

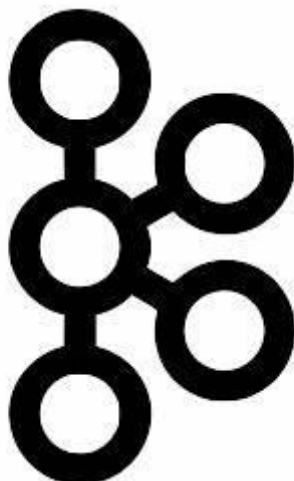


Université Abdelmalek Essaadi  
Faculté des Sciences et techniques de Tanger  
Département Génie Informatique



MST SITBD

# Rapport Technique sur la Prédiction des Sentiments :



Intitulé:

Département Génie Informatique

Encadré Par :

Pr. EL Yusufi Yasyn

Réalisé Par :

STITOU Fatima

HADDAD Alae

HAYOUN Fatima Zohra

## I. Introduction :

Ce rapport expose le développement d'une application Web destinée à l'analyse des sentiments en temps réel à partir des flux de données provenant de Prédiction de désabonnement des clients d'opérateur de Télécommunication en temps réel, en utilisant l'API Apache Kafka Streams. L'objectif primordial est de prédire la nature du sentiment (Négatif, Positif, Neutre, Irrelevant) en temps réel, en employant des outils pour l'analyse linguistique et pyspark pour le traitement des données. Ce projet s'inscrit dans le cadre du programme de Master en Sécurité IT et Big Data l'Université Abdelmalek Essaadi.

## II. Architecture Implémenté :

### 1. Les Outils / Frameworks Utilisés :

#### Kafka Stream :

Kafka Streams constitue le noyau du traitement des flux de données en temps réel dans notre application. Il permet de consommer les données en continu depuis les sujets Kafka, d'appliquer diverses transformations et analyses, et de produire des résultats en temps réel. Grâce à ses capacités de traitement distribué, Kafka Streams assure une scalabilité et une tolérance aux pannes optimales, garantissant un traitement efficace et résilient des données .

#### Cassandra :

Cassandra est utilisée comme base de données NoSQL pour stocker et gérer les données de sentiment analysées. Sa structure flexible et son support des documents JSON permettent de stocker des données complexes et hétérogènes, facilitant ainsi les requêtes et les analyses ultérieures. La capacité de Cassandra à gérer de grandes quantités de données et à fournir des performances élevées est cruciale pour l'archivage et la récupération rapide des résultats d'analyse.

## Flask:

Flask est utilisé comme framework web pour le développement de notre application. Ce micro-framework en Python est léger et flexible, permettant de créer des API RESTful facilement. Dans notre projet, Flask joue un rôle crucial en tant que couche d'interaction entre l'utilisateur et le système. Il expose les services d'analyse de sentiment via des endpoints HTTP, permettant ainsi aux clients de soumettre des données textuelles et de recevoir des résultats d'analyse en temps réel.

## Docker

Docker est employé pour containeriser l'ensemble des composants de notre application, assurant une portabilité et une reproductibilité maximales. Chaque composant, qu'il s'agisse de Kafka Streams, de Cassandra ou des outils de machine learning, est encapsulé dans un conteneur Docker. Cette approche facilite le déploiement, la gestion des dépendances et la scalabilité de notre application à travers différents environnements.

