Sommaire

Abstract

Cryptocurrencies are becoming increasingly popular and especially the most famous of them: the Bitcoin. The currencies have the particularity to be highly volatile which proves to be a challenge for all the investors. Many made their entire wealth betting on Bitcoin while many others lost theirs doing the exact same thing.

The price behavior of cryptocurrencies is still largely unexplored. As such, this problem attracts economists and researchers alike due to its potential. Indeed, the few previous methods used to predict the prices of other markets cannot be applied to the cryptocurrency market because of its numerous peculiarities (such as social media influence).

In this study, we discuss how to implement artificial intelligence to predict the price of the Bitcoin and more precisely for the thirty next days.

In the first place, we compare various algorithms and neural networks based on previous studies to determine which ones are the most likely to produce accurate predictions. During this endeavor, we consider different metrics such as MAE (Mean absolute error), RMSE (Root mean square error), MAPE (Mean absolute percentage error) and Accuracy.

As a result, we selected three algorithms: SLR (Simple Linear Regression), MLR (Multiple Linear Regression) and RNN LSTM (Recurrent Neural Network Long-Short-Term Memory).

The SLR produced subpar predictions due to its inability to model multiples trend changes as well as rapid increases and decreases in price after a long period of stability. The growth of the Bitcoin depends on many factors and thus cannot be predicted only with one of them.

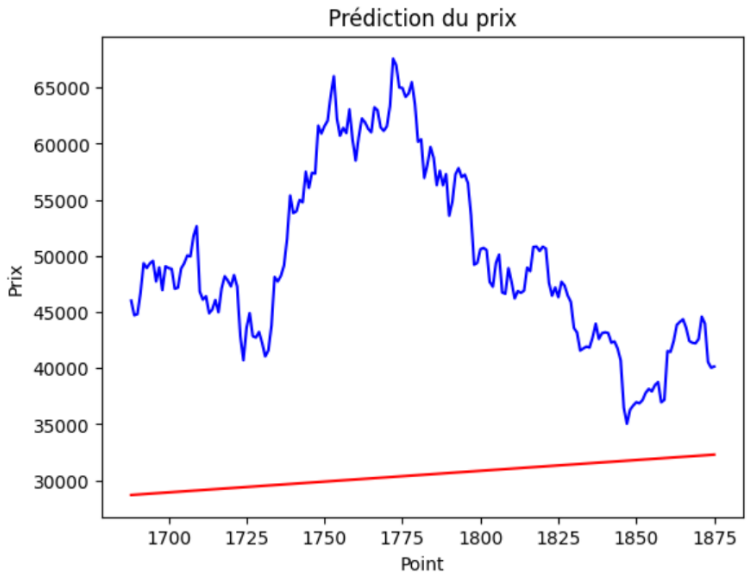

Table 1

On the other hand, the MLR uses multiple features and was therefore able to produce accurate results. It managed to detect the fluctuations in the price and even to predict the actual price except during extreme crashes. The features we chose are the volume of tweets talking about the Bitcoin, the attention on Google Trends and the price of gold in the thirty last days together with the close price of the Bitcoin the previous day.

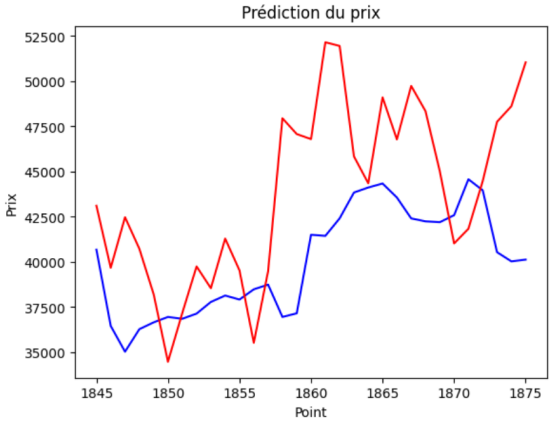

Table 2

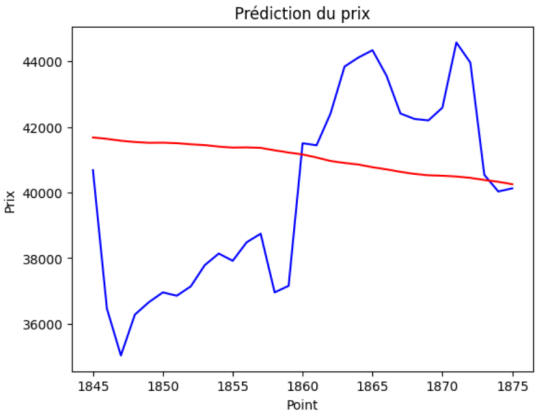

Table 3

The last algorithm we tried to implement, the LSTM, didn't provide coherent results. We were unable to implement this algorithm due to its complexity as well as our lack of data.

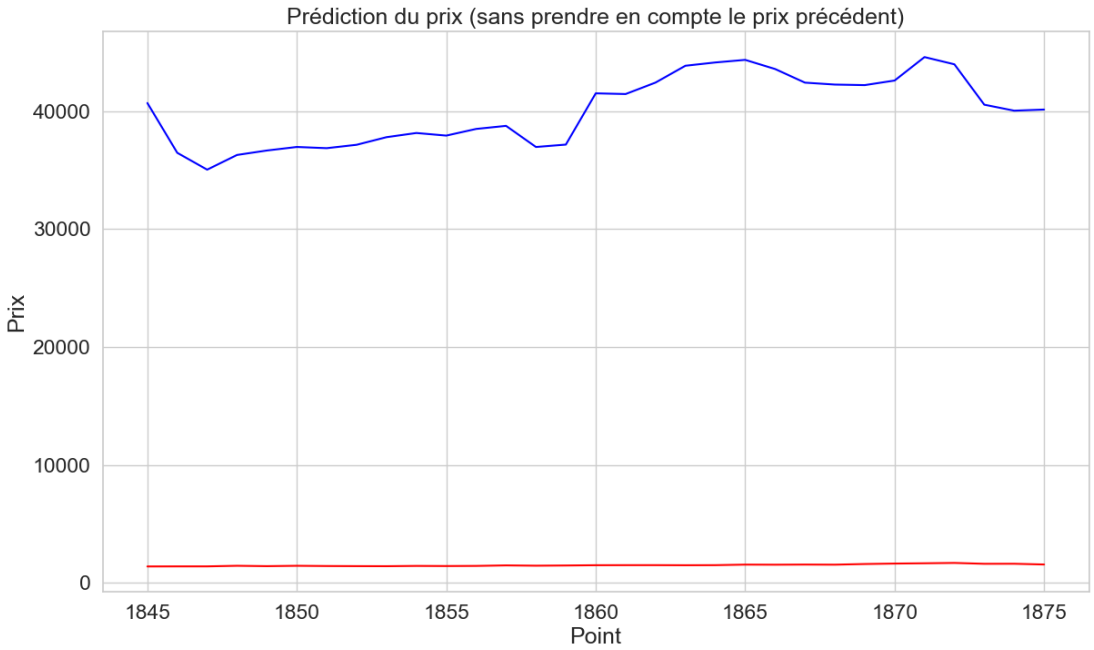

Table 4

In conclusion, the SLR cannot be used to predict Bitcoin market prices while the MLR produces interesting results even with lacking data. We could not determine if the LSTM is able to perform well in this case. Further study is required to get a definitive result.

Introduction

Les cryptomonnaies sont des monnaies virtuelles échangées sur internet qui permettent d'acheter des biens ou des services sans nécessité de banque centrale. Elles utilisent à la place un réseau informatique décentralisé pour vérifier l'authenticité des transactions.

Le Bitcoin est la cryptomonnaie la plus utilisée dans le monde et domine actuellement le marché des cryptomonnaies, représentant plus de 40% de celui-ci. Il a été inventé par une ou plusieurs personnes utilisant le pseudonyme Satoshi Nakamoto et mis en service en 2009. Cependant, il n'a commencé à être véritablement utilisé par le public qu'en 2013. Par conséquent, il est encore très récent sur les marchés et encore plus sur celui des monnaies. L'évolution de son prix reste donc indéterminée, ce qui offre de nouvelles opportunités de recherche non seulement pour les chercheurs mais aussi pour les économistes.

Bien que le Bitcoin soit la cryptomonnaie la plus importante, il en existe de nombreuses autres dont Ethereum et XRP. Ces dernières sont plus récentes que le Bitcoin mais figurent néanmoins parmi les 10 cryptomonnaies ayant les plus grandes parts du marché des cryptomonnaies. Nous avons choisi ces cryptomonnaies en plus du Bitcoin car elles ont la particularité d'être beaucoup plus stables que la grande majorité de leurs concurrentes.

Cependant, même les cryptomonnaies les plus stables sont hautement volatiles, bien plus que les monnaies traditionnelles. La volatilité inhérente à ce marché complique fortement les prédictions et plus particulièrement à long terme. Mais celle-ci offre aussi des avantages certains. En effet, elle permet aux investisseurs de réaliser des bénéfices beaucoup plus conséquents que sur des marchés matures (ou des pertes tout aussi conséquentes).

Par conséquent, la prédiction des prix des cryptomonnaies permettrait de limiter fortement les risques que prennent les investisseurs tout en augmentant leurs bénéfices si elle s'avérait précise.

Durant cette étude, nous essaierons donc de prévoir le cours du Bitcoin en utilisant divers algorithmes d'intelligence artificielle. Nous chercherons d'abord à déterminer quels sont les algorithmes les plus adaptés pour réaliser cela en nous appuyant sur les travaux existants. Nous étudierons ensuite chacun de ces algorithmes ainsi que les résultats obtenus dans différentes études, selon les paramètres utilisés. Enfin, nous mettrons en place ces algorithmes dans une situation pratique et ajusterons les paramètres selon nos recherches.

Notre étude ayant pour but de permettre aux investisseurs de prédire les variations de prix des cryptomonnaies, nous chercherons principalement à prédire les hausses et les baisses des prix plutôt que le prix exact de chaque cryptomonnaie.

Choix des algorithmes

Présentation

Le machine learning se divise en deux catégories principales d'algorithmes : supervisés et non supervisés.

