

h e g

Haute école de gestion
Genève

- Copie Juré
- Copie Conseiller
- Copie Infothèque

Intelligence Artificielle appliquée à la gestion de portefeuille

Travail de Bachelor réalisé en vue de l'obtention du Bachelor HES

par :

Nicolas Jean René RICHARD

Conseiller au travail de Bachelor :

Istvan NAGY

Genève, le 16 juillet 2021

Haute École de Gestion de Genève (HEG-GE)

Filière Économie d'Entreprise, orientation Banque et Finance

Déclaration

Ce travail de Bachelor est réalisé dans le cadre de l'examen final de la Haute école de gestion de Genève, en vue de l'obtention du titre de Bachelor of Science en économie d'entreprise.

L'étudiant a envoyé ce document par email à l'adresse d'analyse remise par son conseiller au travail de Bachelor pour analyse par le logiciel de détection de plagiat URKUND.
<http://www.orkund.com/fr/student/392-orkund-faq>

L'étudiant atteste avoir réalisé seul-e le présent travail, sans avoir utilisé des sources autres que celles citées dans la bibliographie

L'étudiant accepte, le cas échéant, la clause de confidentialité. L'utilisation des conclusions et recommandations formulées dans le travail de Bachelor, sans préjuger de leur valeur, n'engage ni la responsabilité de l'auteur, ni celle du conseiller au travail de Bachelor, du juré et de la HEG.

« J'atteste avoir réalisé seul le présent travail, sans avoir utilisé des sources autres que celles citées dans la bibliographie. »

Fait à Genève, le 16 juillet 2021

Nicolas Jean René RICHARD

Remerciements

Je souhaite tout d'abord remercier Monsieur Istvan NAGY, qui aura su m'aiguiller lors de mon travail, ainsi que pour ses précieux conseils tout au long de la réalisation de ces recherches et travaux pratiques.

Je remercie également monsieur Julien RIBON, pour ses conseils et son enseignement lors du cursus de Banque et Finance, qui m'ont grandement aidé pour Python.

Je remercie ensuite un Data Engineer d'une banque privée genevoise pour son témoignage aussi intéressant qu'utile dans le cadre de ce travail.

J'adresse également mes remerciements à ma famille et à mes amis, qui m'ont conseillé, soutenu et relu avec la plus grande attention.

J'ai une pensée particulière pour mes professeurs de spécialité lors de mon Baccalauréat, qui ont suscité chez moi cet intérêt pour la programmation, fort utile dans ce cadre.

Je souhaite enfin remercier le corps enseignant de la HEG Genève, notamment dans le cadre de l'orientation Banque et Finance, qui par sa grande expérience, m'aura aidé à comprendre, vulgariser et appliquer des notions financières ardues dans le cadre de mon travail de bachelor.

Résumé

L'objectif de ce travail est de visualiser de la manière la plus simple, les différents tenants et aboutissants lorsque l'on parle d'intelligence artificielle et de gestion de portefeuille.

Nous étudions dans un premier temps la question de manière théorique, en définissant les différents termes, les styles de gestion de portefeuille.

Dans un deuxième temps, nous voyons un rapide état des lieux actuel par le biais de diverses statistiques ainsi que des témoignages ou pratiques actuelles.

Dans un dernier temps, nous effectuons une analyse pratique de la situation en utilisant un langage de programmation, Python, afin de comparer dans un cadre fixé (style de gestion buy and hold), différents portefeuilles d'actions gérés avec et sans intelligence artificielle. Nous comparons ensuite les diverses caractéristiques des portefeuilles et cherchons à les optimiser, par du stockpicking avec algorithme (traditionnel et IA) et par l'analyse de ces différents portefeuilles. Nous comparons la performance de ces deux portefeuilles (IA ou standard) afin de savoir si dans notre cas, une méthode d'optimisation est significativement meilleure que l'autre. Finalement, nous déduisons des conclusions et des points d'amélioration.

Nous conseillons aux acteurs de bien saisir le changement qui s'annonce, avec la globalisation des données car il s'agit d'une opportunité forte.

Table des matières

Déclaration	i
Remerciements	ii
Résumé	iii
Liste des tableaux	vi
Liste des figures	vi
1. Introduction	1
1.1 Préambule	1
1.2 Cadre historique de l'intelligence artificielle	1
1.3 Cadre historique de la gestion de portefeuille	2
1.4 Définitions	4
1.4.1 Notions liées à l'intelligence artificielle.....	4
1.4.2 Notions liées à la gestion de portefeuille	8
2. État des lieux de l'intelligence artificielle et de la gestion de portefeuille 11	
2.1 État des lieux mondial	11
2.1.1 Un exemple de cadre légal : États-Unis	11
2.1.2 Constatations mondiales passées et actuelles.....	11
2.1.3 Tendances mondiales prévues pour l'avenir	16
2.2 État des lieux en Suisse	17
2.2.1 Cadre légal Suisse.....	17
2.2.2 Constatations Suisses passées et actuelles.....	17
2.2.3 Tendances Suisses prévues pour l'avenir	19
2.2.4 Un témoignage d'une banque privée genevoise	20
2.3 Synthèse basée sur la recherche	22
3. Comparaison pratique de portefeuilles gérés avec et sans intelligence artificielle	23
3.1 Buts recherchés et postulats de départ	23
3.2 Création des divers algorithmes	25
3.2.1 Algorithme de stockpicking simple.....	25
3.2.2 Algorithme d'optimisation Markowitz et Deep Learning.....	26
3.2.2.1 Détail et fonctionnement de l'algorithme	27
3.2.2.2 Vérification de la validité de l'algorithme	27
3.2.2.3 Création des portefeuilles	29
3.2.2.4 Intégration de l'algorithme dans une boucle, output, mise en forme.....	30
3.2.3 Algorithme de frontière efficiente	31
3.3 Utilisation théorique des algorithmes	32
3.3.1 Tirage des données : portefeuilles.....	32
3.3.2 Principe du stockpicking par intelligence artificielle	33

3.4	Optimisation des portefeuilles et stockpicking avec intelligence artificielle	34
3.4.1	Premier tirage : SP500, pas de spécialisation	34
3.4.2	Deuxième tirage : spécialisation IA x1	36
3.4.3	Troisième tirage : spécialisation IA x2	36
3.4.4	Évolution des Sharpe, Rendements et Risques des portefeuilles	37
3.4.5	Alphas et Bêtas des portefeuilles	38
3.5	Comparaison du stockpicking Traditionnel vs IA	40
3.5.1	Comparaison des valeurs liées aux portefeuilles	41
3.5.2	Comparaison des alphas et bêtas	44
3.6	Synthèse des résultats	45
3.6.1	Constatations liées aux données	45
3.6.2	Recommandations	46
3.6.3	Points d'amélioration	46
4.	Conclusion	49
	Bibliographie	50
	Annexe 1 : Codes PYTHON d'IA DeepDow et de mise en forme	54
	Annexe 2 : Codes PYTHON de stockpicking	61
	Annexe 3 : Code PYTHON de frontière efficiente.....	65

Liste des tableaux

Tableau 1 : Répartition de l'optimisation des portefeuilles : non spécialisée	35
Tableau 2 : Répartition de l'optimisation des portefeuilles : spécialisation IA x1	36
Tableau 3 : Répartition de l'optimisation des portefeuilles : spécialisation IA x2	37
Tableau 4 : Comparaison Stockpick : Minimum Variance.....	41
Tableau 5 : Comparaison Stockpick : Maximum Return	41
Tableau 6 : Comparaison Stockpick : Maximum Utility	42
Tableau 7 : Résumé de performance : stockpick Normal vs IA, optimiseur, annualisé ..	44

Liste des figures

Figure 1 : Illustration de la frontière efficiente selon Markowitz	3
Figure 2 : Couches neuronales d'algorithmes machine et deep learning	6
Figure 3 : Comparaison entre neurone humain et neurone artificiel	7
Figure 4 : Décomposition des neurones de deep learning pour identifier un portrait.....	7
Figure 5 : Usage de l'intelligence artificielle en gestion de portefeuille, monde, 2020	12
Figure 6 : Part d'amélioration des secteurs internes par l'intelligence artificielle	14
Figure 7 : Points de développement principaux pour le déploiement d'intelligence artificielle.....	15
Figure 8 : Disponibilité de l'intelligence artificielle dans les entreprises financières CH / ALL / AUT.....	19
Figure 9 : Portefeuille stockpické optimisé avec l'algorithme DeepDow	28
Figure 10 : Tableau de comparaison de solveur Excel / DeepDow	29
Figure 11 : Extrait de code Python qui calcule les log-returns	29
Figure 12 : Code de création des portefeuilles	30
Figure 13 : Snapshot du tableau résultant de l'optimisation.....	31
Figure 14 : Frontière efficiente du portefeuille stockpické standard.....	32
Figure 15 : Extrait du tableau de stockpicking intermédiaire IA	34
Figure 16 : Boxplot des alphas selon les optimisations, 2017 - 2020	39
Figure 17 : Boxplot des bêtas selon les optimisations, 2017 - 2020	40
Figure 18 : Résumé de performance : stockpick Normal vs IA, équipondéré	43
Figure 19 : Boxplot des alphas selon les optimisations, 2020 - 2021	44
Figure 20 : Boxplot des bêtas selon les optimisations, 2020 - 2021	45

1. Introduction

1.1 Préambule

Il est difficile de dire en quelle année, ou même quelle période l'intelligence artificielle est apparue de manière notable dans le domaine de la finance. Il existait plusieurs produits et logiciels conçus pour des applications spécifiques. On peut notamment trouver par exemple, la commercialisation d'un produit en 1986, PlanPower, pour la planification financière sur mesure d'individus avec un revenu supérieur à \$75'000¹. On peut toutefois affirmer que le développement de cette technologie s'est largement accentué ces dernières années. Pour preuve, on considérait assez récemment que 9% de tous les Hedge Funds utilisent du Machine Learning (une technique d'intelligence artificielle) pour créer des modèles statistiques². Selon Preqin, cette valeur, difficile à quantifier, a augmenté et le nombre de Hedge Funds axés intelligence artificielle lancés en 2018 était en hausse de 77% par rapport à 2016³.

En finance, il existe donc bien des applications pour l'intelligence artificielle, du front au back office. Dans notre cas, nous nous intéressons à son application dans le cadre de la gestion de portefeuilles : la situation actuelle, les tendances, et en pratique.

1.2 Cadre historique de l'intelligence artificielle

On note que les premiers travaux importants qui traitaient la question de l'intelligence d'une machine remontent aux années 1950. C'est justement en 1950 qu'Alan Turing, mathématicien britannique, publie un article, « Computing Machinery and Intelligence », qui commence par : « I propose to consider the question : 'Can machines think ?' » (Turing, 1950, p.1). Cette question marquera la pensée derrière la notion d'intelligence artificielle. Turing, dans ce même article, définit la notion de « Imitation Game⁴ » : un jeu où un homme et une femme discutent avec un interrogateur, qui ne connaît pas leur genre et doit le découvrir en posant des questions. Turing propose ainsi de remplacer l'un des deux participants par une machine : le but de la machine ne serait pas dans l'absolu de répondre correctement aux questions, mais plutôt de se faire passer pour un humain aux yeux de l'interrogateur : le Test de Turing. 6 ans après, des scientifiques se réunissant à l'université de Dartmouth définiront les bases de « l'intelligence artificielle ». La finance fut l'un des domaines les plus anciens à s'intéresser à l'intelligence artificielle : réduction

¹ <https://medium.com/district3/the-history-of-ai-in-finance-7a03fcb4a498>

² https://www.turing.ac.uk/sites/default/files/2019-04/artificial_intelligence_in_finance_-_turing_report_1.pdf (p. 2)

³ <https://www.preqin.com/insights/research/blogs/the-rise-of-the-machines-ai-funds-are-outperforming-the-hedge-fund-benchmark>

⁴ <https://academic.oup.com/mind/article/LIX/236/433/986238>

des biais cognitifs, rapidité d'exécution, intégration de gros sets de données... La commercialisation de produits explose dans les années 80 et 90, et le développement de nouvelles méthodes renforcera la place de l'intelligence artificielle en finance.

L'adoption de l'intelligence artificielle en finance s'est accélérée récemment, et permet entre autres d'éviter des décisions irrationnelles basées sur des émotions mais aussi d'intégrer d'autres caractéristiques telles que le Sentiment Analysis⁵. Ces dernières années, partout dans le monde, des acteurs financiers ont développé des solutions avec intelligence artificielle axées client, comme les robo-advisors.

1.3 Cadre historique de la gestion de portefeuille

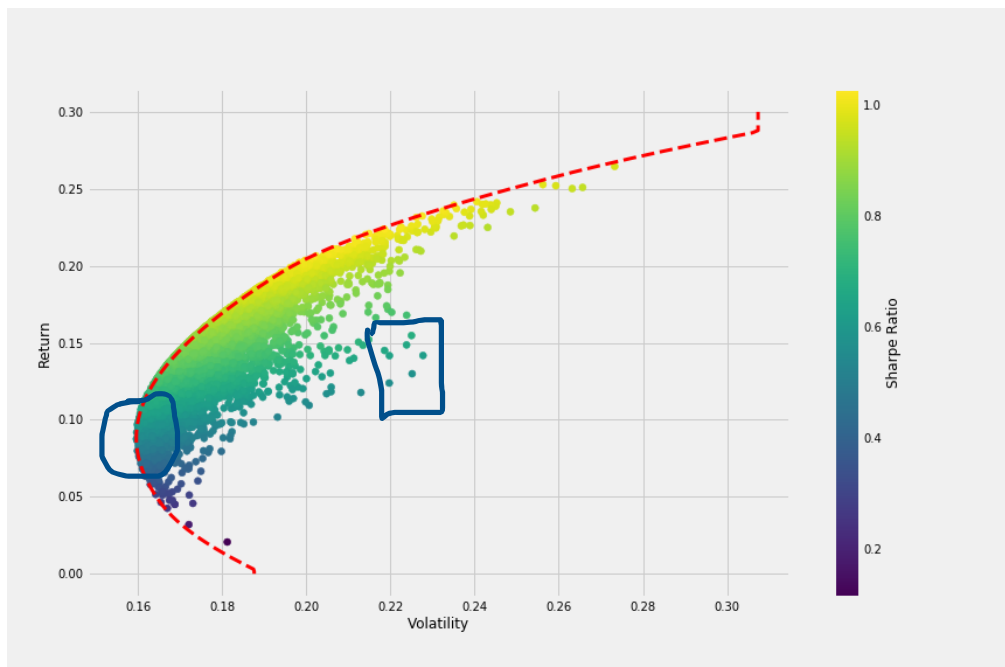
Gérer son portefeuille est un sujet assez large. Dans les années 30, beaucoup d'investisseurs avaient pour objectif d'acheter une ou plusieurs actions, à un prix qui leur semblait correct. La rareté ou le manque d'information rendait cette tâche compliquée, et investir en 2021 semble étonnamment simple par rapport à cette période. De plus, certaines caractéristiques n'étaient pas encore pleinement considérées, telles que l'analyse des « fondamentaux » d'entreprise ou même la notion de risque.

En 1952, Harry Markowitz écrit un article qui sera publié dans le prestigieux « Journal of Finance », nommé « Portfolio Selection ». Il indiquera deux pratiques évidentes mais pas forcément mises en œuvre en ces temps : obtenir un certain rendement implique de prendre des risques, et la diversification est une clé d'un bon portefeuille. Il explique le premier principe en justifiant que les investisseurs ont une aversion au risque. A rendement égal, un investisseur choisira le portefeuille le moins risqué, et de ce fait, un investisseur acceptera un risque supérieur si son portefeuille est mieux rémunéré. Pour le deuxième principe, il avance la notion de corrélation. Pour rappel, si deux actifs sont positivement corrélés (ie. une corrélation proche de 1), alors dans le cas où les rendements de l'un sont croissants, alors les rendements de l'autre le seront également. A l'inverse, si ces actifs sont négativement corrélés (ie. une corrélation proche de -1), alors les rendements croissants de l'un impliqueront des rendements décroissants pour l'autre. Dans le cas où la corrélation est proche de 0, alors cela implique une décorrélation des deux actifs : la situation d'un actif n'impliquera pas de mouvement sur un autre actif. Dans le cas de la Théorie Moderne du Portefeuille de Markowitz (Modern Portfolio Theory, MPT), une bonne diversification se base sur le choix d'actifs qui ne sont pas totalement corrélés positivement. Cela permet d'éviter une chute du cours de tous les actifs (donc une chute brutale du portefeuille) si l'un des actifs devait perdre de sa valeur.

⁵ Selon témoignage d'un Data Engineer, banque privée genevoise

La théorie moderne du portefeuille se base sur analyse moyenne-variance et met en rapport le rendement d'un actif (moyenne) par rapport à son risque (écart-type, découlant de la variance). Cela implique qu'avec plusieurs actifs, il est possible de les pondérer (leur donner une plus ou moins grande importance dans le portefeuille en investissant plus ou moins du capital total dans la position), ce qui influera donc sur le rendement et le risque du portefeuille. Cela implique également que suivant les portefeuilles construits, pour un certain taux de risque il existe un ou plusieurs taux de rendement possible. Si l'on suit le premier principe de cette théorie, alors un investisseur devrait choisir, à risque égal, le portefeuille lui procurant le plus de rendement. De ce fait, il existe donc des portefeuilles dits « inefficients », et un portefeuille dit « efficient », pour un même niveau de risque. Lorsque l'on crée une multitude de portefeuilles d'actifs différemment pondérés, nous allons chercher à fixer un niveau de risque, ensuite calculer par optimisation le portefeuille au meilleur rendement, puis incrémenter ce risque et recommencer. Finalement, sur un graphique avec en abscisse le risque, et en ordonnée le rendement, on obtient une courbe qui contient les portefeuilles dits « efficaces » : la frontière efficiente. Il faudrait donc choisir un portefeuille qui se situe sur cette courbe. Le choix d'un portefeuille plus ou moins risqué dépendra ensuite de l'aversion au risque de l'investisseur ou d'autres critères de sélection.

Figure 1 : Illustration de la frontière efficiente selon Markowitz



(NEVES Fabio, Plotting Markowitz Efficient Frontier with Python, 2018)

On remarque, sur la figure ci-dessus, la frontière efficiente tracée en pointillés rouge. Le cercle bleu, indiqué aux coordonnées approximatives (0.16 ; 0.08) indique le portefeuille

dit « Variance minimale » (minimum variance portfolio). En effet, pour ce choix d'actifs, il n'existe pas d'autre portefeuille avec un risque plus faible. Les points contenus dans le carré bleu, aux coordonnées (0.225 ; 0.125) sont des portefeuilles dits « inefficients » : il existe d'autres portefeuilles avec le même niveau de risque mais avec un rendement supérieur, qui se situent pile à la verticale, au nord : sur la frontière efficiente.

D'autres modèles sont venus compléter la gestion de portefeuille, qui peuvent intégrer de l'intelligence artificielle afin d'améliorer des prédictions ou ingérer de grandes quantités de données. Il existe donc bien des méthodes pour gérer un ou plusieurs portefeuilles, mais aucune de ces méthodes ne peut garantir une gestion parfaite.

1.4 Définitions

1.4.1 Notions liées à l'intelligence artificielle

Commençons par définir ce qu'est l'intelligence artificielle. Il existe plusieurs définitions pour définir une notion très large et complexe, donc nous décidons de choisir la définition proposée par Russell & Norvig dans leur ouvrage « **Artificial Intelligence – A Modern Approach** ». Ils se basent sur des constatations historiques et proposent 4 dimensions (Russell & Norvig, 2016, p.2 – 3) :

- Agir de manière humaine : tel le test de Turing, le but est de pouvoir répliquer un comportement humain le plus fidèlement possible, en adoptant certains points : le langage (parler correctement une certaine langue), la mémoire (enregistrer les actions et l'écoute), la capacité de raisonnement (pour utiliser les informations dans le but de répondre à des questions), l'apprentissage (pour s'adapter aux informations et en déduire des schémas), la vision (pour percevoir l'environnement) et la robotique (pour manipuler l'environnement).
- Penser de manière humaine : le but est de répliquer le fonctionnement du cerveau humain afin de simuler le processus de pensée de l'Homme, en utilisant par exemple les sciences cognitives.
- Penser de manière rationnelle : cette dimension se base sur la logique et sur le principe de pensée basée sur des faits avérés pour déduire des conclusions.
- Agir de manière rationnelle : le but est d'obtenir, en connaissance de cause et avec les données à dispositions, la meilleure issue possible (agent rationnel).

Passons maintenant au Machine Learning. En effet, si un algorithme souhaite répliquer un comportement humain, il doit pouvoir apprendre. On définit souvent deux grandes catégories d'apprentissage : l'apprentissage supervisé et l'apprentissage non supervisé. Cet apprentissage peut porter sur plusieurs points d'une même situation. Par exemple, dans le cas d'une simulation de conduite (Russell & Norvig, 2016, p.694), lorsque le copilote donne l'ordre de freiner, l'intelligence artificielle peut apprendre plusieurs points

de cette situation : la recherche des causes de cet ordre, les conséquences de cet ordre, ainsi que les caractéristiques d'une situation où cet ordre n'est pas donné.

Afin de définir l'apprentissage supervisé, nous pouvons faire l'analogie avec le comportement d'un enfant, comme l'explique sur son site l'entreprise d'informatique Oracle. Par exemple, un professeur expliquerait à un enfant, par le biais d'un livre, à quoi ressemblent les fruits, ainsi que leur nom, couleur, caractéristiques... Un enfant qui verrait ensuite une pomme, comparerait ce fruit et ses caractéristiques avec les différentes notions qu'il a vu dans ce livre, et établirait une correspondance. Dans le cas d'un algorithme, le responsable de l'algorithme va enseigner à l'algorithme ce « livre » avec les conclusions qu'il doit tirer selon ce qu'il voit⁶.

L'apprentissage non-supervisé se différencie en n'ayant pas de « valeurs théoriques » pour servir de base de comparaison. Ainsi, l'enfant ne verrait pas de livre avec des fruits, mais il verrait d'abord les fruits, et lui-même va les comparer et former des groupes. Dans le cas d'un algorithme, ce dernier va grouper les données entre elles et définir à quoi elles correspondent.

On pourrait également définir une notion d'apprentissage par renforcement (Russell & Norvig, 2016, p.695). Il s'agit de d'indiquer à l'algorithme si son fonctionnement est correct ou incorrect. Par exemple, dans le cadre d'une voiture autonome, si le véhicule est à l'origine d'une amende, alors ce point est considéré comme négatif donc l'intelligence artificielle a mal agi dans une certaine situation, qu'il faudra corriger. Dans le cas d'un match de foot, si l'équipe contrôlée par intelligence artificielle marque un but, alors l'action est positive et le comportement est positif. On pourrait également faire le parallèle avec un portefeuille : il serait possible de fixer (selon nos recherches, directives internes, constatations, ou objectifs par exemple) un taux minimal de performance positive pour notre algorithme, afin de déterminer si la gestion est correcte ou non : dans notre exemple, si le portefeuille réalise une performance positive sur au moins $x\%$ des périodes, alors la gestion est correcte.

Une nouvelle notion, qui fait suite au Machine Learning et qui concerne notre partie pratique, est le Deep Learning. Il s'agit d'une sous-méthode d'apprentissage qui n'a pas besoin de supervision et qui peut utiliser données non structurées⁷. Cela se base sur l'usage de réseaux neuronaux, qui sont sensés répliquer le cerveau humain. Là où un algorithme standard analyserait des données de manière linéaire (unes à unes), ces neurones vont analyser une caractéristique de plus par neurone. De plus, ces neurones

⁶ <https://www.oracle.com/ch-fr/data-science/machine-learning/what-is-machine-learning/>

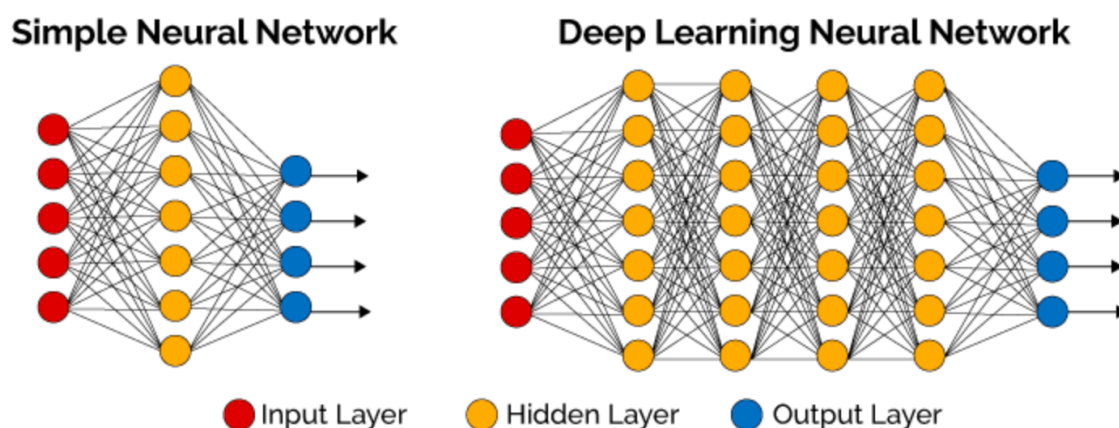
⁷ <https://www.investopedia.com/terms/d/deep-learning.asp>

sont liés entre eux, ce qui rend l'analyse liée bien plus rapide. Ainsi, les données sont comparées et l'algorithme peut déterminer des structures. Par exemple, si l'on prend le cas d'un usager qui privilégie la mobilité douce, deux manières de faire un choix entre bus et vélo seraient possible avec machine learning ou deep learning. Dans le cas du machine learning, l'algorithme aurait besoin de connaître des caractéristiques de décision : météo, retards, accidents, température, distance. Ensuite, il les analyserait une par une pour prendre cette décision. Dans le cas du deep learning, les neurones analyseraient toutes les caractéristiques qu'il peut comprendre des données qui lui sont fournies pour prendre sa décision.

