

**h e g**

Haute école de gestion  
Genève

# **Étude et implémentation d'un Bot de trading intelligent**

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

par :

**Anushan Alvaipillai**

Conseiller au travail de Bachelor :

**David BILLARD, enseignant HES**

**Haute École de Gestion de Genève, le 27 juin 2022**

**Haute École de Gestion de Genève (HEG-GE)**

**Filière IG**

## 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 «Bachelor of Science HES-SO en Informatique de gestion».

L'étudiant a envoyé ce document par email à l'adresse remise par son conseiller au travail de Bachelor pour analyse par le logiciel de détection de plagiat URKUND, selon la procédure détaillée à l'URL suivante : <https://www.orkund.com>.

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 27 juin 2022

Anushan Alvaipillai



## Remerciements

Je tiens d'abord à remercier Mr. Billard pour m'avoir accompagné durant ce travail de Bachelor, ainsi que pour les cours qu'il a enseignés pendant ma formation à l'HEG

Je tiens aussi à remercier ma famille et mes amis qui m'ont soutenu durant toutes mes études.

## Résumé

Une des activités les plus angoissantes qui existe est le trading. Selon wealth within, 90% des personnes qui pratiquent cette activité perdent de l'argent. C'est une activité qui exige du travail et surtout de pouvoir gérer ses émotions. C'est ce qui fait la réussite de nombreux traders.

C'est pourquoi dans ce travail de recherche, nous allons voir comment créer un programme qui se positionnera automatiquement sur des cryptomonnaies pour nous. Pour ça nous allons d'abord voir les différentes stratégies qu'utilisent les traders. Les différentes stratégies sont l'analyse fondamentale, l'analyse sentimentale et l'analyse technique.

Ensuite nous allons reproduire ces différentes stratégies dans notre programme. Concernant l'analyse fondamentale, il va récupérer des informations fondamentales d'une cryptomonnaie sur une plateforme grâce à une API. Pour l'analyse sentimentale, pour chaque cryptomonnaie auquel nous souhaitons investir, il va récupérer les derniers tweets et les derniers articles délivrés par google news pour évaluer le sentiment de ces cryptomonnaies. Concernant l'analyse technique, le programme va utiliser la librairie TA pour reproduire les différents indicateurs techniques qui vont permettre au Bot de prendre des positions.

Puis nous allons voir comment le Bot peut nous envoyer un message telegram, lorsqu'il y a une prise de position. On regardera aussi pour l'héberger sur un serveur Ubuntu. Après ça, nous effectuerons une simulation pour voir la performance du Bot dans le passé. Finalement, nous terminerons par une conclusion.

# Table des matières

<b>Déclaration.....</b>	<b>i</b>
<b>Remerciements .....</b>	<b>ii</b>
<b>Résumé .....</b>	<b>iii</b>
<b>Table des matières.....</b>	<b>iv</b>
<b>Liste des tableaux .....</b>	<b>vii</b>
<b>Liste des figures.....</b>	<b>vii</b>
<b>1. Introduction.....</b>	<b>1</b>
<b>2. Trading .....</b>	<b>3</b>
<b>2.1 Les différents types de trading .....</b>	<b>3</b>
<b>2.2 Les différents types d'analyses .....</b>	<b>3</b>
2.2.1 Analyse fondamentale.....	3
2.2.1.1 Nombre de transaction .....	6
2.2.2 Frais de transaction.....	6
2.2.2.1 Nombre d'adresses actives .....	6
2.2.2.2 Taux de hachage (Hash rate) et preuve de participation (proof of stake) ....	6
2.2.2.3 Whitepaper (livre plan) .....	7
2.2.2.4 L'équipe .....	7
2.2.2.5 Tokenomics et inflation.....	8
2.2.2.6 Network Value to Transactions Ratio (NVT) .....	10
2.2.2.7 Market value to realized value ratio. (MVRV).....	10
2.2.3 Analyse Sentimentale .....	11
2.2.4 Analyse Technique.....	13
2.2.4.1 Définition.....	13
2.2.4.2 Structure .....	13
2.2.4.3 Indicateur technique .....	15
<b>3. Bot de trading .....</b>	<b>18</b>
<b>3.1 Définition .....</b>	<b>18</b>
<b>3.2 Pourquoi choisir un Bot de trading .....</b>	<b>18</b>