L'apprentissage supervisé consiste à entraîner un modèle à partir d'exemples étiquetés (ou labelisés), où chaque exemple est associé à une étiquette ou à une classe connue. L'objectif de l'apprentissage supervisé est de permettre au modèle d'apprendre à prédire l'étiquette ou la classe de nouveaux exemples non étiquetés. Lorsque les données d'entrée sont alimentées dans le modèle, celui-ci ajuste ses poids jusqu'à ce que le modèle soit viable.

L'apprentissage non supervisé implique la recherche de structures, de modèles et de relations cachées dans des données sans étiquette ou aide extérieure préalable. Il diffère de l'apprentissage supervisé, où le modèle est fourni avec des exemples accompagnés de leurs étiquettes correspondantes pour apprendre à prédire des résultats spécifiques, l'apprentissage non supervisé cherche à trouver des informations cachées et des structures inhérentes, le tout sans connaître dès le départ les catégories ou les étiquettes des exemples.

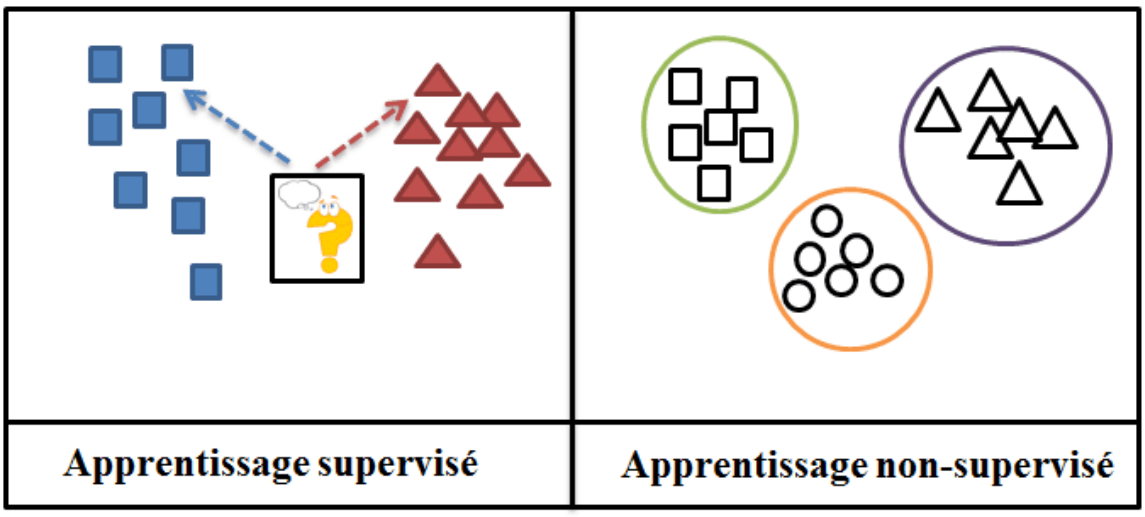

Figure 1

Pour notre cas de prévision des cours des marchés d'une cryptomonnaie, de nombreux algorithmes peuvent être utilisés. Cependant nous allons nous baser sur quelques études démontrant les performances de differents algorithmes de machine learning pour selectionner ceux que nous allons approfondir par la suite.

Recherche d'un ou plusieurs algorithmes

Dans une étude de Juillet 2019 publié par Suhwan Ji, Jongmin Kim et Hyeonseung Im, les performances de nombreux modèles ont été étudiées, autant en classification qu'en régression mais aussi avec différentes couches pour chaque modèle ainsi que des combinaisons entre différents modèles pour balayer un maximum de possibilités.

Deep Neural Networks

Le premier modèle à être étudié est le Deep Neural Network, qui consiste en une ou plusieurs couches d'entrée, de multiples couches cachées et une couche de sortie.

Tous les neurones d'une couche sont connectés à la couche suivante.

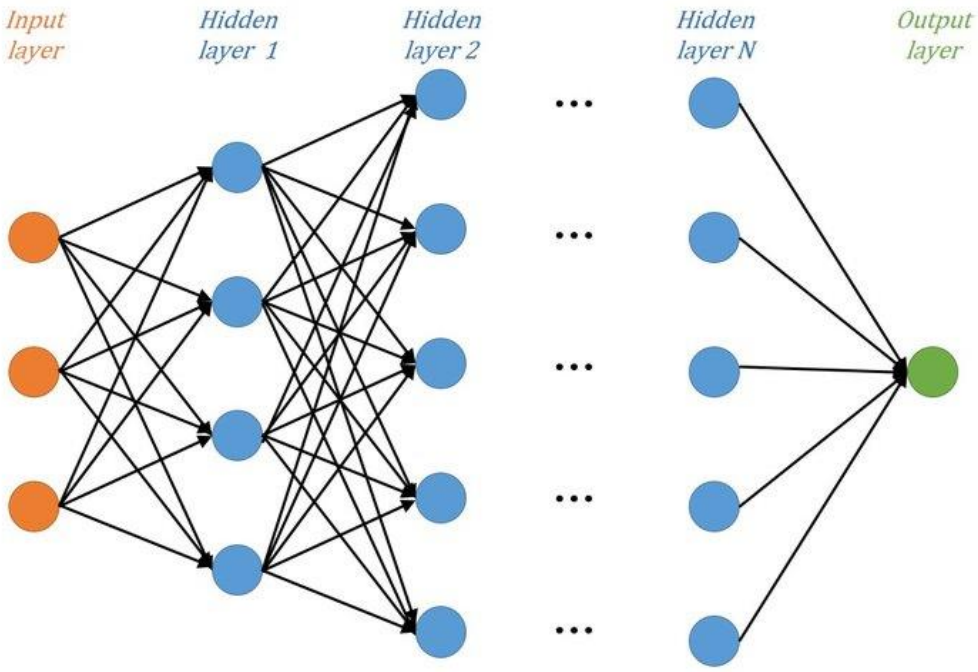

Figure 2

Recurrent Neural Networks et Long Short-Term Memory

Les réseaux neuronaux récurrents (RNN) et Long Short-Term Memory (LSTM) ont également été analysés dans l'étude. Les RNN sont particulièrement adaptés aux données séquentielles, ce qui est courant dans les prévisions de prix de cryptomonnaie. Ils sont capables de capturer des dépendances temporelles dans les données. Cependant, ils ont tendance à oublier les informations à long terme, un problème qui a été résolu avec l'introduction des LSTM. Ces derniers, une variante des RNN, sont capables de capturer et de retenir des informations sur de longues périodes, ce qui peut être extrêmement bénéfique pour prédire les tendances à long terme des prix des cryptomonnaies.

Convolutional Neural Networks

Les réseaux neuronaux convolutionnels (CNN) sont souvent mis à profit pour le traitement d'images, car ils sont aptes à discerner des particularités locales et spatiales. On leur a trouvé des applications fructueuses dans les analyses de séries temporelles en considérant celles-ci comme des images à dimension unique. Les CNN sont en mesure de saisir des spécificités régionales et des tendances dans les renseignements qui pourraient s'avérer utiles pour prédire les fluctuations à court terme.

Deep Residual Networks

Les réseaux neuronaux résiduels profonds (Deep Residual Networks, ou ResNets) sont une amélioration des CNN. Ils utilisent des connexions résiduelles ou des "sautes" pour aider à former des réseaux neuronaux profonds. Ces connexions résiduelles aident à combattre le problème de la disparition du gradient, ce qui rend difficile l'entraînement de réseaux neuronaux profonds.

Combinaisons de CNNs et RNNs

En mélangeant les points forts des CNN (bon pour analyser les structures) et des RNN (bon pour analyser les séquences temporelles), ces chercheurs ont crée un modèle appelé CRNN. Le même ensemble de données est transmis à un CNN et à un LSTM. Le CNN fait une convolution 2D et normalise les données, tandis que le LSTM applique une technique appelée "dropout". Ensuite, les résultats des deux modèles sont fusionnés en un seul vecteur qui est transformé en une seule prédiction par une couche entièrement connectée. Plusieurs manières de fusionner les résultats ont été testées, et la soustraction a fonctionné le mieux pour les problèmes de régression et la concaténation pour les problèmes de classification.

Ensemble de modèles

L'idée derrière les modèles d'ensemble est de combiner plusieurs modèles "faibles" pour créer un modèle "fort" avec une meilleure précision de prédiction. Dans cette étude, un modèle d'ensemble est construit en deux étapes. D'abord, on entraîne trois modèles de base (DNN, LSTM et CNN) avec un premier ensemble de données. Ensuite, ces modèles font des prédictions sur un deuxième ensemble de données. Un quatrième modèle, le DNN "meta", est alors entraîné pour faire des prédictions finales basées sur les prédictions des trois modèles de base.

Résultats de l'étude

Cette étude a examiné différents modèles de Deep Learning pour prédire le mouvement des prix du Bitcoin. Les modèles comparés inclus des réseaux de neurones profonds (DNN), des modèles à longue mémoire à court terme (LSTM), des réseaux de neurones convolutionnels (CNN) et des réseaux résiduels profonds (ResNet), ainsi que plusieurs mélanges de ces modèles.

Chacun de ces modèles a été testé selon deux approches différentes : les problèmes de régression et de classification. La régression consiste à prévoir le prix futur du Bitcoin en se basant sur des données passées. Elle permet d'établir une prévision précise du prix, ce qui peut être particulièrement utile pour les stratégies d'investissement à court terme.

D'autre part, la classification vise à prédire l'évolution du prix, soit à la hausse soit à la baisse. Cela pourrait ne pas donner une prédiction précise du prix, mais offre une indication générale de la direction que le prix pourrait prendre. Cette information est particulièrement utile pour les décisions d'investissement à long terme.

Les résultats de cette étude montrent que les modèles LSTM et SVM sont légèrement supérieurs aux autres pour les tâches de régression. Autrement dit, lorsqu'on essaie de prédire avec exactitude le futur prix du Bitcoin, les modèles LSTM et SVM génèrent des résultats légèrement plus précis que les autres solutions.

Pour ce qui est des problèmes de classification, le modèle DNN a démontré une performance légèrement supérieure. Cela signifie que le DNN a été légèrement plus précis dans la prédiction de l'orientation générale de l'évolution du prix du Bitcoin.