Pour compléter le point précédent et tenter d'expliquer simplement le fonctionnement de ces algorithmes, voici quelques figures comparatives dans le cas d'algorithmes d'intelligence artificielle Machine Learning et Deep Learning.

Schématiquement, le machine learning prend en compte trois catégories de couches. Une couche, *Layer* en anglais, est une fonction mathématique qui prend des données, *inputs*. La première catégorie de couche, *input layer*, prend en compte les données que nous souhaitons analyser. La deuxième catégorie, *hidden layer*, permet d'effectuer les calculs liés à nos données à analyser (à voir : prochaines figures, prochains paragraphes). La dernière couche, *output layer*, propose les données analysées en sortie et un résultat, *output*. Ces couches sont liées entre elles afin d'interagir et d'échanger les données.

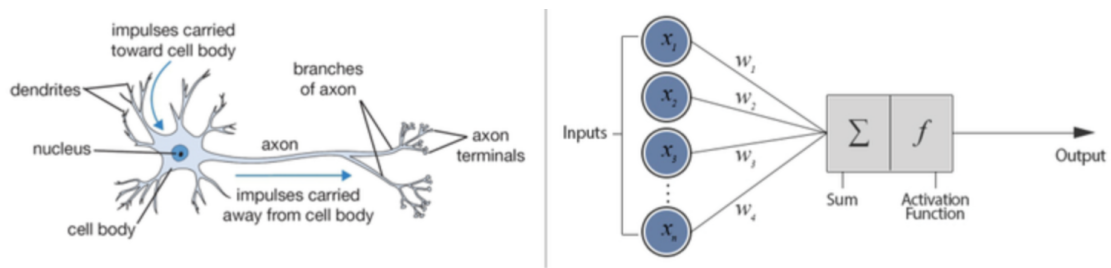
Figure 2 : Couches neuronales d'algorithmes machine et deep learning



(Towards Data Science, How I was able to speed up my deep learning algorithm, 2021)

On constate la différence entre un algorithme de machine learning et un algorithme de deep learning : l'analyse de données se fait sur plusieurs *hidden layers* afin de travailler les données plusieurs fois et d'analyser toutes les caractéristiques. Mais comment fonctionne cette couche intermédiaire ?

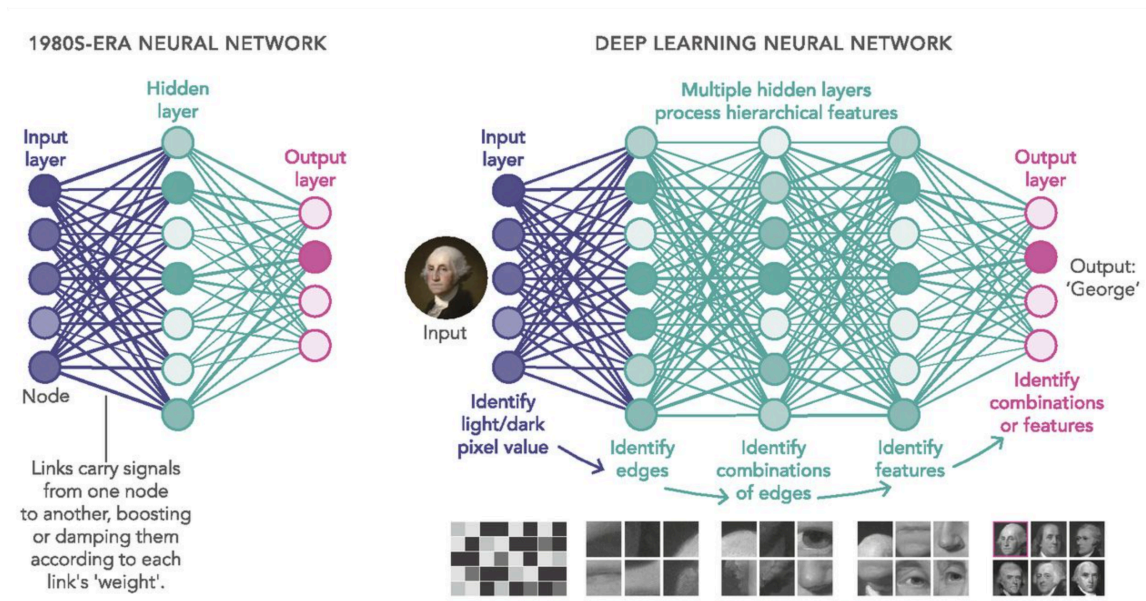
Figure 3 : Comparaison entre neurone humain et neurone artificiel



(Karlijn Willems, Deep Learning in Python, 2019)

Dans un neurone humain (à gauche), les informations entrent en input dans les dendrites, sont traitées dans les axons puis sortent en output dans les terminaux des axons afin d'être transférées vers d'autres neurones. Dans un neurone artificiel (à droite), les données entrent en input avec un certain poids, sont traitées dans la fonction d'activation (qui attribuera une valeur pour chaque input) puis transférées en output. Un exemple simple permet de mieux comprendre les réseaux neuronaux : l'identification d'un visage⁸. Par exemple, plusieurs neurones vont identifier les yeux : couleur, forme, iris... Seuls, ces neurones ne permettent pas d'identifier un visage. Mais avec d'autres neurones qui interagissent tous entre eux, alors on peut identifier le nez, les sourcils, et bien d'autres caractéristiques qui vont entrer en input d'autres neurones pour produire un output : l'identification du visage. La figure suivante illustre ce propos avec l'exemple d'un portrait de George Washington, décomposé pour mieux le reconnaître.

Figure 4 : Décomposition des neurones de deep learning pour identifier un portrait



⁸ <https://deepai.org/machine-learning-glossary-and-terms/hidden-layer-machine-learning>

Passons maintenant à une autre notion intéressante liées à notre sujet : le Big-Data. La définition qui semble être celle de référence⁹ est la définition de Gartner en 2001 : il s'agit de flux de données en *volumes croissants*, avec une grande *variété* et à grande *vitesse* (Gartner, 2001). Ces trois « V » sont l'enjeu principal de ce sujet : comment analyser des données très différentes, arrivant vite et en grande quantité ? Ces données sont importantes car elles permettent de déduire des conclusions très importantes, quel que soit le domaine, du marketing à la finance.

Finalement, nous pouvons mentionner cette notion que beaucoup ont entendu ces derniers mois : le robo-advisor. Il s'agit d'un algorithme qui peut être conçu avec une intelligence artificielle afin d'optimiser, conseiller ou gérer un portefeuille. Par exemple, la banque Suisse Swissquote propose un Robo-Advisor qui peut proposer des actifs ou gérer des positions de manière autonome, afin d'éviter à l'investisseur de garder les yeux sur les marchés de manière permanente¹⁰.

1.4.2 Notions liées à la gestion de portefeuille

Afin de mieux comprendre les problématiques et les critères que nous allons utiliser lors de notre analyse pratique, il convient de définir quelques notions qui sont liées à la gestion de portefeuille, que ce soient des notions théoriques ou calculatoires.

L'Alpha (α) représente la performance supplémentaire d'un actif ou portefeuille par rapport à l'indice, ajustée au risque. Elle est tirée du modèle du CAPM, ce dernier étant défini par l'équation de régression linéaire suivante :

$$R_p - R_f = \alpha + \beta * (R_i - R_f) + \varepsilon^{11}$$

Avec R_p le rendement du portefeuille / actif, R_f le rendement sans risque, R_i le rendement de l'indice, α l'alpha du portefeuille / actif, β le bêta du portefeuille / indice, ε les erreurs résiduelles.

La notion de Bêta (β) est liée à l'alpha. Il s'agit de la sensibilité de l'actif ou du portefeuille à l'indice de référence. Nous calculons le Bêta ainsi :

$$\beta_{actif} = \rho(actif ; SP500) * \frac{\sigma_{actif}}{\sigma_{SP500}}$$

Avec :

⁹ <https://www.oracle.com/ch-fr/big-data/what-is-big-data/>

¹⁰ <https://fr.swissquote.com/robo-advisory>

¹¹ SEILER Robert, Sélection de Gérants 20-21, Cours N°4, Analyse Quantitative, p.3

- $\rho_{(actif ; SP500)}$ La corrélation entre l'actif et l'indice
- σ Le risque (écart type) de l'actif ou de l'indice.

Nous pouvons définir des ratios, qui sont une bonne base de comparaison entre les portefeuilles ou indices. Le ratio de Sharpe correspond à la surperformance par rapport au taux sans risque, ajustée par rapport au risque. On le calcule comme suit :

$$Sharpe = \frac{r_{actif} - r_f}{\sigma_{actif}}$$

Afin de calculer les rendements et risques de nos portefeuilles, il est possible d'effectuer des calculs matriciels pour simplifier les choses. Le calcul du rendement du portefeuille est une simple multiplication des rendements moyens avec les poids dans le portefeuille.

$$r_{portefeuille} = R * w$$

Avec R la matrice des rendements moyens des actifs, w la matrice des poids des actifs. Pour calculer cette matrice de poids, il faut calculer les rendements moyens à partir des rendements logarithmiques. Ainsi, on peut calculer pour chaque période le rendement comme suit :

$$logreturn = \ln \left(\frac{Prix_n}{Prix_{n-1}} \right)$$

Pour calculer la variance du portefeuille, nous utilisons la formule¹² matricielle suivante :

$$\sigma^2_{pf} = w^T * VarCoVar * w$$

Avec :

- σ^2_{pf} La variance du portefeuille
- w et w^T Respectivement la matrice des poids et la matrice transposée des poids
- $VarCoVar$ La matrice de Variance-Covariance

Le risque (volatilité) du portefeuille se calcule en prenant la racine carrée de la variance.

La matrice de Variance – Covariance s'obtient aisément sur Excel en calculant les covariances des rendements entre eux.

Il est désormais intéressant de parler de notions de gestion de portefeuille. Il existe plusieurs manières de gérer des actifs, et afin de simplifier l'analyse des données de notre partie pratique, nous décidons d'appliquer une stratégie « Buy and Hold », c'est-à-dire

¹² DUC François, Finance de Marché, Introduction à la gestion de portefeuille, 2019

acheter des actifs en début de période et définir une allocation (poids) selon une méthode particulière, et de garder cette composition intacte. L'allocation va forcément varier sachant que le cours des actifs va également varier, donc leur part dans le portefeuille variera. Il existe une autre stratégie qui permet de rééquilibrer les poids des actifs : le rebalancement. Le but est de vendre les actifs dont le poids après variation est supérieur au poids d'allocation initiale (à la suite d'une augmentation de cours), et d'acheter les actifs sous-pondérés. Il existe également d'autres stratégies, qui peuvent induire un changement de pondération ou d'actifs pendant la période de gestion.

Afin de trouver l'allocation des actifs du portefeuille, il existe des méthodes qui vont dépendre de l'investisseur (profil privé, âge, horizon-temps...)¹³, des objectifs de rendements, de volatilité, d'indicateurs ou de stratégie. Il est également possible d'utiliser un solveur afin d'obtenir une allocation qui corresponde à un certain critère, en se basant sur des rendements passés. On a notamment la création d'un portefeuille « Minimum Variance », qui, selon les critères de poids par actif, cherche à minimiser la formule de variance du portefeuille vue plus tôt. On peut chercher un portefeuille « Maximum Return », « Maximum Sharpe » ou même « Maximum Utility ». Ce dernier portefeuille n'est pas aussi évident dans son appellation, alors il nous faut l'expliquer. Cette « utilité » dépend d'un coefficient, le « Gamma », qui correspond à l'aversion au risque d'un investisseur. Ce coefficient multiplie le risque de l'actif ou du portefeuille, car si l'investisseur possède une grande aversion au risque, alors cette volatilité possède un impact plus fort mentalement et donc une plus grande importance. Maximiser cette fonction revient à chercher un très grand rendement (à risque limité) ou réduire le risque du portefeuille. Le but est donc de calculer :

$$\text{Max} (r_{\text{portefeuille}} - \gamma * \sigma_{\text{portefeuille}})$$

Enfin, nous pouvons parler du stockpicking, qui est le fait de sélectionner des actifs pour constituer un portefeuille. Il peut dépendre de modèles (internes à l'entreprise par exemple), d'intuitions, de prévisions ou même d'algorithmes de sélection. Ils peuvent être programmés selon plusieurs critères, tels que les secteurs GICS du SP500 (selon l'évolution de ces secteurs et des actions qui le composent), de données dites « fondamentales » telles que le P/E (Cours / bénéfice par action), le P/B (Cours / valeur comptable), l'analyse des Cash Flows et bien d'autres ratios. Pour notre partie pratique, nous créerons un algorithme de stockpicking qui se basera sur l'évolution de secteurs GICS, de valeurs clés liées ainsi que sur le P/E.

¹³ SNOPEK Lukasz, Construction et gestion de portefeuilles, p. 50 – 55, 2021

2. État des lieux de l'intelligence artificielle et de la gestion de portefeuille

Avant de passer à la partie pratique et de voir ce qu'il est possible d'effectuer avec des données réelles et mettre en œuvre nos connaissances, il est intéressant de connaître l'état d'intégration de l'intelligence artificielle dans le monde mais aussi en Suisse, dans le domaine de la finance (gestion de portefeuille, wealth et asset management).

2.1 État des lieux mondial

Nous allons dans un premier temps regarder l'adoption de l'intelligence artificielle au niveau mondial, afin d'obtenir une vue globale de quelques places financières.

2.1.1 Un exemple de cadre légal : États-Unis

Aux États-Unis, pays dans lequel l'intelligence artificielle possède une place très importante, les lois couvrant ce point ne sont pas abondantes. En effet, il n'existe à proprement parler pas de loi qui couvre ce sujet¹⁴. Il existe quelques articles et résolutions qui sont en cours de traitement dans chaque état (16 ont été publiées à ce jour en 2021 aux É-U) mais qui sont spécifiques à chaque utilisation de l'intelligence artificielle : confidentialité des données (Massachusetts : MA H.B. 142 : en cours), création de task forces (Nevada : NV S.B. 110 : en cours), vérification des décisions prises par l'intelligence artificielle (Michigan : MI H.B. 4439 : en cours)¹⁵, et bien d'autres.

Toutefois, l'intelligence artificielle ne doit évidemment pas violer certaines lois, certains actes ou principes fondamentaux : entre autres, Fair Credit Reporting Act, Civil Rights Act... Bien sûr, certaines propositions de lois ou de bills sont en cours de discussion mais il faudra attendre avant de les voir entrer en vigueur.

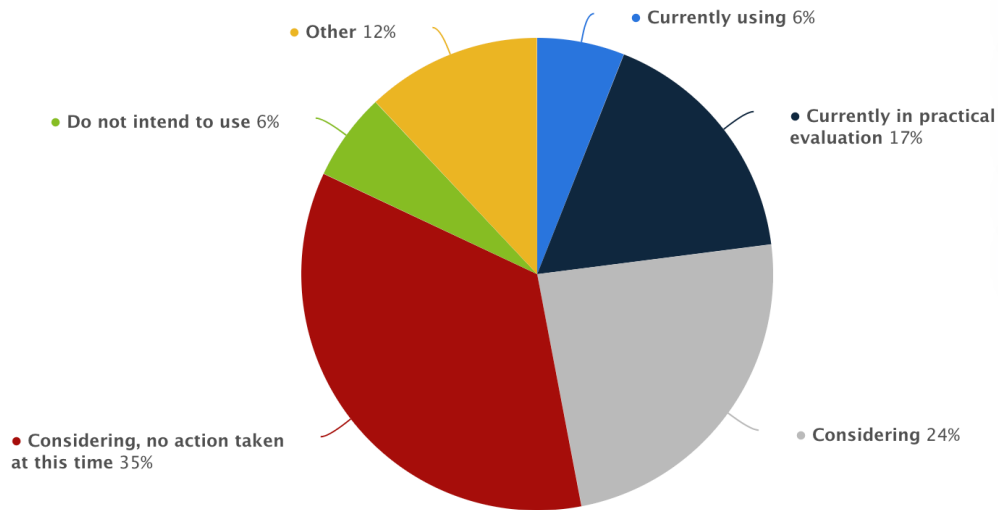
2.1.2 Constatations mondiales passées et actuelles

Afin de mieux se rendre compte de l'usage de l'intelligence artificielle dans le cadre de la gestion de portefeuilles dans le monde, voici un graphique en secteurs :

¹⁴ <https://www.mondaq.com/unitedstates/technology/1059776/artificial-intelligence-comparative-guide>

¹⁵ <https://www.ncsl.org/research/telecommunications-and-information-technology/2020-legislation-related-to-artificial-intelligence.aspx>

Figure 5 : Usage de l'intelligence artificielle en gestion de portefeuille, monde, 2020



(NORRESTAD F., Usage of artificial intelligence for portfolio management worldwide in 2020, Statista)

Nous constatons que d'après cette étude qui porte sur 47 assets et wealth managers, seulement 6% des gérants d'actifs utilisent une intelligence artificielle pour gérer les portefeuilles¹⁶. On remarque toutefois que 35% d'entre eux pensent à l'utiliser sans prendre de mesures pour autant, et 24% pensent et commencent à prendre des mesures, ce qui signifie tout de même que 59% évoquent cette possibilité. Pour 17% d'entre eux, la méthode est en cours d'évaluation. On remarque donc qu'en pratique, dans seulement 6% des cas l'intelligence artificielle est utilisée en pratique (avec tests : 23% au total), ce qui est faible. Il serait intéressant de mettre ces constatations en abyme en les confrontant à d'autres études et témoignages.

Si l'on souhaite se faire une idée des domaines financiers plus larges et de leur intégration de l'intelligence artificielle, une étude de McKinsey publiée en novembre 2020 nous indique que dans 8% et 6% des cas liés à la stratégie et finance d'entreprise, l'intelligence artificielle est la plus utilisée. Pour les domaines liés au risque (risk modeling, fraude et analyse de crédits), ces chiffres s'élèvent à 16% et 12%¹⁷.

¹⁶ <https://www.statista.com/statistics/1209707/usage-of-ai-for-portfolio-management-worldwide/>

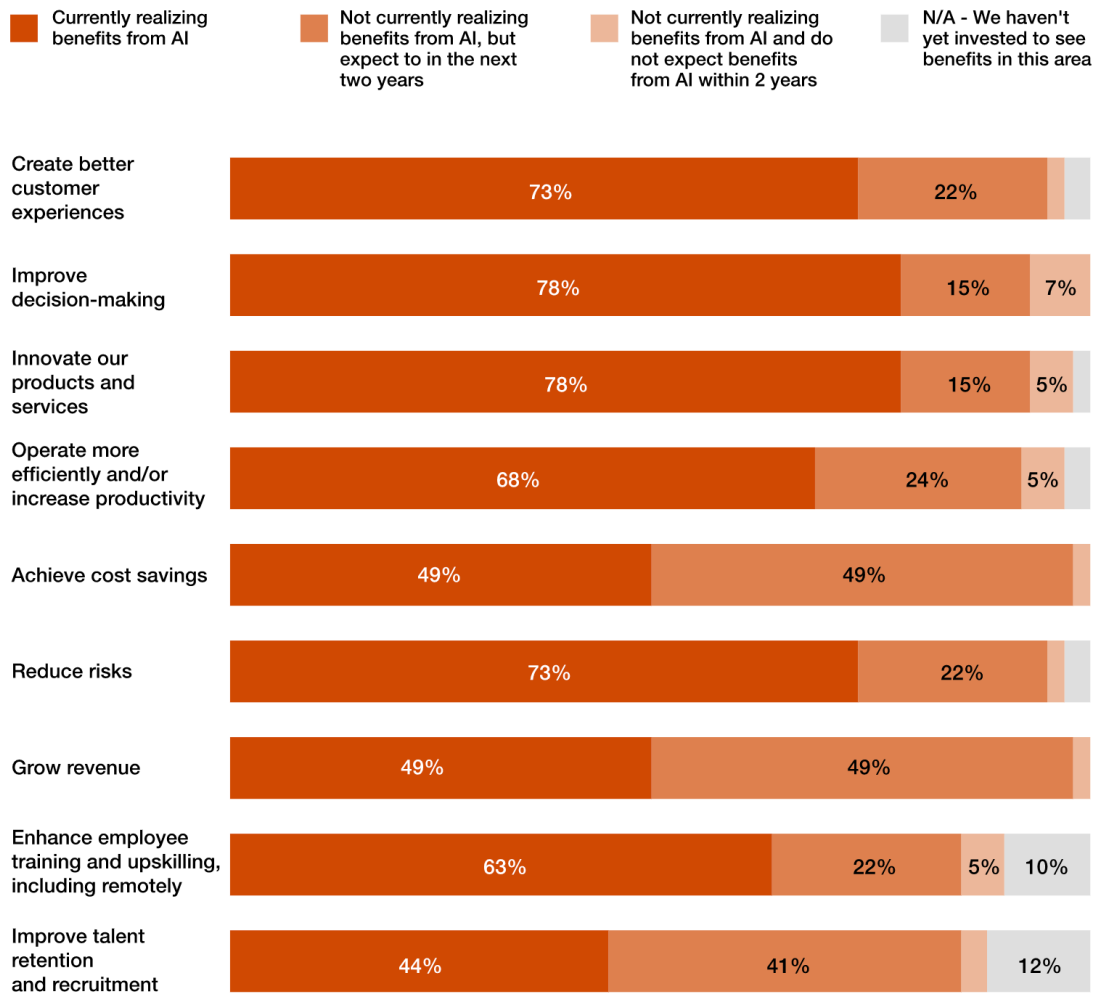
¹⁷ <https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/global-survey-the-state-of-ai-in-2020#>

Une étude de PwC pour 2021¹⁸, s'intéressant aux Wealth et Asset Managers des États-Unis et leur adoption de l'intelligence artificielle, nous donne des renseignements complémentaires fort intéressants. Une grosse majorité de répondants indique qu'utiliser de l'intelligence artificielle permet de prendre de meilleures décisions basées sur de l'information (donc en théorie, avec moins de biais humains), cela permet également une meilleure innovation mais aussi prendre moins de risque tout en mettant le client au centre et en améliorant les relations. Tout cela n'est possible que grâce aux quantités phénoménales de données que les assets et wealth managers possèdent. Il est également mentionné dans l'étude, donc par les institutions financières, que l'intelligence artificielle permet en effet de prédire des transactions, de mettre en place des prévisions assez fiables mais également d'optimiser des portefeuilles. Ce dernier point est important dans le cadre de ce travail, car ce sentiment avéré et constaté par de nombreux acteurs financiers est un point que nous souhaitons vérifier par l'analyse de données réelles. Dans le recueil des données effectué par PwC, une question nous semble très intéressante :

¹⁸ <https://www.pwc.com/us/en/tech-effect/ai-analytics/ai-predictions/asset-and-wealth-management.html>

Figure 6 : Part d'amélioration des secteurs internes par l'intelligence artificielle

AI helps AWM firms enhance the investor experience



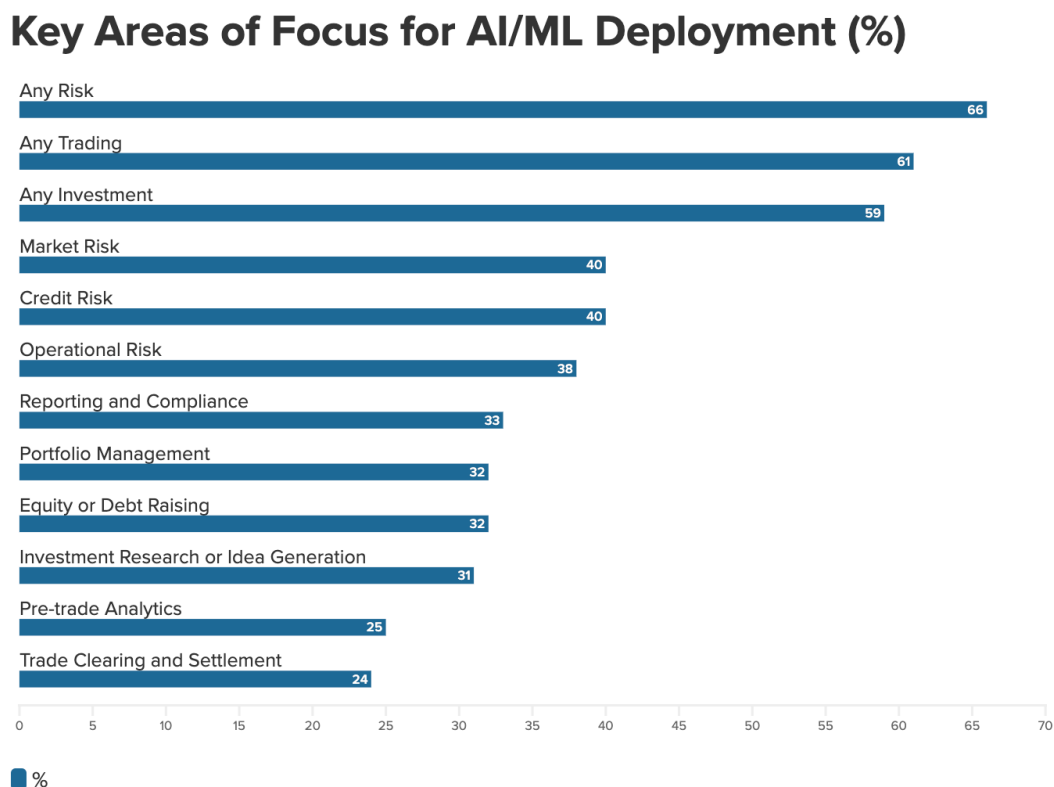
(PwC, AI for Asset and Wealth Managers in 2021, 2021)

Dans la catégorie « improve decision-making », 78% des entreprises voient des améliorations après l'utilisation de l'intelligence artificielle, 73% pour la réduction de risques. On remarque donc, d'après cette étude, que $\frac{3}{4}$ des entreprises jouissent de décisions éclairées et moins risquées en faisant confiance à l'intelligence artificielle dans le domaine de l'asset et wealth management. Toutefois, il existe certains risques qui ne sont pas négligeables. Il est mentionné dans l'étude, que la majorité des participants pense à un manque de confiance envers cette intelligence artificielle qui mènerait à de nouveaux risques. Ainsi, 71% des participants indiquent qu'un risque principal serait lié aux cyber-attaques ainsi qu'à la confidentialité des données. Un autre risque clé est que certains participants craignent de ne pas comprendre comment l'intelligence artificielle puisse prendre des décisions parfois critiques. Nous pouvons faire le parallèle avec la

citation suivante : « Never invest in a business you cannot understand » (Warren BUFFETT). Il faudrait en effet éviter d'investir dans un domaine que l'on ne comprend pas.

