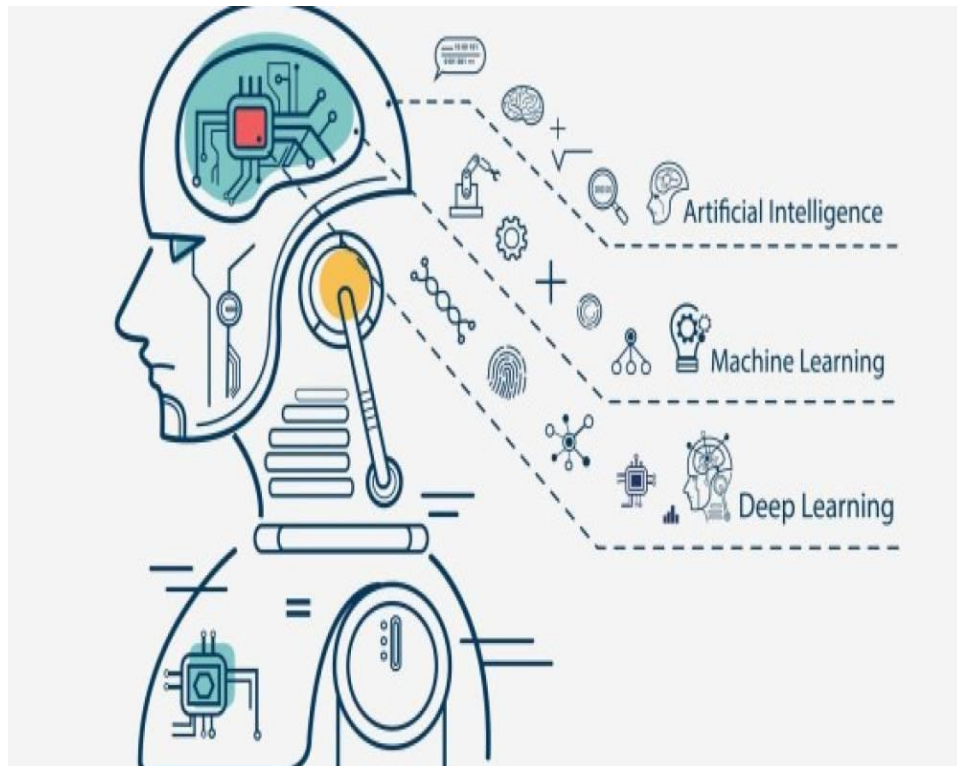


Long Short-Term Memory (LSTM) pour la prédiction du prix d'un actif

Introduction

Les Réseaux de Neurones LSTM offrent une méthode sophistiquée pour anticiper les fluctuations des prix d'actifs financiers en modélisant efficacement les séries temporelles. Leur capacité à capturer les tendances complexes fait d'eux un outil puissant pour les analyses prédictives dans le domaine financier



Objectif

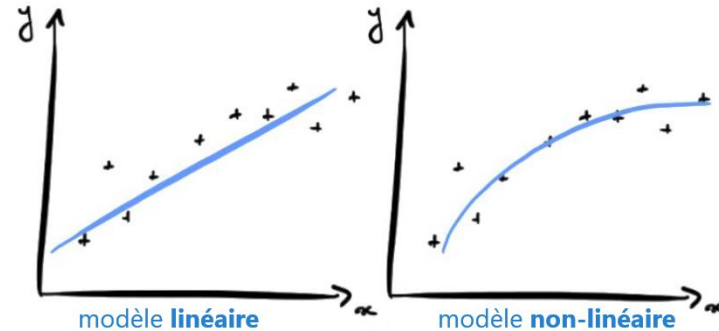
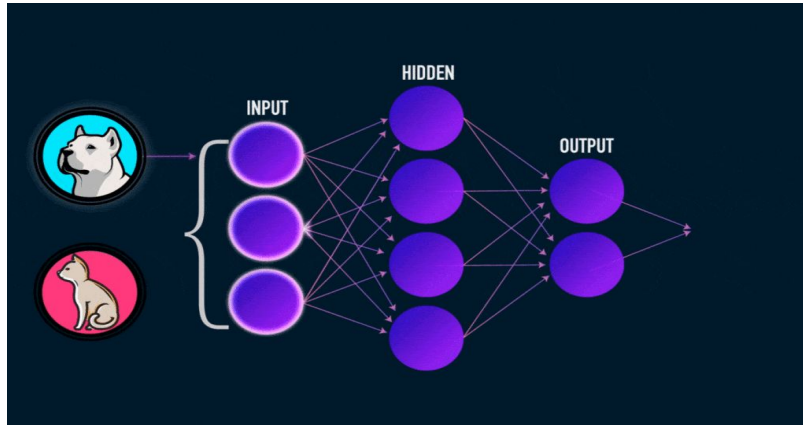
Création d'un modèle de prédiction de l'évolution du prix d'un actif