<b>3.3</b>	<b>Différents Bots de trading et stratégie .....</b>	<b>18</b>
3.3.1	Grid trading .....	18
3.3.2	DCA trading .....	19
3.3.3	B-Cube Ai .....	19
3.3.4	Constat .....	19
<b>4.</b>	<b>Construire son Bot de trading.....</b>	<b>20</b>
<b>4.1</b>	<b>Comment obtenir les cours d'un crypto dans le passé ?.....</b>	<b>20</b>
<b>4.2</b>	<b>Analyse fondamentale – Bot de trading .....</b>	<b>23</b>
<b>4.3</b>	<b>Analyse Sentimental .....</b>	<b>27</b>
4.3.1	Twitter .....	27
4.3.2	News api .....	29
4.3.3	Analyse de texte .....	29
4.3.4	Analyse sentimental final cryptomonnaie .....	30
<b>4.4</b>	<b>Analyse technique .....</b>	<b>30</b>
4.4.1	Libraire TA .....	30
4.4.2	Stratégie d'analyse.....	31
4.4.2.1	Awesome Ocillator .....	31
4.4.2.2	Stochastic RSI.....	31
4.4.2.3	William %R .....	31
4.4.3	Condition d'achats et de ventes .....	33
4.4.4	Constat .....	34
<b>4.5</b>	<b>Exchange FTX .....</b>	<b>35</b>
<b>4.6</b>	<b>Notifications – Telegram.....</b>	<b>36</b>
<b>4.7</b>	<b>Résumé fonctionnement du Bot de trading .....</b>	<b>38</b>
<b>4.8</b>	<b>Serveur AWS .....</b>	<b>41</b>
<b>5.</b>	<b>Backtesting .....</b>	<b>48</b>
<b>5.1</b>	<b>Résultat 2022.....</b>	<b>49</b>
<b>5.2</b>	<b>Résultat 2022 sans l'analyse sentimentale (fear and gread).....</b>	<b>50</b>
<b>5.3</b>	<b>Performance durant chaque mois.....</b>	<b>51</b>

<b>6. Conclusion .....</b>	<b>52</b>
<b>Bibliographie .....</b>	<b>53</b>
<b>Annexe 1 : Informations proposées par Messari – Bitcoin .....</b>	<b>55</b>
<b>Annexe 2 : Bot fondamental.....</b>	<b>62</b>
<b>Annexe 3 : Refus pour l’API étudiante de twitter .....</b>	<b>67</b>
<b>Annexe 4 : Tableau des commissions .....</b>	<b>68</b>
<b>Annexe 5 : Bot de trading.....</b>	<b>70</b>

## Liste des tableaux

Tableau 1 : Analyse du P/S .....	4
Tableau 2 : Analyse du P/B .....	5
Tableau 3 : Crypto A circulation.....	10
Tableau 4 : Analyse sentimentale - Bonne nouvelle .....	11
Tableau 5 : Analyse sentimentale - Mauvaise nouvelle .....	12

## Liste des figures

Figure 1 : Funding Rounds Solana .....	8
Figure 2 : Calendrier de distributions jetons - Solana .....	9
Figure 3 : Graphique Solana .....	9
Figure 4 : Bitcoin - Tendence haussière .....	14
Figure 5 : Bitcoin - Tendence Baissière .....	14
Figure 6 : Ethereum - sans tendance .....	15
Figure 7 : RSI - Surachat et survente .....	16
Figure 8 : Calcul EMA .....	17
Figure 9 : EMA - prise de positions.....	17
Figure 10 : Stratégie Grid .....	18
Figure 11 : Instanciation du tableau.....	21
Figure 12 : Ajouts des OHCVL .....	21
Figure 13 : Exportation fichier CSV .....	22
Figure 14 : Récupération des cryptomonnaies du top 100.....	24
Figure 15 : Récupération des informations fondamentales .....	25
Figure 16 : Analyse fondamentale d'une cryptomonnaie .....	26

