REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Larbi Ben M'Hidi Oum El Bouaghi

Cours traitement du langage naturel

CHAPITRE 5 : MODÈLES DE LANGUE ET CLASSIFICATION DE TEXTE

Dr.Belhouchette K

Contenu du Chapitre :

01

Modèles de langue n-gram

02

Classification de texte

03

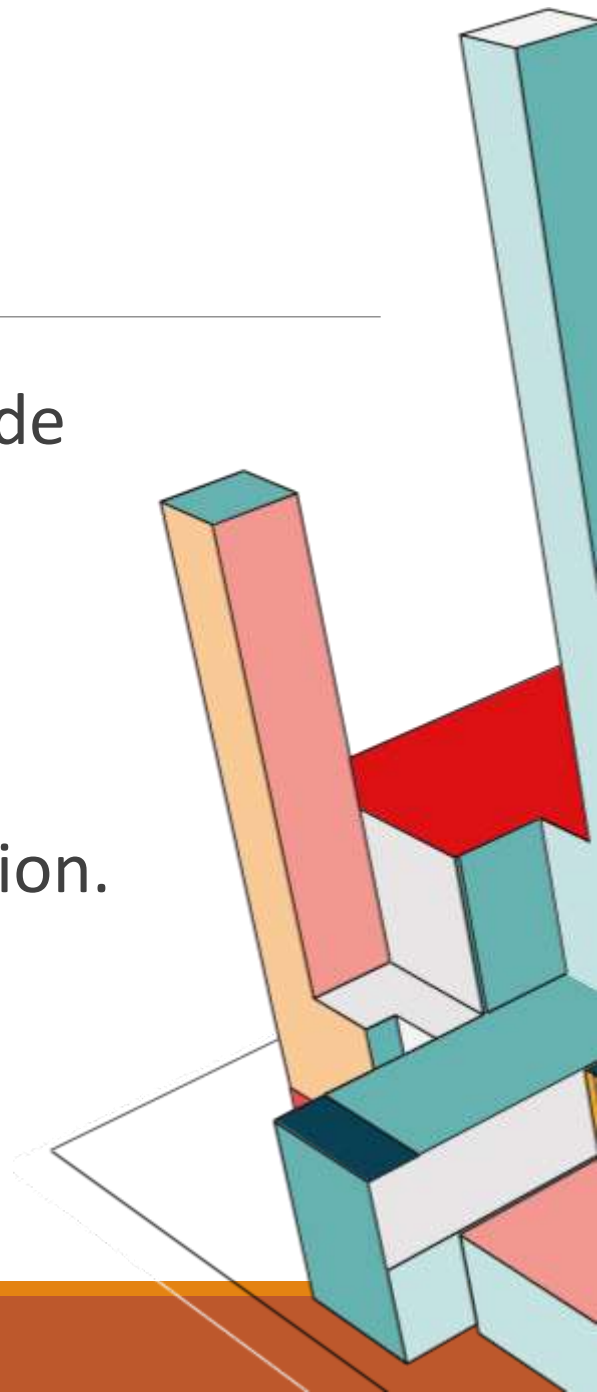
Évaluation des performances des modèles de classification

Introduction

- Un **modèle de langue** estime la probabilité d'une séquence de mots :

$$P(w_1, w_2, ..., w_n) = P(w_1)P(w_2|w_1)...P(w_n|w_1, ..., w_{n-1})$$

- Mais ce calcul est coûteux → n-gram propose une simplification.



01

Modèles de langue n-gram

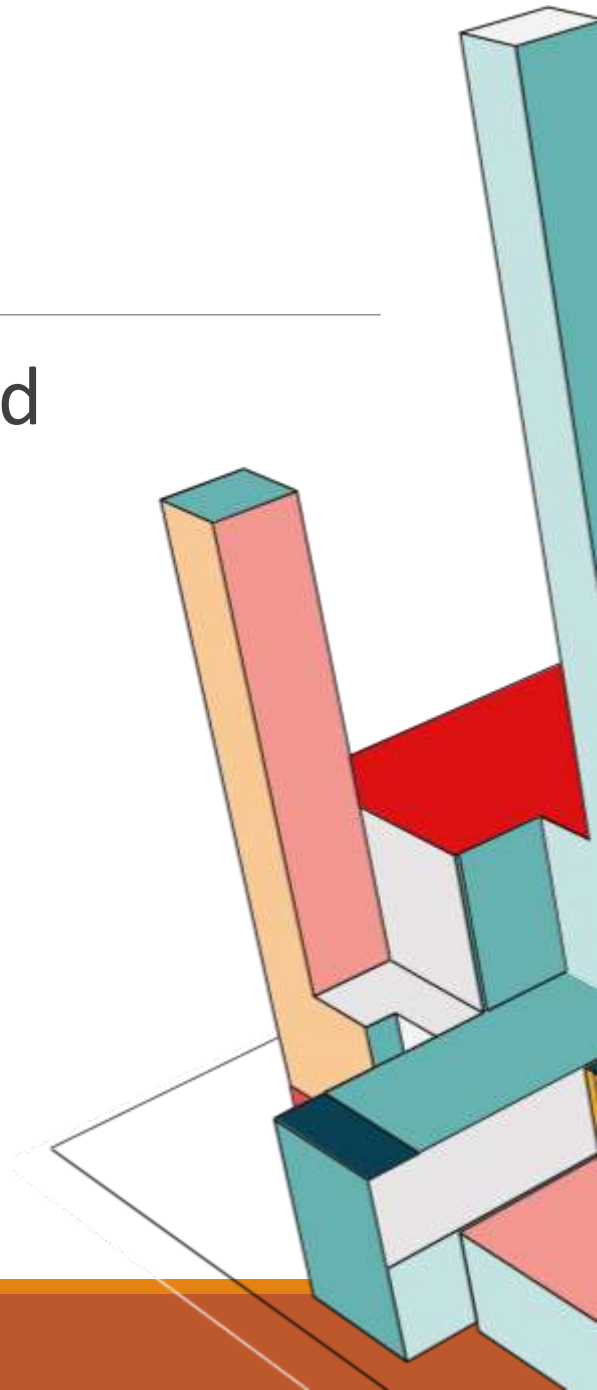
Modèles de langue n-gram

- Un modèle **n-gram** suppose que chaque mot dépend uniquement des **n-1 mots précédents**.

Unigram (n=1) : $P(w_i)$

Bigram (n=2) : $P(w_i|w_{i-1})$

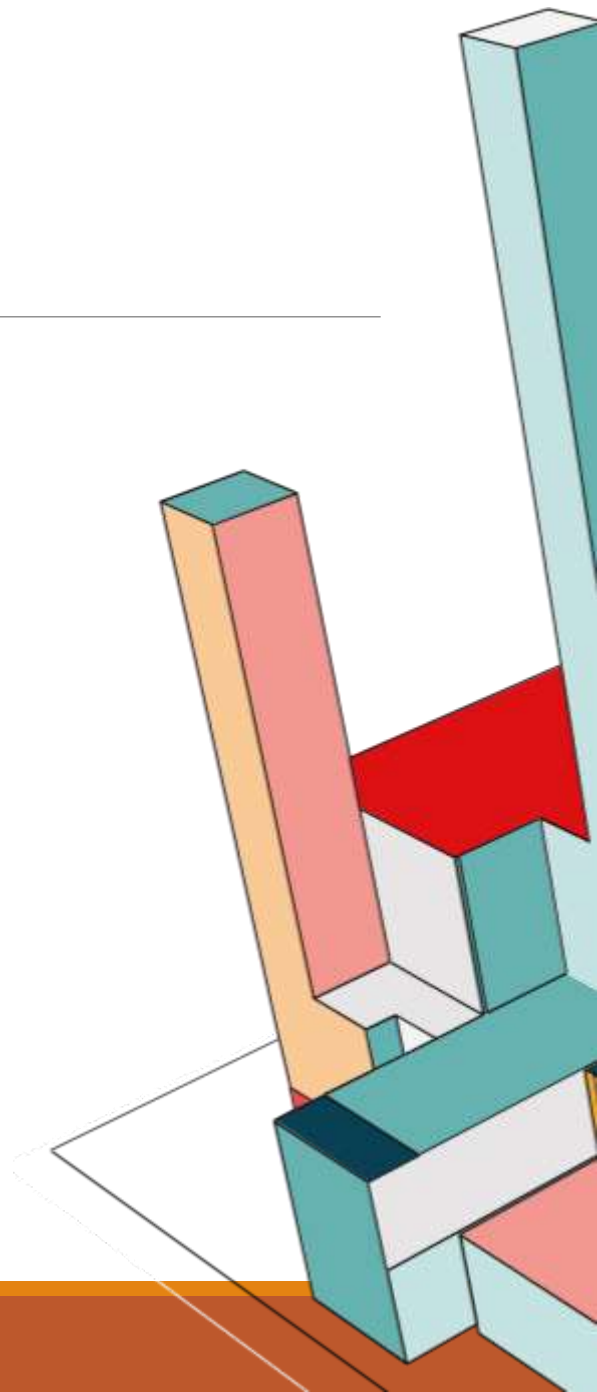
Trigram (n=3) : $P(w_i|w_{i-2}, w_{i-1})$



Calcul de probabilité dans les modèles n-gram

Pour un modèle n-gramme, la probabilité d'une séquence de mots est estimée en fonction des n-1 derniers mots :

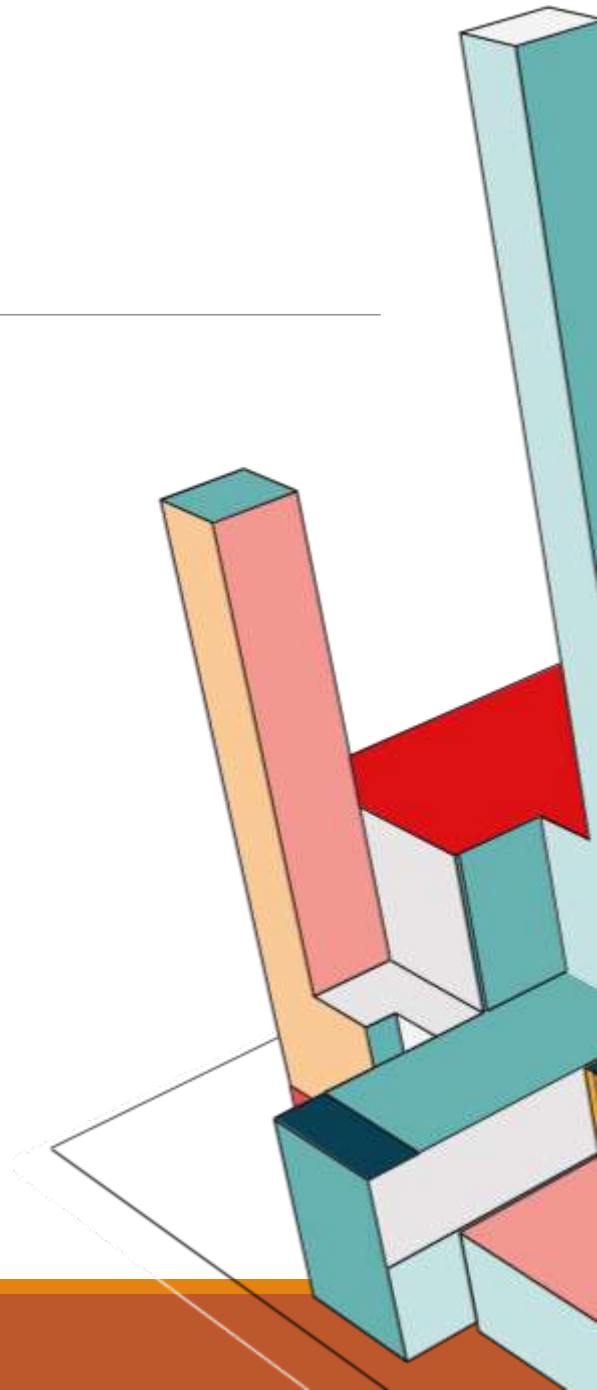
$$P(w_1, w_2, \ldots, w_N) = \prod_{i=1}^{N} P(w_i | w_{i-1}, w_{i-2}, \ldots, w_{i-n+1})$$



Exemple illustratif

Supposons que nous avons le corpus suivant, constitué de 3 phrases liées à l'informatique :

1."les algorithmes sont efficaces"
2."les structures de données sont essentielles"
3."les algorithmes sont utilisés"

