



**IMT Nord Europe**  
École Mines-Télécom  
IMT-Université de Lille

# Rapport UV Projet

## Développement d'un module d'optimisation de portefeuille et de prédiction de la rentabilité des actions en bourses



**Élèves :**

Romain MRAD - Yoan HAKOUN

**Tuteurs :**

Radhouane KAMMOUN - Christelle GARNIER - Vincent ITIER

# Remerciements

Avant de réaliser le développement de ce rapport, il est nécessaire de remercier les personnes qui nous ont permis de réaliser ce projet et qui en ont fait une belle expérience.

Nous tenons donc d'abord à remercier madame Christelle Garnier et monsieur Vincent Itier pour leur suivi et leur aide quant à la partie du projet en lien avec la Data Science.

Nous tenons ensuite à remercier monsieur Radhouane Kammoun pour son suivi et son aide quant à la partie du projet en lien avec la finance de marché.

Enfin, il se doit de remercier l'IMT Nord Europe pour nous avoir permis de réaliser cet UV Projet en 2e année de cycle ingénieur, dans le cadre de la cinquième période d'études.

# Table of Contents

<b>Remerciements</b>	<b>1</b>
<b>Table of Contents</b>	<b>2</b>
<b>Glossaire</b>	<b>3</b>
<b>Introduction</b>	<b>4</b>
<b>Définition du projet</b>	<b>4</b>
<b>Présentation du travail effectué</b>	<b>4</b>
Hypothèses financières	4
Stratégie d'investissement	5
LSTM	5
Premières approches	6
Évaluation des modèles	8
Construction et entraînement des modèles prédictifs à tester	9
<b>Critique des résultats</b>	<b>11</b>
<b>Analyse financière et axes d'amélioration</b>	<b>12</b>
<b>Conclusion</b>	<b>12</b>
<b>Bibliographie</b>	<b>13</b>
<b>Guide d'utilisation</b>	<b>14</b>
Téléchargement	14
Fonctionnement	14
Jour 0	14
Jour N	14
<b>Annexes</b>	<b>15</b>
Diagrammes UML	15
Visualisations	15
Code source	15
Configuration	16

# Glossaire

Portfolio : “portefeuille” en anglais.

Bullish : traduit une tendance haussière.

Bearish : traduit une tendance baissière.

Diagramme UML : Diagramme “Unified Modelling Language” est un diagramme qui permet de visualiser la structure d’une classe, ou la séquence d’un programme.

Cours : le cours boursier est le prix auquel s’échange un titre.

Titre : un titre financier correspond à une partie du capital d'une société (action), ou à une partie de sa dette (obligation).

Cours de clôture ajusté : le cours de clôture ajusté est le cours de clôture après ajustement de tous les fractionnements et distributions de dividendes applicables

Rentabilité journalière : évolution d’un cours du jour  $N$  au jour  $N + 1$ . Donné par la formule :

$$rentabilité = \frac{cours_{N+1}}{cours_N} - 1$$

# Introduction

Dans un monde où les avancées technologiques révolutionnent sans cesse les pratiques financières, le mariage entre la finance de marché et le Machine Learning représente une frontière fertile en innovations. Dans ce contexte dynamique, nous avons entrepris le développement d'un module d'optimisation de portefeuille et de prédiction de la rentabilité des actions en bourse.

Les actions en bourse constituent des séries temporelles intrinsèques. Chaque transaction sur le marché crée un flux continu de données qui évolue au fil du temps. Ces données historiques fournissent une source d'information riche sur les tendances passées et permettent d'anticiper les mouvements futurs du marché. Les séries temporelles présentent des caractéristiques telles que la saisonnalité, les tendances et les cycles, ce qui les rend idéales pour l'application de techniques de modélisation temporelle comme les réseaux de neurones.

Ainsi, notre projet s'inscrit dans ce contexte d'innovation, visant à exploiter les avancées du Machine Learning pour optimiser la gestion de portefeuille et anticiper les mouvements du marché boursier. Dans les sections suivantes, nous détaillerons la définition de notre projet, les travaux réalisés ainsi que les résultats obtenus.

## Définition du projet

Dans le cadre de la P5, nous avons décidé de réaliser un UV Projet liant la finance de marché et le Machine Learning.

Le projet consiste à développer un module Python qui permet d'abord d'initialiser un portefeuille par rapport à un capital donné et d'optimiser sa répartition en évaluant un paramètre ou en prédisant les cours des actions en bourse.

Ensuite, le module devra entraîner un modèle de prédiction pour chaque action afin de prédire leur cours et l'évolution générale du portefeuille.

Enfin, le programme suggérera un changement dans le portefeuille s'il est prédit à la baisse, et permettra d'agir sur ce changement en demandant à l'utilisateur.

## Présentation du travail effectué

Avant d'aborder la partie technique, il est nécessaire d'explicitier nos hypothèses financières et notre stratégie d'investissement. Ces dernières seront intégrées dans la programmation.