Figure 17 : Classement fondamental des cryptomonnaies .....	26
Figure 18 : Obtenir les récents tweets d'une cryptomonnaie .....	27
Figure 19 : Nettoyer le texte .....	28
Figure 20 : Récupérer les articles d'une cryptomonnaie .....	29
Figure 21 : Prise de la pipeline - sentiment-analysis.....	29
Figure 22 : Obtenir le sentiment du texte.....	30
Figure 23 : Formule Willam % R.....	32
Figure 24 : Placement des indicateurs techniques dans le data frame .....	32
Figure 25 : Condition d'achat technique .....	33
Figure 26 : Condition de vente technique .....	33
Figure 27 : Clés personnelles FTX pour passer les ordres .....	35
Figure 28 : Message de félicitations pour avoir terminé la configuration .....	36
Figure 29 : Bot - Achat .....	37
Figure 30 : Bot - Vente .....	37
Figure 31 : Diagramme de séquence - Bot.....	40
Figure 32 : AWS - EC2.....	41
Figure 33 : AWS - Instances.....	41
Figure 34 : AWS - serveur Ubuntu 20.04.....	42
Figure 35 : AWS - Paire de clés .....	42
Figure 36 : AWS - Serveur en cours d'exécution .....	43
Figure 37 : Putty - Sauvegarde clé privée.....	43
Figure 38 : AWS - DNS IPv4 public .....	44
Figure 39 : Putty - Paramétrage du Host Name.....	44
Figure 40 : Importation clé privée .....	45

Figure 41 : Connexion SSH réussie .....	46
Figure 42 : Crontab - planificateur de tâches .....	47
Figure 43 : Résultat 2022 avec le Fear and Gread .....	49
Figure 44 : Résultat 2022 sans le Fear and Gread .....	50
Figure 45 : Graphique capitalisation boursière totale cryptomonnaie.....	51
Figure 46 : Performance sur chaque mois.....	51

# 1. Introduction

Faire de l'argent en dormant : ce sont les quelques mots que recherchent de plus en plus de personnes sur Google. On appelle ça le revenu passif, c'est-à-dire : pouvoir continuer à gagner de l'argent, même après avoir terminé de travailler. Sur internet, on a pu voir différentes méthodes émerger dans le but d'obtenir des revenus passifs. L'idée d'écrire un livre est la première nous venant à l'esprit quand on pense au revenu passif mais le blog ou le marketing digitale ont le vent en poupe de nos jours. Cependant, ces revenus passifs demandent tout de même un entretien particulier pour assurer leur pérennité. Par exemple, pour un blog ou un livre, il faut, dans la plupart des cas, le promouvoir régulièrement pour continuer de le vendre. C'est la raison pour laquelle je considère que ces différents projets ne constituent pas un revenu strictement passif.

Cependant, il existe certains types de services que je considère comme des revenus totalement passifs. Parmi eux : les Bots de trading. En effet, les Bots de trading permettent de faire travailler son argent en ne faisant rien. Ceci, grâce à un programme informatique. Malheureusement, comme nous allons le voir durant ce travail de recherche, la plupart de ces robots n'inspectent qu'une partie de ce qui peut être inspecté et prennent des commissions assez importantes. Ainsi, j'ai décidé de créer mon propre Bot de trading intelligent de A à Z et je vais vous montrer le processus de la phase de la création jusqu'à la phase de déploiement.

L'objectif de ce travail de recherche est de montrer la méthodologie qu'utilisent les traders. Ensuite, de réussir à les reproduire de manière automatisée par notre Bot. Ce Bot réussira à prendre en compte dans son analyse globale les différentes analyses du trader qui sont l'analyse fondamentale, l'analyse sentimentale et l'analyse technique à l'aide de plusieurs outils. Dans ce travail, nous allons nous concentrer sur le domaine de la cryptomonnaie puisque ce domaine recense beaucoup d'informations en open source qui aide les programmeurs à créer leur Bot. Ce travail n'est absolument pas un conseil financier et n'incite en aucun cas à acheter de la cryptomonnaie. Ma démarche est surtout orientée vers une vulgarisation de la simplification qu'apporte les outils informatiques.

Ce travail de recherche se déroulera de la manière suivante : Nous allons tout d'abord voir le fonctionnement du trading avec les différentes analyses. Ensuite, nous allons expliciter le fonctionnement des différents Bots de trading qui existent sur le marché de la cryptomonnaie. Puis nous verrons les différentes étapes à suivre afin de construire

son propre Bot de trading intelligent. Par la suite, on fera une simulation de notre Bot pour analyser les résultats qu'il aurait produits cette année. Enfin, nous terminerons par une conclusion.

