Chapitre III

Méthodes d’ évaluation

Evaluation d'un modèle de régression linéaire

Coefficient de détermination R2

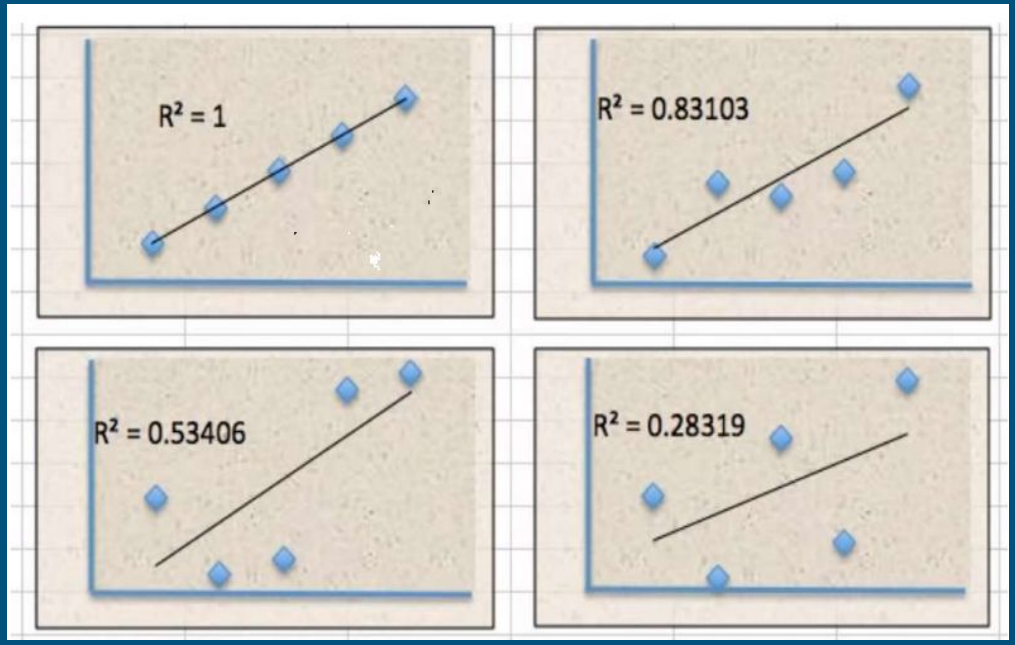
– Un indicateur de la qualité d'une régression linéaire simple

– à quel point l'équation de régression est adaptée pour décrire la distribution des points.

0 ≤ R2 ≤ 1 : plus R2 est proche de 1, plus le modèle semble pertinent

• La valeur 0 : indique un pouvoir de prédiction faible

• La valeur 1 : indique un pouvoir de prédiction fort.



Evaluation d'un modèle de classification

Les métriques de mesures de la performances

- Accuracy : Elle calcule le nombre de prédictions correctes parmi toutes les prédictions.

$$\text{accuracy} = \frac{TrueNegative + TruePositive}{Nombre\ d'observations}$$

- Recall /Sensitivity : Le rappel mesure la capacité d'un modèle à identifier tous les exemples positifs

$$recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

- Précision: mesure la proportion des exemples identifiés comme positifs parmi tous les exemples classés comme positifs.
  ⇒ mesure la capacité du modèle à classifier correctement les exemples positifs, en évitant les faux positifs.

