

*Rapport Technique d’évaluation*

Py\_OLPOS

Python Online Portfolio Optimisation Strategies

Promotion octobre 2021

Participants :

Marouan Maghzaoui

Tudor Haruta

# Contexte

Le problème de l’optimisation du portefeuille est un problème difficile pour des algorithmes de Machine Learning, algorithmes statistiques ainsi que pour la finance computationnelle. Néanmoins, la solution de ce problème permettra l’élaboration et la gestion automatisée d’un portefeuille financier, composé de différents actifs, sans l’impact de l’émotion humaine, qui exerce souvent des influences négatives sur la profitabilité d’un portefeuille.

Posséder un portefeuille varié des actions, indices et devises d’état permettrait la protection d’un patrimoine contre l’inflation ainsi que contre la volatilité si le portefeuille est suffisamment varié. Si, par exemple, certaines actions perdaient de la valeur, les autres composants du portefeuille pourraient les compenser. Cela implique choisir des parties composantes le moins corrélés entre elles, mais aussi ceux que l'on considère avoir du potentiel de croissance. La répartition des différentes parties composantes doit aussi être faite en fonction du profil de risque du détenteur du portefeuille, s’il souhaite être exposé à plus ou moins de volatilité.

Ce projet est purement éducatif et a pour but notre familiarisation avec ce sujet, ainsi que faire nos premiers pas dans le monde du Machine Learning et de la Data Science. Nous ne conseillons pas l’utilisation de ces algorithmes dans le but de faire des investissements financiers ou du trading.

Objectifs

L’objectif de ce projet consiste à créer un modèle d’allocation de portefeuille qui adapte sa stratégie “online”, qui va effectuer des achats et ventes des actifs en fonction de ses prédictions pour maximiser le gain du portefeuille.

En partant de la philosophie que tous les modèles sont faux, certains sont utiles, cet algorithme « pourrait » servir d’appui ou indicateur pour effectuer des achats et ventes d’actions sur l’échelle journalière, même si on le déconseille fortement.

Nous allons nous poser l’objectif suivant : créer un modèle de Machine Learning qui essaye de trier des actions en termes de gain ou de pertes le jour prochain. Avec ce classement on change la répartition des actions dans portefeuille pour maximiser sa valeur.

Nous sommes tous les deux intéressés par le monde de la blockchain et des cryptomonnaies avec des petites expériences d’investissement et/ou de trading. Cela nous a permis de gratter la surface du monde de la finance, mais notre connaissance du sujet reste limitée.

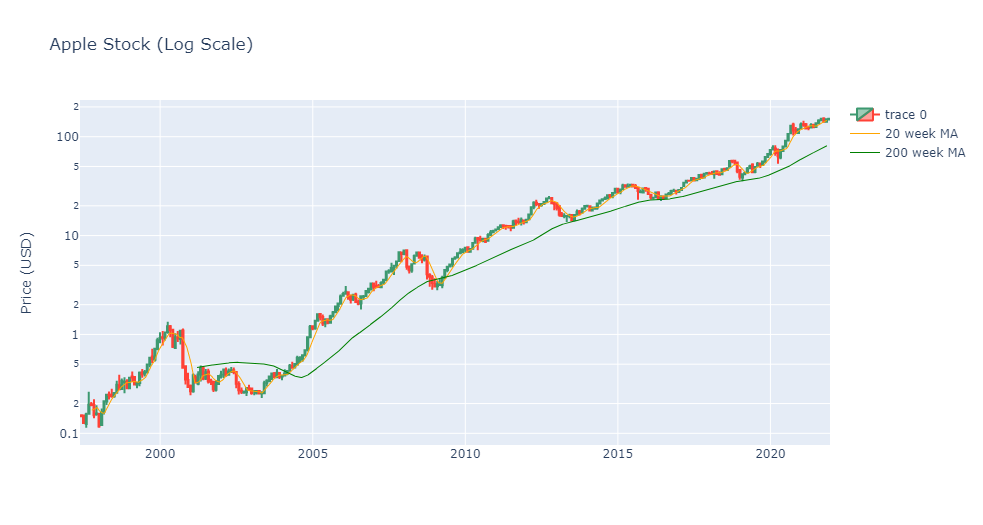
Nous n’avons pas de connaissance de projets similaires dans notre entourage, mais nous avons pu regarder des approches statistiques a la solution de ce problème implémentés dans la librairie Universal algos disponible sur GitHub.