## 2. Trading

Le trading est un mot en anglais qui consiste simplement à acheter et à vendre des produits financiers (crypto, actions, devises, options financières, etc.). Ces échanges sont effectués par des traders depuis la salle de marchés d'une institution financière ou depuis Internet pour les traders indépendants. Il existe 2 types de position dans le trading. Il y a les « long » qui permet de parier à la hausse. Il y a les « short » qui permet de parier à la baisse.

### 2.1 Les différents types de trading

Selon Thami Kabbaj, il existerait 3 types de trading :

**Swing trading** → Le swing trading est de prendre une position et de la garder de cinq jours à un mois.

**Day trading** → Le day trading consiste à se focaliser sur les ouvertures et les fermetures d'une position d'un cours. La durée de la position peut varier entre 15 minutes à une journée.

**Le scalping** → Le scalping est similaire à la stratégie du day trading, mais la durée de la position est inférieure à 15 minutes. Cependant les traders qui utilisent cette méthode, se positionnent avec des petits montants par rapport à leur capital, mais ils effectuent plein de positions (en moyenne 300 positions par jour)

### 2.2 Les différents types d'analyses

Dans ce domaine, il existe trois types d'analyse. Les différents types d'analyses sont les suivantes :

#### 2.2.1 Analyse fondamentale

L'analyse fondamentale permet d'étudier l'état économique d'un actif financier (devises, matières premières, actions, etc.). En effet, pour une entreprise, l'analyse fondamentale peut être établie avec certains indicateurs économiques comme le chiffre d'affaires, la marge brute, etc. et en faisant des calculs avec ces indicateurs économiques. Grâce à cela, on peut voir si le cours de l'action est sous-évalué ou surévalué. De nombreux traders se basent uniquement sur l'analyse fondamentale et se positionnent sur les actifs sous-évalués et vendent lorsque les actifs sont surévalués. On les surnomme les fundamentalistes.

Pour analyser l'état de santé économique d'une entreprise, je vais vous énumérer quelques calculs que conseille Warren Buffet (Warren Buffet est un célèbre investisseur qui a fait fortune grâce à ces investissements) :

$P/S = \text{Capitalisation boursière (nombre d'actions en circulation * prise de l'action)} / \text{Chiffre d'affaires}$

Le résultat de ce calcul nous permet de savoir si un actif est sous-évalué ou surévalué. La condition de cette surévaluation est que le résultat doit être supérieur à 1, si elle est inférieure ça veut dire que le prix de l'action est sous-évalué. Exemple :

Tableau 1 : Analyse du P/S

Entreprise	A	B
Capitalisation boursière	\$80 M	\$80 M
Chiffres d'affaires	\$40 M	\$160 M
P/S (Price / sales)	2	0.5
Résultat de l'analyse	Surévalué	Sous-évalués

(Anushan Alvaipillai, 2022)

$P/B = \text{Capitalisation boursière} / \text{valeur comptable (total actif – total passif)}$

La valeur comptable va nous permettre de savoir si l'entreprise a plus de bien que de dettes. En effet, plus ce nombre positif est élevé, plus elle montre qu'elle a une certaine autonomie.

Le résultat de ce calcul nous permet de savoir si un actif est sous-évalué ou surévalué. La condition de cette surévaluation est que le résultat doit être supérieur à 1, si elle est inférieure ça veut dire que le prix de l'action est sous-évalué. Exemple :

Tableau 2 : Analyse du P/B

Entreprise	A	B
Capitalisation boursière	\$100 M	\$100 M
Valeur comptable	\$60 M	180 M
P/B (Price / Book)	1.67	0.56
Résultat de l'analyse	Surévalué	Sous-évalués

(Anushan Alvaipillai, 2022)

Ces deux calculs servent à illustrer le fonctionnement de l'analyse fondamentale pour une action et à comprendre la méthodologie qu'il faut entamer pour réussir une bonne analyse. On peut constater que ce sont des simples calculs qu'il nous permet de savoir si un actif est sous-évalué ou surévalué. Plus, on a des calculs qui correspondent à notre analyse, plus il est probable que le cours de l'actif aille dans notre direction.