Il convient de souligner que bien que ces différences de performance aient été observées, elles étaient mineures. Ainsi, bien que le LSTM, SVM et le DNN aient été légèrement supérieurs dans leurs domaines respectifs, tous les modèles évalués ont globalement

démontré une performance comparable. Ainsi, le choix du modèle à adopter pourrait être influencé par divers facteurs, notamment la complexité des données à traiter ou l'aisance d'implémentation du modèle. Un modèle plus complexe pourrait être nécessaire pour traiter des données complexes et capter des tendances non linéaires.

	DNN	LSTM	CNN	ResNet	CRNN	Ensemble	SVM	Base	Random
regression	6755.55	8806.72	6616.87	7608.35	8102.71	5772.99	**9842.95**	–	–
classification	**10877.07**	**10359.42**	**10422.19**	**10619.98**	**10315.18**	**10432.44**	9532.43	**9532.43**	**9918.70**

Figure 3

Gain en partant d'un portefeuille de 10 000$ sur les 20 derniers jours pour la régression et les 50 derniers jours pour la classification

Une seconde étude publiée en juillet 2020 par Nicola Uras, Lodovica Marchesi, Michele Marchesi, et Roberto Tonelli a étudié les régressions linéaires simple (SLR), les régressions linéaires multiples (MLR), et le Long Short-Term Memory (LSTM).

Ici, deux nouveaux modèles ont été évalué en plus du LSTM qui a été vu précédemment.

Régressions Linéaires

La régression linéaire simple est une méthode de Machine Learning, qui recherche une relation entre la donnée d'entrée (variable indépendante) et de sortie (variable dépendante).

Cet algorithme cherche à définir la courbe qui représente au mieux la tendance des points.

La régression linéaire multiple est une extension de la régression linéaire simple qui accepte plusieurs variables indépendantes. Elle est utilisée quand plusieurs facteurs pourraient influencer la variable dépendante, permettant de capturer des relations plus complexes dans les données.

Long Short-Term Memory

Le LSTM dans cette étude reprend les principes discutés précédemment, avec ici une distinction entre les Univariate LSTM et Multivariate LSTM. Un LSTM univarié est un LSTM qui ne tient compte que d'une seule caractéristique d'entrée, dans notre cas, la variable « close ».

Toutefois, l'étude a révélé que de meilleurs résultats étaient obtenus en incluant une deuxième caractéristique, la variable « volume ». C'est ce que l'on appelle un LSTM multivarié, un LSTM possédant plusieurs caractéristiques au niveau de la couche d'entrée.

Résultats de l'étude

Les résultats obtenus lors de cette étude sont très satisfaisants, de nombreux paramètres ont été testés notamment le lag avec différentes valeurs.

Il a été montré que la régression linéaire simple, la régression linéaire multiple, le LSTM Univariate et le le LSTM Multivariate sont tous les quatre viables avec une erreur absolue moyenne en pourcentage (MAPE) de 0.007 et une racine de l'erreur quadratique moyenne relative (rRMSE) de 0.010 à 0.011.

LR and MLR results with time regimes.

		Linear regression			Multiple Linear Regression			
Series	**h**	**MAPE**	**rRMSE**	k_p	**MAPE**	**rRMSE**	k_p	k_v
BTC	0	0.015	0.025	4	0.012	0.014	8	10
	120	0.007	0.010	7	0.007	0.011	1	1
	240	0.029	0.050	4	0.031	0.052	5	1
	360	0.034	0.041	1	0.037	0.045	1	2
	480	0.041	0.062	2	0.039	0.061	2	1
	600	0.065	0.082	2	0.065	0.080	2	2
	720	0.028	0.035	1	0.026	0.035	1	5
	840	0.017	0.024	7	0.018	0.024	7	1
	960	0.030	0.040	4	0.029	0.040	1	10
	1.080	0.029	0.039	1	0.022	0.031	3	3
	1.200	0.018	0.025	8	0.021	0.026	8	2
	1.320	0.020	0.026	5	0.021	0.027	7	7

Figure 4

Univariate and multivariate LSTM results with time regimes.

		Univariate LSTM			Multivariate LSTM			
Series	**h**	**MAPE**	**rRMSE**	k_p	**MAPE**	**rRMSE**	k_p	k_v
BTC	0	0.022	0.034	3	0.021	0.030	3	1
	120	0.007	0.011	4	0.007	0.010	2	1
	240	0.044	0.058	3	0.065	0.077	3	1
	360	0.088	0.105	2	0.187	0.233	3	3
	480	0.043	0.066	4	0.041	0.061	1	1
	600	0.068	0.088	1	0.078	0.127	2	1
	720	0.027	0.035	2	0.027	0.043	1	2
	840	0.017	0.023	1	0.017	0.031	3	1
	960	0.027	0.035	6	0.033	0.067	2	1
	1.080	0.025	0.038	3	0.030	0.106	3	1
	1.200	0.021	0.028	1	0.024	0.033	1	1
	1.320	0.018	0.025	1	0.020	0.028	1	2

Figure 5

Résultats des tests pour des lags de valeurs différentes ou Kp est le nombre de point « close » précèdent pris en compte et Kv est le nombre de point « volume » précèdent pris en compte.

Régression linéaire simple

Principe de la régression linéaire simple

La régression linéaire simple est une méthode statistique qui permet d'étudier la relation entre deux variables. Dans le contexte du Machine Learning, elle est souvent utilisée pour prédire une variable cible (ou dépendante) en fonction d'une variables explicative (ou indépendante).

La régression linéaire simple repose sur l'hypothèse qu'il existe une relation linéaire entre la variable dépendante et la variable indépendante, c'est-à-dire que la variable dépendante peut être exprimée comme une fonction linéaire de la variable indépendante. Cette relation est représentée par l'équation suivante :

$$y = a + bx + \epsilon$$

Où :

- *y* est la variable dépendante,
- *x* est la variable indépendante,
- *a* est l'ordonnée à l'origine (ou le biais),
- *b* est le coefficient directeur (ou le poids),
- ϵ est l'erreur aléatoire.

Estimation des paramètres

L'estimation des paramètres *a* et *b* se fait généralement par la méthode des moindres carrés. Cette méthode cherche à minimiser la somme des carrés des résidus, c'est-à-dire la somme des carrés des différences entre les valeurs observées de *Y* et les valeurs prédites par le modèle. Les formules pour calculer *a* et *b* sont les suivantes :

$$b = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Où $\bar{x}$ et $\bar{y}$ sont respectivement la moyenne des *x* et des *y*.

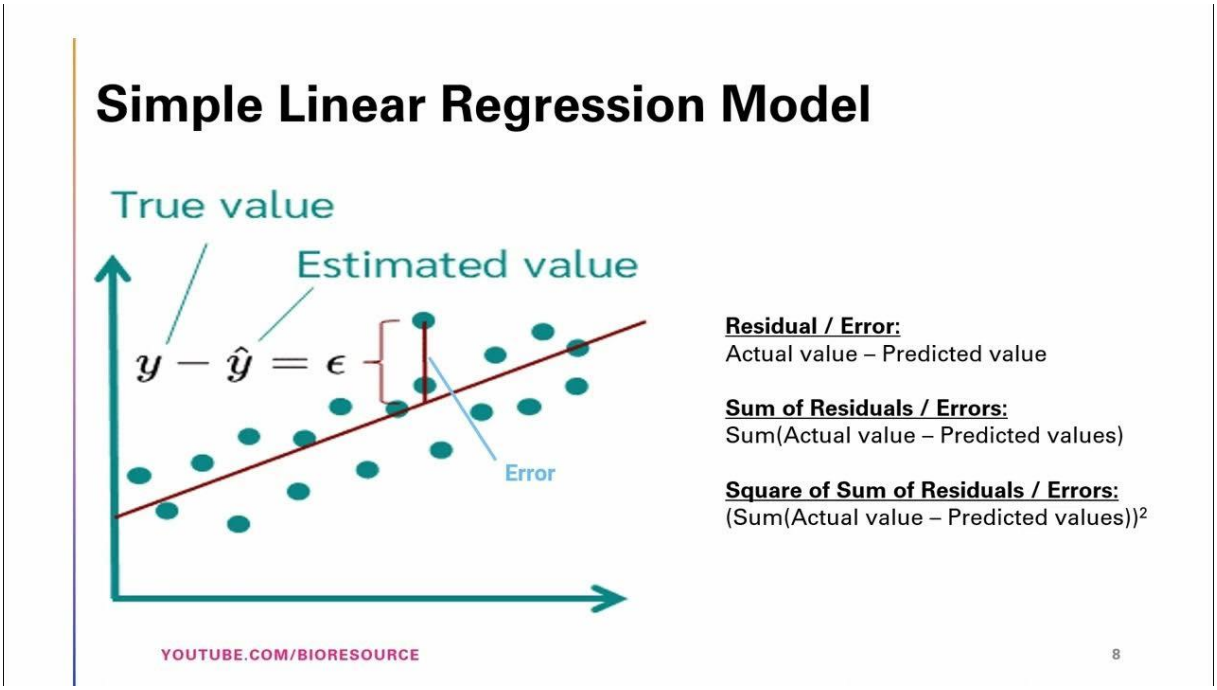

Figure 6

Mesure de l'erreur : La racine de l'erreur quadratique moyenne (RMSE)

La qualité du modèle de régression peut être évaluée à l'aide de plusieurs mesures d'erreur. L'une des plus couramment utilisées est la racine de l'erreur quadratique moyenne (RMSE). Cette mesure donne une idée de la quantité d'erreur que le modèle fait en moyenne. Elle est définie par la formule suivante :

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

Où y_i est la valeur observée, $\hat{y}_i$ est la valeur prédite par le modèle et n est le nombre d'observations.

Grâce à l'élévation au carré des erreurs cette mesure punit fortement les grands écarts de valeur.

La RMSE a la même unité que la variable dépendante, ce qui facilite son interprétation. Plus la RMSE est petite, plus le modèle de régression est précis.

Limites

La régression linéaire simple postule l'existence d'une relation linéaire entre la variable dépendante (le prix du Bitcoin) et la variable indépendante. Toutefois, dans la réalité, cette

relation peut ne pas être linéaire. Par exemple, le prix du Bitcoin peut être affecté par un ensemble complexe de facteurs tels que la dynamique du marché, les régulations gouvernementales, les événements mondiaux, et bien d'autres.