Un article dérivé d'une étude menée par Refinitiv en 2020 nous éclaire davantage sur les diverses tendances actuelles.

Figure 7 : Points de développement principaux pour le déploiement d'intelligence artificielle en finance



(Refinitiv, Artificial Intelligence / Machine Learning survey, Août 2020, p.9)

On constate que le déploiement d'intelligence artificielle dans un but de gestion de portefeuilles concerne 32% des participants¹⁹. De plus, la partie « any investment » qui comporte également la gestion de portefeuille au sens large s'élève à 59% des répondants. Ainsi, on constate qu'au sens strict du terme, un tiers des interrogés considèrent que dans le cadre d'intégration d'intelligence artificielle, la gestion de portefeuilles serait un point important. Cela semble aller dans le sens de nos constatations précédentes : une certaine part de gérants est intéressée, ce n'est pas la majorité mais il existe tout de même cette envie. Pour rebondir sur le chiffre de « 9% » de hedge funds utilisant de l'intelligence artificielle mentionné plus tôt, il semblerait que ces hedge funds

¹⁹ https://www.refinitiv.com/content/dam/marketing/en_us/documents/gated/reports/the-rise-of-the-data-scientist.pdf , p.9

utilisant l'intelligence artificielle performant mieux que les hedge funds ne l'utilisant pas : « 34% de rendement avec IA, Mai 2017 – Mai 2020 contre 12% sans IA » (Cerulli Associates, 2020). Attention toutefois aux chocs et marchés en dent de scie, où l'intelligence artificielle semble sous-performer selon l'article. Nous pouvons faire le parallèle avec la notion de Cygne Noir / Black Swan, nommée par Nassim Nicholas TALEB, qui signifie des chocs et crises, dont la probabilité d'occurrence est fortement sous-estimée (par la loi normale, majoritairement utilisée en finance).

Ces articles et études semblent nous montrer principalement qu'encore assez récemment en Amérique du Nord, l'intelligence artificielle dans l'asset et wealth management est assez discutée mais encore moyennement adoptée à grande échelle. Beaucoup considèrent cette option et tentent de préparer le terrain, mais sauter le pas semble difficile. Il en ressort toutefois que les avantages sont nombreux et que les bénéfices se font vite ressentir, mais il faut garder à l'esprit les divers risques et comprendre ce que l'on fait. Pour le domaine plus spécifique de la gestion de portefeuilles, l'adoption semble encore assez limitée. Il existe des solutions d'intelligence artificielle pour gérer les portefeuilles mais il semble que la majeure partie des solutions reste humaine.

La tendance dans les régions d'Asie, d'Australie et de Nouvelle-Zélande semble se confirmer : une étude de Shoreline indique que, parmi des asset managers et des investisseurs, 21% d'entre eux utilisent de l'intelligence artificielle (90% de ceux-ci sont des asset managers)²⁰, beaucoup prévoient d'investir dans le domaine et ceux qui ne l'utilisent pas n'en font pas une priorité ou ne savent comment l'intégrer.

2.1.3 Tendances mondiales prévues pour l'avenir

Grandview Research a publié un rapport intéressant sur l'intelligence artificielle dans l'asset management, des prévisions jusqu'en 2028. D'après le rapport, la taille de l'intelligence artificielle dans le marché de l'asset management s'élève à 1.48 milliards de dollars en 2020. Les prévisions semblent très optimistes, avec un taux de croissance annuel estimé à environ 37.1%, pour atteindre une taille de 13.43 milliards de dollars en 2028²¹. En 2019, 54.2% du marché de l'intelligence artificielle dans l'asset management était représenté par l'Amérique du Nord, qui semble être la partie mondiale la plus développée dans ce sujet depuis un certain temps. Ces premières prévisions semblent nous indiquer une tendance mondiale forte pour le développement de l'intelligence artificielle en finance, idéale pour prendre le train en marche.

²⁰ <https://shorelineawc.com/use-of-ai-in-asset-management/>

²¹ <https://www.grandviewresearch.com/industry-analysis/artificial-intelligence-asset-management-market>

Selon un article de Funds-Europe publié en décembre 2020, dérivé d'un article de leur journal dans l'édition de novembre 2020, une citation attire notre attention : « Applying AI in portfolio management is yet to take off on a wider scale, despite increasing use of the technology overall. »²². Il semblerait que les acteurs de la gestion de portefeuille préfèrent observer les résultats de gestions avec intelligence artificielle avant de se précipiter et remplacer leur gestion par de l'intelligence artificielle : le facteur humain reste clé et il sera difficile pour la gestion de portefeuille de passer au tout numérique.

2.2 État des lieux en Suisse

Nous allons désormais passer en revue le statut de l'intelligence artificielle au niveau Suisse à travers diverses études et divers sondages, puis au niveau du canton de Genève grâce à un témoignage d'une banque privée genevoise.

2.2.1 Cadre légal Suisse

Un projet important pour l'intelligence artificielle en Suisse est en cours : la révision de la Loi Fédérale sur la Protection des Données (LPD). Cette loi régit actuellement les points importants concernant l'utilisation de l'intelligence artificielle en Suisse. Cette révision, qui devrait prendre fin en 2022, pourrait inclure des points importants pour la finance. Un groupe de travail du SEFRI (Secrétariat d'État à la formation, à la recherche et à l'innovation) a été formé par la Confédération afin de songer à des propositions de modification de réglementations concernant l'intelligence artificielle. Ainsi, selon ce groupe et monsieur Christian Busch, il existe des améliorations possibles pour des décisions liées à l'intelligence artificielle et ayant un impact. Cette révision de la LPD pourrait inclure l'obligation d'informer une personne concernée par une décision prise par l'intelligence artificielle, si cette dernière implique une conséquence juridique ou affecte la personne concernée de manière forte. Ainsi, il ne serait pas impossible que dans le cadre de gestion de portefeuilles, une gestion avec intelligence artificielle pure implique une plus grande communication vis-à-vis des clients si cela implique certaines conséquences. On remarque également que l'intelligence artificielle est régulée en Suisse grâce à l'Ordonnance sur les infrastructures des marchés financiers (OIMF), qui interdit entre autres la manipulation des marchés avec des algorithmes de trading gérés par l'intelligence artificielle.

2.2.2 Constatations Suisses passées et actuelles

Un rapport très intéressant, publié par la Haute École de Lucerne, conjointement avec l'Asset Management Association Switzerland pour l'année 2020 nous indique quelques

²² <https://www.funds-europe.com/news/machine-driven-investing-the-future-of-portfolio-management>

points concernant la gestion d'actifs, et on peut trouver des mentions intéressantes concernant l'intelligence artificielle. Ainsi, 37.4% des sociétés suisses de technologies financières (FinTechs) utilisent des données provenant d'intelligence artificielle²³. En Europe et aux Etats-Unis, sur 59 asset managers, environ deux tiers d'entre eux utilisent de l'intelligence artificielle pour la gestion de leur données. Il est mentionné dans ce rapport que globalement, en Suisse ou dans le monde, l'intelligence artificielle en finance est un enjeu de taille.

Selon une étude d'Accenture en Suisse publiée en mai 2021, la majorité des gérants de fortune en Suisse reconnaissent les avantages de l'intelligence artificielle mais éprouvent de la difficulté pour la mettre en place (Accenture, 2021). Les gérants de fortune Suisses voient comme priorité liée à l'intelligence artificielle l'intégration du client dans ce processus, afin de mieux comprendre ses besoins et d'utiliser des données de qualité pour anticiper les besoins²⁴. Ainsi, il semble que les gérants de fortune en Suisse reconnaissent qu'un usage de l'intelligence artificielle soit bénéfique mais il n'est pas évident de la mettre en place.

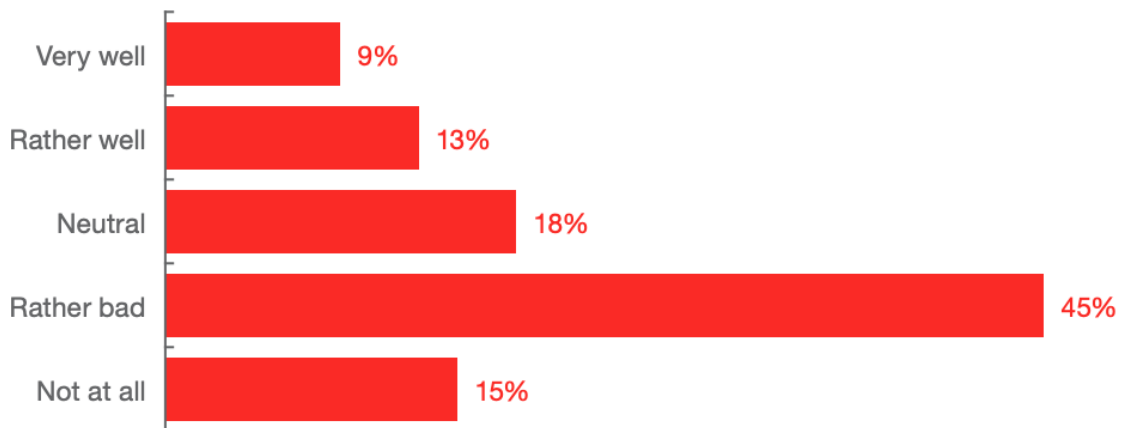
Une autre étude, de FintechNewsSwitzerland en 2020 nous indique d'autres éléments de compréhension pour les marchés Suisses / Allemands / Autrichiens : 80% des participants souhaiteraient utiliser l'intelligence artificielle afin d'effectuer de petites améliorations ou du marketing prédictif²⁵. Il est également mentionné que l'adoption de l'intelligence artificielle en Suisse est en retard par rapport à l'Asie ou l'Amérique du Nord (PwC, 2020). Cette information, qui provient d'une étude de PwC, est couplée à un autre point intéressant concernant la région Suisse / Allemagne / Autriche :

²³ https://www.am-switzerland.ch/assets/content/images/Swiss-AM-Study-2020_Webversion_EN.pdf

²⁴ <https://www.accenture.com/ch-en/insights/capital-markets/wealth-management-artificial-intelligence>

²⁵ <https://fintechnews.ch/aifintech/3-key-ai-capabilities-for-asset-management-applications/39904/>

Figure 8 : Disponibilité de l'intelligence artificielle dans les entreprises financières
CH / ALL / AUT



(PwC, How mature is AI adoption in financial services, 2020)

On remarque que 60% (45% + 15%)²⁶ des entreprises financières interrogées n'ont pas ou très peu d'intelligence artificielle prête à l'emploi.

Ces différentes études nous ouvrent les yeux sur la situation Suisse : cela nous confirme donc, que l'intelligence artificielle suscite un grand intérêt mais que la disponibilité et l'application ne sont pas totalement au rendez-vous. Même si la Suisse possède une belle part de marché dans la gestion d'actifs et de portefeuilles, elle semble être moins en avance que ses concurrents sur ce créneau.

2.2.3 Tendances Suisses prévues pour l'avenir

Plusieurs articles et témoignages nous indiquent clairement un point : l'intelligence artificielle dans la finance Suisse va prendre une place de plus en plus importante. Selon un premier article publié dans FintechNewsSwitzerland et BusinessInsider, UBS estimait sur un échantillon de 86 banques que les revenus des banques en 3 ans (2018 – 2021) pourraient augmenter de 3.4% et les coûts diminuer de 3.9% grâce à l'intelligence artificielle²⁷. On voit également de nombreuses entreprises développer des robo-advisors (des intelligences artificielles pour aider les clients dans des domaines tels que le support informatique ou l'investissement). De plus, des banques comme UBS décident de mettre en place des centres informatiques (UBS va créer un centre en Suisse) pour la gestion de données liées au Big Data²⁸.

²⁶ <https://www.pwc.de/de/future-of-finance/how-mature-is-ai-adoption-in-financial-services.pdf>

²⁷ <https://www.businessinsider.com/ai-in-financial-services-2017-11?IR=T>

²⁸ <https://fintechnews.ch/fintech/swiss-banks-accelerate-ai-adoption/16329/>

Un article publié par l'Association Suisse des Banquiers (ASB) indique que le volume de données devrait tripler entre 2021 et 2025, passant de 60 milliards de Téraoctet à 175 milliards de Téraoctet²⁹. A titre de comparaison, un disque dur standard actuel peut généralement stocker 1 Téraoctet de données. Cela indique l'opportunité grandissante liée à l'intelligence artificielle. Pour compléter cela, une interview de Damir Filipovic (Swissquote) nous révèle un autre acteur important pour l'avenir de l'intelligence artificielle : les caisses de pensions³⁰, qui pourraient bien recourir à cette méthode bénéfique.

Finalement, il semble que la tendance soit assez claire : l'intelligence artificielle va s'inscrire de manière notable dans les années à venir, mais il faudra pour cela que les acteurs financiers Suisse saisissent le changement.

2.2.4 Un témoignage d'une banque privée genevoise

Afin de mieux saisir les enjeux, les tendances à saisir et l'application pratique de l'intelligence artificielle, nous avons posé des questions à un Data Engineer d'une banque privée genevoise.

Nous commençons par la question : « que constatez-vous vis-à-vis de l'intelligence artificielle dans le secteur bancaire Suisse ? Et dans votre banque ? ». Notre contact nous indique avoir l'impression d'une modernisation des banques suisses sans toutefois innover de manière notable. Les banques mettent en effet en place des initiatives technologiques pour moderniser leurs applications ou leur business, mais peu de changements forts et disruptifs tels que l'intelligence artificielle sont effectués. Notre contact nous indique également que la notion d'intelligence artificielle est assez floue dans certaines banques, la confusion entre intelligence artificielle, machine learning ou même simple automatisation étant aisée. Cela nous permet de faire le parallèle avec la tendance mentionnée précédemment : la peur de se lancer dans un domaine tel que l'intelligence artificielle, si l'on ne le comprend pas assez. Nous discutons enfin d'un point tout aussi intéressant : les ChatBots. Ils soulèvent cette même interrogation concernant la compréhension de la technologie : est-ce vraiment de l'intelligence artificielle ? Notre contact nous indique que dans son emploi, cette technologie n'est pas encore utilisée mais elle l'est par exemple chez Revolut. Il termine en nous indiquant que l'idée d'intelligence artificielle existe bel et bien dans les banques suisses, mais il faudra attendre encore car la prudence est de mise.

²⁹ <https://www.swissbanking.ch/en/news-and-positions/opinions/the-future-of-the-financial-industry-banking-on-data>

³⁰ <https://actu.epfl.ch/news/artificial-intelligence-in-finance-3/>

A la question « Que pensez-vous de l'avenir de l'intelligence artificielle dans la gestion de portefeuilles en Suisse et dans votre banque ? », notre interlocuteur nous indique constater une certaine tendance qui se dirige dans ce sens, en mentionnant l'exemple du robo-advisor / trading bot de la banque Swissquote. Il voit également une tendance liée aux FinTechs, qui pourraient à l'avenir proposer ce genre de services pour concurrencer des traders et portfolio managers. Toutefois, il pense qu'à moyen terme, les activités de trade sont plus concernées que les portfolio managers. Il termine en indiquant que dans sa banque, l'humain est au centre et qu'il s'agit d'un point qui fait la différence.

Nous avons posé une question tout aussi financière que sociétale à notre interlocuteur : « A votre avis, l'intelligence artificielle peut-elle remplacer complètement l'humain dans la gestion de portefeuilles ? Pourquoi ? ». Il nous a répondu qu'en effet, l'intelligence pourrait remplacer un humain dans ce cadre, mais de manière partielle. Il mentionne les avantages, tels qu'éviter de prendre des décisions irrationnelles biaisées par les émotions, et que la possibilité de surperformance de l'intelligence artificielle face à un trader est possible. Notre contact nous indique que même si à long terme, l'intelligence artificielle serait utilisée par de nombreux acteurs, l'humain (le gestionnaire de portefeuilles) pourrait toutefois garder sa place avec plusieurs buts : améliorer son algorithme, l'entraîner, et même modifier ses paramètres pour que son algorithme batte les autres algorithmes.

Concernant les avantages de l'intelligence artificielle, notre interlocuteur reprend les arguments concernant la rationalité des décisions et ajoute la possibilité d'intégrer l'analyse de sentiments du marché de manière algorithmique. Il voit toutefois un inconvénient majeur, qui se trouve être la difficulté de prédiction d'évènements à venir. Il nous indique que l'intelligence artificielle se base sur des données passées. Notre contact termine en ajoutant que l'humain et l'intelligence artificielle sont complémentaires : l'humain peut aider à améliorer l'algorithme, et l'algorithme peut aider l'humain pour automatiser et prendre des décisions.

Dans sa banque privée, notre interlocuteur nous indique que l'intelligence artificielle n'est proposée à ses clients, ni dans un cadre de gestion de portefeuilles, ni dans un cadre de support, même si des solutions ont été envisagées et testées. Il pense que les jeunes préféreront la machine et que les générations plus âgées préféreront l'humain. Il nous confie qu'aujourd'hui, il choisirait l'humain car l'intelligence artificielle peut être encore limitée pour des raisons qui sont souvent techniques : sécurité ou droits d'accès notamment. La part d'intelligence artificielle dans sa banque est encore faible mais elle risque de croître avec les années à venir, cette proportion étant dépendante de

contraintes techniques mais également (et fortement) politiques. La crise du COVID n'a également pas eu d'impact notable sur ce secteur dans la banque.

Ce témoignage confirme les tendances précédentes tout en ajoutant une mention importante, tirée de témoignages du quotidien : l'adoption de l'intelligence artificielle dans les banques suisses n'est pas aussi forte que l'on pense.

2.3 Synthèse basée sur la recherche

A la lumière des études, articles, sondages et interviews mentionnées, nous comprenons mieux la situation et les tendances majeures qui sont liées.

On constate, aussi bien d'un point de vue mondial que d'un point de vue Suisse, la tendance de digitalisation qui est croissante. Les indicateurs et analyses prévoient des flux de données de plus en plus importants et il semblerait que des économies soient possibles en utilisant l'intelligence artificielle. Dans un contexte où le Big-Data est omniprésent, où les acteurs financiers resserrent leurs équipes, il serait intéressant de saisir ce changement. Intégrer de l'intelligence artificielle de manière réfléchie et compréhensible va probablement devenir un atout et dans certains domaines, tels que la gestion de portefeuilles, beaucoup d'acteurs pensent à sauter le pas et la tendance ne devrait probablement pas diminuer.

Même si les témoignages laissent penser qu'une grande implication et de forts développements sur le thème de l'intelligence artificielle sont en cours, nous constatons qu'en pratique, la mise en place est plutôt difficile et l'adoption à grande échelle, qui n'est toutefois pas impossible, est à prévoir pour dans plusieurs années.

Il existe de nombreuses difficultés, et cela commence par une compréhension difficile de ce qui est, ce qui n'est pas, ce que peut ou ne peut pas faire de l'intelligence artificielle. Les possibilités sont grandes et la vulgarisation, ou des cours pour mieux comprendre le domaine sont, selon nous, des possibilités intéressantes à mettre en place.

3. Comparaison pratique de portefeuilles gérés avec et sans intelligence artificielle

L'idée derrière cette partie est de mettre en pratique les connaissances notamment vues à la HEG et de se forger une opinion à partir de données réelles. Dans ce cadre, nous utiliserons le langage de programmation Python. Nous le compléterons suivant nos besoins à travers l'usage de bibliothèques à importer, comme DeepDow, une bibliothèque spécialisée dans la résolution de problèmes liés à l'optimisation financière avec deep learning. Nous utiliserons également Pandas afin de mettre en forme des tableaux pour la récupération et l'analyse des données. Le reste de l'analyse se fera sur Excel, et nous regrouperons les données intéressantes dans des tableaux simplifiés.

3.1 Buts recherchés et postulats de départ

Cette comparaison cherche à déterminer, si dans le cadre d'actions américaines (tirées du SP500), notre portefeuille stockpické de manière traditionnelle (algorithme avec critères) est plus efficace (de plusieurs points de vue : risque, rendement entre autres) qu'un portefeuille stockpické par intelligence artificielle après optimisation.

Afin de simplifier nos démarches et pour faciliter l'implémentation des données (notamment dans l'algorithme d'optimisation), nous définissons plusieurs postulats de départ.

Le premier postulat implique que le style de gestion sera de type « Hold Only ». Ainsi, ce style de gestion évitera les changements de positions pour donner suite à des variations du marché, et permettra de garder un même portefeuille du début de la simulation jusqu'à la fin de la période, et permettra une analyse plus aisée des actifs suivant le type d'optimisation.

Le deuxième postulat est dépendant de l'algorithme d'optimisation DeepDow. Afin d'optimiser les portefeuilles, l'algorithme a besoin de données moyennes journalières ainsi qu'une matrice de Variance – Covariance. De ce fait, avec le rendement journalier moyen, le calcul des ratios de Sortino et d'Information est plus compliqué, car le ratio de Sortino implique dans son calcul la volatilité négative (volatilité des rendements négatifs), et il faut donc utiliser toutes les données de rendement et non les valeurs moyennes. Pour le ratio d'Information, il faut calculer le Tracking Error, qui se calcule également avec les données complètes. Il reste toutefois possible de les calculer pour des portefeuilles ou des classes d'actifs qui sortent de l'ordinaire, en prenant après optimisation les données journalières et de calculer ces ratios pour ensuite ajouter une nouvelle base de comparaison.

Le troisième postulat dépend donc du deuxième : afin de comparer nos portefeuilles après optimisation, ayant des données moyennes, nous allons donc utiliser le ratio de Sharpe. Afin d'entrer dans les détails, nous verrons sur Excel au cas par cas d'autres détails de comparaison tels que les actifs par portefeuille ou les données liées à ces actifs.

Le quatrième postulat est lié au premier, et concerne le rebalancement des positions. Dans notre cas, les positions ne seront pas rebalancées. Cela évitera de changer les pondérations en plein calcul d'algorithme à la suite de l'évolution des marchés et permettra de simplifier l'analyse.

Le cinquième postulat concerne les portefeuilles simulés. Étant donné que notre algorithme de stockpicking permet de sélectionner 18 actions (les 3 meilleures actions selon leur P/E sélectionnées par secteur, parmi les 6 meilleurs secteurs GICS), nous choisissons donc 18 actions pour constituer nos portefeuilles afin de simplifier les comparaisons.