Data

## Cadre

Pour obtenir nos jeux de données, nous avons utilisé l’API de yahoo finance qui nous a permis de télécharger les historiques des prix des actions et indices qu’Nous avons considérés comme intéressants. Nous avons utilisé les valeurs d’ouverture, fermeture, maximum et minimum de chaque jour pour 6 actions (Google, Amazon, Apple, Facebook, Pfizer et Tesla) qui étaient les cibles initiaux*.*

Les graphiques suivants montrent l’évolution de la valeur de ces 6 actions cibles en dollars depuis l’an 1997. Nous avons utilisé la fonction « resample('M') » pour regrouper les données journalières en mois. Nous avons aussi utilisé la méthode de plot Candlestick pour afficher le plus bas et le plus haut de chaque mois et la couleur vert ou rouge pour indiquer s’il y a un gain ou une perte dans chaque mois. Nous avons tracé deux moyennes mobiles « *Moving Averege* » (Ma) la première en jaune sur 20 semaines et la deuxième sur 200 semaine (à peu près 5 ans)

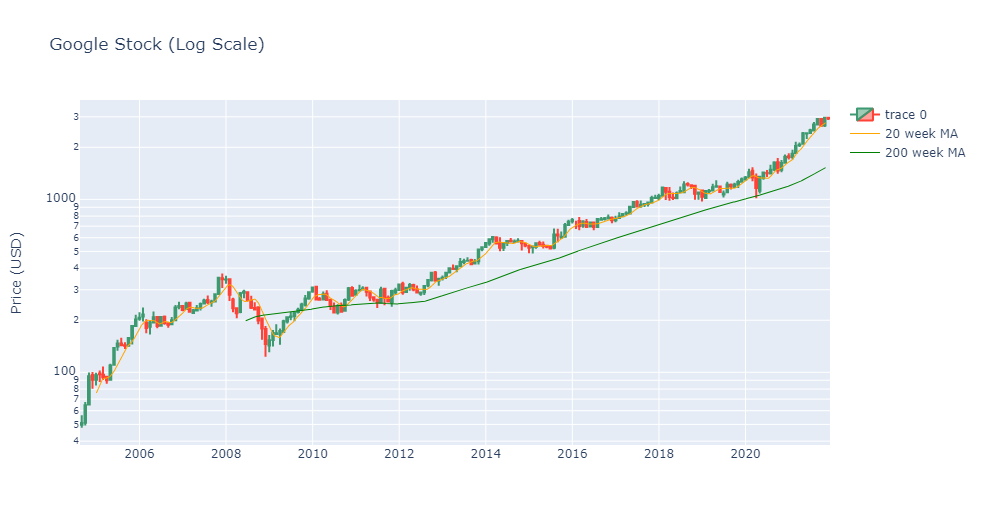


***Fig 1. Valeur de l’action de Apple depuis 1997***

******

***Fig. 2 Valeur de l’action de Amazon depuis 1997***

***Fig. 3 Valeur de l'action de Pfizer depuis 1997***



***Fig. 4 Valeur de l'action de Google depuis 2004***



***Fig. 5 Valeur de l'action de Tesla depuis 2010***



***Fig. 6 Valeur de l'action de Facebook depuis 2012***

Aussi nous avons utilisé les valeurs d’ouverture, fermeture, maximum et minimum des indexes de marché tel que NASDAQ, NYSE, VIX (volatilité de marché) et DXY (Dollar strength index). Nous avons ajouté aussi ces données (taux de change Euro / Dollar, Bitcoin / Dollars, prix de Baril de pétrole / dollars et pris de l’Or par rapport au dollars) (voir l’annexe).

