Université IBN TOFAIL – Kénitra
ECOLE NATIONALE DES SCIENCES APPLIQUEES

Cours Intelligence Artificielle

CH2: Machine Learning
- Concepts de Base

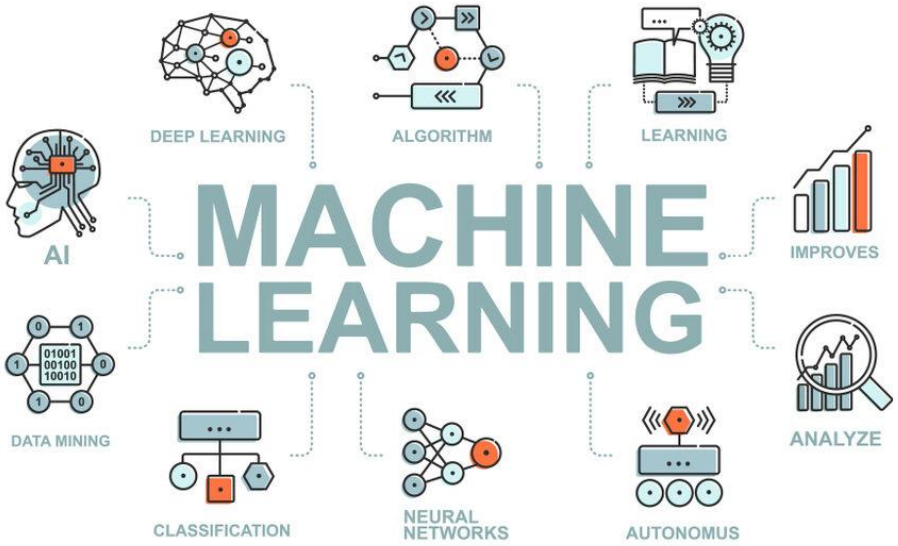
Par:

