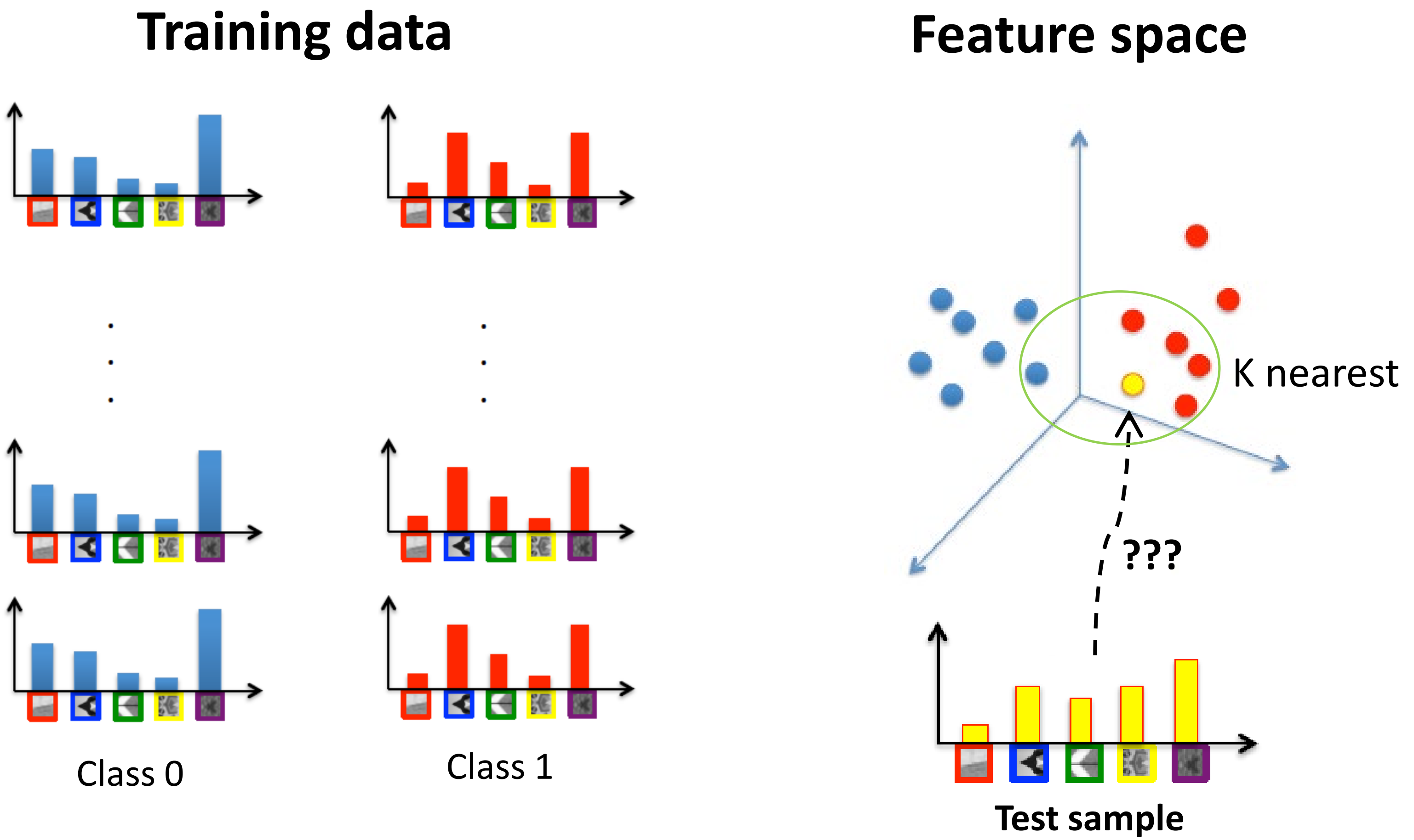
K-nearest Neighbours (kNN)



K-nearest Neighbours (kNN)

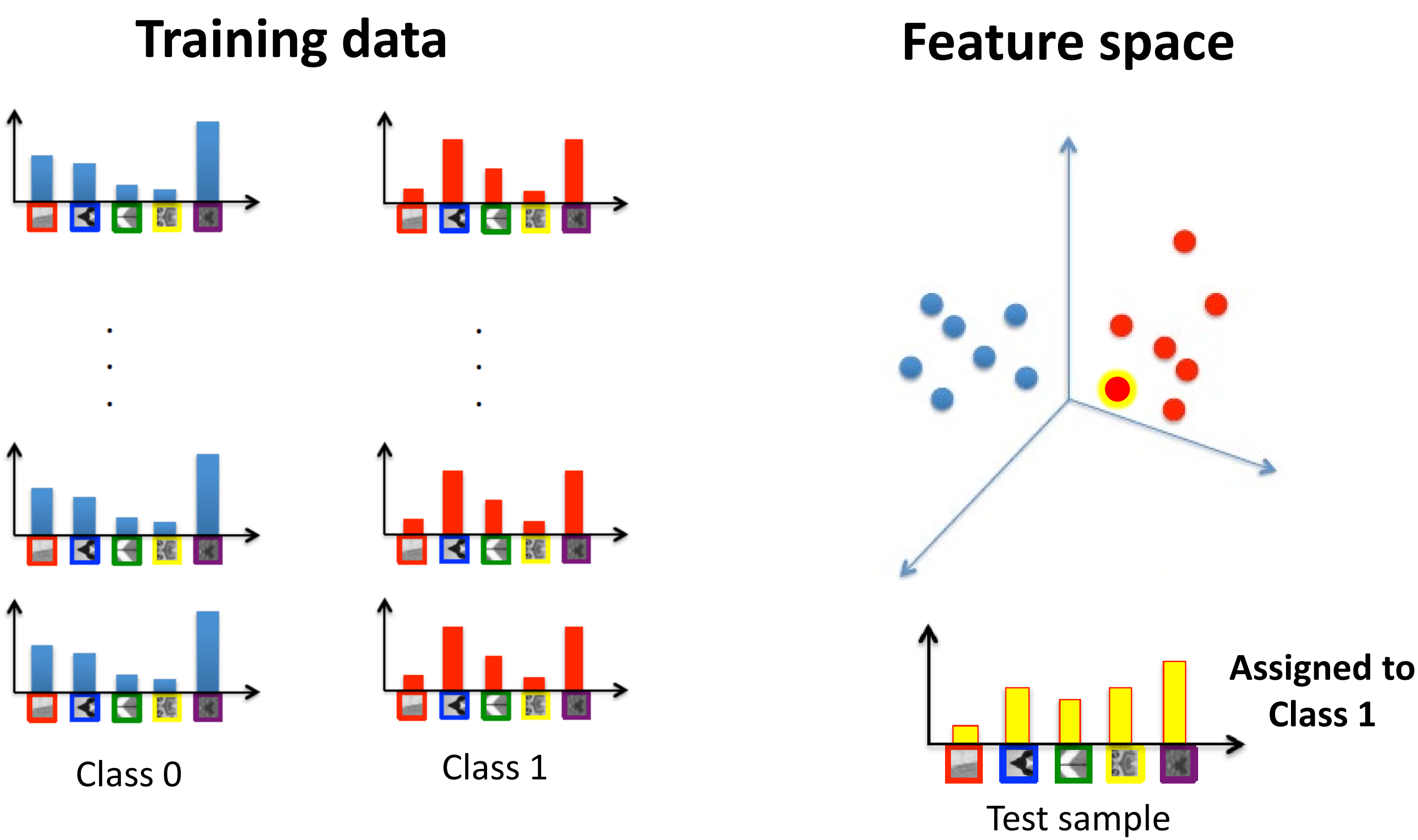


K-nearest Neighbours (kNN)



universidad de león

K-nearest Neighbours (kNN)



K-nearest Neighbours (kNN)