Le sixième postulat concerne l'optimisation et les poids. Nos portefeuilles seront optimisés suivant les deux contraintes ci-dessous (Bien qu'il existe des contraintes, DeepDow appelle cette optimisation *unconstrained*) :

- $\sum_{n=0}^N w_n = 1$: pas de levier
- $w_n \in [0 ; 1]$: pas de short selling

C'est-à-dire, la somme de tous les poids doit être égale à 1 (portefeuille 100% investi, pas de levier), et les poids des actifs varient entre 0 inclus et 1 inclus (pas de short selling) : il est donc possible de ne pas investir dans un actif, et il est possible d'être investi dans un seul actif parmi tout le portefeuille. Cela permet de ne pas investir dans des actifs qui apportent peu à notre portefeuille mais également de réduire la charge du processeur, qui dans une volée d'actifs, doit optimiser 6 fois chaque portefeuille (3 types d'optimisation recherchée : Minimum Variance, Maximum Return, Maximum Utility ; 2 méthodes d'optimisation : Markowitz et Deep Learning, donc $3 * 2 = 6$) de 18 actifs (notre algorithme de stockpicking traditionnel sélectionne les 3 meilleurs actifs selon leur P/E pour chacun des 6 meilleurs secteurs GICS, donc $3 * 6 = 18$), et ce sur 100 portefeuilles, mais également de faciliter l'analyse des données. On constate ici qu'avec un processeur 8 cœurs, le temps d'optimisation d'un portefeuille et d'écriture des données est d'environ 20 secondes. Cela nous donne une moyenne de 33 minutes pour optimiser nos 100 portefeuilles.

Un septième postulat vient détailler notre analyse de données. Notre but est de comparer des portefeuilles entre eux, et il convient de détailler ce que l'on considère être une

différence statistiquement significative. Diverses recherches et lectures, notamment un rapport de 1999 de l'Union Européenne concernant l'introduction de l'Euro et l'arrondi des monnaies, nous indique qu'aucun consensus n'est fixé pour ce domaine, et cela n'a pas encore changé³¹. Dans notre cas, étant donné que nous avons utilisé un arrondi à 6 chiffres après la virgule afin de vérifier la concordance et l'exactitude des calculs effectués par l'algorithme de Deep Learning, nous allons réutiliser cet arrondi. De plus, nous nous basons sur des portefeuilles de 18 actifs, il existe donc beaucoup de possibilités et même si la répartition des actifs est assez différente, il se peut que les résultats à comparer (ratios par exemple) soient proches.

3.2 Création des divers algorithmes

Pour utiliser Python de manière ordonnée et simplifiée, nous utiliserons Jupyter, un logiciel qui permet de programmer en Python sur des « Notebook », des fichiers pouvant comporter plusieurs algorithmes ou parties d'algorithmes, exécutables un par un ou en ensemble. Cela rendra également plus simple la visualisation des données ainsi que l'exportation de résultats.

Dans un souci de simplification et de facilité pour exécuter les algorithmes suivant le besoin, ces derniers ont été disposés dans plusieurs fichiers notebook Jupyter.

L'algorithme lié à l'optimisation Deep Learning a été importé depuis le site de développeur du module, et a été adapté et en partie vérifié afin d'intégrer nos données et nos boucles.

3.2.1 Algorithme de stockpicking simple

A titre de comparaison avec l'algorithme de Deep Learning et le potentiel portefeuille stockpické par intelligence artificielle que nous allons obtenir, nous avons programmé un algorithme de stockpicking traditionnel simple sur la base du SP500. Après des discussions avec le conseiller au travail de bachelor et à la lumière de lectures d'articles financiers, nous avons décidé d'inclure les P/E d'actions ainsi que les données de rendement des secteurs GICS du SP500, mais aussi les données de commodités ou d'indices liés spécifiquement à ces secteurs, donc liés aux actifs.

Cet algorithme est divisé en 5 parties afin de rendre le code plus clair, plus facile à débiter (il est dépendant de données externes, ce qui nous rend dépendant à ces changements) et plus simple à exécuter pas à pas.

La première partie importe les tickers, les P/E et les secteurs GICS correspondants afin de servir de base pour le stockpicking.

³¹ https://ec.europa.eu/economy_finance/publications/pages/publication1224_en.pdf

La deuxième partie importe des données de rendement (journalier, mensuel et recommandations d'achat) des actifs et les ajoute à la liste correspondante.

La troisième partie importe les données de rendement des secteurs, commodités et indices et les ajoute à la liste correspondante.

La quatrième partie compare les secteurs par le biais d'un barème de notation. Pour ce faire, nous nous sommes basés sur nos précédentes discussions et lectures, et nous avons adapté ces points mais aussi cette notation. Ainsi, un rating de « fort achat » donne 25 points, tandis qu'une « vente » rapporte 0 points. Pour les rendements, un rendement supérieur ou égal à 10% donne 25 points par exemple, un rendement compris entre 3% inclus et 5% non inclus rapporte 9 points, tandis qu'un rendement négatif n'en apporte pas. Sachant qu'il y a 1 rating, 1 rendement de secteur et 2 rendements de valeurs liées, il y a 4 points à évaluer et la somme donne 100. Ainsi, un secteur possède une note variant de 0 sur 100 (pire) à 100 sur 100 (meilleur).

La cinquième partie effectue le stockpicking. Nous prenons la moyenne de la note des secteurs et sélectionnons ceux qui ont une note supérieure ou égale à cette moyenne. Après avoir sélectionné ces secteurs, nous sélectionnons les 3 meilleurs actifs (selon leur P/E, le plus faible possible) par secteur, et dans notre cas, l'algorithme sélectionne 6 secteurs et crée donc un portefeuille de 18 actifs.

Pour faire suite à une impossibilité de télécharger certains rendements d'actifs, nous remplaçons ceux-ci par les suivants dans la liste. La liste finale est donc celle-ci :

['CE', 'SEE', 'NEM', 'ALL', 'UNM', 'AFL', 'KIM', 'BXP', 'EQR', 'INTC', 'WU', 'HPQ', 'DISH', 'OMC', 'VZ', 'OKE', 'COG', 'KMI']

3.2.2 Algorithme d'optimisation Markowitz et Deep Learning

Ayant des connaissances en économie d'entreprise et non en Intelligence Artificielle, après lecture de divers articles de programmation d'optimiseurs utilisant l'intelligence artificielle sous Python, il nous est apparu de manière claire qu'apprendre les connaissances nécessaires serait assez ardu. Après des recherches dans les diverses bibliothèques présentes sous Python, DeepDow est apparue comme la solution de notre algorithme. Il s'agit d'une bibliothèque complète spécialisée dans le deep learning appliqué à des problèmes financiers. Sur leur site, les développeurs ont mis à disposition un exemple d'algorithme, utilisant leur bibliothèque afin d'optimiser des portefeuilles en utilisant deux méthodes : un solveur simple (Markowitz) et du Deep Learning, afin de prédire les données. Cet algorithme servira donc de base afin de créer nos comparaisons.

3.2.2.1 Détail et fonctionnement de l'algorithme

Cet algorithme d'optimisation sous Markowitz et Deep Learning fonctionne principalement par fonctions. Il prend en input des données moyennes (matrice des rendements moyens journaliers et matrice Var-CoVar).

On retrouve plusieurs classes, notamment la classe « Net » : elle va créer les réseaux neuronaux en se basant sur nos données de rendements moyens afin d'effectuer des prédictions et s'occuper de l'entraînement et apprentissage de l'algorithme. La classe « Markowitz » quant à elle, définit les critères de solveur pour nos portefeuilles : minimum variance, maximum return et maximum utility.

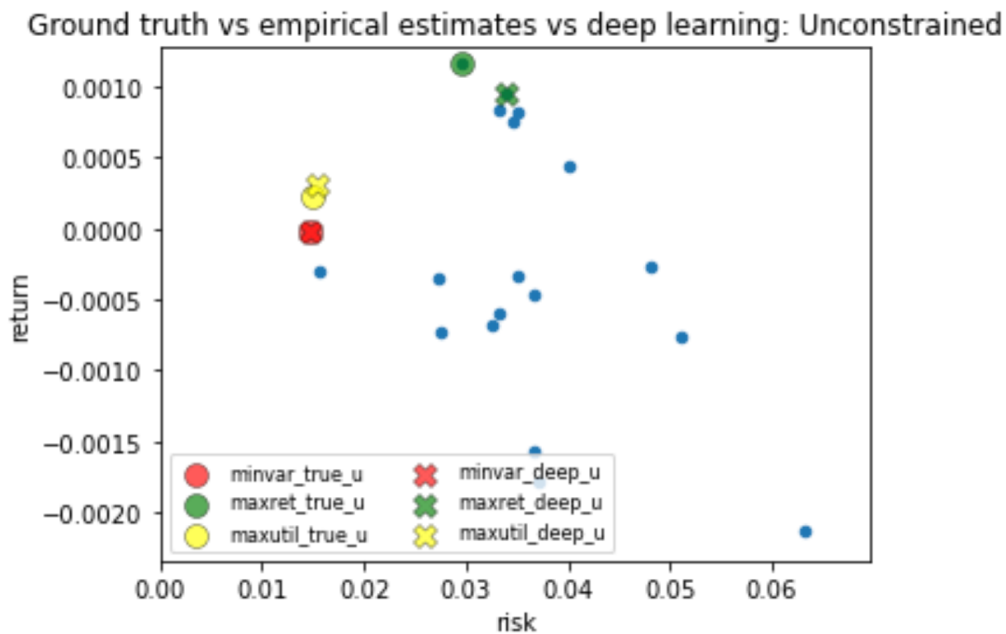
Pour chaque portefeuille et chaque type, l'algorithme les optimise avec Markowitz et avec Deep Learning et nous donne en output les données demandées ainsi que des graphiques pour comparer les valeurs.

3.2.2.2 Vérification de la validité de l'algorithme

Cet algorithme n'étant pas programmé et écrit par nos soins, il convient donc de vérifier si les calculs qu'il effectue (et qui sont vérifiables, ie. Les calculs de solveur selon Markowitz) sont reproductibles sur un logiciel tel qu'Excel. De ce fait, nous choisissons un portefeuille de 18 actions et nous allons l'optimiser sur Excel et par l'algorithme. Nous choisissons à titre d'exemple, le portefeuille généré par notre algorithme de stockpicking.

Il faut quelques mises en forme avant de commencer le calcul des données, notamment la création d'une matrice Var-CoVar ainsi qu'une matrice des rendements moyens journaliers par actif. Une fois cela fait, l'algorithme ayant été adapté au préalable par nos soins afin de pouvoir utiliser n'importe quel portefeuille d'actifs, nous exécutons le code et obtenons le graphique suivant :

Figure 9 : Portefeuille stockpické optimisé avec l'algorithme DeepDow



(DeepDow, RICHARD Nicolas, 2021)

Une modification de la fonction de graphique de l'algorithme nous permet d'extraire les données, que nous avons inscrites dans un fichier excel. Sur ce fichier, nous calculons la matrice Var-CoVar ainsi que le rendement journalier moyen. Nous préparons deux colonnes, l'une pour le portefeuille minimum variance et l'autre pour le portefeuille maximum return. L'algorithme DeepDow calcule un troisième portefeuille nommé « maximum utility », que nous n'optimiserons pas ici.

Nous lançons donc le solveur Excel avec les paramètres suivants pour nos cas :

- Min Var : objectif : minimiser le risque ; somme des poids = 1 ; poids compris entre 0 et 1 inclus
- Max Ret : objectif : maximiser le rendement ; somme des poids = 1 ; poids compris entre 0 et 1 inclus

Une fois les résultats obtenus, nous calculons l'arrondi à 6 chiffres des valeurs Excel et de l'algorithme, puis nous vérifions si elles concordent. Dans notre cas, nous constatons qu'elles concordent effectivement. Nous en déduisons que cet algorithme fonctionne correctement pour la partie solveur simple et nous formulons l'hypothèse que les calculs liés au Deep Learning, difficiles à vérifier sans connaissances dans le sujet, seront effectivement corrects.

Figure 10 : Tableau de comparaison de solveur Excel / DeepDow

	Min Var	Max Ret
Risk (std)	1,4631%	2,9554%
Return	-0,0016%	0,1158%
Risk (std) DeepDow	1,4631%	2,9554%
Return DeepDow	-0,0016%	0,1158%
Risk Arrondi : 6	VRAI	VRAI
Return Arrondi : 6	VRAI	VRAI

(RICHARD Nicolas, BackTest-algo-IA, 2021)

3.2.2.3 Création des portefeuilles

Dans le but de stockpicker des actifs grâce à l'intelligence artificielle, il nous faut des portefeuilles à optimiser de manière traditionnelle (Markowitz) et par intelligence artificielle (Deep Learning) dans notre algorithme, pour ensuite sélectionner les actifs.

Afin de simplifier les calculs liés à la moyenne de rendement des actifs, nous avons créé un algorithme simple qui parcourt la liste des actifs du SP500, puis les insère dans un tableau créé avec la bibliothèque Pandas. Ensuite, nous calculons dans un autre tableau les log-returns des actifs et nous enregistrons ce tableau dans le dossier Python qui comporte nos fichiers, afin de ne pas avoir à télécharger 1 an de données journalières pour un peu moins de 500 actifs, à chaque itération de notre algorithme.

Figure 11 : Extrait de code Python qui calcule les log-returns

```
Entrée [47]: ##### Récup liste rdts pour chaque actif du SP
import pandas as pd
import json
import yfinance as yf

all_stocks = pd.DataFrame()

for ticker in liste_sp500:
    tmp_close = yf.download(ticker,
                           start='2020-02-15',
                           end='2021-02-15',
                           progress=False)['Close']
    all_stocks = pd.concat([all_stocks, tmp_close], axis=1)
all_stocks.columns=liste_sp500

log_returns = np.log(all_stocks/all_stocks.shift(1))
log_returns = log_returns.tail(log_returns.shape[0] -1) #Eliminer première ligne de NaN
log_returns.to_csv(r'TEMPTEST/logreturns-SP.csv', index=False)
```

(RICHARD Nicolas, IA - Modifiée + algo pour boucle, 2021)

Le but est ensuite de créer les portefeuilles. Nous avons décidé de créer 100 portefeuilles contenant 18 actifs (comme notre portefeuille stockpické) tirés au hasard depuis la liste du SP500 modifiée, tout en fixant trois conditions afin d'assurer une certaine logique et stabilité :

- Les portefeuilles doivent être tous différents
- Pas de doublons d'actif dans un même portefeuille
- Éliminer les actifs dont les données ne peuvent être importées via Python

Après avoir tiré au hasard ces 100 portefeuilles uniques de 18 actifs uniques, ils sont enregistrés afin de les utiliser plus tard.

Figure 12 : Code de création des portefeuilles

```
Entrée [6]: ##### Charger les actifs du SP500
import pandas as pd
import json
from random import randrange

tbl_sp = pd.read_csv('Tableau SP500 GICS avec PE.csv')
liste_sp500 = tbl_sp['Symbol'].tolist()
liste_sp500.remove('BRK.B')
liste_sp500.remove('BF.B')
liste_sp500.remove('STX')
liste_sp500.remove('TDY')
len_sp = len(liste_sp500)

liste_actifs = []
liste_pfs = []

while len(liste_pfs) < 100:
    for nb_actif in range(18):
        alea = randrange(len_sp)
        while liste_sp500[alea] in liste_actifs:
            alea = randrange(len_sp)
        liste_actifs.append(liste_sp500[alea])
    if not liste_actifs in liste_pfs:
        liste_pfs.append(liste_actifs)
    liste_actifs = []

with open("TEMPTEST/liste_pfs.json", 'w') as f:
    json.dump(liste_pfs, f, indent=2)
```

(RICHARD Nicolas, IA - Modifiée + algo pour boucle, 2021)

3.2.2.4 Intégration de l'algorithme dans une boucle, output, mise en forme

Pour traiter nos portefeuilles, nous programmons une boucle qui, pour chaque portefeuille, s'intègre à l'algorithme de deep learning et enregistre les outputs dans un tableau de comparaison en effectuant une comparaison sur la base du ratio de Sharpe.

Figure 13 : Snapshot du tableau résultant de l'optimisation

Risk Max util deep	Return Min var deep	Return Max return deep	Return Max util deep	Sharpe minvar deep	Sharpe maxret deep	Sharpe maxutil deep	Better pf minvar	Better pf maxret	Better pf maxutil	Better pfs	Nb Better pfs
0.0185397267258655	0.0004131784059202852	0.002640224940930113	0.0015646199286527198	-0.0348664542298359	0.02716253132380968	0.030707766945379425	Traditionnel	Traditionnel	IA	Traditionnel	2.0
0.018840920312641538	9.494623145016173e-05	0.0018685317845349887	0.0006245794729652262	-0.05032057124704552	0.02931208810133865	-0.019676991290764332	Traditionnel	IA	Traditionnel	Traditionnel	2.0
0.0200262549512208	0.0007758712431704559	0.005902057443884513	0.0018263123094736469	-0.0121212834704965	0.1057052870335374	0.0414958196580846	Traditionnel	IA	IA	IA	2.0
0.019722165993975936	0.00039157147640626546	0.0025198759992937545	0.0005328786841460115	-0.03137647896430827	0.039642330867082534	-0.023447108273310358	IA	IA	Traditionnel	IA	2.0
0.01813898637690235	0.0005976407113783507	0.0025198759992937545	0.0012252624531258033	-0.024140785874314143	0.039642330867082534	0.012678843262940838	IA	Traditionnel	IA	IA	2.0
0.01908909357785373	0.00029108100491557365	0.0015122489092389447	0.00027910235989390926	-0.0374325522330575	0.01589453317315043	-0.03751902010307928	Traditionnel	Traditionnel	Traditionnel	Traditionnel	3.0
0.01696217970589631	0.0004148667036351383	0.002219741459911124	0.0007775456842362186	-0.03565093780854816	0.0314402248921984	-0.012838017585620172	Traditionnel	Traditionnel	Traditionnel	Traditionnel	3.0
0.01812549285591744	0.00033466241760456465	0.0023960440568334355	0.0006409524681234675	-0.03733770993598296	0.0323175459411809	-0.01955003266859093	Traditionnel	Traditionnel	Traditionnel	Traditionnel	3.0
0.014990769705972575	0.0010305888435742652	0.002279790265686836	0.0019058107315069325	0.002702569220628684	0.07134095178865059	0.06073766026559505	Traditionnel	IA	IA	IA	2.0
0.018440559186137663	0.0006578151533222692	0.002012105590763135	0.0015035613537276885	-0.02022398533643672	0.03229150047520457	0.02756179479192769	IA	Traditionnel	IA	IA	2.0
0.017570230213231986	0.00035876149609365496	0.0016953716843777144	0.0007114559205520779	-0.03771212812370545	0.02118973254573476	-0.016155196693161258	IA	Traditionnel	Traditionnel	Traditionnel	2.0

(DeepDow, RICHARD Nicolas, Tableau Comparaison Normal vs Deep, 2021)

Nous complétons l'algorithme en programmant une partie qui calcule l'alpha et le bêta de chaque actif de chaque portefeuille. Nos portefeuilles étant basés sur le SP500, nous utilisons les valeurs du SP500 pour calculer les alphas et bêtas. Enfin, nous indiquons quel type d'optimisation est majoritaire dans le portefeuille (après optimisation, pour chaque portefeuille, des 6 types, nous regardons si le type Markowitz ou Deep Learning possède le plus grand Sharpe).

3.2.3 Algorithme de frontière efficiente

Grâce au module FinQuant, nous programmons un algorithme qui permet de tracer une frontière efficiente, afin d'illustrer cette notion financière en pratique mais aussi de l'appliquer à nos portefeuilles stockpickés.

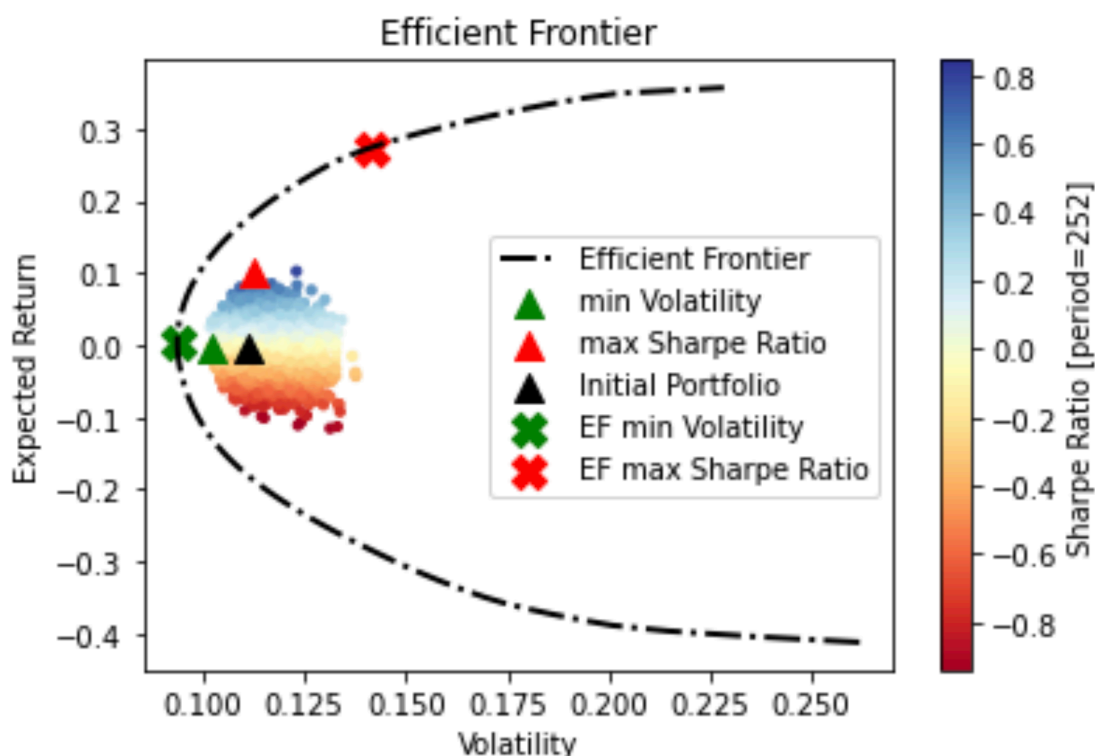
Le but est de simuler des portefeuilles, selon les actifs que nous lui donnons, avec des poids différents pour constituer le nuage de points (ici, du rouge au bleu) avec la méthode de Monte Carlo, pour tracer cette frontière efficiente. Nous limitons le nombre de portefeuilles simulés à 5'000 tout en sachant que nous cherchons avant tout à tracer la frontière.

Comme mentionné en introduction, pour tracer cette frontière efficiente, il faut résoudre une équation qui, pour chaque niveau de volatilité, donnera le rendement maximal (en fonction des poids).

Cet algorithme indique les portefeuilles « minimum variance » et « maximum Sharpe » dans les portefeuilles simulés et sur la frontière efficiente, ainsi que les Sharpe Ratio pour chaque portefeuille simulé.

Ainsi, pour notre liste stockpick, nous obtenons le graphe suivant :

Figure 14 : Frontière efficiente du portefeuille stockpické standard



(RICHARD Nicolas, RICHARD-Nicolas_TB_PYTHON V3.0.0, 2021)

Nous constatons que le rendement maximal du portefeuille s'établit à environ 35% pour 23.5% de risque. En se basant sur la frontière efficiente, notre portefeuille stockpické peut s'attendre à une volatilité minimale d'environ 10%. Il est intéressant de noter que les portefeuilles simulés à partir de ces actifs stockpickés de manière traditionnelle ont un ratio de Sharpe (du moins, ces 5000 portefeuilles) inférieur à 1 mais une majorité de portefeuilles possède une valeur aux alentours de 0.5 : cette valeur est à améliorer mais il s'agit d'un bon début.

3.3 Utilisation théorique des algorithmes

Cette partie permet de comprendre ce que nous avons effectué afin d'obtenir nos données. Plus tard, nous adapterons le code et le tirage de portefeuille au besoin afin de créer une nouvelle liste d'actifs à importer, afin de spécialiser nos tirages et de créer des cas spécifiques.

3.3.1 Tirage des données : portefeuilles

Nous avons tout d'abord exécuté le code permettant d'obtenir 100 portefeuilles. A ce stade, nous tirons 100 portefeuilles au hasard de la liste complète du SP500 (retranchée des actifs dont les données ne sont pas disponibles par notre algorithme). Il s'agit de la situation initiale.

Suivant nos résultats, nous avons choisi de spécialiser notre portefeuille en sélectionnant certains actifs, c'est-à-dire en n'utilisant à la place de la liste d'actifs du SP500 qu'une certaine liste d'actifs qui répondent à un critère pour confirmer ou infirmer des hypothèses.

3.3.2 Principe du stockpicking par intelligence artificielle

Afin de procéder à notre stockpicking pour créer un portefeuille axé optimisation par intelligence artificielle (ici, deep learning), à chaque analyse de 100 portefeuilles nous procédons à une sélection avec le critère mentionné précédemment : le type majoritaire d'optimisation par portefeuille.