Plan

- Définition du Deep Learning et des Séries Temporelles
- Les Réseaux Neuronaux Récurrents (RNN)
- LSTM
- Modèle de Prédiction
- Axes d'amélioration

Qu'est ce que le Deep Learning?



Pour illustrer le concept, prenons un problème de classification entre chien et chat à partir d'image. Lors de l'apprentissage, l'algorithme va ajuster les poids des neurones de façon à diminuer l'écart entre les résultats obtenus et les résultats attendus. Le modèle pourra apprendre à détecter les triangles dans une image puisque les chats ont des oreilles beaucoup plus triangulaires que les chiens.

Deep Learning vs Machine learning

Machine learning vs. deep learning		
	MACHINE LEARNING	DEEP LEARNING
Optimal data volumes	Thousands of data points	Big data: millions of data points
Outputs	Numerical value, like a classification or score	Anything from numerical values to free-form elements, like free text and sound
How it works	Uses various types of automated algorithms that learn to model functions and predict future actions from data	Uses neural networks that pass data through many processing layers to interpret data features and relationships
How it's managed	Algorithms are directed by data analysts to examine specific variables in data sets	Algorithms are largely self-directed on data analysis once they're put into production

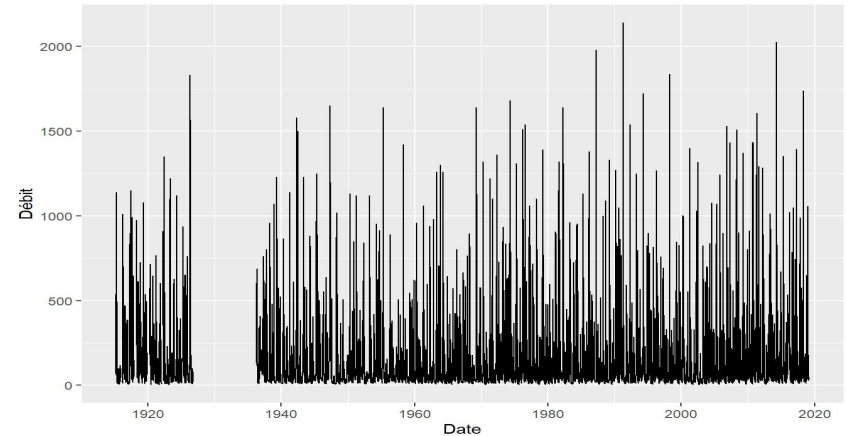
©2017 TECHTARGET. ALL RIGHTS RESERVED. TechTarget

Qu'est ce qu'une série temporelle?

Les séries temporelles couvrent un large éventail de phénomènes de la vie réelle et se retrouvent dans de nombreux domaines.

Mathématiquement une série temporelle c'est une série de données indexée par le temps.

L'analyse et la prédiction de ces séries sont donc d'un intérêt primordial pour certaines industries ou secteurs d'activités car concrètement prédire une série temporelle c'est prédire le futur.



Comment se décompose une série temporelle?