Cependant, pour le monde de la cryptomonnaie, l'analyse fondamentale est totalement différente de celui d'une action. Puisque, dans ce domaine, il n'y a pas les mêmes informations qu'on pourrait avoir sur une action. En effet, beaucoup de cryptomonnaies sont décentralisées, ce qui veut dire que personne n'a vraiment le contrôle sur cet actif. Alors comment faire pour établir une analyse fondamentale sur une cryptomonnaie. Il y a des indicateurs propres à la cryptomonnaie pour nous faire savoir si une cryptomonnaie est sous-évaluée ou surévaluée.

Mais avant de vous énumérer les indicateurs fondamentaux dédiés à la cryptomonnaie, je souhaite parler d'un indicateur que beaucoup de personnes dans cet écosystème conseillent, mais qui est facilement manipulable. Cet indicateur est le nombre d'abonnés qu'a l'entreprise qui a créé la cryptomonnaie. En effet, les abonnés, les likes et les commentaires sont facilement achetables sur internet. Malgré cela, plus le nombre d'abonnés est élevé, plus certaines personnes considèrent que le projet est fiable. C'est pourquoi pas mal de personnes qui sont dans ce domaine recommandent cette

métrique. Mais Binance Academy ne le recommande pas. C'est pourquoi, je vais prendre en considération durant ce travail, l'avis de Binance Academy.

Maintenant, nous pouvons commencer par énumérer les indicateurs pour une bonne analyse fondamentale :

### **2.2.1.1 Nombre de transaction**

Le nombre de transaction est un bon indice, puisqu'il montre l'activité qu'à cette cryptomonnaie. Plus il est élevé, plus c'est positive.

### **2.2.2 Frais de transaction**

Lorsqu'on fait une transaction avec une cryptomonnaie, nous devons payer des frais (fees). Même si cette transaction n'est pas acceptée. Par exemple, pour le cas d'Ethereum, au moment où j'écris (23.05.2022), les frais moyens sont à 0.7913 USD. C'est une belle avancée depuis un an, puisque le 19 mai 2021, les frais moyens s'élevaient à 71.72 \$ sur Ethereum. C'est pourquoi, c'est important de bien connaître les frais de transactions que procure une cryptomonnaie avant d'investir sur celle-ci. Puisque les gens seront rédhitoires pour effectuer des transactions en ayant des frais élevés et ceci peut même freiner l'activité économique de cette cryptomonnaie.

#### **2.2.2.1 Nombre d'adresses actives**

Le nombre d'adresses actives nous permet de savoir si une cryptomonnaie est tenue par une petite poignée de personnes (baleines) ou par une vaste majorité de personnes. Il est important de combiner cette métrique avec la capitalisation boursière pour avoir un ratio. Plus le nombre d'adresses est élevé, plus le prix de cette cryptomonnaie est difficilement manipulable.

#### **2.2.2.2 Taux de hachage (Hash rate) et preuve de participation (proof of stake)**

Le taux de hachage représente le taux de sécurité que représentent les blockchains qui font du proof of work. Plus il y a des personnes qui utilisent ces ressources matérielles pour contribuer à la blockchain, plus cela renforcera la sécurité de cette blockchain. Le même raisonnement s'applique sur les blockchains qui font du proof of stake. Cependant, pour augmenter la sécurité des blockchains proof of stake, il faut stacker ces cryptomonnaies, à l'instar d'utiliser ses ressources. Plus les gens épargnent leur cryptomonnaie, plus cela renforce la sécurité de la blockchain.

### **2.2.2.3 Whitepaper (livre plan)**

Le whitepaper regroupe toutes les informations nécessaires pour comprendre au projet, ainsi que plus de détails. Il est primordial de le consulter avant d'investir sur un projet. En effet, c'est l'élément indispensable du projet. Ce document permet de savoir si le projet est fiable. Elle permet de comprendre ce qu'ils apportent de plus sur le marché. Elle permet de voir les différents use case. De parcourir aussi les roadmaps pour voir s'ils sont ambitieux. Globalement le livre blanc permet de cerner la fiabilité du projet.