En d'autres termes, à la fin de l'optimisation des 100 portefeuilles, nous regardons pour chacun d'entre eux : quelle méthode, entre Markowitz et Deep Learning, permet d'obtenir le meilleur ratio de Sharpe pour les portefeuilles Min Var, Max Ret et Max Util ? Nous notons (entre Markowitz et Deep Learning) quelle méthode était le plus souvent la plus performante, et notons, pour chaque actif de chaque portefeuille, son nom ainsi que la méthode la plus performante.

A titre d'exemple, un portefeuille constitué de TSLA, AAPL et AMZN, dont les 3 types (Min Var, Max Ret et Max Util) ont un ratio de Sharpe supérieur avec la méthode Deep Learning, sont notés dans un tableau à part, avec le suffixe « IA » : TSLA IA, AAPL IA, AMZN IA.

En répétant l'opération pour chaque portefeuille, nous obtenons un tableau qui regroupe les actifs, le type (Markowitz et Deep Learning) et leur occurrence. Si pour un actif donné, il est apparu plus de fois dans des portefeuilles mieux optimisés par Deep Learning (ie son occurrence avec le suffixe « IA » est supérieure à son occurrence avec le suffixe « Traditionnel »), alors nous notons le ticker à part et il sera sélectionné pour servir de liste de base pour la constitution des portefeuilles suivants.

Notre but est donc d'optimiser nos portefeuilles plusieurs fois, en réduisant la liste des actifs au fur et à mesure des optimisations, pour ne garder que des actifs qui sont mieux optimisés par Deep Learning : notre portefeuille stockpické.

Figure 15 : Extrait du tableau de stockpicking intermédiaire IA

Actifs	Traditionnel	IA	Total	Type principal	Rang si IA	Ticker si IA
BBY	2	4	6	IA	1	BBY
LW	2	3	5	IA	2	LW
FLT	1	0	1	Traditionnel		
IPG	2	1	3	Traditionnel		
ED	2	3	5	IA	3	ED
GPN	4	1	5	Traditionnel		
GWW	1	0	1	Traditionnel		
PNR	2	0	2	Traditionnel		
NLOK	7	1	8	Traditionnel		
PNC	2	1	3	Traditionnel		
EQR	5	0	5	Traditionnel		
CPRT	6	1	7	Traditionnel		
CVX	2	1	3	Traditionnel		
TAP	3	1	4	Traditionnel		
EQIX	3	1	4	Traditionnel		

(RICHARD Nicolas, Actifs Spé IA x0, 2021)

3.4 Optimisation des portefeuilles et stockpicking avec intelligence artificielle

Cette partie mettra en valeur l'utilisation de nos données afin d'effectuer un stockpicking pour obtenir un portefeuille axé intelligence artificielle en sélectionnant des actifs qui sont mieux optimisés par Deep Learning. Nous optimisons nos portefeuilles en prenant la période de rendements d'actifs : 01/01/2017 → 01/01/2020.

3.4.1 Premier tirage : SP500, pas de spécialisation

Avant d'entrer dans les résultats des tirages, il convient de discuter les notions de supériorité entre Markowitz et Deep Learning. En effet, un solveur (comme pour Markowitz) va chercher l'optimum local en résolvant une équation fournie, par exemple les poids des actions pour que le portefeuille possède la plus grande fonction d'utilité. Ainsi, les données utilisées pour l'optimisation sont les données historiques de rendement que l'on entre en input. Pour le Deep Learning, la méthode est différente. Par le biais des réseaux de neurones récurrents (des neurones liés entre eux, comme le LSTM : Long – Short Term Memory), une partie du set de données est utilisé en tant que set pour effectuer des prédictions³² sur les prix futurs des actifs³³. La construction du portefeuille est effectuée sur ce set et la partie suivante des données sert de mise en pratique réelle avec les données effectives. Dans notre cas, avec DeepDow, les matrices de rendement et Variance – Covariance sont effectivement apprises et prédites.

³² https://deepdow.readthedocs.io/en/latest/auto_examples/end_to_end/iid.html#sphx-glr-auto-examples-end-to-end-iid-py

³³ https://www.researchgate.net/publication/338475546_Portfolio_Optimization-Based_Stock_Prediction_Using_Long-Short_Term_Memory_Network_in_Quantitative_Trading

Comme mentionné précédemment, nous avons créé 100 portefeuilles selon la liste du SP500 pour ne pas biaiser notre analyse. Une première optimisation nous donne les valeurs suivantes :

Tableau 1 : Répartition de l'optimisation des portefeuilles : non spécialisée

Optimisation principale du pf	Traditionnelle	IA	Égalité
Total des portefeuilles	50	50	0
Minimum Variance	38	62	0
Maximum Return	56	44	0
Maximum Utility	65	35	0

(RICHARD Nicolas, Tableau Spé IA x0, 2021)

On constate une répartition plutôt égale de l'optimisation des portefeuilles, c'est-à-dire 50% de portefeuilles mieux optimisés par Markowitz, 50% mieux optimisés par Deep Learning (pour rappel : Sharpe supérieur). Cela nous indique donc la nécessité de sélectionner des actifs pour obtenir un meilleur portefeuille. On constate également que sur nos 100 portefeuilles, aucun d'entre eux n'ont eu d'égalité (c'est-à-dire ratio de Sharpe égal (6 chiffres après la virgule) sur les optimisations (Min Var, Max Ret, Max Util tous les trois égaux) ou chaque type d'optimisation avait une catégorie différente (Traditionnel + IA + égalité). Il semblerait donc que l'optimisation soit assez tranchée et que la différence entre Markowitz et Deep Learning soit assez notable.

En répétant l'expérience, les résultats varient peu : la répartition des portefeuilles semble assez équitable entre Markowitz (« traditionnel ») et Deep Learning (« IA »). On constate également qu'une partie des actifs qui sont mieux intégrés de manière Markowitz ou Deep Learning, semble le rester en réitérant l'optimisation, ce qui pourrait à première vue nous indiquer que le stockpicking est possible et consistant dans la durée.

A partir de ces résultats et de la liste d'actifs du SP500 (481 actifs), nous créons une liste avec les actifs qui sont le mieux optimisés par deep learning, ce qui nous donne une

nouvelle liste avec 91 actifs prêts pour un deuxième tirage. Ce premier stockpicking nous permet de réduire de plus de 5 fois le nombre d'actifs disponibles.

3.4.2 Deuxième tirage : spécialisation IA x1

Afin de continuer notre processus de stockpicking (pour rappel : sélectionner des actifs qui sont mieux optimisés et intégrés par le Deep Learning), nous sélectionnons la liste de 91 actifs et créons 100 portefeuilles à partir de celle-ci.

Tableau 2 : Répartition de l'optimisation des portefeuilles : spécialisation IA x1

Optimisation principale du pf	Traditionnelle	IA	Égalité
Total des portefeuilles	60	40	0
Minimum Variance	49	51	0
Maximum Return	53	47	0
Maximum Utility	65	35	0

(RICHARD Nicolas, Tableau Spé IA x1, 2021)

On constate une légère baisse des portefeuilles mieux optimisés par IA mais les types sont cette fois-ci plus équilibrés. De nouvelles optimisations font varier ces valeurs mais les actifs semblent être les mêmes, encore une fois le type d'optimisation semble être durable. Nous obtenons cette fois-ci 21 actifs mieux optimisés par le Deep Learning, ce qui sera notre nouvelle liste pour créer les prochains portefeuilles. Cette spécialisation réduit de plus de 4 fois la taille de la liste disponible pour créer nos prochains portefeuilles, et étant donné que nous sommes censés stockpicker un portefeuille de 18 actifs, il nous apparaît clairement que nous atteignons bientôt notre but.

3.4.3 Troisième tirage : spécialisation IA x2

Les 100 portefeuilles créés une troisième fois, à partir cette fois-ci de 21 actifs, nous donne la répartition suivante :

Tableau 3 : Répartition de l'optimisation des portefeuilles : spécialisation IA x2

Optimisation principale du pf	Traditionnelle	IA	Égalité
Total des portefeuilles	34	66	0
Minimum Variance	66	34	0
Maximum Return	27	73	0
Maximum Utility	35	65	0

(RICHARD Nicolas, Tableau Spé IA x2, 2021)

Cette spécialisation semble orienter les valeurs en direction de l'intelligence artificielle, et ce, même en réitérant l'optimisation avec d'autres portefeuilles de la même liste d'actifs.

Les actifs mieux optimisés par deep learning que nous obtenons sont toujours au nombre de 21, et réitérer l'optimisation ne change ni la liste, ni le nombre. Ainsi, nous en déduisons que le stockpicking ne pourra être amélioré en continuant dans ce sens. Nous décidons donc de sélectionner les 18 meilleurs actifs selon leur occurrence dans les portefeuilles mieux optimisés par deep learning. Ainsi, nous obtenons un portefeuille stockpické de 18 actifs par intelligence artificielle :

['TSLA', 'IEX', 'IP', 'NKE', 'FDX', 'WAB', 'FMC', 'ALGN', 'FOXA', 'PHM', 'BSX', 'FTNT', 'D', 'COST', 'FFIV', 'EW', 'MAR', 'TTWO']

3.4.4 Évolution des Sharpe, Rendements et Risques des portefeuilles

Sur cette période 01/01/2017 → 01/01/2020, nous décidons de comparer les ratios de Sharpe, les rendements et les risques de nos portefeuilles au fur et à mesure de l'optimisation / stockpicking. Pour rappel, ces données sont des moyennes journalières.

Afin de simplifier cette partie, car nous avons 2 méthodes d'optimisations sur 3 types de portefeuilles, nous nous intéressons à l'un d'entre eux seulement : l'évolution du portefeuille Maximum Utility optimisé par deep learning.

Pour la composante risque, on remarque que le risque minimal passe de 0.6982% à 0.7698% : l'augmentation de celui-ci pourrait être liée à une recherche de rendement. Le risque moyen passe de 0.8178% à 0.9661%. L'augmentation peut également signifier une recherche de rémunération supplémentaire du risque. Pour les risques maximaux, ils augmentent de 0.9603% à 1.1515 %, une augmentation assez notable.

Dans la catégorie rendement, nous cherchons donc à savoir si cette augmentation de risque s'accompagne d'une augmentation de rendement. C'est bel et bien le cas pour les rendements minimaux, passant de 0.0447% à 0.0728%. En moyenne, les rendements varient entre 0.0828% et 0.1115%. Les rendements maximaux stagnent entre 0.1370% et 0.1381%, et on constate bel et la suite de cette augmentation de risque. Il semblerait donc que notre stockpicking semble induire une augmentation de rendement en prenant plus de risque. Toutefois, le minimum des rendements nous montre que les actifs à rendement plus faible sont mis de côté et que le cœur des actifs varie d'une manière notable.

En ce qui concerne le Sharpe ratio, les ratios minimaux restent toujours négatifs, ce qui n'est pas une excellente nouvelle même si ces derniers s'approchent de 0. Toutefois, en moyenne, ils passent d'une valeur négative (-0.0210) à une valeur positive après stockpicking (0.0114), ce qui semble confirmer la recherche de rendement supplémentaire et une tendance bénéfique. Pour les valeurs maximales, elles diminuent légèrement tout en restant positives (de 0.0421 à 0.0353), ce qui signifierait peut-être l'ajout d'un actif très risqué par rapport à son rendement (comme en témoignerait l'augmentation de +0.15% de risque maximal).

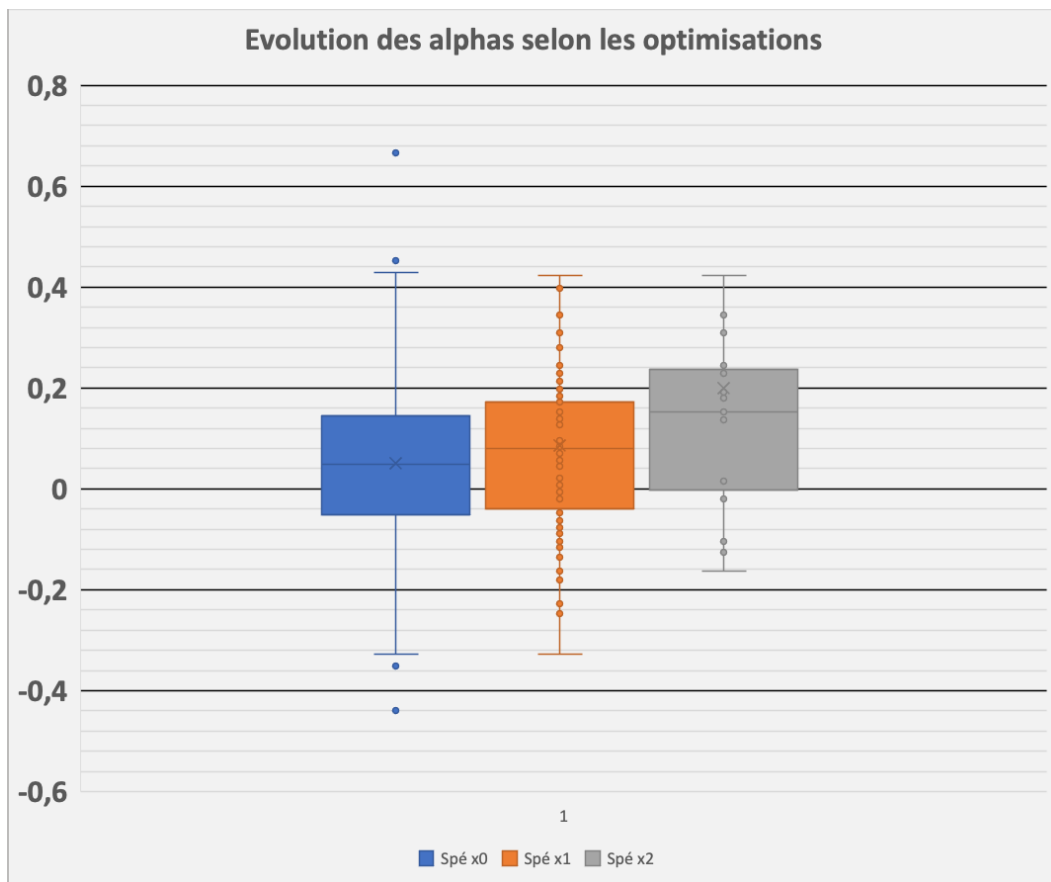
Finalement, pour ce type de portefeuille mais aussi pour les autres, le stockpicking semble avoir un effet bénéfique, et cela porte l'ensemble des portefeuilles dans une tendance haussière, que ce soit au niveau du rendement, du Sharpe mais aussi du risque. Le stockpicking par intelligence artificielle semble donc créer un portefeuille plus performant.

3.4.5 Alphas et Bêtas des portefeuilles

Comme effectué dans la partie précédente, nous comparons les alphas et bêtas au fur et à mesure de nos optimisations pour tenter de comprendre comment le deep learning « choisit » les actifs.

Un boxplot des alphas suivant les optimisations ainsi qu'une liste détaillée des valeurs nous donnent quelques pistes de réflexion :

Figure 16 : Boxplot des alphas selon les optimisations, 2017 - 2020

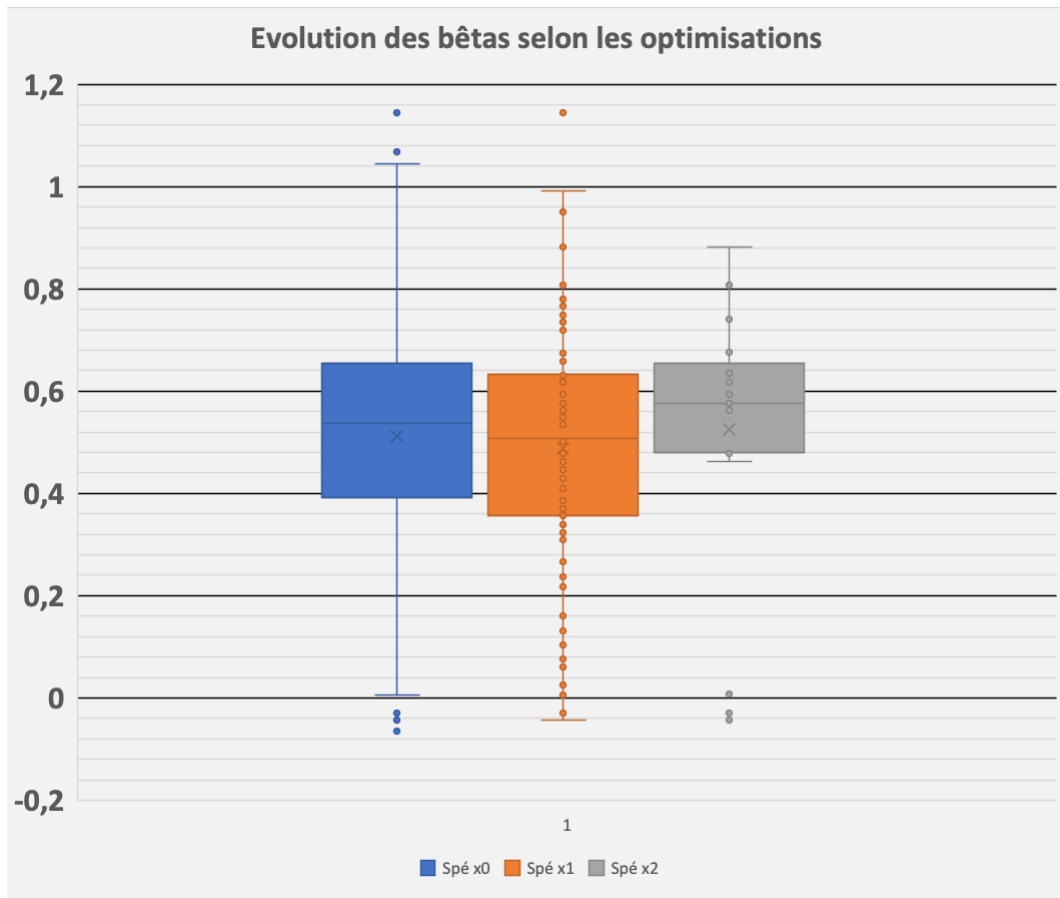


(RICHARD Nicolas, Résumé Alpha + Bêta, 2021)

On remarque une évolution croissante des alphas au fur et à mesure des optimisations. On constate que toutes les valeurs importantes (Q1, Médiane, Q3) augmentent. Il semblerait que le stockpicking lié à l'intelligence artificielle sélectionne des actifs dont l'alpha (surperformance ajustée au risque par rapport à l'indice) est plus élevé que la moyenne, la tendance serait donc à la création de valeur. Sur ce graphique (les valeurs extrêmes ont été masquées pour plus de lisibilité), on constate également que les actifs à fort alpha sont sélectionnés dans les 3 cas. Cela semble confirmer cette recherche de création de valeur. Ces différences semblent significatives (+200% d'augmentation pour la médiane, +60% pour le Q3). Un autre point intéressant est concernant le premier quartile : on remarque qu'au final, ce premier quartile est légèrement supérieur : 75% des valeurs sont donc plus grandes que 0, ce qui semble nous dire que le stockpicking par intelligence artificielle recherche bel et bien une création de valeur et une meilleure performance ajustée au risque.

Pour le bêta, une conclusion se dessine une fois le boxplot dessiné :

Figure 17 : Boxplot des bêtas selon les optimisations, 2017 - 2020



(RICHARD Nicolas, Résumé Alpha + Bêta, 2021)

On remarque, même si l'étape intermédiaire ne semble pas le montrer, que les bêtas se resserrent. La conclusion laisse à penser que l'algorithme sélectionne des actifs à volatilité plus faible que le marché (maximum inférieur à 1, 1^{er} quartile proche de 0.5), et de manière concentrée. Ainsi, il semblerait que le portefeuille stockpické soit assez défensif et moins volatile que le SP500.

Ce stockpicking, en ajoutant les conclusions établies concernant les rendements, risques et Sharpe, semble avoir pour effet la recherche d'un rendement plus élevé en essayant de contenir la volatilité.

3.5 Comparaison du stockpicking Traditionnel vs IA

Nous avons désormais deux portefeuilles stockpickés de manière différente. Le but est désormais de comparer leurs rendements sur une période différente, ici 01/01/2020 → 01/01/2021, pour savoir si l'intelligence artificielle ou une méthode traditionnelle de stockpicking est plus performante, et dans le cadre de la gestion, si l'optimisation est plus performante par Markowitz ou en Deep Learning.

3.5.1 Comparaison des valeurs liées aux portefeuilles

Après avoir mis en input les portefeuilles stockpickés dans notre algorithme DeepDow, nous obtenons des résultats intéressants à discuter.

Pour l'optimisation du portefeuille Minimum Variance, voici le tableau qui résume les valeurs obtenues :

Tableau 4 : Comparaison Stockpick : Minimum Variance

	Risque Markowitz	Rendement Markowitz	Sharpe Markowitz	Risque IA	Rendement IA	Sharpe IA
Stockpick Normal	1.4451 %	0.0094 %	- 0.0624	1.4457 %	0.0076 %	- 0.0636
Stockpick IA	1.6748 %	0.0942 %	- 0.0032	1.6864 %	0.0974 %	- 0.0012

(RICHARD Nicolas, Tableau comparaison Normal vs IA, 2021)

Si l'on compare les portefeuilles stockpickés entre eux, on remarque que dans le cas du stockpick par intelligence artificielle, les risques sont plus élevés mais ils sont également compensés par un rendement plus élevé, que ce soit par l'optimisation Markowitz ou par l'optimisation Deep Learning. Les ratios de Sharpe confirment cette constatation : le portefeuille stockpické par intelligence artificielle possède un meilleur ratio de Sharpe dans les deux cas d'optimisation. On constate également que l'optimisation par Deep Learning est meilleure dans le portefeuille stockpické par IA (- 0.0012 contre - 0.0032) mais pas dans le cas du stockpick normal, où l'optimisation Markowitz est plus efficace (- 0.0624 contre - 0.0636).

Pour l'optimisation du portefeuille Maximum Return, un autre tableau nous permet de faire une comparaison :

Tableau 5 : Comparaison Stockpick : Maximum Return

	Risque Markowitz	Rendement Markowitz	Sharpe Markowitz	Risque IA	Rendement IA	Sharpe IA
Stockpick Normal	2.8853 %	0.1268 %	0.0095	2.8853 %	0.1268 %	0.0095

Stockpick IA	5.6391 %	0.8428 %	0.1318	5.6391 %	0.8428 %	0.1318
-----------------	----------	----------	--------	----------	----------	--------

(RICHARD Nicolas, Tableau comparaison Normal vs IA, 2021)

Lorsque l'on vérifie les valeurs obtenues, on constate qu'à 6 chiffres après la virgule (notre arrondi pour définir la notion de significativité statistique), les valeurs sont identiques. Il faut aller à 8 chiffres pour voir que les Sharpe de Deep Learning sont meilleurs que les Sharpe Markowitz, mais la différence n'est pas statistiquement significative. Ainsi, dans le cas du Maximum Return, pour un portefeuille donné le deep learning ne bat pas Markowitz. Mais pour les portefeuilles entre eux, la différence est significative : 0.1318 contre 0.0095 dans les deux cas : le portefeuille stockpické par IA performe mieux, avec un plus gros risque (quasiment doublé), mais une performance qui s'aligne en conséquence.

Le portefeuille Maximum Utility possède des caractéristiques également intéressantes :

Tableau 6 : Comparaison Stockpick : Maximum Utility

	Risque Markowitz	Rendement Markowitz	Sharpe Markowitz	Risque IA	Rendement IA	Sharpe IA
Stockpick Normal	1.4682 %	0.0229 %	- 0.0522	1.5210 %	0.0321 %	- 0.0443
Stockpick IA	1.8355 %	0.1876 %	0.0480	1.8760 %	0.1979 %	0.0524

(RICHARD Nicolas, Tableau comparaison Normal vs IA, 2021)

Nous constatons deux points : le premier point est que, dans les deux portefeuilles, l'optimisation par deep learning performe mieux qu'avec Markowitz. Le deuxième point notable est que le portefeuille stockpické par intelligence artificielle surperforme dans les deux optimisations le stockpicking normal, et de loin.

Finalement, dans les 3 cas d'optimisation selon les types (min var, max ret, max util), le portefeuille stockpické par intelligence artificielle bat le stockpicking normal en termes de Sharpe ratio. Pour le type d'optimisation, dans 3 cas sur 6 le deep learning bat Markowitz, dans 2 cas sur 6 il y a égalité et dans 1 cas sur 6, Markowitz bat le deep learning. On constate un léger avantage pour le Deep Learning. Il semblerait également que le

portefeuille stockpické avec intelligence artificielle soit meilleur en termes de rendements ajustés au risque.

Après ces constatations, il nous vient une question en tête : comment expliquer de si grandes différences ?