De plus, la régression linéaire simple n'est pas adaptée pour modéliser des motifs complexes. En particulier, dans le cas de séries temporelles financières comme le prix du Bitcoin, les erreurs (différences entre les valeurs prédites et les valeurs réelles) sont souvent corrélées dans le temps. Or, la régression linéaire simple suppose que ces erreurs sont indépendantes, ce qui n'est pas le cas ici. Par conséquent, l'utilisation de la régression linéaire simple peut conduire à des prédictions inexactes dans ce contexte.

Régression linéaire multiple

Présentation de l'algorithme

Le second algorithme que nous avons retenu pour notre étude est la régression linéaire multiple. Il s'agit d'un modèle linéaire multidimensionnel dans lequel une variable quantitative Y est considérée comme étant la solution, la variable à expliquer. Elle est habituellement représentée sur l'axe des ordonnées. Les autres variables, nommées Xi, sont des variables explicatives ou prédictives et on les représente communément sur l'axe des abscisses. Il s'agit donc d'une généralisation de la régression linéaire simple pour permettre d'évaluer les relations linéaires entre une variable réponse et plusieurs variables explicatives.

La formule mathématique de la régression linéaire multiple est la suivante :

$$Y = ß0 + ß1X1 + ß2X2 + \cdots + ßiXi + \varepsilon$$

Cette équation est donc similaire à celle de la régression linéaire simple à la différence qu'il y a plus d'une variable indépendante (X1, X2…)

L'estimation des paramètres ß0…ßi par la méthode des moindres carrés est basé sur le même principe que celui de la régression linéaire simple mais appliqué à i dimensions. On ne trouve ainsi plus la meilleure ligne (celle qui passe le plus près de la ligne de points [xi,yi]) mais le meilleur plan i-dimensionnel qui passe le plus près des points.

Ce résultat est obtenu en minimisant la somme des carrés des déviations des points sur le plan.

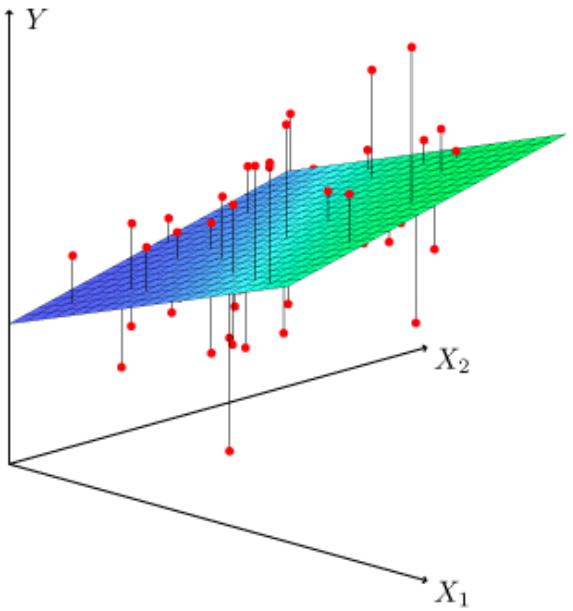

Figure 7

La régression linéaire multiple appliquée à la prédiction des prix

La régression linéaire multiple est un algorithme fréquemment utilisé pour prédire les prix des marchés financiers dans le monde. Dans cette étude, les auteurs utilisent diverses régressions pour prédire le cours des marchés, y compris la régression linéaire multiple. Dans leur cas, la complexité des données a favorisé les modèles de deep learning. Cependant, le marché des cryptomonnaies étant plus récent, la quantité de données est bien plus réduite.

Dans une étude de 2020, les auteurs ont utilisé la régression linéaire multiple pour prédire le prix du Bitcoin et d'autres cryptomonnaies dans les prochains jours.

Pour se faire, ils ont utilisé deux données d'entrées principales : le prix et le volume des jours précédents. Les valeurs les plus hautes et les plus basses du jour ont aussi été ajoutés pour certains essais.

Les meilleurs résultats obtenus sont visibles dans la figure 8 :

Multiple linear regression

Series	MAPE	rRMSE	k_p	k_v
BTC	0.026	0.037	1	1
ETH	0.039	0.053	6	3
LTC	0.045	0.058	2	2
MSFT	0.011	0.015	1	1
INTC	0.013	0.017	1	1
NKSH	0.013	0.018	7	5

Figure 8

Les variables Kp et Kv représentent le lag des deux features : le prix et le volume respectivement.

On peut donc voir que les valeurs de MAPE (Erreur absolue moyenne en pourcentage) et de rRMSE (Erreur quadratique moyenne relative) sont faibles, ce qui suggère donc des prédictions assez précises et une bonne performance du modèle.

Dans cette même étude, les performances de ce modèle sont comparées à celles d'autres modèles utilisant respectivement la régression linéaire simple et les réseaux de neurones récurrents LSTM (Long Short-Term Memory) simples et multiples. Les performances varient selon la cryptomonnaie choisie mais restent extrêmement similaires, ce qui confirme notre choix d'utiliser la régression linéaire multiple.

Une manière d'affiner encore les résultats est de modifier la fenêtre de temps utilisée et notamment de la fixer à 120 points. Cela a pour conséquence de réduire l'aléatoire et de mettre en évidence des tendances. Cette fenêtre est notée h dans la figure 9 :

		Multiple Linear Regression			
Series	**h**	**MAPE**	**rRMSE**	k_p	k_v
BTC	0	0.012	0.014	8	10
	120	0.007	0.011	1	1
	240	0.031	0.052	5	1
	360	0.037	0.045	1	2
	480	0.039	0.061	2	1
	600	0.065	0.080	2	2
	720	0.026	0.035	1	5
	840	0.018	0.024	7	1
	960	0.029	0.040	1	10
	1.080	0.022	0.031	3	3
	1.200	0.021	0.026	8	2
	1.320	0.021	0.027	7	7

Figure 9

On constate donc que l'on obtient bien de meilleurs résultats avec h = 120.

Dans une autre étude, les chercheurs ont tenté de prédire les variations de prix du Bitcoin et de l'Ethereum, les deux plus importantes cryptomonnaies. Pour cela, ils ont utilisé une tout autre approche et se sont basés sur la présence de ces cryptomonnaies dans les discussions sur les réseaux sociaux. En effet, l'intérêt du public pour une cryptomonnaie exerce une forte influence sur son prix.

Les inputs choisis sont les tweets (leur volume) ainsi que les données de Google Trends qui permettent de connaître le nombre de recherches pour tel ou tel sujet. Des études préalables ont mis en évidence le fait que le contenu des tweets était en fait assez peu important, qu'il soit négatif ou positif. Il semblerait en effet que dans le cas des cryptomonnaies, toute publicité a un impact positif sur leur prix.

Les corrélations entre l'intérêt du public et le prix de la cryptomonnaie deviennent évidentes lorsque l'on examine les données des figures 10 et 11 :

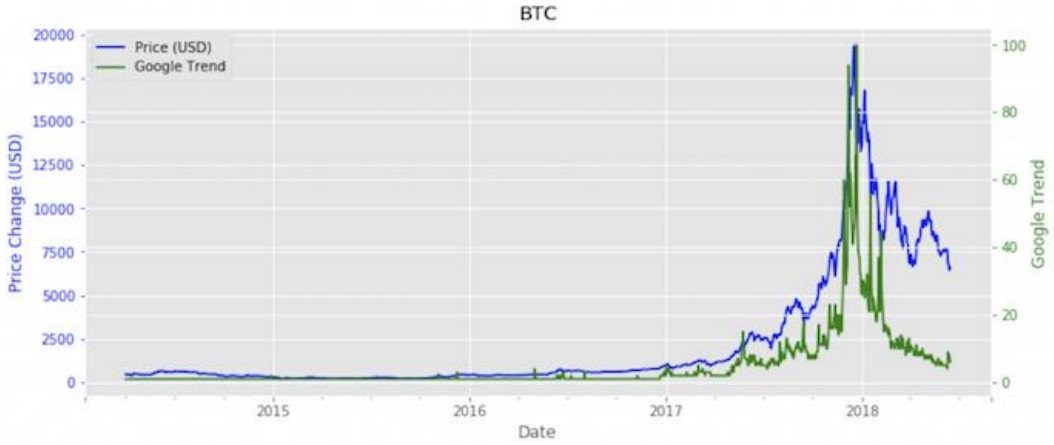

Figure 10

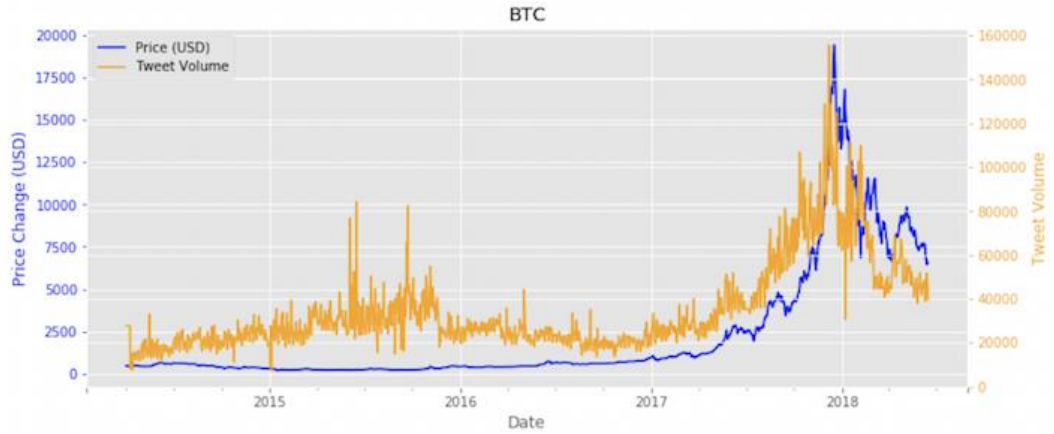

Figure 11

Les résultats obtenus par cette étude sont obtenus sous forme de graphique :