Traditionnellement, une série temporelle est décomposée selon trois éléments :

- Une tendance (T_t): La tendance correspond à un comportement croissant ou décroissant d'une série au cours du temps
- Une saisonnalité (S_t): La saisonnalité reflète la présence d'un phénomène périodique qui se répète au long de la série temporelle.
- Un résidu ou erreur (ϵ_t)

Mathématiquement, on peut donc traduire une série temporelle par $X_t = T_t + S_t + \epsilon_t$

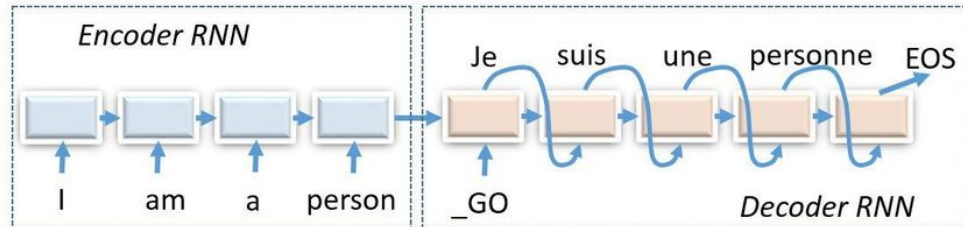
Les Réseaux Neuraux Récurrents (RNN)

Qu'est ce qu'un RNN?

- Réseau de neurones artificiels adapté à l'analyse de données séquentielles.
- Différent des réseaux de neurones classiques car les RNN possèdent des connexions récurrentes -> prise en compte des informations passées lors du traitement des données.

Applications des RNN

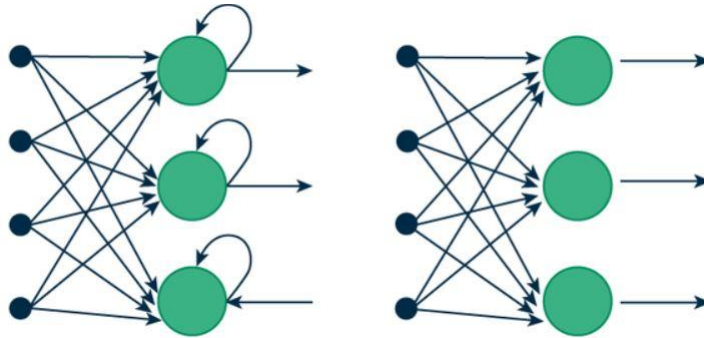
- Reconnaissance de la parole
- Traduction automatique
- Analyse de sentiments
- Prévion de séries temporelles



Les Réseaux Neuronaux Récurrents (RNN)

Structure d'un RNN

Constitué de cellules récurrentes qui prennent en entrée à la fois les données d'entrée à un instant donné et l'état caché de la cellule à l'instant précédent.



A gauche un RNN avec les cellules récurrentes

A droite un réseau de neurones classique

Les Réseaux Neuraux Récurrents (RNN)

Fonctionnement d'un RNN

Propagation avant (forward propagation) :

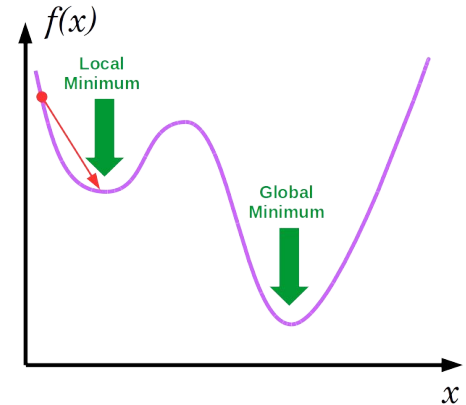
- À chaque pas de temps, les données d'entrée sont combinées avec l'état caché de la cellule précédente.
- La sortie est soumise à une fonction d'activation non linéaire pour introduire de la complexité.
- Exemples de fonctions d'activation : sigmoïde, tangente hyperbolique.

Rétropropagation du gradient (backpropagation) :

- Calcul de l'erreur entre la sortie prédite et les vraies données (fonction de perte, comme l'erreur quadratique moyenne)
- Les poids sont ajustés pour minimiser l'erreur (algorithmes d'optimisation comme la descente de gradient stochastique)

Propagation arrière dans le temps :

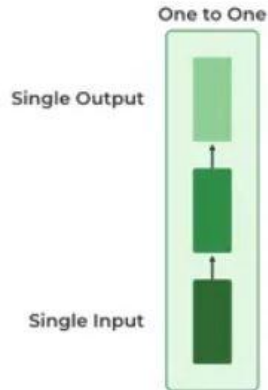
- Les RNN possédant des connexions récurrentes, le calcul du gradient nécessite de remonter dans le temps à travers toutes les étapes de la séquence.
- Cela peut poser des problèmes d'explosion de gradients, qui peuvent être atténués par des techniques comme le gradient tronqué ou le gradient clipping.



