

# Projet d'Apprentissage Statistique Appliqué : Prédiction de ratios de Sharpe d'un portefeuille de stratégies

Tom Ducard, Antonin Devalland

Janvier 2026

## Introduction

Le sujet que nous avons choisi de traiter concerne la prédiction du ratio de Sharpe d'un mix de stratégies quantitatives. Ce projet est inspiré d'un des Data Challenges du Collège de France de Napoleon X. Napoleon Crypto, branche de Napoleon X, est une entreprise spécialisée dans la conception de solutions d'investissement quantitatives, basées sur des algorithmes. Ayant développé plusieurs stratégies performantes, l'entreprise cherche à optimiser l'allocation entre ces stratégies pour maximiser les performances. L'objectif de ce problème est donc d'aider à la construction d'un mélange optimal de stratégies quantitatives d'investissement à partir de l'ensemble des stratégies soumises. Trouver la meilleure allocation parmi les stratégies quantitatives chaque semaine revient à déterminer la combinaison maximisant le ratio de Sharpe (sur les 5 prochains jours de trading).

Le ratio de Sharpe est une métrique de performance qui mesure le rendement excédentaire pour chaque point de risque pris. Nous supposerons un taux sans risque nul. Avec  $Lr_{i,s} = \ln\left(\frac{I_s}{I_{s-1}}\right)$  les rendements logarithmiques d'une stratégie  $i$  à un temps  $s$  (avec une valeur de la stratégie au temps  $s$  qui vaut  $I_s$ ) et  $w_i$  le poids de la stratégie  $i$  dans le portefeuille, le ratio de Sharpe annualisé objectif est calculé selon la formule suivante, légèrement modifiée pour éviter une volatilité trop proche de 0 :

$$S_t^*(w_1, \dots, w_7) = \frac{\frac{252}{5} \sum_{i=1}^7 w_i \times \left( \sum_{s=t+1}^{t+5} Lr_{i,s} \right)}{\text{Max} \left( \sqrt{252 \times \sum_{i=1}^7 \sum_{j=1}^7 w_i w_j \sum_{s=t-20}^{t+5} (Lr_{i,s} - \bar{Lr}_i)(Lr_{j,s} - \bar{Lr}_j)}; 0.005 \right)}$$

L'enjeu est donc, finalement, de prédire le ratio de Sharpe  $S^*$  d'une combinaison donnée sur les 5 prochains jours de trading. Avec les outils de machine learning, nous allons chercher à prévoir ce ratio de Sharpe.

## Les données

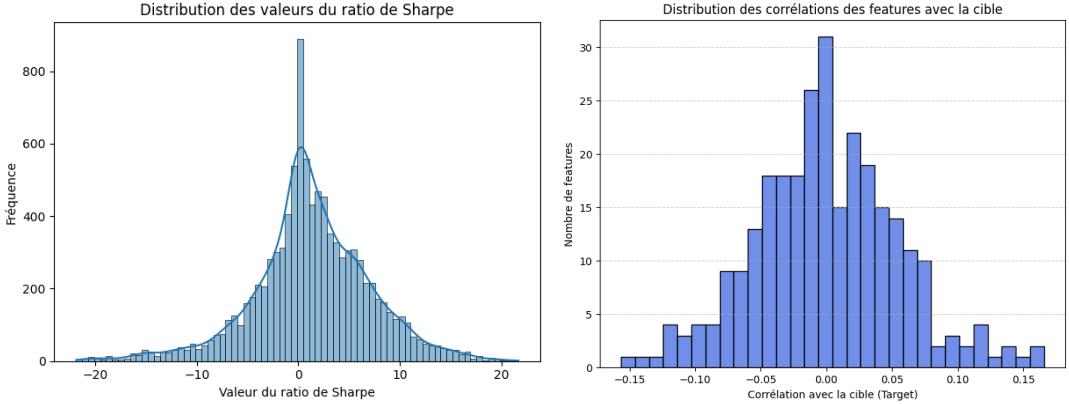
Pour répondre à ce sujet, Napoleon X nous fournit des données concernant 7 stratégies quantitatives qu'ils ont mis en place. Les features initiales (10 000 samples) sont constituées pour chaque observation des poids alloués à chaque stratégie, de la valeur de chacune des stratégies sur 21 jours de trading, et des valeurs de 3 instruments financiers (inconnus) sur 21 jours de trading. Toutes les séries temporelles sont alignées sur une même échelle de temps de 21 jours de trading et sont normalisées à 100 au départ, assurant la comparabilité entre échantillons. En plus de cela, les valeurs du label nous sont fournies, ce sont les ratios de Sharpe de ces mix de stratégies, au nombre de 10 000 (1 par sample). Pour chaque observations, les données sont représentées par les colonnes suivantes.