Euclidean distance: $$d(\mathbf{x}^{(r)}, \mathbf{x}^{(s)}) = \sqrt{\sum_{i=1}^{n} (x_i^{(r)} - x_i^{(s)})^2}$$

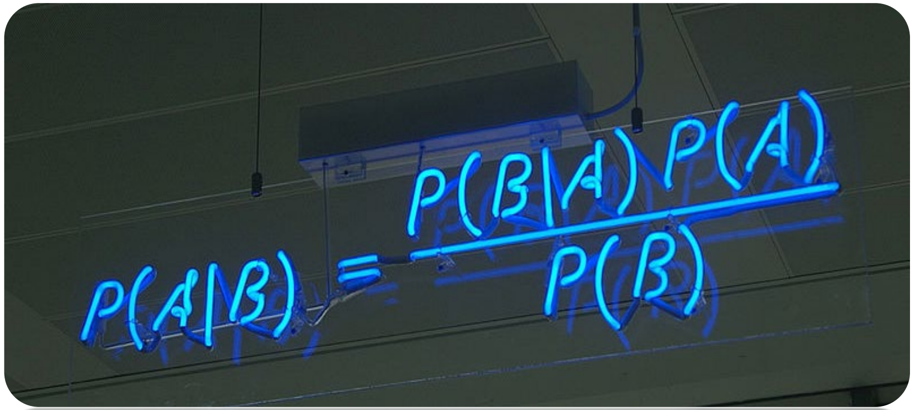
Manhattan distance: $$d(\mathbf{x}^{(r)}, \mathbf{x}^{(s)}) = \sum_{i=1}^{n} \left| x_i^{(r)} - x_i^{(s)} \right|$$

Chebyshev distance: $$d(\mathbf{x}^{(r)}, \mathbf{x}^{(s)}) = \max_{i=1,2,\ldots n} \left| x_i^{(r)} - x_i^{(s)} \right|$$

Cosine distance: $$d(\mathbf{x}^{(r)}, \mathbf{x}^{(s)}) = \arccos\left(\frac{\sum_{i=1}^{n} x_i^{(r)} x_i^{(s)}}{\sqrt{\sum_{i=1}^{n} x_i^{(r)}} \cdot \sqrt{\sum_{i=1}^{n} x_i^{(s)}}} \right)$$

SUPERVISED LEARNING: NAIVE BAYES

Naive Bayes Classifier

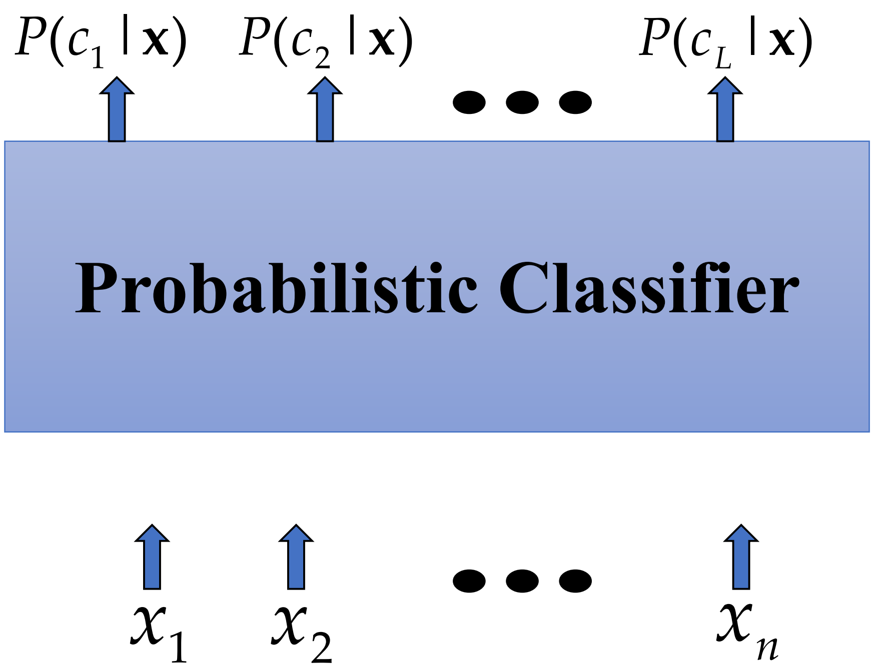


universidad de león

Probabilistic Classification

- Establishing a probabilistic model for classification

$$P(c|\mathbf{x}) \quad c = c_1, \cdots, c_L, \qquad \mathbf{x} = (x_1, \cdots, x_n)$$



$$\mathbf{x} = (x_1, x_2, \cdots, x_n)$$

Probability Basics

REGINNA 4.0

Prior, conditional and joint probability for random variables

- Prior probability: $P(c)$
- Conditional probability: $P(x_1 | x_2), P(x_2 | x_1)$
- Joint probability: $\mathbf{x} = (x_1, x_2), P(\mathbf{x}) = P(x_1, x_2)$
- Relationship: $P(x_1, x_2) = P(x_2 | x_1)P(x_1) = P(x_1 | x_2)P(x_2)$
- Independence: $P(x_2 | x_1) = P(x_2)$ $\quad$ $P(x_1, x_2) = P(x_1)P(x_2)$

  $P(x_1 | x_2) = P(x_1)$

Bayesian Rule

$$P(c | \mathbf{x}) = \frac{P(\mathbf{x} | c)P(c)}{P(\mathbf{x})}$$

$$Posterior = \frac{Likelihood \times Prior}{Evidence}$$

universidad de león

Probabilistic Classification

- **M**aximum **A P**osterior (**MAP**) classification rule
  - For an input ***x***, find the largest one from L probabilities output by a probabilistic classifier $P(c_1 \mid \mathbf{x}), \ldots, P(c_L \mid \mathbf{x})$.
  - Assign ***x*** to label *c** if $P(c^* \mid \mathbf{x})$ is the largest.

- Classification with the MAP rule
  - Apply Bayesian rule to get posterior probabilities

$$P(c_i \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid c_i)P(c_i)}{P(\mathbf{x})} \propto P(\mathbf{x} \mid c_i)P(c_i)$$

Common factor for all L probabilities

$$\text{for } i = 1, 2, \cdots, L$$

  - Then apply the MAP rule to assign a label

Naive Bayes

Bayes classification

$$P(c\,|\,\mathbf{x}) \propto P(\mathbf{x}\,|\,c)P(c) = P(x_1,\cdots,x_n\,|\,c)P(c) \ \text{ for } c = c_1,\ldots,c_L.$$

Difficulty: learning the joint probability $P(x_1,\cdots,x_n\,|\,c)$

Naïve Bayes classification

– Assume all input features are class conditionally independent!

$$\begin{aligned} P(x_1,x_2,\cdots,x_n\,|\,c) &= \underline{P(x_1\,|\,x_2,\cdots,x_n,c)}P(x_2,\cdots,x_n\,|\,c) \\ &= \underline{P(x_1\,|\,c)}P(x_2,\cdots,x_n\,|\,c) \\ &= P(x_1\,|\,c)P(x_2\,|\,c)\cdots P(x_n\,|\,c) \end{aligned}$$

Applying the independence assumption

– Apply the MAP classification rule: assign $\mathbf{x}' = (a_1,a_2,\cdots,a_n)$ to c^* if

$$\underbrace{[P(a_1\,|\,c^*)\cdots P(a_n\,|\,c^*)]}_{\text{estimate of } P(a_1,\cdots,a_n\,|\,c^*)}P(c^*) > \underbrace{[P(a_1\,|\,c)\cdots P(a_n\,|\,c)]}_{\text{esitmate of } P(a_1,\cdots,a_n\,|\,c)}P(c), \quad c \neq c^*, c = c_1,\cdots,c_L$$

Naive Bayes Algorithm (for discrete input attributes)