Hassan EL FADIL

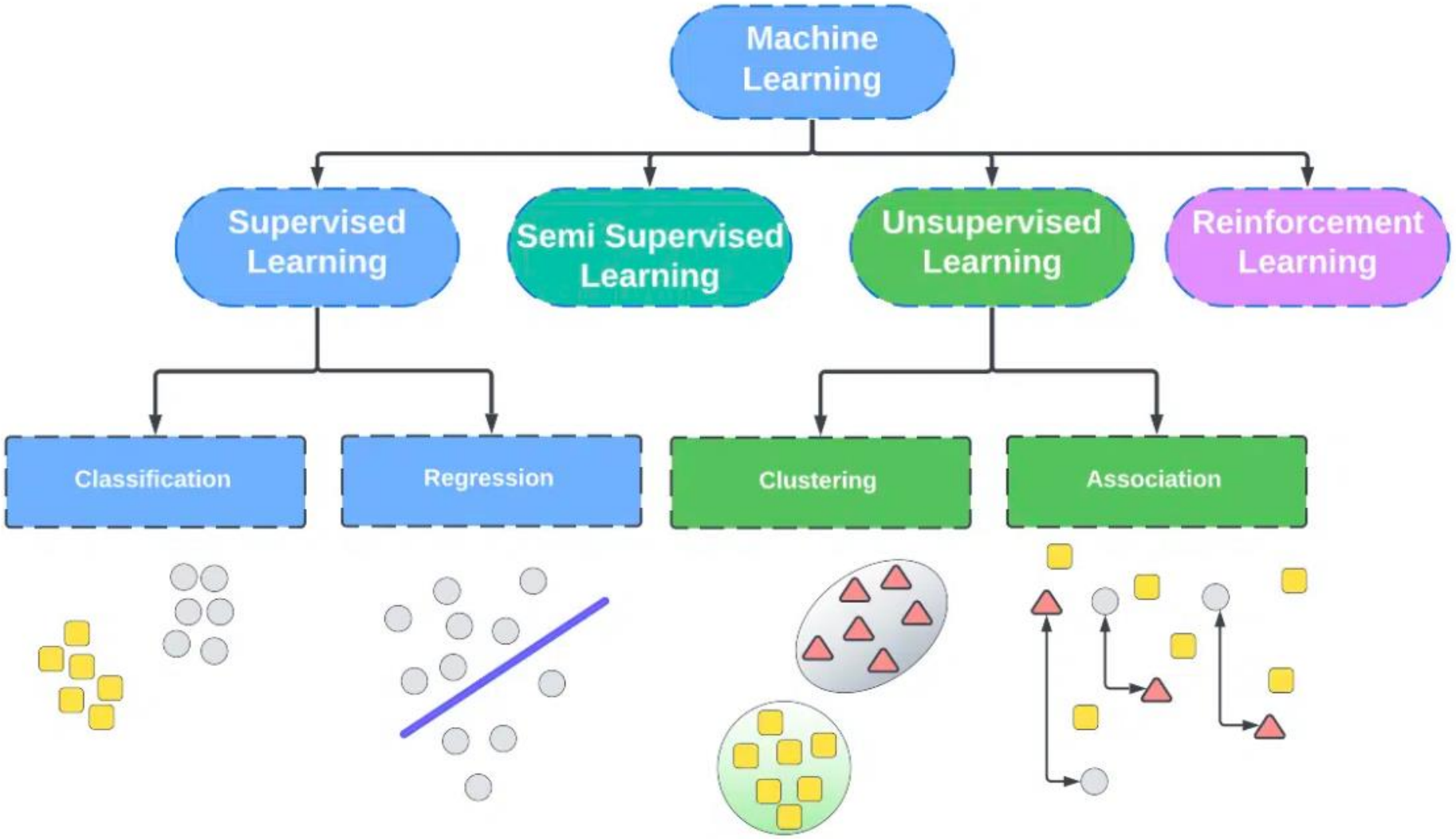
elfadil.h@gmail.com

1. Définition de ML

- Le **Machine Learning (ML)**, ou **apprentissage automatique**, est une branche de l'intelligence artificielle (IA) qui permet aux ordinateurs d'apprendre à partir de données, sans avoir besoin d'être explicitement programmés.
- Le principe fondamental du ML repose sur l'utilisation d'algorithmes capables d'analyser des données, d'en extraire des modèles ou des structures, puis de faire des prédictions ou des prises de décision basées sur ces modèles.