Nous décidons de créer deux tableaux Excel avec les rendements totaux des actifs de nos portefeuilles stockpickés sur la période concernée. La différence élevée de performance s'explique à la vue d'un seul rendement : Tesla. En effet, entre début 2020 et début 2021, l'action TSLA a fourni une performance de + 743%. Pour le cas d'un portefeuille Maximum Return, il suffit juste de sélectionner cet actif et il est quasiment impossible de rivaliser. Cela nous prouve également que le stockpicking par IA fut judicieux car il a inclus TSLA dans le portefeuille bien avant le rallye (pour rappel, le prix de TSLA au 31 décembre 2019 était de 83.67\$). Mais cela n'explique pas tout. En effet, si l'on calcule le rendement pour les deux portefeuilles de manière équipondérée (poids d'un actif = 1/18), alors on obtient - 11.63% pour le stockpick normal contre + 61.49% avec le stockpick par IA. Le choix des actifs semble être bien plus judicieux, et certes, le choix de Tesla aide en partie. Mais même lorsque l'on retire TSLA du portefeuille, la performance obtenue reste de + 21.37%, ce qui nous indique un bon choix d'actifs. A titre de comparaison, la performance du SP500 entre janvier 2020 et janvier 2021 est de 16.26%³⁴.

Figure 18 : Résumé de performance : stockpick Normal vs IA, équipondéré

STOCKPICK NORMAL	CE	SEE	NEM	ALL	UNM	AFL	KIM	BXP	EQR	INTC	WU	HPQ	DISH	OMC	VZ	OKE	COG	KMI
31/12/2019	123,12	39,83	43,45	112,45	29,16	52,9	20,71	137,86	80,92	59,85	26,78	20,55	35,47	81,02	61,4	75,67	17,41	21,17
31/12/2020	129,94	45,79	59,89	109,93	22,94	44,47	15,01	94,53	59,28	49,82	21,94	24,59	32,34	62,37	58,75	38,38	16,28	13,67
Rdt Total	5,54%	14,96%	37,84%	-2,24%	-21,33%	-15,94%	-27,52%	-31,43%	-26,74%	-16,76%	-18,07%	19,66%	-8,82%	-23,02%	-4,32%	-49,28%	-6,49%	-35,43%
Poids Equipondéré	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%
R Tot Equipondéré	-11,63%																	
R Tot 1 actif partout	-13,45%																	
STOCKPICK IA	TSLA	IEK	IP	NKE	FDX	WAB	FMC	ALGN	FOXA	PHM	BSX	FTNT	D	COST	FFIV	EW	MAR	TTWO
31/12/2019	83,666	172	46,05	101,31	151,21	77,8	99,82	279,04	37,07	38,8	45,22	106,76	82,82	293,92	139,65	77,76334	151,43	122,43
31/12/2020	705,67	199,2	49,72	141,47	259,62	73,2	114,93	534,38	29,12	43,12	35,95	148,53	75,2	376,78	175,94	91,23	131,92	207,79
Rdt Total	743,44%	15,81%	7,97%	39,64%	71,69%	-5,91%	15,14%	91,51%	-21,45%	11,13%	-20,50%	39,13%	-9,20%	28,19%	25,99%	17,32%	-12,88%	69,72%
Poids Equipondéré	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%	5,56%
R Tot Equipondéré	61,49%																	
R Tot 1 actif partout	61,09%																	

(RICHARD Nicolas, Rendements Stockpick Normal vs IA, 2021)

Pour résumer cette partie, nous pouvons créer un tableau avec les valeurs annualisées de chaque portefeuille et de chaque optimisation par l'algorithme :

³⁴ <https://www.macrotrends.net/2526/sp-500-historical-annual-returns>

Tableau 7 : Résumé de performance : stockpick Normal vs IA, optimiseur, annualisé

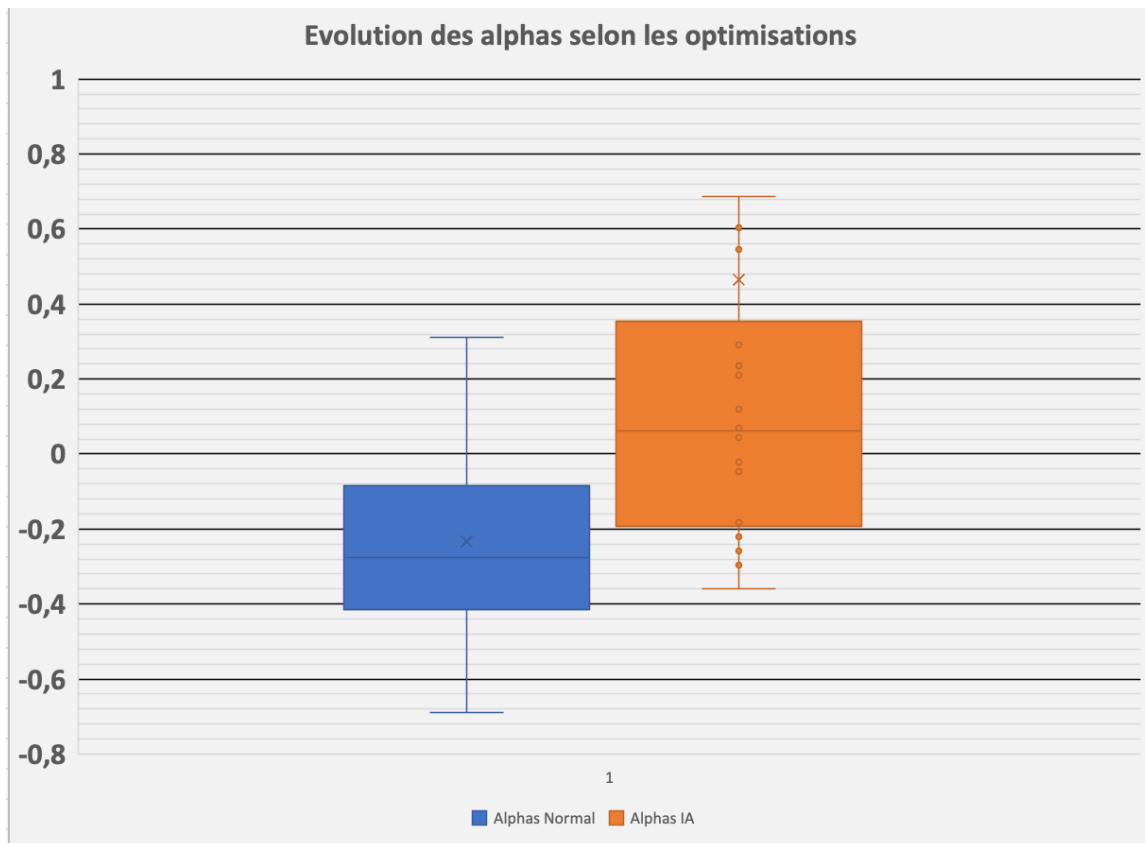
	Markowitz : (risque ; rendement)			Deep Learning : (risque ; rendement)		
DONNÉES ANNUALISÉES	<i>Minimum Variance</i>	<i>Maximum Return</i>	<i>Maximum Utility</i>	<i>Minimum Variance</i>	<i>Maximum Return</i>	<i>Maximum Utility</i>
Stockpick Normal	(22,94% ; 2,4%)	(45,8% ; 37,63%)	(23,31% ; 5,93%)	(22,95% ; 1,93%)	(45,8% ; 37,63%)	(24,15% ; 8,43%)
Stockpick IA	(26,59% ; 26,77%)	(89,52% ; 728,95%)	(29,14% ; 60,37%)	(26,77% ; 27,82%)	(89,52% ; 728,95%)	(29,78% ; 64,58%)

(RICHARD Nicolas, Résumé Perf Stockpick, 2021)

3.5.2 Comparaison des alphas et bêtas

Il est maintenant temps de constater l'évolution des alphas et bêtas à la suite de notre stockpicking, pour la période 2020 – 2021.

Figure 19 : Boxplot des alphas selon les optimisations, 2020 - 2021

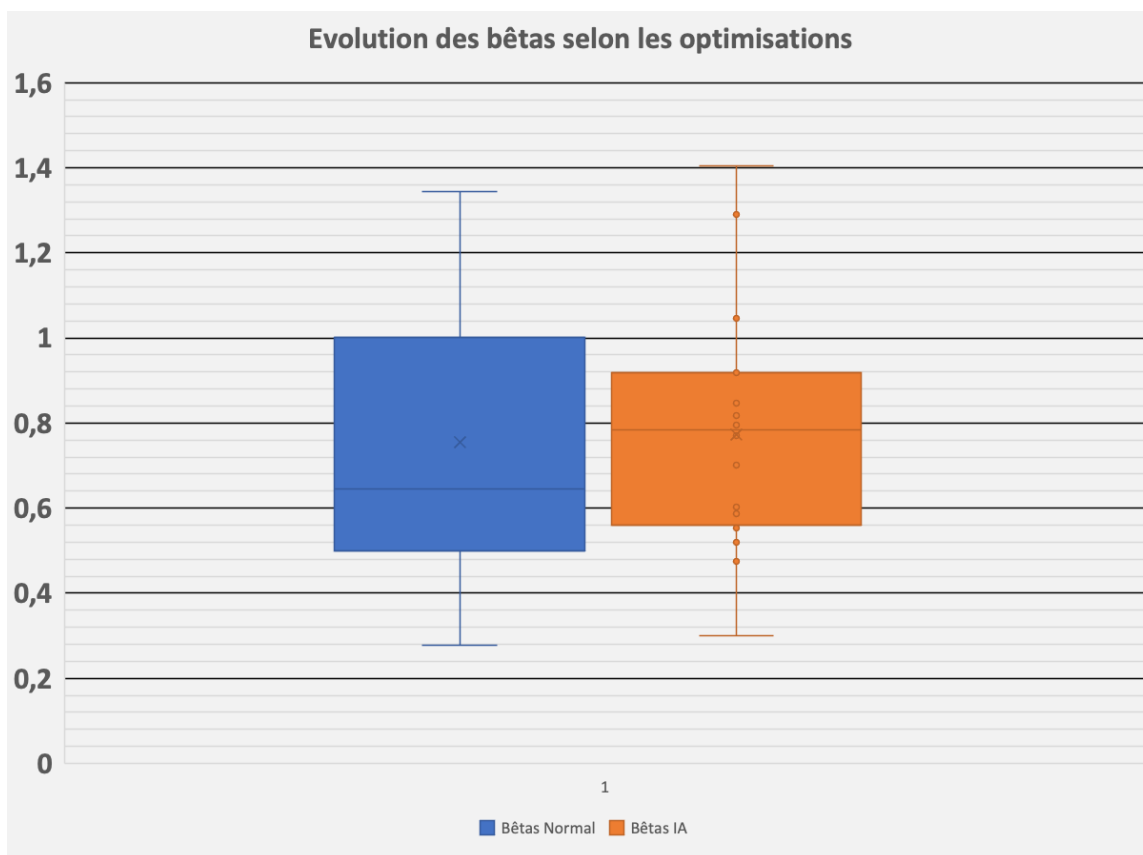


(RICHARD Nicolas, Résumé Alpha Bêta Stockpick, 2021)

On constate de manière flagrante, que pour 2020 – 2021, le portefeuille de stockpicking selon l'intelligence artificielle a su prédire des actifs avec un alpha plus élevé. Ils étaient, entre 2017 et 2020, également élevés : la conservation d'alpha et de valeur est

effective. Le premier quartile est plus faible mais le troisième est plus élevé, il semblerait donc que l'intervalle de ces alphas ne soit plus aussi resserré mais les valeurs semblent très correctes.

Figure 20 : Boxplot des bêtas selon les optimisations, 2020 - 2021



(RICHARD Nicolas, Résumé Alpha Bêta Stockpick, 2021)

Pour les bêtas, ils semblent toujours aussi resserrés avec toutefois une légère augmentation : il semblerait que les actifs aient gagné en volatilité, mais cela est certainement dû au marché tumultueux de 2020, notamment lié à la crise du COVID.

3.6 Synthèse des résultats

A la lumière de ces analyses, il est intéressant de noter les diverses constatations et remarques afin de se faire une idée plus éclairée du sujet. Bien sûr, ces constatations et recommandations se basent sur l'analyse des données présentées ci-dessus et dépendent de la qualité de ces dernières.

3.6.1 Constatations liées aux données

Aux vues des optimisations effectuées (dans l'optique de stockpicking) par notre algorithme, nous avons constaté que la part de portefeuilles mieux optimisés par Markowitz ou Deep Learning semble être égale et constante, de l'ordre de 50%/50%.

Ainsi, il semble difficile d'affirmer que le Deep Learning optimise mieux les portefeuilles que Markowitz, et inversement. Bien sûr, une fois un début de stockpicking mis en place, les valeurs varient mais cela est normal car nous sélectionnons des actifs qui sont mieux gérés par un certain type d'optimisation. Mais en temps normal, sans sélection, aucun type (ie Markowitz ou Deep Learning) ne sort du lot. De ce fait, nous ne pouvons pas affirmer, sur la base de nos données analysées, que certains portefeuilles en « hold only » sur la base du SP500 peuvent être mieux optimisés par l'intelligence artificielle ou par un optimiseur de type Markowitz.

En ce qui concerne le stockpicking, le portefeuille stockpické par intelligence artificielle sort clairement du lot et propose de meilleurs rendements et ratios de Sharpe. Même si la présence de Tesla dans ce portefeuille biaise très certainement les valeurs, la suppression de cette dernière du portefeuille n'exerce pas une influence extrême : les valeurs de rendement restent très positives.

On constate que l'algorithme de stockpicking par intelligence artificielle a eu un pouvoir prédictif assez bon (attention toutefois au biais de l'année 2020, exceptionnelle) par rapport au stockpicking normal.

3.6.2 Recommandations

Il nous est possible d'émettre des recommandations qui vont être assez conservatrices.

Nous recommandons la mise en place d'un algorithme d'optimisation avec intelligence artificielle afin de créer des portefeuilles stockpickés et de les backtester avec des données passées. Nous ne conseillons pas de l'utiliser afin d'investir directement, mais plutôt dans l'optique de vérifier si l'intelligence artificielle peut mieux performer qu'une sélection d'actifs traditionnelle sur des périodes passées, ou même pour tester le comportement en période de crise. Il faudrait bien évidemment le comparer avec un algorithme de stockpicking traditionnel afin de constituer des comparaisons documentées.

Nous ne recommandons toutefois pas la sélection pure d'intelligence artificielle pour optimiser des portefeuilles, car nos analyses ne nous ont clairement pas permis d'affirmer que ce style est supérieur aux autres. Il serait néanmoins possible de l'utiliser comme base de comparaison.

3.6.3 Points d'amélioration

Ces constatations peuvent être biaisées par l'année 2020, à son rallye et bull market très forts qui ont donné aux investisseurs l'occasion de réaliser de très jolis rendements malgré un début d'année assez chaotique.

Afin de vérifier la validité de nos conclusions, il serait tout d'abord possible de comparer ces données avec des algorithmes ou modèles existants. Il semblerait que beaucoup de Hedge Funds utilisent notamment l'intelligence artificielle avec des algorithmes et modèles. Attention toutefois, car le monde du Hedge Fund est opaque et mettre la main sur un outil de ce genre ne sera pas facile.

Nous pourrions également, avec l'aide d'une personne ayant des connaissances en intelligence artificielle, programmer un algorithme qui effectuerait cette optimisation en utilisant le deep learning et/ou une autre méthode d'IA, afin de savoir si les constatations dépendent de la méthode d'IA appliquée ou du code en lui-même.

Une possibilité liée aux données serait d'agrandir notre champ de calculs en ne se basant plus seulement sur le SP500, mais également sur d'autres listes d'actifs, tout en augmentant le nombre de portefeuilles et/ou leur taille : FTSE, CAC, MSCI World... Ainsi, cela permettrait de savoir si les constatations dépendent d'un facteur « région » ou pas.

La base de comparaison pourrait être également changée : nous nous basons sur le ratio de Sharpe, car les données en input de l'algorithme sont des données moyennes. En se basant sur les données complètes (ce qui fera ralentir l'algorithme), nous aurons la possibilité de changer de ratio et d'obtenir une comparaison plus forte avec plus de significativité (ratio de Sortino, ratio d'Information par exemple).

Toujours sur cette lancée, en plus de la comparaison d'alphas, nous pouvons également intégrer d'autres données liées aux actifs, comme des données internes (ratios du style P/E par exemple, ou P/B), secteur, ou même des controverses. Cela permettrait d'identifier (ou non) d'éventuels facteurs influençant l'actif et donc l'optimisation par l'algorithme.

Finalement, nos tirages pourraient être améliorés et complétés par deux méthodes. La première s'apparente à une distribution d'échantillonnage. En effet, pour un portefeuille spécialisé, nous avons la possibilité d'adapter le code et d'effectuer, par exemple, 20 tirages de ce tirage. Ainsi, nous aurons 20 groupes de 100 portefeuilles, ce qui ressemble à une distribution de plusieurs échantillons afin de calculer les valeurs de ces derniers (moyenne, variance entre autres) pour en déduire des caractéristiques théoriques sur l'échantillon de base. Cela nous permettrait de savoir dans quels intervalles varient nos taux d'optimisation IA et bien d'autres valeurs, afin de mieux cerner les variations et savoir si une part de hasard ne reste pas liée. Cela impliquerait un temps de calcul très long (30 minutes pour 100 portefeuilles * 20 groupes = 10 heures de calcul non-stop). La deuxième méthode pourrait être d'utiliser un solveur traditionnel ou cet optimiseur traditionnel / deep

learning afin d'optimiser nos groupes de 100 portefeuilles. Ainsi, il serait intéressant de voir s'il existe une optimisation / spécialisation possible depuis cet algorithme et si la spécialisation qu'il effectuerait est en rapport avec la nôtre. Comme pour la première méthode, cela serait également très long en termes de calculs.

4. Conclusion

De notre revue des articles et autres témoignages, il en ressort une forte tendance de digitalisation (flux de données multiplié par 3 en 2025). Beaucoup d'acteurs semblent intéressés et considèrent des solutions d'intelligence artificielle. Avec cette digitalisation et cette tendance croissante, il serait intéressant pour eux de sauter le pas s'ils ont en effet bien saisi les enjeux et les risques, et qu'ils comprennent ce qu'ils s'appêtent à mettre en place. En pratique, on constate toutefois que cette intégration n'est pas aussi complète qu'on le croit et le chemin reste long, même si les opportunités sont actuellement grandes.

La partie pratique nous ouvre les yeux sur le fonctionnement en situation du deep learning. On peut utiliser cet algorithme afin d'effectuer une sélection d'actifs, et bien que cela ne constitue pas une source infaillible, cette partie pratique nous montre bien des points. Dans notre cas, sans spécification et sans consignes, il n'y a pas de vraie distinction entre une optimisation standard (Markowitz) ou avec intelligence artificielle.

Cet exercice de recherche et de pratique fut une expérience très enrichissante : elle nous ouvre les yeux sur une application directe et réelle de plusieurs algorithmes et elle permet de saisir les enjeux futurs alors que nous serons potentiellement diplômés dès cette année. En plus d'enrichir ma culture personnelle, cela me sert d'introduction à un thème d'avenir qui dépasse le cadre de la finance et qui sera probablement un enjeu sociétal des années à venir.

Bibliographie

ACCENTURE, [sans date]. *AI in Wealth Management: Built to Scale* | Accenture [en ligne]. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://www.accenture.com/us-en/insights/capital-markets/wealth-management-artificial-intelligence>

ASSOCIATION SUISSE DES BANQUIERS (ASB) et AERNI, Andrea Luca, [sans date]. The future of the financial industry: banking on data. *Swiss Bankers Association* [en ligne]. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://www.swissbanking.ch/en/news-and-positions/opinions/the-future-of-the-financial-industry-banking-on-data>

BEATTIE, rew, [sans date]. Understanding The History Of The Modern Portfolio. *Investopedia* [en ligne]. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://www.investopedia.com/articles/07/portfolio-history.asp>

BODNAR, Taras, OKHRIN, Yarema, VITLINSKY, Valdemar et ZABOLOTSKY, Taras, 2018. Determination and estimation of risk aversion coefficients. *Computational Management Science*. 1 juin 2018. Vol. 15, n° 2, pp. 297-317. DOI [10.1007/s10287-018-0317-x](https://doi.org/10.1007/s10287-018-0317-x).

BUCHANAN, Bonnie G, 2019. *Artificial intelligence in finance* [en ligne]. Zenodo. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://zenodo.org/record/2626454>

BUSCH, Christian, 1 apr. J.-C. *Intelligence artificielle : comment réagit la Confédération ?* [en ligne]. 30 novembre 1 apr. J.-C. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://dievolkswirtschaft.ch/fr/2020/03/busch-12-2019fr/>

COBBAUT, Robert, 2015. *La gestion de portefeuille: instruments, stratégie et performance*. Bruxelles : De Boeck. ISBN 978-2-8041-9012-5.

CONFÉDÉRATION SUISSE, 1992. RS 235.1 - *Loi fédérale du 19 juin 1992 sur la protection des données (LPD)* [en ligne]. [Consulté le 4 juillet 2021 a]. Disponible à l'adresse : https://www.fedlex.admin.ch/eli/cc/1993/1945_1945_1945/fr

CONFÉDÉRATION SUISSE, 2015. RS 958.11 - *Ordonnance du 25 novembre 2015 sur les infrastructures des marchés financiers et le comportement sur le marché en matière de négociation de valeurs mobilières et de dérivés (Ordonnance sur l'infrastructure des marchés financiers, OIMF)* [en ligne]. [Consulté le 4 juillet 2021 b]. Disponible à l'adresse : <https://www.fedlex.admin.ch/eli/cc/2015/854/fr>

DEEPAI.ORG, 2019. Hidden Layer. [en ligne]. 17 mai 2019. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://deepai.org/machine-learning-glossary-and-terms/hidden-layer-machine-learning>

FAUSCH, Jürg et ANKENBRAND, Thomas, [sans date]. Swiss Asset Management Study 2020. . pp. 126.

FINTECHNEWS SWITZERLAND, 2018. Swiss Banks Accelerate AI Adoption. *Fintech Schweiz Digital Finance News - FintechNewsCH* [en ligne]. 13 mars 2018. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://fintechnews.ch/fintech/swiss-banks-accelerate-ai-adoption/16329/>

FINTECHNEWS SWITZERLAND, 2020. 3 Key AI Capabilities for Asset Management Applications. *Fintech Schweiz Digital Finance News - FintechNewsCH* [en ligne]. 16 octobre 2020. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://fintechnews.ch/aifintech/3-key-ai-capabilities-for-asset-management-applications/39904/>

GDS, Direction opérationnelle Suisse numérique, [sans date]. *Stratégie Suisse numérique - Données, contenus numériques et intelligence artificielle* [en ligne]. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://www.digitaldialog.swiss/fr/champs-d-action/donnees-contenus-numeriques-et-intelligence-artificielle>

GORDON, Clara-Ann et GUROVITS, Andras, [sans date]. AI, Machine Learning & Big Data Laws and Regulations | Switzerland | GLI. [en ligne]. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://www.globallegalinsights.com/practice-areas/ai-machine-learning-and-big-data-laws-and-regulations/switzerland>

GRANDVIEW RESEARCH, [sans date]. *AI In Asset Management Market Size Report, 2020-2027* [en ligne]. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://www.grandviewresearch.com/industry-analysis/artificial-intelligence-asset-management-market>

JOLLIEN, Nathalie, 2019. Artificial Intelligence in Finance. [en ligne]. 14 novembre 2019. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://actu.epfl.ch/news/artificial-intelligence-in-finance-3/>

KREPL, Jan, 2021. *DeepDow* [en ligne]. Python. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://github.com/jankrepl/deepdow>

KREPL, Jan, 2021. *DeepDow* Independent and identically distributed returns [en ligne]. Python. [Consulté le 4 juillet 2021]. Disponible à l'adresse : https://deepdow.readthedocs.io/en/latest/auto_examples/end_to_end/iid.html#sphx-glr-auto-examples-end-to-end-iid-py

MARKOWITZ, Harry, 1952. Portfolio Selection. *The Journal of Finance*. 1952. Vol. 7, n° 1, pp. 77-91. DOI [10.2307/2975974](https://doi.org/10.2307/2975974).

MCKINSEY, [sans date]. *Global survey: The state of AI in 2020* | McKinsey [en ligne]. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/global-survey-the-state-of-ai-in-2020#>

MILTHALER, Frank, [sans date]. *FinQuant: A program for financial portfolio management, analysis and optimisation* [en ligne]. Python. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://github.com/fmilthaler/FinQuant>

NEVES, Fábio, 2018. Plotting Markowitz Efficient Frontier with Python. *Medium* [en ligne]. 14 octobre 2018. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://towardsdatascience.com/python-markowitz-optimization-b5e1623060f5>

ORBIUM et ACCENTURE, [sans date]. *Orbium Wealth Management C-Level Survey* [en ligne]. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://www.orbium.com/c-level-survey/>

PRICEWATERHOUSECOOPERS, [sans date]. *Asset & Wealth Management* [en ligne]. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://www.pwc.com/us/en/tech-effect/ai-analytics/ai-predictions/asset-and-wealth-management.html>

PWC, [sans date]. *How mature is AI adoption in financial services?* [en ligne]. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://www.pwc.de/de/future-of-finance/how-mature-is-ai-adoption-in-financial-services.pdf>

REFINITIV, 2021. *Using AI and big data in asset management* [en ligne]. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://www.refinitiv.com/perspectives/market-insights/how-ai-and-big-data-are-reshaping-asset-management/>

ROLANDI, Alex, [sans date]. *Machine-driven investing: the future of portfolio management?* [en ligne]. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://www.funds-europe.com/news/machine-driven-investing-the-future-of-portfolio-management>

RUSSELL, Stuart J., NORVIG, Peter et DAVIS, Ernest, 2016. *Artificial intelligence: a modern approach*. 3rd ed. Upper Saddle River : Prentice Hall. Prentice Hall series in artificial intelligence. ISBN 978-0-13-604259-4.

