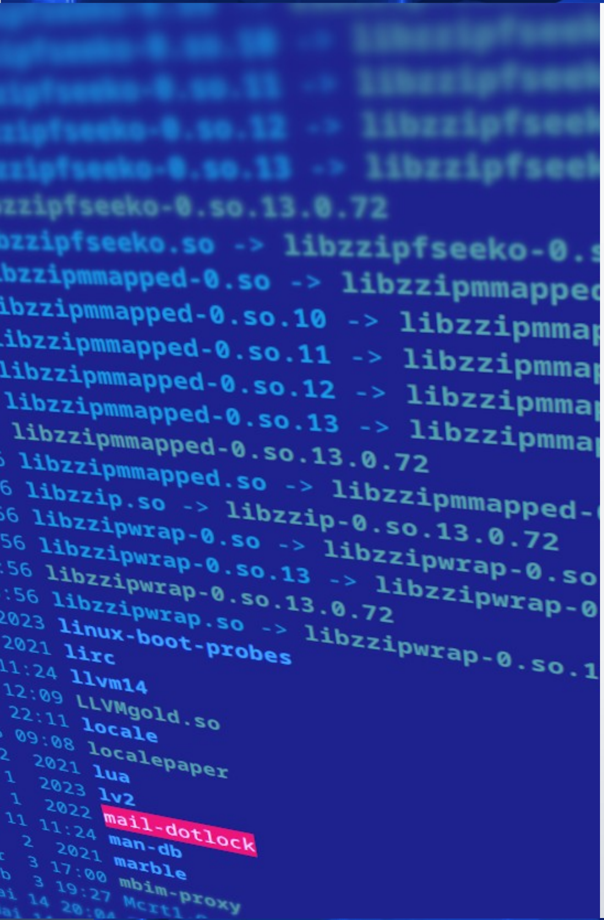
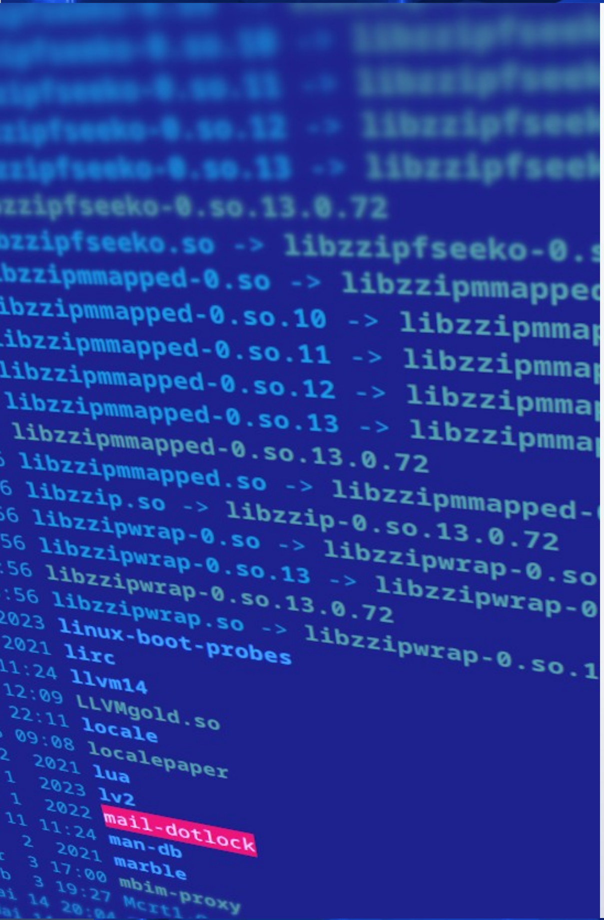
Issy-les-Moulineaux
26/02/2024