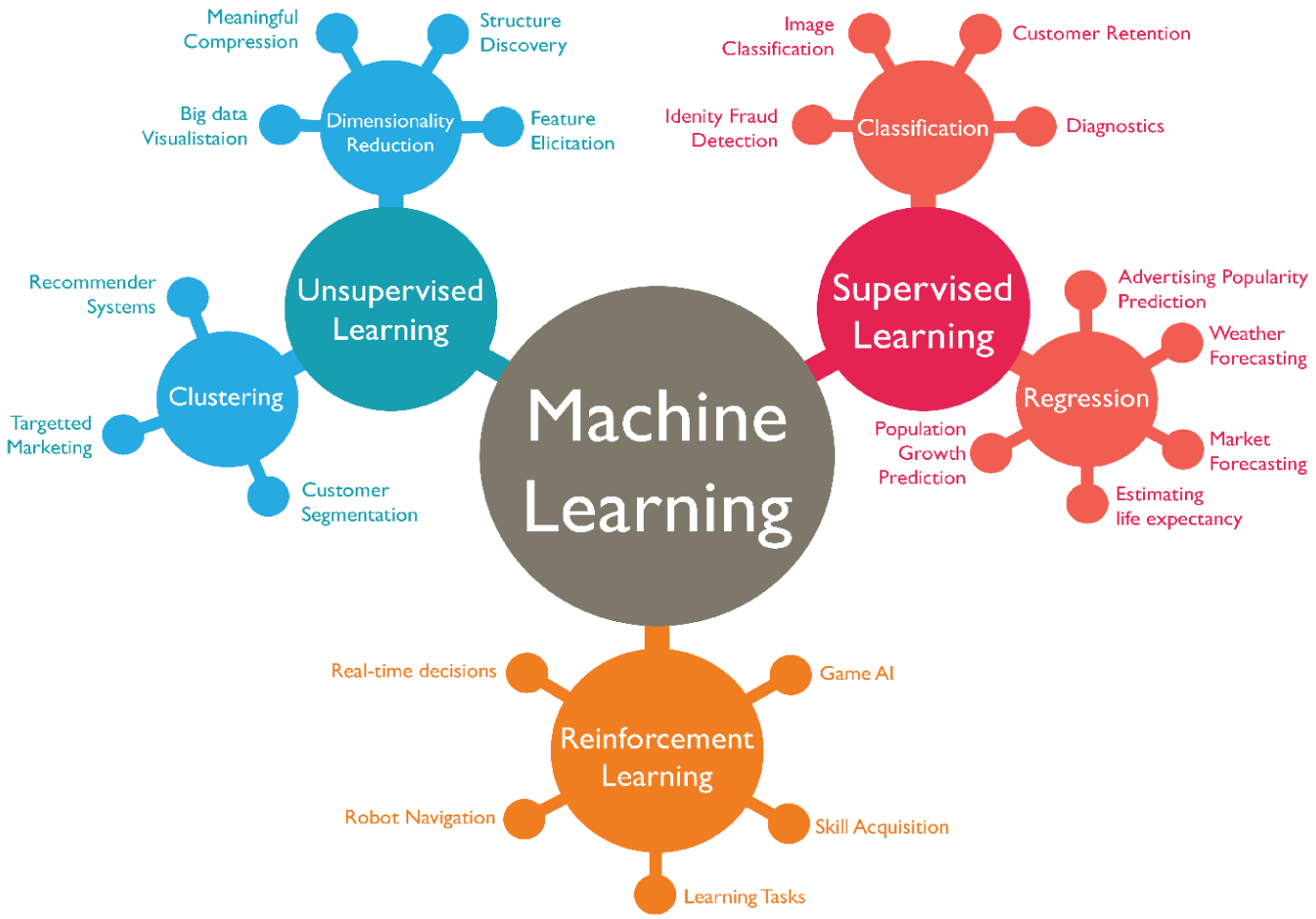


2. Types d'apprentissage de ML



…Types d'apprentissage de ML

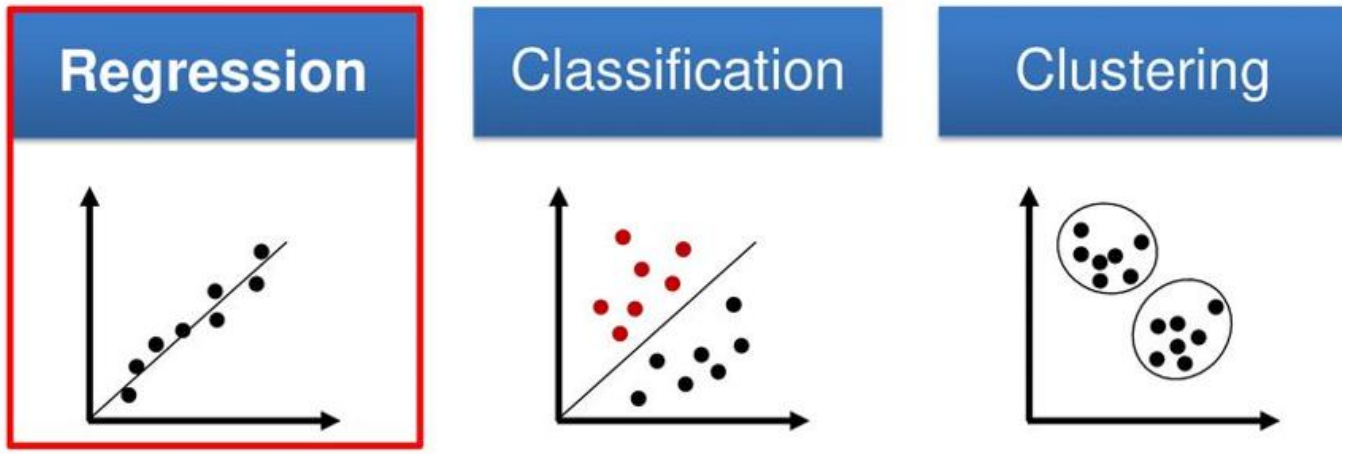
- **Supervisé** : L'algorithme apprend à partir de données annotées (par exemple, des exemples de questions et leurs réponses).
- **Non supervisé** : L'algorithme trouve des structures dans les données sans labels (par exemple, regrouper des clients).
- **Renforcement** (**Reinforcement Learning**, ou RL): L'algorithme **apprend par essais et erreurs**, en recevant des récompenses ou des punitions pour ses actions.



…Types d'apprentissage de ML

- **Classification** : Décide **à quelle catégorie** appartient quelque chose.
  - *Prédire si un email est **spam** ou **non-spam** (2 classes).*
  - *Classer des images de fruits en **pommes**, **bananes**, ou **oranges** (plusieurs classes).*
- **Régression** : Prédit **une valeur numérique**.
  - *Prédire le prix d'une maison en fonction de sa surface et de son emplacement.*
  - *Prédire la température pour demain à partir des données météorologiques actuelles.*
- **Clustering** : Regroupe des éléments similaires **sans labels** prédéfinis.
  - *Regrouper les clients d'une entreprise en fonction de leur comportement d'achat: les types de produits qu'ils achètent , la fréquence d'achat (clients réguliers ou occasionnels), le montant qu'ils dépensent (petits, moyens ou gros acheteurs).*
  - *Identifier des groupes dans des données de génétique ou de géographie: Regrouper des individus ayant des profils génétiques similaires, Identifier des populations ayant une origine commune ou des traits partagés, Découvrir des groupes liés à des risques spécifiques de maladies.*

…Types d'apprentissage de ML

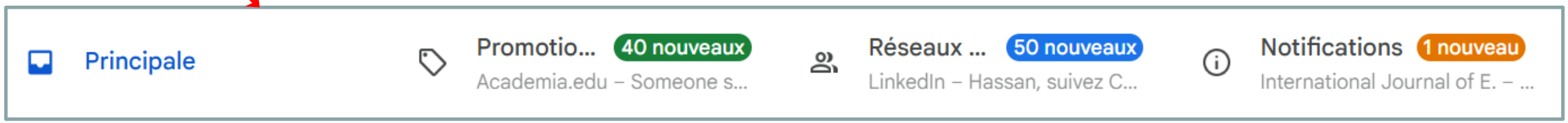


- House pricing
- Sales
- Persons weight

- Object detection
- Spam detection
- Cancer detection

- Genome patterns
- Google news
- Pointcloud (Lidar) processing

Classification des emails dans Gmail



3. Métriques de performance (Performance Metrics)

- Les métriques de performance mesurent l'efficacité et la pertinence d'un modèle de Machine Learning par rapport aux objectifs fixés. Elles permettent d'évaluer objectivement si le modèle répond aux besoins.
- **Classification** : Accuracy, F1-Score, ROC-AUC (selon classes).
- **Régression** : RMSE, MAE, R^2.
- **Clustering** : Silhouette Score, ARI, Dunn Index.
- **Renforcement** : Cumulative Reward, Discounted Return.

