

# Le deep learning et l'analyse prédictive





### **Table of Content**

```
rzipfseeko-0.so.13 -> libzzipfsee
zzinfseeko-0.so.13.0.72
ozzipfseeko.so -> libzzipfseeko-0.s
bzzipmmapped-0.so -> libzzipmmapped
ibzzipmmapped-0.so.10 -> libzzipmmar
ibzzipmmapped-0.so.11 -> libzzipmma
libzzipmmapped-0.so.12 -> libzzipmma
libzzipmmapped-0.so.13 -> libzzipmmaı
libzzipmmapped-0.so.13.0.72
libzzipmmapped.so -> libzzipmmapped-
6 libzzip.so -> libzzip-0.so.13.0.72
6 libzzipwrap-0.so -> libzzipwrap-0.so
56 libzzipwrap-0.so.13 -> libzzipwrap-0
S6 libzzipwrap-0.so.13.0.72
:56 libzzipwrap.so -> libzzipwrap-0.so.1
11:24 11vm14
12:09 LLVMgold.so
2:11 locale
09:08 localepaper
  2022 mail-dotlock
```

- **1.Introduction to ML and DS in Python**
- 2.Deep Learning with Tensorflow
- 3. Containerized Machine Learning
- 4.Azure IA
- 5.Dataiku
- 6.Assessment

#### zipfseeko-0.so.13 -> libzzipfsee zzipfseeko-0.so.13.0.72 ozzipfseeko.so -> libzzipfseeko-0.s bzzipmmapped-0.so -> libzzipmmapped ibzzipmmapped-0.so.10 -> libzzipmmar ibzzipmmapped-0.so.11 -> libzzipmmap libzzipmmapped-0.so.12 -> libzzipmma libzzipmmapped-0.so.13 -> libzzipmma libzzipmmapped-0.so.13.0.72 libzzipmmapped.so -> libzzipmmapped-6 libzzip.so -> libzzip-0.so.13.0.72 6 libzzipwrap-0.so -> libzzipwrap-0.so 56 libzzipwrap-0.so.13 -> libzzipwrap-0 56 libzzipwrap-0.so.13.0.72 :56 libzzipwrap.so -> libzzipwrap-0.so.1 11:24 **11vm14** 12:09 LLVMgold.so 22:11 locale 09:08 localepaper 2021 **lua** 2022 mail-dotlock 11:24 man-db 2021 marble 17:00 mbim-proxy

## 1. Introduction to ML and DS in Python

### Al vs ML vs DL

### Artificial Intelligence

Development of smart systems and machines that can carry out tasks that typically require human intelligence

### **2** Machine Learning

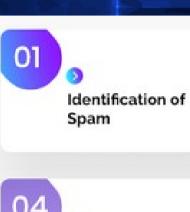
Creates algorithms that can learn from data and make decisions based on patterns observed

Require human intervention when decision is incorrect

### 3 Deep Learning

Uses an artificial neural network to reach accurate conclusions without human intervention