### **2.2.2.4 L'équipe**

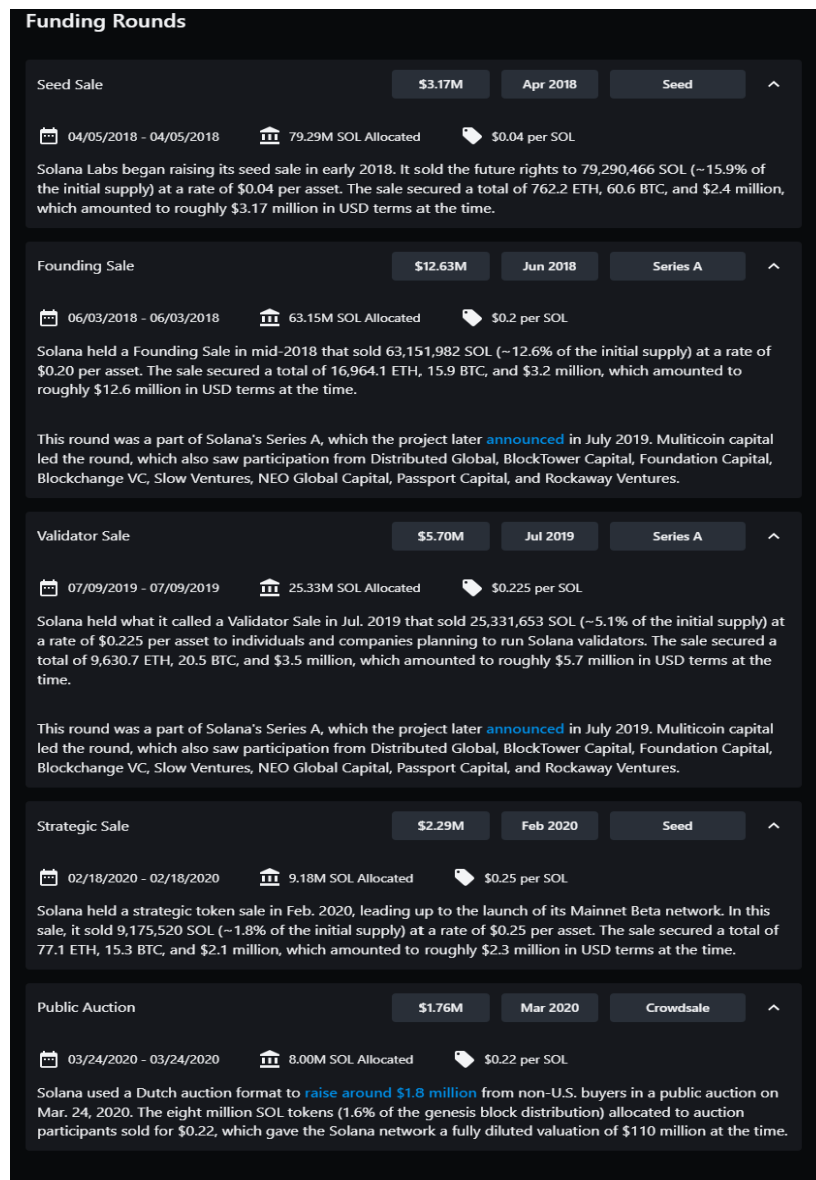
L'équipe d'un projet nous permet de savoir si un projet est une potentielle arnaque ou pas, parce que dans le monde de la cryptomonnaie, les scams (arnaques) sont assez courantes. En effet, plus une équipe est composée de personnes qui ont de l'expérience dans le domaine de la cryptomonnaie, plus cela montre une certaine fiabilité au niveau du projet.

### 2.2.2.5 Tokenomics et inflation

L'analyse de la tokenomics est très importante pour savoir si une cryptomonnaie est inflationniste. Pour ça, je vais vous illustrer la méthodologie qu'il faut avoir pour détecter une cryptomonnaie qui est inflationniste. À ce titre je vais vous montrer l'exemple de Solana.