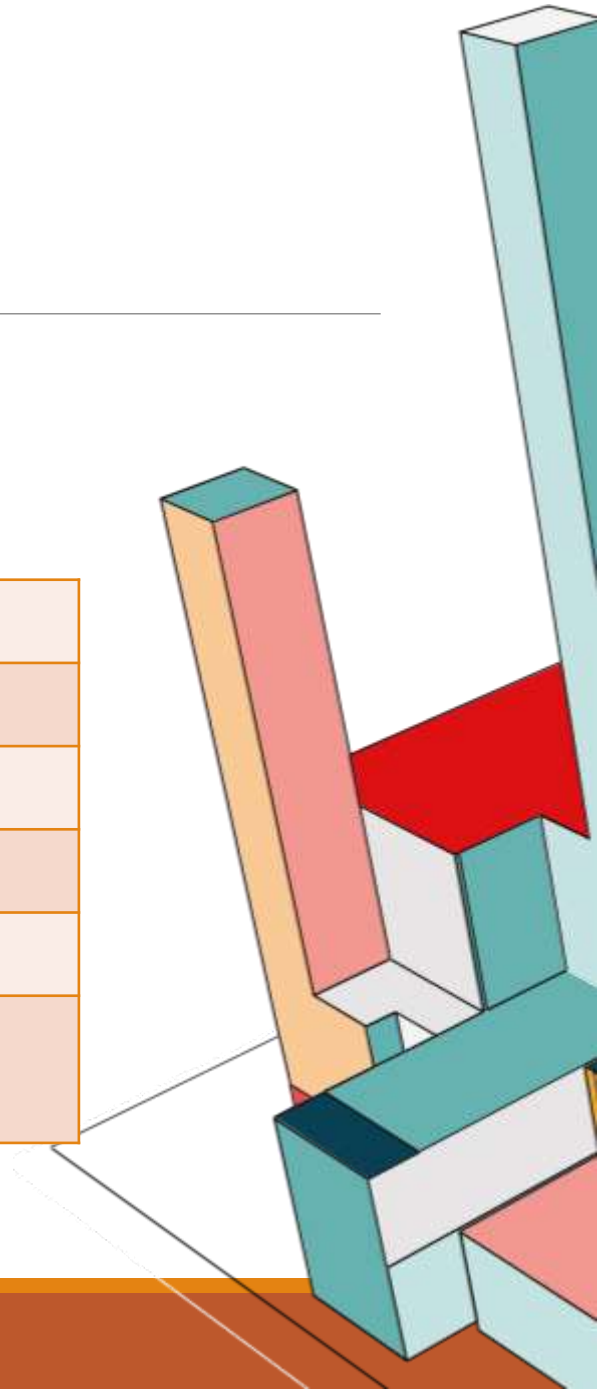


Exemple illustratif

Étape 1 : Construction du modèle Bigramme

Les bigrammes présents dans notre corpus sont :

("les", "algorithmes")	("algorithmes", "sont")
("sont", "efficaces")	("les", "structures")
("structures", "de")	("de", "données")
("données", "sont")	("sont", "essentielles")
("les", "algorithmes")	("algorithmes", "sont")
("sont", "utilisés")	

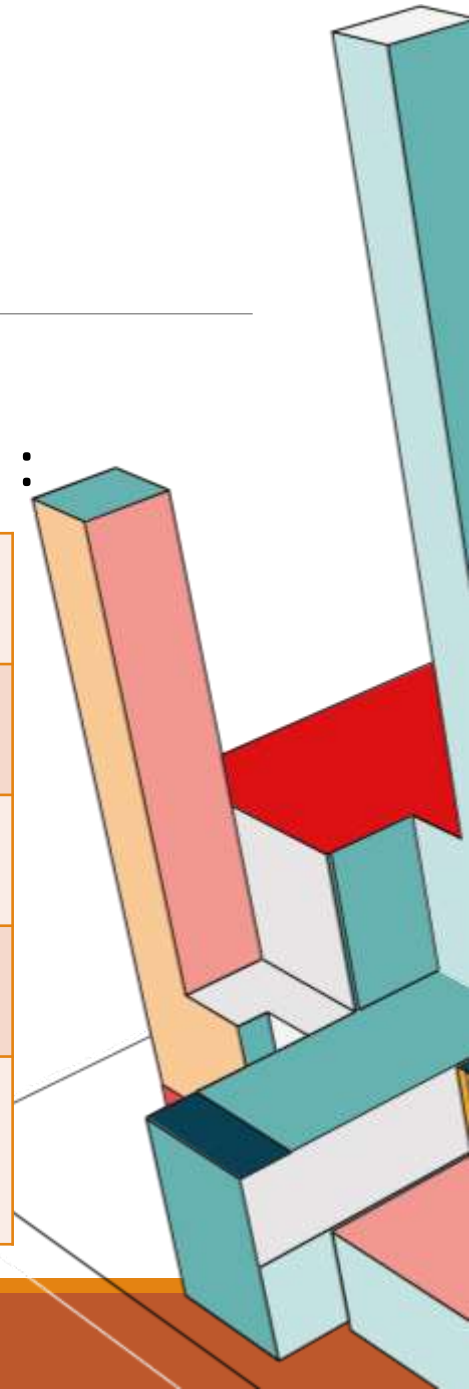


Exemple illustratif

Étape 2 : Calcul des fréquences de chaque bigramme

Nous comptons les occurrences de chaque bigramme dans le corpus :

"les« → "algorithmes"	**2 fois**	**"algorithmes"→"sont"**	**2 fois**
"sont"→ "efficaces"	1 fois	**"les", "structures"**	1 fois
"structures"→ "de"	1 fois	**"de", "données"**	1 fois
"données"→ "sont"	1 fois	**"sont"→ "essentielles"**	1 fois
"sont"→"utilisés"	1 fois		

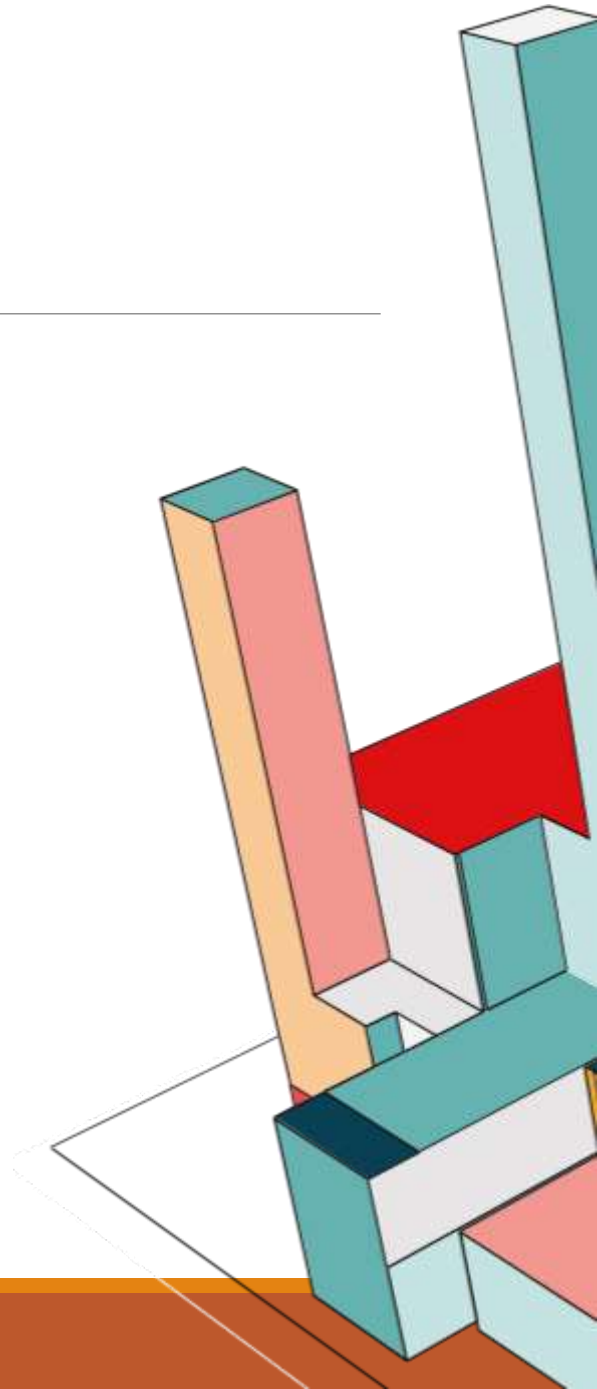


Exemple illustratif

Étape 3 : Calcul des probabilités des bigrammes

La probabilité d'un bigramme $P(w_i|w_{i-1})$ est donnée par :

$$P(w_i|w_{i-1}) = \frac{\text{compte de } (w_{i-1}, w_i)}{\text{compte de } w_{i-1}}$$



Exemple illustratif

Calcul des probabilités pour chaque bigramme :

"les« → "algorithmes"	**P(algorithmes|les)=2/3 (car "les" apparait 3 fois dans le corpus : "les algorithmes", "les structures", "les algorithmes")**
"sont"→ « algorithmes"	P(sont|algorithmes)=1(car "algorithmes" apparait 2 fois, chaque fois suivi de "sont")
« sont"→ « efficaces"	P(efficaces|sont)=1/3(car "sont" apparait 3 fois dans le corpus : "sont efficaces", "sont essentielles", "sont utilisés"
« les"→ "structures"	P(structures|les)=1/3(car "les" apparait 3 fois)
"structures"→« de"	P(de|structures)=1(car "structures" apparait une seule fois, suivi de "de")

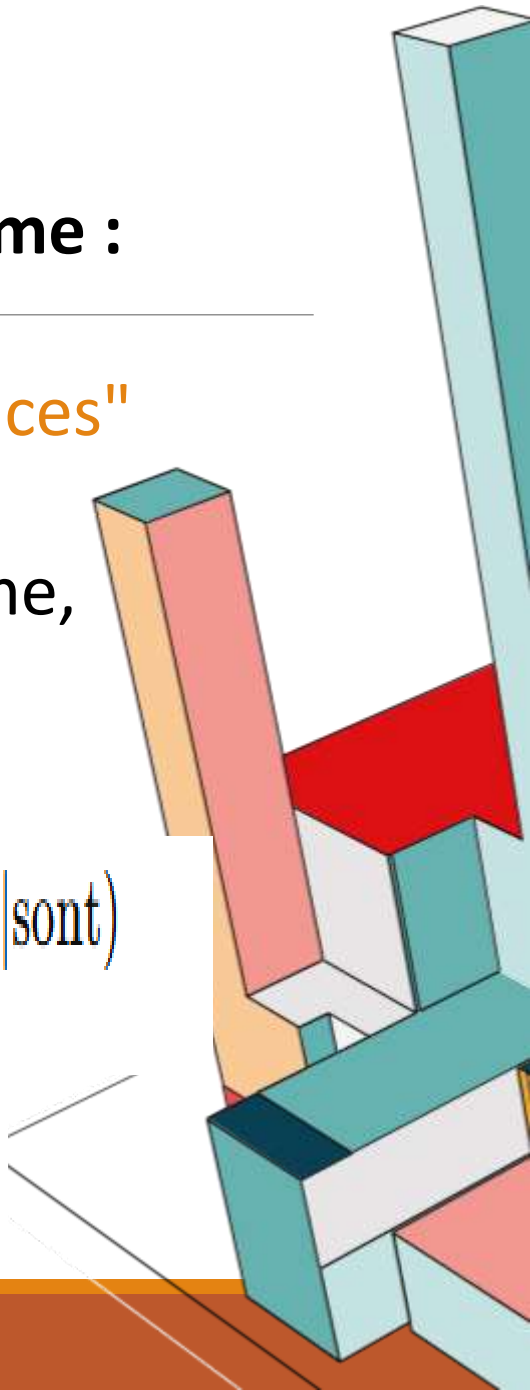
Exemple illustratif **Calcul des probabilités pour chaque bigramme :**

Probabilités totales pour la phrase "les algorithmes sont efficaces"

Si nous voulons calculer la probabilité de la phrase "les algorithmes sont efficaces" en utilisant notre modèle bigramme, cela revient à multiplier les probabilités des bigrammes qui la composent :

$$P(\text{"les algorithmes sont efficaces"}) = P(\text{algorithmes}|\text{les}) \times P(\text{sont}|\text{algorithmes}) \times P(\text{efficaces}|\text{sont})$$

$$P(\text{"les algorithmes sont efficaces"}) = \frac{2}{3} \times 1 \times \frac{1}{3} = \frac{2}{9}$$