## Some AI Applications





02 Recommending Products



O3 Customer Segmentation





Image and Video Recognition



05

Fraudulent Transactions



06

Demand Forecasting



07

Virtual Personal Assistant



08

Sentiment Analysis

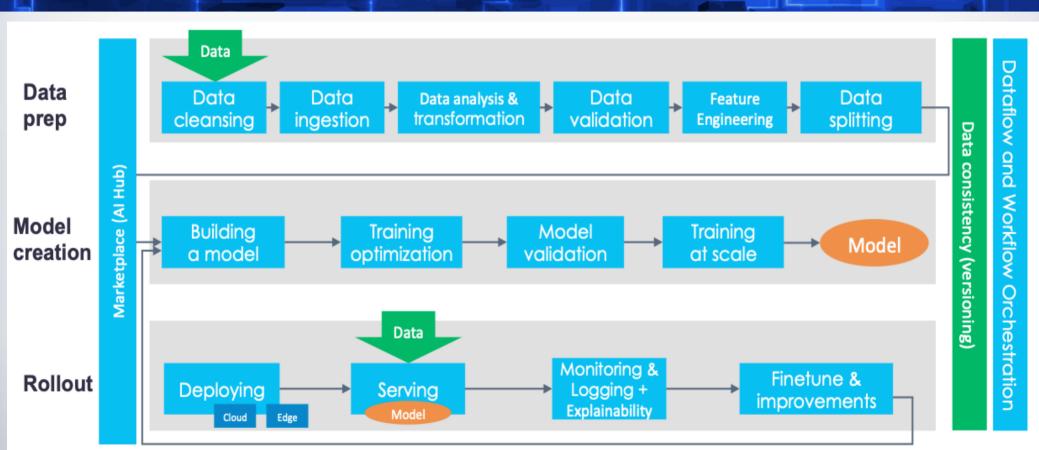


09

Customer Service Automation

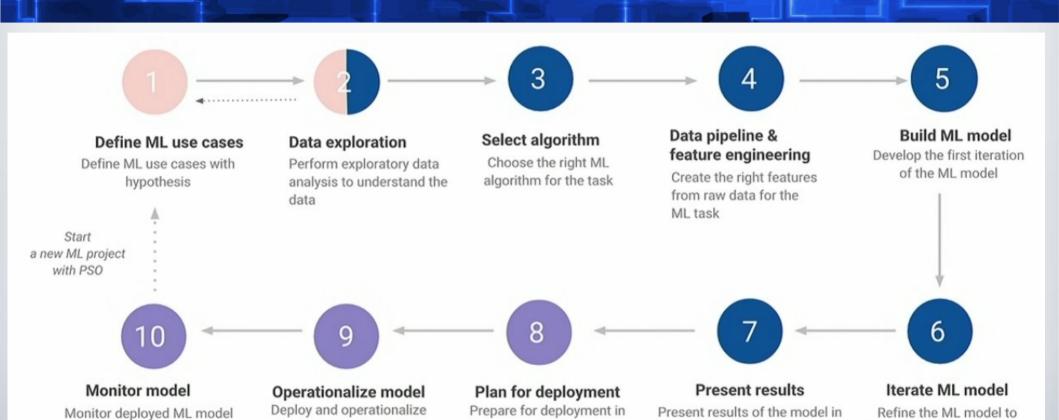


## **AI-ML Pipelines**



Lotfi Hocini

## **AI-ML Pipelines**



production

ML model in production

and retrain or rebuild when

performance degrades

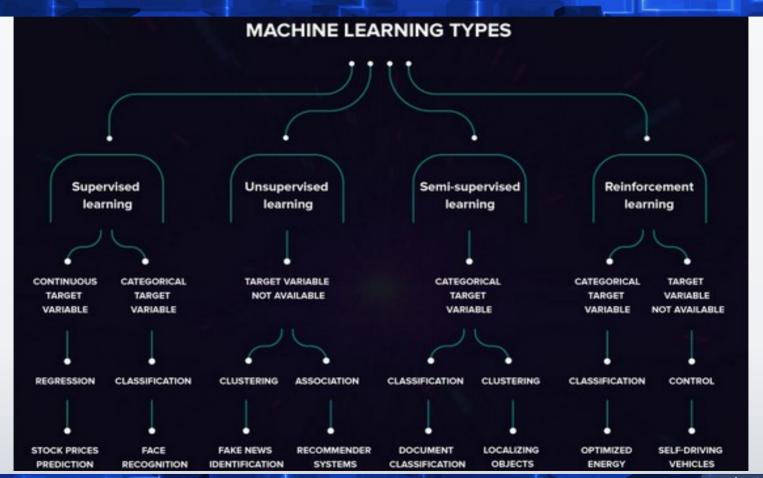
improve performance and

efficacy

a way that demonstrates its

value to stakeholders

## Some ML Approaches



Lotfi Hocini

### Supervised vs Unsupervised ML

Input data is labeled

Has a feedback mechanism

Data is classified based on the training dataset

Divided into Regression & Classification

Used for prediction

#### Unsupervised learning

Input data is unlabeled

Has no feedback mechanism

Assigns properties of given data to classify it

Divided into Clustering & Association

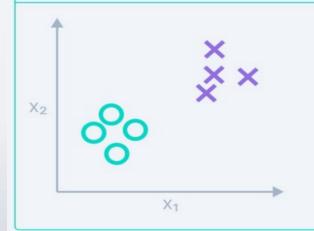
Used for analysis

## Supervised vs Unsupervised ML

#### Supervised learning

Algorithms include: decision trees, logistic regressions, support vector machine

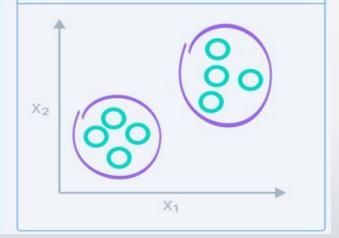
A known number of classes



#### Unsupervised learning

Algorithms include: k-means clustering, hierarchical clustering, apriori algorithm