### Hypothèses financières

Sur le marché **Euronext Paris**, nous avons entrepris une collecte exhaustive de toutes les actions disponibles afin de les intégrer dans notre analyse. Toutefois, pour garantir la qualité des données et la pertinence de nos modèles, nous avons jugé nécessaire d'effectuer un processus de filtrage. Ce filtrage consiste principalement à exclure deux catégories d'actions : celles considérées comme récentes et celles dont le prix était jugé relativement bas.

D'une part, les actions récentes, définies comme celles pour lesquelles il n'existait pas suffisamment de données historiques pour permettre un entraînement adéquat d'un réseau de neurones ou d'autres modèles d'analyse, ont été retirées de notre échantillon. Cette décision découle de la nécessité d'avoir un historique de données significatif pour que nos analyses puissent produire des résultats fiables et informatifs.

D'autre part, nous avons également exclu les actions dont le prix de marché était inférieur à 5€. Ce seuil a été choisi de manière arbitraire, mais il est basé sur une évaluation du marché selon laquelle les actions cotées en dessous de ce seuil sont souvent considérées comme étant à risque de disparition ou de sous-performance à long terme. En éliminant ces actions, nous cherchons à construire un ensemble de données plus robuste et représentatif, en nous concentrant sur des titres qui sont généralement plus liquides et qui ont une probabilité plus élevée de fournir des signaux d'analyse significatifs.

Parallèlement, nous avons décidé d'utiliser exclusivement le cours de clôture ajusté de chaque action. Cette mesure, qui prend en compte les ajustements tels que les dividendes et les émissions d'actions, offre une vision plus précise de la performance réelle de l'action.

Enfin, dans le but de refléter au mieux les conditions du marché réel, nous avons fixé les frais de transaction à un montant de 2€. Cette décision vise à prendre en compte les coûts associés à l'achat et à la vente d'actions, et à garantir que nos simulations et nos analyses tiennent compte de ces aspects pratiques de l'investissement en actions.

## Stratégie d'investissement

La stratégie mise en œuvre repose sur un investissement à long terme. En se basant uniquement sur les cours de clôture ajustés, le programme doit être activé quotidiennement après la clôture du marché parisien. Par la suite, des ajustements ne sont suggérés que si le programme prévoit une tendance à la baisse dans le portefeuille.

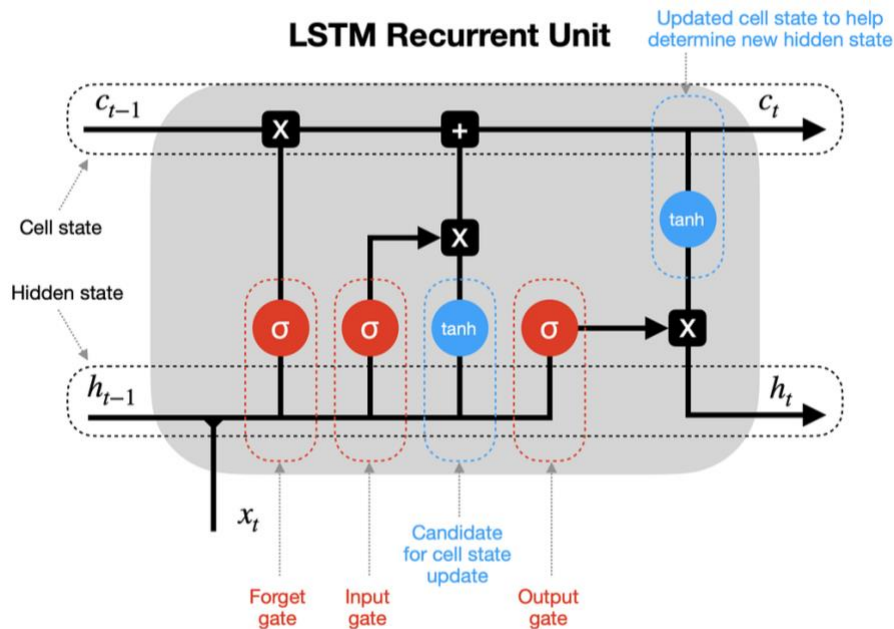
D'un côté, si le cours de clôture ajusté du jour  $N + 1$  (prédit) est inférieur à celui du jour  $N$  (connu), il est opportun de vendre cette action à l'ouverture du marché le jour  $N + 1$ .

D'un autre côté, si le cours de clôture ajusté du jour  $N + 1$  (prédit) est supérieur à celui du jour  $N$  (connu), il est préférable de maintenir cette action dans le portefeuille, car elle est susceptible de prendre de la valeur.