Q335 .R86 2010

SHORELINE, 2018. *Survey - Use of AI in Asset Management 2018*. *Shoreline* [en ligne]. 20 mai 2018. [Consulté le 4 juillet 2021]. Disponible à l'adresse : <https://shorelineawc.com/use-of-ai-in-asset-management/>

SNOPEK, Lukasz, 2012. *The Complete Guide to Portfolio Construction and Management* [en ligne]. Wiley. [Consulté le 4 juillet 2021]. ISBN 978-1-119-97688-2. Disponible à

l'adresse :

<http://gen.lib.rus.ec/book/index.php?md5=332e75d99e9f78070166cdf70b171b63>

SWAINE-SIMON, Sydney, 2018. The History of AI in Finance. *Medium* [en ligne]. 8 mars 2018. [Consulté le 4 juillet 2021]. Disponible à l'adresse :

<https://medium.com/district3/the-history-of-ai-in-finance-7a03fcb4a498>

TALEB, Nassim Nicholas, 2007. *The Black Swan: The Impact of the Highly Improbable* [en ligne]. 1. [Consulté le 4 juillet 2021]. ISBN 978-1-4000-6351-2. Disponible à l'adresse :

<http://gen.lib.rus.ec/book/index.php?md5=799cdaf29ec65e01fe4191219f3deaf0>

TURING, A. M., 1950. I.—COMPUTING MACHINERY AND INTELLIGENCE. *Mind*. 1 octobre 1950. Vol. LIX, n° 236, pp. 433-460. DOI [10.1093/mind/LIX.236.433](https://doi.org/10.1093/mind/LIX.236.433).

WILLIAMS-GRUT, Oscar, [sans date]. Banks are looking to use artificial intelligence in almost every part of their business: Here's how it can boost profits. *Business Insider* [en ligne]. [Consulté le 4 juillet 2021]. Disponible à l'adresse :

<https://www.businessinsider.com/ai-in-financial-services-2017-11>

Annexe 1 : Codes PYTHON d'IA DeepDow et de mise en forme

Algorithme de Deep Learning :

La première classe, Markowitz, comporte trois fonctions : minvar, maxret et maxutil. Ces trois fonctions calculent les poids selon le type d'optimisation souhaité. Ainsi, le minvar cherche à minimiser le risque et le maxutil cherche à maximiser (rendement – gamma * risque). On définit ensuite les deux types de contraintes : unconstrained (poids max = 1) et constrained (poids max < 1).

Puis est définie la fonction qui va créer les graphiques, que nous avons adapté afin de pouvoir enregistrer les données calculées pour l'export.

Ensuite, la classe « Net » est définie : il s'agit des réseaux d'apprentissages des matrices de rendement et de Var-CoVar. S'en suivent les diverses boucles qui constituent l'apprentissage et la prédiction des données.

Algorithme de création de portefeuilles :

Nous construisons deux listes vides : une qui inclura nos 18 actifs, et une autre qui inclura nos 100 portefeuilles. Ensuite, nous créons une boucle « while », qui opérera tant que la taille de la liste de portefeuilles (ie. Le nombre de portefeuilles) sera inférieure à 100. Une boucle « for » se situe à l'intérieur de notre « while », qui opère du 1^{er} au 18^{ème} actif, et qui :

- Sélectionne un nombre au hasard compris entre 0 (en Python, les valeurs se situe au rang « n – 1 ») et le nombre d'actifs du SP500 corrigés – 1
- Tant que le nombre au hasard (qui correspond donc à un actif parmi la liste) nous donne un actif déjà présent dans le portefeuille, répéter le choix au hasard.
- Ajouter l'actif correspondant au rang au hasard choisi, dans le portefeuille.
- Et ce, 18 fois pour 18 actifs.

A titre d'information, nous avons lu sur la page du développeur de la bibliothèque « random » qu'il existe deux types de tirage aléatoire : un tirage aléatoire pur, qui peut servir pour de la sécurité ou de la cryptographie, et un autre tirage aléatoire pour de l'analyse de données mais moins sécurisé. Dans notre cas, la deuxième option suffit amplement et c'est donc le module que nous avons utilisé. Ensuite, à la fin de cette boucle « for », quand notre portefeuille est prêt, nous vérifions qu'il n'existe pas déjà et s'il existe,

alors il n'est pas ajouté. Cette boucle « while » permet de s'assurer qu'il y a bien 100 portefeuilles conçus. Pour finir, cette liste de portefeuilles est enregistrée dans un fichier « json » pour les récupérer plus tard. Nous avons également conçu ce code nous-même afin de l'adapter simplement lors de nouvelles itérations et pour pouvoir le modifier en cas de besoin, si nous souhaitons modifier les tailles de portefeuilles ou leur nombre.

Algorithme de boucle et mise en forme :

Maintenant que nos données de rendement et nos portefeuilles sont prêts, il faut les intégrer à notre algorithme de deep learning. Le but est de créer une boucle qui, pour les 100 portefeuilles, les optimisera et inscrira les données dans un tableau de résumé, et comparera selon le ratio de Sharpe.

De ce fait, nous commençons par déclarer deux variables qui contiennent le numéro de portefeuille de début et de fin. Ainsi, nous pouvons utiliser l'algorithme pour un nombre de portefeuilles différent de 100 mais aussi pour découper le calcul en tranches afin de libérer le processeur (à titre d'information, notre processeur 8 cœurs effectue les calculs de 50 portefeuilles en environ 16 minutes à une température moyenne de 76° C).

Ensuite, nous importons les données log-return pré-calculées et créons le tableau de résumé si celui-ci n'existe pas encore, n'est pas encore enregistré ou que le portefeuille de départ de la boucle est le premier. Après avoir importé les portefeuilles selon le fichier « json » créé précédemment, nous déclarons la boucle qui va passer de portefeuille en portefeuille. Une autre boucle va sélectionner les données nécessaires pour créer un tableau de log-return avec les actifs du portefeuille puis prendre la moyenne des log-returns pour obtenir le rendement moyen journalier (ce qui est permis avec le log-return). On fait de même pour obtenir la matrice Var-CoVar et la suite du programme est notre algorithme DeepDow adapté.

Une fois les portefeuilles optimisés, on inscrit les valeurs dans notre tableau résumé : actifs du portefeuille, rendement minvar, risque minvar, sharpe minvar, etc... Finalement, on regarde pour les trois types (minvar, maxret, maxutil) si l'IA ou le solveur simple génère un meilleur Sharpe, et on en déduit quelle méthode a été plus efficace pour ce portefeuille.

Une fois notre boucle principale terminée (que ce soit avec 10, 20 ou plus de portefeuilles), le tableau est enregistré en format « csv » pour analyse sur Excel.

Afin de faciliter notre analyse, nous avons créé un autre algorithme qui convertit ce fichier « csv » en format Excel « xlsx ».

Nous avons nous-même créé un autre algorithme afin de calculer l'alpha et le bêta des actions de chaque portefeuille et de noter quel était le type de chaque optimisation pour chaque actif. Étant donné que nos actifs sont tirés du SP500, nous calculerons nos bêtas et alphas selon le SP500, ainsi nous téléchargeons via Python les données du tracker « SPY ».

```

#### Charger les actifs du SP500
import pandas as pd
import json
from random import randrange
import caffeine

typ = "IA x4"
#sp500_total = "IA x1"
if typ == "sp500_total":
    tbl = pd.read_csv('Tableau SP500 GICS avec PE.csv')
    liste = tbl['Symbol'].tolist()
    liste.remove('BRK.B')
    liste.remove('BF.B')
    liste.remove('STX')
    liste.remove('TDY')
    liste.remove('CARR')
    liste.remove('LUMN')
    liste.remove('OTIS')
    liste.remove('VAR')
    liste.remove('VNT')
    len_liste = len(liste)
elif typ == "IA x1":
    tbl = pd.read_excel('Comparaison NORMAL vs DEEP/Actifs sp6 IA x0.xlsx', sheet_name='Stockpick')
    liste = tbl['Liste stockpickée'].tolist()
    len_liste = len(liste)
elif typ == "IA x2":
    tbl = pd.read_excel('Comparaison NORMAL vs DEEP/Actifs sp6 IA x1.xlsx', sheet_name='Stockpick')
    liste = tbl['Liste stockpickée'].tolist()
    len_liste = len(liste)
elif typ == "IA x3":
    tbl = pd.read_excel('Comparaison NORMAL vs DEEP/Actifs sp6 IA x2.xlsx', sheet_name='Stockpick')
    liste = tbl['Liste stockpickée'].tolist()
    len_liste = len(liste)
elif typ == "IA x4":
    tbl = pd.read_excel('Comparaison NORMAL vs DEEP/Actifs sp6 IA x3.xlsx', sheet_name='Stockpick')
    liste = tbl['Liste stockpickée'].tolist()
    len_liste = len(liste)

liste_actifs = []
liste_pfs = []

print(liste)

while len(liste_pfs) < 100:
    for nb_actif in range(18):
        alea = randrange(len_liste)
        while liste[alea] in liste_actifs:
            alea = randrange(len_liste)
        liste_actifs.append(liste[alea])
    if not liste_actifs in liste_pfs:
        liste_pfs.append(liste_actifs)
        liste_actifs = []

with open('TEMPTEST/liste_pfs.json', 'w') as f:
    json.dump(liste_pfs, f, indent=2)

```

```

#### Boucle principale : for nb_pf ...
import matplotlib inline
import cvxpy as cp
from deepdow.benchmarks import Benchmark
from deepdow.callbacks import EarlyStoppingCallback
from deepdow.data import InRAMDataset, RigidDataLoader
from deepdow.layers import NumericalMarkowitz
from deepdow.losses import MeanReturns, StandardDeviation
from deepdow.experiments import Run
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import torch
import math
import json
import caffeine

debut_pf = 0 #départ = 0. Nb pf = début + 1. 0 ==> pf 1
fin_pf = 100 #dernier pf = 100 ; dans boucle il s'arrête à fin_pf - 1. Debut = 0 ==> fin 100 ==> 0 à 9 mais pf 1 à 10
liste_pfs = []
liste_colonnes = ["Num pf", "Actifs",
                  "Risk Min var true", "Risk Max return true", "Risk Max util true",
                  "Return Min var true", "Return Max return true", "Return Max util true",
                  "Sharpe minvar true", "Sharpe maxret true", "Sharpe maxutil true",
                  "Risk Min var deep", "Risk Max return deep", "Risk Max util deep",
                  "Return Min var deep", "Return Max return deep", "Return Max util deep",
                  "Sharpe minvar deep", "Sharpe maxret deep", "Sharpe maxutil deep",
                  "Better pf minvar", "Better pf maxret", "Better pf maxutil",
                  "Better pfs", "Nb Better pfs"]

log_ret = pd.read_csv('TEMPTEST/logreturns-SP.csv')

if debut_pf == 0 or pd.read_csv('Comparaison NORMAL vs DEEP/Tableau sp6 IA x4.csv') is None:
    tableau_comparaison = pd.DataFrame("", index = list(range(0, 100)), columns = liste_colonnes)
else:
    tableau_comparaison = pd.read_csv('Comparaison NORMAL vs DEEP/Tableau sp6 IA x4.csv')

with open('TEMPTEST/liste_pfs.json', 'r') as f:
    liste_pfs = json.load(f)

for nb_pf in range(debut_pf, fin_pf):
    actifs = liste_pfs[nb_pf]
    log_returns = pd.DataFrame()

    for nb_actif in range(len(liste_pfs[nb_pf])):
        log_returns[actifs[nb_actif]] = log_ret[actifs[nb_actif]]

    VarCoVar = log_returns.cov()
    log_returns = log_returns.T
    dailly = pd.DataFrame(index=log_returns.index)
    dailly['Daily Return'] = log_returns.mean(axis=1)

    torch.manual_seed(4)
    np.random.seed(21)
    #####
    mean = dailly.copy()
    covmat = VarCoVar.copy()
    n_assets = len(actifs)

```

```

mean.index = actifs
covmat.index = actifs
mean_values = mean['Daily Return'].tolist()
#####
df_risk_ret = pd.DataFrame({'risk': np.diag(covmat) ** (1 / 2),
                           'return': mean_values})

x_lim = (0, df_risk_ret['risk'].max() * 1.1)
y_lim = (df_risk_ret['return'].min() * 1.1, df_risk_ret['return'].max() * 1.1)
ax = df_risk_ret.plot.scatter(x='risk', y='return')
ax.set_xlim(x_lim)
ax.set_ylim(y_lim)
n_timesteps = 1500
returns = pd.DataFrame(np.random.multivariate_normal(mean_values,
                                                    covmat.values,
                                                    size=n_timesteps,
                                                    check_valid='raise'), columns=mean.index)

class Markowitz:
    """Solutions to markowitz optimization problems.

    Parameters
    -----
    mean : np.ndarray
        1D array representing the mean of returns. Has shape `(n_assets,)`.

    covmat : np.ndarray
        2D array representing the covariance matrix of returns. Has shape `(n_assets, n_assets)`.

    """

    def __init__(self, mean, covmat):
        if mean.ndim != 1 and covmat.ndim != 2:
            raise ValueError('mean needs to be 1D and covmat 2D.')

        if not (mean.shape[0] == covmat.shape[0] == covmat.shape[1]):
            raise ValueError('Mean and covmat need to have the same number of assets.')

        self.mean = mean
        self.covmat = covmat

        self.n_assets = self.mean.shape[0]

    def minvar(self, max_weight=1.):
        """Compute minimum variance portfolio."""
        w = cp.Variable(self.n_assets)
        risk = cp.quad_form(w, self.covmat)
        prob = cp.Problem(cp.Minimize(risk),
                          [cp.sum(w) == 1,
                           w <= max_weight,
                           w >= 0])

        prob.solve()

        return w.value

    def maxret(self, max_weight=1.):
        """Compute maximum return portfolio."""
        w = cp.Variable(self.n_assets)
        ret = self.mean @ w
        prob = cp.Problem(cp.Maximize(ret),
                          [cp.sum(w) == 1,
                           w <= max_weight,
                           w >= 0])

        prob.solve()

        return w.value

```

```

        prob.solve()

        return w.value

    def maxutil(self, gamma, max_weight=1.):
        """Maximize utility."""
        w = cp.Variable(self.n_assets)
        ret = self.mean @ w
        risk = cp.quad_form(w, self.covmat)
        prob = cp.Problem(cp.Maximize(ret - gamma * risk),
                          [cp.sum(w) == 1,
                           w <= max_weight,
                           w >= 0])

        prob.solve()

        return w.value

    def compute_portfolio_moments(self, weights):
        """Compute mean and standard deviation of some allocation.

        Parameters
        -----
        weights : np.array
            1D array representing weights of a portfolio. Has shape `(n_assets,)`.

        Returns
        -----
        mean : float
            Mean (return)
        std : float
            Standard deviation (risk)
        """
        pmean = np.inner(weights, self.mean)
        pmean = pmean.item()
        pvar = weights @ self.covmat @ weights
        pvar = pvar.item()

        return pmean, pvar ** (1 / 2)

meannp = np.array(mean_values)
markowitz = Markowitz(meannp, covmat.values)
max_weight = 2 / n_assets
gamma = 10

optimal_portfolios_u = {'minvar': markowitz.minvar(max_weight=1.),
                        'maxret': markowitz.maxret(max_weight=1.),
                        'maxutil': markowitz.maxutil(gamma=gamma, max_weight=1.),
                        }
optimal_portfolios_c = {'minvar': markowitz.minvar(max_weight=max_weight),
                        'maxret': markowitz.maxret(max_weight=max_weight),
                        'maxutil': markowitz.maxutil(gamma=gamma, max_weight=max_weight)}

color_mapping = {'minvar': 'r',
                 'maxret': 'g',
                 'maxutil': 'yellow'}

mpl_config = {'s': 100,
              'alpha': 0.65,
              'linewidth': 0.5,
              'edgecolor': 'black'}

marker_mapping = {'true_c': '*',
                  'true_u': 'o',
                  'emp_c': 'v',

```



```

        'emp_u': 'p',
        'deep_c': 'p',
        'deep_u': 'x'}

def plot_scatter(title='', **risk_ret_portfolios):
    ax = df_risk_ret.plot.scatter(x='risk', y='return')
    ax.set_xlim(x_lim)
    ax.set_ylim(y_lim)
    plt.title(title)
    plt.tight_layout()

    all_points = []
    all_names = []

    global liste_noms
    global liste_risk
    global liste_ret
    liste_noms = []
    liste_risk = []
    liste_ret = []

    for name, portfolios in risk_ret_portfolios.items():
        for objective, w in portfolios.items():
            y, x = markowitz.compute_portfolio_moments(w)
            all_points.append(ax.scatter(x,
                                         y,
                                         c=color_mapping[objective],
                                         marker=marker_mapping[name],
                                         **mpl_config))
            all_names.append("{}_{}".format(objective, name))
            liste_noms.append("{}_{}".format(objective, name))
            liste_risk.append(x)
            liste_ret.append(y)

    plt.legend(all_points,
               all_names,
               scatterpoints=1,
               loc='lower left',
               ncol=len(risk_ret_portfolios),
               fontsize=8)

    markowitz_emp = Markowitz(returns.mean().values, returns.cov().values)
    emp_portfolios_u = {'minvar': markowitz_emp.minvar(max_weight=1.),
                       'maxret': markowitz_emp.maxret(max_weight=1.),
                       'maxutil': markowitz_emp.maxutil(gamma=gamma, max_weight=1.)}
    emp_portfolios_c = {'minvar': markowitz_emp.minvar(max_weight=max_weight),
                       'maxret': markowitz_emp.maxret(max_weight=max_weight),
                       'maxutil': markowitz_emp.maxutil(gamma=gamma, max_weight=max_weight)}

class Net(torch.nn.Module, Benchmark):
    """Learn covariance matrix, mean vector and gamma.

    One can enforce max weight per asset.
    """

    def __init__(self, n_assets, max_weight=1.):
        super().__init__()

        self.force_symmetric = True
        self.matrix = torch.nn.Parameter(torch.eye(n_assets), requires_grad=True)
        self.exp_returns = torch.nn.Parameter(torch.zeros(n_assets), requires_grad=True)

```

```

        self.gamma_sqrt = torch.nn.Parameter(torch.ones(1), requires_grad=True)

        self.portfolio_opt_layer = NumericalMarkowitz(n_assets, max_weight=max_weight)

    def forward(self, x):
        """Perform forward pass.

        Parameters
        -----
        x : torch.Tensor
            Tensor of shape `(n_samples, n_channels, lookback, n_assets)`.

        Returns
        -----
        weights_filled : torch.Tensor
            Of shape `(n_samples, n_assets)` representing the optimal weights as determined by the
            convex optimizer.
        """
        n = len(x)

        cov_sqrt = torch.mm(self.matrix, torch.t(self.matrix)) if self.force_symmetric else self.matrix

        weights = self.portfolio_opt_layer(self.exp_returns[None, ...],
                                          cov_sqrt[None, ...],
                                          self.gamma_sqrt,
                                          torch.zeros(1).to(device=x.device, dtype=x.dtype))

        weights_filled = torch.repeat_interleave(weights, n, dim=0)

        return weights_filled

    lookback, gap, horizon = 2, 0, 5 # lookback does not matter in this case, just used for consistency
    batch_size = 256
    n_epochs = 100 # We employ early stopping

    X_list, y_list = [], []

    for i in range(lookback, n_timesteps - horizon - gap + 1):
        X_list.append(returns.values[i - lookback:i, :])
        y_list.append(returns.values[i + gap:i + gap + horizon, :])

    X = np.stack(X_list, axis=0)[:, None, ...]
    y = np.stack(y_list, axis=0)[:, None, ...]

    dataset = InRAMDataset(X, y, asset_names=returns.columns)
    dataloader = RigidDataLoader(dataset, batch_size=batch_size)

    all_losses = {'minvar': StandardDeviation() ** 2,
                  'maxret': MeanReturns(),
                  'maxutil': MeanReturns() + gamma * StandardDeviation() ** 2}

    deep_portfolios_c = {}
    deep_portfolios_u = {}

    for mode in ['u', 'c']:
        for loss_name, loss in all_losses.items():
            network = Net(n_assets, max_weight=max_weight if mode == 'c' else 1.)
            run = Run(network,
                      loss,
                      dataloader,
                      val_dataloaders={'train': dataloader},
                      callbacks=[EarlyStoppingCallback('train', 'loss', patience=3)])

    run.launch(n_epochs=n_epochs)

```

```

# Results
w_pred = network(torch.ones(1, n_assets)).detach().numpy().squeeze() # the input does not matter

if mode == 'c':
    deep_portfolios_c[loss_name] = w_pred
else:
    deep_portfolios_u[loss_name] = w_pred

#####
plt_scatter(title='Ground truth vs empirical estimates vs deep learning: Unconstrained',
            true_u=optimal_portfolios_u,
            emp_u=emp_portfolios_u,
            deep_u=deep_portfolios_u)
#plt.savefig('Comparison NORMAL vs DEEP/'+ 'pf_' + str(nb_pf + 1) + '.png')
#####
rf_us_yearly = 0.0158 #US 10Y 30 May 2021
rf_daily = rf_us_yearly / math.sqrt(252) #Désannualiser en divisant par racine(252)

tableau_comparaison.iat[nb_pf,0] = nb_pf + 1 #car commence à 0
tableau_comparaison.iat[nb_pf,1] = actifs

tableau_comparaison.iat[nb_pf,2] = liste_risk[liste_noms.index("minvar_true_u")]
tableau_comparaison.iat[nb_pf,3] = liste_risk[liste_noms.index("maxret_true_u")]
tableau_comparaison.iat[nb_pf,4] = liste_risk[liste_noms.index("maxutil_true_u")]
tableau_comparaison.iat[nb_pf,5] = liste_ret[liste_noms.index("minvar_true_u")]
tableau_comparaison.iat[nb_pf,6] = liste_ret[liste_noms.index("maxret_true_u")]
tableau_comparaison.iat[nb_pf,7] = liste_ret[liste_noms.index("maxutil_true_u")]

tableau_comparaison.iat[nb_pf,8] = ((tableau_comparaison.iat[nb_pf,5] - rf_daily) / tableau_comparaison.iat[nb_pf,6] - rf_daily) / tableau_comparaison.iat[nb_pf,7] - rf_daily
tableau_comparaison.iat[nb_pf,9] = ((tableau_comparaison.iat[nb_pf,5] - rf_daily) / tableau_comparaison.iat[nb_pf,6] - rf_daily) / tableau_comparaison.iat[nb_pf,7] - rf_daily
tableau_comparaison.iat[nb_pf,10] = ((tableau_comparaison.iat[nb_pf,5] - rf_daily) / tableau_comparaison.iat[nb_pf,6] - rf_daily) / tableau_comparaison.iat[nb_pf,7] - rf_daily

tableau_comparaison.iat[nb_pf,11] = liste_risk[liste_noms.index("minvar_deep_u")]
tableau_comparaison.iat[nb_pf,12] = liste_risk[liste_noms.index("maxret_deep_u")]
tableau_comparaison.iat[nb_pf,13] = liste_risk[liste_noms.index("maxutil_deep_u")]
tableau_comparaison.iat[nb_pf,14] = liste_ret[liste_noms.index("minvar_deep_u")]
tableau_comparaison.iat[nb_pf,15] = liste_ret[liste_noms.index("maxret_deep_u")]
tableau_comparaison.iat[nb_pf,16] = liste_ret[liste_noms.index("maxutil_deep_u")]

tableau_comparaison.iat[nb_pf,17] = ((tableau_comparaison.iat[nb_pf,14] - rf_daily) / tableau_comparaison.iat[nb_pf,15] - rf_daily) / tableau_comparaison.iat[nb_pf,16] - rf_daily
tableau_comparaison.iat[nb_pf,18] = ((tableau_comparaison.iat[nb_pf,14] - rf_daily) / tableau_comparaison.iat[nb_pf,15] - rf_daily) / tableau_comparaison.iat[nb_pf,16] - rf_daily
tableau_comparaison.iat[nb_pf,19] = ((tableau_comparaison.iat[nb_pf,14] - rf_daily) / tableau_comparaison.iat[nb_pf,15] - rf_daily) / tableau_comparaison.iat[nb_pf,16] - rf_daily

if tableau_comparaison.iat[nb_pf,8] > tableau_comparaison.iat[nb_pf,17]:
    tableau_comparaison.iat[nb_pf,20] = "Traditionnel"
elif tableau_comparaison.iat[nb_pf,8] < tableau_comparaison.iat[nb_pf,17]:
    tableau_comparaison.iat[nb_pf,20] = "IA"
else:
    tableau_comparaison.iat[nb_pf,20] = "Egal"

if tableau_comparaison.iat[nb_pf,9] > tableau_comparaison.iat[nb_pf,18]:
    tableau_comparaison.iat[nb_pf,21] = "Traditionnel"
elif tableau_comparaison.iat[nb_pf,9] < tableau_comparaison.iat[nb_pf,18]:
    tableau_comparaison.iat[nb_pf,21] = "IA"
else:
    tableau_comparaison.iat[nb_pf,21] = "Egal"

if tableau_comparaison.iat[nb_pf,10] > tableau_comparaison.iat[nb_pf,19]:
    tableau_comparaison.iat[nb_pf,22] = "Traditionnel"
elif tableau_comparaison.iat[nb_pf,10] < tableau_comparaison.iat[nb_pf,19]:
    tableau_comparaison.iat[nb_pf,22] = "IA"
else:
    tableau_comparaison.iat[nb_pf,22] = "Egal"