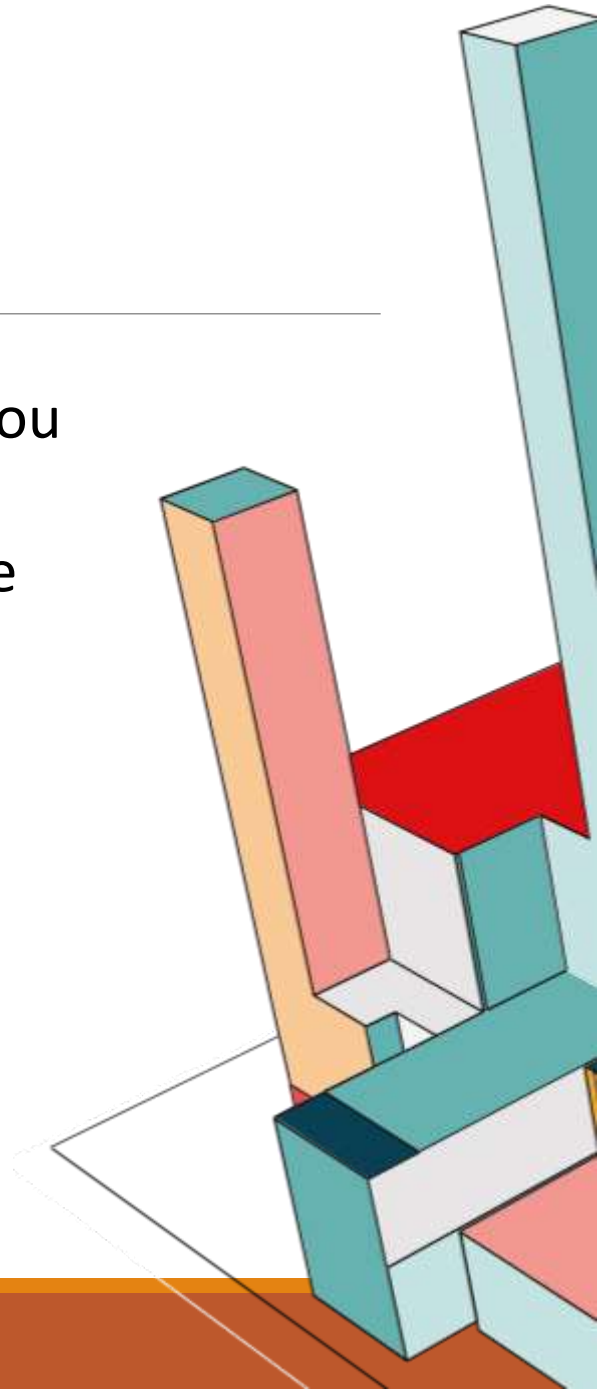
02

Modèles Classification de texte

Définition

- La classification de texte est une tâche de TLN consistant à assigner une ou plusieurs étiquettes à un texte en fonction de son contenu.
- CAD apprendre une **fonction f : Texte → Classe(s)** à partir d'un ensemble d'exemples étiquetés.

Par exemple le texte : Ce produit est excellent ⟶ Classe : POSITIF

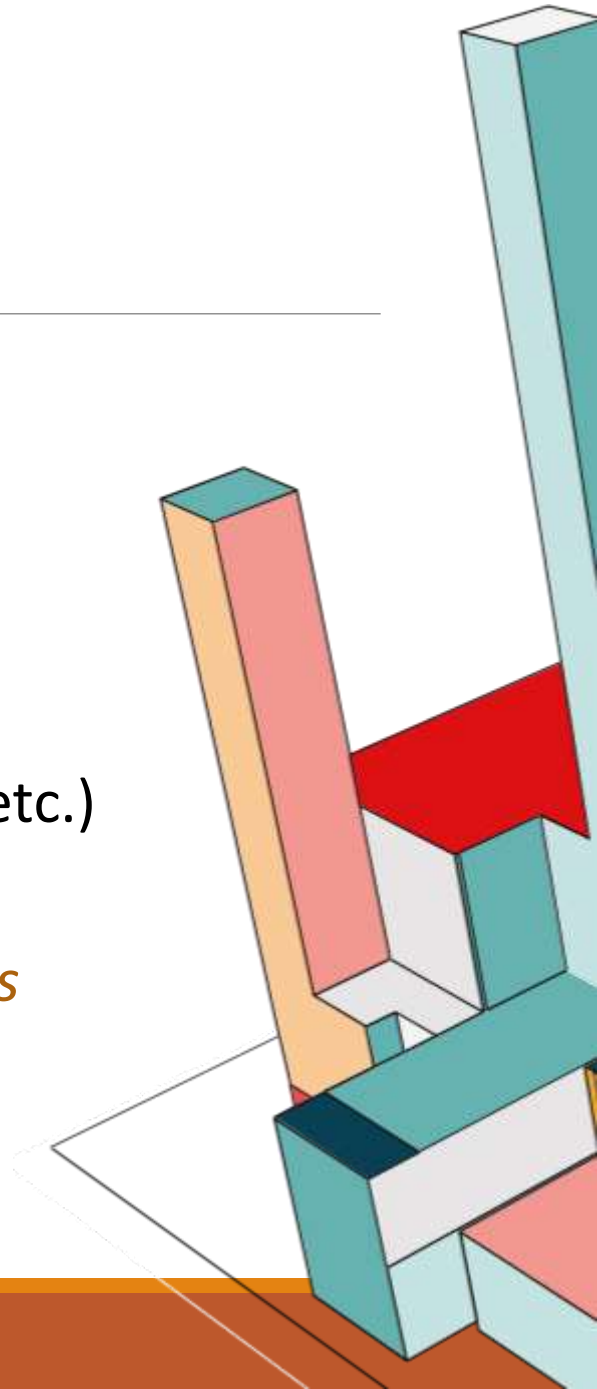


Définition

- Elle est utilisée dans plusieurs applications :

  - ✓Filtrage d'e-mails (spam vs non spam)
  - ✓Analyse de sentiments (positif, négatif, neutre)
  - ✓Catégorisation d'articles (politique, sport, santé, technologie, etc.)

*NB : Les algorithmes ont besoin que le texte soit représenté sous forme **vectorielle** (BoW, TF-IDF, embeddings) pour être exploitable.*

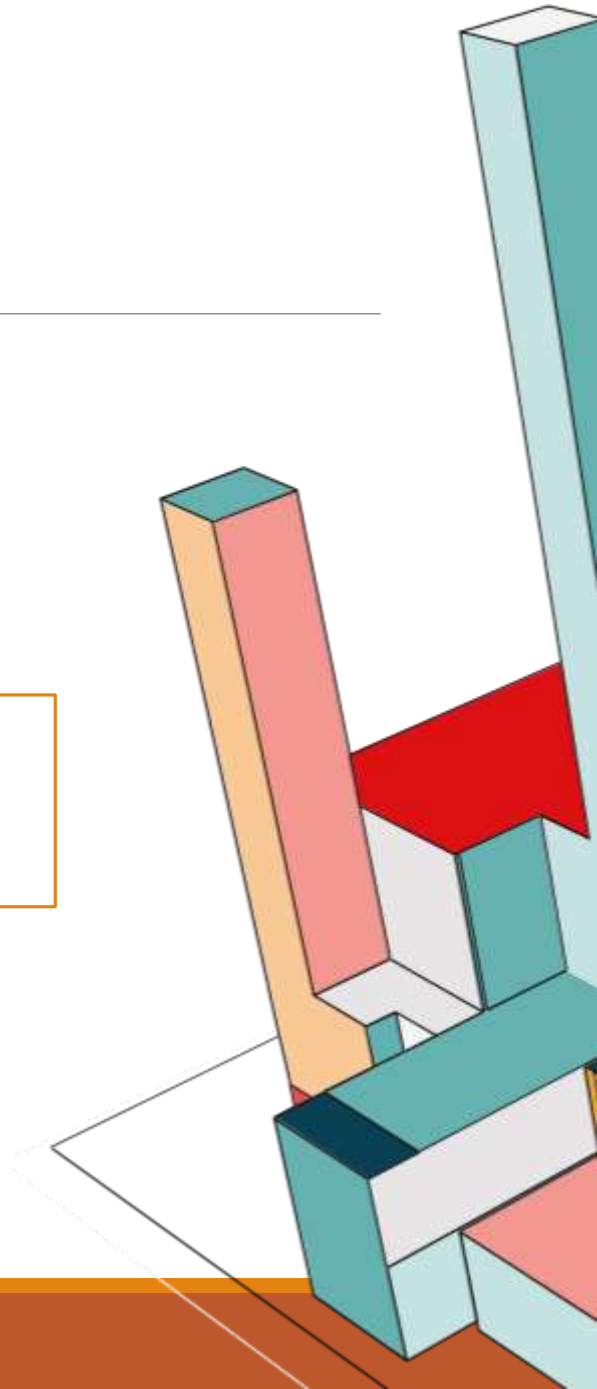


Classification du texte

Approches de classification

Méthodes basées sur des modèles statistiques

Méthodes basées sur l'apprentissage profond



Approches de classificatio

Méthodes basées sur des modèles statistiques

1. Bayes Naïf

- Suppose que les mots sont indépendants conditionnellement à la classe du texte

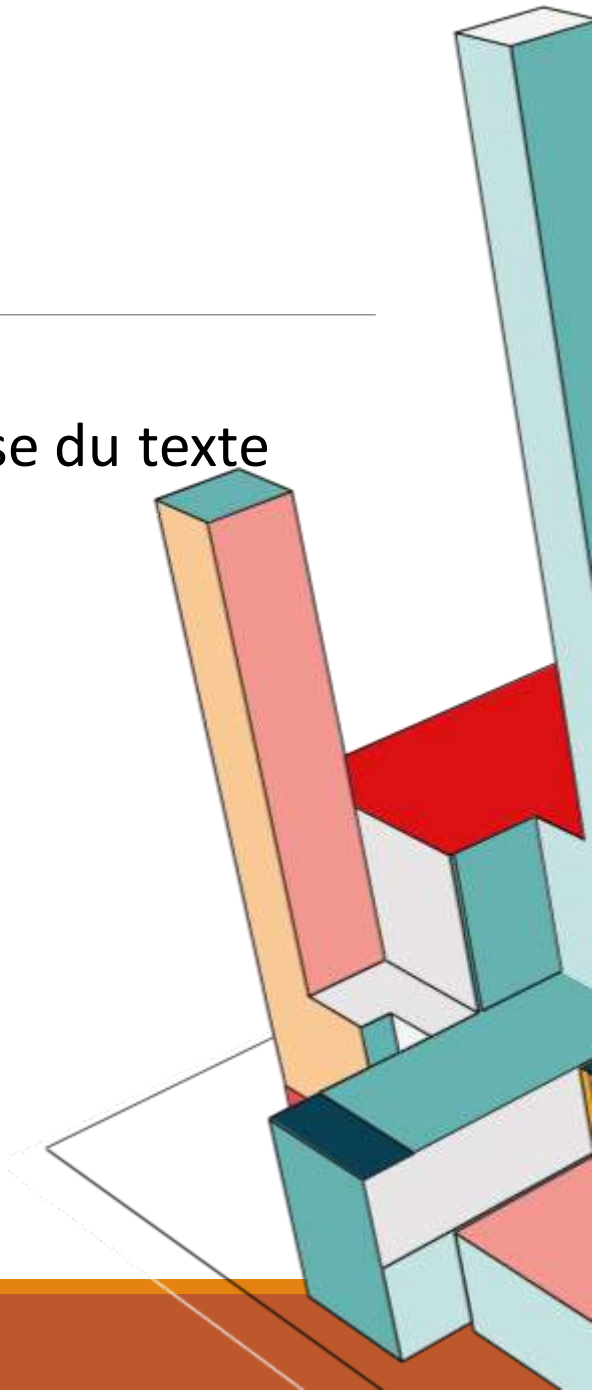
- Probabilité d'une classe C donnée un document d

$$P(c|\vec{x}) \propto P(c) \prod_{i=1}^{n} P(x_i|c)$$

$P(c)$: probabilité a priori de la classe

$P(x_i|c)$: probabilité d'observer le mot x_i dans la classe c

- Rapide et efficace pour les grandes bases de données textuelles.