## LSTM

Les réseaux de neurones récurrents (**RNN**) constituent une classe de modèles particulièrement adaptés à la modélisation des séries temporelles. Les couches **LSTM** (Long Short-Term Memory) sont un type spécial de **RNN** conçues pour surmonter les limitations des réseaux traditionnels en matière de mémorisation à long terme. Contrairement aux **RNN** classiques, les **LSTM** sont capables de maintenir une mémoire à long terme et de gérer les dépendances dans les données séquentielles. Chaque unité **LSTM** comporte une cellule mémoire, des portes d'entrée, de sortie et d'oubli, qui régulent le flux d'informations à travers le réseau. Cette architecture permet aux **LSTM** de capturer des dépendances complexes dans les données temporelles, ce qui en fait un outil puissant pour la prédiction de séries chronologiques comme les cours des actions en bourse.

## Schéma d'une unité LSTM :



- La donnée d'entrée  $x_t$  traverse initialement la **Forget gate**  $f_t$  qui décide si la donnée est conservée ( $f_t = 1$ ) ou oubliée ( $f_t = 0$ ). Cela impacte l'état de l'unité précédente  $c_{t-1}$  comme il est multiplié avec le résultat de  $f_t$ .
- La donnée d'entrée  $x_t$  traverse aussi la **Input gate**  $i_t$  qui décide si le **Candidate for cell state update** (candidat pour la mise à jour de l'état de la cellule) est retenu ( $i_t = 1$ ) ou oublié ( $i_t = 0$ ). La mémoire à long terme est donc  $c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t$ .
- La donnée d'entrée  $x_t$  traverse aussi la **Output gate**  $o_t$  qui décide si le **Updated cell state to help determine new hidden state** (mise à jour de l'état de la cellule) est retenu ( $o_t = 1$ ) ou oublié ( $o_t = 0$ ).
- Enfin, les sorties  $c_t$  et  $h_t$  sont passés en entrée à la prochaine cellule.

Ce mécanisme permet donc de conserver des données sur le long terme, un aspect vital aux séries temporelles telles que les données boursières.

## Premières approches

Ayant déjà travaillé avec la rentabilité journalière de titres lors de l'UV IMF, nous avons décidé d'entraîner nos premiers modèles sur ce type de données. La rentabilité journalière correspond au pourcentage d'évolution du cours du jour  $N$  au jour  $N + 1$  en suivant cette formule :

$$\text{rentabilité} = \frac{\text{cours}_{N+1}}{\text{cours}_N} - 1$$

En travaillant avec la rentabilité, nous avons accès à deux informations clés. D'une part, son signe nous indiquait si le cours de clôture ajusté du titre allait évoluer de manière **Bullish** ( $\text{rentabilité} > 0$ ) ou **Bearish** ( $\text{rentabilité} < 0$ ). D'autre part, la valeur de la rentabilité et le cours de clôture actuel nous donnaient le cours de clôture prédit :

$$\text{cours}_{N+1} = (\text{rentabilité} + 1) \times \text{cours}_N$$

Par ailleurs, les valeurs des rentabilités sont souvent comprises dans l'intervalle  $[-1; 1]$ . Cet ordre de grandeur permet d'améliorer la vitesse d'entraînement des réseaux de neurones **LSTM** et l'efficacité de prédiction.

Cependant, c'est en entraînant nos premiers modèles que nous avons remarqué que les résultats étaient très insatisfaisants (notamment sur les métriques de **score R2** et **MSE**). En effet, les valeurs de rentabilité journalières ne traduisaient pas assez de variance afin d'avoir un jeu de données d'entraînement efficace.

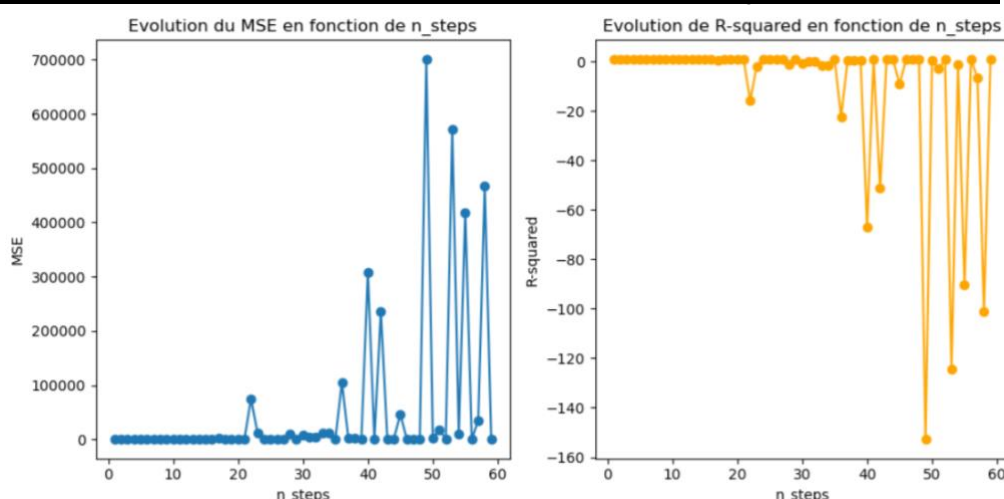
Nous avons donc essayé de mettre en œuvre une autre solution : la classification binaire. Il était possible de labelliser les données en fonction de leur tendance. En effet, un mouvement **Bullish** pourrait être labellisé 1 et un mouvement **Bearish** serait 0. Après la construction du jeu de données labellisées, nous avons testé plusieurs classifieurs (**Random Forest**, **Gradient Boosting**) mais les résultats étaient très mauvais : environ 70% de classification incorrecte. Nous étions donc de retour à la case départ.