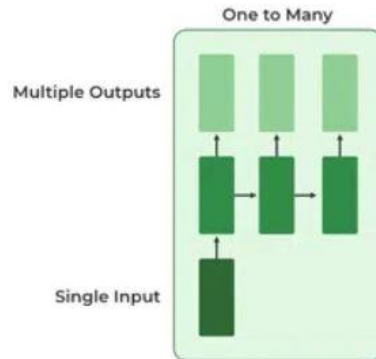
Les Réseaux Neuronaux Récurents (RNN)

Types de RNN

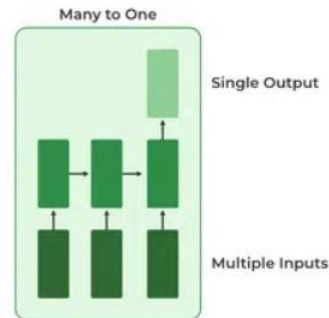
Un par un : une seule entrée et une seule sortie.



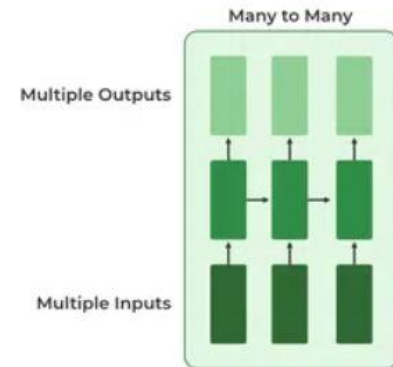
Un à plusieurs : une entrée et de nombreuses sorties



Plusieurs contre un : nombreuses entrées générant une seule sortie



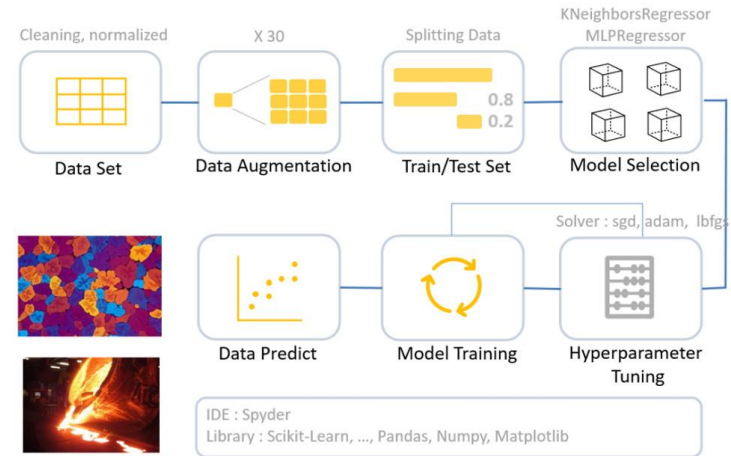
Plusieurs à plusieurs : plusieurs entrées et plusieurs sorties



Les Réseaux Neuronaux Récurrents (RNN)

Entraînement des RNN

- **Prétraitement des données** : Normalisation, padding, etc.
- **Initialisation des poids** : Des méthodes d'initialisation appropriées sont nécessaires pour éviter les problèmes de convergence
- **Choix de la fonction de perte** : MSE ou MAE par exemple
- **Choix de l'algorithme d'optimisation** : Optimisation avec moment ou l'optimisation adaptative
- **Réduction du surapprentissage** : Dropout ou régularisation L2.
- **Validation croisée** : Evaluation de la performance du modèle sur des données non vues.
- **Tuning des hyperparamètres** : Optimisation des paramètres du modèle pour obtenir de meilleures performances.



Les Réseaux Neuronaux Récurents (RNN)

Limites des RNN

- **Dépendance à long terme difficile à apprendre** : Difficulté à capturer les dépendances à long terme dans les données.
- **Calculs coûteux** : Le calcul du gradient sur de longues séquences peut être intensif en termes de ressources.
- **Sensibilité au choix de l'architecture** : Le choix de l'architecture (nombre de couches, nombre d'unités, etc.) peut avoir un impact significatif sur les performances du modèle.