3.1. Métriques pour la Classification

3.1.1. Matrice de confusion:

- Une matrice de confusion est utilisée pour évaluer les performances des algorithmes de classification
- Une matrice de confusion comporte deux lignes et deux colonnes pour la classification binaire.
- Le nombre de lignes et de colonnes est égal au nombre de classes.
- Les colonnes sont les classes prédites et les lignes sont les classes réelles.

	Predicted (0)	Predicted (1)
Actual (0)	True Negatives (TN)	False Positives (FP) Type 1 error
Actual (1)	False Negatives (FN) Type 2 error	True Positives (TP)

1. **Vrais positifs (TP):** Valeur réelle est 1 et la valeur prédite est également 1
2. **Vrais négatifs (TN):** Valeur réelle est 0 et la valeur prédite est également 0
3. **Faux positifs (FP) (erreur de type 1):** Valeur réelle est 0 mais la valeur prédite est 1
4. **Faux négatifs (FN) (erreur de type 2):** Valeur réelle est 1 mais la valeur prédite est 0

3.1.2. Classes équilibrées

Précision Globale (Accuracy): Utile si toutes les classes ont une importance égale.

$$Acuracy = \frac{Number\ of\ Correct\ Predictions\ (\sum TP + \sum TN)}{Total\ Population}$$

Exemple:
Valeurs réelles: y=[0,1,1,0,1,0,0,1,1,1]
valeurs prédites: $\hat{y}$=[0,0,1,0,1,0,1,1,0,1] :

$$Acuracy = \frac{1+0+1+1+1+1+0+1+0+1}{10} = 70\%$$

3.1.3. Classes déséquilibrées

Precision : Pour mesurer combien de prédictions positives sont correctes.

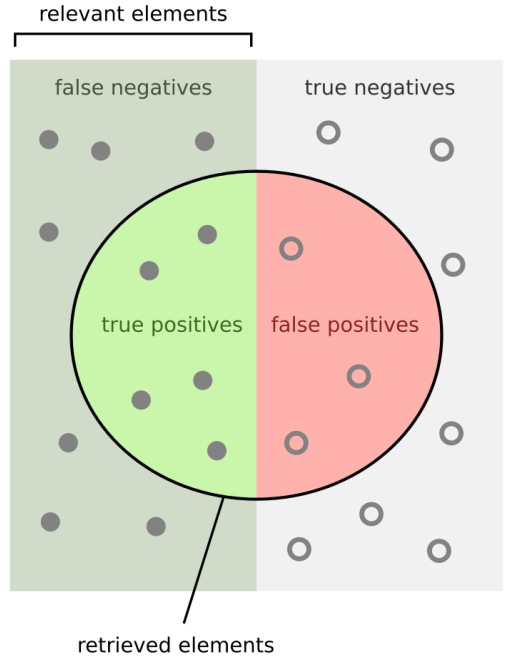
$$Precision = \frac{\sum TP}{\sum TP + \sum FP}$$

Recall (Rappel) :Pour mesurer combien de cas positifs réels sont capturés.

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

F1-Score *:* (surtout si les classes sont très déséquilibrées) :Moyenne harmonique de la précision et du rappel.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$



