Ynov M2 data - 2024 – 2024
N. El Malki & M. El malki

Plateforme d’Analyse des Marchés Boursiers et des Événements Impactants

1. Contexte

Les marchés financiers sont influencés par de nombreux facteurs extérieurs tels que les décisions politiques, les innovations technologiques, les annonces économiques et les crises mondiales. Un investisseur doit constamment suivre ces informations et analyser leur impact pour prendre des décisions éclairées.

Aujourd’hui, il est difficile d’exploiter cette grande quantité de données en temps réel pour comprendre leur influence sur le marché. Ce projet propose de développer une plateforme intelligente qui collecte, analyse et structure ces informations pour assister les investisseurs dans leurs décisions.

2. Objectif du Projet

L’objectif est de concevoir une architecture de traitement de données temps réel qui :

- ❑ Collecte les données financières (indices boursiers, actions, crypto) et les actualités économiques, politiques et technologiques.
- ❑ Analyse et enrichit ces données avec un LLM (large langage model) pour extraire des tendances et liens de causalité.
- ❑ Stocke et organise ces données dans des bases relationnelles et NoSQL .
- ❑ Propose une interface utilisateur permettant d’interroger ces informations en langage naturel, converti en requêtes SQL/NoSQL via un LLM.

Étape 1 : Collecte et Ingestion des Données (NiFi & Web Scraping)

La première étape du pipeline consiste à collecter des données en temps réel provenant de plusieurs sources. L'objectif est d'obtenir des cours de bourse, des actualités financières, et des événements économiques, technologiques et politiques susceptibles d'affecter les marchés.

Les données en temps réel sur les cours de bourse peuvent être collectées à l'aide des bibliothèques Python `yfinance` et `yahooquery`. Les informations sur les actualités financières, économiques, technologiques, politiques, et les événements mondiaux peuvent être récupérées via des flux RSS provenant de divers médias français et étrangers, tels que :

- Les Échos : https://www.lesechos.fr/rss/finance-marches.xml

- La Tribune : https://www.latribune.fr/rss/rubrique/finance.xml

Apache NiFi est utilisé pour automatiser la collecte et l'ingestion des données. Il permet :

- D'extraire les données via des API, du scraping, et des flux RSS.
- De router les données vers Kafka pour le traitement en streaming.

Dans un premier temps, on peut s'appuyer sur un script python qui sera ensuite exploité/utilisé par nifi

Étape 2 : Streaming et Transformation (Kafka + Spark)

Une fois les données collectées via NiFi et le scraping, elles sont envoyées vers Kafka sous forme de topics. Kafka agit comme un bus de messages pour distribuer ces données aux consommateurs (Spark et le LLM).

Les données sont organisées sous plusieurs topics, selon leur type, par exemple :

- topic_cours_bourse → contient les prix des actions, crypto et indices.
- topic_actualites_finance → contient les nouvelles économiques et financières.
- topic_evenements_mondiaux → regroupe les annonces politiques, technologiques et réglementaires.

Vous avez la possibilité d'ajouter ou d'affiner d'autres topics. Pour chaque topic, réfléchissez au schéma de données à adopter pour le stockage.

Les données sont ensuite traitées par Spark Streaming, où elles sont structurées dans un format adapté au stockage, puis enrichies en créant de nouvelles mesures et en établissant des associations entre les données.

Mesures

De nouvelles métriques sont calculées à partir des données boursières, telles que la volatilité, la tendance et le volume des transactions :

- **Volatilité** : Calculée comme l'écart-type des rendements logarithmiques sur une période donnée (par exemple, 30 jours). Plus l'écart est grand, plus la volatilité est élevée.
- **Tendance** : Utilise les moyennes mobiles (SMA ou EMA) pour observer la direction du prix. Une pente positive de la régression linéaire indique une tendance haussière.
- **Volume de Transactions** : Représente le nombre d'unités échangées. Une moyenne mobile du volume sur plusieurs jours permet de détecter des tendances de forte ou faible activité.

Associations

Après avoir collecté les données sur les bourses et les actualités, vous devez établir des associations entre ces informations. Voici un exemple de création d'association :

1. **Extraction des entités financières dans les flux RSS**
    a. Identifier les entreprises mentionnées dans les articles.
    b. Extraire des noms d'entreprises et symboles boursiers.
    c. Utiliser des bases de données de correspondance pour vérifier si ces entités sont cotées en bourse.

2. **Récupération des données boursières associées**
    a. Vérifier l'évolution du cours de l'action concernée avant et après l'événement.
    b. Déterminer si l'événement a eu un impact significatif sur le marché.

