Pratique de la data science

April 4, 2025

1 TP 8 - Classification de News et impact sur les variations de stocks

Librairies à installer :

```
import json
import os
from datetime import datetime
import matplotlib.pyplot as plt
import yfinance as yf
import pandas as pd
import pytz
from transformers import BertTokenizer, BertForSequenceClassification
import torch
from collections import defaultdict
from matplotlib.lines import Line2D
```

Objectif : Visualiser l'effet potentiel des News financières sur les prix des actions (on regarde cette fois-ci les variations toutes les heures). On utilisera la modèles finetunés dans le TP7 pour classifier les news scrappés sur l'année 2025 afin de visualiser des possibles corrélations entre les sentiments prédits sur ces news et les variations de stocks.

Tout doit être codé sous forme de fonctions. L'objectif est de pouvoir utiliser nos fonctions pour chaque compagnie.

L'étude sera faite sur les cmpagnies pour lesquelles on a le plus de news (Microsoft, Apple, Tesla, Amazone...)

1.1 Extraction des textes et timestamps (date et heure) des actualités

Créer une fonction `get_texts_timestamps(news_data)` qui permet de **transformer le fichier JSON d'une compagnie en deux listes (text et timestamps)** pour permettre l'analyse de sentiments et la visualisation).

Étapes :

1. Pour chaque article du fichier JSON :
   - Convertir le `timestamp` UTC (format ISO) en timezone de New York (`America/New_York`).
   - Arrondir les timestamps à l'heure pleine précédente (en supprimant minutes, secondes et millisecondes).
   - Construire un texte complet en concaténant les textes du `title` et de la `description`.
2. Retourner deux listes parallèles :
   - `news_texts` : liste des textes prêts pour l'analyse de sentiment.
   - `news_timestamps` : liste des timestamps alignés avec le fuseau horaire de marché.

1.2 Analyse de sentiments grâce aux modèles finetunés en TP7

Créer une fonction `get_sentiments(model_path, texts)` qui permet d'appliquer le modèle finetuné à notre liste de textes pour prédire leurs sentiments.

1. Charger le tokenizer et modèle à utiliser, initialiser une liste vide pour conserver les prédictions :

```
tokenizer = BertTokenizer.from_pretrained("ProsusAI/finbert")
model = BertForSequenceClassification.from_pretrained(model_path)
model.eval()
sentiments = []
```

2. Pour chaque texte dans la liste :
   (a) Transformer les inputs en tokens (`tokenizer...`).
   (b) Appliquer le modèle (sans calcul de gradient (`with torch.no_grad()`).
   (c) Extraire les prédictions de sentiment (`argmax` sur les logits).
   (d) Ajouter la prédiction dans la liste `sentiments`.

1.3 Alignement des timestamps avec les heures d'ouverture des marchés

Créer une fonction `align_timestamps(timestamps)` qui permet d'aligner les horaires des News avec les horaires des marchés (pour rendre possible superposition cohérente entre les événements textuels et les données de prix.
Les News publiées en dehors des horaires de marchés sont mappés sur la dernière heure d'ouverture

- Si la news est publiée entre **9h30 et 15h** ⇒ associée à l'heure de publication.
- Si la news est publiée entre **15h et minuit** ⇒ associée à **15h le même jour**.
- Si la news est publiée entre **minuit et 9h30** ⇒ associée à **15h la veille**.

1.4 Visualisation comparative des sentiments et du prix de l'action

Créer une fonction `plot_comparison(df, sentiments_a, sentiments_b, timestamps, title_a, title_b)` qui affiche deux graphiques côte à côte permettant de comparer visuellement les prédictions de deux modèles d'analyse de sentiments (par exemple FinBERT base vs fine-tuné) sur des News et leurs impacts sur l'évolution du prix d'une action.

Entrées :

- `df` : DataFrame contenant l'historique des prix avec des intervalles de 60 mins (colonnes = `Datetime` et `Close`).
- `sentiments_a`, `sentiments_b` : listes de sentiments prédits (`0`, `1`, `2`) pour deux modèles différents.
- `timestamps` : liste des dates de publication des articles (déjà converties en timezone marché).
- `title_a`, `title_b` : titres des sous-graphiques.

Étapes de la fonction :

1. Aligner les timestamps des news avec les horaires de marché à l'aide de la fonction `align_timestamps()`.
2. Grouper les sentiments par timestamp (il peut y avoir plusieurs news au même moment).
3. Tracer deux sous-graphiques :
   - Tracer la courbe des prix de l'action.

- Superposer les news sous forme de points colorés :
  - `Vert` : sentiment positif
  - `Or` : sentiment neutre
  - `Rouge` : sentiment négatif
- Décaler légèrement les points en ordonnée pour distinguer si plusieurs news à la même heure.

4. Ajouter une légende explicite avec les couleurs et la courbe de prix.

L'objectif de cette visualisation est de comparer l'interprétation des sentiments par deux modèles sur une même période, et de détecter visuellement des liens entre événements médiatiques et variations du prix de l'action.

1.5 Mise en commun des fonctions pour la visualisation

Pour les compagnies avec suffisament de News :

1. Extraire les variations de stock :

```
ticker = yf.Ticker("...")
df = ticker.history(start="2025-01-01", interval="60m")
df = df.reset_index()
```

2. Extraire les News et timestamps du fichier JSON correspondant.
3. Extraire les sentiments.
4. Afficher les graphiques pour deux modèles (Exemple : FinBERT et FinBERT finetuned).