En essayant un modèle **LSTM** sur les valeurs du cours de clôture ajusté (et non pas la rentabilité) de l'action LVMH, nous avons remarqué des résultats assez satisfaisants : un score R2 de l'ordre de 0,9.

L'étape suivante était donc l'optimisation du paramétrage de ce modèle. Pour ce faire, nous avons mis en œuvre une méthode vue en cours : tracer l'évolution du score du modèle en fonction de différentes valeurs du paramètre afin de repérer la valeur optimale.

Il était nécessaire d'identifier deux grandeurs importantes. D'une part, le nombre d'unités LSTM à l'intérieur de chaque couche. D'autre part, le nombre de jours passés considérés pour la construction d'un vecteur de données.

### Évolution du MSE et du score R2 en fonction du nombre de jours considérés :



L'allure des courbes ne donnaient aucune indication quant à la valeur optimale du nombre de jours considérés pour un vecteur d'entrée. En traçant les mêmes graphiques pour le nombre d'unités **LSTM**, nous avons aussi observé des résultats non concluants.

La littérature disponible en ligne ne nous indiquait pas de valeur optimale pour ces deux paramètres. Cependant, une majorité des modèles trouvés utilisaient environ 50 unités **LSTM** par couche. Quant au nombre de jours considérés, il variait entre 50 et 100.

Enfin, nous avons fixé le nombre d'unités **LSTM** par couche et le nombre de jours considérés à 50.



Ensuite, un autre problème auquel nous avons fait face est celui de la composition du modèle séquentiel. Tous les modèles prédictifs trouvés en ligne avaient des architectures différentes : ils étaient composés d'une combinaison de couches **LSTM**, **Dropout** et **Dense**.

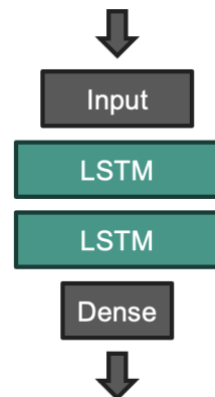
D'une part, le **Dropout** est une technique qui est destinée à empêcher le surapprentissage sur les données d'entraînement en abandonnant des unités dans un réseau de neurones. D'autre part, une couche **Dense** est utilisée afin de modifier la dimension du vecteur produit par la couche précédente.

Contraints par la durée du projet, nous n'étions pas en mesure de tester plusieurs combinaisons. Parallèlement, nous avions un modèle à deux couches **LSTM** qui avait des résultats satisfaisants sur les données LVMH. Nous l'avons donc généralisé à l'ensemble des actions.

Cependant, les résultats de ce dernier après son entraînement sur d'autres actions étaient très insatisfaisants. En effet, comme chaque action avait un comportement différent (plus ou moins complexe), un modèle pouvait donc être très bon pour certaines actions et totalement inefficace pour d'autres.

Finalement, la solution envisagée était d'entraîner plusieurs modèles pour chaque action et procéder à la sélection du modèle le plus performant.

#### Modèle LVMH :



## Évaluation des modèles

Avant d'aborder les différents modèles testés, il est nécessaire de présenter les différentes manières d'évaluer nos modèles.

Premièrement, La **MSE** (Mean Squarred Error) mesure la moyenne des carrés des écarts entre les valeurs réelles et prédites dans un ensemble de données. Elle est calculée comme suit :

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Où :

- $n$  est le nombre d'observations dans l'ensemble de données
- $y_i$  sont les valeurs réelles
- $\hat{y}_i$  sont les valeurs prédites par le modèle

Plus la valeur de la **MSE** est faible, meilleure est la performance du modèle.

Deuxièmement, le **coefficient de détermination**, souvent appelé score **R2**, est une mesure statistique de la proximité des données à la droite de régression ajustée. Il représente la proportion de la variance dans la variable dépendante (cible) qui est expliquée par la variable indépendante (prédicteur). Il est défini comme suit :

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2}$$

Où :

- $n$  est le nombre d'observations dans l'ensemble de données
- $y_i$  sont les valeurs réelles
- $\hat{y}_i$  sont les valeurs prédites par le modèle
- $\bar{y}_i$  est la moyenne des valeurs réelles

Un **R2** proche de 1 indique que le modèle ajuste bien les données, tandis qu'un R2 proche de 0 indique que le modèle n'est pas capable d'expliquer la variance des données.

