TP2 – Classification de texte avec `scikit-learn` et `sentence-transformers`

Liwaa Zebian et Tarek Radwan
IFT3335 - Intelligence Artificielle

Hiver 2025

Partie 1 – BoW vs TF-IDF

1. Quelle méthode donne les meilleurs résultats ? Pourquoi ?

Les résultats mis à jour montrent que la méthode **TF-IDF** donne les meilleures performances avec le modèle MLP, tandis que **BoW** reste compétitif pour Logistic Regression. Voici le tableau comparatif :

Modèle	**Accuracy (BoW)**	**F1 (BoW)**	**Accuracy (TF-IDF)**	**F1 (TF-IDF)**
Logistic Regression	0.9793	0.9182	0.9596	0.8237
Random Forest	0.9735	0.8938	0.9733	0.8929
MLP Classifier	0.9811	0.9306	**0.9829**	**0.9337**

Analyse : TF-IDF permet au modèle MLP d'extraire des caractéristiques plus discriminantes et d'atteindre un F1-score supérieur. BoW est néanmoins plus performant pour Logistic Regression, probablement parce qu'il reflète mieux la fréquence brute des mots caractéristiques du spam, ce qui profite à un modèle linéaire.

2. Que se passe-t-il si on diminue `max_features` ?

Nous avons testé plusieurs valeurs de `max_features` pour la vectorisation TF-IDF avec un modèle Logistic Regression. Le graphique obtenu montre que :

— L'**accuracy reste stable** au-dessus de 0.96, même avec peu de features.
— Le **F1-score diminue légèrement** avec l'augmentation de `max_features`, indiquant que des vocabulaires trop larges peuvent introduire du bruit.

Ces résultats suggèrent qu'un vocabulaire modéré (1000–2000 mots) est suffisant pour cette tâche, et qu'ajouter des mots rares ne contribue pas à une meilleure performance.

Partie 2 – Sentence Transformers

1. Comparaison avec BoW et TF-IDF

Voici un tableau comparatif des F1-scores obtenus avec les trois approches :

Modèle	F1 (BoW)	F1 (TF-IDF)	F1 (Embeddings)
Logistic Regression	0.9182	0.8237	0.8753
Random Forest	0.8938	0.8929	0.8079
MLP Classifier	0.9306	**0.9337**	0.9329

Analyse : Les embeddings de phrases donnent des performances très compétitives, notamment avec le modèle MLP. Toutefois, TF-IDF offre un léger avantage dans ce cas précis. Pour les modèles plus simples, BoW reste solide grâce à sa simplicité et son efficacité sur des textes courts comme les SMS.

2. Avantages et inconvénients des embeddings

Avantages :
— Capturent la **sémantique** des mots et phrases.
— Meilleure généralisation sur les reformulations ou synonymes.
— Peu sensibles à la variation de vocabulaire.
Inconvénients :
— Plus ;lents à générer que BoW/TF-IDF.
— Requiert plus de ;ressources (RAM, CPU).
— Moins faciles à interpréter.

Partie 3 – Super Learner

Nous avons implémenté un **Super Learner**, une méthode ensembliste qui combine les prédictions de plusieurs modèles pour améliorer la performance globale. Dans notre cas, nous avons utilisé trois modèles de base :

— **Logistic Regression**
— **Random Forest**
— **SVC (avec StandardScaler)**

Ces modèles ont été entraînés sur les représentations **TF-IDF** des SMS.

Le modèle **méta** utilisé est une régression logistique, entraînée à partir des prédictions des trois modèles de base sur des validations croisées (StratifiedKFold à 5 plis).

Résultats du Super Learner

— **Accuracy** : 0.9832
— **F1 Score** : 0.9344

Ces résultats dépassent légèrement les meilleurs scores obtenus avec un seul modèle, démontrant l'intérêt d'une combinaison pondérée.

Poids attribués par le méta-modèle

Modèle	Poids appris
Logistic Regression	4.72
Random Forest	6.23
SVC (linéaire)	2.99

Cela signifie que le Super Learner fait davantage confiance aux prédictions du Random Forest et de la régression logistique, tout en conservant une contribution significative du SVC.

Commentaires sur les graphiques

Le graphique comparatif des **F1-scores** montre clairement que le Super Learner surpasse tous les modèles individuels :

— Logistic Regression : 0.9182
— Random Forest : 0.8938
— SVC linéaire : 0.9100
— Super Learner : **0.9344**

On constate donc que la combinaison pondérée permet d'exploiter les forces complémentaires des modèles de base, ce qui se traduit par une meilleure généralisation.