A unknown number of classes



## **Some Metrics** - Regression -

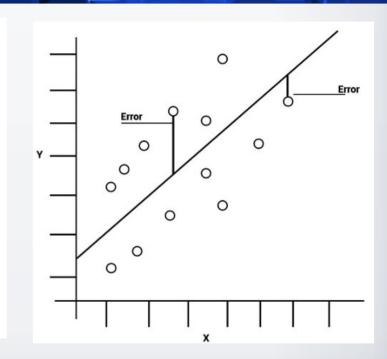
### 1.3 Minimizing the MSE

First, we find the gradient of the MSE with respect to  $\beta$ :

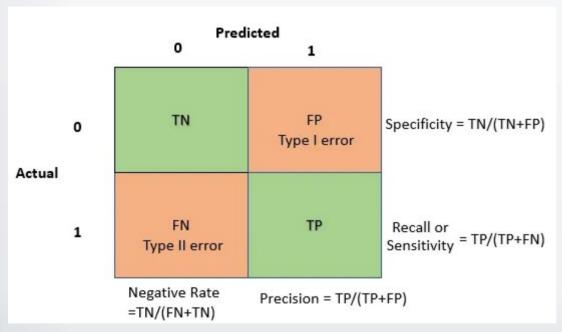
$$\nabla MSE(\beta) = \frac{1}{n} \left( \nabla \mathbf{y}^T \mathbf{y} - 2\nabla \beta^T \mathbf{x}^T \mathbf{y} + \nabla \beta^T \mathbf{x}^T \mathbf{x} \beta \right)$$
$$= \frac{1}{n} \left( 0 - 2\mathbf{x}^T \mathbf{y} + 2\mathbf{x}^T \mathbf{x} \beta \right)$$
$$= \frac{2}{n} \left( \mathbf{x}^T \mathbf{x} \beta - \mathbf{x}^T \mathbf{y} \right)$$

We now set this to zero at the optimum,  $\widehat{\beta}$ :

$$\mathbf{x}^T \mathbf{x} \widehat{\beta} - \mathbf{x}^T \mathbf{y} = 0$$



## **Some Metrics**- Binary Classification -

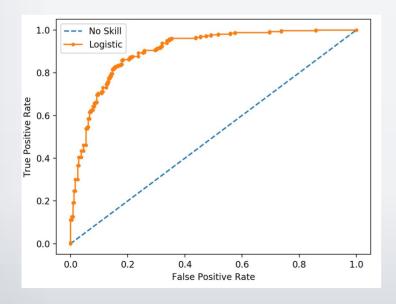


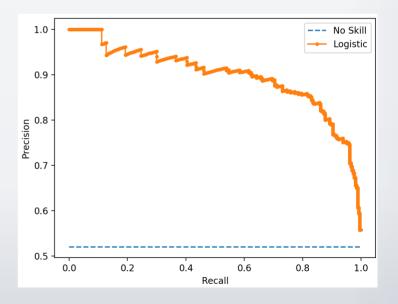
TP+TN

#### **Confusion Matrix**

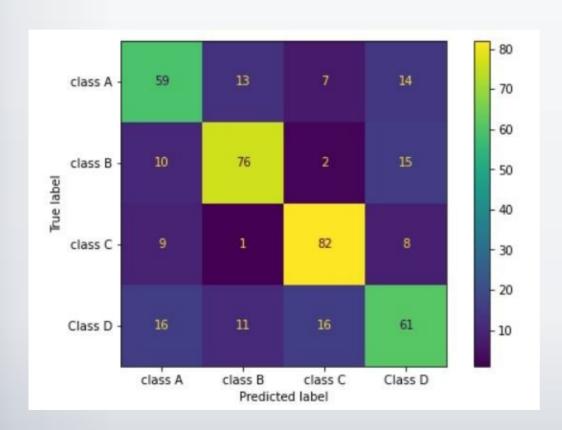
## **Some Metrics**- Binary Classification -

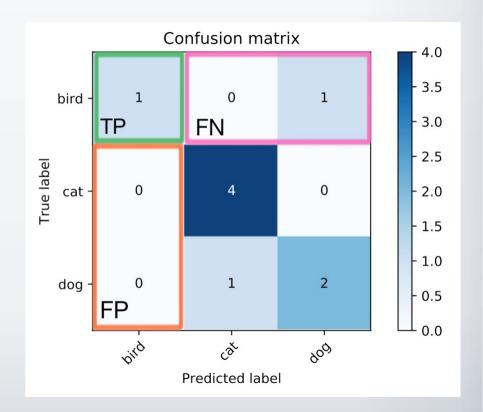
- ROC curves should be used when there are roughly equal numbers of observations for each class.
- Precision-Recall curves should be used when there is a moderate to large class imbalance.