REGINNA 4.0

Learning Phase: Given a training set **S**, with L classes and n features

For each target value of c_i $(c_i = c_1, \cdots, c_L)$

$\hat{P}(c_i) \leftarrow$ estimate $P(c_i)$ with examples in $\mathbf{S}$;

For every attribute value x_{jk} of each attribute x_j $(j = 1, \cdots, n;\ k = 1, \cdots, N_j)$

$\hat{P}(x_{jk} \mid c_i) \leftarrow$ estimate $P(x_{jk} \mid c_i)$ with examples in $\mathbf{S}$;

Output: conditional probability tables; for x_j, $N_j \times L$ elements

Test Phase: Given an unknown instance $\mathbf{x}' = (a'_1, \cdots, a'_n)$

Look up tables to assign the label c^* to **X'** if

$$[\hat{P}(a'_1 \mid c^*) \cdots \hat{P}(a'_n \mid c^*)]\hat{P}(c^*) > [\hat{P}(a'_1 \mid c) \cdots \hat{P}(a'_n \mid c)]\hat{P}(c), \quad c \neq c^*, c = c_1, \cdots, c_L$$

universidad de león

Example NB: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Example NB : Play Tennis

Learning Phase: Given a training set **S**, with L classes and n features

For each target value of c_i $(c_i = c_1, \cdots, c_L)$

$\hat{P}(c_i) \leftarrow$ estimate $P(c_i)$ with examples in **S**;

For every attribute value x_{jk} of each attribute x_j $(j = 1, \cdots, n; k = 1, \cdots, N_j)$

$\hat{P}(x_{jk} \mid c_i) \leftarrow$ estimate $P(x_{jk} \mid c_i)$ with examples in **S**;

Output: conditional probability tables; for x_j, $N_j \times L$ elements

L= ?

n= ?

Example: Play Tennis

Learning Phase: Given a training set **S**, with L classes and n features

For each target value of c_i $(c_i = c_1, \cdots, c_L)$

$\boxed{\hat{P}(c_i) \leftarrow \text{estimate } P(c_i) \text{ with examples in } \mathbf{S};}$

For every attribute value x_{jk} of each attribute x_j $(j = 1, \cdots, n; k = 1, \cdots, N_j)$

$\hat{P}(x_{jk} \mid c_i) \leftarrow \text{estimate } P(x_{jk} \mid c_i) \text{ with examples in } \mathbf{S};$

Output: conditional probability tables; for x_j, $N_j \times L$ elements

$$P(\text{Play}=\textit{Yes}) = 9/14 \qquad P(\text{Play}=\textit{No}) = 5/14$$

Example: Play Tennis

Learning Phase: Given a training set **S**, with L classes and n features

For each target value of c_i $(c_i = c_1, \cdots, c_L)$

$\hat{P}(c_i) \leftarrow$ estimate $P(c_i)$ with examples in $\mathbf{S}$;

For every attribute value x_{jk} of each attribute x_j $(j = 1, \cdots, n; k = 1, \cdots, N_j)$

$\hat{P}(x_{jk} \mid c_i) \leftarrow$ estimate $P(x_{jk} \mid c_i)$ with examples in $\mathbf{S}$;

Output: conditional probability tables; for x_j, $N_j \times L$ elements

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Example: Play Tennis

Learning Phase:

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=*Yes*	Play=*No*
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

$P(\text{Play}=\mathit{Yes}) = 9/14$ $\quad$ $P(\text{Play}=\mathit{No}) = 5/14$

Example: Play Tennis

Test Phase: Given a new instance, predict its label

- $\mathbf{x}'$=(Outlook=***Sunny***, Temperature=***Cool***, Humidity=***High***, Wind=***Strong***)
- Look up tables achieved in the learning phrase

P(Outlook=*Sunny*|Play=*Yes*) = 2/9

P(Temperature=*Cool*|Play=*Yes*) = 3/9

P(Huminity=*High*|Play=*Yes*) = 3/9

P(Wind=*Strong*|Play=*Yes*) = 3/9

P(Play=*Yes*) = 9/14

P(Outlook=*Sunny*|Play=*No*) = 3/5

P(Temperature=*Cool*|Play==*No*) = 1/5

P(Huminity=*High*|Play=*No*) = 4/5

P(Wind=*Strong*|Play=*No*) = 3/5

P(Play=*No*) = 5/14

- Decision making with the MAP rule

$P(Yes|\mathbf{x}') \approx [P(Sunny|Yes)P(Cool|Yes)P(High|Yes)P(Strong|Yes)]P(Play=Yes) = 0.0053$

$P(No|\mathbf{x}') \approx [P(Sunny|No)\,P(Cool|No)P(High|No)P(Strong|No)]P(Play=No) = 0.0206$

Given the fact $P(Yes|\mathbf{x}') < P(No|\mathbf{x}')$, we label $\mathbf{x}'$ to be "*No*".

REGINNA 4.0

Pros and cons of Naive Bayes

Advantages

- It's relatively simple to understand and build
- It's easily trained, even with a small dataset
- It's fast!
- It's not sensitive to irrelevant features
- Test is straightforward; just looking up tables

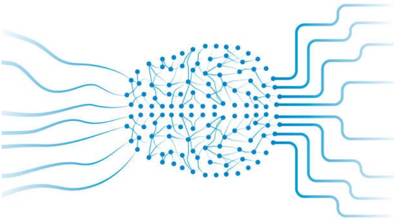
Disadvantages

- It assumes every feature is independent, which isn't always the case

SUPERVISED LEARNING: NEURAL NETWORKS

Artificial Neural Networks

Inspired by the biological processes: Nervous system



- Simplified mathematical models
- Try to mimic neurons in a very basic setting
- Without claiming to faithfully reflect the real behaviour of the nervous system

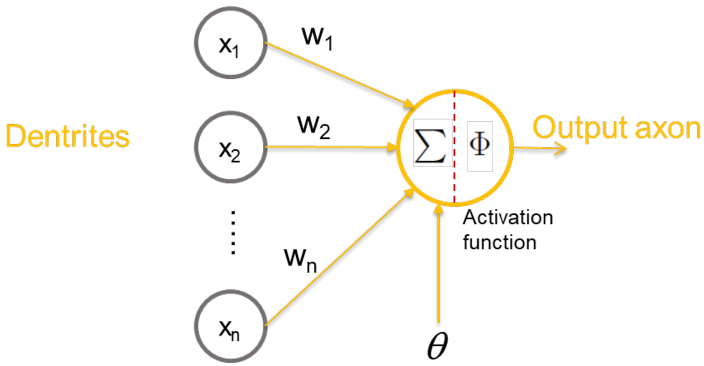
- Capable of handling uncertainty
- Robust Solutions
- Not an algorithm
- Non- linear models