```

```

somme_tradi = 0
somme_IA = 0
somme_egal = 0

for nb_col in range(20,23): #de 20 à 22
    if tableau_comparaison.iat[nb_pf,nb_col] == "Traditionnel":
        somme_tradi = somme_tradi + 1
    elif tableau_comparaison.iat[nb_pf,nb_col] == "IA":
        somme_IA = somme_IA + 1
    else:
        somme_egal = somme_egal + 1

sommess = {"IA": somme_IA, "Traditionnel": somme_tradi, "Egalité": somme_egal}

values = list(sommess.values())
keys = list(sommess.keys())

valeur1 = max(values)
index1 = values.index(valeur1)
nom1 = (keys[index1])
val1 = valeur1
sommess.pop(keys[index1], None)

values = list(sommess.values())
keys = list(sommess.keys())

valeur2 = max(values)
index2 = values.index(valeur2)
nom2 = (keys[index2])
val2 = valeur2

#print(nom1, val1)
#print(nom2, val2)

if somme_tradi == somme_IA == somme_egal:
    tableau_comparaison.iat[nb_pf,23] = ("Egalité")
    tableau_comparaison.iat[nb_pf,24] = (somme_egal)
elif somme_tradi == somme_IA == 1 or somme_tradi == somme_egal == 1 or somme_IA == somme_egal == 1:
    tableau_comparaison.iat[nb_pf,23] = (nom1 + " " + nom2)
    tableau_comparaison.iat[nb_pf,24] = (val1)
else:
    tableau_comparaison.iat[nb_pf,23] = (nom1)
    tableau_comparaison.iat[nb_pf,24] = (val1)

tableau_comparaison.to_csv('Comparison NORMAL vs DEEP/Tableau spé IA x4.csv',index=False)
tableau_comparaison.to_excel('Comparison NORMAL vs DEEP/Tableau spé IA x4.xlsx',
                             sheet_name='Liste Valeurs')

```

```

#Tableau occurrence actifs
import pandas as pd
import ast
import ast
import numpy as np
import yfinance as yf
import math

tableau_comparaison = pd.read_csv('Comparaison NORMAL vs DEEP/Tableau spé IA x4.csv')

liste_col = ["Actifs"]
liste_actifs = []
liste_stockpicke = []
type_stockpicke = "IA" #Traditionnel Egalité
nb_stockpicke = 1

for nb in range(0,18):
    liste_col.append("Actif_" + str(nb + 1))
for nb in range(0,18):
    liste_col.append("Type" + "Actif_" + str(nb + 1))

tableau_actifs = pd.DataFrame("", index = list(range(0, 100)), columns = liste_col)

for nb_pf in range(0,100):
    liste = tableau_comparaison.iat[nb_pf,1]
    liste = ast.literal_eval(liste)
    tableau_actifs.iat[nb_pf,0] = liste
    for nb_act in range(0,18):
        tableau_actifs.iat[nb_pf,nb_act+1] = liste[nb_act]
        tableau_actifs.iat[nb_pf,nb_act+19] = liste[nb_act] + " " + tableau_comparaison.iat[nb_pf,22] #22 max util ;
        if not liste[nb_act] in liste_actifs:
            liste_actifs.append(liste[nb_act])
#### count
tbl = pd.DataFrame()

for colo in range(19,37):
    nom = tableau_actifs.columns[colo]
    #tbl[nom] = tableau_actifs[nom].value_counts()
    tbl = pd.concat([tbl,tableau_actifs[nom].value_counts()], ignore_index=True, axis=1)
tbl["sum"] = tbl.sum(axis=1)
####
tbl_lst_act = pd.DataFrame(columns=['Actifs', 'Beta', 'Alpha', 'Traditionnel', 'IA', 'Total',
                                   'Type principal', 'Rang si IA', 'Ticker si IA', 'Liste stockpickée'])
tbl_lst_act['Actifs'] = liste_actifs

#Bêta actions
tbl_sp = pd.DataFrame()
indice = ['SPY']

rf_us_yearly = 0.0158 #US 10Y 30 May 2021, conforme au fichier IA.
rf_daily = rf_us_yearly / math.sqrt(252) #Désannualiser en divisant par racine(252)

tmp_close = yf.download(indice,
                        start='2017-01-01',
                        end='2020-01-03',
                        progress=False)['Close']
tbl_sp = pd.concat([tbl_sp, tmp_close], axis=1)
tbl_sp.columns = indice
tbl_sp = np.log(tbl_sp/tbl_sp.shift(1))
tbl_sp = tbl_sp.tail(tbl_sp.shape[0] -1)

tableau_lr_sp = pd.read_csv('TEMPTEST/loge>Returns-SP.csv')
tableau_lr_sp = tableau_lr_sp.tail(tableau_lr_sp.shape[0] -1)
tableau_lr_sp = tableau_lr_sp.fillna(0)

#pd.set_option('display.max_rows', None)
#print(tableau_lr_sp[list_actifs[0]])
#print(tbl_sp[indice[0]])

for nb_actif in range(len(liste_actifs)):
    correl = np.corrcoef(tableau_lr_sp[list_actifs[nb_actif]], tbl_sp[indice[0]])[0,1]
    risk_p = tableau_lr_sp[list_actifs[nb_actif]].std()
    risk_m = tbl_sp[indice[0]].std()
    beta = correl * (risk_p / risk_m)

#Alpha actions
ret_p = tableau_lr_sp[list_actifs[nb_actif]].mean()
ret_m = tbl_sp[indice[0]].mean()
#yearly
y_ret_p = math.pow(1 + ret_p,252)-1
y_ret_m = math.pow(1 + ret_m,252)-1
alpha = (y_ret_p - rf_us_yearly) - beta * (y_ret_m - rf_us_yearly)

#Valeurs inscrites
tbl_lst_act.iat[nb_actif,1] = beta
tbl_lst_act.iat[nb_actif,2] = alpha
#Compter les occurrences IA / tradi d'actifs
#print(str(tbl_lst_act.iat[nb_actif,0] + " " + "Traditionnel"))
if ((tbl.index == str(tbl_lst_act.iat[nb_actif,0] + " " + "Traditionnel")).any()) == True:
    tbl_lst_act.iat[nb_actif,3] = tbl.loc[str(tbl_lst_act.iat[nb_actif,0] + " " + "Traditionnel"), 'sum'] #tradi
else:
    tbl_lst_act.iat[nb_actif,3] = 0
#print(str(tbl_lst_act.iat[nb_actif,0] + " " + "IA"))
if ((tbl.index == str(tbl_lst_act.iat[nb_actif,0] + " " + "IA")).any()) == True:
    tbl_lst_act.iat[nb_actif,4] = tbl.loc[str(tbl_lst_act.iat[nb_actif,0] + " " + "IA"), 'sum'] #ia
else:
    tbl_lst_act.iat[nb_actif,4] = 0
tbl_lst_act.iat[nb_actif,5] = (tbl_lst_act.iat[nb_actif,3] + tbl_lst_act.iat[nb_actif,4]) #total
if tbl_lst_act.iat[nb_actif,3] > tbl_lst_act.iat[nb_actif,4]: #tradi > ia
    tbl_lst_act.iat[nb_actif,6] = "Traditionnel"
elif tbl_lst_act.iat[nb_actif,3] < tbl_lst_act.iat[nb_actif,4]: #tradi < ia
    tbl_lst_act.iat[nb_actif,6] = "IA"
else:
    tbl_lst_act.iat[nb_actif,6] = "Egalité"
if tbl_lst_act.iat[nb_actif,6] == type_stockpicke:
    liste_stockpicke.append(tbl_lst_act.iat[nb_actif,0])
tbl_lst_act.iat[nb_actif,7] = nb_stockpicke
tbl_lst_act.iat[nb_actif,8] = tbl_lst_act.iat[nb_actif,0]
nb_stockpicke = nb_stockpicke + 1

tbl_stockpicke = pd.DataFrame()
tbl_stockpicke['Liste stockpickée'] = liste_stockpicke

writer = pd.ExcelWriter('Comparaison NORMAL vs DEEP/Actifs spé IA x4.xlsx', engine='xlsxwriter')
tableau_actifs.to_excel(writer, sheet_name='Liste Actifs')
tbl_lst_act.to_excel(writer, sheet_name='Actifs')
tbl_stockpicke.to_excel(writer, sheet_name='Stockpick')
writer.save()

```

Annexe 2 : Codes PYTHON de stockpicking

La première partie s'occupe d'importer et créer deux tableaux : le tableau des tickers et des P/E correspondants de liste des actifs du SP500 ; et le tableau des secteurs GICS du SP500. Pour ce faire, la bibliothèque Pandas permet d'importer des tableaux de page web. Ainsi, en choisissant des sites contenant nos données, une requête permet de les importer et de les mettre en forme. Un troisième tableau est créé, qui est la fusion des deux premiers. Ainsi, ce troisième tableau qui comporte le ticker, le secteur correspondant et le P/E de l'actif, est enregistré en format « csv ».

La deuxième partie importe les données journalières, monthly et les recommandations d'achat des secteurs du SP500. Après mise en forme, 4 tableaux sont enregistrés. Le premier, « Tableau GICS Recherche », est le tableau qui comporte les secteurs, la moyenne de variation journalière et mensuelle, le rating ainsi que 2 valeurs (commodités, indices ou ETFs) liées à ce secteur, et dont la variation est bonne indicatrice du secteur. Les trois autres tableaux sont le recueil des données du premier.

La troisième partie importe les données des valeurs liées aux secteurs. Elle importe le premier tableau de la partie 2, recherche le nom des commodités / indices / ETFs et enregistre la valeur moyenne mensuelle de variation. Ce tableau est ensuite enregistré sous le nom « Tableau GICS complet ».

La quatrième partie effectue la comparaison des secteurs. Elle évalue les ratings et rendements du tableau précédent en donnant un certain nombre de points. Une boucle avec plusieurs conditions vérifie chaque point à évaluer et attribue un certain nombre de points, et effectue la somme de ces notations, ce qui donne une note sur 100.

La cinquième partie est le stockpicking final. Sachant que notre stratégie est du style « buy and hold », nous négligeons par choix les variations de cours récentes et ne mettons pas de critères liés. Nos secteurs étant évalués, il nous reste à sélectionner les « meilleurs ». Un tri de note décroissante pour les secteurs puis de valeur croissante pour les P/E nous donne un tableau pour lequel il ne reste plus qu'à sélectionner les 3 meilleurs valeurs pour chacun des 6 meilleurs secteurs et d'enregistrer cette liste d'actifs stockpickés.

```

#### IMPORT DE LA LISTE SP500, ET RECUPERATION DES P/E
#POSITION : 1)
import pandas as pd

tableaux = pd.read_html('https://www.liberatedstocktrader.com/sp-500-companies-list-by-sector-market-cap/')
tbl_sp500_PE = tableaux[1]
tbl_sp500_PE.drop(labels = [2,3],axis = 1, inplace = True)
tbl_sp500_PE.columns = ['Industry Sector', 'Symbol', 'P/E']
tbl_sp500_PE.drop(labels = [0],axis = 0, inplace = True)
tbl_sp500_PE = tbl_sp500_PE.reset_index(drop=True)

for lignes in range(len(tbl_sp500_PE)):
    valeur = tbl_sp500_PE.iat[lignes,2]
    valeur = valeur.replace(".", "")
    valeur = valeur.replace("-", "")
    valeur = valeur.replace("100+", "101")
    tbl_sp500_PE.iat[lignes,2] = valeur
tbl_sp500_PE["P/E"] = pd.to_numeric(tbl_sp500_PE["P/E"])

tableaux = pd.read_html('https://en.wikipedia.org/wiki/List_of_S%26P_500_companies')
tbl_sp500 = tableaux[0]
tbl_sp500.drop(labels = ["SEC filings", "Headquarters Location", "Date first added", "CIK", "Founded"],axis = 1, inplace = True)

tbl_sp500_simple = pd.merge(tbl_sp500,
                            tbl_sp500_PE,
                            on = 'Symbol',
                            how = 'right')
tbl_sp500_simple.to_csv(r'Tableau SP500 Complet avec PE.csv',index=False)

tbl_sp500_simple.drop(labels = ["GICS Sub-Industry", "Industry Sector"],axis = 1, inplace = True)
tbl_sp500_simple.to_csv(r'Tableau SP500 GICS avec PE.csv',index=False)

```

```

#### IMPORT DES DONNEES Journalières , Monthly et recommandations d'achat pour secteurs SP500
#POSITION : 2)
import pandas as pd
import numpy as np

tableaux = pd.read_html('https://fr.tradingview.com/markets/indices/quotes-snp/')
tbl_tradingview = tableaux[0]
tbl_tradingview.columns = ['Sector', 'Last Price', '% 1D', '1D', 'High', 'Low', 'Rating']
for ligne_tableau in range(len(tbl_tradingview)):
    texte_a_modifier = tbl_tradingview.iat[ligne_tableau,0]
    head, sep, tail = texte_a_modifier.partition('S&P 500 ')
    tbl_tradingview.iat[ligne_tableau,0] = str('S&P ' + tail) #str('S&P ' + tail)
tbl_tradingview = tbl_tradingview.sort_values('Sector')
tbl_tradingview = tbl_tradingview.reset_index(drop=True)
tbl_tradingview['% 1D'] = tbl_tradingview['% 1D'].str.rstrip('%').astype('float')

tableaux = pd.read_html('http://www.lazyportfolioetf.com/sp-500-sector-returns/')
tbl_lazypf = tableaux[0]
tbl_lazypf.drop(labels = [3],axis = 0, inplace = True)
tbl_lazypf.drop(labels = ["Unnamed: 0"],axis = 1, inplace = True)
for ligne_tableau in range(len(tbl_lazypf)):
    texte_a_modifier = tbl_lazypf.iat[ligne_tableau,0]
    head, sep, tail = texte_a_modifier.partition(' (X')
    if head == "S&P Healthcare":
        head = "S&P Health Care"
    tbl_lazypf.iat[ligne_tableau,0] = str(head)
tbl_lazypf.columns = ['Sector', '% 1M', '% 3M', '% 6M', '% 1Y', '% 3Y', '% 5Y', '% 10Y']
tbl_lazypf = tbl_lazypf.sort_values('Sector')
tbl_lazypf = tbl_lazypf.reset_index(drop=True)

tbl_secteursSP = pd.merge(tbl_tradingview,
                          tbl_lazypf,
                          on = 'Sector',
                          how = 'right')

tbl_secteursSP.drop(labels = ["Last Price", "1D", "High", "Low", "% 3M", "% 6M", "% 1Y", "% 3Y", "% 5Y", "% 10Y"],axis = 1, inplace = True)
for ligne_tableau in range(len(tbl_secteursSP)):
    texte_a_modifier = tbl_secteursSP.iat[ligne_tableau,0]
    head, sep, tail = texte_a_modifier.partition('S&P ')
    tbl_secteursSP.iat[ligne_tableau,0] = str(tail)

tbl_gics = pd.read_excel(r'Secteurs GICS.xlsx', sheet_name='GICS')
tbl_gics = tbl_gics.replace(np.nan, '', regex=True)
tbl_gics['% 1D'] = tbl_secteursSP['% 1D']
tbl_gics['% 1M'] = tbl_secteursSP['% 1M']
tbl_gics['Rating'] = tbl_secteursSP['Rating']

tbl_gics.to_csv(r'Tableau GICS Recherche.csv',index=False)
tbl_secteursSP.to_csv(r'Tableau SP500 Secteurs.csv',index=False)
tbl_tradingview.to_csv(r'Tableau TradingView GICS.csv',index=False)
tbl_lazypf.to_csv(r'Tableau LazyPortfolio GICS.csv',index=False)

```

```

#### IMPORTER DATA DE COMMODITIES
#POSITION : 3)

#Importer liste des commos à chercher
tbl_recherche = pd.read_csv(r'Tableau GICS Recherche.csv')
tbl_recherche = tbl_recherche.replace(np.nan, '', regex=True)
tbl_valeurs = tbl_recherche.copy()

import investpy
import math

colonne_tbl = 4

for colonne_tbl in range(4, len(tbl_recherche.columns)):
    for ligne_tbl in range(len(tbl_recherche)):
        nom_actif = tbl_recherche.iat[ligne_tbl,colonne_tbl]
        if nom_actif != "":
            if tbl_recherche.columns[colonne_tbl] == "Commodity 1" or tbl_recherche.columns[colonne_tbl] == "Commodity
                tbl_actif = investpy.commodities.get_commodity_historical_data(commodity = nom_actif,
                    from_date = '01/01/2017',
                    to_date = '01/01/2020',
                    country=None,
                    as_json=False,
                    order='ascending',
                    interval='Daily')

            if tbl_recherche.columns[colonne_tbl] == "ETF":
                tbl_actif = investpy.etfs.get_etf_historical_data(etf = nom_actif,
                    country = 'united states',
                    from_date = '01/01/2017',
                    to_date = '01/01/2020',
                    stock_exchange=None,
                    as_json=False,
                    order='ascending',
                    interval='Daily')

            if tbl_recherche.columns[colonne_tbl] == "Indice":
                tbl_actif = investpy.indices.get_index_historical_data(index = nom_actif,
                    country = 'united states',
                    from_date = '01/01/2017',
                    to_date = '01/01/2020',
                    as_json=False,
                    order='ascending',
                    interval='Daily')

            monthly_yield = round(float(((tbl_actif.iat[len(tbl_actif)-1,0]/tbl_actif.iat[0,0]-1)*100),2)
            tbl_valeurs.iat[ligne_tbl,colonne_tbl] = monthly_yield

tbl_valeurs.to_csv(r'Tableau GICS Complet.csv',index=False)

```

```

#### EVALUATION DES SECTEURS
#POSITION : 4)

tbl_final = pd.read_csv(r'Tableau GICS Complet.csv')
tbl_final["Eval GICS"] = ""
tbl_final["Eval Sect 1"] = ""
tbl_final["Eval Sect 2"] = ""
tbl_final["Eval Rating"] = ""
tbl_final["Note du secteur (sur 100)"] = ""
tbl_final = tbl_final.replace(np.nan, '', regex=True)

#col départ = 2

for ligne_tbl in range(len(tbl_final)):
    cptr_pos = 0
    for colonne_tbl in range(2, 8):
        val_a_eval = tbl_final.iat[ligne_tbl,colonne_tbl]
        if val_a_eval != "":
            if tbl_final.columns[colonne_tbl] == "Rating":
                if val_a_eval == "Fort achat":
                    tbl_final.iat[ligne_tbl,11] = 25
                elif val_a_eval == "Acheter":
                    tbl_final.iat[ligne_tbl,11] = 13
                elif val_a_eval == "Vendre" or val_a_eval == "Forte vente":
                    tbl_final.iat[ligne_tbl,11] = 0
            else:
                if tbl_final.columns[colonne_tbl] == "% 1M":
                    position_col = 8
                elif tbl_final.iat[ligne_tbl,9] == "":
                    position_col = 9
                elif tbl_final.iat[ligne_tbl,10] == "":
                    position_col = 10
                if val_a_eval >= 10:
                    tbl_final.iat[ligne_tbl,position_col] = 25
                elif val_a_eval >= 5:
                    tbl_final.iat[ligne_tbl,position_col] = 15
                elif val_a_eval >= 3:
                    tbl_final.iat[ligne_tbl,position_col] = 9
                elif val_a_eval >= 1:
                    tbl_final.iat[ligne_tbl,position_col] = 5
                elif val_a_eval >= 0:
                    tbl_final.iat[ligne_tbl,position_col] = 3
                elif val_a_eval < 0:
                    tbl_final.iat[ligne_tbl,position_col] = 0
            tbl_final.iat[ligne_tbl,12] = tbl_final.iat[ligne_tbl,8]+tbl_final.iat[ligne_tbl,9]+tbl_final.iat[ligne_tbl,10]+tbl

tbl_final_GICS = tbl_final.copy()
tbl_final_GICS.drop(labels = ['% 1D', "% 1M", "Rating", "Commodity 1", "Commodity 2", "ETF", "Indice", "Eval GICS", '
tbl_final_GICS.columns = ['GICS Sector', 'Note du secteur (sur 100)']

tbl_final_GICS.to_csv(r'Evaluation GICS Simple.csv',index=False)
tbl_final.to_csv(r'Evaluation GICS Complète.csv',index=False)

```

```

#### STOCKPICKING
#POSITION : 5

import pandas as pd
import math

nb_act_par_sect = 3 #Nombre d'actifs par secteurs

tbl_sp500 = pd.read_csv(r'Tableau SP500 GICS avec PE.csv')
tbl_eval = pd.read_csv(r'Evaluation GICS Simple.csv')
tbl_eval = tbl_eval.sort_values(['Note du secteur (sur 100)'], ascending=[False])

moyenne_secteur = int(tbl_eval["Note du secteur (sur 100)"].mean() + 1)
quartilesup_sect = []
sup_sect = []
liste_stockpick = []

for ligne in range(len(tbl_eval)):
    if tbl_eval.iat[ligne,1] >= moyenne_secteur:
        sup_sect.append(str(tbl_eval.iat[ligne,1]) + " " + str(tbl_eval.iat[ligne,0]))
sup_sect.sort(reverse = True)
for secteurs in range(len(sup_sect)):
    secteur = sup_sect[secteurs]
    head, sep, tail = secteur.partition(' ')
    quartilesup_sect.append(tail)

tbl_stockpick = pd.merge(tbl_sp500,
                        tbl_eval,
                        on = 'GICS Sector',
                        how = 'right')

tbl_stockpick["Tri temporaire"] = ""
for ligne in range(len(tbl_stockpick)):
    tbl_stockpick.iat[ligne,5] = (str(tbl_stockpick.iat[ligne,4]) + " " + str(tbl_stockpick.iat[ligne,2]))
tbl_stockpick = tbl_stockpick.sort_values(['Tri temporaire', 'P/E'], ascending=[False, True])
tbl_stockpick.drop (labels = ["Tri temporaire"],axis = 1, inplace = True)
tbl_stockpick.to_csv(r'Tableau Stockpicking.csv',index=False)
tbl_stockpick = tbl_stockpick.reset_index(drop=True)

nb_actions = len(quartilesup_sect)*3

nb_ligne = 0
for nb_sect in range(len(quartilesup_sect)):
    while tbl_stockpick.iat[nb_ligne,2] != quartilesup_sect[nb_sect]:
        nb_ligne = nb_ligne + 1
    if tbl_stockpick.iat[nb_ligne,2] == quartilesup_sect[nb_sect]:
        liste_stockpick.append(tbl_stockpick.iat[nb_ligne,0])
        liste_stockpick.append(tbl_stockpick.iat[nb_ligne+1,0])
        if nb_act_par_sect >= 3:
            liste_stockpick.append(tbl_stockpick.iat[nb_ligne+2,0])
            if nb_act_par_sect >= 4:
                liste_stockpick.append(tbl_stockpick.iat[nb_ligne+3,0])

print(liste_stockpick)

```

Annexe 3 : Code PYTHON de frontière efficiente

Grâce à la bibliothèque FinQuant, qui simplifie en temps et en nombre de lignes les calculs d'optimisation, il est possible de mettre en place une frontière efficiente en se basant sur des données de rendement téléchargeables avec une API.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import quandl
import yfinance as yf
import scipy.optimize as sco
quandl.ApiConfig.api_key = 'z4D7GxttMD5fLxFFxNdz'
from finquant.portfolio import build_portfolio

liste_stockpick = ['CE', 'SEE', 'NEM', 'ALL', 'UNM', 'AFL', 'KIM', 'BXP', 'EQR', 'INTC', 'WU', 'HPQ', 'DISH', 'OMC',

names = []
for nbr in range(len(liste_stockpick)):
    names.append("WIKI/"+liste_stockpick[nbr])

start_date = '2017-01-01'
end_date = '2021-01-01'
pf = build_portfolio(names=names,
                    start_date=start_date,
                    end_date=end_date)

# performs and plots results of Monte Carlo run (5000 iterations)
opt_w, opt_res = pf.mc_optimisation(num_trials=5000)
pf.mc_plot_results()
pf.ef_plot_efrontier()
pf.ef.plot_optimal_portfolios()
```