Ces données représentent l’évolution du prix pour chaque journée de travail (du lundi au vendredi) et sont globalement comprises entre 1971-02-05 et 2021-11-03.

Les données sont disponibles principalement sur le site de Yahoo.com en libre accès. Quelques autres indices ont été téléchargé d’autre site.

La volumétrie de jeu de donnée initiale est 12800 lignes x 75 colonnes au début mais à la fin on a utilisé que 6157 lignes (depuis l’introduction de l’action Amazon)

Nous avons fait le diagramme de corrélation (Fig. 17) de toutes nos variables. Pour cela, nous avons fait un nouveau tableau contenant que les moyennes mobiles de nos variables. Ce diagramme nous indique une forte corrélation entre les actions des entreprises IT. On observe que le VIX et le DXY exercent une faible influence sur la grande majorité des actions. Les actions sont aussi corrélées avec les indices NASDAQ et NYSE, mais cela est attendu car leur prix fait aussi partie de ces indices par construction. Mais même avec ces corrélations, nous avons eu des problèmes à trouver une variable cible qui nous aide à optimiser le portefeuille.

## Pertinence

Pour nettoyer les données (généralement des valeurs naan pour des actions ou indexes qui ne sont pas encore introduits dans la bourse), nous avons remplacé les naan par des 0.

Nous avons aussi ajouté des colonnes qui indiquent les gains ou les pertes de chaque action des 6 que nous avons. Nous avons commencé par classer les 6 actions cibles au début mais à la fin nous avons décidé de laisser que les 3 plus anciens (Apple, Pfizer et Amazon) pour diminuer la quantité de donnée à entrainer. Nous avons créé une variable qui est une combinaison linéaire entre 0 et 5 en se basant sur les rangs des actions cible en utilisant cette formule.

**target=( Apple\_Rank-1) +( Pfizer\_Rank-1)\*3 + ('Amazon\_Rank-1)\*9**

Cette formule nous donne 6 valeurs (5,7,11,15,19,21) et avec un dictionnaire on obtient une distribution entre 0 et 5.

A la fin nous avons fait un décalage de cette dernière variable sur la ligne précédente pour qu’on essaye de déterminer cette variable avec du Machine Learning.

# Projet

## Classification du problème

Nous avons au début essayé faire résoudre le problème avec la régression de 6 vecteurs (déduire la bonne répartition des actions dans le portefeuille) mais avec les mauvais scores qu’on a obtenu qui était toujours négative, nous avons changé le problème vers une classification sur 6 valeurs.

La tâche de Machine Learning est de déduire le rang des gains des actions cible pour savoir quelle action on doit acheter et quelle action on doit vendre d’un jour à l’autre.

La métrique principale dont nous avons choisi pour noter notre modèle est l’accuracy via la méthode score.

Nous avons aussi créé une fonction « value wallet » qui déduit la valeur du portefeuille à la fin du Machine Learning en attribuant chaque jour à la 1ere action 4/7 de valeur du portefeuille, la deuxième 2/7 et la 3eme 1/7.

Pour la partie entrainement on a pris les premier 80% des données dans l’ordre et on a laissé les derniers 20 % pour le test.

## Choix du modèle & Optimisation

Nous avons essayé différents modèles. Au début un modèle simple de « LogisticRegression » (LR) .Puis nous avons testé beaucoup de modèles de « K-nearest neighbors » ( KNN) en changeant le nombre de voisins de 1 à 50 et la valeur p de 1 à 10. Nous avons utilisé une boucle *for* au lieu d’un GridSearchCV au début pour pouvoir à la fois voir le score et la valeur de portefeuille à la fin.