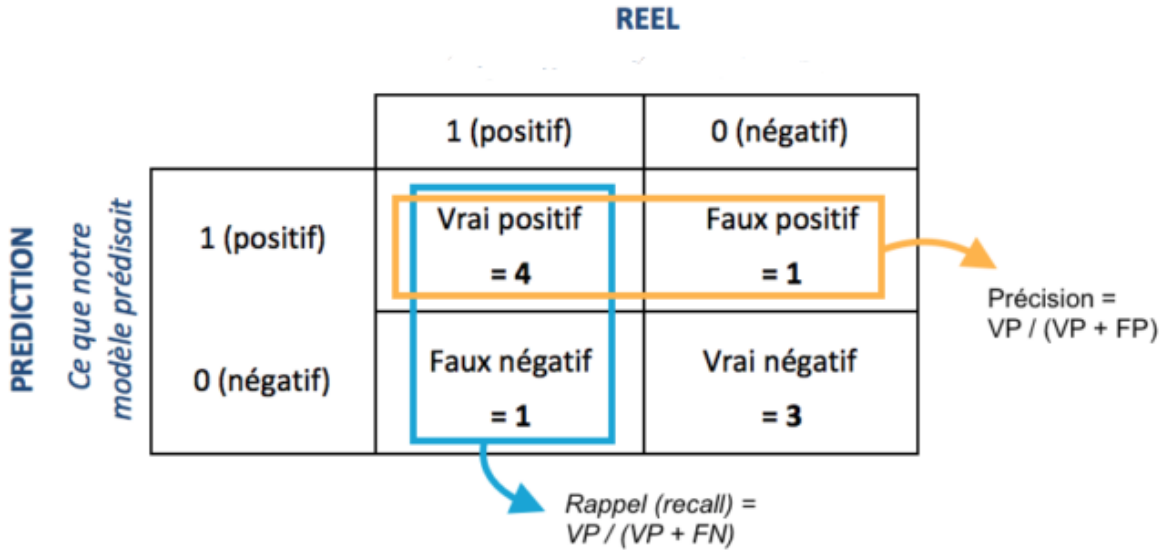
… Métriques pour la Classification

Exemple:
Valeurs réelles: y = [0,1,1,0,1,0,0,1,1,1]
valeurs prédites: $\hat{y}$ = [0,0,1,0,1,0,1,1,0,1]

$$Precision = \frac{\sum TP}{\sum TP + \sum FP} = \frac{4}{5} = 0.8$$

$$Recall = \frac{\sum TP}{\sum TP + \sum FN} = \frac{4}{5} = 0.8$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 0.8 = 80\%$$



À comparer avec $Acuracy = 70\%$

3.1.4. ROC-AUC (Receiver Operating Characteristic - Area Under Curve)

Taux de vrais positifs (True Positive Rate, TPR) : Le TPR peut être calculé à l'aide de la formule ci-dessous. Un taux Vrai positif plus élevé signifie que tous les points positifs sont classés correctement.

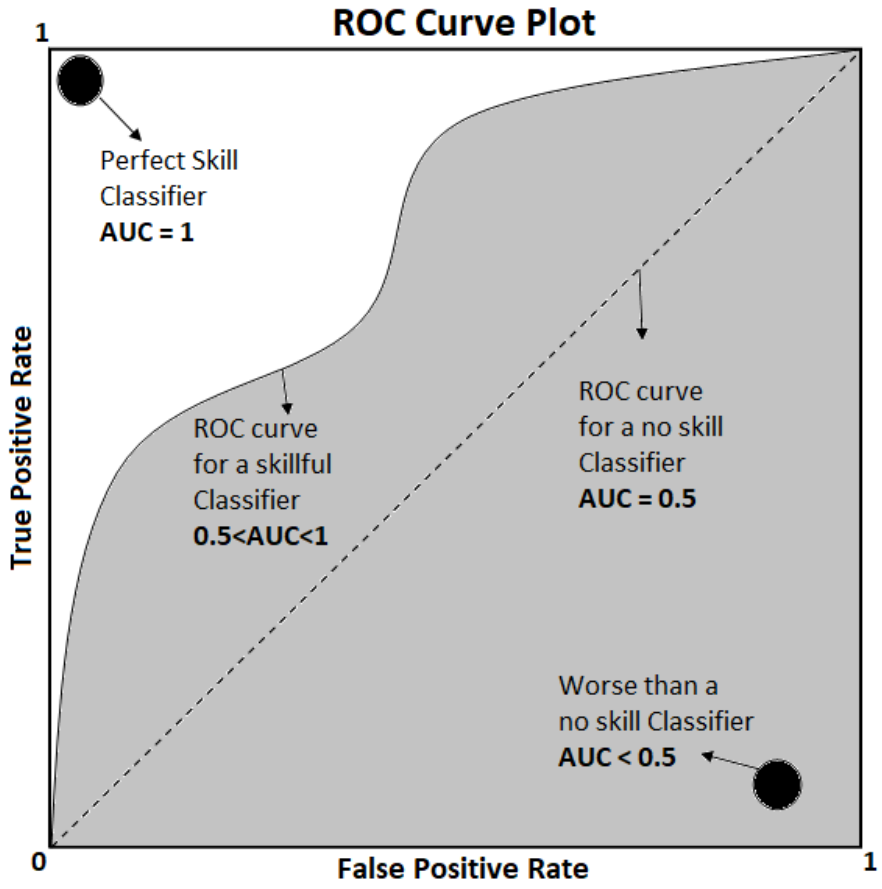
$$True\ Positive\ Rate(TPR) = Reacall = \frac{\sum TP}{\sum TP + \sum FN}$$

Taux de faux positifs (False Positive Rate, FPR) : Le FPR peut être calculé à l'aide de la formule ci-dessous. Plus petit le taux de faux positifs signifie que tous les points négatifs sont classés correctement.

$$False\ Positive\ Rate(FPR) = \frac{\sum FP}{\sum FP + \sum TN}$$

3.1.4. ROC-AUC (Receiver Operating Characteristic - Area Under Curve)

- La courbe ROC est une métrique utilisée pour visualiser les performances d'un problème de classification binaire.
- ROC est une courbe de probabilité, c'est un tracé du taux de faux positifs par rapport au taux de vrais positifs.
- Le score AUC est la zone sous la courbe ROC (**Area Under Curve**).
- Plus le score AUC est élevé, meilleures sont les performances du classificateur.
- Les valeurs du score AUC peuvent varier de **0 à 1**.



