Saibi Mohamed Encadré par Mr.Célestin Coquidé 22/04/2025

Rapport Individuel - Projet CC3 Printemps : Exploration du jeu de données ORBITAAL pour l'analyse des transactions Bitcoin

2024-2025

Saibi Mohamed Encadré par Mr.Célestin Coquidé 22/04/2025

Introduction

L'objectif principal de ce projet est d'analyser les transactions Bitcoin à l'aide du jeu de données ORBITAAL couvrant la période de 2009 à 2021. Ce projet a pour ambition d'explorer et d'étudier en profondeur les dynamiques du réseau Bitcoin à travers ses données de transaction.

Sur le plan personnel, ce projet représente une opportunité précieuse de découvrir le domaine de la science et de l'ingénierie des données des domaines vers lesquels je souhaite m'orienter tout en approfondissant mes compétences en programmation Python, un langage que je n'avais que peu eu l'occasion d'utiliser avant. C'est également l'occasion d'aborder des concepts avancés tels que le traitement de données massives (Big Data) et la visualisation de données.

Compréhension du contexte : Blockchain & Bitcoin

En premier lieu, il a fallu avant d'entamer le travail comprendre le fonctionnement de la blockchain et la structure logique du réseau Bitcoin pour avoir une base solide. Cela a été réalisé grace a un long travail de recherche et de lecture des papiers de recherche qui étaient fait en relation avec ce sujet

Analyse Exploratoire : problème et solution

Un axe majeur de mon travail a été l'exploration des transactions disponibles et leur visualisation de manière interactive. Au départ, la tâche me paraissait simple : je pensais pouvoir lire le fichier, ecrire quelques lignes de code, et obtenir mes graphiques sans difficulté. Cependant, la réalité s'est révélée bien différente, et plusieurs problèmes sont survenus .

Durée longue de test et erreur de connexion de session Spark:

Le premier défi rencontré concernait le temps d'exécution des requêtes. Même pour une simple agrégation, certaines pouvaient prendre entre 5 et 7 minute. Ce temps de traitement excessif entraînait fréquemment, au bout de quelques requêtes, une erreur de type (ConnectionRefusedError) , souvent liée à une surcharge de la mémoire. Pour contourner ce problème, je devais relancer l'exécution complète du fichier, ce qui a considérablement ralenti ma progression, notamment lors des premières phases du projet..

Première Solution : Nettoyage Ciblé

Pour contourner ce problème, j'ai dû revenir à l'un des fondements de l'analyse de données : la préparation des données. En effet, je réalisais des requêtes directement sur des fichiers contenant près de 200 millions de lignes, sans les filtrer ni les orienter en fonction de mes objectifs. Grâce à

une documentation approfondie, j'ai compris l'importance de retravailler les données en amont afin d'optimiser leur traitement, réduire l'utilisation de la mémoire et limiter les erreurs de connexion, tout en diminuant les temp d'exécution

La première étape a consisté à exclure les mineurs « c'est-à-dire les transactions de validation ou de frais »qui n'étaient pas pertinentes pour mon analyse du réseau de transactions réel. Cette opération a permis d'alléger conssidérablement le dataset en supprimant un grand nombre de lignes, tout en conservant l'essentiel de la valeur en satoshis échangés

Deuxiéme Solution : Exploration et Distribution des Bitcoins

2.1Pourcentage Cumulé et échantillons

J'ai ensuite décidé de me concentrer uniquement sur une portion réduite du dataset, en le triant selon le montant des transactions. Bien que cette approche puisse sembler risquée en termes de perte d'informations, un graphique que j'ai réalisé m'a permis de justifier ce choix. En visualisant la répartition des montant de transaction et leur pourcentage cumulé, j'ai constaté qu'environ 60 % des bitcoins étaient concentrés entre les 20 premières adresses. Cette forte concentration m'a donc permis de travailler éfficacement sur un échantillon restreint, tout en conservant une représentativité significative du réseau

2.2Courbe de Lorenz et indice de Gini

L'analyse de la courbe de Lorenz, accompagnée du calcul de l'indice de Gini, est venue renforcer la validité de mon choix méthodologique. En effet, un indice de Gini élevé « éloigné de 0 » indique une forte inégalité dans la répartition des bitcoins, confirmant que la majorité des fonds est concentrée dans un nombre très restreinte d'adresses. Cela justifie pleinement le focus sur les top adresses, tout en conservant la pertinence de l'analyse.

