

PROJET CRYPTOBOT

Contexte :

De nos jours, le monde des cryptos commence à prendre une place importante et grossi. Il s'agit tout simplement de marchés financiers assez volatiles et instables se basant sur la technologie de la Blockchain.

Le but principal de ce projet est de créer un bot de trading basé sur un modèle de Machine Learning et qui investira sur des marchés crypto.

Participants au projet : Thomas Saliou, Florent Rigal, Philippe Kirstetter-Fender et Nancy Frémont

1^{ère} étape : Récolte des données, extraction et transformation

Explication des traitements appliqués :

Lors de la première étape de récolte des données, nous avons collecté et préparé les données de marché nécessaires à l'entraînement du futur modèle de prédiction.

Deux types de données ont été récupérées : des données historiques provenant de l'API publique de Binance (url : <https://api.binance.com>) et des données en temps réels (via websocket de Binance) ou en streaming (url : <wss://stream.binance.com:9443/ws>).

Via l'API, le site Binance fournit des données publiques comme les prix actuels(ticker/price), les historiques des trades, le carnet d'ordres (/depth), les volumes sur 24h et les données de marchés (klines, etc.). Les Klines ou chandeliers (/api/v3/klines) représentent l'évolution de prix d'un actif (paire ex : BTC-USDT, BTC-ETH, etc.) sur une période donnée. Chaque chandelier représente un OHLCV (Open High Low Close Volume), soit le prix d'ouverture de la période, le prix le plus haut atteint sur la période, le prix le plus bas atteint sur la période, le prix de clôture et le volume de la paire sur la période. Les données sont enregistrées dans des fichiers csv.

Via le websocket du site de Binance, il est possible d'extraire des données comme un changement de prix en 24h, les prix, les OHLCV par minute (ce qui nous intéresse ici). Les données sont enregistrées dans un fichier json.

Pour la création des scripts, nous avons utilisés les librairies requests (pour interagir avec des APIs), pandas (pour le traitement des données), pyspark (permet le traitement efficace de données en continu) et websocket-client (permet de se connecter à des serveurs websocket, d'envoyer et recevoir des messages en temps réels).

Une fonction générique de récupération a été développée pour pouvoir extraire les données depuis n'importe quel marché (exemple : BTC-USDT, ETH-USDT) en s'appuyant sur des paramètres dynamiques tels que le nom du marché ou symbol, l'intervalle de temps (1h) et la limite de transaction. Une fois les données brutes collectées, une fonction de traitement (en transformant les données en dataframe pandas) a été mise en place afin de les rendre exploitables : conversion des timestamps (moment ouverture/fermeture d'un chandelier) en format date, normalisation des prix, gestion des données manquantes, conversion des données numériques en float et suppression des colonnes inutiles.