LSTM

Qu'est ce qu'un LSTM?

Les LSTM (Long Short-Term Memory) sont un type spécifique de réseau de neurones récurrents (RNN)

Objectif : Régler le problème de "disparition du gradient" qui peut survenir lorsqu'on utilise les réseaux de neurones récurrents et qui cause un problème de dépendance à long terme

La disparition du gradient se produit lorsque les gradients, qui indiquent la pente de la fonction de perte par rapport aux poids du réseau, deviennent très petits à mesure qu'ils sont transmis des couches de sortie vers les couches initiales lors de la rétropropagation.

Cela entraîne une mise à jour très lente ou même l'absence de mise à jour des poids dans les couches initiales, rendant l'apprentissage inefficace voire impossible dans les parties profondes du réseau.

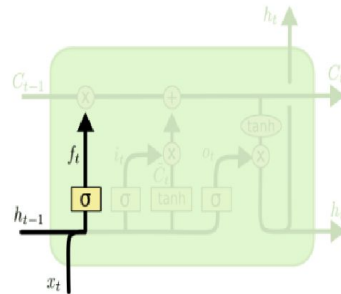
LSTM

Le principe

Les LSTM reposent sur le principe d'unités récurrentes à porte (GRU)

Ces portes contrôlent le flux d'information à travers le réseau. Il en existe trois :

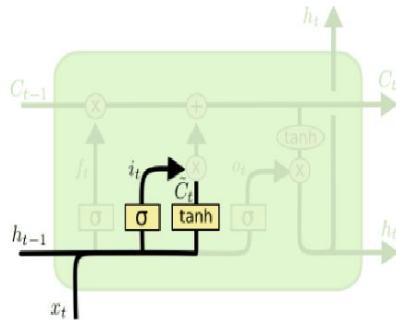
- Portes d'oubli (forget gates),



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

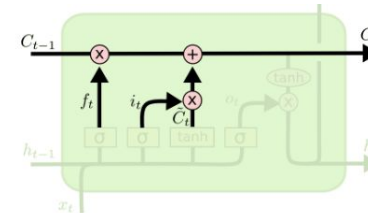
LSTM

- Portes d'entrée (Input gates),



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

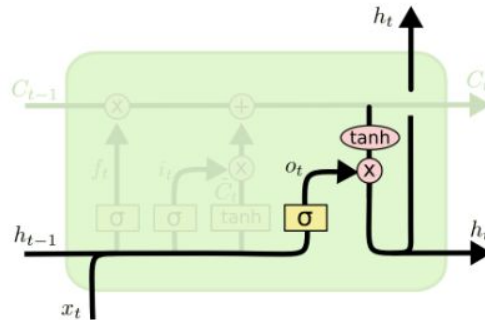
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

LSTM

- Portes de sortie (Output gates),



$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

LSTM

On a aussi d'autres composantes :

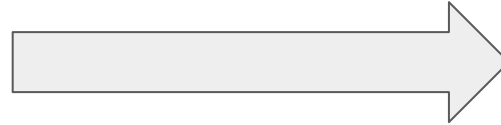
- Cellule mémoire candidate
- Cellule mémoire
- État caché

L'objectif du modèle

Objectif : Fournir le prix d'un actif à J+1**Entrées**

Indicateurs :

- RSI
- EMAF (Fast moving average)
- EMAM (Medium moving average)
- EMAS (Slow moving average)
- Prix antérieurs
- ...

**Sortie**

Prix de l'actif à J+1

Fonctionnement du modèle

Préparation des données :

Création de deux variables cibles :

1.Target : Différence entre le prix d'ouverture actuel et le prix à la clôture future.

2.TargetCloseNext : Le close price à J+1

Target Class : Variable de classification indiquant si le prix augmente ou diminue (1 ou 0).

Normalisation des données

- Normalisation des prix des actifs financiers.
- Normalisation des indicateurs techniques.
- Normalisation de la variable binaire de tendance.