Le deep learning et l'analyse prédictive

Lotfi Hocini

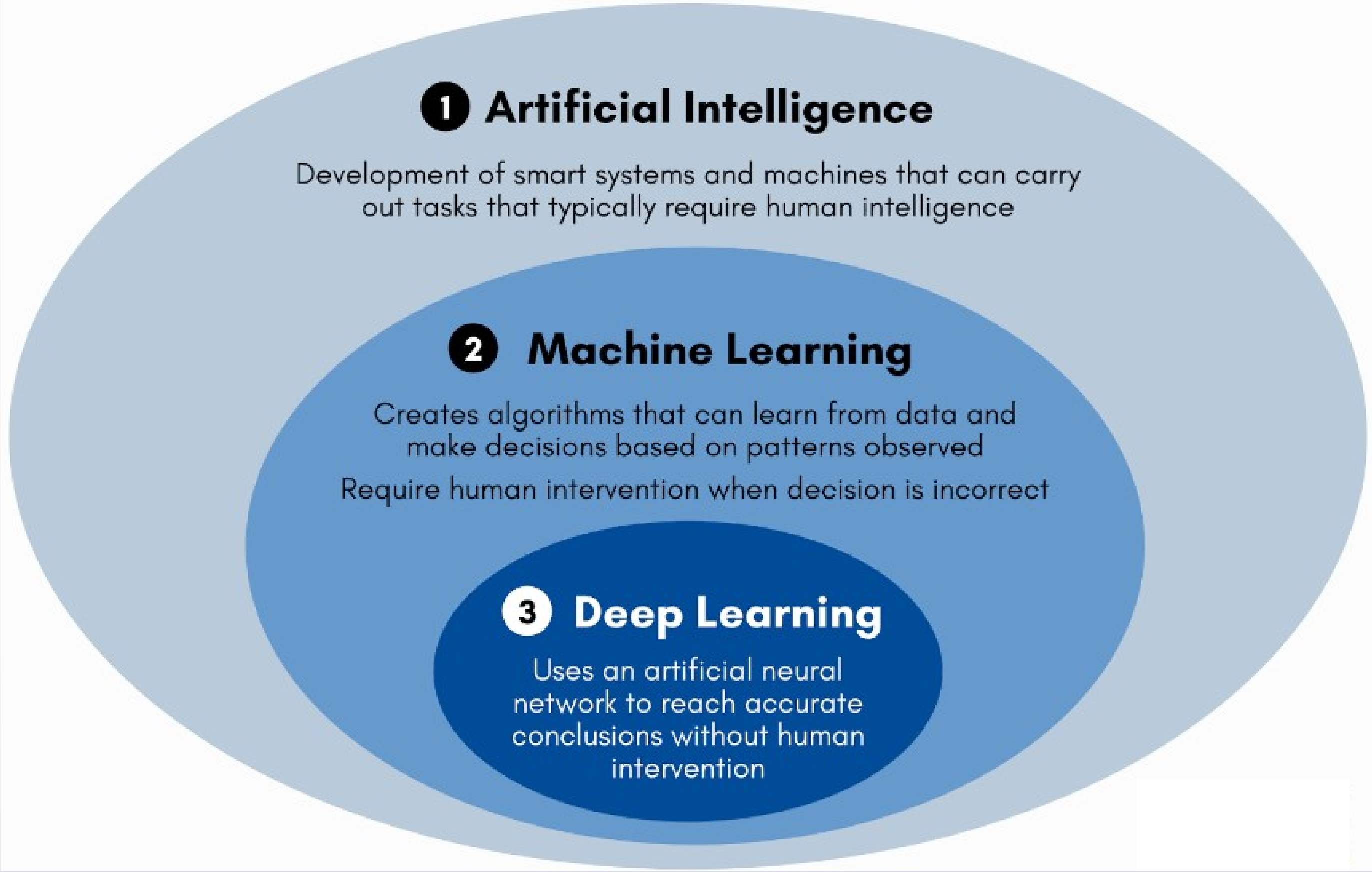
Table of Content



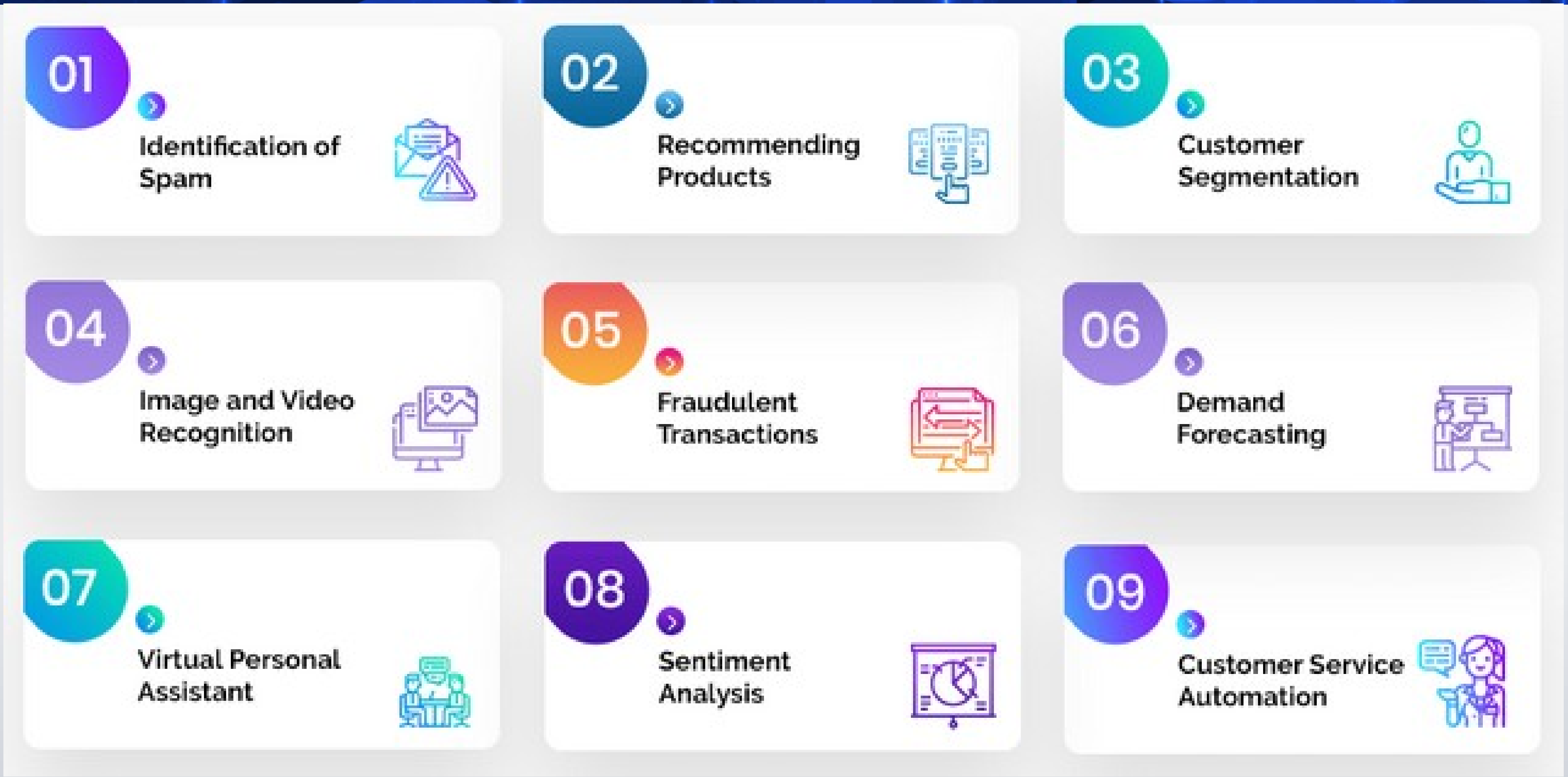


1. Introduction to ML and DS in Python

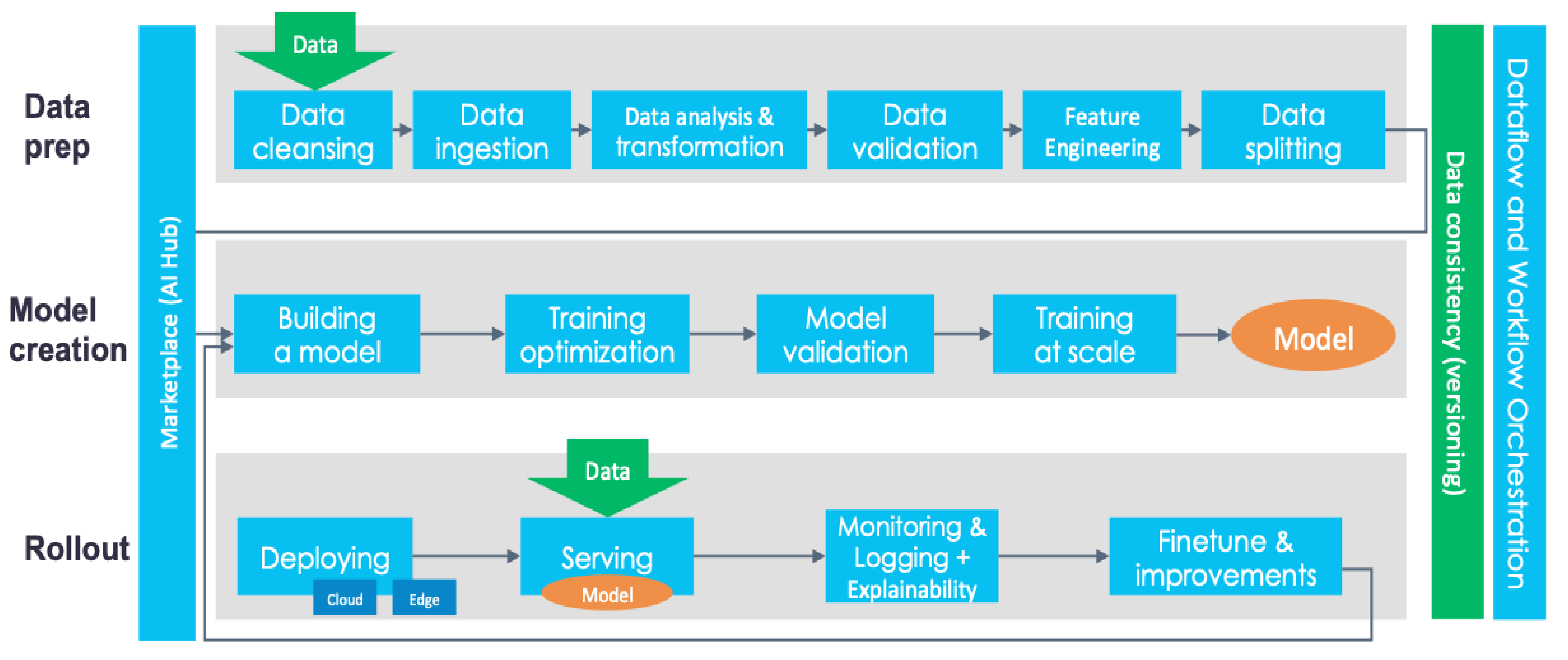
AI vs ML vs DL