$$\underbrace{1}_{ID} + \underbrace{7}_{\text{Poids}} + \underbrace{7 \times 21}_{\text{Valeurs des stratégies sur 21 jours}} + \underbrace{3 \times 21}_{\text{Valeurs des instruments sur 21 jours}} = 218 \text{ colonnes}$$

Concernant le traitement des données initiales, qui sont déjà très propres, nous supprimons les quelques doublons et réalisons des vérifications de cohérence économique.

Les valeurs cibles (ratios de Sharpe), dont la distribution est présentée sur le graphique ci-dessous, sont très dispersés, et très élevés en valeur absolue, allant de -22 à 22. Cette distribution étonnante, car beaucoup plus étalée (sur-pondération de valeurs intermédiaires et extrêmes) qu'une distribution de Sharpe classique souvent très compacte autour de zéro, s'explique par plusieurs raisons. Tout d'abord la nature extrêmement volatile des cryptomonnaies, actifs principaux des stratégies de Napoleon Crypto conduit à des stratégies volatiles et

incertaines. Par ailleurs l'horizon considéré est extrêmement court (5 jours) ce qui renforce l'instabilité du ratio de Sharpe annualisé. Notons que la moyenne est à 1,3 ce qui représente une très bonne performance.



## Les features

A partir des données initiales nous construisons de nouvelles features. Premièrement, nous transformons les valeurs des stratégies et des 3 actifs en rendement logarithmique. Cette transformation permet de supprimer l'autocorrélation présente dans ces séries et extraire l'information potentiellement utile pour la prévision d'un ratio de Sharpe. Ensuite, nous calculons les covariances entre les rendements des stratégies d'une même observation et les volatilités de ces rendements. Ces variables sont nécessaires puisque les variances et covariances des stratégies constituent le dénominateur du ratio de Sharpe. Des ratios de Sharpe hebdomadaires sont aussi calculés pour capturer une potentielle auto-corrélation dans les ratios de Sharpe. Nous ajoutons le kurtosis et skewness des rendements qui pourraient apporter de l'information utile. Enfin notre dernière feature est l'indice Herfindahl-Hirschman (HHI) qui est une mesure de diversification (permet de voir si le ratio de Sharpe est meilleur lorsque l'AUM est alloué de manière diversifiée ou non) :

$$HHI = \sum_{i=1}^N \omega_i^2, \quad 0 < HHI \leq 1 \begin{cases} HHI = 1 : \text{portefeuille totalement concentré} \\ HHI = \frac{1}{N} : \text{portefeuille équitable} \end{cases}$$