Le dataset normalisé devient un np array de 2 dimensions, avec des valeurs entre 0 et 1.

Fonctionnement du modèle

Sélection des features :

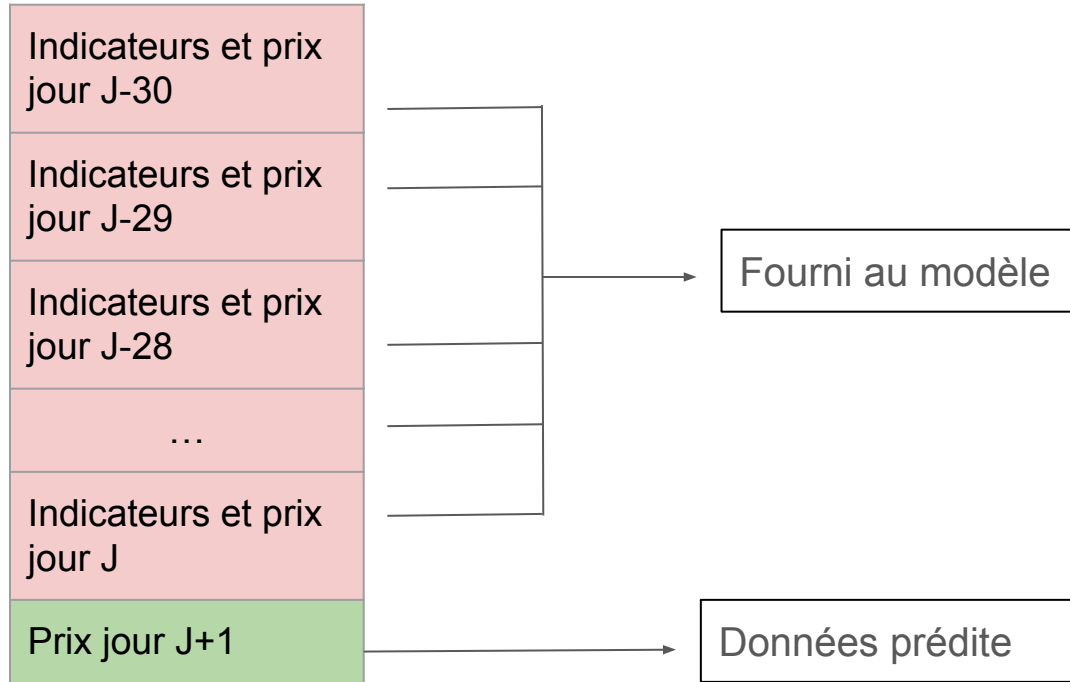
- On conserve les colonnes correspondant aux prix et aux indicateurs (8 premières colonnes)



Entrée du modèle

On split alors le dataset en 20% pour training et 80% de test.