Finalement, le score de direction évalue la proportion de données qui ont été correctement prédites **Bullish** ou **Bearish**. Il est calculé comme suit :

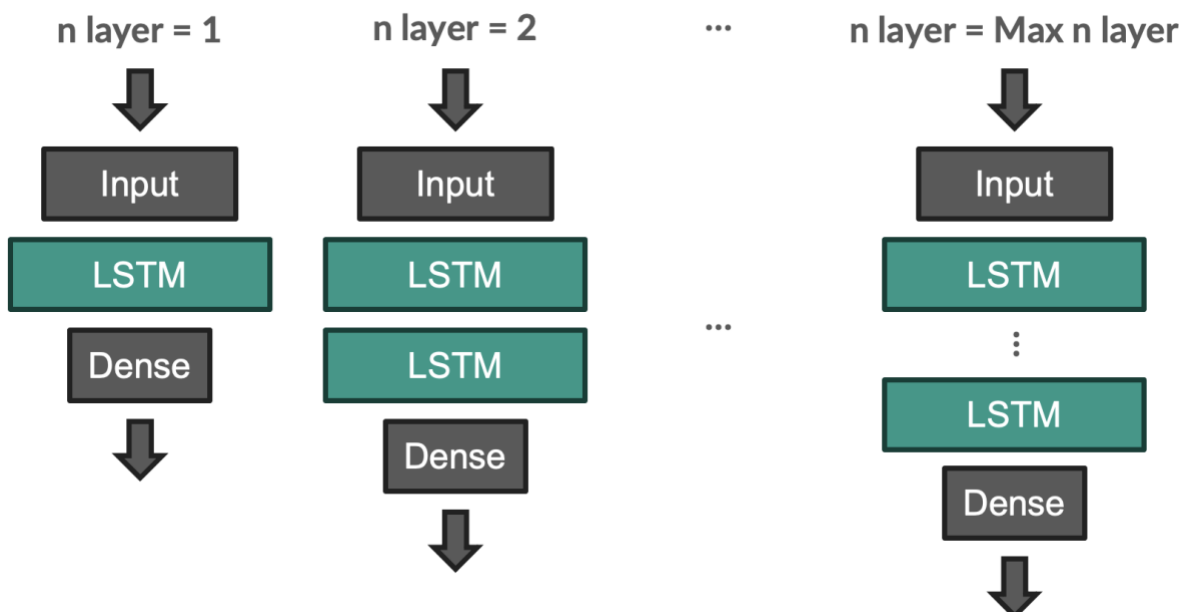
$$direction\ score = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{sign(y_i - y_{i-1}) = sign(\hat{y}_i - y_{i-1})}$$

Un **score de direction** proche de 1 indique que le modèle prédit souvent dans la bonne direction. À l'inverse, un score proche de 0 traduit que le modèle ne prédit presque jamais dans la bonne direction.

## Construction et entraînement des modèles prédictifs à tester

En début d'utilisation, l'utilisateur sera amené à configurer un nombre maximal de couches **LSTM** consécutives à tester  $n\_layers$ . Le programme devra d'abord créer de manière itérative un modèle à  $n$  couches et l'entraîner sur les données. Ensuite, il devra l'évaluer sur les données test.

### Schéma expliquant la construction des modèles de manière itérative :



Enfin, le modèle ayant eu le meilleur score sera sélectionné et sauvegardé pour des utilisations futures (sans besoin de réentraînement). Ce processus est répété pour chaque action considérée.

Nous avons donc entraîné et sélectionné pour chaque titre un modèle prédictif en utilisant comme données d'entrées des vecteurs avec 50 jours passés.

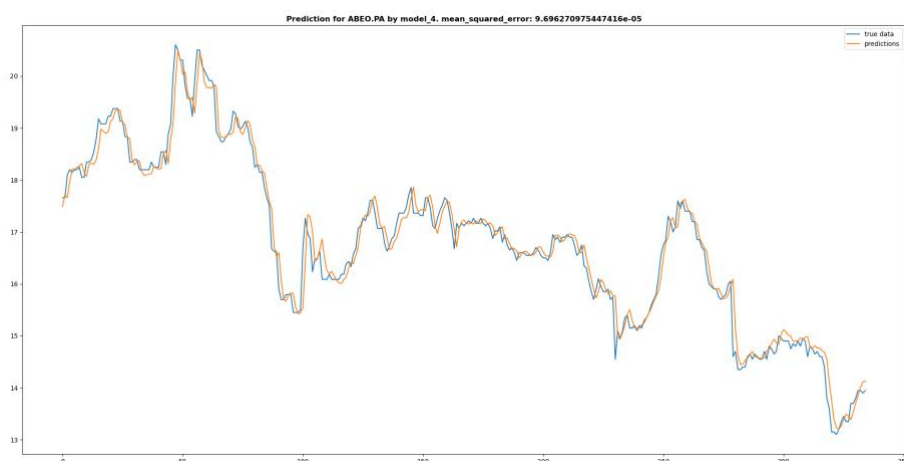
**Vecteur correspondant au jour  $N$  :**