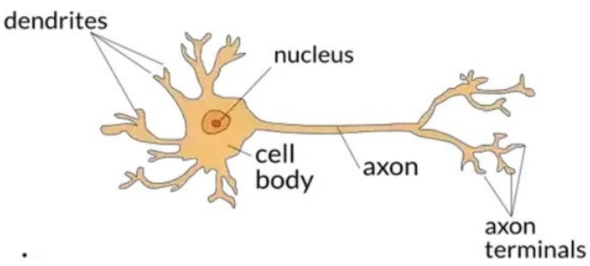
Biological neuron

S. Ramón y Cajal → Neuron discovery

1906 Nobel Prize in Medicine

Artificial Neuron vs. Biological Neuron

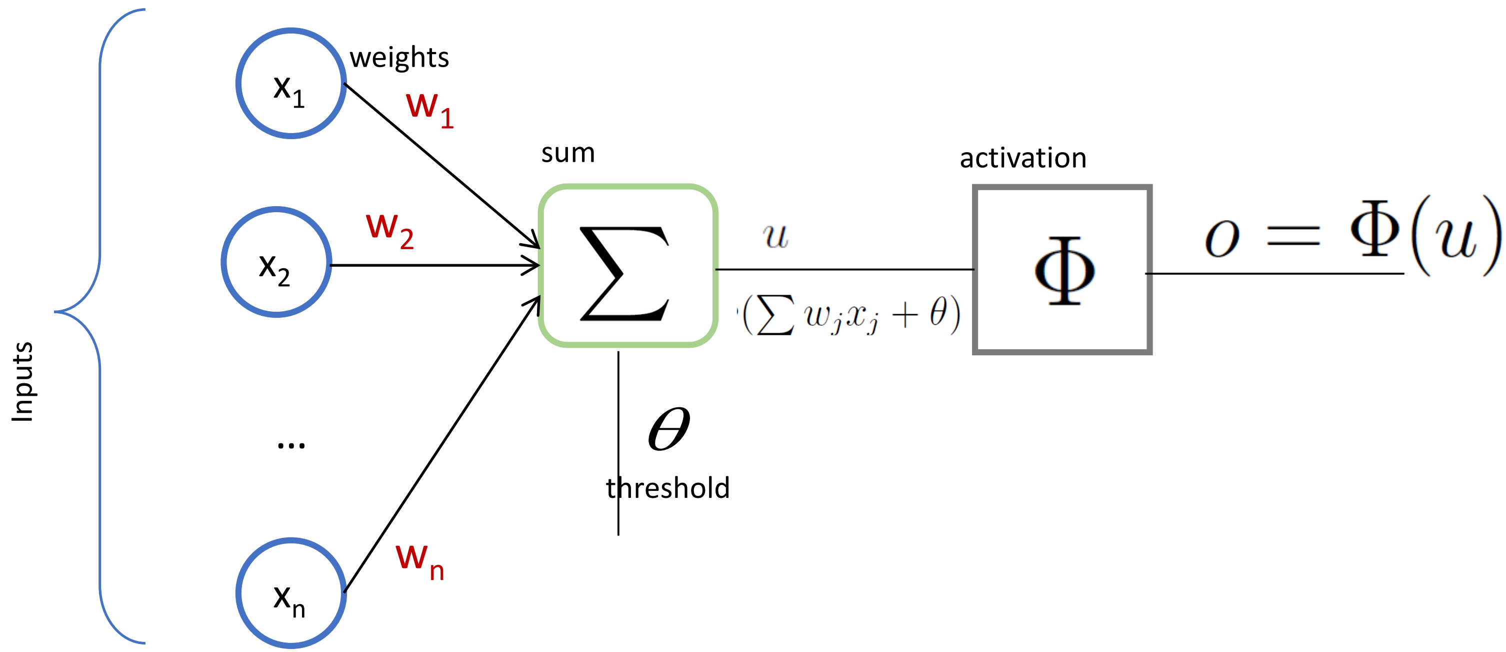




universidad de león

Perceptron (by Rosenblatt)

Architecture



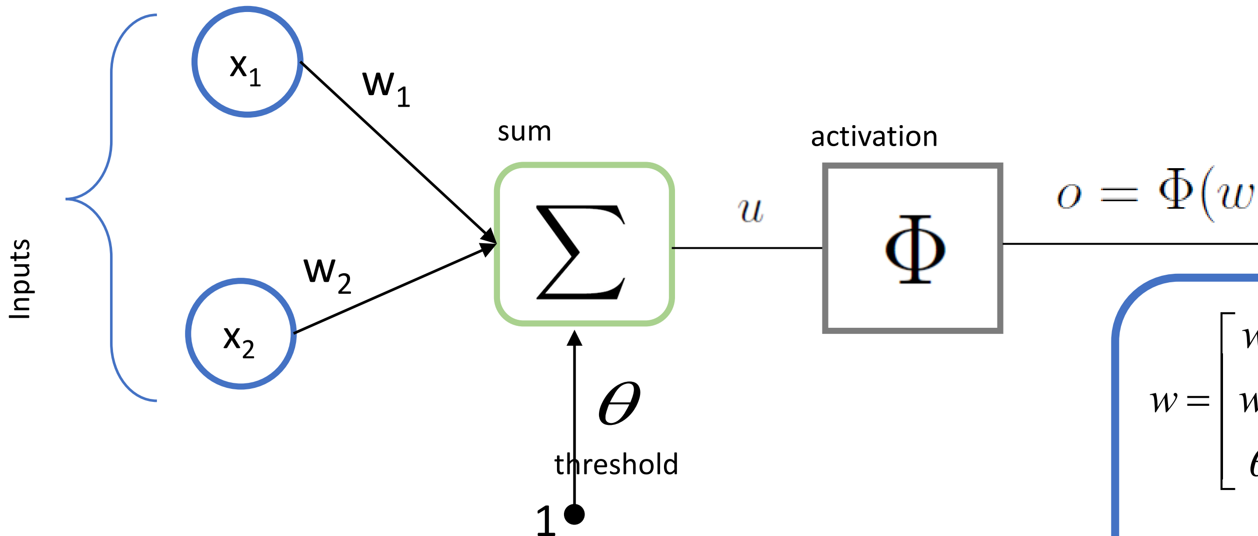
$$o = \Phi(u) = \Phi(\sum_j w_j x_j + \theta)$$

$$\Phi(u) = \operatorname{sgn}(u) = \begin{cases} 1, & u \geq 0 \\ -1, & u < 0 \end{cases}$$

universidad de león

Perceptron (by Rosenblatt)

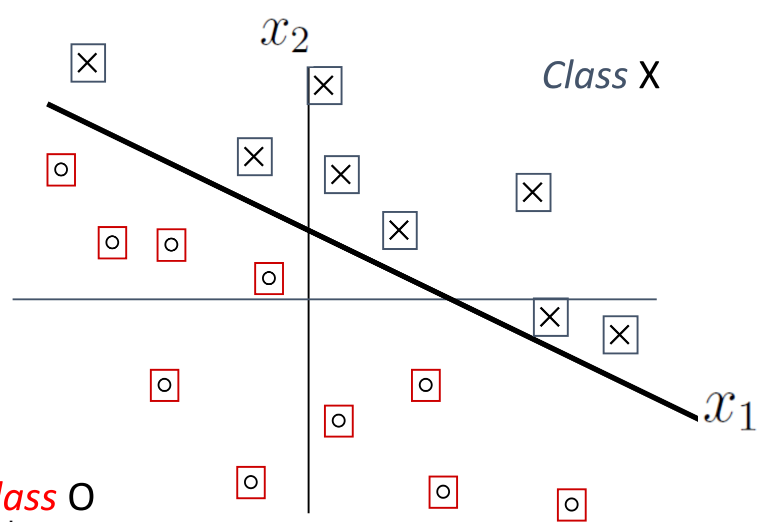
Architecture



$$o = \Phi(w_1 x_1 + w_2 x_2 + \theta)$$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \theta \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix}$$