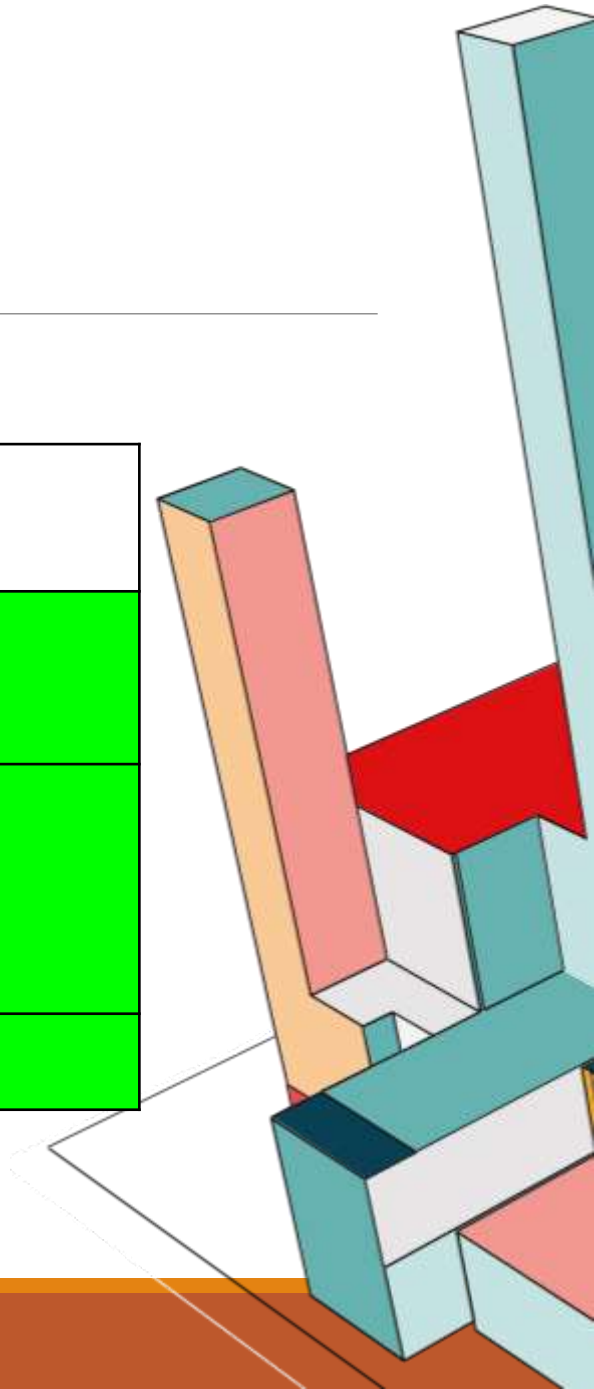


Méthodes basées sur des modèles statistiques

Approches de classificatio

1. Bayes Naïf

Avantages	Inconvénients	Idéal pour
Rapide à entraîner	*Hypothèse forte d'indépendance*	*Spam/ham*
Peu sensible aux données rares	*Performance limitée sur certains types de texte*	*Catégorisation de documents simples*
Interprétable		

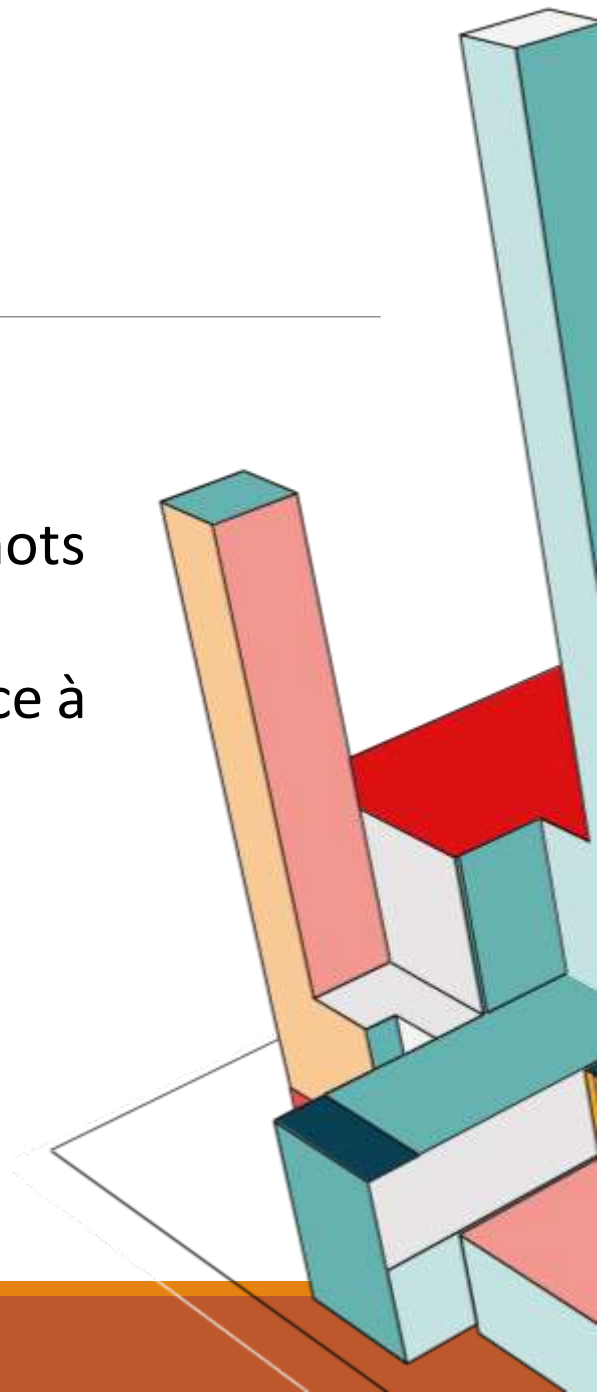


Méthodes basées sur des modèles statistiques

Approches de classificatio

2. Régression logistique

- Modèle linéaire qui prédit la probabilité d'une classe en fonction des mots présents
- Utilise une fonction sigmoïde pour estimer la probabilité d'appartenance à une classe



Méthodes basées sur des modèles statistiques

Approches de classificatio

2. Régression logistique

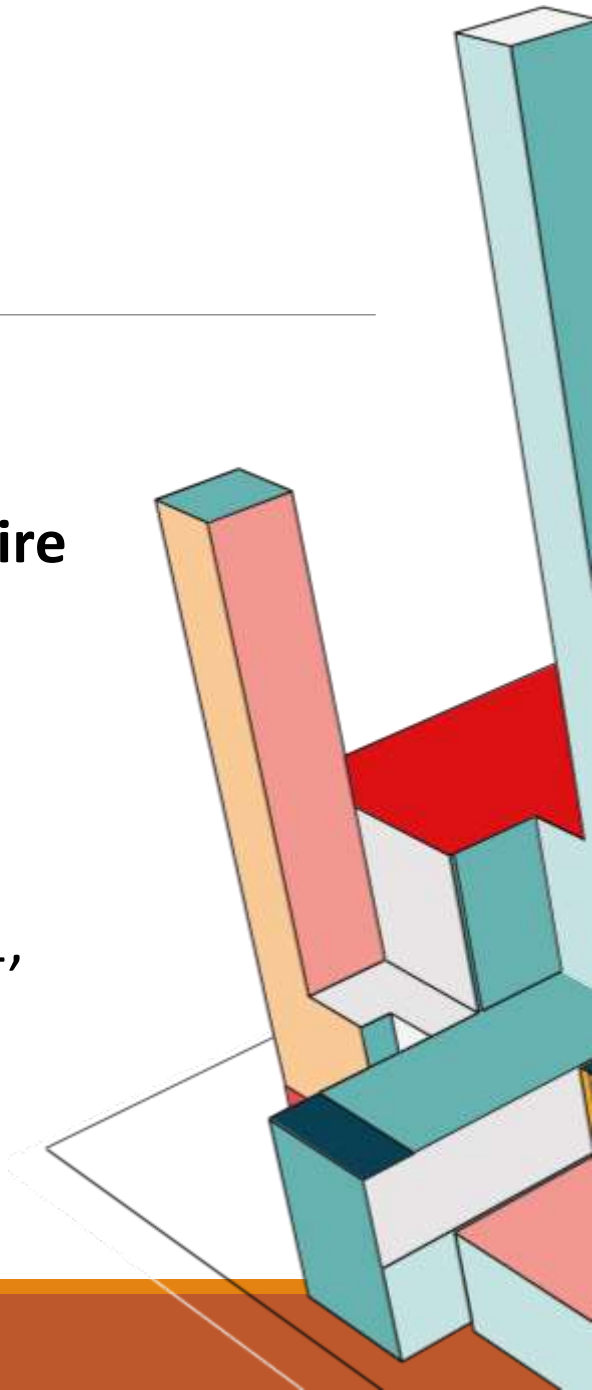
Soit un document représenté par un vecteur $\vec{x}=[x_1,x_2,...,x_n]$
et un vecteur de poids $\vec{w}=[w_1,w_2,...,w_n]$On calcule une **combinaison linéaire pondérée** :

$$z = \vec{w}^T\vec{x} + b = \sum_{i=1}^{n} w_i x_i + b$$

- Puis, on applique la fonction sigmoïde : $\sigma(z) = \frac{1}{1+e^{-z}}$

Qui transforme n'importe quelle valeur réelle z en une valeur entre 0 et 1, interprétée comme une probabilité :

$$P(y=1|\vec{x}) = \sigma(\vec{w}^T\vec{x} + b)$$



Approches de classificatio

Méthodes basées sur des modèles statistiques

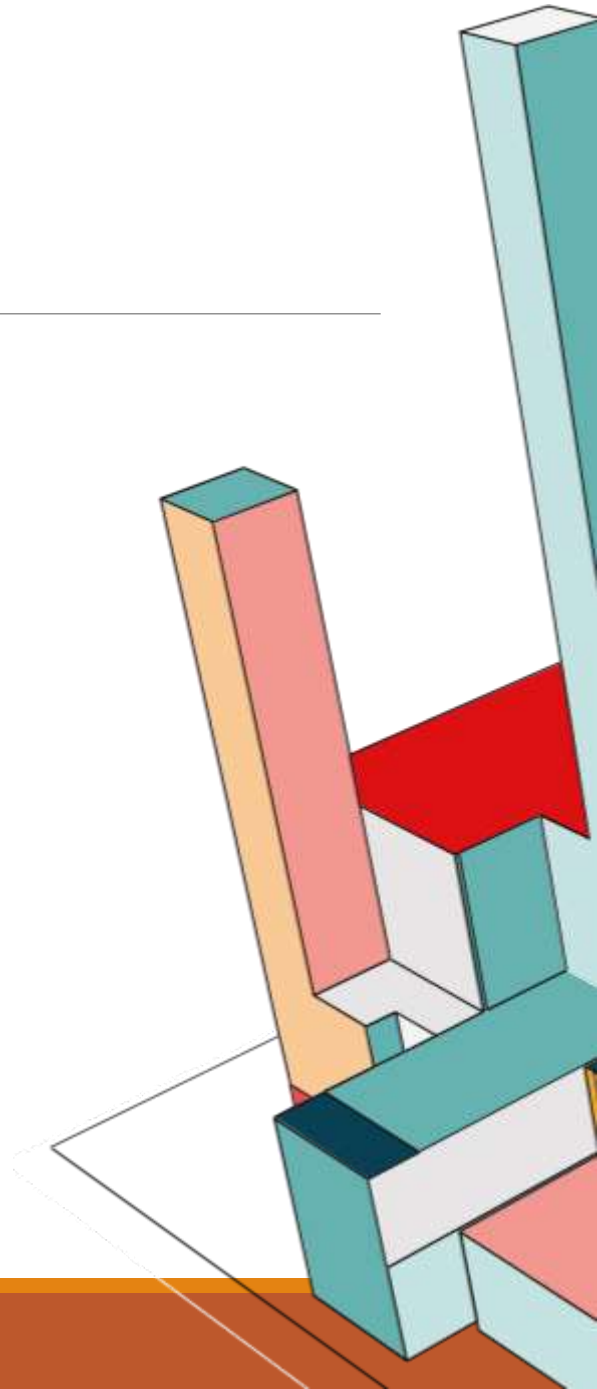
2. Régression logistique

Interprétation

1. Si σ(z)≈1 , le document est très probablement de la classe POSITIF
2. Si σ(z)≈0 , il est probablement NÉGATIF

On utilise un seuil (par défaut 0,5) pour prendre la décision finale :