1 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

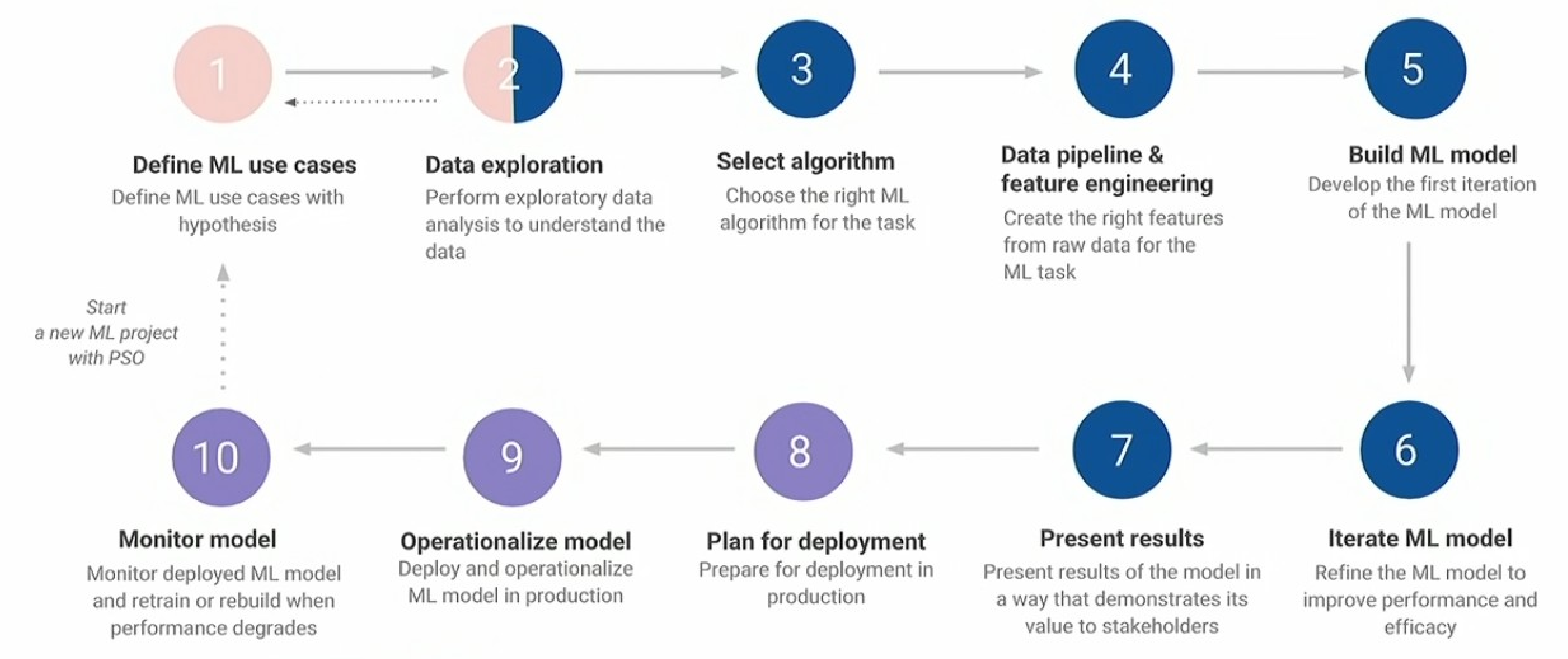
Some AI Applications



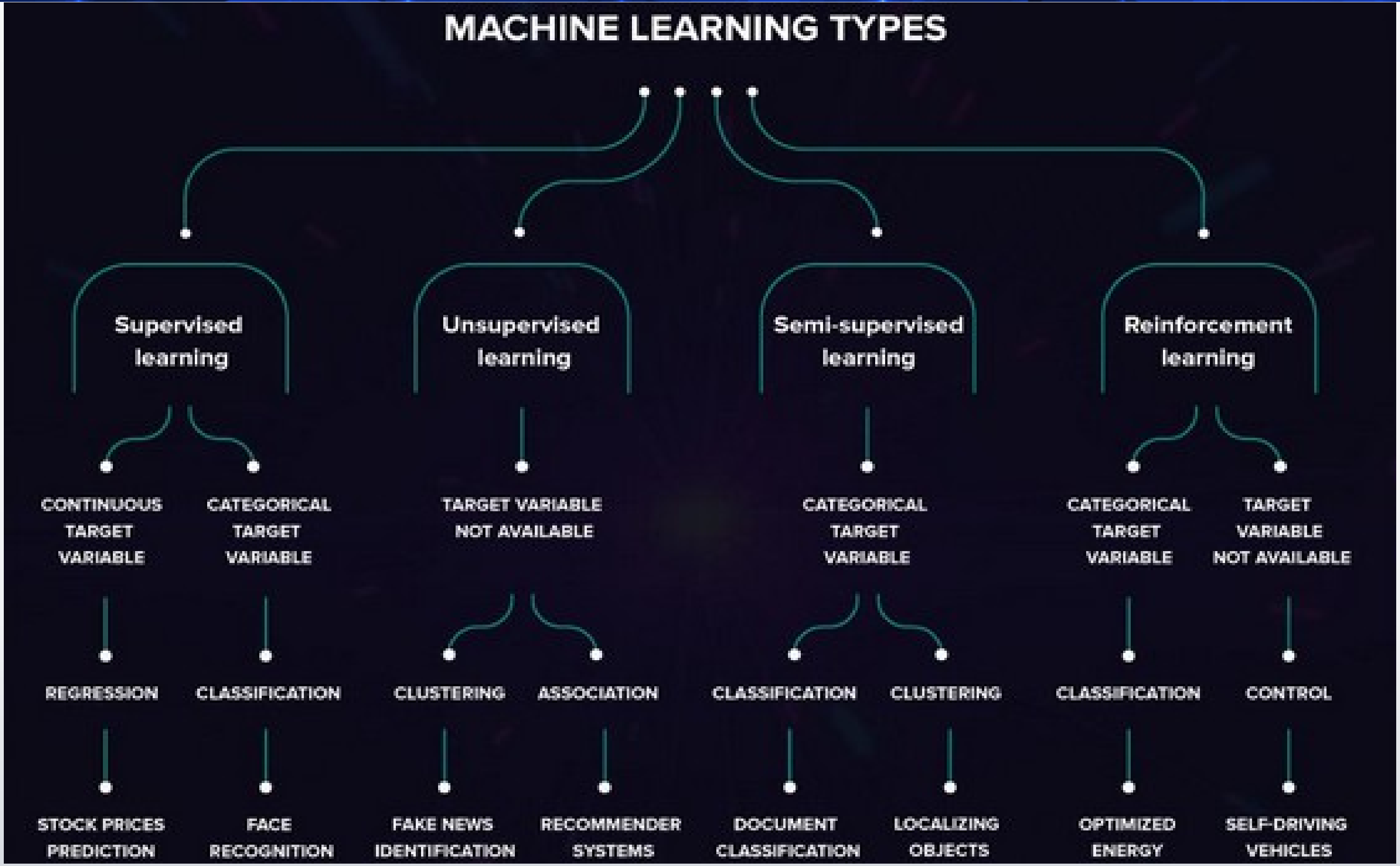
AI-ML Pipelines



AI-ML Pipelines



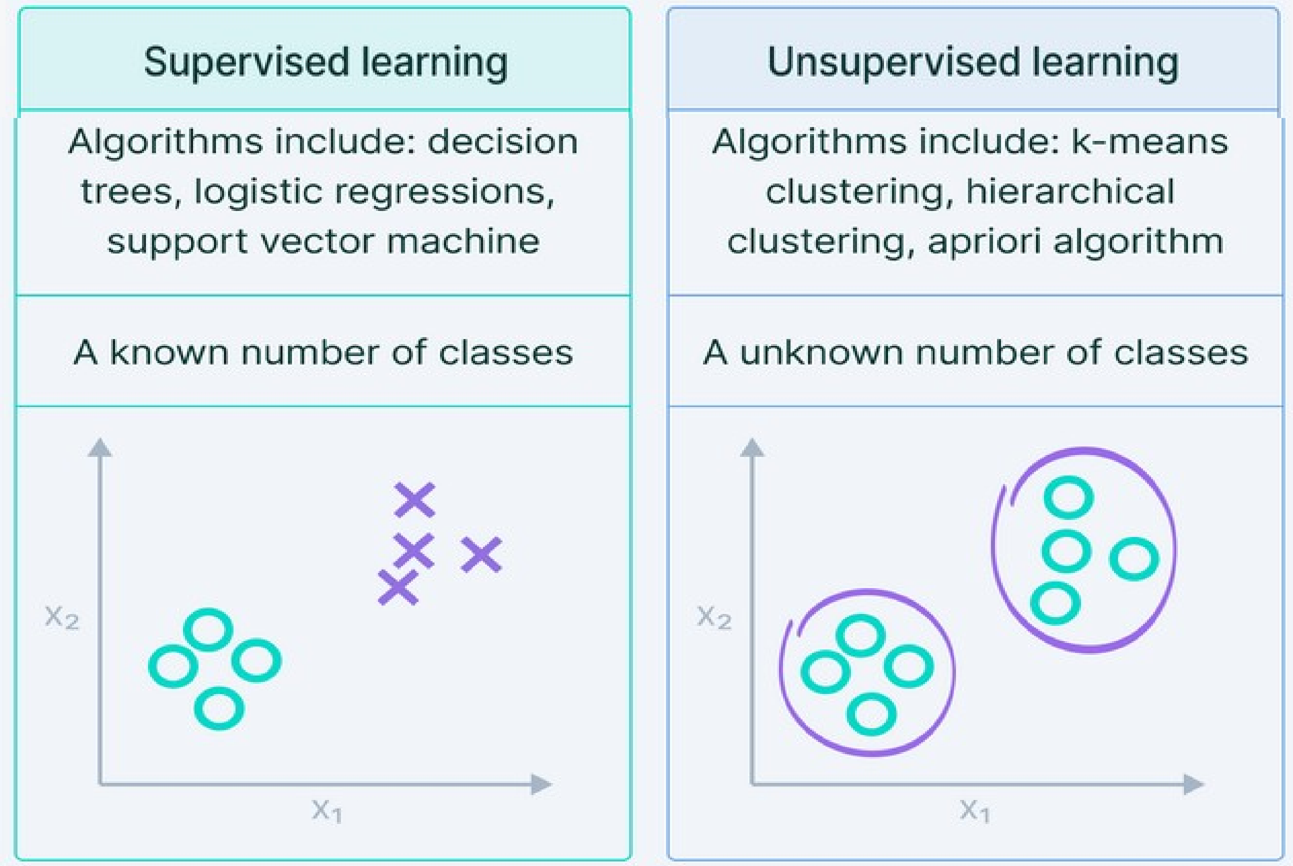
Some ML Approaches



Supervised vs Unsupervised ML