## Some Metrics - Multi Class Classification -





## Some Metrics - Multi Class Classification -

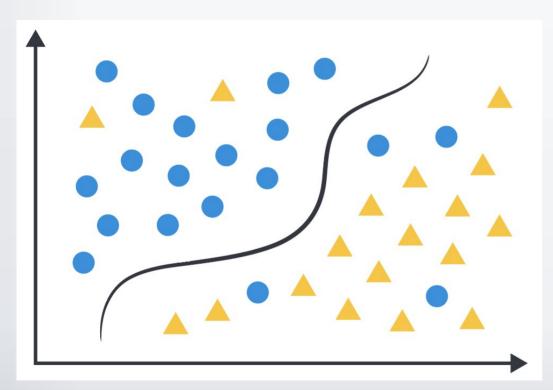
### Micro average method

- Sum up individual tp, fp.
- Micro precision =  $tp_1+tp_2+..tp_n/(tp_1+tp_2+..tp_n+fp_1+fp_2+..fp_n)$
- Micro recall =  $= tp_1 + tp_2 + ...tp_n / (tp_1 + tp_2 + ...tp_n + fn_1 + fn_2 + ...fn_n)$

### Macro average method

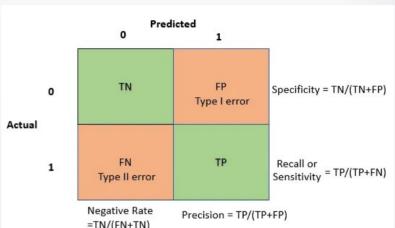
Compute metric independently for each class and then take average

## **Exercice 1**

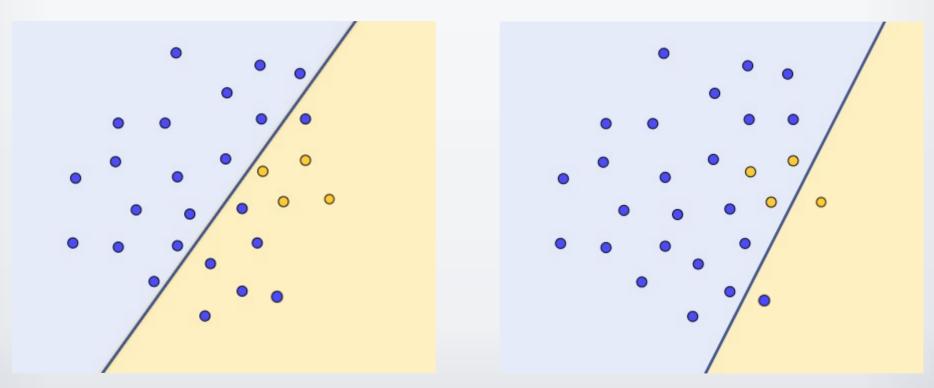


Accuracy =  $\frac{TP+TN}{TP+FP+TN+FN}$ F1 - Score =  $\frac{2*Recall*Precision}{Recall+Precision}$ 

Calculate the classification metrics



## **Exercice 2**



- Calculate the classification metrics
- Derive insights

## **Configure VSCode**

- Install Data Science Profil
- Install Additional extensiions

## **Python - Advanced**

https://www.geeksforgeeks.org/python-programming-language/?ref=lbp



19 Lotfi Hocini

### **Python - Virtual environment**

https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtualents/



20

### Python - FastApi

https://fastapi.tiangolo.com/tutorial/



### **ML Practice**

- Topics: Data Preparation, Missing Data, Imbalanced data, Split Datasets, Metrics, Visualization, model selection, saving ...
- Database: 'data' Folder

#### To install:

- pip install virtualenv, then create virtual environment and activate it
- pip install ipykernel
- pip install pandas
- Pip install numpy
- pip install matplotlib
- pip install seaborn
- pip install scikit-learn

### **Next Sesssion**

2. Deep Learning with Tensorflow