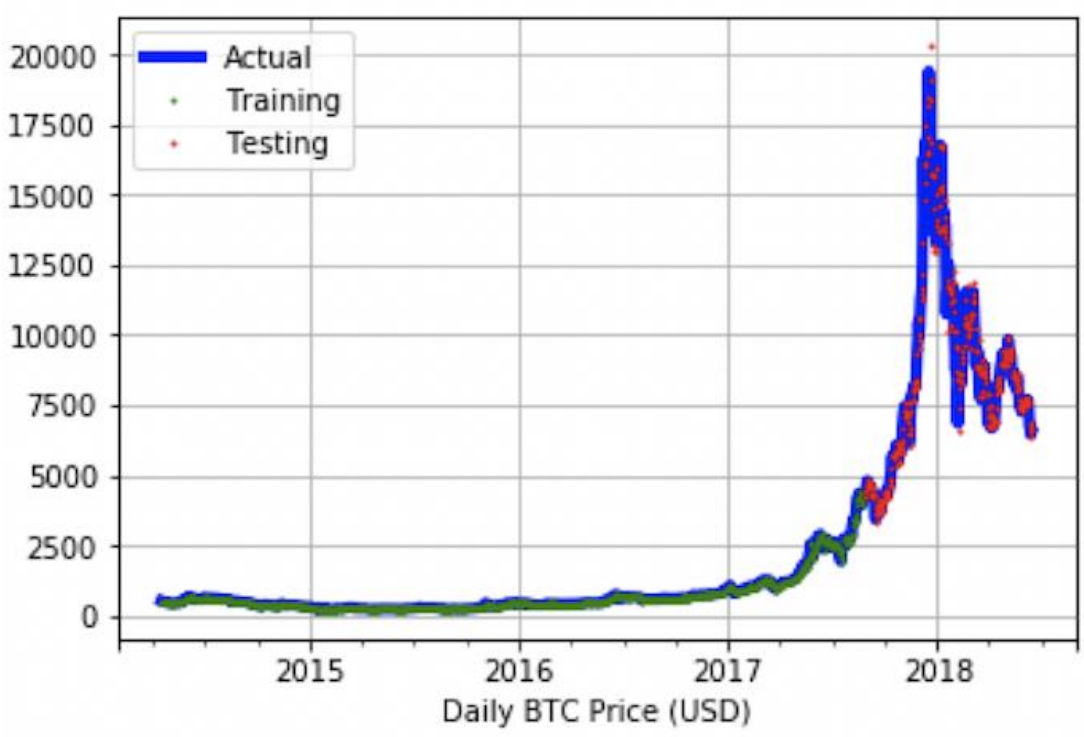

Figure 12

On peut donc constater que les performances du modèle correspondent à nos attentes. Cette approche semble aussi être compatible avec celle de l'étude précédente qui utilisait le prix et le volume de la cryptomonnaie.

Cependant, l'utilisation du volume de tweet dans un modèle présente aussi des inconvénients.

En effet, un tel modèle est facilement manipulable par des personnes externes. Ces dernières pourraient aisément fausser les prédictions en utilisant des bots pour gonfler artificiellement le volume de tweets. Dans notre cas, ce type de risque est inexistant mais si un modèle comme celui-ci était utilisé par le public, il deviendrait rapidement inutilisable.

Le second inconvénient est d'ordre pratique et nous affecte beaucoup plus. En effet, pour analyser le volume des tweets, il faut les récupérer, ce qui demande une puissance de calcul supérieure à celle dont nous disposons. Cependant, il nous est possible d'utiliser Google Trends, qui est un paramètre moins précis mais tout de même fortement corrélé.

Par conséquent, la régression linéaire multiple nous paraît être l'un des meilleurs algorithmes pour prédire le cours des cryptomonnaies, particulièrement si l'on inclut des inputs liés à l'intérêt du public.

Les paramètres de l'algorithme semblent en revanche varier fortement d'une étude à l'autre, ce qui nous pousse à croire que l'implémentation de cet algorithme demandera beaucoup de réglages et d'affinage.

Réseau LSTM

Présentation de l'algorithme

Le dernier algorithme qui a retenu notre attention est le LSTM (Long Short-Term Memory). Il s'agit d'un réseau de neurones artificiels récurrent adapté aux prédictions sur des séries chronologiques, comme les cours des marchés. Ce type de réseau a pour particularité de dépasser le problème de « vanishing gradient ». Ce problème apparaît lorsqu'un réseau utilise un nombre de couches élevé avec des fonctions d'activations comme la fonction sigmoïde. En effet, les gradients de la fonction de perte s'approchent de zéro, ce qui rend le réseau difficile à entraîner, et particulièrement ses premières couches. Les réseaux LSTM utilisent des connexions de rétroaction ce qui les rend différent des réseaux classiques qui n'utilisent que des connexions « vers l'avant ».

Cette propriété permet aux LSTM de traiter des séquences de données entières (comme des séries chronologiques) sans traiter indépendamment chaque point de la séquence. A la place, ils gardent les informations utiles des données précédentes de la séquence pour aider à traiter les données suivantes. Un réseau LSTM peut donc apprendre des patterns récurrents comme des augmentations des prix selon la saison, ce qui a tendance à se produire pour les cryptomonnaies. Il garde un contexte de plus long terme. Plus les patterns sont séparés par de grandes périodes, plus ce type de réseau s'avère efficace.

Pour faire simple, la sortie d'un LSTM à un point particulier dépend de trois choses : la mémoire long terme actuelle du réseau (cell state), la sortie du précédent point (hidden state précédent) et la donnée d'entrée pour le pas actuel.

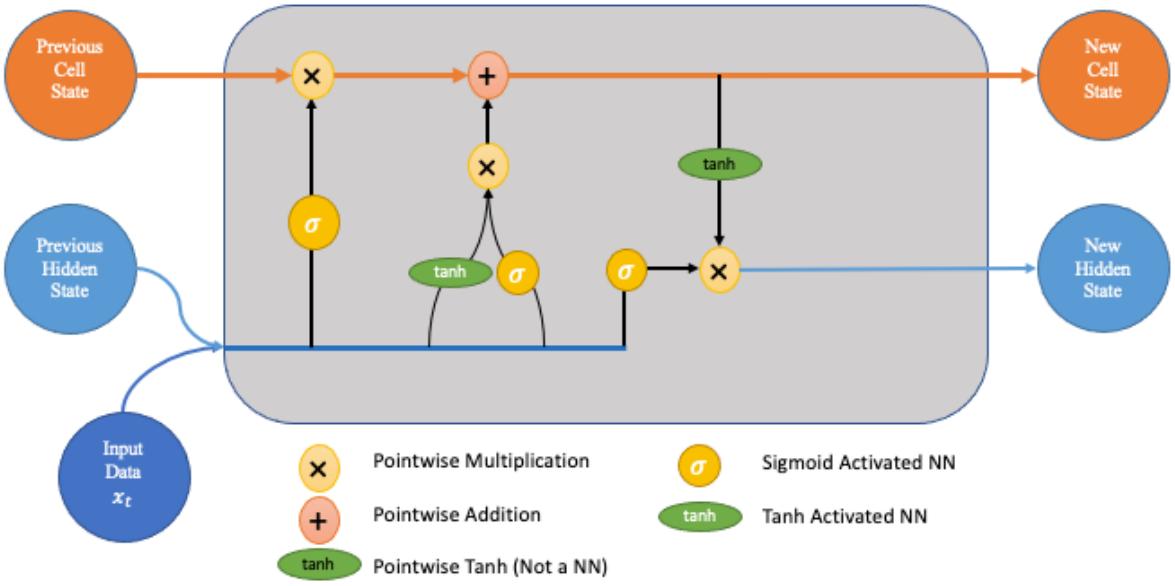

Figure 13

Un LSTM utilise une série de portes qui contrôlent comment l'information d'une séquence de données entre, est stockée et quitte le réseau. Il y a habituellement trois portes : porte

d'oubli, porte d'entrée et porte de sortie. Ces portes sont similaires à des filtres et sont chacune un réseau de neurone distinct.

La porte d'oubli sert à déterminer quelles parties de la mémoire long terme devraient désormais avoir moins d'influence selon la sortie du point précédent ainsi que le point actuel de la séquence.

La porte d'entrée utilise les mêmes entrées que la porte d'oubli mais permet de choisir quelles nouvelles informations il faut ajouter au réseau de mémoire long terme (cell state).

Finalement, la porte de sortie détermine le nouveau hidden state qui sera transmis ensuite. Pour cela, elle utilise le réseau de mémoire long terme qui vient d'être modifié, la sortie du point précédent (hidden state précédent) et le point actuel de la séquence.

Pour obtenir le résultat final (la prédiction), il nous faut convertir le dernier hidden state en sortie exploitable. On utilise alors une couche linéaire comme toute dernière étape. (Cette couche n'est pas représentée sur le diagramme)

Les réseaux LSTM appliqués à la prédiction des prix

Les réseaux LSTM sont les algorithmes les plus utilisés actuellement pour la prédiction des cours des cryptomonnaies. Dans l'étude que nous avons commenté dans la partie précédente, les auteurs ont aussi utilisé des réseaux LSTM, l'un simple et l'autre multiple. Le premier prend en paramètre d'entrée le prix de la journée et le second prend en plus de cela le volume.

Les chercheurs ont testé un large spectre de paramètres pour leur modèle. Ils ont varié les epochs entre 300 et 800 et les batch size entre 22 et 82 avec un pas de 100 et 10 respectivement.