Score AUC <0,5: le classifieur est pire qu'un classifieur sans compétence

AUC = 0,5: il ne s'agit pas d'un classificateur de compétences et ne peut pas classer les classes positives et négatives, il prédit juste au hasard n'importe quelle classe

0,5 <AUC <1: c'est un classificateur habile, il y a plus de chances que ce classificateur puisse classer correctement les classes positives et négatives, car un score AUC plus élevé signifie que le FPR est faible et le TPR est élevé

AUC = 1: c'est un classificateur de compétences parfait et il peut classer tous les points de classe négatifs et positifs avec précision

3.1.5. Log Loss (Logarithmic Loss)

- **Log Loss**, ou **Logarithmic Loss**, est une métrique utilisée pour évaluer les modèles de classification probabiliste. Elle mesure la précision des prédictions en tenant compte des probabilités prédites plutôt que des classes finales.

$$\boldsymbol{Log\ Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{i,j}\log(p_{i,j})$$

- $y_{i,j}$: Indicateur binaire (1 si l'exemple *i* appartient à la classe *j*, sinon 0).
- $p_{i,j}$: Probabilité prédite pour la classe *j* sur l'exemple *i*.
- *N* : Nombre total d'exemples.
- *K*: Nombre de Classes

- Pour une classification binaire (**K=2**) :

$$\boldsymbol{Log\ Loss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i\log(p_i) + (1-y_i)\log(1-p_i)]$$

3.2. Métriques pour la Régression

3.2.1. Mean Absolute Error (MAE) :

Mesure la moyenne des écarts absolus entre les valeurs prédites et réelles

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|$$

3.2.2. Mean Squared Error (MSE) :

MSE: Moyenne des carrés des erreurs (favorise les petites erreurs).

Root Mean Squared Error (RMSE): $RMSE = \sqrt{MSE}$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

3.2.3. R^2 (Coefficient de détermination):

Mesure la proportion de la variance expliquée par le modèle.

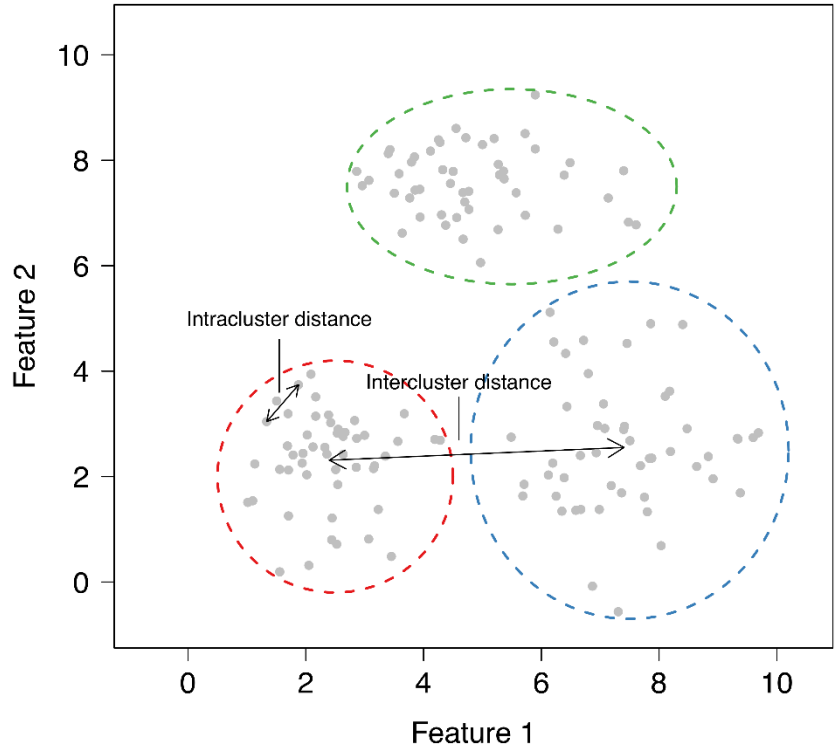
Utile pour évaluer la qualité globale du modèle.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}$$

$\bar{y}_i$ moyenne des valeurs réelles (observées) y_i

3.3. Métriques pour le Clustering

L'objectif du **clustering** est d'identifier des groupes partageant des propriétés similaires.
Les données au sein de chaque groupe doivent être homogènes (minimiser la distance intra-cluster), tandis que chaque groupe doit être suffisamment distinct des autres (maximiser la dissimilarité inter-cluster).



