

Description du Projet : Gestion d'un Fond Multi-Asset

Luna Baladi

Objectif du projet :

Le projet consiste à gérer un fonds multi-asset en mettant en place un pipeline de données robuste et automatisé, permettant une administration efficace et optimisée du fonds. La période d'évaluation du fonds s'étend du 01-01-2023 au 31-12-2024. Le processus d'investissement dans ce fonds se fait chaque lundi, où les décisions d'achat ou de vente d'actifs sont prises. Un "deal" est défini comme l'achat ou la vente d'un actif spécifique au sein du portefeuille.

Acquisition des données (Module : `data_collector.py`) :

Le module `data_collector.py` est responsable de la collecte des données nécessaires au bon fonctionnement du fonds. Une fois téléchargées, les données sont traitées pour les rendre exploitables. Les données historiques des actifs sont récupérées depuis 2010 sur Yahoo Finance.

Configuration des Clients et des Portefeuilles :

Le fonds est destiné à trois profils de clients, chacun ayant un portefeuille adapté à son profil de risque. Les portefeuilles sont considérés comme autofinancés, ce qui signifie qu'aucune contribution ou retrait d'argent n'est effectué durant la période d'évaluation. La valeur du portefeuille dépend de la performance des actifs qui le composent. Chaque client débute avec un portefeuille de 1 000 000 EUR, et à partir de ce portefeuille initial, il peut acheter ou vendre des actifs en fonction des performances de ceux-ci.

Lors de la vente d'un actif, sa part dans le portefeuille est réduite de moitié. L'argent généré par cette vente est alors réinvesti de manière équilibrée dans les autres actifs du portefeuille, en fonction de leur prix actuel sur le marché.

Pour créer un portefeuille initial adapté à chaque profil de risque, j'ai appliqué une optimisation sous contrainte, que l'on retrouve dans le module `optimisations.py`. Ce processus d'optimisation vise à établir la meilleure combinaison d'actifs pour chaque client, en fonction de ses contraintes spécifiques, telles que la minimisation du risque ou la maximisation du rendement.

Optimisation (Module : `optimisations.py`) :

Dans le module `config.py`, des actifs spécifiques ont été définis pour chaque profil de risque, en diversifiant les classes d'actifs et en les adaptant aux stratégies des clients. Pour chacun des clients, un portefeuille initial est optimisé en fonction de son profil de risque. L'optimisation est réalisée sous contrainte, visant à minimiser le risque pour les clients à faible volatilité, ou à maximiser le rendement pour les clients à rendement élevé.

Stratégies de Gestion des Portefeuilles (Module : **strategies.py**) :

Les stratégies sont basées sur des modèles de machine learning, et plus précisément sur un modèle de Random Forest, utilisé pour les trois profils de risque afin de garantir une approche cohérente et solide.

Le Random Forest est un algorithme d'assemblage puissant qui permet de capturer des relations complexes entre les variables (comme la volatilité, le rendement, les indicateurs techniques, etc.) tout en étant moins sujet au surapprentissage que d'autres modèles plus complexes. Ce modèle est idéal pour les prévisions basées sur de nombreuses variables, comme c'est le cas pour nos stratégies de gestion de portefeuille.

1. Stratégie Low Risk :

L'objectif de cette stratégie est de maintenir la volatilité annualisée du portefeuille autour de 10%. Pour y parvenir, un actif est vendu si sa volatilité hebdomadaire dépasse 15% et si son rendement hebdomadaire est négatif. À l'inverse, un actif est acheté si sa volatilité hebdomadaire est inférieure à 15% et si son rendement hebdomadaire est positif. Si ces conditions ne sont pas remplies, l'actif reste dans le portefeuille sans modification.

Cette stratégie vise à éviter un portefeuille trop risqué, en se basant sur des critères de volatilité et de performance.

Cependant, une difficulté rencontrée est que la volatilité du portefeuille en général était plus faible que prévue, en raison des corrélations entre les actifs. Par conséquent, la volatilité cible de 10% n'a pas été atteinte, ce qui a légèrement pénalisé le rendement global.