## Enjeu

La problématique proposée par Napoleon est ambitieuse car prédire un ratio de Sharpe hebdomadaire est une tâche complexe. Il y a plusieurs raisons à cela, premièrement, les rendements financiers sont très peu autocorrélés contrairement à la volatilités ou aux prix. C'est le cas dans nos données, les rendements des stratégies ne présentent pas d'autocorrélations. En revanche, à une même date les rendements de différentes stratégies peuvent être corrélées. En l'absence d'autocorrélation, la prédition de rendement et donc du ratio de Sharpe (qui nécessite une prédition sur cinq jours) est difficile. Les rendements étant presque imprévisibles et la volatilité étant très instable, le ratio de Sharpe futur hebdomadaire est très bruité et sûrement peu lié aux informations passées. Ce bruit est amplifié par l'annualisation. Le graphique ci-dessus montre que les corrélations entre notre cible et nos features sont petites et confirment que le signal est faible. Par ailleurs, le ratio de Sharpe futur issu de sept stratégies différentes est expliquée par un très grand nombre de variables : les rendements des stratégies, leurs variances et structure de covariance ainsi que les poids du portefeuille. Le signal est donc dilué entre de nombreuses variables ce qui ajoute une difficulté pour la prédition.

Nous supposons tout de même qu'il est possible de partiellement prédire ce ratio de Sharpe. La première raison à cela est qu'il existe sur les marchés des dépendances temporelles exploitables, même si ce n'est pas le cas pour les rendements. Par exemple nous savons que la volatilité des actifs financiers (et des stratégies financières) présentent de d'autocorrélation. Des régularités statistiques comme celles-ci peuvent donc être exploités par des modèles pour la prédition. Par ailleurs, la diversification joue un rôle important dans la performance ajustée du risque dans un portefeuille, et nous pensons qu'il est possible d'utiliser l'information des poids des stratégies pour cette prédition.

L'utilisation de méthodes d'apprentissage automatique est particulièrement pertinente ici en raison de la forte dimensionnalité des données, de la nature non linéaire des relations entre les poids, les trajectoires temporelles des stratégies et le ratio de Sharpe futur, ainsi que de la complexité des interactions entre stratégies. Notre cible est une mesure complexe qui dépend directement d'un grand nombre de variables. En effet, le poids de ces stratégies, leurs rendements, les covariances entre ces stratégies et la valeur des trois actifs supplémentaires

impactent directement le ratio de Sharpe du portefeuille. Un aussi grand nombre de variables contenant de l'information est difficilement intégrable dans un modèle économétrique classique. Par ailleurs, il est naturel de supposer que les relations entre les features et la cible sont complexes et non-linéaires, ce qui rend les modèles d'apprentissage automatique plus pertinents que des modèles de régressions simples.

Les données présentent un réel intérêt pour des modèles d'apprentissage automatique par leur structure et leur dimension. Chaque observation contient beaucoup de variables, ce qui rend le problème difficile à traiter avec des modèles linéaires simples. Cette richesse des données permet en revanche d'exploiter efficacement des modèles comme les Random Forest, le boosting ou les réseaux de neurones, qui sont conçus pour gérer des espaces de variables complexes. Le nombre élevé d'observations (10 000) rend également leur utilisation possible.

Enfin, l'intérêt de cette tâche est clair, la prédiction permettrait de savoir comment allouer optimalement le portefeuille de stratégie.

## Séparation du data set

Le data set est construit à partir de fenêtres temporelles glissantes de 21 jours, chacune pouvant donner lieu à 50 observations différentes correspondant à des pondérations distinctes appliquées aux mêmes stratégies. Bien que les trajectoires de prix sous-jacentes soient strictement identiques dans ces groupes d'observations, les pondérations différentes produisent des ratios de Sharpe distincts. Autrement dit, il est possible d'avoir 50 observations avec les mêmes séries temporelles mais seulement des poids différents. Dans ce contexte, une séparation classique des données entraînerait inévitablement du data leakage.

Pour éviter ce problème, les observations issues d'une même trajectoire de marché sont regroupées et placées dans un même jeu de données (entraînement, validation ou test). Nous procérons à une séparation explicite d'un jeu de test représentant 20 % des groupes et faisons de même pour la validation set, une proportion volontairement élevée afin de disposer d'un nombre suffisant de trajectoires réellement indépendantes, malgré la redondance des séries temporelles.