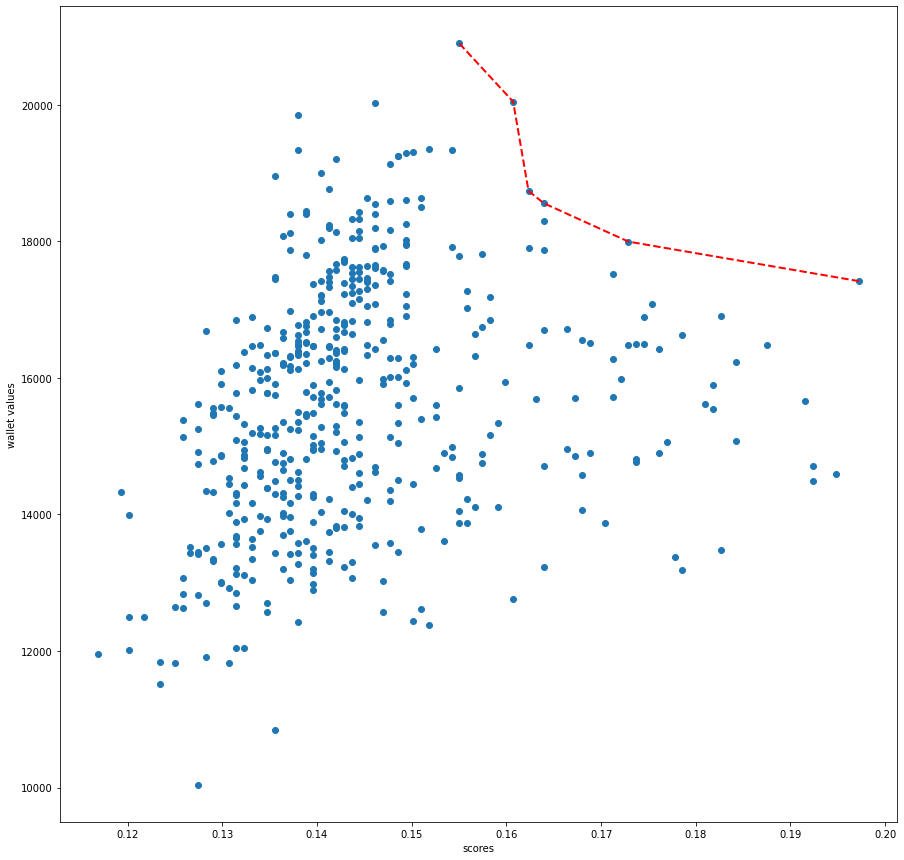
*Chart, scatter chart

Description automatically generated*

***Fig. 7 Nuage de point des résultats des modèles KNN (scores /wallet values)***

On voit dans le nuage des points des résultats que les scores varient entre 0.1 et 0.22 et les valeurs des portefeuilles varient entre 9000 et 20000 dollars. La ligne rouge est la ligne de pareto. C’est sur cette ligne qu’on trouve les meilleurs modèles.

Nous avons aussi fait la même chose avec « Random Forest Classifier » (RF) . Nous avons testé avec des différents nombres d'estimateurs (entre 100 et 1000 avec un pas de 100) et aussi en variant le paramètre « *min\_samples\_leaf* » entre 1 et 50.



***Fig. 8 Nuage de point des résultats des modèles RF (scores /wallet values)***

On voit dans le nuage des points des résultats que les scores varient entre 0.11 et 0.20 et les valeurs des portefeuilles varient entre 1000 et 21000 dollars. La ligne rouge est la ligne de Pareto. C’est sur cette ligne qu’on trouve les meilleurs modèles. Dans ce modèle on trouve le meilleur gain de portefeuille (plus de 10 milles dollars)

Aussi nous avons testé rapidement quelques autres modèles tels que « Passive Aggressive Classifier », « Perceptron », « GradientBoostingClassifier » et « Support Vector Machine »  (SVM)

Nous avons gardé les modèles KNN et RF parce qu’ils donnent des meilleurs scores ou des meilleures valeurs de portefeuille.

