Document technique : Outil de traitement de texte et d'analyse de sentiment

# Introduction

Cet outil de traitement de texte est conçu pour automatiser le nettoyage et l'analyse de texte, avec une spécialisation dans la prédiction de sentiments à partir de commentaires de films. Il permet de traiter des données en plusieurs langues, de les nettoyer, puis de les classifier en utilisant un modèle d'apprentissage automatique. L'architecture repose sur plusieurs modules : un API FastAPI pour le traitement backend, un modèle de classification Bernoulli pour la prédiction, un fichier de prétraitement pour le nettoyage du texte, et une interface utilisateur Streamlit.

Cet outil est particulièrement utile pour les applications nécessitant un traitement rapide et automatisé de grandes quantités de texte (comme les avis de films) afin d'en extraire des informations utiles telles que le sentiment du texte.

# Composants principaux

## 1. API FastAPI (`api.py`)

L'API FastAPI joue un rôle central dans l'outil de traitement de texte. Elle assure l'interface entre l'utilisateur et le backend de l'application, en fournissant des points d'accès pour nettoyer et analyser le texte. FastAPI est utilisé pour son efficacité, sa rapidité et sa simplicité de déploiement.

Fonctionnalités de l'API :  
- Téléchargement de fichiers : L'API permet aux utilisateurs de charger des fichiers CSV ou TSV, contenant des critiques de films ou autres textes.  
- Prétraitement : L'API applique les étapes de prétraitement du texte en utilisant le script `preprocessing.py`.  
- Classification des sentiments : Une fois le texte nettoyé, l'API transmet les données au modèle de classification Bernoulli.  
- Retour des résultats : L'API renvoie les résultats sous forme de réponses JSON avec la classe de sentiment prédit.

Exemple de point d'accès :  
- `POST /predict/`: Ce point d'accès reçoit soit un fichier texte soit une chaîne de caractères et renvoie la prédiction de sentiment après nettoyage du texte.

## 2. Script de prétraitement (`preprocessing.py`)

Le fichier `preprocessing.py` contient un ensemble de fonctions de nettoyage de texte. Ce module est essentiel pour garantir que les données fournies au modèle de prédiction sont cohérentes, propres et sans bruit inutile.

Les étapes incluent :  
- Suppression des doublons : La fonction supprime les occurrences répétées d'une même phrase ou d'un même commentaire.  
- Détection de la langue : Utilisation de `langdetect` pour détecter les langues présentes et ne conserver que celles rédigées en anglais.  
- Nettoyage des textes : Suppression des caractères spéciaux, des chiffres, des stopwords.  
- Lemmatisation : Réduction des mots à leur forme de base.  
- Remplacement des valeurs vides : Les chaînes vides sont remplacées par des valeurs `NaN`.

## 3. Modèle de prédiction Bernoulli (`bernoulli\_model.joblib`)

Le modèle Bernoulli Naive Bayes est utilisé pour classer les critiques de films en différentes catégories de sentiments. Ce modèle est idéal pour traiter des données textuelles binaires, comme celles résultant des étapes de nettoyage et de vectorisation du texte.

Caractéristiques du modèle :  
- Entraînement : Le modèle a été entraîné sur un jeu de données Rotten Tomatoes, contenant cinq classes de sentiment.  
- Prédiction : Le modèle classifie chaque critique en fonction de son sentiment global.  
- Enregistrement et déploiement : Le modèle est enregistré dans un fichier `.joblib` et est chargé au moment des prédictions.

## 4. Interface utilisateur (Streamlit)

L'interface utilisateur construite avec Streamlit permet d'offrir un accès facile aux utilisateurs non techniques pour interagir avec l'outil de traitement de texte et de prédiction de sentiment. Cette interface est hébergée sur Streamlit Cloud.

Fonctionnalités :  
- Téléchargement de fichiers : Les utilisateurs peuvent charger un fichier CSV ou TSV.  
- Entrée manuelle de texte : Un champ de texte permet aux utilisateurs d'entrer directement un texte à analyser.  
- Nettoyage du texte : Le texte est nettoyé à l'aide des fonctions du fichier `preprocessing.py`.  
- Affichage des résultats : Les résultats de la prédiction de sentiment sont affichés à l'utilisateur.

## 5. Script de nettoyage de données (`Script\_nettoyage.ipynb`)

Ce notebook est un espace de travail utilisé pour développer et tester les étapes de nettoyage et d'analyse de données textuelles. Il permet de valider le bon fonctionnement des fonctions de nettoyage avant leur intégration dans `preprocessing.py`.

Étapes couvertes :  
- Suppression des doublons.  
- Gestion des valeurs vides (remplacement par `NaN`).  
- Nettoyage des données multilingues.  
- Test des algorithmes de détection de langue.

# Fonctionnement général de l'outil

1. Chargement et nettoyage des données : Les utilisateurs chargent un fichier ou entrent un texte dans l'interface Streamlit.  
2. Analyse et prédiction : Le texte nettoyé est passé au modèle Bernoulli pour prédire la classe de sentiment.  
3. Affichage des résultats : Les résultats sont renvoyés via l'API ou affichés dans l'interface Streamlit.

# Déploiement

1. \*\*FastAPI\*\* : L'API peut être déployée localement pour des tests, ou sur des plateformes cloud comme Microsoft Azure.  
2. \*\*Streamlit\*\* : L'interface est hébergée sur Streamlit Cloud.

# Cas d'utilisation

Cet outil est conçu pour des scénarios tels que :  
- Analyse de critiques de films  
- Surveillance des réseaux sociaux  
- Support client

# Conclusion

Cet outil est une solution complète pour le nettoyage, le traitement et l'analyse des données textuelles. Grâce à FastAPI, au modèle Bernoulli et à l'interface Streamlit, il offre une solution fluide pour l'analyse de sentiment en temps réel.