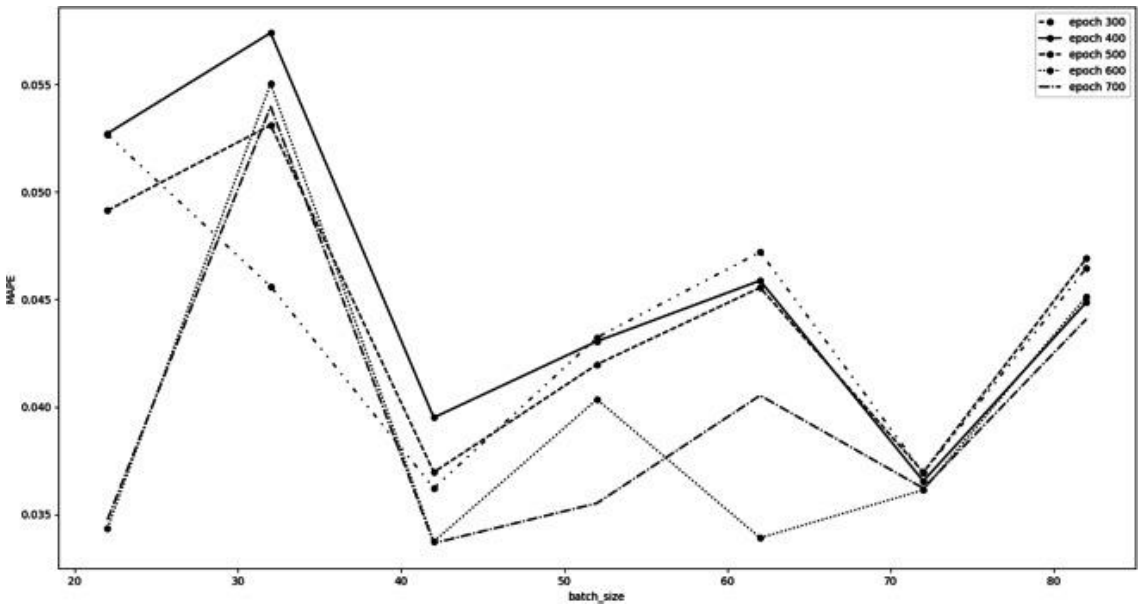

Figure 14

On peut donc voir qu'avec un batch size de 32 (qui est la valeur de base lorsque l'on utilise Keras), le modèle est beaucoup moins performant. La batch size retenue dans cette étude est 72 puisqu'elle permet d'obtenir des résultats intéressants peu importe le nombre d'epochs. Une batch size de 42 semble aussi pouvoir fournir de bons résultats tout en étant plus instable. En revanche, le choix du nombre d'epoch est moins ardu puisqu'à 600 epoch, on obtient les meilleures performances presque pour chaque batch size.

Chaque série chronologique est composée de 200 prix, dont 80 en commun avec la série suivante, et ce pour mettre en évidence des patterns dans l'évolution des prix. Avec les données utilisées dans cette étude, cela donne 12 séries pour le Bitcoin et l'Ethereum.

Les résultats obtenus avec ces paramètres sont visibles dans la figure 15 :

	Univariate LSTM			**Multivariate LSTM**			
Series	**MAPE**	**rRMSE**	k_p	**MAPE**	**rRMSE**	k_p	k_v
BTC	0.027	0.041	1	0.038	0.048	2	1
ETH	0.034	0.052	6	0.057	0.076	2	1
LTC	0.035	0.051	1	0.039	0.054	1	1
MSFT	0.012	0.015	1	0.012	0.015	1	2
INTC	0.013	0.017	2	0.013	0.017	1	1
NKSH	0.014	0.020	7	0.013	0.018	1	2

Figure 15

Les variables Kp et Kv représentent le lag des deux features : le prix et le volume respectivement.

On constate que les résultats sont assez proches de ceux obtenus avec la régression linéaire multiple. Le modèle offre donc une bonne performance. On peut cependant voir que le réseau LSTM multiple est moins précis que le réseau LSTM simple, sauf pour la monnaie NKSH (que nous n'utilisons pas puisqu'il ne s'agit pas d'une cryptomonnaie).

En testant avec différentes séries chronologiques, les écarts de performances changent :

		Univariate LSTM			Multivariate LSTM			
Series	**h**	**MAPE**	**rRMSE**	k_p	**MAPE**	**rRMSE**	k_p	k_v
BTC	0	0.022	0.034	3	0.021	0.030	3	1
	120	0.007	0.011	4	0.007	0.010	2	1
	240	0.044	0.058	3	0.065	0.077	3	1
	360	0.088	0.105	2	0.187	0.233	3	3
	480	0.043	0.066	4	0.041	0.061	1	1
	600	0.068	0.088	1	0.078	0.127	2	1
	720	0.027	0.035	2	0.027	0.043	1	2
	840	0.017	0.023	1	0.017	0.031	3	1
	960	0.027	0.035	6	0.033	0.067	2	1
	1.080	0.025	0.038	3	0.030	0.106	3	1
	1.200	0.021	0.028	1	0.024	0.033	1	1
	1.320	0.018	0.025	1	0.020	0.028	1	2

Figure 16

Le réseau LSTM simple reste plus performant sur la majorité des séries mais les meilleurs résultats sont obtenus avec le réseau LSTM multiple pour les valeurs h=120 et h=0.

Ces résultats sont donc légèrement meilleurs que ceux obtenus avec la régression linéaire. Cette différence est corroborée par d'autres études récentes. CH, U. K., Deekshith, K., & Alekhya, V. (2022) ont ainsi obtenu une précision légèrement plus élevée avec un réseau LSTM.

Li, Y., & Dai, W. (2020) ont réalisés des prédictions avec un réseau LSTM composé de 30 nœuds avec un taux d'apprentissage de 0.01 et une fonction de perte MSE.

Celles-ci sont modélisées dans la figure 17 :

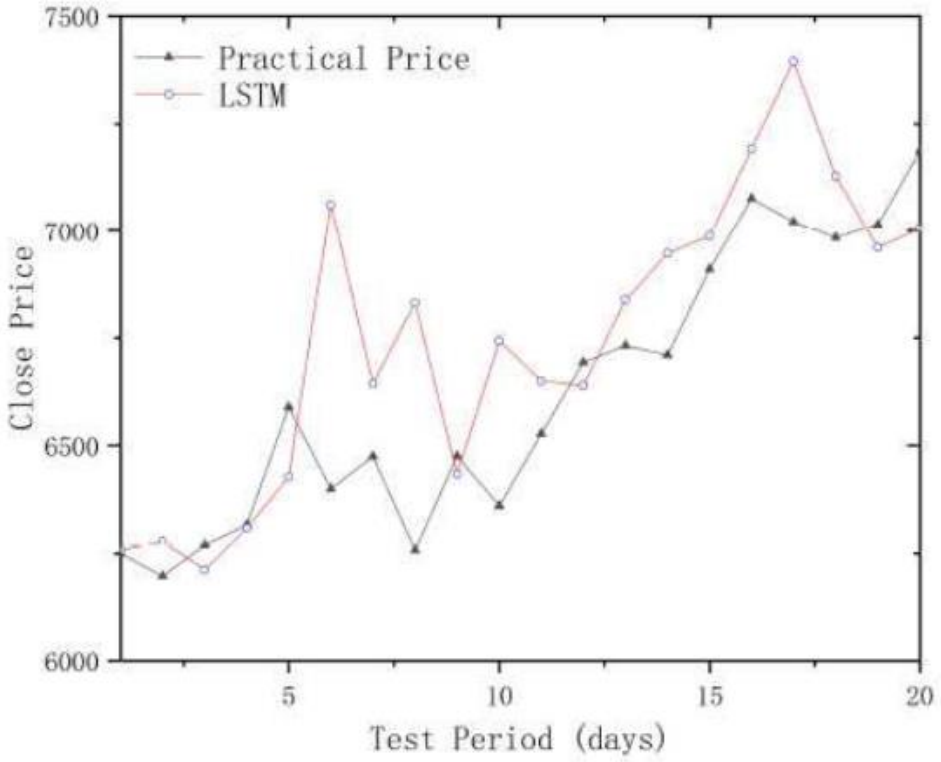

Figure 17

Ces résultats nous paraissent être les plus vraisemblables lorsque l'on prend en compte la volatilité du Bitcoin. Ce modèle prend ainsi en compte l'attention des investisseurs ainsi que des variables macroéconomiques comme le prix de l'or. Ce type d'approche nous semble être la meilleure puisque la prédiction des prix du Bitcoin selon le volume de tweets et Google Trends fournissait déjà des résultats intéressants.

Table 1 List of input attributes

Transaction information	Technical indicators	Macroeconomic variables	Investor attention
lowest price	RSI	gold price	Baidu Index
closing price	MFI	exchange rate	—
highest price	OBV	NYSE Index	—
opening price	—	NASDAQ Index	—
trading volume	—	S&P 500 Index	—
transaction amount	—	federal funds rate crude oil futures price	—

Figure 18

On remarque que certaines améliorations pourraient être apportées, comme la représentation de l'attention des investisseurs. En effet, les auteurs n'utilisent que l'attention des investisseurs chinois. Il serait donc plus judicieux d'y ajouter l'attention sur Twitter ou Instagram par exemple.

Un autre modèle est proposé dans cet étude, mélangeant un réseau CNN et un réseau LSTM :

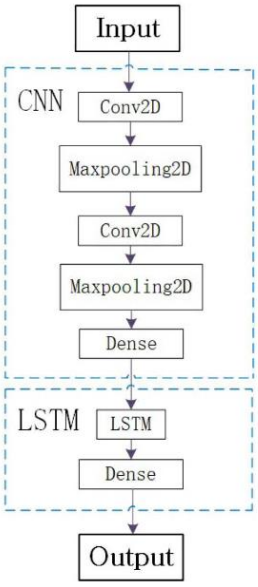

Figure 19

Les chercheurs obtiennent notamment une meilleure précision avec ce modèle. Cela nous montre que pour obtenir les meilleures performances, il peut être judicieux d'utiliser plusieurs algorithmes, selon les paramètres d'entrées.

Pour conclure, le réseau LSTM et plus particulièrement multiple est l'algorithme qui apparaît comme le plus prometteur. Il est cependant bien plus complexe que la régression linéaire, ce qui pourrait rendre difficile son application.

Applications des modèles

A la suite de cela nous avons donc décidé de préciser nos recherches en appliquant les modèles vus précédemment.

Régression linéaire simple

La régression linéaire simple est le premier axe d'application que nous allons développer.

Comme vu dans les études précédentes la régression linéaire simple pourrait apporter des résultats bien que le modèle ne soit peut-être pas assez complexe pour capter les tendances sous-jacentes du bitcoin.

