

Rapport Final - Projet Advanced Python, Git, Linux for Bloomberg

Gilles Delpy SOP et Barthélémy BENGLOAN

Contents

1	Introduction	2
2	Fonctionnement détaillé du projet	2
2.1	Récupération initiale des données historiques	2
2.2	Automatisation des données intraday	2
2.3	Traitement et affichage interactif des données	2
2.4	Déploiement en ligne via AWS	3
3	Méthodes de travail adoptées	3
4	Résultats obtenus	3
5	Conclusion et futures améliorations	3

1 Introduction

Dans le cadre de ce projet, nous avons décidé de créer un outil concret permettant à chacun de suivre en direct les évolutions du CAC40. Pour cela, nous avons choisi d'utiliser les données du site **Bourse Direct**, reconnu pour la fiabilité de ses informations financières. Nous voulions non seulement récupérer ces données, mais surtout les exploiter de manière claire et intuitive à travers un dashboard interactif accessible à tous. Le cœur de notre démarche était l'automatisation complète : du scraping initial des données historiques jusqu'à la mise à jour continue toutes les cinq minutes des données intraday, en passant par un rapport synthétique quotidien généré automatiquement à 20h.

2 Fonctionnement détaillé du projet

Notre projet est structuré autour de quatre étapes principales : récupération initiale des données historiques, automatisation des données intraday, traitement et affichage interactif des données, ainsi que la mise en ligne du dashboard via AWS.

2.1 Récupération initiale des données historiques

La première étape du projet consistait à obtenir une base complète de données historiques pour alimenter notre dashboard. Nous avons écrit un script Bash `scrape_history.sh` qui interroge directement l'API de Bourse Direct `history`, afin de récupérer l'intégralité des données historiques du CAC40 disponibles depuis l'année 2000. L'idée derrière cette étape était de disposer d'une base de données pour analyser les tendances historiques et améliorer la pertinence du dashboard. Une fois récupérées, ces données sont stockées localement sous forme de fichiers JSON puis immédiatement converties en fichiers (CSV) `data_history.csv`.

2.2 Automatisation des données intraday

Pour garantir une actualisation continue et automatique des données, nous avons conçu un second script Bash nommé `scrape_intraday.sh` qui interroge aussi un autre l'API (intraday) de Bourse Direct `intraday`. Ce script est exécuté toutes les cinq minutes par une tâche Cron `cron_intraday.txt`. L'idée de cette étape était d'assurer la fraîcheur constante des données affichées sur le dashboard. Le script collecte les dernières informations disponibles sur Bourse Direct, utilise des expressions régulières (Regex) pour extraire précisément les valeurs pertinentes telles que les prix d'ouverture, de clôture, le plus haut et le plus bas du jour, puis les organise immédiatement dans un autre fichier (CSV) `data_intraday.csv` prêts à être utilisés.

2.3 Traitement et affichage interactif des données

Les fichiers CSV produits sont intégrés dans notre application Web Dash (il permet de développer des applications Web interactives de visualisation entièrement en Python). Le site s'organise en plusieurs blocs bien distincts. En haut, un bandeau affiche les principales informations de la dernière ligne de données intraday : le prix d'ouverture, le prix de clôture, le plus haut et le plus bas. Vient ensuite une section dédiée au résumé statistique de la journée : on y retrouve la volatilité calculée sur l'ensemble des données

disponibles, ainsi que les ratios de Sharpe et Sortino, qui sont recalculés chaque soir à 20h dans le cadre du rapport journalier.

Puis une autre section graphique où les utilisateurs peuvent observer l'évolution horaire du CAC40 et de sa volatilité :

- tout au long de la journée
- Suivant la période définie par le calendrier (c'est modifiable)
- Suivant la subdivision souhaitée (journalière, hebdomadaire et mensuelle)

Le graphique est mis à jour automatiquement.

2.4 Déploiement en ligne via AWS

Afin de rendre accessible notre solution à un large public, nous avons choisi AWS pour héberger notre dashboard. Le choix d'AWS a été motivé par sa fiabilité, sa flexibilité et son accessibilité. C'était nouveau pour nous mais on s'est facilement adapté. Notre site est accessible à l'adresse suivante : <http://13.60.74.31:8050/> et notre repository Github sur `Project_ADV_Python_git_linux`

3 Méthodes de travail adoptées

Notre méthodologie de travail s'est basée sur une collaboration étroite et structurée, nous avons partagé équitablement les tâches donc du 50/50.

- **Versionnement avec GitHub** : Nous avons constamment utilisé GitHub pour suivre nos développements et faciliter notre coopération.
- **Réunions régulières** : Chaque deux jours, nous nous réunissions pour définir les objectifs précis et vérifier l'avancement du projet.
- **Communication efficace** : Teams et Whatsapp nous permettait de communiquer en permanence et de résoudre rapidement toute difficulté rencontrée et parfois quand le besoin se faisait sentir nous nous rencontrions au pôle pour mieux échanger.

4 Résultats obtenus

Notre dashboard est aujourd'hui opérationnel, répondant parfaitement aux exigences initiales fixées. Il constitue un outil pratique et puissant, facilitant l'accès à des données financières actualisées et analysées en temps réel.

5 Conclusion et futures améliorations

Ce projet nous a apporté une expérience précieuse dans l'automatisation de processus financiers et le développement web. Nous envisageons plusieurs axes d'amélioration pour rendre notre dashboard encore plus robuste et performant :

- Intégration de multiples sources de données financières pour diversifier et sécuriser notre système.
- Développement de fonctionnalités prédictives grâce à des algorithmes de Machine Learning, notamment le réseau de neurones.

Nous sommes convaincus que ces améliorations permettront à notre projet de continuer à évoluer positivement et à rester pertinent sur le long terme. Et nous remercions notre prof BOGROFF Alexis et notre chargé de TD MAURICE Steven pour ce projet très enrichissant et pour l'accompagnement tout au long de ce semestre.

Références et ressources

- Bourse Direct
- Documentation Dash
- Répertoire GitHub du projet