- Si σ(z)≥0,5→ prédire classe 1
- Sinon → prédire classe 0



Approches de classificatio

Méthodes basées sur des modèles statistiques

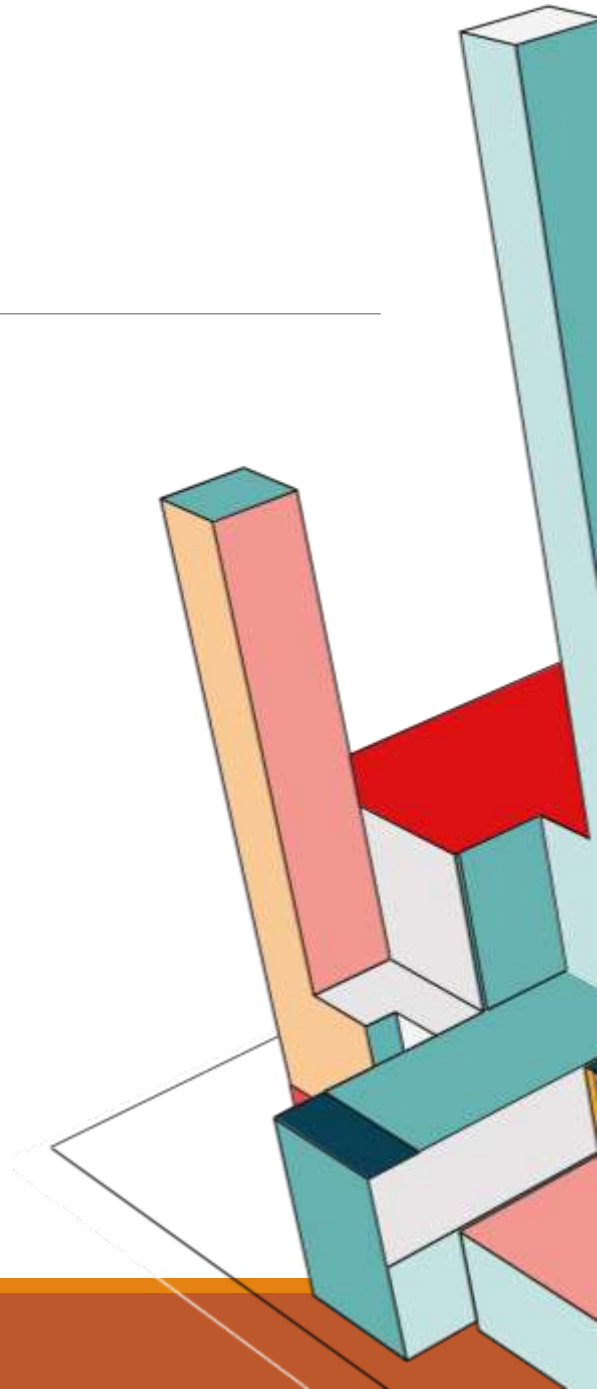
2. Régression logistique

Exemple illustratif

Supposons que chaque document est représenté par 2 caractéristiques :
x1 = nombre de fois que le mot "excellent" apparaît
x2= nombre de fois que le mot "mauvais" apparaît

Soit un document avec x=[3,0], et un modèle entraîné avec : w=[0.8,−1.2], b=−0.5
Alors :
z=0.8·3+(−1.2)·0−0.5=1.9 et $\sigma(1.9) = \frac{1}{1+e^{-1.9}} \approx 0.87$

Probabilité POSITIF = **87 %** → prédiction : **POSITIVE**



Méthodes basées sur des modèles statistiques

Approches de classificatio

3. SVM (Support Vector Machine)

SVM est un algorithme de classification supervisée qui cherche à trouver l'hyperplan qui sépare au mieux les classes tout en maximisant la marge entre les points des deux classes.

Exemple :
Dans un espace 2D, un hyperplan est une droite qui sépare les exemples positifs des négatifs.

Approches de classificatio

Méthodes basées sur des modèles statistiques

3. SVM (Support Vector Machine)

SVM est un algorithme de classification supervisée qui cherche à trouver l'hyperplan qui sépare au mieux les classes tout en maximisant la marge entre les points des deux classes.

Exemple :
Dans un espace 2D, un hyperplan est une droite qui sépare les exemples positifs des négatifs.

Approches de classificatio

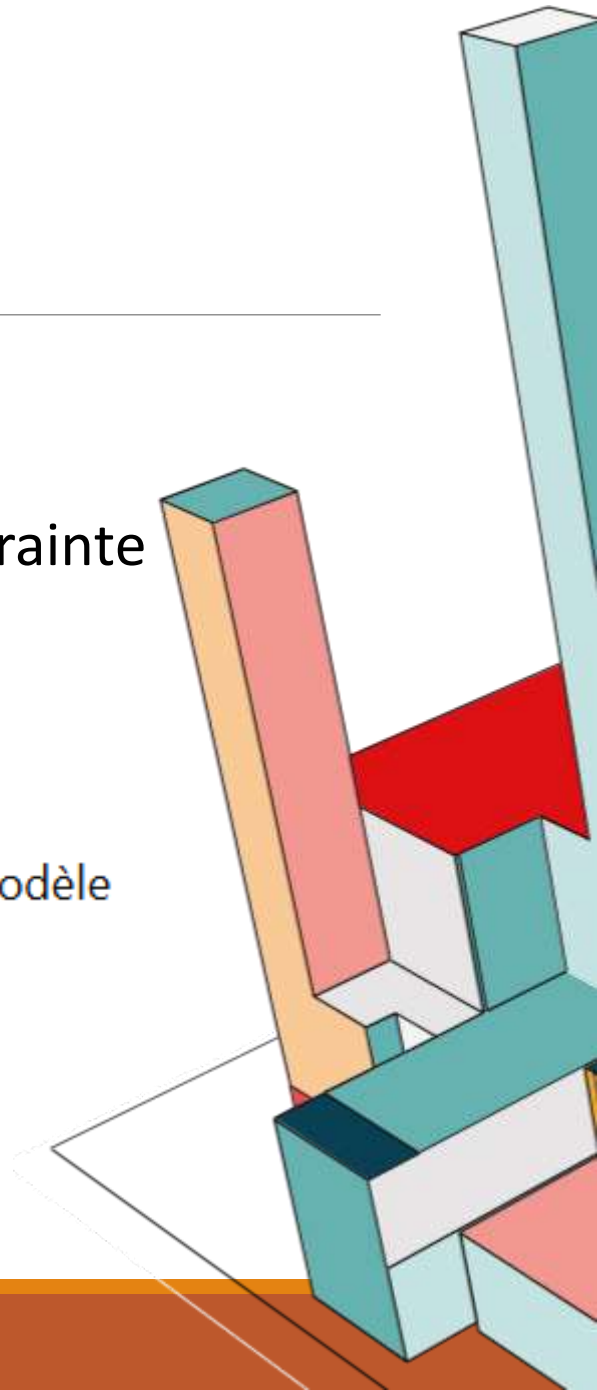
Méthodes basées sur des modèles statistiques

3. SVM (Support Vector Machine)

Le problème est formulé comme une **optimisation quadratique** sous contrainte

$$\min_{\vec{w},b} \frac{1}{2}||\vec{w}||^2 \quad \text{sous la contrainte} \quad y_i(\vec{w}^T x_i + b) \geq 1$$

où : x_i est un exemple, $y_i \in \{-1, +1\}$ est sa classe, $\vec{w}$ est le vecteur de poids du modèle



Approches de classificatio

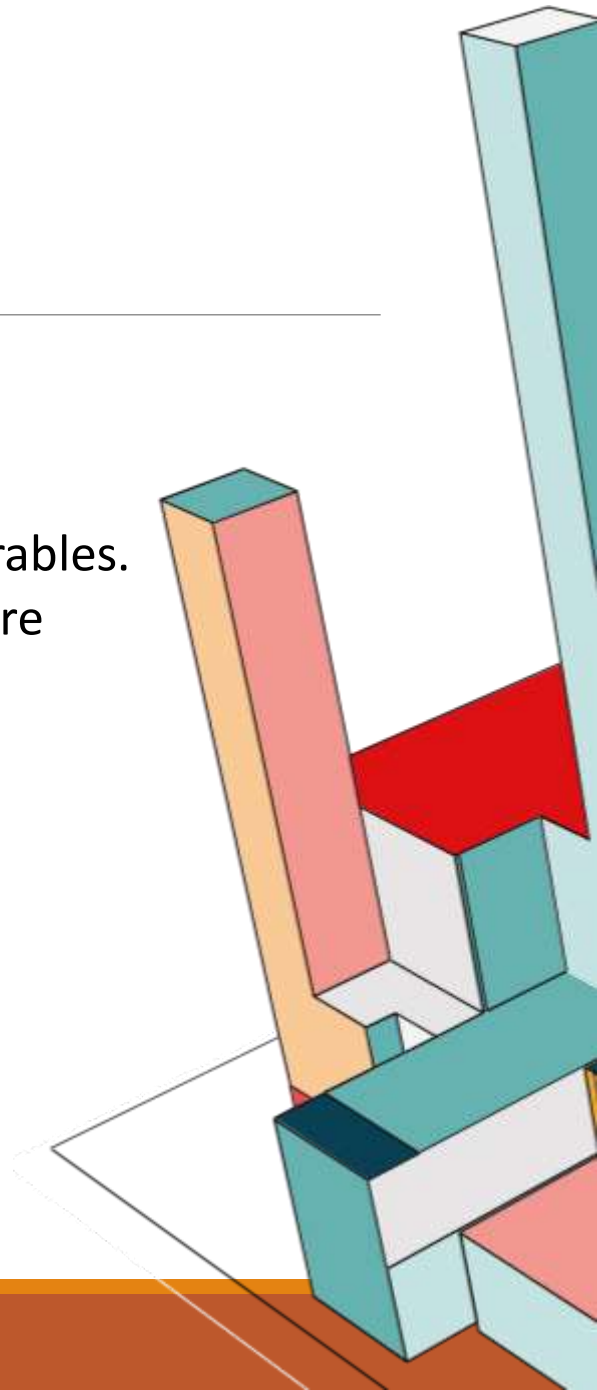
Méthodes basées sur des modèles statistiques

3. SVM (Support Vector Machine)

Dans la réalité (et surtout en texte), les données ne sont pas toujours linéairement séparables.

→ On introduit donc des variables de relâchement et un paramètre C pour permettre quelques erreurs (soft margin SVM):

Ce paramètre C contrôle le **compromis entre marge large et erreur de classification**



Classification du texte

Approches de classificatio

Méthodes basées sur des modèles statistiques

3. SVM (Support Vector Machine)

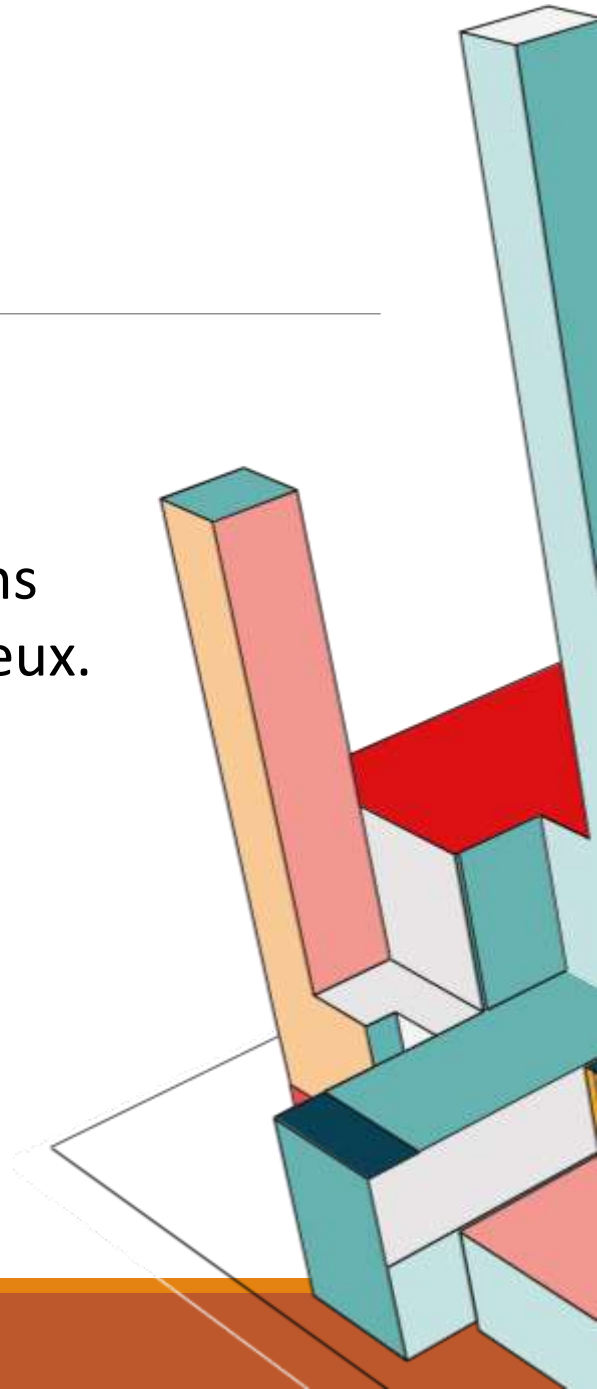
Avantages	Inconvénients	Cas d'usage typiques en NLP
➢ Très efficace dans les espaces de grande dimension ➢ Fonctionne bien même avec peu d'exemples ➢ Robuste au surapprentissage ➢ Peut être utilisé avec différents noyaux (linéaire, RBF, polynomial), mais le noyau linéaire suffit pour les textes	➢ Plus lent à entraîner que Naïve Bayes ou régression logistique ➢ Moins interprétable ➢ Paramétrage (choix de C, du noyau, etc.) peut être délicat	➢ Détection de spam ➢ Classification de sentiments ➢ Catégorisation de documents ➢ Détection de discours haineux ou de fake news

Approches de classificatio

Méthodes basées sur des modèles statistiques

4. k-Nearest Neighbors (k-NN)

- Classer un document en regardant les k documents les plus proches dans l'ensemble d'entraînement et en prenant la majorité des classes parmi eux.