2. Stratégie Low Turnover :

La stratégie Low Turnover cherche à minimiser les transactions, afin de réduire les coûts associés aux achats et ventes d'actifs. Les actifs sont sélectionnés pour être stables, diversifiés et appartiennent à des secteurs considérés comme sûrs. Pour cette stratégie, j'ai intégré des facteurs macroéconomiques tels que l'indice des prix à la consommation, le VIX, et les taux d'intérêt, ainsi que des indicateurs techniques tels que le RSI et la volatilité historique.

La condition d'achat/vente est la suivante :

- Si la volatilité sur 5 jours est inférieure à celle sur 10 jours ET si le rendement à 10 jours est positif, acheter.
- Si la volatilité sur 5 jours est supérieure à celle sur 10 jours ET si le rendement à 10 jours est négatif, vendre.
- Si le RSI est inférieur à 30, acheter (surachat).
- Si le RSI est supérieur à 80, vendre (survente).

Cette stratégie n'a pas donné les résultats escomptés en termes de performance. Un ajustement des critères d'achat et de vente pourrait être nécessaire pour améliorer la rentabilité tout en minimisant le turnover des actifs.

3. Stratégie High Yield Equity Only :

La stratégie High Yield se concentre sur l'optimisation du rendement, sans se soucier de la volatilité. Aucun critère de gestion du risque n'est appliqué dans cette stratégie. Elle est exclusivement axée sur l'investissement en actions (equity). L'objectif est d'identifier les actifs offrant un rendement hebdomadaire élevé, sans contrainte de volatilité.

J'ai utilisé des indicateurs pour déterminer les décisions d'achat ou de vente, basés sur les rendements hebdomadaires et les volumes de transactions. En raison de contraintes d'accès aux données fondamentales (PER et autres ratios financiers, qui n'étaient que disponible à un moment donné et non sur tout l'historique), je me suis limité à des indicateurs techniques plus simples. J'avais également l'intention d'utiliser les données de recherche Google pour chaque action afin de suivre les tendances du marché, mais j'ai rencontré des difficultés avec l'API, et avec d'autres API qui ne fournissent pas d'informations pertinentes via la version gratuite. En conséquence, j'ai dû me limiter aux indicateurs techniques plus classiques.

La règle d'achat/vente appliquée est la suivante :

- Si le rendement hebdomadaire dépasse un seuil (quantile 75%), acheter.
- Si le rendement hebdomadaire est inférieur à un seuil (quantile 25%), vendre.
- Sinon, ne rien faire.

Cette stratégie a montré de bons résultats en termes de rendement, bien que la volatilité ait été plus élevée que celle des autres portefeuilles, ce qui était prévu, car aucune contrainte sur la volatilité n'a été imposée.

Modélisation et Validation :

Les données antérieures à 2023 ont été divisées en 80% pour l'entraînement et 20% pour le test, en respectant la structure chronologique des données. Le modèle a été entraîné sur 80% des données disponibles et testé sur le reste pour évaluer sa capacité à prédire les décisions d'achat et de vente. Après l'entraînement, les stratégies ont été appliquées sur la période d'observation (01-01-2023 à 31-12-2024).

À chaque itération, les décisions prises (achats/ventes) sont mises à jour dans les tables de la base de données : **portfolios**, **deals**, et **clients**. La valeur des portefeuilles des clients est mise à jour en conséquence.

Évaluation des Performances :

Les rendements des portefeuilles sont calculés à partir de la table `returns` et utilisés pour générer des statistiques et des graphiques de performance. Des graphiques tels que les rendements cumulés, la volatilité historique, et les drawdowns sont générés à l'aide de Python et enregistrés sous forme d'images. Ces résultats sont automatiquement enregistrés dans un rapport détaillé dans un fichier reports.

Conclusion :

Ce projet démontre comment une gestion de fonds multi-asset peut être automatisée avec des techniques avancées d'optimisation, de machine learning et d'analyse de performance. Il illustre l'importance de l'adaptation des stratégies de gestion aux profils de risque des clients, en combinant des décisions basées sur des facteurs techniques et macroéconomiques, et en évaluant rigoureusement les performances à travers des outils statistiques et graphiques.