$N - 49$	$N - 48$	$N - 47$	$N - 46$	.....	$N - 3$	$N - 2$	$N - 1$	$N$
----------	----------	----------	----------	-------	---------	---------	---------	-----

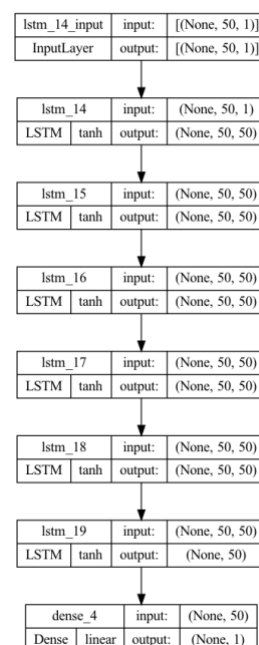
En mettant à l'échelle  $[0; 1]$  les données, cela améliorerait la vitesse d'entraînement et les résultats sur les données tests. En effet, il s'avère que les unités **LSTM** réagissent mieux face aux valeurs comprises entre 0 et 1.

Afin de mieux analyser la structure et la performance des modèles prédictifs, nous avons automatisé la génération de visualisations de l'architecture du modèle et de la prédiction des données test.

**Visualisation de la prédiction des données test**



**Représentation du modèle**

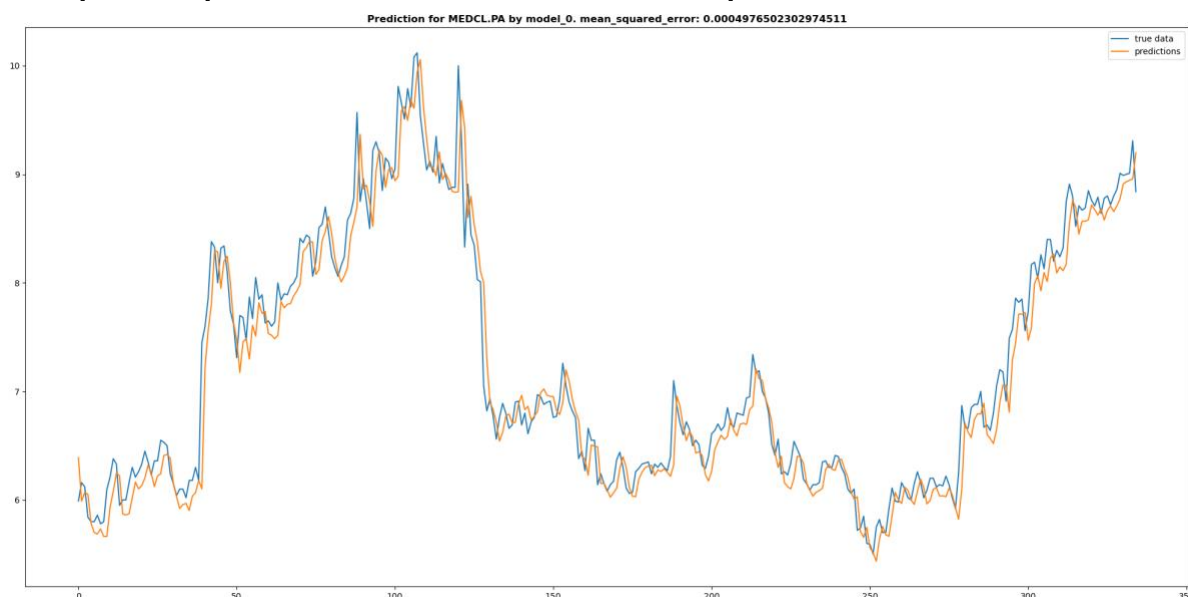


Ces deux graphiques sont donc générés pour chaque modèle entraîné sur environ 240 actions considérées. Cela permet d'analyser visuellement la performance des modèles.

# Critique des résultats

Afin de procéder à une évaluation de la prédiction, il était nécessaire d'analyser les graphes générés par l'entraînement de chaque réseau de neurones **LSTM**.

## Exemple de la prédiction test lors de l'entraînement pour Medincell S.A. :



En regardant ce graphique d'exemple, nous remarquons que l'allure de la **courbe de prédiction** est quasi-identique à celle des **données réelles**. Nous faisons donc face à un modèle prédictif bien entraîné et précis. Cependant, nous remarquons également un retard sur la courbe de prédiction, qui influe la direction de la prédiction.