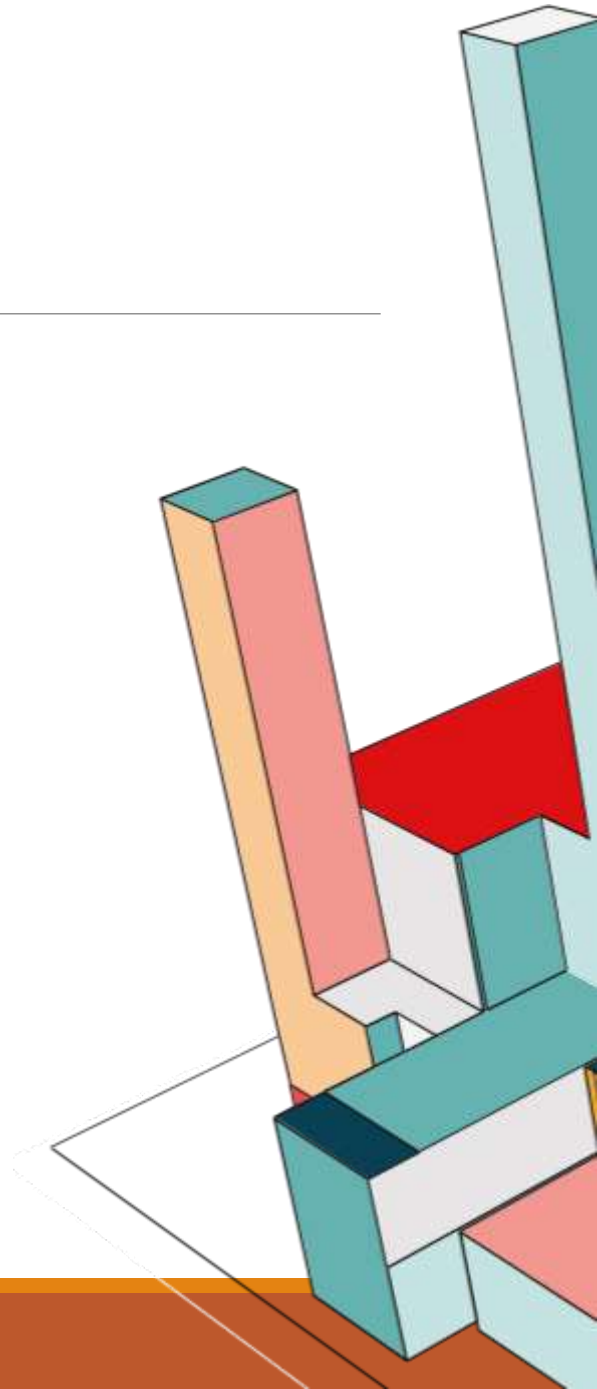
Méthodes basées sur des modèles statistiques

Approches de classificatio

4. k-Nearest Neighbors (k-NN)

Pour un nouveau document à classer :

- On calcule la distance entre ce document et tous les documents du jeu d'entraînement
- On sélectionne les k plus proches voisins (selon la distance choisie)
- On affecte la classe majoritaire parmi ces voisins au document



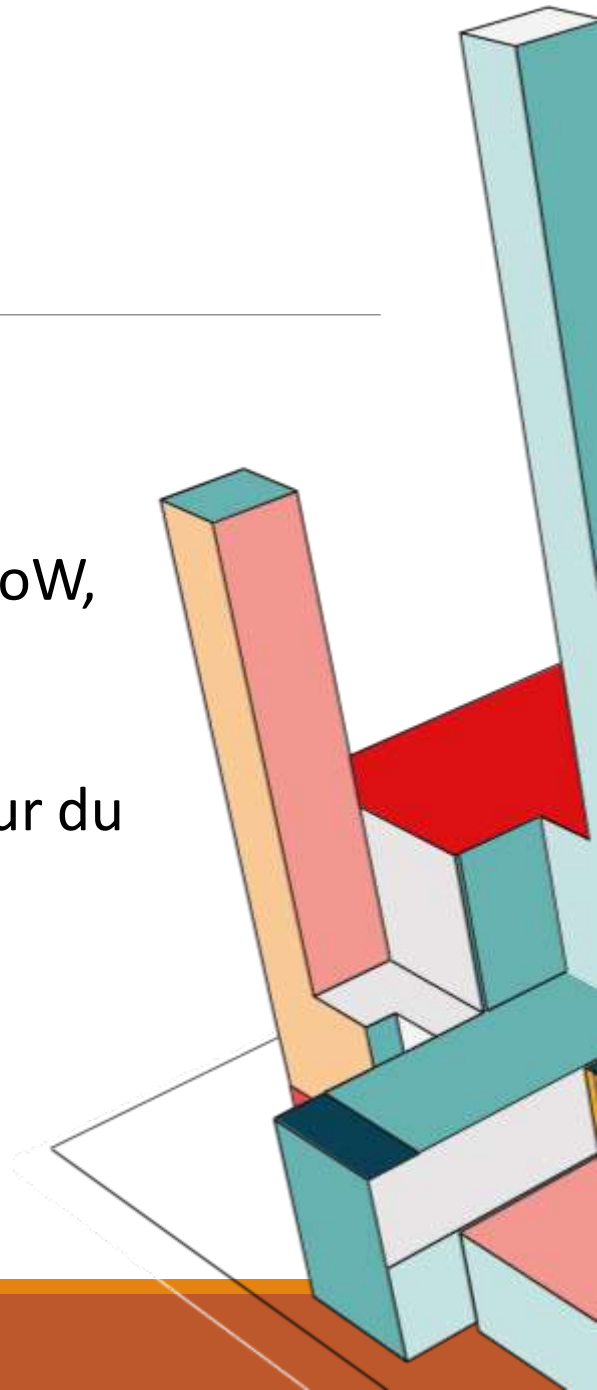
Méthodes basées sur des modèles statistiques

Approches de classificatio

4. k-Nearest Neighbors (k-NN)

Le choix de la mesure de similarité est crucial. Pour les textes vectorisés (BoW, TF-IDF), les plus utilisées sont :

- Distance euclidienne : rarement utilisée en NLP car sensible à la longueur du texte
- Distance cosinus : mesure l'angle entre deux vecteurs → plus robuste



Classification du texte

Approches de classificatio

Méthodes basées sur des modèles statistiques

4. k-Nearest Neighbors (k-NN)

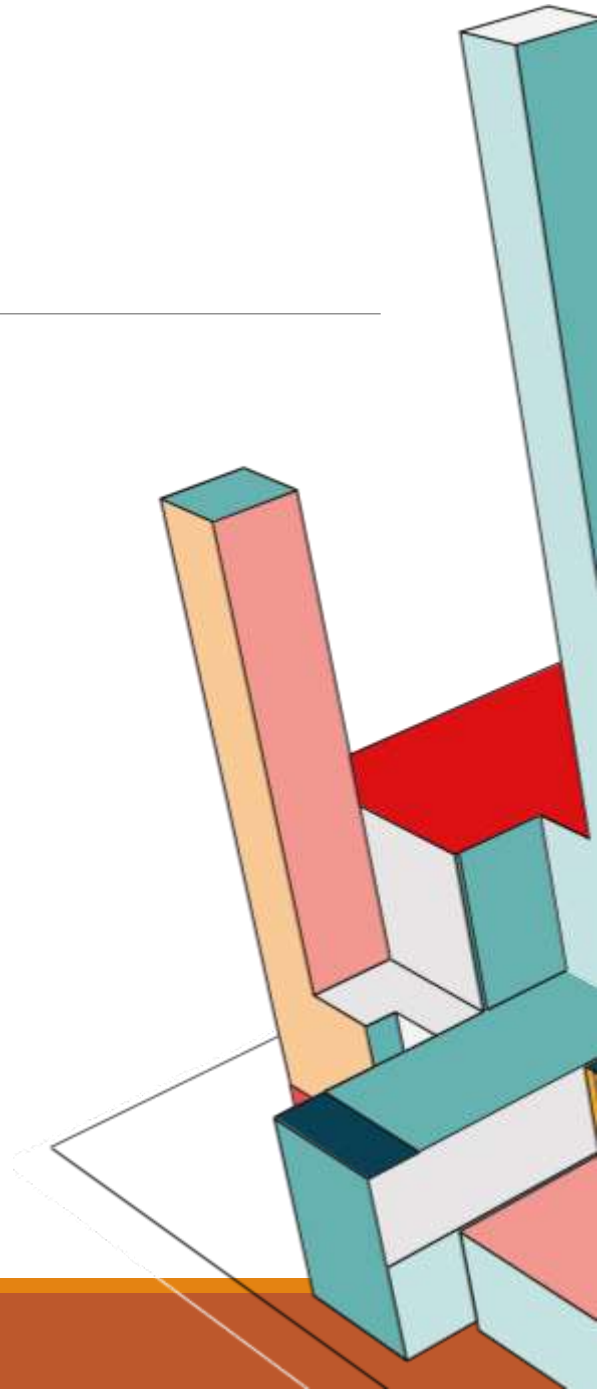
Avantages	Inconvénients	Cas d'usage typiques en NLP
➢ Simple à comprendre et implémenter ➢ Pas d'entraînement → utile pour les petits jeux de données ➢ S'adapte bien à la forme réelle des données ➢ Peut utiliser toute mesure de similarité	➢ Lent à prédire (doit comparer avec tous les exemples) ➢ Peu scalable pour les grands corpus ➢ Sensible au choix de la distance et de kkk ➢ Ne fournit pas de probabilités directement ➢ Performances médiocres si les données ne sont pas bien vectorisées	➢ Systèmes de recommandation de documents ➢ Classification de textes avec peu de données ➢ Réponses automatiques par analogie (FAQ)

Classification du texte

Approches de classification

Méthodes basées sur l'apprentissage profond

- **CNN (Convolutional Neural Networks)**
- **Réseaux de neurones classiques (MLP)**
- **Transformers (BERT, GPT, T5, etc.)**
- **RNN (Recurrent Neural Networks) et LSTM**

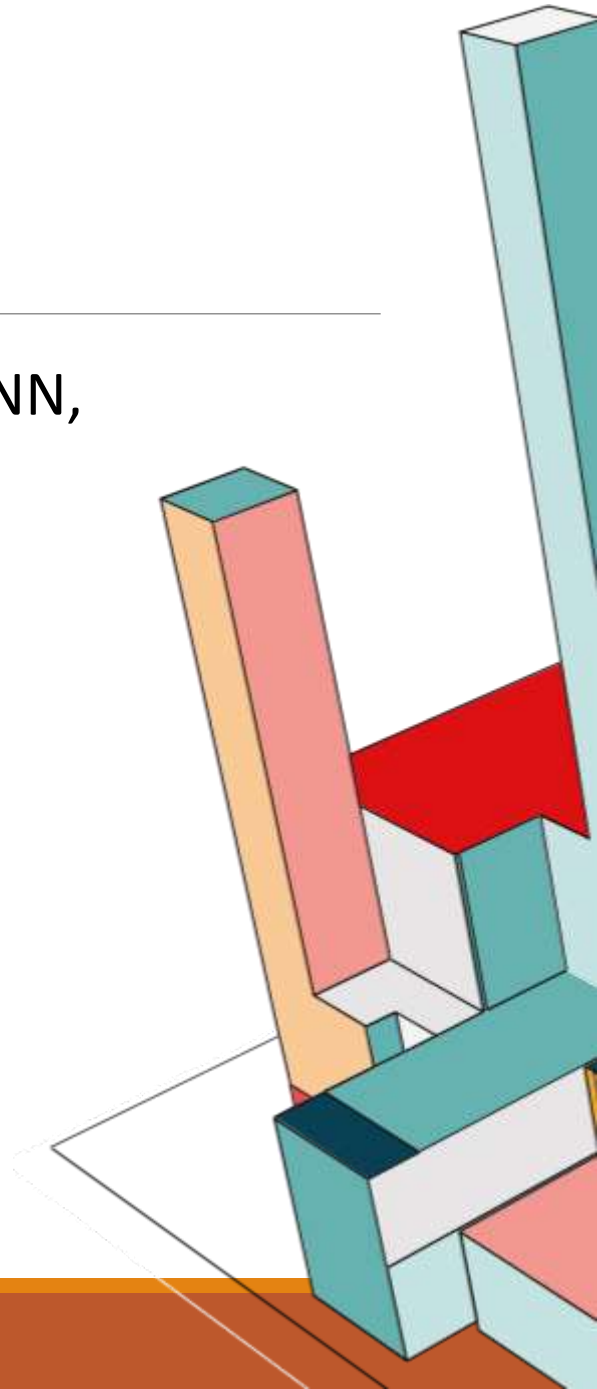


Évaluation des performances des modèles de classification

Après avoir entraîné un modèle de classification: SVM, Naïve Bayes, k-NN, ou autres , il faut savoir mesurer sa performance de façon fiable