$$o = \Phi(\mathbf{w}^\top \mathbf{x})$$



Class O

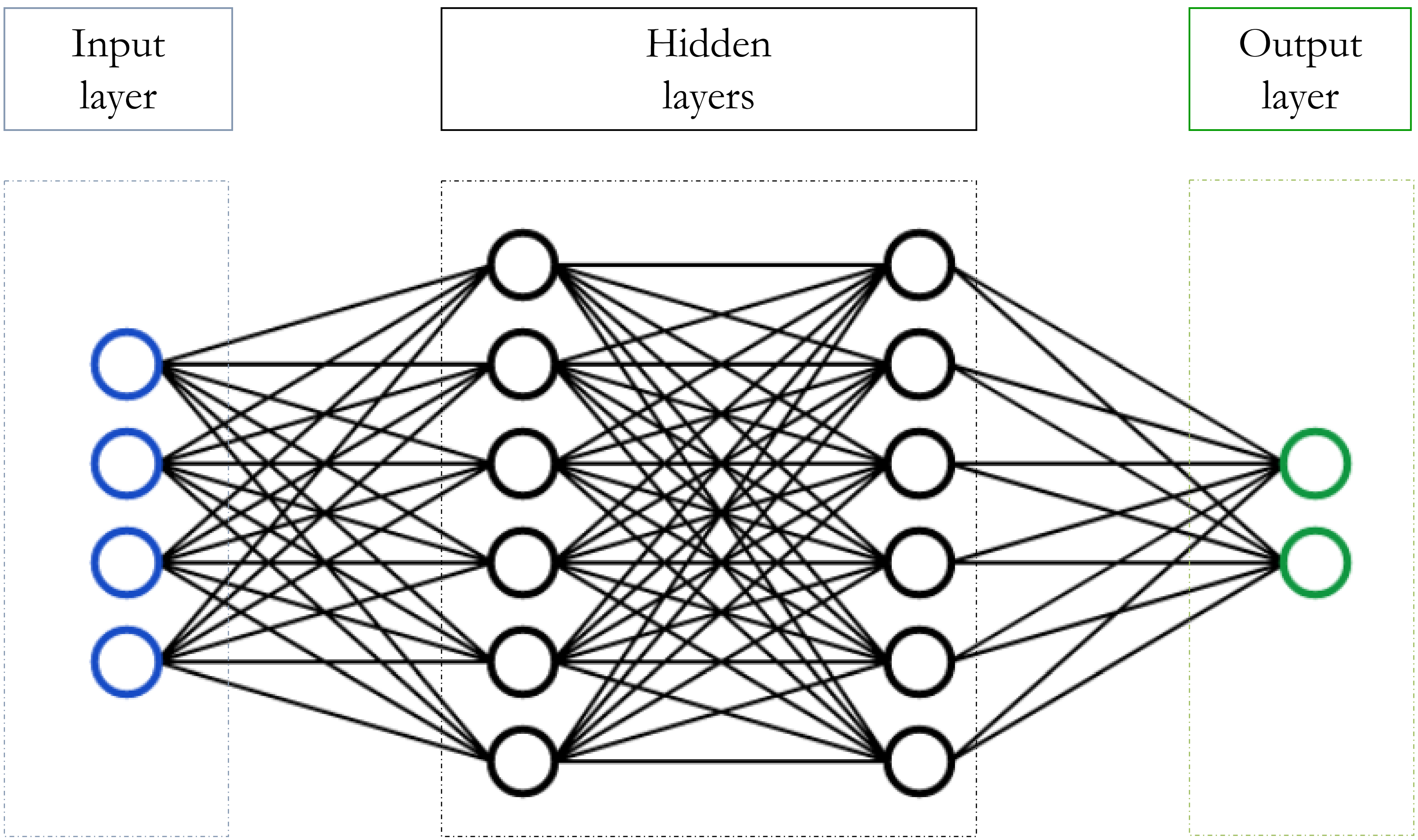
Decision boundary

$$(\mathbf{w}^\top \mathbf{x}) = 0$$

$$w_1 x_1 + w_2 x_2 + \theta = 0$$

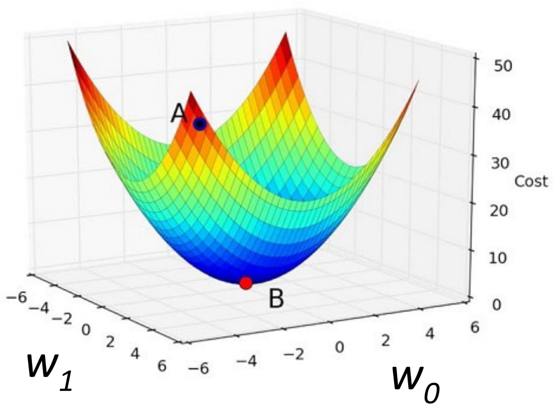
MultiLayer Perceptron(MLP)

Architecture



… Training Neural Networks

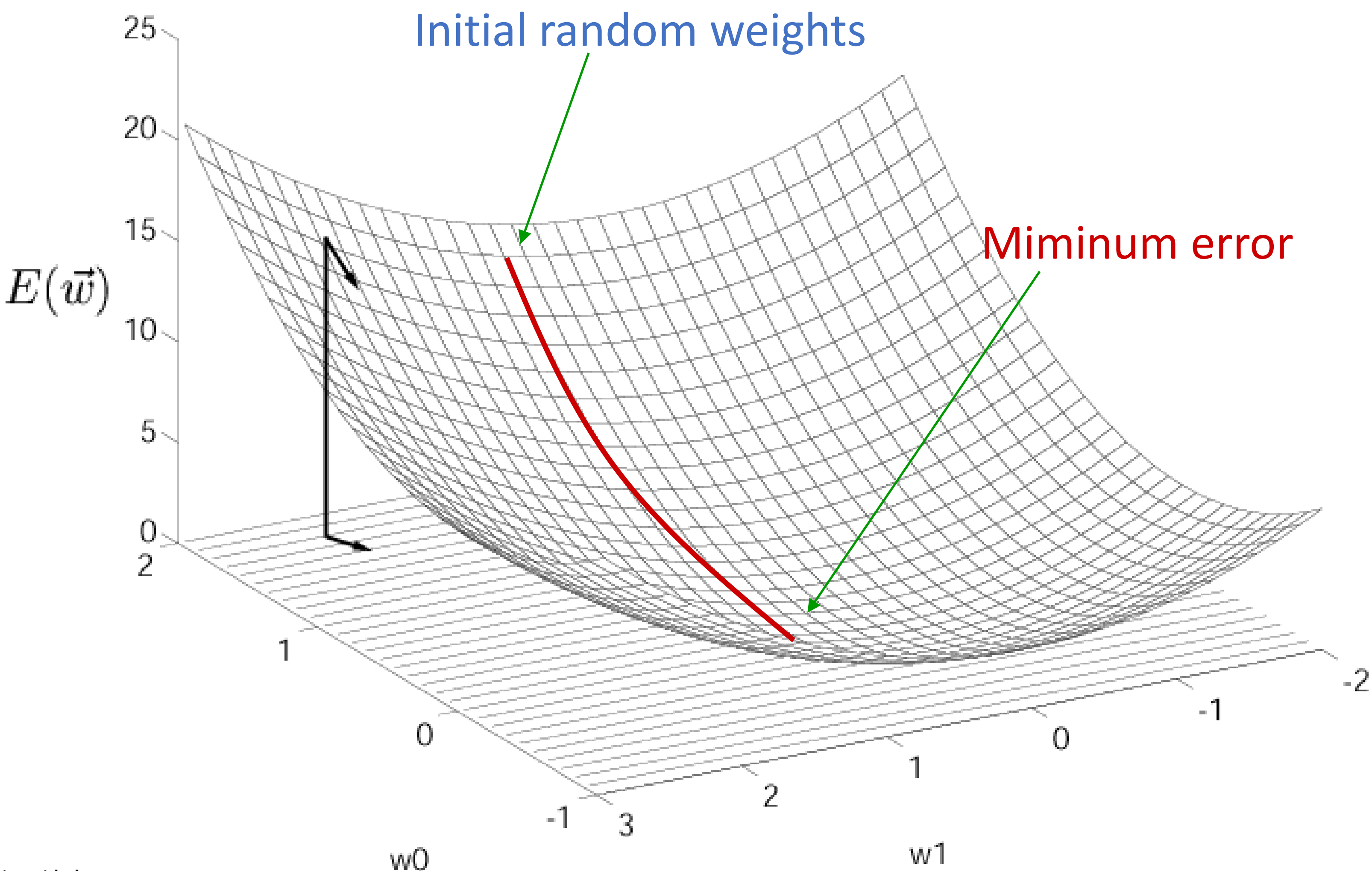
Cost function



- The key idea is to use gradient descent to search the hypothesis space of possible weights to find the network weights that best fit the training examples.