Supervised learning	Unsupervised learning
Input data is labeled	Input data is unlabeled
Has a feedback mechanism	Has no feedback mechanism
Data is classified based on the training dataset	Assigns properties of given data to classify it
Divided into Regression & Classification	Divided into Clustering & Association
Used for prediction	Used for analysis

Supervised vs Unsupervised ML



Some Metrics
- Regression -

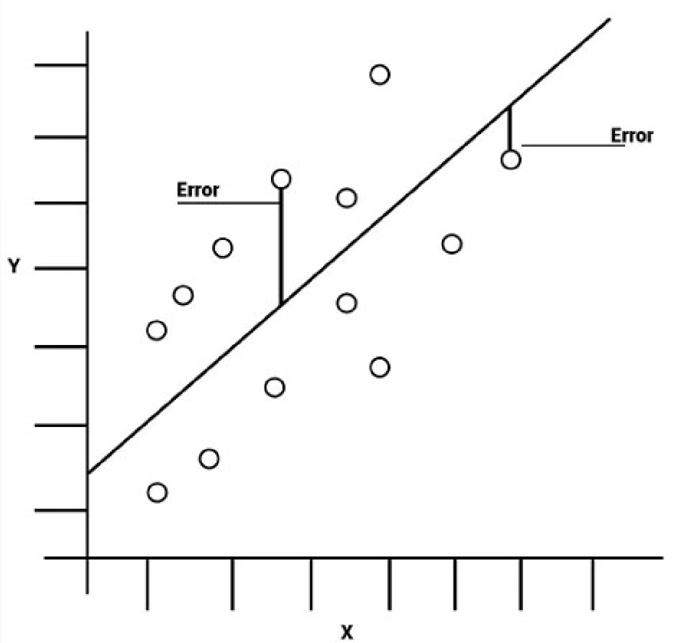
1.3 Minimizing the MSE

First, we find the gradient of the MSE with respect to β:

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

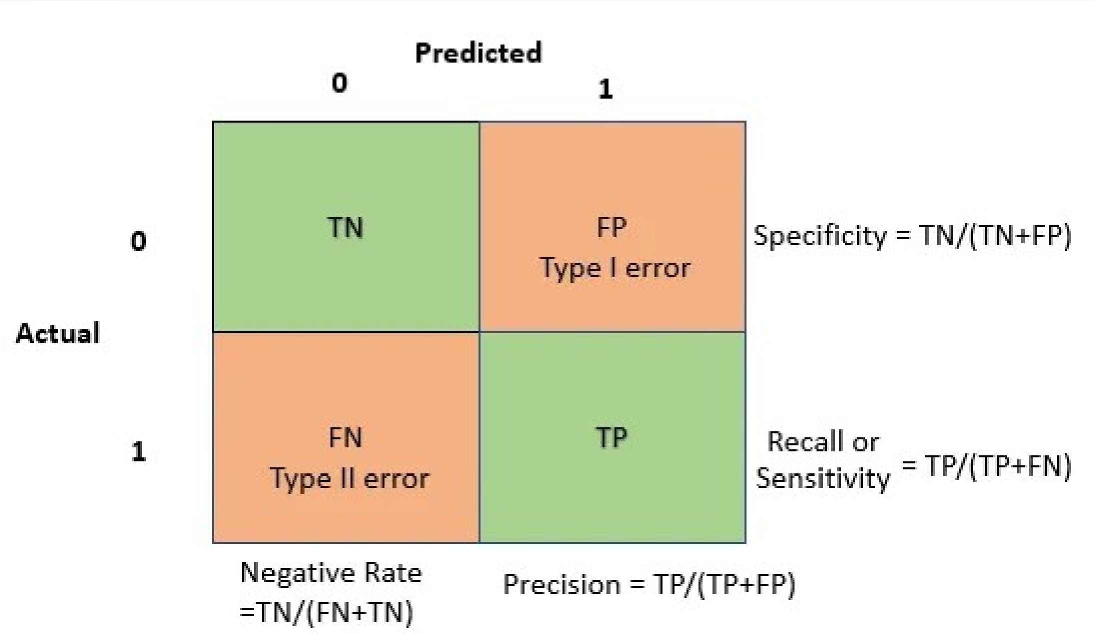
We now set this to zero at the optimum, $\widehat{\beta}$:

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



$$\widehat{\beta} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y}$$

Some Metrics
- Binary Classification -



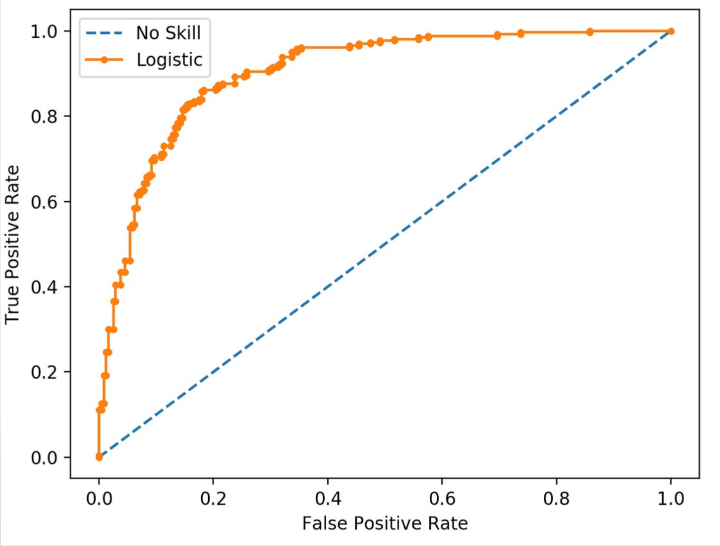
Confusion Matrix

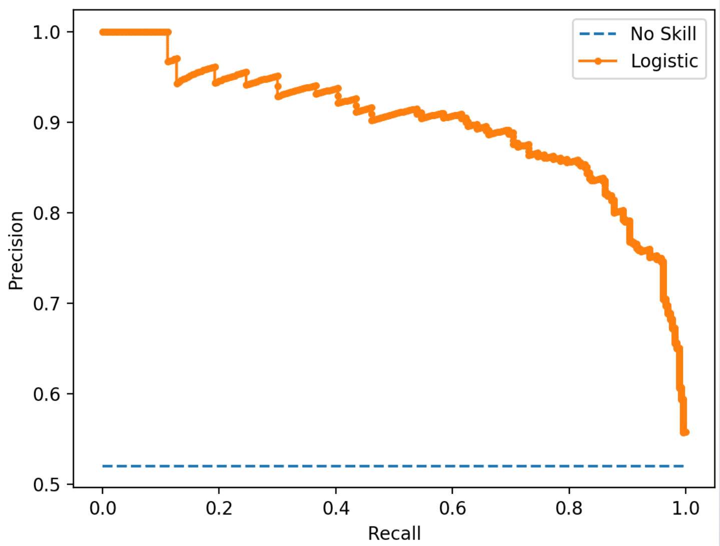
$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