Entrainement du modèle



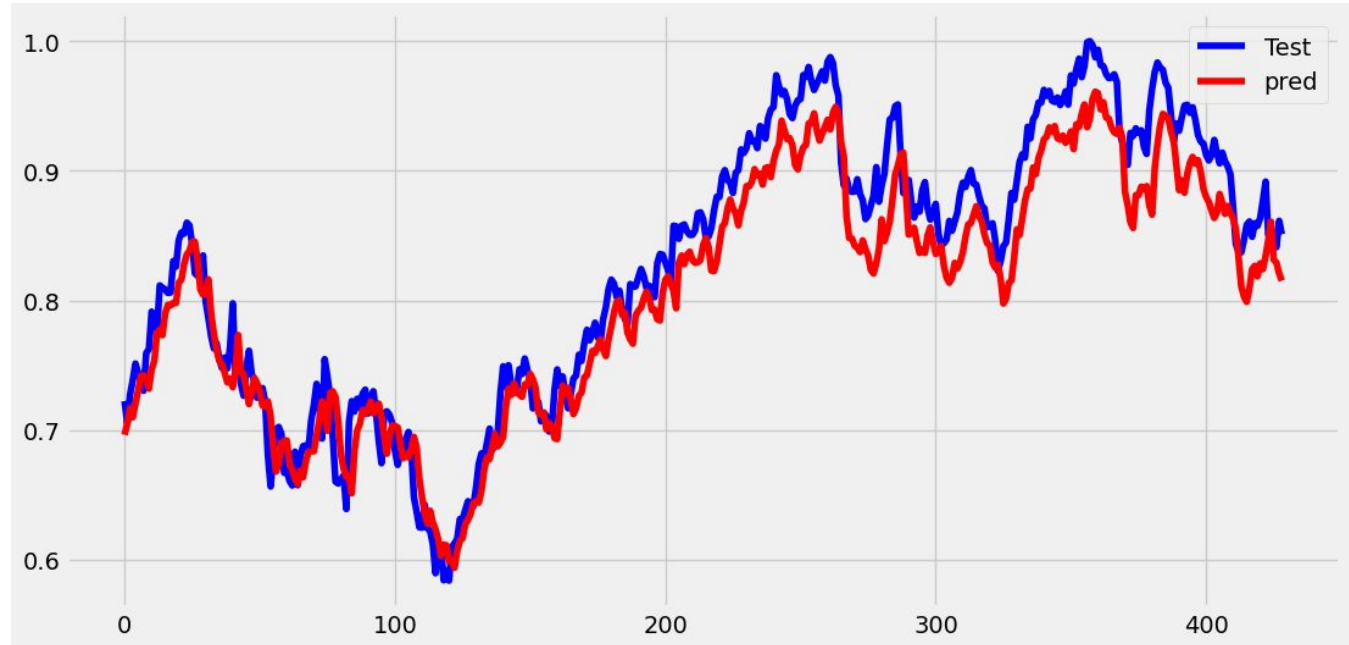
Entraînement du modèle

Date	Open	High	Low	Close	Adj Close
2012-03-12	759.429993	760.619995	756.840027	759.130005	759.130005
2012-03-13	761.450012	773.030029	761.450012	772.979980	772.979980
2012-03-14	773.190002	774.679993	769.280029	771.630005	771.630005
2012-03-15	772.380005	776.469971	770.739990	776.419983	776.419983
2012-03-16	776.559998	778.099976	775.640015	777.130005	777.130005
2012-03-19	777.119995	782.549988	776.070007	780.090027	780.090027
2012-03-20	778.030029	778.299988	773.119995	777.369995	777.369995

- Splitter le dataset on va
- Entraîner le modèle :
 - en lui donnant les 30 derniers jours de données
 - Prédire le prix du prochain jour et ainsi de suite jusqu' à la fin du dataset

Test du modèle

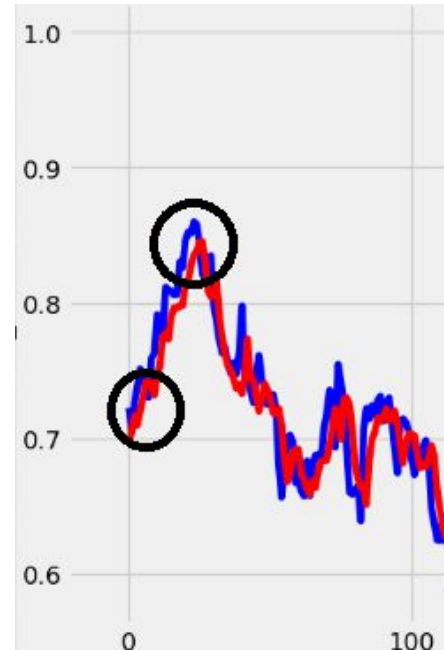
On observe que le modèle semble donner de bonnes estimations quant aux prix. Même si il existe quand même une différence entre les prédictions journalières et le valeurs réelles.



Limites du modèle

Limites :

Le modèle LSTM a tendance à prédire des valeurs similaires au prix actuel, ce qui peut donner l'impression de simplement répliquer la courbe des prix précédente.



Axes d'amélioration

Axes d'amélioration :

- Ajouter des couches et des neurones pour une meilleure précision.
- Intégrer ou améliorer les indicateurs techniques.
- Déterminer le nombre optimal de couches et de nœuds.
- Trouver le nombre idéal de jours précédents pour les données d'entrée.
- Explorer d'autres variables cibles pour améliorer la prédiction.



**Merci de votre
attention !**