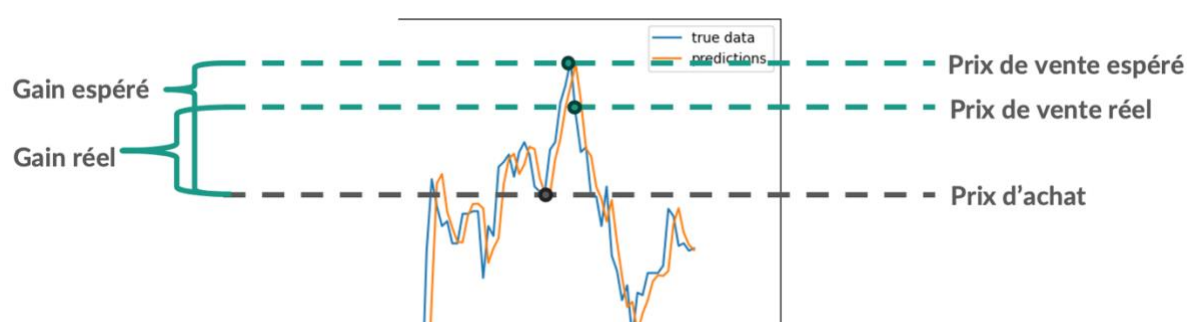
Après analyse de l'entièreté des graphiques générés, nous remarquons que ce retard sur la courbe de prédiction est omniprésent. Ce dernier peut être interprété comme un poids très important sur la valeur du cours au jour  $N$  lors de la prédiction. Cela pourrait dire que les modèles entraînés ne capturent pas les tendances à court terme. Une amélioration pourrait donc être envisageable.

Nous remarquons par contre que même si la prédiction au jour  $N + 1$  est dans la mauvaise direction, celle du jour  $N + 2$  est souvent correcte. En effet, le poids important de la valeur réelle du cours en  $N + 1$  (**Bearish**) corrige souvent le sens la prédiction.

Néanmoins, en ignorant l'origine de ce retard et en se basant sur la stratégie d'investissement explicitée précédemment, il est possible de dire que ce retard est négligeable dans certaines situations.

En effet, en cas d'erreur de prévision (prévision **Bullish** au jour  $N + 1$  alors que la tendance est réellement **Bearish**), cela n'a pas d'impact, car la prédiction du jour  $N + 2$  est souvent correcte (**Bearish**). Certes, on ne vendra pas l'action avant qu'elle ne commence à décliner, mais lors de la vente, on le fera à un cours plus élevé que celui de l'achat, assurant ainsi un bénéfice considérable.

### Explication de la stratégie par rapport au retard :



## Analyse financière et axes d'amélioration

D'une part, si les prédictions sont dans la bonne direction, le programme devrait nous suggérer un portefeuille rentable qui va générer des bénéfices sur le long terme.

D'une autre part, si les prédictions sont dans la mauvaise direction pour le jour  $N$  mais correctes pour le jour  $N + 1$ , il est toujours possible de réaliser des bénéfices, même si ces derniers ne sont pas les plus grands. Dans le cas contraire, le programme suggèrera des actions qui perdront de la valeur. Une amélioration des modèles dans le but d'éliminer ce retard serait intéressante. On pourrait tester l'ajout de couches **Dropout** ou l'entraînement d'un réseau de neurones qui détermine le paramétrage optimal.

Ensuite, en ayant fixé les frais de transaction à 2€, nous nous retrouvons qu'avec des actions à cout élevé (souvent supérieur à 50€). On pourrait donc passer outre certaines actions à cout faible (inférieur à 50€) qui auraient un grand potentiel de gain. Il serait plus intéressant de mettre en œuvre des frais de transaction proportionnelles aux cours des actions.

Par ailleurs, il serait aussi opportun de prédire plus loin que le jour  $N + 1$ . On pourrait envisager de prédire le jour  $N + 5$  avec les données  $[N - 49; N]$  ou de prédire le jour  $N + 2$  en prédisant d'abord le jour  $N + 1$  et ainsi de suite.

Finalement, il est nécessaire de garder en tête que nous ne nous attendons pas à une prédiction réelle et précise de l'évolution de n'importe quelle action. Cela reste impossible comme le marché a un comportement très complexe et souvent imprévisible.

## Conclusion

En conclusion, au cours de ces quatre semaines de projet, nous avons réalisé un travail significatif dans le domaine de la gestion de portefeuille. Notre principal accomplissement réside dans le développement d'un module complet intégrant à la fois l'optimisation de portefeuille et la prédiction des actions en bourse. Grâce à notre travail, nous avons automatisé des processus tels que l'optimisation sous contrainte, l'entraînement de réseaux de neurones et la génération de graphiques.

Ces fonctionnalités combinées offrent une solution complète permettant d'optimiser la constitution d'un portefeuille, de prédire son évolution future, de recommander des ajustements stratégiques et de visualiser de manière intuitive les données, les prédictions et les modèles sous-jacents.