## Modèles

Les données utilisées sont de dimension élevée, avec un grand nombre de variables explicatives. Malgré la transformation en rendements, qui permet de réduire l'autocorrélation temporelle, une corrélation significative subsiste entre certaines variables, notamment entre stratégies observées à une même date. De plus, les relations entre les prédicteurs et la variable cible sont susceptibles d'être non linéaires et d'impliquer des interactions complexes. Dans ce contexte, il est nécessaire de recourir à des modèles capables de gérer à la fois la forte dimension, la colinéarité et la complexité des relations, tout en limitant le risque de sur-apprentissage.

Le premier modèle testé est un modèle de régression Elastic Net, qui est linéaire et interprétable. Ce modèle de régression combine pénalités L1 et L2 afin de gérer un grand nombre de variables potentiellement corrélées. Les variables explicatives sont standardisées avant estimation. L'Elastic Net est particulièrement adapté à nos données car le grand nombre de variables et leurs corrélations empêchent l'utilisation de modèles interprétables de régression sans régularisation. Sa régularisation combinant les modèles Ridge et Lasso permet une sélection de variables souple, évitant ainsi d'écartier arbitrairement des prédicteurs proches. Sa structure linéaire permet en outre une interprétation économique claire des effets estimés.

Nous considérons ensuite un modèle de Random Forest, un modèle qui repose sur l'agrégation d'arbres de décision construits sur des sous-échantillons aléatoires des données et des variables. Ce modèle est très adapté ici car il y a beaucoup de variables et car elles sont redondantes et possède un signal faible. Chaque arbre est entraîné sur un sous-ensemble de variables, ce qui permet de capté beaucoup de petits signaux qui sont ensuite agrégés. Cela permet aussi de ne pas avoir à se concentrer sur certaines variables. Par ailleurs les arbres pourront capter des relations non linéaires.

Nous testons également un modèle XGBoost, fondé sur le boosting d'arbres de décision, afin de capturer des relations non linéaires et des interactions plus complexes entre les variables. Grâce à l'entraînement successif sur les erreurs ce modèle peut exploiter des effets plus subtils que les précédents.

Enfin nous utilisons un multilayer perceptron (MLP) qui est un réseau de neurone feedforward pleinement connecté. Ce modèle est encore plus général que les précédents et pourrait apporter des gains de prédictions puisque notre cible est une fonction complexe et sûrement très non-linéaire. Le potentiel avantage par rapport aux modèles basés sur des arbres est que celui-ci peut appliquer des fonctions continus (par rapport à des splits discrets) sur l'ensemble des variables (par rapport à des sous-ensembles), ce type de relation peut être plus

naturel.

## Méthodologie

Les modèles sont entraînés exclusivement sur un jeu d'apprentissage, tandis que leurs hyperparamètres sont sélectionnés sur un jeu de validation disjoint, afin de contrôler la complexité et d'éviter toute fuite d'information. Le modèle retenu est ensuite ré-entraîné sur l'ensemble apprentissage-validation avant d'être évalué une seule fois sur un jeu de test indépendant.

La métrique retenue est la MAE qui est moins sensible aux erreurs extrêmes que le RMSE, car celles-ci ne sont pas amplifiées par une mise au carré. Cela est particulièrement pertinent dans notre contexte, où le ratio de Sharpe présente une forte dispersion et peut prendre des valeurs très élevées en valeur absolue. La RMSE pourrait trop pénaliser certaines grosses erreurs et être moins représentatif que la MAE.

## Résultats

Comme benchmark de référence, nous utilisons un modèle naïf qui prédit pour toutes les observations la moyenne du ratio de Sharpe observée dans l'échantillon d'entraînement, ce qui fournit un point de comparaison minimal permettant de vérifier que les modèles apportent un réel gain prédictif.

Nous renseignons nos résultats sur la table ci-dessous, obtenus avec la seed 42 (sauf pour les MLP). Pour vérifier la stabilité des modèles nous effectuons plusieurs itérations avec des seeds aléatoires.