3. **Association événement-actif**
    a. Si un événement cite explicitement une entreprise cotée, on l'associe directement à son indice en bourse.
    b. Si un événement concerne un secteur spécifique (ex : "Nouvelle régulation sur l'IA"), on l'associe aux entreprises du secteur concerné (ex : NVDA, GOOGL, MSFT).
    c. Si l'événement concerne un pays ou une économie, on l'associe à des indices (S&P 500, CAC 40, DAX).

L'étape 1 peut être réalisée en utilisant un LLM. Pour cela, vous pouvez exploiter les bibliothèques Python `huggingface_hub` ou `transformers` afin d'interroger un LLM en langage naturel et obtenir une réponse.

En complément de ces associations, pour chaque actualité, créez des métadonnées à l'aide du LLM. Les métadonnées sont des informations qui décrivent, résument ou expliquent les données brutes, facilitant ainsi leur organisation, analyse et recherche. Dans notre cas, les métadonnées que nous allons générer à l'aide du LLM peuvent inclure :

- **Type d'événement** : politique, économique, technologique, etc.
- **Actifs concernés** : les actions, indices boursiers ou crypto-monnaies affectés par l'événement.
- **Impact attendu** : augmentation ou baisse de prix, volatilité accrue, etc.
- **Localisation géographique** : pays ou régions affectés par l'événement (par exemple, un changement politique en Europe).
- **Date de l'événement** : moment où l'événement a eu lieu ou est prévu.
- **Détails supplémentaires** : informations spécifiques fournies par l'événement, telles que des chiffres clés ou des annonces précises.

Ces méta-données sont fournies à titre d'exemple, mais vous pouvez les enrichir avec d'autres types de métadonnées.

Toutes les données collectées ainsi que celles que vous avez générées doivent être stockées dans des bases de données relationnelles et/ou NoSQL.

Étape 3 : Création d'un Moteur de Recherche en Langage Naturel avec Transformation en Requêtes SQL/NoSQL

Créez un moteur de recherche qui permet à un utilisateur de poser des questions en langage naturel sur les actifs boursiers et événements mondiaux. Ces questions sont ensuite transformées en requêtes SQL ou NoSQL pour interroger les bases de données, et les réponses sont formulées en langage naturel via le LLM, basées sur les résultats des requêtes.

Les types de questions peuvent être :

- *"Quels sont les impacts des élections françaises sur les actions des entreprises cotées ?"*
- *"Donne-moi la tendance des actions Tesla cette semaine."*

- *"Quels sont les indices boursiers affectés par les nouvelles économiques sur la crise énergétique ?"*
- *"Quels événements mondiaux ont impacté le marché des cryptomonnaies cette année ?"*

Le moteur de recherche se compose de plusieurs éléments :

- **Interface Utilisateur (UI)** : Permet à l'utilisateur de poser des questions en langage naturel.
- **LLM (Large Language Model)** : Transforme les questions en requêtes SQL ou NoSQL adaptées à la base de données.
- **Base de Données** : Contient les informations sur les actifs boursiers, événements mondiaux, et autres données pertinentes. Les bases peuvent être relationnelles (par. ex. MySQL) ou NoSQL (par. Ex. MongoDB).
- **Backend de Traitement des Requêtes** : Exécute les requêtes SQL ou NoSQL générées par le LLM et renvoie les résultats.
- **Générateur de Réponse en Langage Naturel** : Formule les réponses aux utilisateurs, basées sur les résultats des requêtes en utilisant le LLM.

Avant que le LLM ne transforme les requêtes en langage naturel en requêtes SQL ou NoSQL, il est crucial de lui fournir des informations sur la structure des bases de données. Cela garantit une cohérence entre les requêtes SQL/NoSQL générées par le LLM et le contenu des bases de données.

L'utilisateur pourra poser des questions via une interface que vous développerez, par exemple un chatbot ou une interface web utilisant Streamlit.

Tâches :

Le projet doit être réalisé en groupe de 4 à 5 personnes.

1. **Proposez une architecture détaillée de votre solution**, puis discutez-en avec l'enseignant.
2. **Pour chaque base de données**, définissez le schéma de données associé pour les bases relationnelles et la structure générale des données pour les bases NoSQL. Vous êtes libres de choisir le type de stockage ainsi que le système de gestion de bases de données.
3. **Développez chacune des étapes** décrites ci-dessous.

Soutenance :

durera **30 minutes**, réparties comme suit :

- **20 minutes de présentation** de votre solution.
- **10 minutes d'échanges** avec le jury.

Vous devez préparer une présentation PowerPoint et une démonstration en direct de votre solution.

Livrables :

Les livrables à rendre sont les suivants :

- Un **rapport** détaillant votre solution, les choix effectués, ainsi que les difficultés rencontrées pendant le projet.
- Le **code source** de votre projet.
- Les **supports de présentation** utilisés pour la soutenance.

Note importante : La note est individuelle. Dans le rapport, chaque membre du groupe doit spécifier les tâches réalisées et sa contribution au projet.