# Description des travaux réalisés

## Bibliographie

Au début nous avons utilisé les articles de Borodin et al, 2004 et Li et al, 2012 qui décrivent les algorithmes Anticor et OLMAR respectivement pour nous informer sur ce sujet et nous avons exploré le répertoire Universal-algos sur GitHub qui contient ces algorithmes. Mais nous avons décidé contre ces approches et d’élaborer des algorithmes de régression et, après, de classification pour appliquer nos nouvelles connaissances de Machine Learning.

## Difficultés rencontrées lors du projet

Le principal problème que nous avons eu c’est le choix de la variable cible.

Au début nous avons opté vers une target qui est composé de vecteurs avec la part de chaque action dans notre portefeuille. Mais les scores étaient négatifs. Après nous avons changé vers une combinaison linéaire des rangs des actions. Mais aussi le nombre des classes était très élevé avec 6 actions (720 classes) avec des scores <0.01 et même avec 4 actions (24 classes) nous obtenons des scores <0.1. A la fin nous avons opté pour un modèle avec 3 actions (Apple, Pfizer et Amazon) ou nous avons que 6 classes et des scores qui atteignent 0.22 > 0.166 qui est la valeur du choix aléatoire.

# Bilan & Suite du projet

Nous avons analysé les deux meilleurs modèles, le modèle avec le meilleur score (KNN avec n=31 et p =2) ou le score =0.22 et le modèle avec la plus grande valeur de portefeuille (RF avec n\_estimators = 100 et min\_samples\_leaf=39 ) pour donner une valeur de portefeuille finale égale à 20896 dollars.

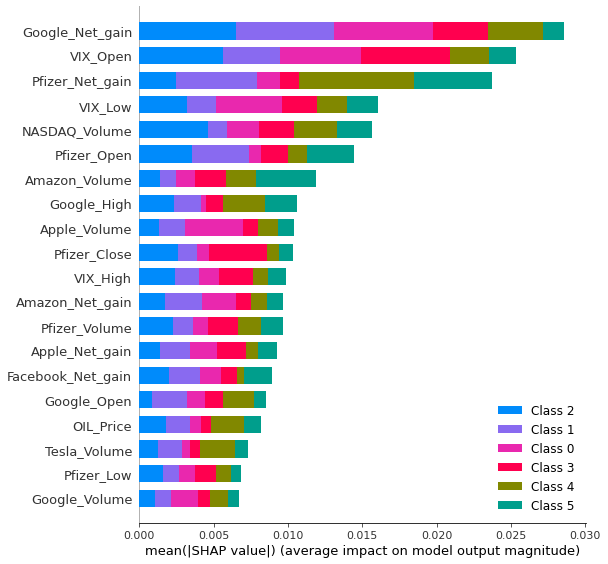
Pour l’analyse des 2 modèles, nous avons utilisé la bibliothèque Shap qui analyse l’impact de chaque variable sur les résultats.

*Graphical user interface, text

Description automatically generated*

***Fig. 9 Résultat de shap du modèle avec le meilleur score***

On voit que pour le premier modèle (KNN) les indices qui ont une relation avec le bitcoin et surtout le volume de bitcoin vendue chaque jour qui a le plus d’influence sur ce modèle.