... Training Neural Network

Cost function



Building Machine Learning Models

Pipeline

I-Problem Definition and Data Collection → II-Data Preparation → III-Model Selection → IV-Model training → V- Model Fine-Tuning → VI - Model Testing → VII Deployment

I-Problem Definition and Data Collection → II-Data Preparation → III-Model Selection → IV-Model training → V- Model Fine-Tuning → VI - Model Testing → VII Deployment

I-Problem Definition and Data Collection

Clearly define the problem statement and objective of the machine learning model.

Identify the type of machine learning task (classification, regression, clustering, etc.).

Collect and preprocess the relevant data required for training and evaluation.

II-Data Exploration and Preparation

- Perform exploratory data analysis (EDA) to gain insights into the dataset.
- Handle missing values, outliers, and inconsistencies in the data.
- Feature selection, feature extraction.
- Split the data into training, validation, and test sets.

I-Problem Definition and Data Collection → II-Data Preparation → III-Model Selection → IV-Model training → V- Model Fine-Tuning → VI - Model Testing → VII Deployment

III-Model Selection

Understand different types of machine learning algorithms (decision trees, neural networks, etc.).

Consider factors such as model complexity, interpretability, and scalability.

Choose an appropriate model based on the problem requirements and dataset characteristics.

universidad de león

I-Problem Definition and Data Collection → II-Data Preparation → III-Model Selection → IV-Model training → V- Model Fine-Tuning → VI - Model Testing → VII Deployment

IV-Model training

- Prepare the data for model training (normalization, encoding categorical variables, etc.).
- Optimize the model parameters using a suitable optimization algorithm (e.g., gradient descent).
- Evaluate the model's performance on the validation set to monitor its progress.

REGINNA 4.0

I-Problem Definition and Data Collection → II-Data Preparation → III-Model Selection → IV-Model training → V- Model Fine-Tuning → VI - Model Testing → VII Deployment

V-Model Fine-Tuning

Adjust hyperparameters (e.g., learning rate, regularization) to improve model performance.

Measure the model's performance using appropriate evaluation metrics (accuracy, precision, recall, etc.).

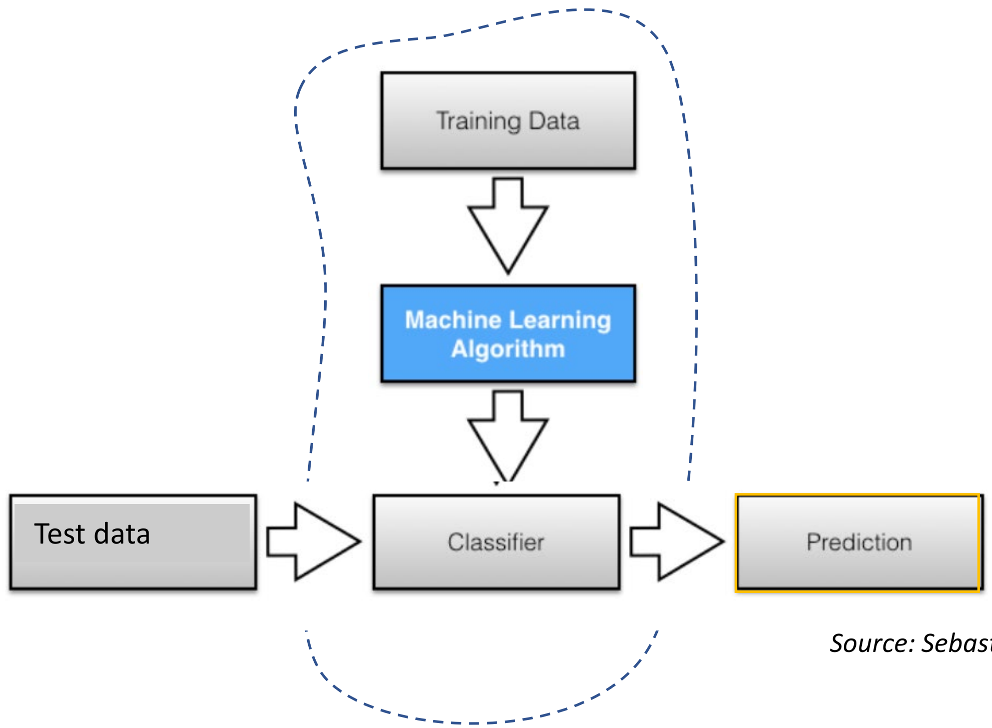
Use techniques like cross-validation or grid search to find the optimal hyperparameter values.

Analyze the model's strengths, weaknesses, and potential sources of error.

universidad de león

I-Problem Definition and Data Collection → II-Data Preparation → III-Model Selection → IV-Model training → V- Model Fine-Tuning → VI - Model Testing → VII Deployment

VI-Model testing



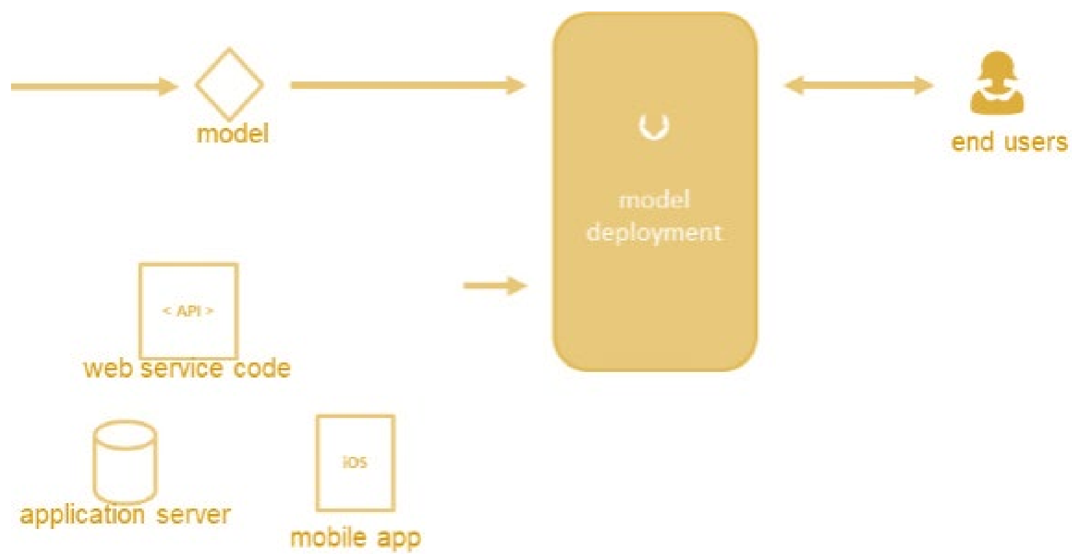
Source: Sebastian Raschka

- Assess the final model's performance on the test set.