$$précision = \frac{TruePositive}{TruePositive + FalsePositive}$$

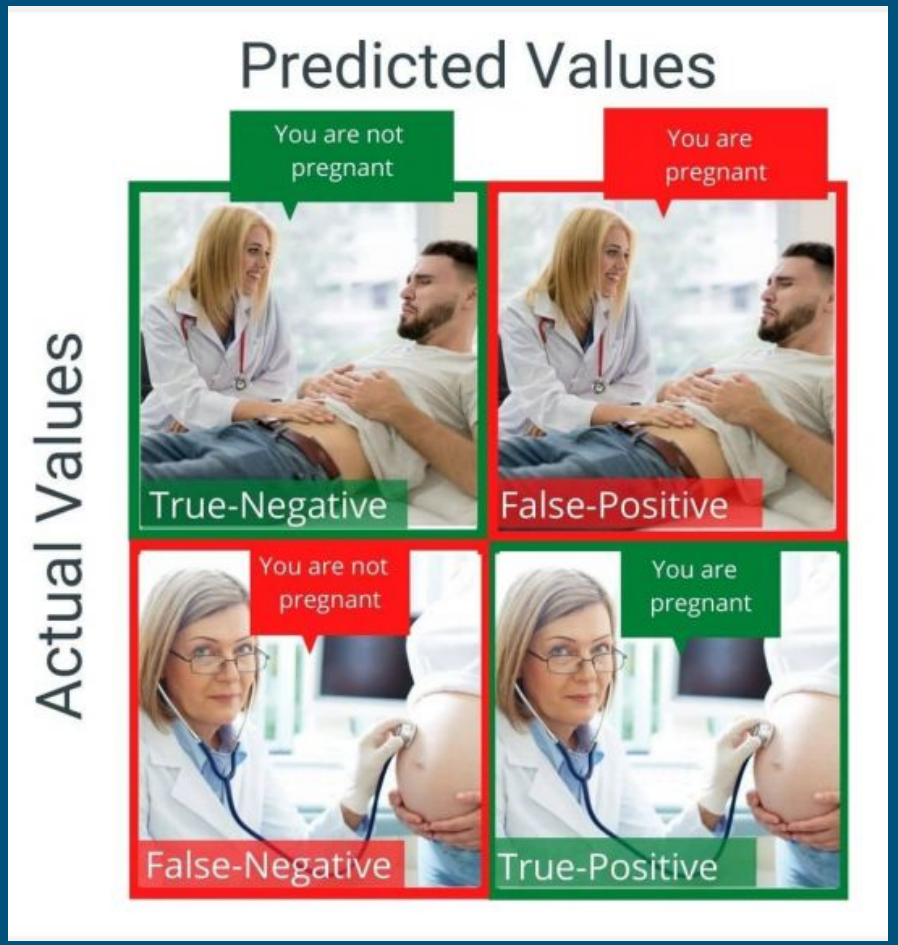
- Le score F1: calculé en utilisant la précision et le rappel.
  ➔ Si l'une des deux mesures (précision ou rappel) est faible, le score F1 sera également faible, même si l'autre mesure est élevée.
  ➔ Le score F1 est compris entre 0 et 1, 1 indique une performan parfaite en termes de précision et de rappel, tandis qu'une valeur indique une performance nulle.

$$\text{F1 score} = 2 \times \frac{recall \times précision}{recall + précision}$$

Les mesures de performances

Exemple : La personne est-elle enceinte ? (classification binaire)

- True Positive : Le modèle prédit enceinte (+) et effectivement la personne est enceinte (+).
- True Negative : Le modèle prédit non enceinte (-) et effectivement la personne n'est pas enceinte (-).
- False Positive : Le modèle prédit enceinte (+) alors que la personne ne l'est pas (-).
- False Negative : Le modèle prédit non nceinte (-) alors qu'en réalité elle l'est (+)



Matrice de Confusion

C’est un matrice carrée de taille n (nombre de classes)

Permet d’avoir une représentation visuelle de la performance d’un classificateur

Les indicateurs de performance sont calculés à partir de quatre nombres qui correspondent aux nombre de Vrais Positifs (VP), le nombre de Faux Positifs (FP), le nombre de Vrais Négatifs (VN), et le nombre de Faux Négatifs (FN).

		Valeur Réelles	
Valeurs prédites		Malade	Non malade
	Malade	TruePositive	FalsePositive
	Non Malade	FalseNegative	TrueNegative

Overfitting / underfitting

Sur-apprentissage (Overfitting)

– Apprentissage par cœur : incapacité de généralisation
– L'erreur d'apprentissage est très faible vs erreur phase de test est élevée

Causes

- Le modèle est trop complexe
- Un très grand nombre de paramètres par rapport à la quantité de données
- Le modèle apprend les particularités (par ex. le bruit) des données

Solutions

Limiter la complexité du modèle, diversifier les données d'entraînement

Sous-Apprentissage (underfitting)

- La fonction d'apprentissage n'est pas assez riche pour pouvoir décrire la diversité dans les données
- Complexité de modèle insuffisante,

Causes

-Un nombre de paramètres trop faible par rapport à la complexité des données
- Une quantité de données d'entraînement insuffisante.