3.3.1. Silhouette Score (S):

Mesure le degré de séparation entre les clusters

$$S = \frac{b - a}{max(a, b)}$$

- ***a*** est la distance moyenne entre un point et les autres points de son cluster.
- ***b*** est la distance moyenne entre un point et les points du cluster le plus proche.
- S proche de 1 = bons clusters, S proche de -1 = mauvais clustering

3.3.2. Davies-Bouldin Index:

- Évalue la compacité et la séparation des clusters.
- Plus la valeur est faible, mieux c'est.

$$DB = \frac{1}{N}\sum_{i=1}^{N} \max_{j \neq i} \frac{\sigma_i + \sigma_j}{d(c_i + c_j)}$$

où σ mesure la dispersion dans un cluster, et $d(c_i + c_j)$ est la distance entre les centres des clusters *i* et *j*.

3. Etapes d'un projet de Machine Learning

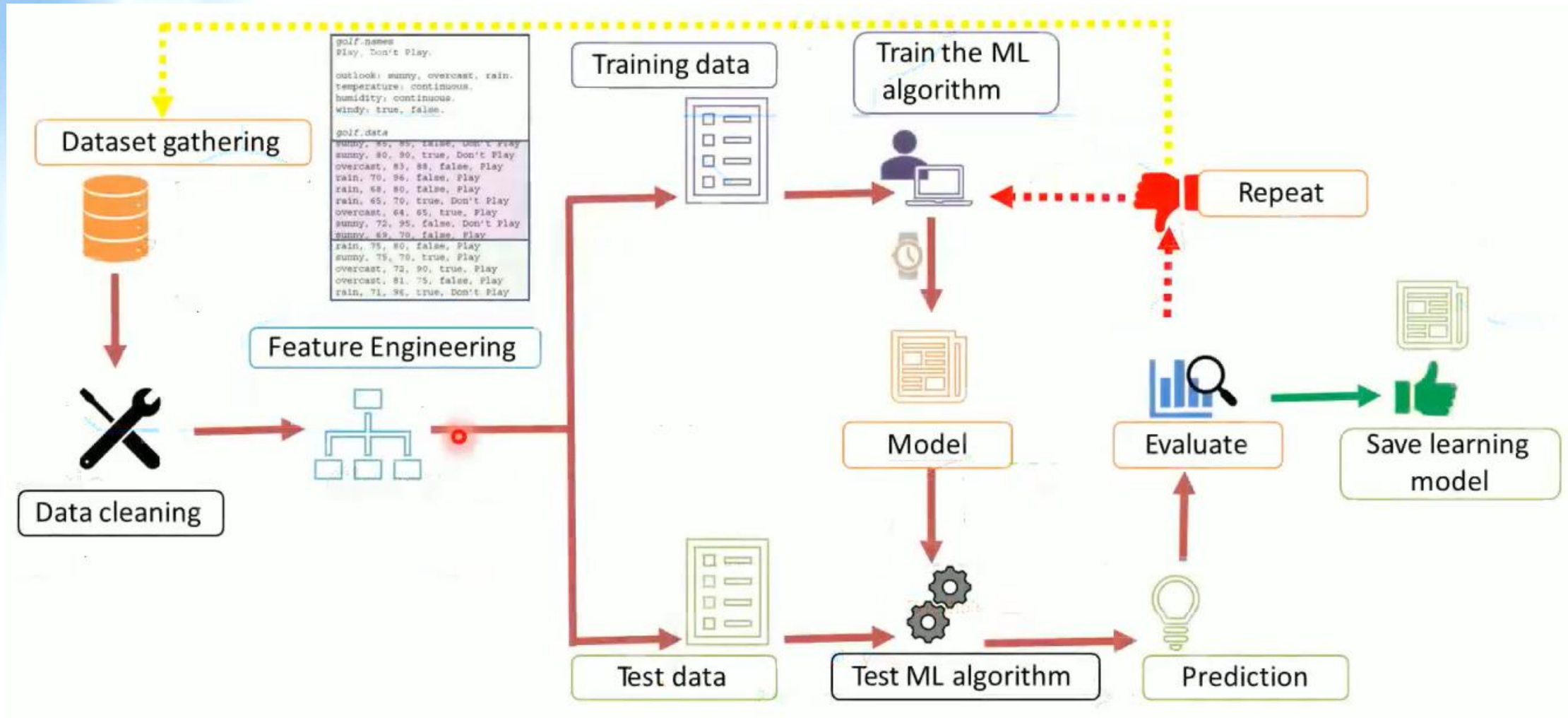
3.1. Définition du problème

- Identifier le problème à résoudre (classification, régression, clustering, etc.).
- Déterminer les objectifs métiers: **Améliorer la satisfaction client,** Réduire les coûts, Augmenter les revenus, Automatiser une tâche, …
- Déterminer les métriques de performance (précision, RMSE, F1-score, etc.).