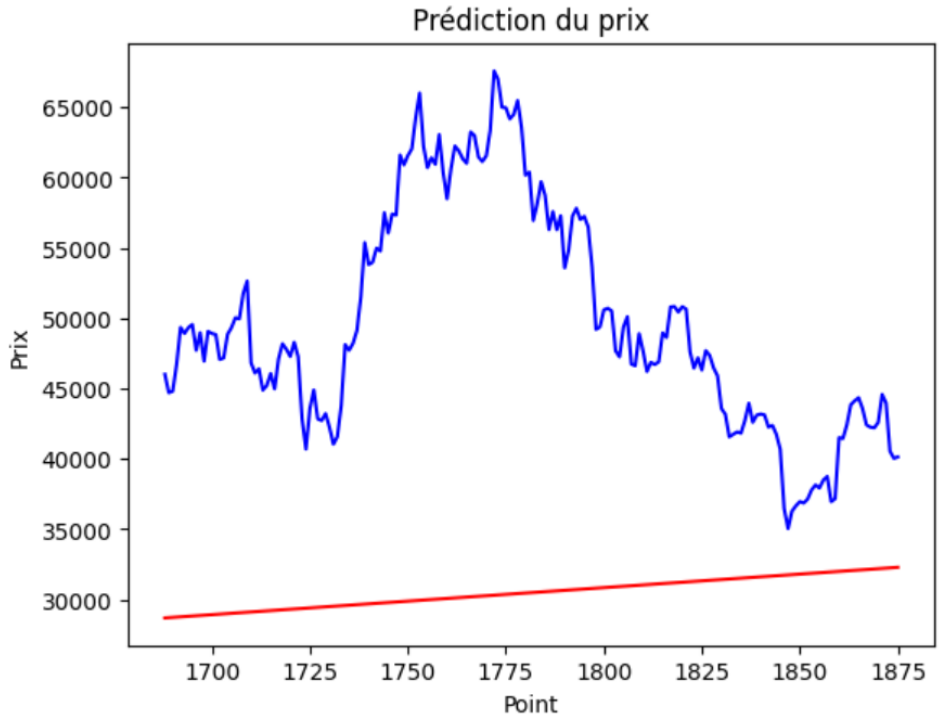

Figure 20

La droite que nous avons réussi à prédire passe certes proche du maximum de points du jeu de donnée mais ne reflète pas la réalité.
Un modèle tel que la régression linéaire simple ne permet pas de suivre précisément l'évolution du prix du bitcoin dû à sa volatilité.

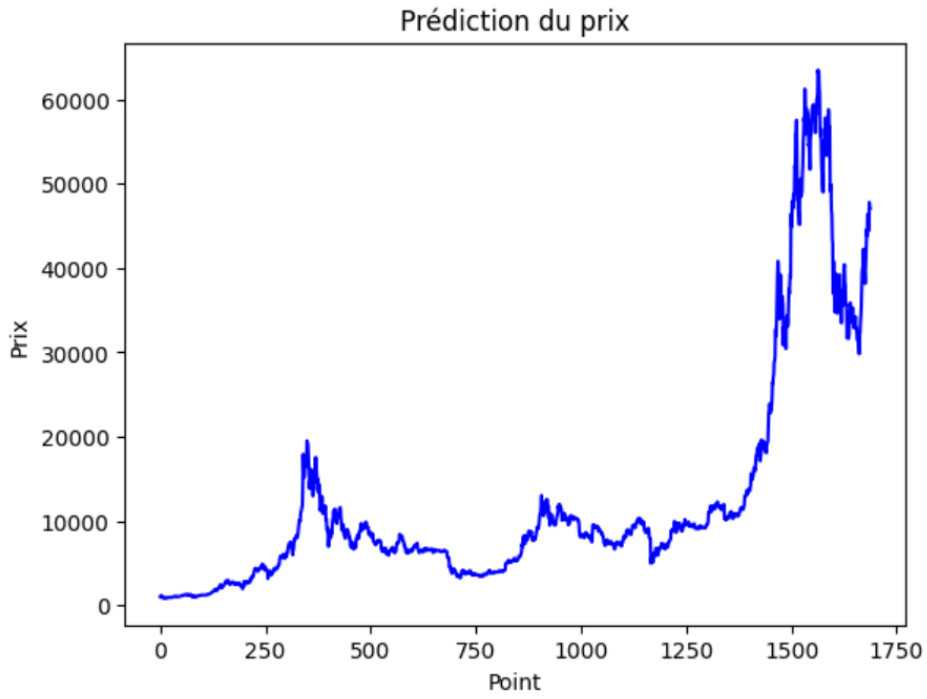

Figure 21

En effet, la majorité des points sont en dessous de 20 000$ puis passent très rapidement jusqu'à 60 000$ avant de redescendre aussi vite.

Il est impossible de modéliser de tels changements avec la régression linéaire simple.

La régression linéaire simple ne permet même pas de modéliser des changements de tendance.

Régression linéaire multiple

Nous allons donc étudier maintenant la régression linéaire multiple qui sera plus apte à prédire les changements de tendance.

Dans un premier temps nous avons réalisé la régression linéaire multiple avec uniquement pour caractéristiques le volume de tweet en rapport avec le bitcoin, le taux d'attention du bitcoin sur google trends et le prix de l'or.

Toutes ces caractéristiques sont décalées de 30 jours avant le jour de prédiction actuelle, pour nous permettre de prédire avec un décalage sur les 30 prochains jours.

	Tweet	Attention	Gold	index
1683	98470	31	1805.9	1683
1684	141201	31	1805.9	1684
1685	98812	31	1809.9	1685
1686	92346	31	1825.0	1686
1687	86926	31	1829.0	1687

Figure 22

Avec cette expérience nous avons obtenu une MAPE (Mean Absolute Percentage Error) de 0.23.

Les résultats sont globalement satisfaisant et réussissent à suivre la tendance du marché même si ce modèle perçoit mal les extrêmes locaux.

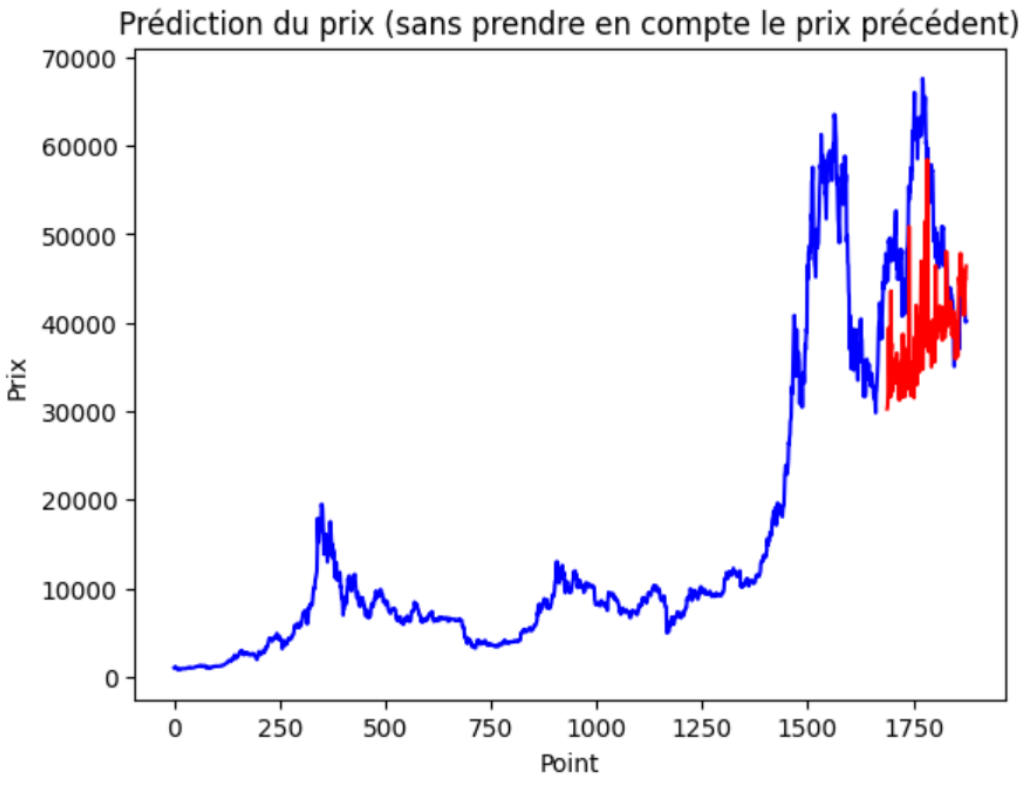

Figure 23

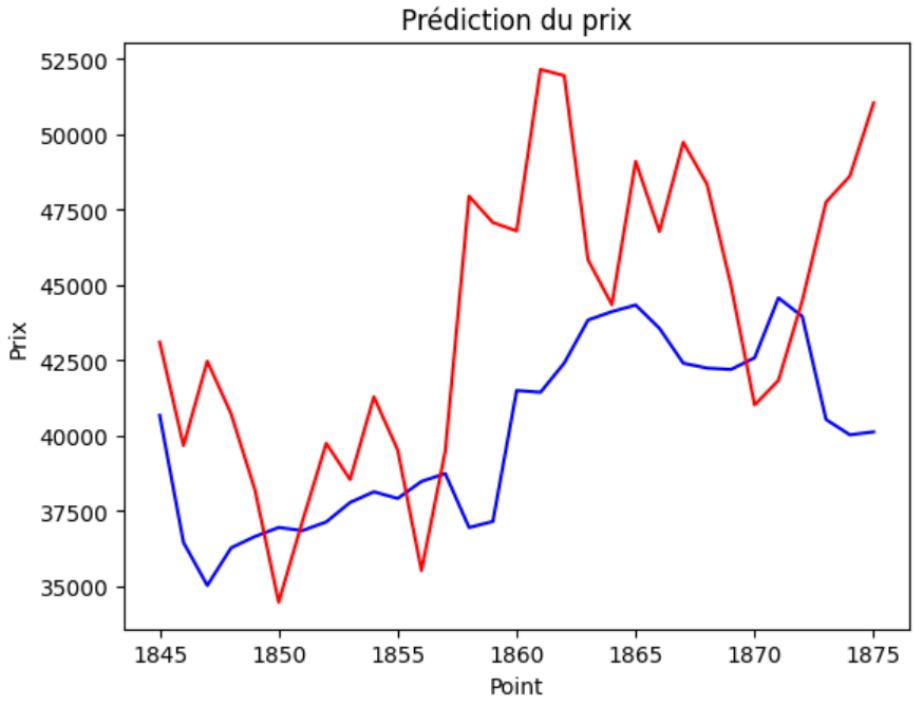

Figure 24

Vue zoomée sur 30 jours.