Table 1: Performances des différents modèles

Modèle	MAE	Corr(pred, y)	Std(pred)
Moyenne (benchmark)	4,45	x	x
Elastic Net (1)	4,34	0,23	1,46
Random Forest (2)	4,43	0,09	0,78
ExtraTrees (3)	4,37	0,17	0,67
XGBoost (4)	4,39	0,19	0,37
XGBoost (5)	4,49	0,06	1,48
MLP (6)	4,31	0,20	0,71

Si l'on considère ces résultats dans leur ensemble, nous pouvons conclure que les gains de MAE et les corrélations prédition/cible indiquent que les modèles réussissent à extraire de l'information des features et améliorent la prédition par rapport au benchmark. Cette amélioration est modeste mais significative et cohérente avec le sujet et la difficulté de prédire un ratio de Sharpe futur. Les écarts-types des prévisions sont faibles (l'écart-type de la cible est 5,9) ce qui indique que les modèles parviennent peu à expliquer la variance de la cible et que les prédictions sont très regroupées.

Le modèle Elastic Net (1) retient un paramètre de régularisation (alpha) de 0,6 ce qui est élevé. Le modèle est donc fortement régularisé. Le ratio l1 sélectionné est de 0,5 la régularisation est donc mixte, ce qui indique à la fois une sélection de variable et des coefficients faibles. C'est un résultat logique au vu du nombre de features et du faible signal. Par curiosité nous testons également le même modèle mais avec une grille d'hyperparamètres plus "agressive" : alpha maximum est 0,1. L'idée est de "forcer" le modèle à être moins régularisé et à produire des données plus dispersées pour tenter de répliquer la variance et donc la distribution de notre cible. Cette tentative échoue ( $MAE > 5$ ) ce qui indique un fort overfitting de ce modèle qui capte trop de bruits propres aux données d'entraînement.

Le modèle Random Forest (2) avec sélection d'hyperparamètres sur le validation set est aussi un modèle assez peu complexe et peu flexible. Cela est confirmé par la très faible variance de la prédition qui indique que la considération de relations non linéaires permise par une Random Forest n'améliore pas la prévision. Si des relations non-linéaires existent entre les features et la cible alors elles sont trop instables pour être utilisées. Ce résultat montre que les relations entre notre cible et nos features sont instables et bruitées ce qui obligent les modèles à être fortement régularisés. Le modèle sélectionne un grand nombre d'arbre, et un nombre de variable candidates à chaque split égale à  $\log_2(n)$ . Ce dernier paramètre indique que peu de variables sont considérées pour chaque split, ce qui ajoute de l'aléa dans le modèle qui est donc assez fortement régularisé.

Les conclusions précédentes sont confirmées par le résultat du modèle ExtraTrees qui introduit de l'aléa supplémentaire par rapport à une Random Forest, réduisant la variance. Ce modèle se prête donc mieux à

nos données qu'une Random Forest car il se généralise mieux sur nouvelles observations.

Par défaut, les modèles de gradient boosting tels que XGBoost sont généralement entraînés en minimisant une erreur quadratique moyenne (Mean Squared Error, MSE), qui pénalise fortement les erreurs de grande amplitude. Ce choix est bien adapté à des variables cibles proches d'une distribution gaussienne et peu sujettes aux valeurs extrêmes. Toutefois, dans notre cas, la distribution est fortement étalée et présente des valeurs extrêmes dues à la nature très volatile des actifs sous-jacents. L'utilisation de la MSE conduit les modèles à sur-pondérer ces observations extrêmes, souvent dominées par le bruit, au détriment de la structure moyenne du signal. Afin de limiter cet effet, nous avons remplacé la loss quadratique par une loss robuste de type Pseudo-Huber, qui combine un comportement quadratique pour les erreurs modérées et linéaire pour les erreurs importantes. Ce compromis permet de réduire l'influence des valeurs aberrantes tout en conservant une fonction de perte différentiable, adaptée à l'optimisation par gradient. En testant un modèle avec la MSE nous confirmons que la performance est moins bonne.