Troisieme solution : Optimisation de la gestion de la mémoire

Une autre solution que j'ai adoptée pour résoudre ce problème était d'ajuster les paramètres de la session en augmentant l'espace mémoire alloué. Cela m'a permis de réduire les erreurs liées à l'abus de memoire et d'éviter les coupures fréquentes des connexions, assurant ainsi une exécution plus fluide et plus rapide des requêtes.

Conclusion

Ce projet m'a permis de franchir plusieurs étapes importantes dans mon apprentissage de Python et Pyspark ainsi que des concepts fondamentale dans la science , l'analyse et l'ingénierie de données

Saibi Mohamed Encadré par Mr.Célestin Coquidé 22/04/2025

ANNEXE

```
total_transactions = df2.count()
print(f"Nombre total de transactions : {total_transactions}")
```

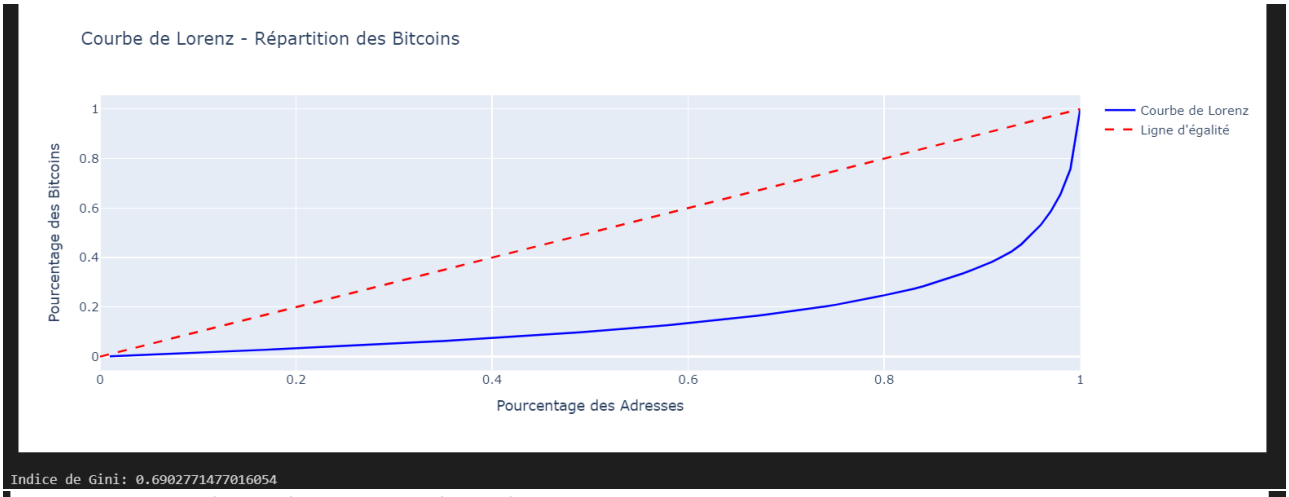
✓ 1.4s

```
[Stage 10:==============================>                    (20 + 8) / 37]
Nombre total de transactions : 292969302
```

```
transactions_by_sender = df2.groupBy("SRC_ID").count()
transactions_by_sender.show(10)  # Affiche les 10 premiers émetteurs dans le tableau
```

34.9s

```
Total sans mineurs (Satoshi) :  74965419710627
Total avec mineurs Satoshi :  74967004055431
Transacation des mineurs en Satoshi:  1584344804
```




```
Indice de Gini: 0.6902771477016054
```