C’est crucial pour :

- Comparer plusieurs modèles
- Détecter le surapprentissage
- Adapter les hyperparamètres
- Choisir le modèle le plus pertinent pour l’usage réel



Évaluation des performances des modèles de classification

Métriques d'évaluation :

1. Précision (Precision) Mesure la proportion des vrais positifs parmi les prédictions positives

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

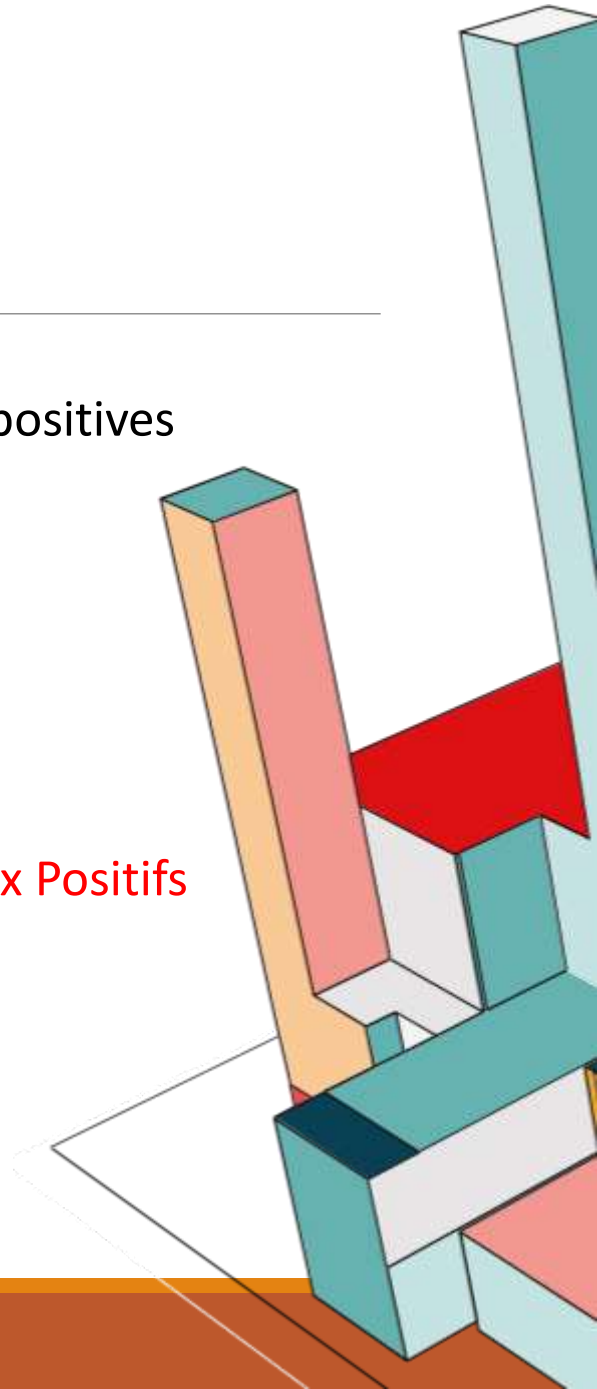
Imaginons un modèle de classification de spam
Il a classé 100 emails comme spam:

- Parmi eux, 80 sont vraiment des spams → ce sont des Vrais Positifs (VP)
- Les 20 autres sont des emails normaux, que le modèle a mal classés → ce sont des Faux Positifs (FP)

Alors : $\text{Précision} = \frac{80}{80 + 20} = \frac{80}{100} = 0{,}80$

Donc : **80 %** des emails prédits comme spam étaient effectivement du spam

→ C'est une bonne **précision** !



Métriques d'évaluation :

2. Rappel (Recall) Indique la capacité du modèle à détecter tous les exemples pertinents

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

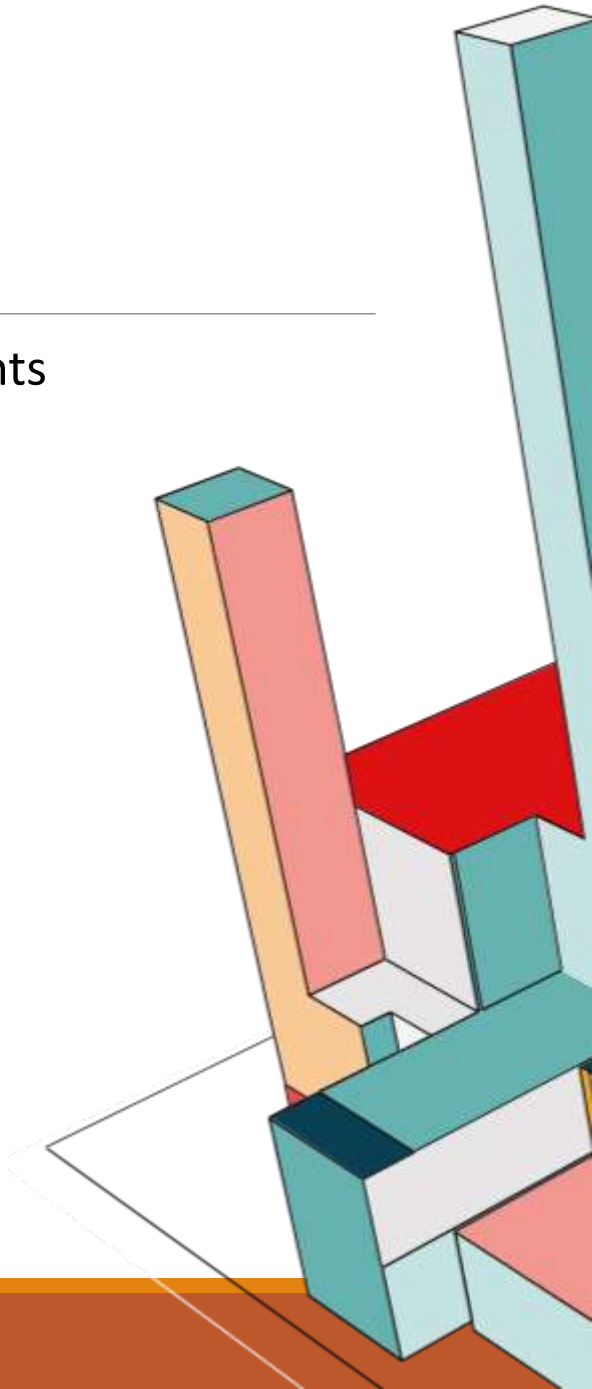
Imaginons toujours une tâche de détection de spam :
Il y a 100 emails qui sont réellement du spam dans notre jeu de test

- Le modèle en a correctement détecté 70 → ce sont les Vrais Positifs (VP)
- Il en a raté 30 : il les a classés comme "non spam → ce sont les Faux Négatifs (FN)

Donc :

$$\text{Rappel} = \frac{70}{70 + 30} = \frac{70}{100} = 0{,}70$$

Donc : le modèle a détecté 70 % des spams. Il en a manqué 30 %.



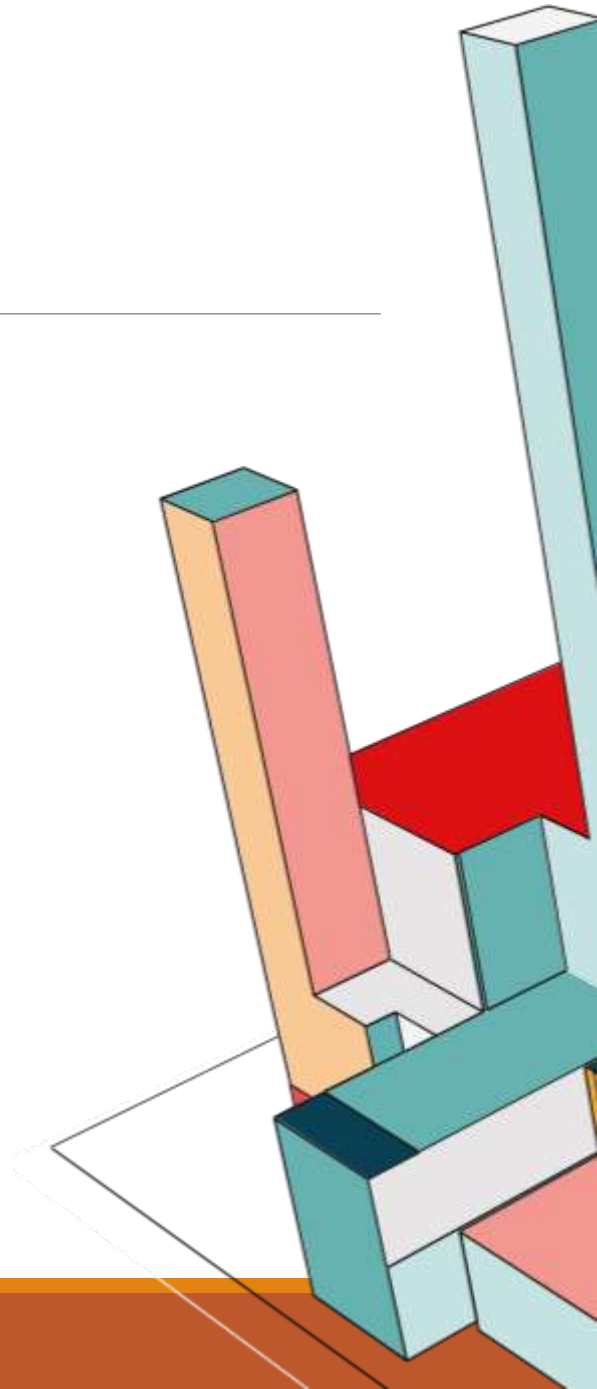
Métriques d'évaluation

3. F-mesure (F1-score)

La F-mesure est une métrique qui combine à la fois :

- la précision : qualité des prédictions positives
- et le rappel : capacité à ne pas rater de cas positifs
- Donc elle donne une valeur unique qui équilibre les deux.

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

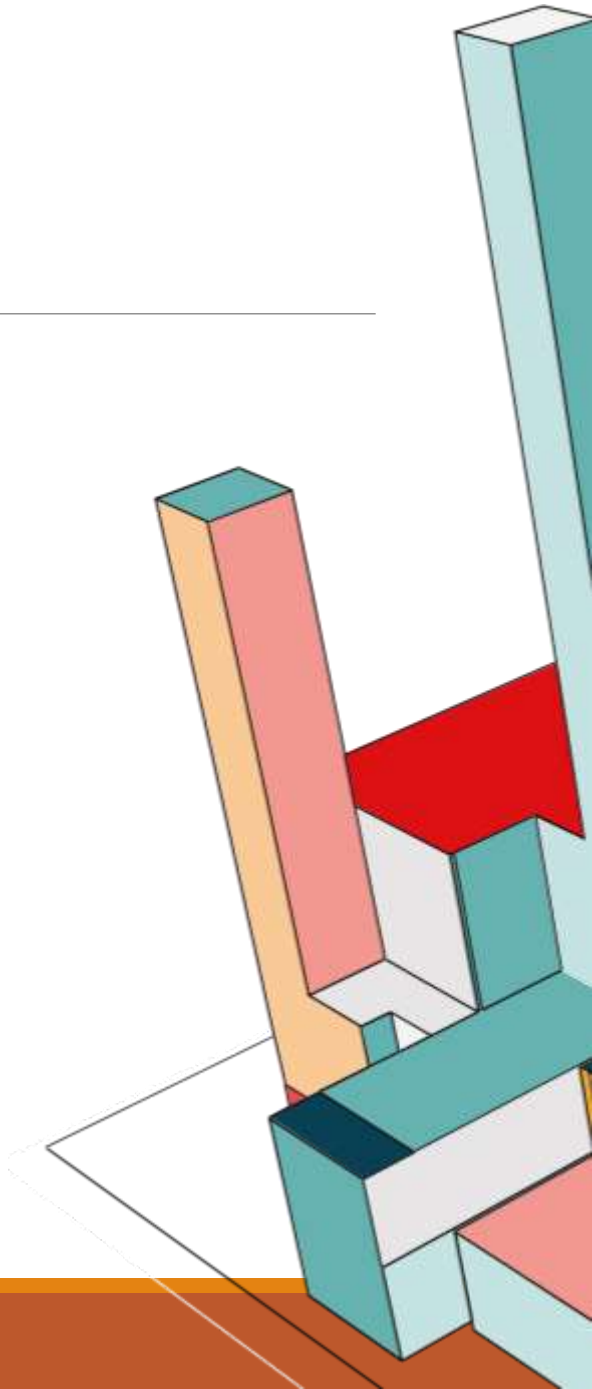


Métriques d'évaluation

4. Exactitude (Accuracy)

mesure la proportion de prédictions correctes parmi toutes les prédictions faites

