Rapport de Projet : Pipeline Data Lakehouse HackerNews

Membres du groupe : Manne KITSOUKOU, Julien MILLION, Aymane OURAQ, Jana ZEBIAN

Technologies utilisées : Spark, Kafka, Delta Lake, Spark NLP, Garage (S3), Pandas/Seaborn.

Date : janvier 2026
Repo GitHub : https://github.com/aymaneo/HackerNews

1. Introduction

Ce projet implémente un pipeline **Data Lakehouse** complet pour l'analyse en temps réel du flux HackerNews. L'enjeu est de transformer un flux de données textuelles brutes en insights structurés.

L'API HackerNews (Firebase) : Nous avons exploité l'API temps réel qui diffuse les articles (*stories*) et les commentaires. L'intérêt majeur réside dans la richesse textuelle, traitée ici via **Spark NLP** pour extraire une dimension sémantique (analyse de sentiment) des titres et contenus, permettant ainsi de dépasser la simple analyse statistique.

Fonctionnalités du pipeline :

- Collecte en temps réel via un Producer Kafka.
- Stockage structuré suivant l'architecture **Médaillon** (Bronze, Silver, Gold) sur Delta Lake.
- Traitement analytique avec Spark et Spark NLP pour l'extraction de sentiments ou de thématiques.
- Visualisation des tendances via des outils de Data Science (Pandas/Seaborn).

2. Description technique de l'API et du flux

Commandes et Endpoints utilisés

L'API HackerNews est interrogée via l'interface Firebase. Les principaux points de terminaison utilisés sont :

- `https://hacker-news.firebaseio.com/v0/topstories.json` : Récupère les IDs des articles les plus populaires.
- `https://hacker-news.firebaseio.com/v0/item/{id}.json` : Récupère les détails complets d'un item (article, commentaire, etc.).

Liste des attributs du flux de données

Chaque "item" récupéré contient les attributs suivants que nous traitons dans notre pipeline :

- `id` : Identifiant unique de l'item.
- `type` : Type de contenu (story, comment, job, poll).

- `by` : Nom de l'auteur.
- `time` : Timestamp de publication (format Unix).
- `text` : Contenu textuel (essentiel pour Spark NLP).
- `score` : Nombre de points de l'article.
- `title` : Titre de l'article.
- `descendants` : Nombre total de commentaires.

3. Résultats obtenus et explications

L'exécution du pipeline a permis d'extraire des informations concrètes sur la dynamique de la communauté HackerNews. Voici l'analyse des résultats issus de nos traitements Spark et Spark NLP :

A. Traitement Batch et SparkSQL

Grâce à la structuration des données dans le Data Lakehouse, nous avons pu effectuer des requêtes croisées complexes :

- **Analyse par Domaine :** Nous avons classé les sources externes les plus partagées (ex: *github.com*, *youtube.com*, *huggingface.co*). GitHub arrive en tête avec 132 commentaires associés, mais nous observons que certains domaines comme *shreevatsa.net* (86.67%) ou *youtube.com* (73.68%) génèrent des discussions beaucoup plus positives que la moyenne.
- **Performance des Utilisateurs :** Un classement des contributeurs a été établi selon leur "positive_ratio". Des utilisateurs comme *ChrisArchitect* ou *rahimnathwani* maintiennent un taux de 100% de commentaires positifs sur leurs interventions, ce qui permet d'identifier les profils les plus constructifs de la plateforme.

B. Streaming en temps réel et Fenêtrage

L'intégration de la bibliothèque Spark NLP a permis de transformer le texte brut en données structurées exploitables :

- **Reconnaissance d'Entités Nommées (NER) :** Le pipeline a identifié les sujets dominants. Sans surprise, l'entité **"AI"** est la plus mentionnée (97 occurrences), suivie de géants comme **Apple** et **Google**, ainsi que des technologies émergentes comme les **LLMs**.

- **Extraction de Mots-Clés :** Les termes les plus fréquents dans les discussions ont été isolés. Outre les verbes d'action (*use, like, get*), on note une forte précurrence des termes liés au "code", au "work" et au "time", reflétant la culture technique et productive des utilisateurs.
- **Analyse de Sentiment :** Le modèle a classé chaque message. Les résultats montrent une prédominance de sentiments **négatifs** (environ 800 commentaires) par rapport aux **positifs** (environ 600), avec une très faible portion de messages neutres. Cela s'explique souvent par l'aspect critique et analytique des débats sur HackerNews.

C. Visualisations (Pandas & Seaborn)

Les données traitées ont été exportées vers un environnement de Data Science pour produire des synthèses visuelles claires :

- **Distribution des Sentiments :** Un histogramme montre visuellement que la communauté est très polarisée, avec un volume de commentaires négatifs/critiques supérieur au volume positif.
- **Top Entités :** Un bar chart horizontal met en évidence l'omniprésence de l'Intelligence Artificielle dans les flux actuels, surpassant largement les autres entités technologiques ou organisationnelles.
- **Fréquence des Mots-Clés :** Une visualisation des 20 mots-clés les plus utilisés permet de comprendre en un coup d'œil les préoccupations majeures des développeurs (utilisation d'outils, qualité du code, gestion du temps).

4. Conclusion et Perspective

Le projet démontre la viabilité d'une architecture Lakehouse pour le traitement de données Le projet a atteint l'ensemble de ses objectifs initiaux. Nous avons réussi à transformer un flux de données instable et non structuré en un système de connaissance organisé. L'architecture **Médaillon** assure la fiabilité des données, tandis que **Kafka** permet une ingestion sans perte.

L'apport de **Spark NLP** est ici fondamental : il permet de ne plus seulement compter des interactions, mais de mesurer l'opinion et d'identifier les tendances technologiques (comme l'hégémonie de l'IA) de manière automatisée. Ce pipeline constitue une base solide pour n'importe quelle application de veille technologique ou d'analyse de marché en temps réel.