

## 1. Introduction

Ce programme est une application de détection de SMS SPAM utilisant Python et Streamlit.

Il permet de :

- Nettoyer et analyser les données textuelles.
- Entraîner un modèle de machine learning pour la classification.
- Offrir une interface interactive pour tester des SMS.

## 2. Importation des bibliothèques

- streamlit : Créer l'interface utilisateur interactive.
- pandas et numpy : Manipuler les données et effectuer des calculs.
- matplotlib et seaborn : Générer des graphiques pour visualiser les résultats.
- scikit-learn : Entraîner le modèle (régression logistique) et évaluer les performances.
- FPDF : Générer des fichiers PDF.

## 3. Nettoyage des messages

La fonction `clean_message` nettoie les données textuelles en :

- Convertissant le texte en minuscules.
- Supprimant les caractères non alphanumériques.
- Retirant les mots vides (stopwords).

## 4. Chargement et fusion des données

- `load_data` : Charge un fichier CSV contenant les SMS (colonnes `label` et `message`), nettoie les messages et renvoie un DataFrame.
- `merge_and_clean_data**` : Fusionne deux bases de données, nettoie et supprime les doublons.

## 5. Entraînement du modèle

- `train_model` :
  - Convertit les messages en vecteurs numériques avec TF-IDF.
  - Divise les données en ensembles d'entraînement et de test.
  - Entraîne un modèle de régression logistique pour prédire les SMS SPAM.

## 6. Visualisation des résultats

La fonction `display_graphs` génère :

- Une distribution des classes (HAM vs SPAM).
- Une matrice de confusion.
- Une courbe précision-rappel.

## 7. Interface utilisateur Streamlit

- Tester un SMS : L'utilisateur peut entrer un SMS et voir s'il est classifié comme SPAM ou HAM.
- Télécharger une nouvelle base : Charger un fichier CSV, fusionner et réentraîner le modèle.
- Réentraîner : Réentraîner le modèle avec les données fusionnées.

## Conclusion

Ce code démontre l'intégration de techniques de machine learning et d'une interface utilisateur pour résoudre un problème réel.