Royaume du Maroc

Université Abdelmalek Essaâdi

Faculté des Sciences et Techniques – Tanger

Big Data

Rapport

Projet: Prédiction de sentiments en temps réel des flux de données de réseau social Twitter

Cycle Master en Sciences et Techniques

Filière : Intelligence Artificielle et Sciences de Données

Encadré par

Yasyn EL YUSUFI

Réalisé par

Idriss KHATTABI

Ayman BOUFARHI

Abdelali IBN TABET

2023-2024

1. Introduction

L'analyse de sentiments est un domaine important de l'apprentissage automatique qui vise à déterminer les sentiments exprimés dans un texte, souvent des opinions ou des réactions sur les médias sociaux. Dans ce rapport, nous présentons un projet qui combine l'apprentissage automatique, les technologies web et les systèmes de streaming pour réaliser une analyse de sentiments en temps réel.

2. Description du Projet

Le projet consiste en une application web permettant aux utilisateurs de saisir un tweet dans un champ dédié, qui est ensuite traité par un modèle d'apprentissage automatique pour prédire le sentiment associé au tweet. Les résultats prédits sont affichés instantanément sur l'interface web, accompagnés de statistiques sur la distribution des sentiments dans les tweets soumis. De plus, les résultats sont stockés dans une base de données MongoDB pour une analyse ultérieure.

3. Technologies Utilisées

3.1 PySpark

Nous avons utilisé PySpark pour le prétraitement des données et l'entraînement des modèles d'apprentissage automatique.

3.2 Django

Django a été utilisé pour créer l'interface web permettant aux utilisateurs d'uploader leurs données de validation. Django offre une architecture MVC robuste et des fonctionnalités intégrées pour gérer les requêtes HTTP et générer des réponses dynamiques, ce qui en fait un choix idéal pour le développement web.

3.3 Kafka

Kafka a été utilisé pour la diffusion en temps réel des résultats prédits à partir du modèle d'apprentissage automatique vers l'interface web. Kafka est un système de messagerie distribué qui permet le streaming de données en temps réel avec une faible latence et une grande fiabilité.

3.4 MongoDB

MongoDB a été utilisé comme base de données pour stocker les résultats prédits. MongoDB est une base de données NoSQL flexible et évolutive, bien adaptée pour stocker des données semi-structurées telles que les résultats de l'analyse de sentiments.

4. Réalisation

4.1 Prétraitement des Données avec PySpark

Le prétraitement des données avec PySpark a joué un rôle crucial dans la préparation de nos données pour l'entraînement des modèles d'analyse de sentiments.

Nous avons utilisé PySpark pour effectuer une série d'opérations de nettoyage et de transformation sur nos données brutes. Ce qui est essentiel pour l'entraînement des modèles d'apprentissage automatique.

Cela comprenait notamment :

- La suppression des lignes vides
- La suppression des caractères spéciaux
- La mise en minuscules
- La tokenization
- La suppression des stop-word

4.2 Entraînement du Modèle

Après avoir prétraité nos données, nous avons procédé à l'entraînement de plusieurs modèles d'analyse de sentiments en utilisant PySpark MLlib. Nous avons expérimenté avec différents algorithmes de classification. Pour évaluer les performances de chaque modèle, nous avons utilisé des techniques telles que la validation croisée et l'Accuracy.

```
Logistic Regression Test Accuracy = 0.7797757418296185
Random Forest Test Accuracy = 0.37857240530562014
Decision Tree Test Accuracy = 0.34828387802543415
```

Après avoir comparé les performances de chaque modèle, nous avons sélectionné le modèle de régression logistique comme étant le meilleur modèle pour notre ensemble de données.

4.3. Interface Web Django

Nous avons choisi le framework Django pour créer l'interface web de notre projet.

Nous avons structuré notre application Django en utilisant des modèles pour représenter les données, des vues pour gérer la logique métier et des templates HTML pour générer les pages web dynamiques.

Nous avons également utilisé les fonctionnalités de formulaire de Django pour permettre aux utilisateurs de saisir des tweets directement sur l'interface.

4.4. Streaming en Temps Réel avec Kafka

L'intégration de Kafka dans notre projet a permis de gérer le streaming en temps réel des tweets, de leur traitement à leur stockage.

Voici comment nous avons mis en œuvre cette intégration.

Producteur Kafka

Nous avons utilisé un producteur Kafka pour lire les tweets à partir d'un fichier CSV et les envoyer au topic Kafka nommé 'numtest'. Le producteur Kafka, écrit en Python, lit chaque ligne du fichier CSV et envoie les données au topic toutes les trois secondes.

Ce mécanisme garantit que les tweets sont diffusés en continu pour être traités en temps réel.

Consommateur Kafka

Du côté consommateur, nous avons mis en place un consommateur Kafka qui récupère les tweets en temps réel à partir du topic 'numtest'. Le consommateur utilise PySpark pour charger un modèle de régression logistique précédemment entraîné et stocké.

Chaque tweet reçu est nettoyé et prétraité avant d'être transformé par le pipeline PySpark. Le résultat de la prédiction est ensuite affiché et stocké dans une base de données MongoDB pour une analyse ultérieure.

L'intégration avec PySpark nous permet d'exploiter la puissance de Spark pour le traitement des données en masse, tandis que Kafka assure une transmission efficace et en temps réel des données.

4.5 Stockage et Requêtes avec MongoDB

Pour stocker les résultats prédits ainsi que d'autres données pertinentes, nous avons utilisé MongoDB comme base de données NoSQL.

Après avoir prédit le sentiment d'un tweet à l'aide du modèle d'apprentissage automatique, nous créons un document contenant le texte du tweet et le résultat de la prédiction. Ce document est ensuite inséré dans la collection dans MongoDB.

5. Analyse des Résultats

L'analyse des résultats a montré que le modèle de régression logistique entraîné sur les données de validation a obtenu des performances satisfaisantes en termes de prédiction des sentiments des tweets.

Les statistiques affichées sur l'interface web ont fourni des informations utiles sur la répartition des sentiments dans les tweets, permettant aux utilisateurs de mieux comprendre les tendances dans les données.

6. Conclusion

Ce projet démontre avec succès l'intégration de différentes technologies pour réaliser une analyse de sentiments en temps réel. L'utilisation de PySpark pour le prétraitement des données et l'entraînement des modèles, combinée à Django pour l'interface web et à Kafka pour le streaming des résultats, a permis de créer une application robuste et performante. L'intégration de MongoDB comme base de données a également facilité la gestion et l'analyse des résultats.