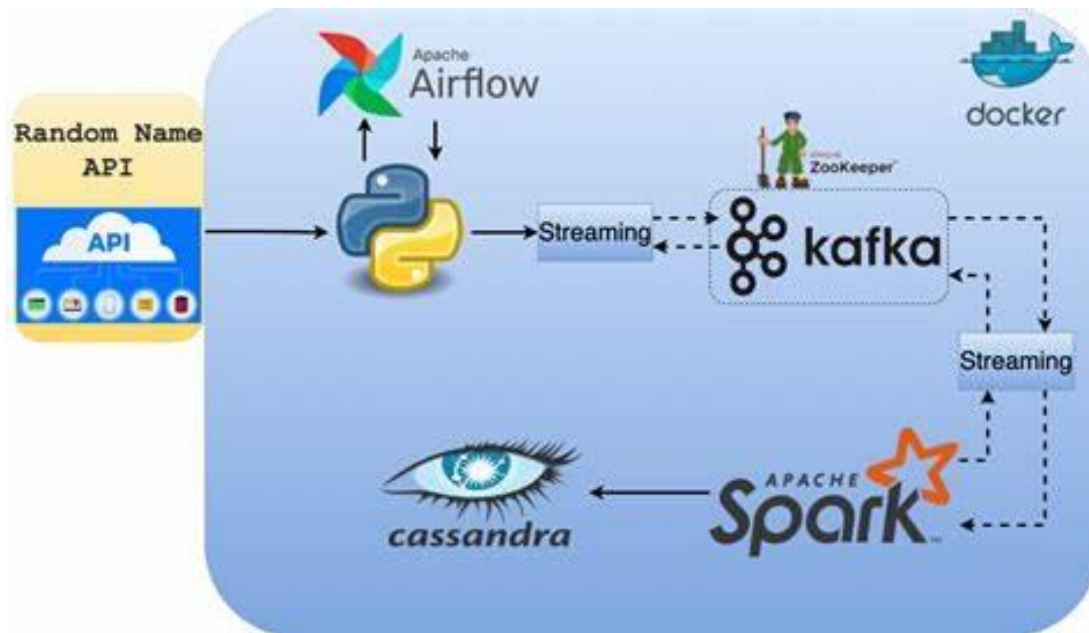
## PySpark.ml

PySpark.ml est utilisé pour les tâches de machine learning sur les données prétraitées. Cette bibliothèque de la suite Apache Spark permet de construire, d'entraîner et de déployer des modèles de machine learning à grande échelle. Grâce à ses capacités de traitement distribué, PySpark.ml assure des performances élevées même avec des volumes importants de données, facilitant la prédiction précise et rapide des sentiments.

## GitHub pour la Collaboration

GitHub sert de plateforme de collaboration et de gestion de version pour notre projet. Il permet à l'équipe de développement de travailler de manière coordonnée, de suivre les modifications de code, et de gérer les branches et les fusions. Les fonctionnalités de revue de code et de suivi des issues de GitHub assurent une qualité de code élevée et une résolution efficace des problèmes, tout en facilitant la collaboration entre les membres de l'équipe.

## Le Processus:



## Explication du Processus :

Le processus d'architecture impliquant Spark, Cassandra, Docker et Kafka est conçu pour permettre le traitement efficace des données en flux, leur stockage persistant et leur visualisation interactive. Dans ce cadre, Kafka agit en tant que middleware pour la réception et la distribution des données en temps réel. Les données sont ensuite traitées par Spark, avec des scripts Python exécutés dans des conteneurs Docker, permettant l'utilisation d'algorithmes de machine learning pour des analyses avancées. Une fois traitées, les données sont enregistrées dans Cassandra, offrant une solution de stockage NoSQL robuste pour gérer les grandes quantités de données générées. Docker facilite le déploiement et la gestion de l'ensemble de l'architecture, permettant une orchestration fluide des services dans un environnement conteneurisé. Cette approche garantit une scalabilité, une flexibilité et une résilience optimales, tout

en simplifiant le processus de développement, de déploiement et de maintenance de l'infrastructure de traitement et de stockage des données en flux.

### III. Visualisation :

#### 1- DASHBOARD:



Ce tableau de bord affiche des visualisations en temps réel pour l'analyse des sentiments à partir des flux de données des réseaux sociaux. Il comprend quatre sections principales:

- **Graphique à barres** : Montre l'évolution des valeurs de prédiction des sentiments au fil du temps.
- **Diagramme circulaire** : Ce diagramme affiche la répartition des sentiments en différentes classes (positif/négatif, par exemple).
- **Graphique en ligne** : Représente la distribution des niveaux de confiance des prédictions de sentiment.

- [Graphique à Barres Horizontales](#): Ce graphique illustre comment les prédictions de sentiments sont réparties entre différents états.

Ce tableau de bord permet de visualiser rapidement l'analyse des sentiments et la performance du système en temps réel.

## 2- Architecture :

Cette interface présente l'architecture d'un Système d'Analyse de Sentiments. Voici une description concise :

### Outils Utilisés

- [Logos des outils](#) : Kafka, Docker, Machine Learning, Spark, Cassandra, GitHub
- [Diagramme d'Architecture](#) : Illustration du flux de travail.
- [Documentation](#) : Liens vers les documentations des outils utilisés