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



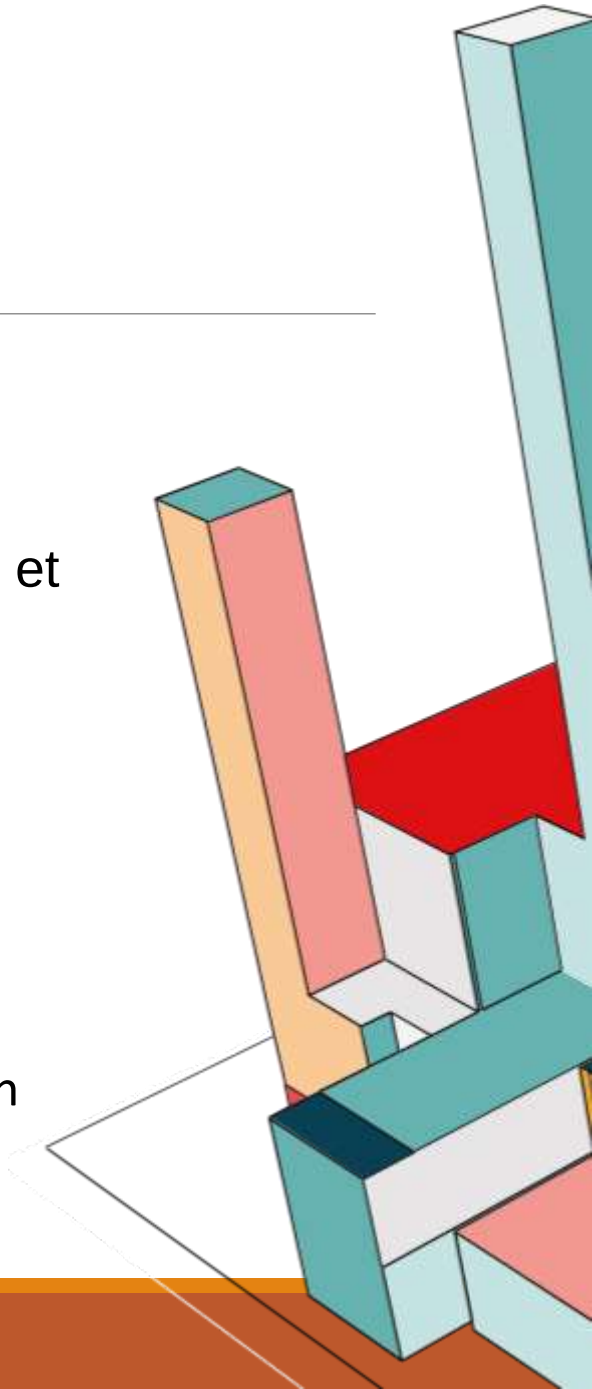
Métriques d'évaluation

5. Matrice de confusion

- Tableau qui affiche les vrais positifs (TP), vrais négatifs (TN), faux positifs (FP) et faux négatifs (FN)
- Aide à identifier les erreurs spécifiques du modèle

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

L'AUC-ROC mesure la capacité d'un modèle à distinguer entre les classes positives et négatives, en tenant compte des scores de confiance: probabilités ou scores de décision



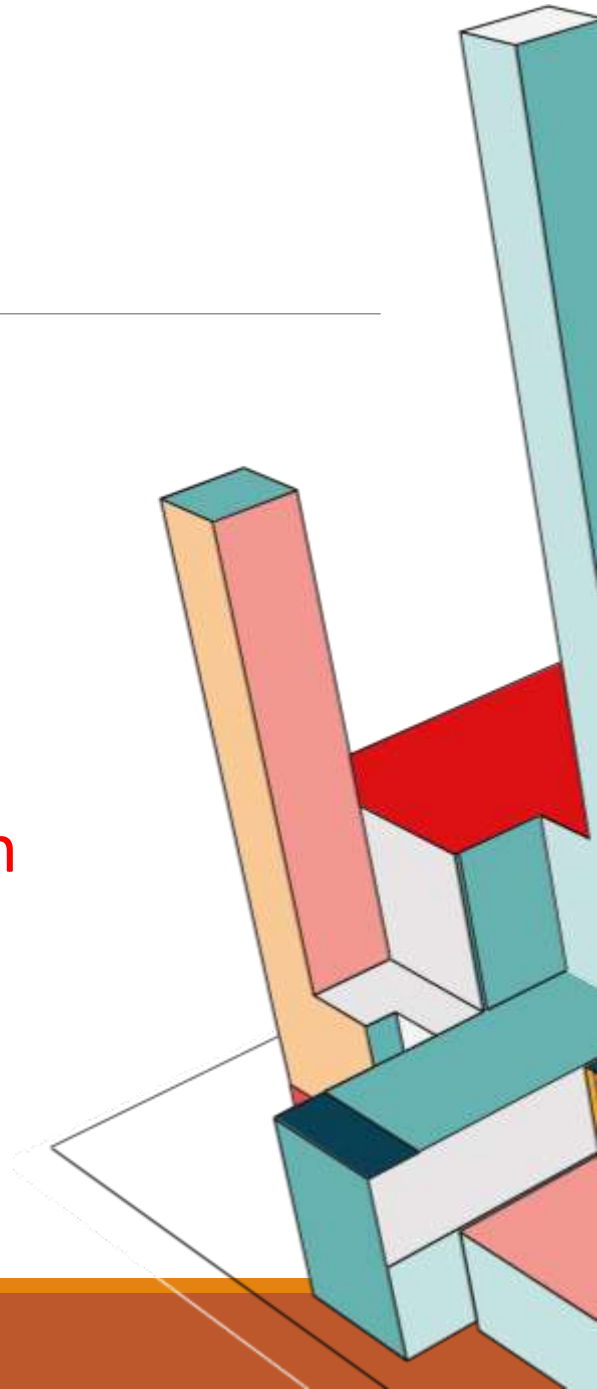
Validation du modèle

Entraînement/test split (train-test split)

On divise le jeu de données en deux parties :

- 70–80 % pour l'entraînement
- 20–30 % pour le test

Simple et rapide MAIS les résultats dépendants d'un seul split, donc peu robustes



Validation du modèle

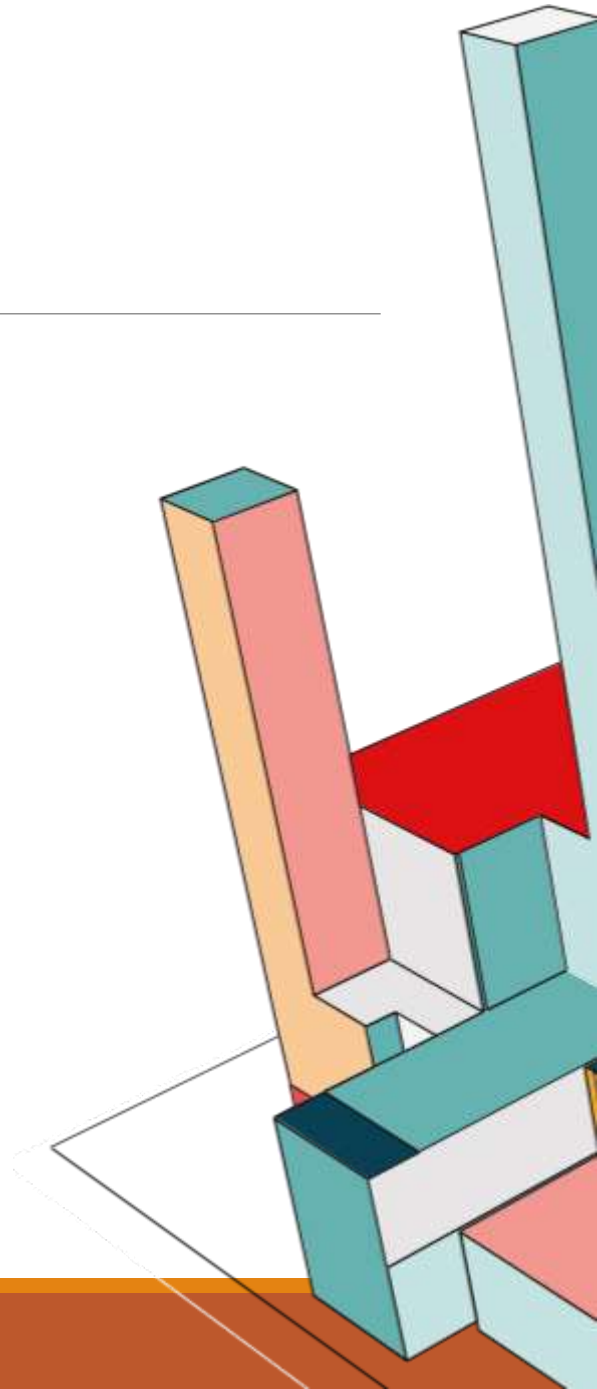
Validation croisée (k-fold cross-validation)

- Le jeu est divisé en k parties égales
- Le modèle est entraîné k fois, à chaque fois avec :

k–1 parties pour l'entraînement
1 partie pour la validation

- L'évaluation finale est la moyenne des scores obtenus

Robuste, réduit la variance des résultats
MAIS Plus coûteux en calcul



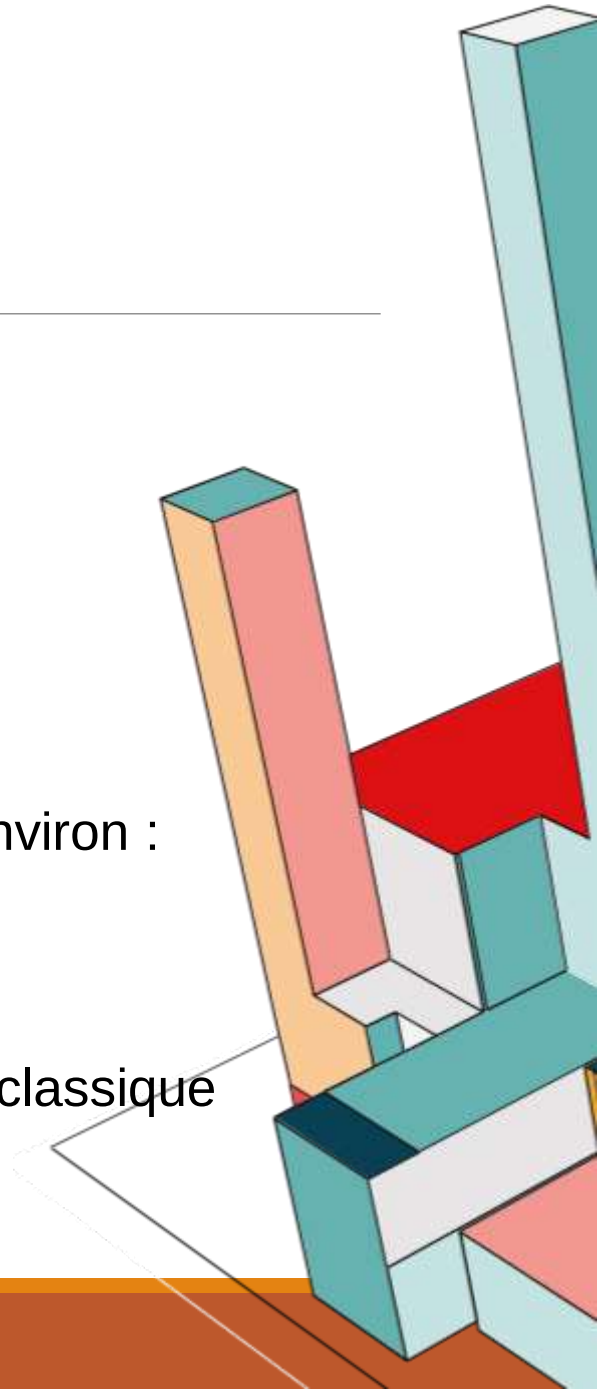
Validation du modèle

StratifiedKFold

est une variante de la validation croisée qui veille à conserver les proportions de chaque classe dans chaque sous-échantillon ou fold

Si notre dataset contient : 900 e-mails ham et 100 e-mails spam
Et que nous utilisons StratifiedKFold avec k = 5, alors chaque fold contiendra environ : 180 ham et 20 spam

Les rapports de classe sont conservés dans chaque fold, contrairement à KFold classique qui peut créer des folds très déséquilibrés.



Validation du modèle

En classification de texte :

- Utiliser StratifiedKFold pour respecter les proportions des classes
- Toujours mesurer plusieurs métriques : accuracy, F1, recall, précision
- Prévoir un bon découpage entre documents pour éviter le data leakage