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

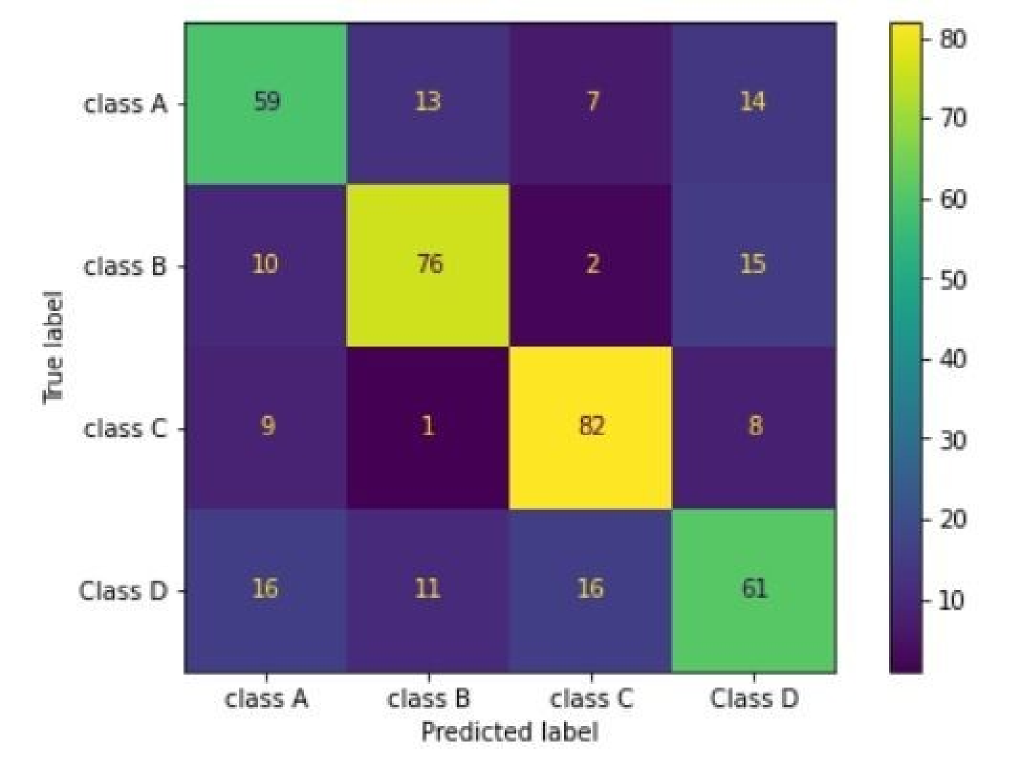
Some Metrics
- Binary Classification -

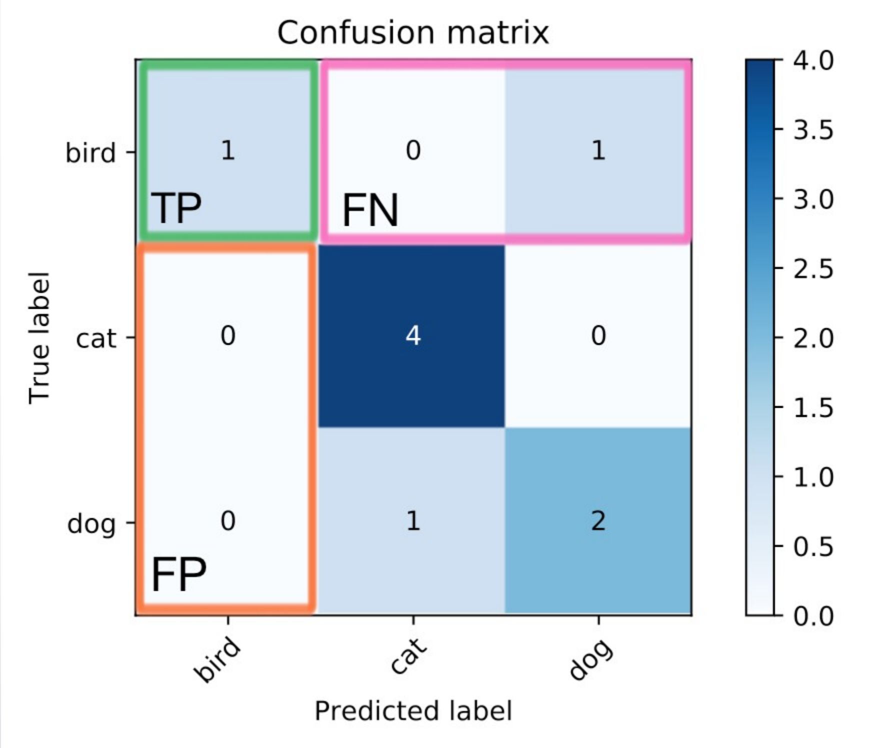
- **ROC curves** should be used when there are roughly equal numbers of observations for each class.
- **Precision-Recall** curves should be used when there is a moderate to large class imbalance.