I-Problem Definition and Data Collection → II-Data Preparation → III-Model Selection → IV-Model training → V- Model Fine-Tuning → VI - Model Testing → VII Deployment

VII-Model deployment

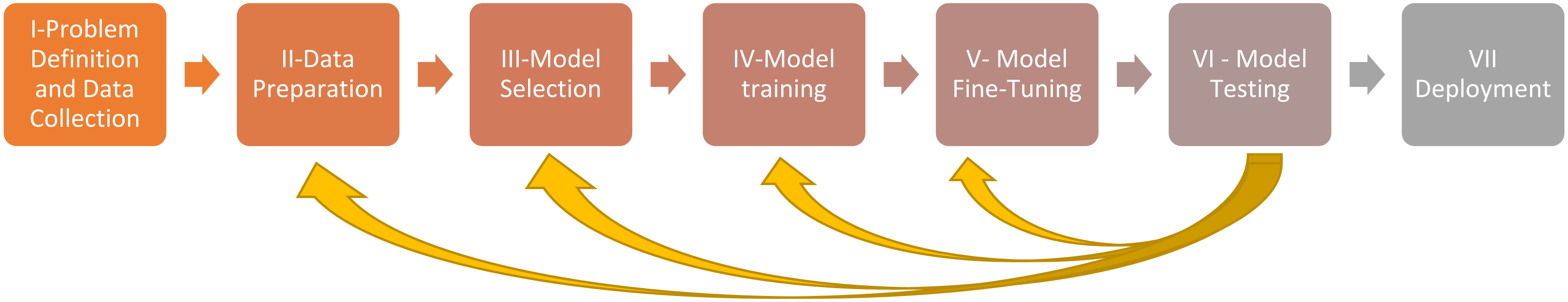
- Implement the model in a production environment.
- Monitor the model's performance in production and iterate on improvements as needed.



universidad de león

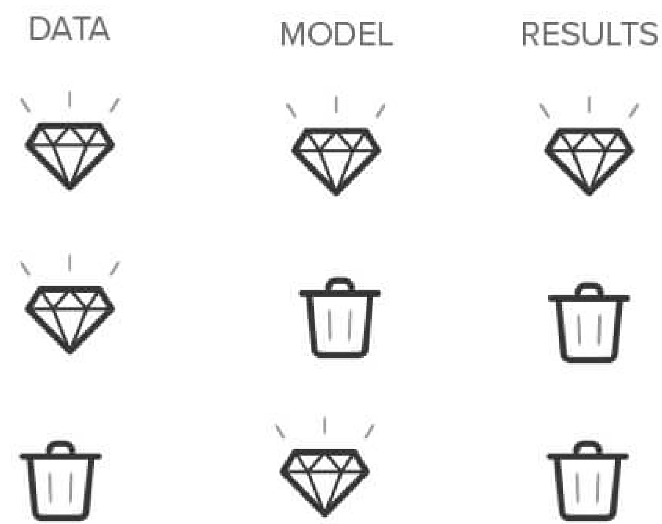
Machine Learning — An iterative process

The development of Machine Learning-based applications is a highly empirical process.



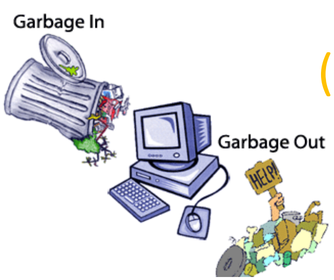
Remarks: Data quality

The quality of the results strongly depends on the quality of the training data.



GIGO

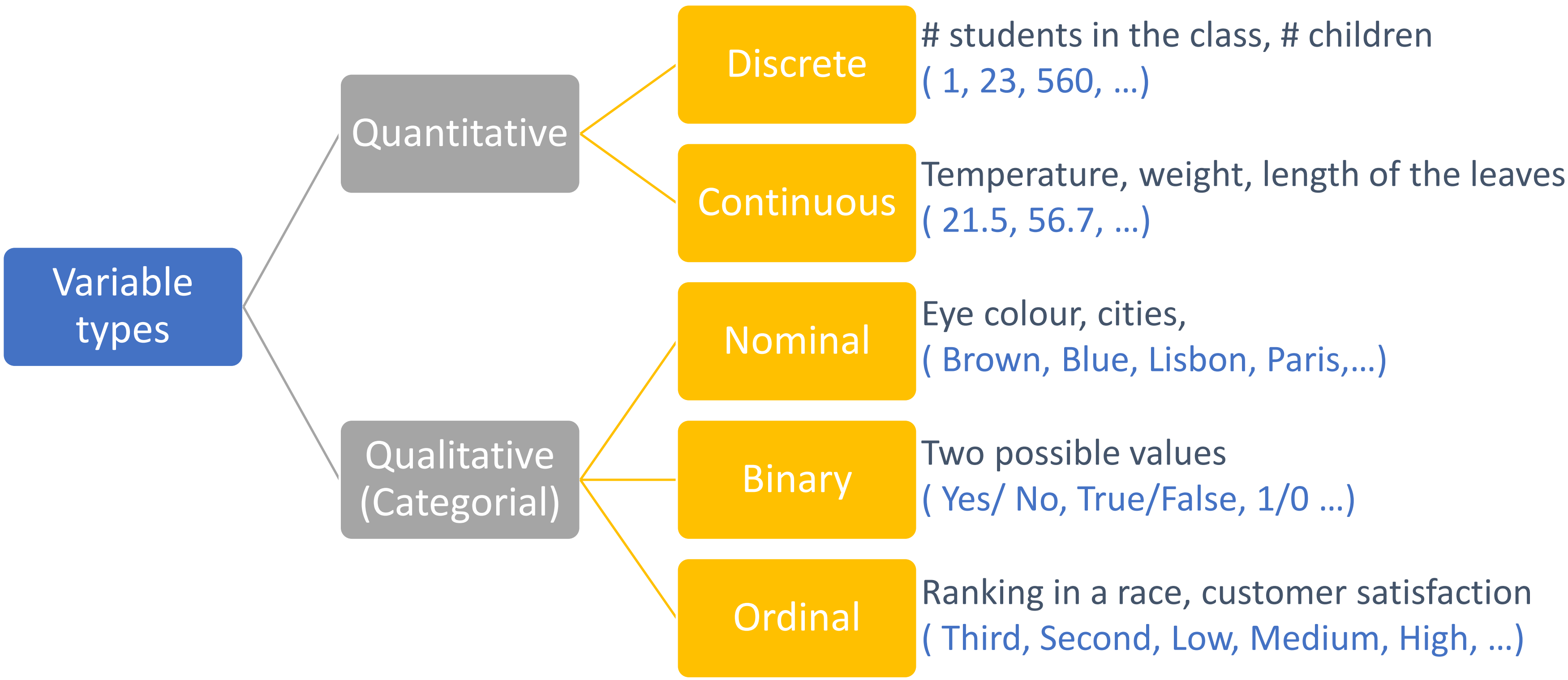
(Garbage In, Garbage Out)



universidad de león

Remarks: Variable (Feature, Attribute) Types

- Objective function: It is the true function f that we aim to learn.



MODEL EVALUATION

How can we evaluate model?

What if I use the training set to assess the quality of the model?

In that case, we would be rewarding models that MEMORIZE TRAINING DATA

PROBLEM!!!! We want our classifier to generalize well for new (unknown) cases.

SOLUTION Evaluate the models on a dataset different from the training dataset.