**

***Fig. 10 Résultat de shap du modèle avec le meilleur score***

*Dans le deuxième modèle (RF) on vois que les facteurs sont diversifié un peu venant de diffèrent actions et indices (Google, VIX, Pfizer,nasdaq..). On vois aussi que les impact sont très faible (0.03) pour le 1er facteur ici par rapport à 0.6 pour le premier facteur dans le modele KNN.*

Les modèles de Machine Learning que nous avons créé malgré leur faible score, fournissent des gains au niveau de la valeur de portefeuille qui dépasse 2 fois la valeur initiale. Pourtant, c’est loin de modèle optimale (ou le score = 1) ou avec une valeur initiale de portefeuille de 10000 dollars fini par 176286 dollar avec un gain de 17.6 fois.

Etant donne qu’on effectue des prédictions d’achat ou de vente pour les prix du jour au lendemain, ce modèle pourrait être utile a un trader, même si on le déconseille fortement.

Concernant les pistes à améliorer, il faut peut-être choisir un meilleur modèle de répartition des actions dans le portefeuille à partir de rang au lieu du modèle simple que nous avons choisi.

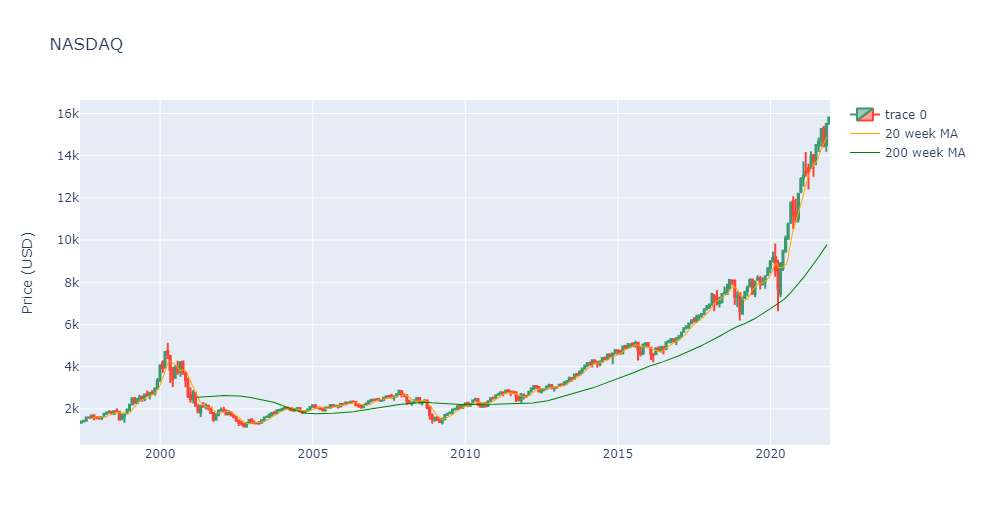
Aussi, nous n’avons pas eu le temps pour tester un modèle de Deep Learning et pour voir si le modèle peut donner des meilleurs scores et des meilleures valeurs de portefeuille.

“En ce monde rien n’est certain, à part la mort et les impôts.” Benjamin Frankin

Dans le cadre de notre modèle, nous avons exclu un aspect assez important, les différentes taxes qu’on doit payer pour effectuer des transactions. Les plateformes d’échange peuvent prendre des taxes jusqu’à plus de 1% de la valeur de la transaction. Si on fait des transactions assez souvent, comme est le cas ici, chaque jour, cette somme devient vite très imposante et va diminuer les possibles gains de l’algorithme, même le rendant pas profitable dans certains cas. De plus, en fonction du pays de résidence, il faut aussi ajouter les impôts et taxes sur les plus-values obtenues. Donc par la suite il faudrait implémenter une méthode qui prend en compte les différentes taxes possibles et choisit si un changement de la proportion des actions du portefeuille est judicieuse ou non.