3.2. Collecte des données

- Identifier les sources de données (bases de données, capteurs, API, web scraping).
- Collecter des données suffisantes en quantité et en qualité nécessaires pour entraîner et évaluer le modèle.

…Etapes d'un projet de ML



3.3. Exploration et préparation des données :

- **Exploration :** Visualiser les données pour détecter des tendances, schémas ou anomalies.
- **Nettoyage :** Traiter les données manquantes, supprimer les doublons, corriger les incohérences.
- **Transformation :**
  - **Normaliser** (voir ci-après) ou standardiser les données: ajuster les valeurs de chaque caractéristique pour qu'elles se situent dans un intervalle spécifique [0, 1] ou [-1, 1]. Cela permet de réduire l'impact des grandes valeurs
  - Encoder les variables catégoriques.: **Exemple**: Rouge → 0; Vert → 1; Bleu → 2

- **Division des données en ensembles** :
  - **Entraînement (exemple : 70 %) :** Pour entraîner le modèle.
  - **Validation (exemple : 15 %) :** Pour ajuster les hyperparamètres.
  - **Test (exemple : 15 %) :** Pour évaluer la performance finale.

3.3. Exploration et préparation des données :

Deux façons pour **normaliser** les données:

1. Normalisation (Min-Max)

La **normalisation Min-Max** transforme les données pour qu'elles soient dans une plage spécifique, typiquement entre 0 et 1

$$Valeur\ normalisée = \frac{X - X_{min}}{X_{max} - X_{min}}$$

2. Standardisation: centrage-réduction (Z-score) :

- Elle est calculée en soustrayant la **moyenne** des données et en divisant par l'**écart-type**.
- Cette méthode transforme les données pour qu'elles aient une **moyenne de 0** et un **écart-type de 1**.

$$Valeur\ normalisée = \frac{X - \mu}{\sigma}$$

Ici, μ est la moyenne et σ est l'écart-type de la variable.

3.4. Choix du modèle et de l'algorithme :

- Sélectionner le modèle adapté au type de problème (régression, classification, clustering, etc.).
- Identifier les algorithmes appropriés (ex. arbres de décision, SVM, réseaux neuronaux).

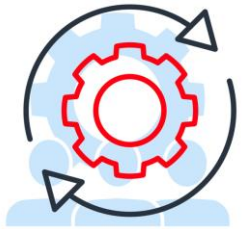
3.5. Entraînement du modèle :

- Former le modèle en utilisant l'ensemble d'entraînement.
- Ajuster les paramètres internes pour minimiser l'erreur.
- Surveiller les performances sur l'ensemble de validation.

3.6. Évaluation du modèle :

- Utiliser l'ensemble de test pour mesurer la performance réelle.
- Calculer des métriques pertinentes (précision, F1-score, RMSE, etc.).
- Vérifier la généralisation pour éviter le sur-apprentissage (overfitting).

3.7. Optimisation:



- Ajuster les hyperparamètres pour améliorer les performances (ex. taux d'apprentissage, profondeur des arbres).
- Tester des techniques avancées (ex. régularisation, bagging, boosting).

3.8. Déploiement :

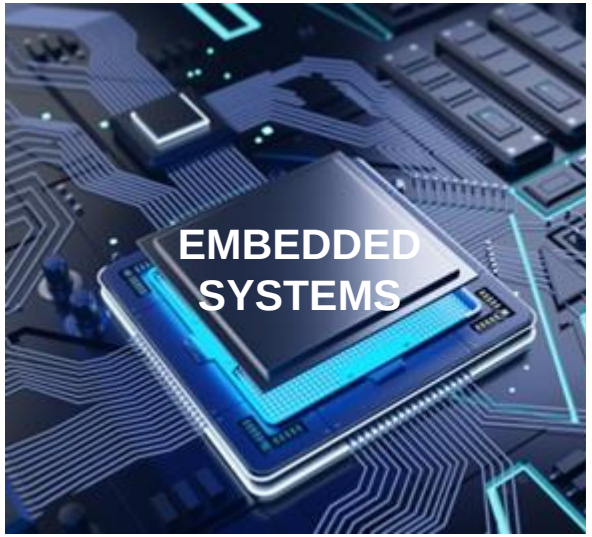
- Intégrer le modèle dans un système opérationnel ou une application.
- Configurer des API ou pipelines pour une utilisation en production.

3.9. Surveillance et maintenance :

- Suivre les performances du modèle en production.
- Réentraîner le modèle avec de nouvelles données si nécessaire.
- Identifier et corriger les dérives des données (data drift).

3.10. Documentation et communication :

- Documenter le projet de manière complète (approches, choix, résultats).
- Partager les résultats et les insights avec les parties prenantes.



FIN !
Questions ?