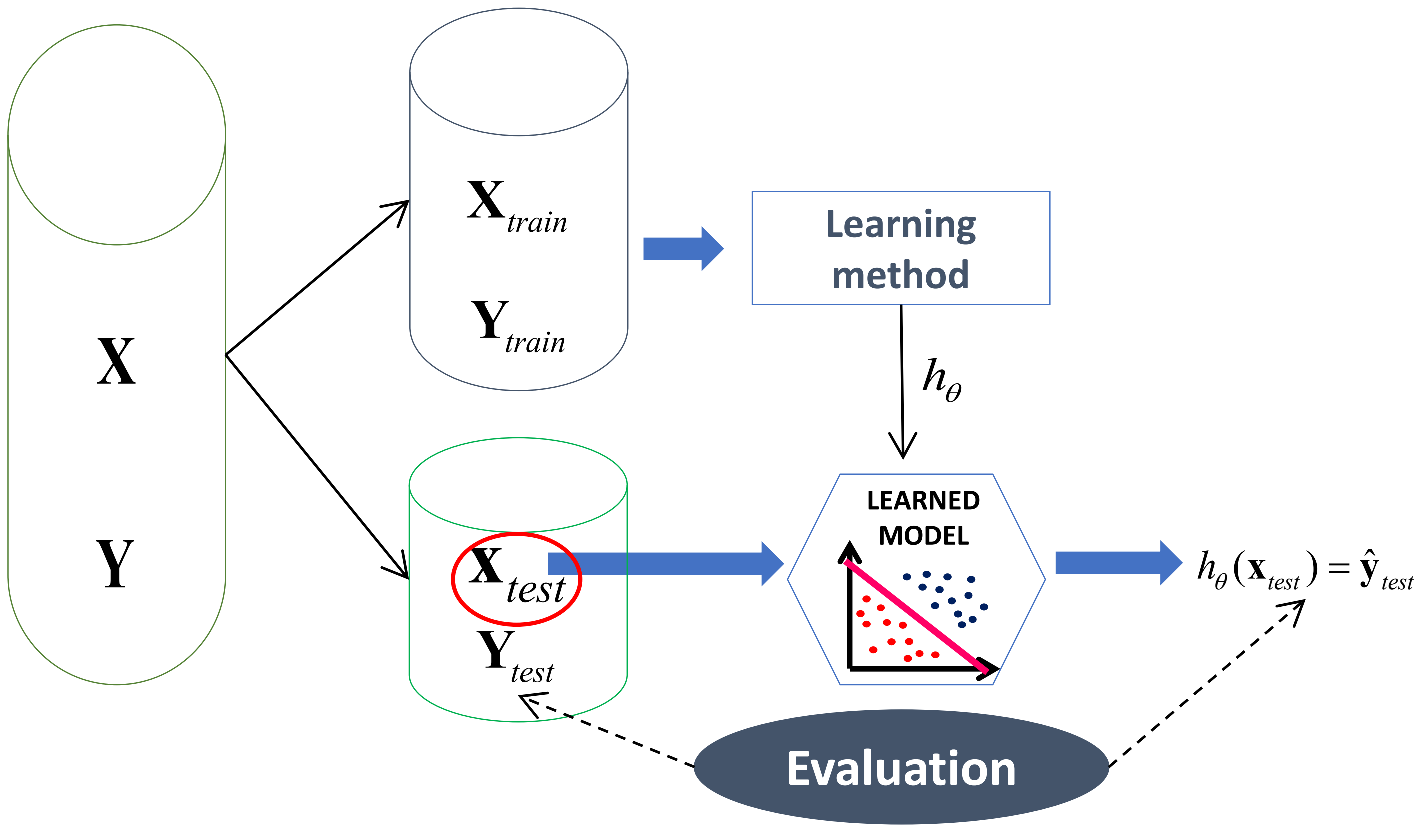
How can we get an **unbiased estimate** of the **accuracy of a learned model**?

universidad de león

Evaluation techniques

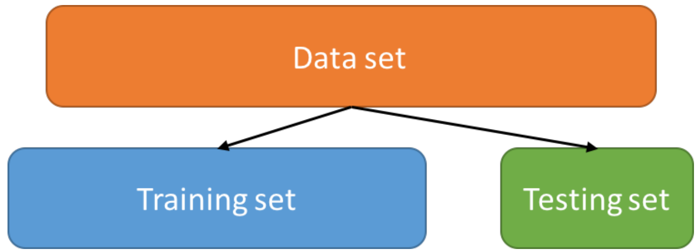
Train-Test split

Splitting the dataset into train and test set



Evaluation techniques

Train-Test split



The available data set D is divided into two disjoint subsets,

- the **training set** Xtrain (for learning a model) (70%)
- the **test set** Xtest (for testing the model) (30%)

This method is mainly used when the data set X is large.

IMPORTANT!!!!

The training set should not be used in testing and the test set should not be used in learning.

Unseen test set provides a unbiased estimate of accuracy.

universidad de león

Evaluation metrics

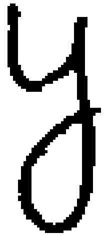
Confusion matrix

It is used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are known.

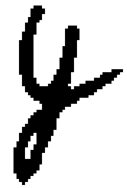
		Real Class	
		1	0
Predicted Class	1	TP	FP
	0	FN	TN

TEST

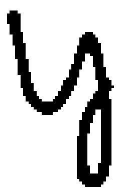
¿ $\widehat{d} = 0$ o $\widehat{d} = 1$?



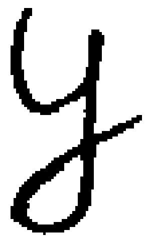
$d = 1$ $\widehat{d} = 1$



$d = 0$ $\widehat{d} = 0$

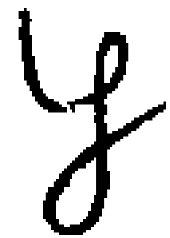


$d = 0$ $\widehat{d} = 1$

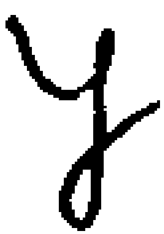


$d = 1$ $\widehat{d} = 0$

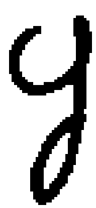
$d = 0$ $\widehat{d} = 0$



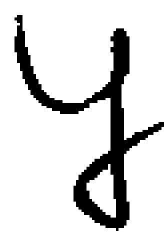
$d = 1$ $\widehat{d} = 1$



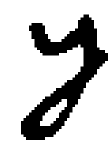
$d = 1$ $\widehat{d} = 0$



$d = 0$ $\widehat{d} = 0$



$d = 1$ $\widehat{d} = 1$



$d = 0$ $\widehat{d} = 0$

Evaluation metrics

Confusion matrix

		Real Class	
		1	0
Predicted Class	1	TP	FP
	0	FN	TN

FN, FP, TN, TP?

When someone performs a test to check if s/he is an authorized user:

False negative: When the model says you are not the user but you actually are.
False positive: When the model says you are the user but you are not.
True negative: When the model says you are not the user and you are not.
True positive: When the model says you are the user and you are.

Evaluation metrics

Confusion matrix

		Real Class	
		1	0
Predicted Class	1	TP	FP
	0	FN	TN

CONFUSION MATRIX

		Real Class	
		1	0
Predicted Class	1	3	1
	0	2	4

Evaluation metrics

Confusion matrix

Accuracy is the number of correctly classified examples

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

		Real Class	
		1	0
Predicted Class	1	TP	FP
	0	FN	TN

CONFUSION MATRIX

Evaluation metrics

Confusion matrix

		Real Class	
		1	0
Predicted Class	1	TP	FP
	0	FN	TN

CONFUSION MATRIX

Precision is the number of correctly classified positive examples divided by the total number of examples that are classified as positive.

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

Recall is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set.

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

$$FScore = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

Evaluation metrics

Confusion matrix

Accuracy is the number of correctly classified examples

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

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$ 0,70

		Real Class	
		1	0
Predicted Class	1	TP	FP
	0	FN	TN

CONFUSION MATRIX

		Real Class	
		1	0
Predicted Class	1	3	1
	0	2	4

Evaluation metrics

Confusion matrix

Precision is the number of correctly classified positive examples divided by the total number of examples that are classified as positive.

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

		Real Class	
		1	0
Predicted Class	1	TP	FP
	0	FN	TN

CONFUSION MATRIX

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

0,75

		Real Class	
		1	0
Predicted Class	1	3	1
	0	2	4

Evaluation metrics

Confusion matrix

Recall is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set.

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

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

0,60

		Real Class	
		1	0
Predicted Class	1	TP	FP
	0	FN	TN

CONFUSION MATRIX

		Real Class	
		1	0
Predicted Class	1	3	1
	0	2	4

Evaluation metrics

Confusion matrix

F-score

$$Precision = \frac{TP}{TP + FP}$$ 0,75

$$Recall = \frac{TP}{TP + FN}$$ 0,60

$$FScore = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$ 0,66

Overview

- Machine Learning
  - Concept
  - Application fields
  - Supervised, Unsupervised & Reinforcement learning
- Approaching a problem of learning from examples
- Building Machine Learning Models
- Supervised Learning models
  - K-NN
  - Naïve Bayes
  - Neural Networks
- Evaluating classifier performance

Introduction to Machine Learning

Thank you!

Rocío Alaiz Rodríguez

rocio.alaiz@unileon.es