Pour donner suite à cela, nous avons ajouté une caractéristique PrevClose.

L'objectif de cette caractéristique est d'inclure dans le jeu d'entrainement, pour chaque jour, le prix du jour précédent. Cela ajoutera une information supplémentaire importante qui permettra d'affiner nos prédictions.

	Tweet	Attention	Gold	PrevClose
1845	137590	35	1794.6	41744.328125
1846	120983	35	1788.7	40680.417969
1847	133215	35	1802.2	36457.316406
1848	122898	35	1811.7	35030.250000
1849	110526	35	1808.8	36276.804688

Figure 25

De cette manière nous avons obtenu une MAPE de 0.07, nous avons donc obtenu une baisse de 70% grâce à la caractéristique prevClose.

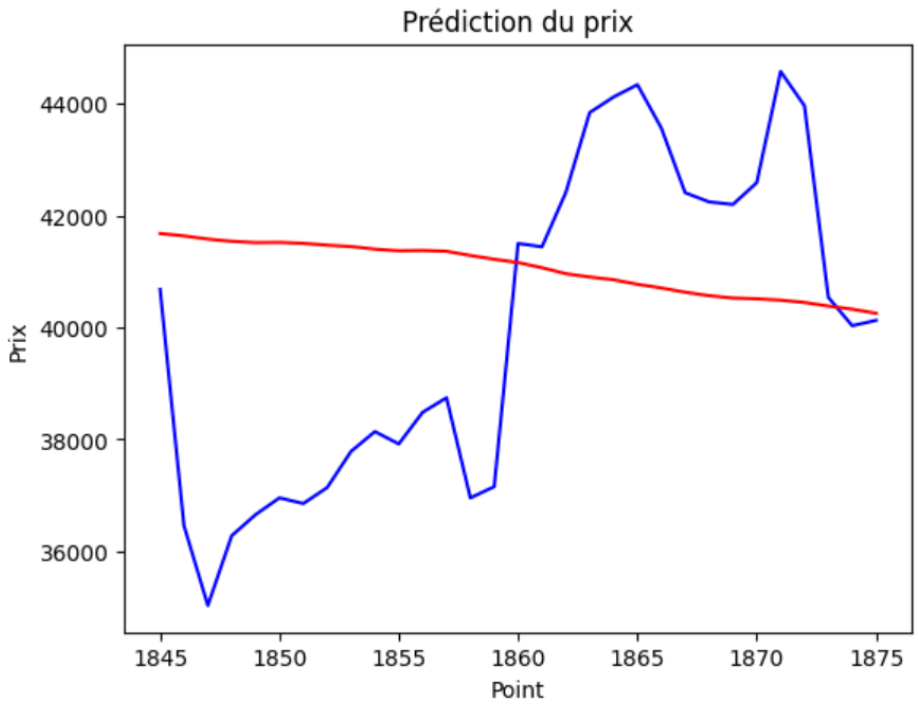

Figure 26

Prédiction sur 30 jours.

Nous pouvons en déduire que cette application de la régression linéaire multiple permet de déterminer les grandes tendances du marché mais pas les tendances à très court terme.

Réseau LSTM

Nous allons maintenant voir si le LSTM permet d'obtenir des prédictions encore plus précises pouvant être appliqué à une situation réelle.

Nous avons tout d'abord essayé de suivre l'étude de Yan Li et Wei Dai publiée en Octobre 2019, cependant nous avons rencontré quelques erreurs nous empêchant de continuer l'expérience, notamment des résultats incohérents ou aberrants.

Par la suite, nous avons testé un modèle très simplifié du LSTM, avec une couche LSTM suivi d'une couche Dense mais encore une fois, les résultats n'ont pas été satisfaissants.

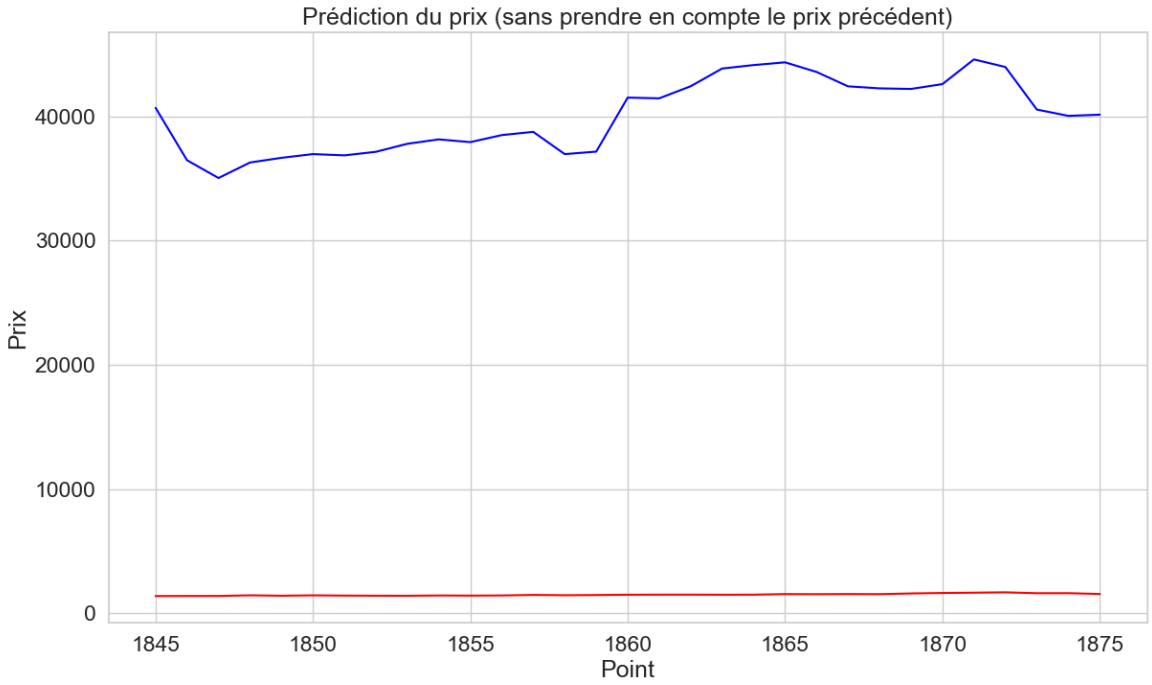

Figure 27

Notre prédiction est en rouge

Limites

Dans le cadre de cette étude, nous avions des limites imposées nous empêchant d'approfondir nos recherches. Ces limites sont notamment dû au temps que nous avons à disposition mais aussi aux données que nous avons pu collecter.

Pour poursuivre cette étude, il faudrait collecter de nombreuses autres caractéristiques telles que le volume de tweet avec Tweepy ou encore faire de l'analyse de sentiment sur ces tweets pour en extraire la tendance (positive ou négative).

Mais aussi le cours d'autres indicateurs tels que le pétrole, l'argent, l'euro, le rouble, le yen, etc…

Nous aurions pu aussi chercher un moyen de collecter le volume et le sentiment de la presse sur le bitcoin et les cryptomonnaies en général ainsi que la tendance des investissements globaux.

Conclusion

Dans le cadre de notre étude, nous avons développé trois modèles dans le but de prédire avec un maximum de précision le cours du bitcoin.

Parmi les trois modèles que nous avons développés, le LSTM aurait théoriquement dû être le plus performant mais nous n'avons pas réussi à l'appliquer à notre cas.

C'est donc la régression linéaire multiple qui a été la plus performante avec une MAPE de 0.07 avec les meilleurs paramètres, suivi de la régression linéaire simple avec 0.37.

Bien que nous ayons eu de bons résultats sur ces deux algorithmes, ils ne permettent pas à eux seuls de prédire avec certitude le cours du bitcoin sans risque.

Ces algorithmes sont utilisables dans un contexte réel en tant qu'indicateur pour avoir une idée de la tendance avant d'effectuer une analyse approfondie.

Cependant lors de nos recherches nous avons réalisé que la plupart des études disponibles sur internet manquent de précisions sur certains points ce qui force à la réalisation d'étapes supplémentaires pour valider le modèle. Cela est probablement dû à la récente démocratisation des cryptomonnaies, c'est un champ de recherche très récent qui nécessite encore quelques années de maturation pour permettre aux chercheurs de mieux comprendre le problème.

La technologie évolue très rapidement, particulièrement dans le domaine du Big Data, ce qui pourrait grandement améliorer la précision des modèles, notamment en prenant en compte en temps réel les sentiments et l'intérêt du grand public et des investisseurs.

L'un des possibles axes d'amélioration de cette étude nous paraît être d'incorporer dans les données d'autres moyens d'évaluer la popularité du Bitcoin, principalement auprès d'acteurs clés comme les influenceurs ou la presse, avec un indice de confiance par exemple.

Sources

https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-l-inf-intRegmult.pdf

https://delladata.fr/regression-lineaire-multiple/

https://statsandr.com/blog/multiple-linear-regression-made-simple/

https://corporatefinanceinstitute.com/resources/data-science/multiple-linear-regression/

http://www.stat.yale.edu/Courses/1997-98/101/linmult.htm

https://arxiv.org/ftp/arxiv/papers/2009/2009.10819.pdf

https://scholar.smu.edu/cgi/viewcontent.cgi?article=1039&context=datasciencereview

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7924725/

https://en.wikipedia.org/wiki/Vanishing_gradient_problem

https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484

https://en.wikipedia.org/wiki/Long_short-term_memory

https://penseeartificielle.fr/comprendre-lstm-gru-fonctionnement-schema/

https://intellipaat.com/blog/what-is-lstm/?US

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

https://sajet.in/index.php/journal/article/download/220/227

https://ietresearch.onlinelibrary.wiley.com/doi/pdfdirect/10.1049/joe.2019.1203

https://medium.com/geekculture/linear-regression-from-scratch-in-python-without-scikit-learn-a06efe5dedb6

https://www.nintyzeros.com/2020/02/linear-regression-from-scratch.html