Solutions

- Augmenter la complexité du modèle,
- Ajouter des variables d'entrée pertinentes,
- Augmenter le nombre de paramètres du modèle
- Collecter plus de données d'entraînement

Exercice

Supposons que nous ayons un modèle de classification binaire qui prédit si un email
est du spam ou non. Lorsque nous évaluons la performance de ce modèle, nous
pourrions calculer à la fois le rappel et la précision.
Imaginons que nous avons un ensemble de données contenant 100 emails, dont 20
sont du spam et 80 ne le sont pas. Nous appliquons notre modèle de classification et
obtenons les résultats suivants :

- Vrais positifs (spam identifié comme spam) : 80
- Faux positifs (non-spam identifié comme spam) : 10
- Vrais négatifs (non-spam identifié comme non-spam) : 5
- Faux négatifs (spam identifié comme non-spam) : 5

Calculer l'accuracy, le recall, la précision et le F1 score. Interpréter le résultat.

Correction

Accuracy = (Vrais positifs + Vrais négatifs) / Nombre total de messages = (80 + 5) / 100 = 0,85

Précision = Vrais positifs / (Vrais positifs + Faux positifs)

= 80 / (80 + 10) = 0,8889

Rappel = Vrais positifs / (Vrais positifs + Faux négatifs)

= 80 / (80 + 5) = 0,9412

F1 score = 2 * (précision * rappel) / (précision + rappel)

= 2 * (0,8889 * 0,9412) / (0,8889 + 0,9412) = 0,9143

Interprétation des résultats

Recall = 0,94

 le modèle est capable d'identifier 94% des messages de spam dans l'ensemble de données
 une faible probabilité de manquer des messages importants

L'accuracy = 0,85

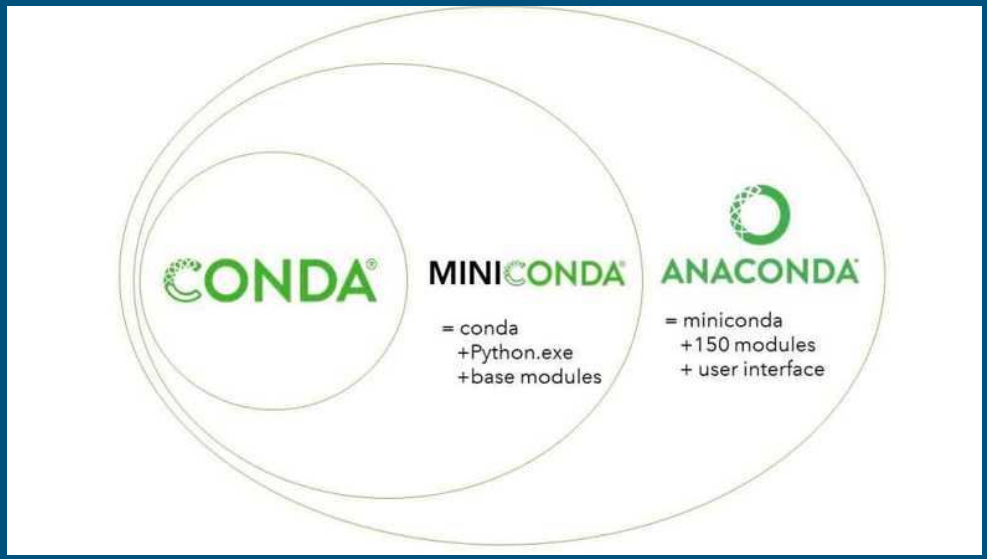
 indique que le modèle a correctement classé 85% des messages
 le modèle a une bonne performance globale (attention mesure trompeuse)

La précision = 0,89

- lorsque le modèle classe un message comme spam, il a 89% de chances que ce message soit réellement du spam.
- faible probabilité de classer à tort un message légitime comme spam.

Le F1 score = 0,91.
 le modèle a un bon équilibre entre la capacité à identifier les messages de spam et la capacité à éviter de classer les messages légi comme spam.