Some Metrics
- Multi_Class Classification -



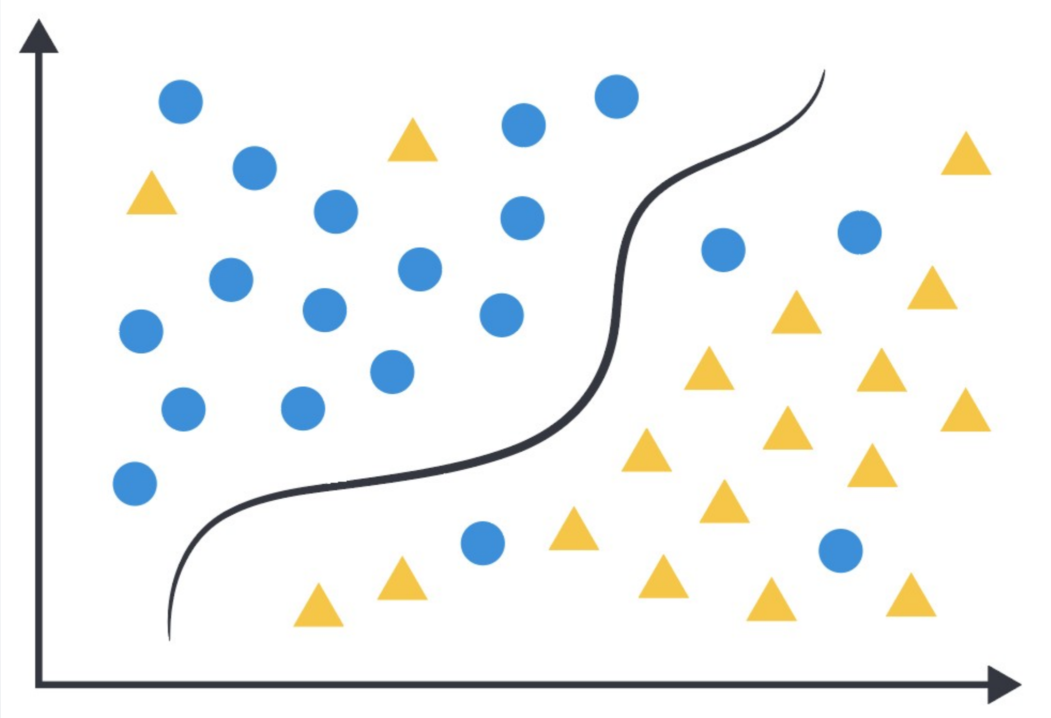


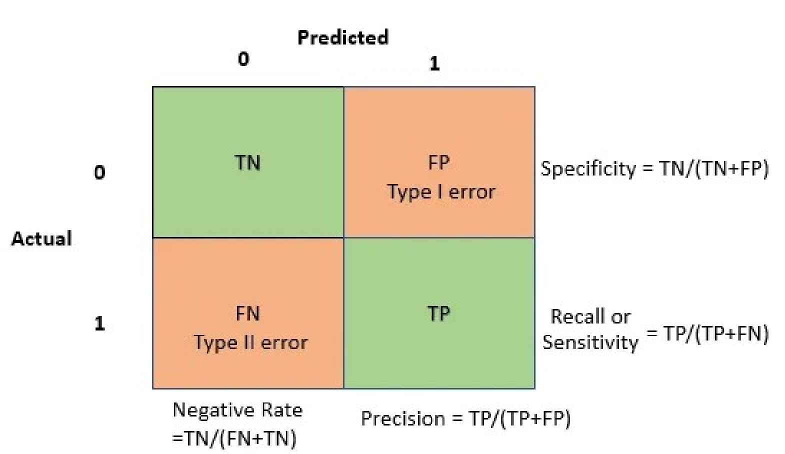
Some Metrics
- Multi_Class Classification -

- **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



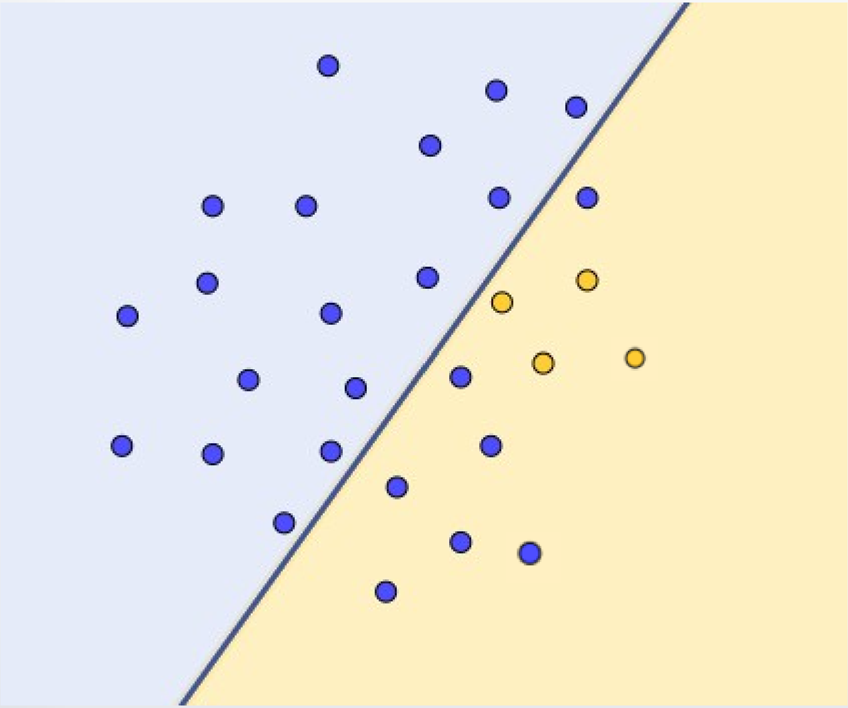


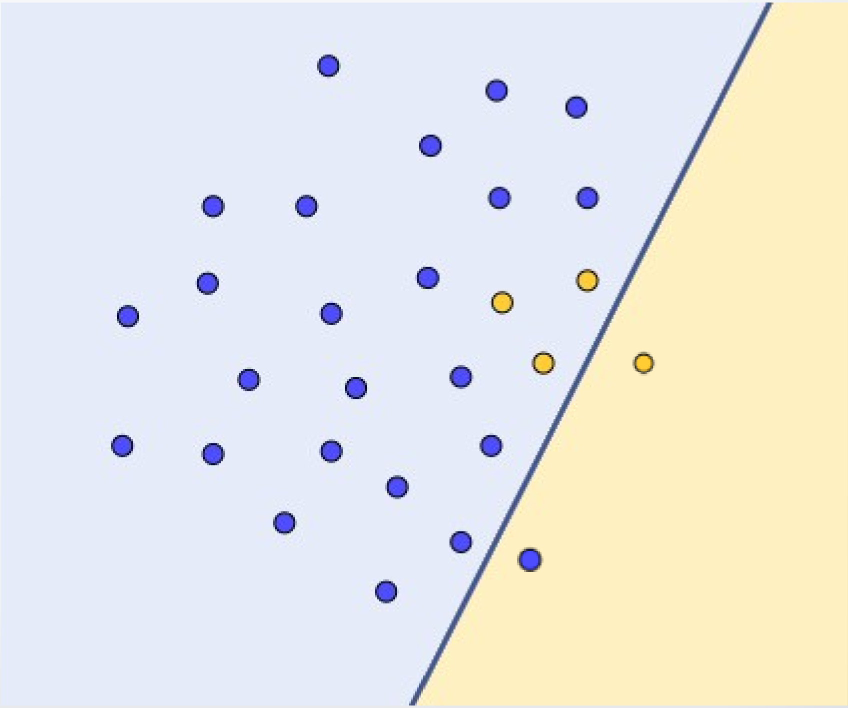
$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

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

- Calculate the classification metrics

Exercice 2





- Calculate the classification metrics
- Derive insights

Configure VSCode

- Install Data Science Profil
- Install Additional extenstions

Python – Advanced

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

Python – Virtual environment

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

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**