# Bibliographie

## Articles :

1. Chris Gallant, **How to Calculate Gain and Loss on a Stock**, *Investopedia*  
14/02/2024. – [\[Lien article\]](#)
2. Armin Lawi, **Implementation of LSTM and GRUs on grouped time-series data to predict stock prices accurately**, *SpringerOpen*  
07/07/2022 – [\[Lien article\]](#)
3. Wei Hao Khoong, **Why Scaling Your Data Is Important**, *Medium*  
03/01/2023 – [\[Lien article\]](#)
4. Shubhang Khandelwal, **Prediction Stock Prices With LSTM and GRU**, *Medium*  
07/09/2023 – [\[Lien article\]](#)
5. Prajwal Chauhan, **Stock Prediction and Forecasting using LSTM**, *Medium*  
08/07/2023 – [\[Lien article\]](#)
6. ShawnYuShuHearn, **Forecasting Stock Market Indices Prices**, *Medium*  
02/08/2023 – [\[Lien article\]](#)
7. Saul Dobilas, **LSTM Recurrent Neural Networks**, *Medium*  
06/02/2022 – [\[Lien article\]](#)
8. Mahesh, **Mean Squared Error**, *Medium*  
13/05/2023 – [\[Lien article\]](#)
9. Erika D, **Looking at R-Squared**, *Medium*  
13/05/2019 – [\[Lien article\]](#)

## Documentation :

1. **Keras LSTM** – [\[Lien documentation\]](#)
2. **Keras Dense** – [\[Lien documentation\]](#)
3. **yFinance API** – [\[Lien documentation\]](#)
4. **Scikit-Learn** – [\[Lien documentation\]](#)
5. **Pandas** – [\[Lien documentation\]](#)
6. **Numpy** – [\[Lien documentation\]](#)
7. **Joblib** – [\[Lien documentation\]](#)

# Guide d'utilisation

Après avoir explicité le travail effectué, il est maintenant nécessaire d'expliquer comment utiliser le module développé.

## Téléchargement

Pour télécharger tout le module, il faudra exécuter la commande suivante dans un terminal :

```
git clone https://github.com/romainmrاد/UVProjet.git
```

Ensuite, il faudra installer toutes les librairies Python nécessaires avec les commandes suivantes :

### En utilisant pip

```
pip3 install -r requirements.txt
```

### En utilisant anaconda

```
conda install --file requirements.txt
```

## Fonctionnement

### Jour 0

Étapes à réaliser :

- Configurer le portefeuille, la prédiction et le marché dans les fichiers de configuration
- Exécuter le script `portfolio_initialisation.py`
- Exécuter le script `portfolio_prediction.py`

### Jour N

Étapes à réaliser si prédiction en  $N - 1$  est **Bullish** :

- Exécuter le script `portfolio_evaluation.py`
- Exécuter le script `portfolio_prediction.py`

Étapes à réaliser si prédiction en  $N - 1$  est **Bearish** :

- Exécuter le script `portfolio_modification.py`
- Exécuter le script `portfolio_prediction.py`

# Annexes

## Diagrammes UML

- [Diagramme UML Class \*\*Stock\*\*](#)
- [Diagramme UML Class \*\*Portfolio\*\*](#)
- [Diagramme UML Class \*\*Market\*\*](#)
- [Diagramme UML Script \*\*portfolio\\_initialisation.py\*\*](#)
- [Diagramme UML Script \*\*portfolio\\_evaluation.py\*\*](#)
- [Diagramme UML Script \*\*portfolio\\_prediction.py\*\*](#)
- [Diagramme UML Script \*\*train\\_models.py\*\*](#)

## Visualisations

- [Visualisation du portefeuille actuel](#)
- [Visualisation de l'évolution du portefeuille](#)
- [Visualisations des prédictions des données test](#)
- [Visualisations des modèles de prédictions](#)

## Code source

- [Code source des différents objet définis](#)
- [Scripts utilisés](#)



## Configuration

### Configuration du portefeuille

Paramètre	Clé JSON	Valeurs
Nombre d'actions dans le portefeuille	<b>numberOfAssets</b>	Nombre entier
Capital initial pour l'investissement	<b>capital</b>	Nombre décimal
Facteur d'optimisation de portefeuille	<b>optimisationFactor</b>	'sharpeRatio', 'profitability', 'risk'
Proportion minimale par action pour l'optimisation	<b>minimalProportion</b>	Nombre décimal dans l'intervall ]0; 1]
Choix des actions initiales	<b>initialisationMethod</b>	'prediction', 'historical'

### Configuration de la prédiction

Paramètre	Clé JSON	Valeurs
Métrique à évaluer lors du test des différents modèles	<b>predictionMetric</b>	'r2_score', 'direction', 'mean_squared_error'
Nombre maximal de couches LSTM à tester pour les modèles	<b>maxLayers</b>	Nombre entier
Nombre de jours passés à considérer pour la prédiction	<b>lookback</b>	Nombre entier

### Configuration du marché

Paramètre	Clé JSON	Valeurs
Période à considérer pour télécharger les données historiques d'une action	<b>period</b>	'1d', '5d', '1mo', '3mo', '6mo', '1y', '2y', '5y', '10y', 'ytd', 'max'
Taux sans risque	<b>riskFreeRate</b>	Nombre décimal dans l'intervall ]0; 1]
Frais de transaction	<b>tradingFee</b>	Nombre entier ou décimal