# 

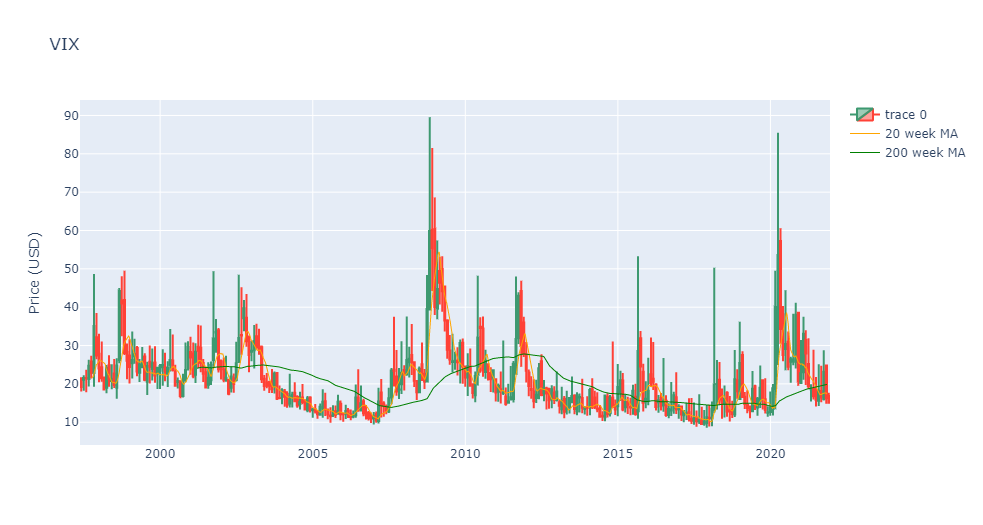
# Annexes



***Fig. 11 Valeur de l’indice NASDAQ depuis 1997***



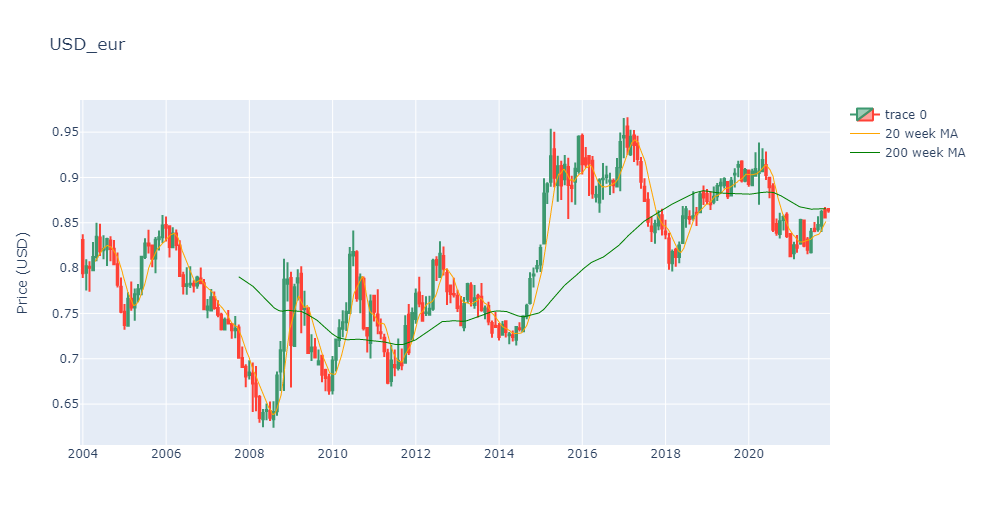
***Fig. 12 Valeur de l’indice NYSE depuis 1997***



***Fig. 13 Valeur de l’indice VIX de Apple depuis 1997***



***Fig. 14 Valeur de l’indice DXY depuis 1997***



***Fig. 15 Valeur de dollars / euro depuis 2004***



***Fig. 16 Valeur de Bitcoin / USD depuis 2014***

Treemap chart

Description automatically generated

## 

***Fig. 17 Heatmap de corrélation***

## Description des fichiers de code

Vous recevrez joint à ce notebook trois notebooks jupyter, un contenant la data visualisation et un autre contenant le modèle, ainsi que le fichier qui regroupe l’ensemble de nos données.