Data Science Python Tools



DATA SCIENCE PYTHON LIBRARIES

NumPy
This is a mathematical library. Has a powerful N-dimensional array object, linear algebra, high-level mathematical functions, etc.

used for data analysis and manipulation. It provides tools for
data cleaning, merging, reshaping,
slicing, and filtering data.

Scikits Learn
used for machine learning tasks such as classification, regression, and clustering. provides a range of algorithms and tools .

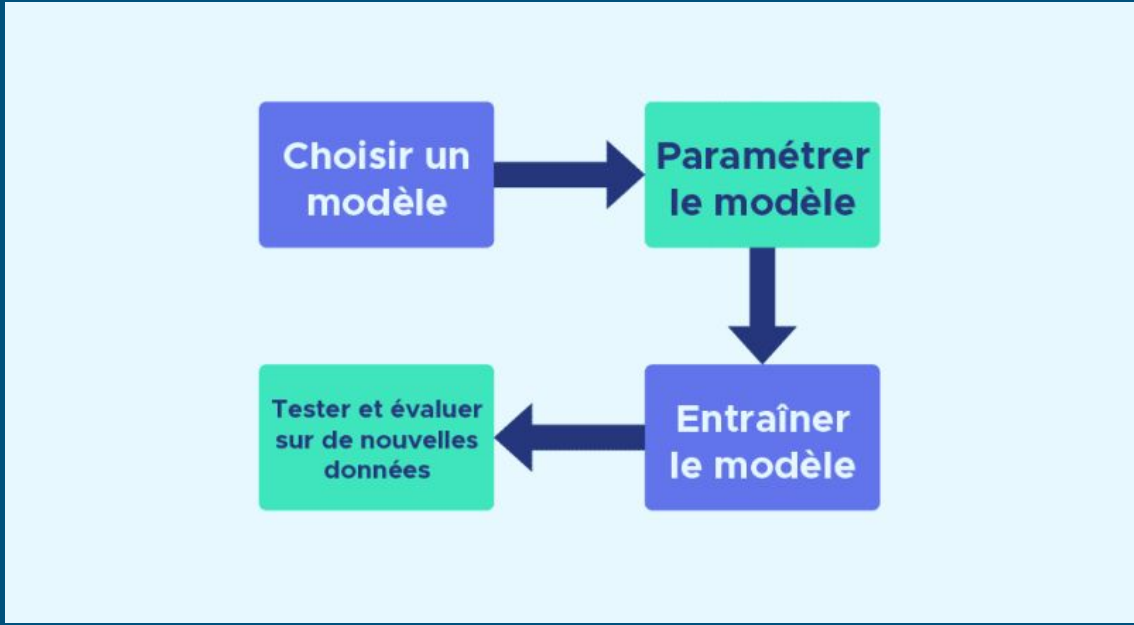
open-source machine learning framework developed by Google that is used to build and train machine learning model

used for creating static, animated, and interactive visualizations of data

for building and training neural networks, and to simplify the process of building and deploying machine learning models

La bibliothèque scikit-learn

C’est une **librairie Python** permet de créer, paramétrer des modèles de machine Learning dans le cadre d’apprentissage supervisé et non supervisé



- Bibliothèque open source de **calcul numérique et de** deep learning compatible avec le langage Python
- Permet de simplifier le processus d'acquisition de données, d'entraînement des modèles de Machine Learning et de génération de prédictions
- Le langage Python offre une API front-end pratique et confortable pour créer des applications à l'aide de ce framework.
- TenforFlow intègre Keras, une API de deep learning de haut niveau qui permet de créer des réseaux de neurones en quelques lignes de code avant de les entrainer et les déployer via TensorFlow

- C'est une API de réseau de neurones écrite en langage Python. Une bibliothèque Open Source, exécutée par-dessus des frameworks tels que Theano et TensorFlow
- **Keras** est modulaire, rapide , simple d'utilisation et intuitive pour créer des modèles de Deep Learning.v
- Permet de créer des **"layers "** pour les RNA à architectures complexes.

Activités 1 , 2 & 3

Objectifs :

1- préparation des données

2- Régression linéaire

3- classification multi-classes K-nn

Utilisation des librairies :

- pandas
- sklearn
- matplotlib

Pour aller plus loin encore : utiliser le modèle XGBoost et SVM dans la 3ème activité

Fin

Merci Pour Votre Attention