Les résultats du XGBoost (4) ont encore la même signification que ceux précédents, c'est particulièrement visible ici. La MAE et la corrélation indiquent que la prédiction est meilleur que le benchmark et que ce n'est pas du bruit. En revanche l'écart-type indique que les prédictions sont très peu dispersées. De plus le modèle retient seulement deux arbres ce qui est très faible. Le modèle est simple et très régularisé. Nous tentons de "forcer" une plus grande flexibilité dans le modèle XGBoost (5) en imposant un nombre d'arbres minimum de 50. Le résultat montre que cela conduit simplement à de l'overfitting et dégrade la performance.

Pour le modèle MLP, les hyperparamètres testés et choisis à l'aide du validation set correspondent à un modèle simple avec peu de couches cachés (1 ou 2) et peu de neurones (entre 16 et 64). Des modèles plus complexes n'auraient pas trop de sens au vu des données et des résultats des modèles précédents.

La première différence majeure que nous observons avec le modèle MLP est l'instabilité de ce-dernier. D'une seed à l'autre les résultats obtenus sont très différents, ce qui n'est pas le cas avec nos autres modèles. Pour Random Forest l'agrégation de nombreux arbres réduit l'aléa, en boosting, l'agrégation séquentielle a le même effet. Mais pour un MLP l'aléa a beaucoup d'impact. Une des raisons est que les fonctions de pertes sont convexes par rapport aux prédictions mais pas par rapport aux paramètres. L'initialisation aléatoire des paramètres conduit donc à des solutions différentes entre deux entraînement (différents minima locaux). Nous observons une instabilité des résultats des modèles MLP, malgré des solutions de régularisation comme la normalisation des données, une régularisation L2, un "early stop", un paramètre de "drop out" et une architecture simple. Comme pour les autres hyperparamètres, l'early stop est déterminé à l'aide du validation set. Par ailleurs, nous n'avons pas modifié l'initialisation des paramètres, par défaut Keras utilise la méthode de Xavier avec fonction uniforme.

Une régularisation trop forte peut provoquer de l'instabilité, c'est le cas lorsque le modèle ne capte pas assez de signal et que l'entraînement est rendu très aléatoire par l'initialisation, un taux de drop out élevé et un "early stop" tôt. Nous essayons donc un second modèle avec une grille d'hyperparamètres (toujours sélectionnés grâce au validation set) raisonnable qui limite la régularisation. Cela stabilise un peu le modèle mais celui-ci reste tout de même assez instable. Le problème qui persiste est que l'early stopping s'active souvent très tôt (parfois à la première "epoch"), en raison du faible signal et du bruit dans la cible. Cela cause l'instabilité du modèle. Pour résoudre ce problème nous fixons alors le nombre d'epoch minimal à 10, mais cela ne stabilise toujours pas le modèle.

Les prédictions finales, modèle (6) sont obtenues en agrégeant cinq réseaux de neurones entraînés avec des initialisations différentes, ce qui constitue une forme simple d'ensemble visant à réduire la variance du modèle. Pris individuellement, les réseaux de neurones présentent une forte variabilité selon l'initialisation et des performances instables. En revanche, l'agrégation des prédictions de plusieurs MLP permet de réduire significativement la variance et d'améliorer la performance globale, confirmant l'intérêt d'une approche en ensemble dans ce contexte bruité. Cela s'observe très bien en comparant la performance du modèle (6) à la moyenne et à l'écart-type des modèles pris individuellement, dont les résultats sont présentés dans la Table 2.

Table 2: Performances des MLP individuels utilisés pour le modèle (6)

Modèle	MAE	Corr(pred, y)	Std(pred)
Moyenne	4,36	0,17	1,00
Ecart-type	0,08	0,09	0,40

Pour conclure, les résultats montrent qu'il est possible d'extraire un signal prédictif faible mais significatif pour le ratio de Sharpe à court terme, malgré un bruit important. Les modèles fortement régularisés et les approches en ensemble apparaissent les plus adaptées, tandis que toute augmentation excessive de la complexité conduit rapidement à de l'overfitting.