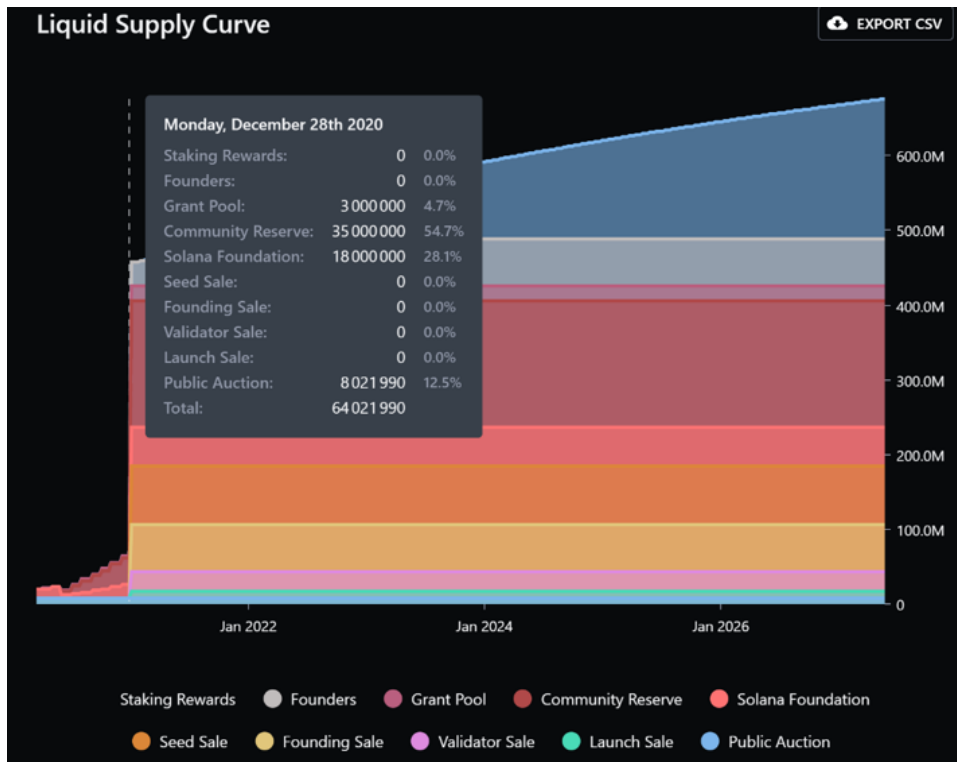
On peut le voir ci-dessous que pour certains investisseurs qui étaient au début du projet, ont eu des Solanas pour une bouché de pains. Cependant, ces jetons ne sont pas distribués immédiatement auprès des investisseurs. Ces Solanas sont donnés peu à peu pendant une période précise et ça s'appelle la supply schedule. Comme on peut le voir sur le graphique suivant :

Figure 1 : Funding Rounds Solana



(<https://messari.io/asset/solana/profile/launch-and-initial-token-distribution>)

Figure 2 : Calendrier de distributions jetons - Solana



(<https://messari.io/asset/solana/profile/launch-and-initial-token-distribution>)

Figure 3 : Graphique Solana



(<https://www.tradingview.com/chart/?symbol=CRYPTOCAP%3ASOL>)

Les jetons ont été débloqué subitement en décembre 2021 et cela a entrainé une hausse du cours. Les premiers investisseurs qui étaient aux prémices peuvent vendre massivement et impacter fortement les autres possesseurs de Solana en faisant chuter les prix. Puisqu'ils sont considérés comme des baleines.

### 2.2.2.6 Network Value to Transactions Ratio (NVT)

Le Network Value to Transactions Ratio est similaire au price-to-earnings ratio mais pour la cryptomonnaie. Elle permet de savoir le nombre de transactions qu'il y a eu en 24h par rapport à sa capitalisation boursière. En effet, en divisant la capitalisation boursière par le volume de transaction en 24 heures, nous obtenons le ratio NVT. Plus ce ratio est élevé, plus cela montre qu'il est surévalué. En moyenne le NVT est à 90-95. S'il est en dessous de cette moyenne, cela montre qu'il est sous-évalué. S'il est au-dessus, cela montre que cet actif est surévalué.

### 2.2.2.7 Market value to realized value ratio. (MVRV)

Pour obtenir le ratio mvr, il faut diviser le market cap par le realized cap. Le realized cap est une métrique qui permet d'avoir la capitalisation boursière réelle. En effet, toutes les personnes qui possèdent du bitcoin ne l'ont pas acheté à un prix précis. Certaines personnes l'ont eu pour 200 \$, d'autres pour 60'000 \$, etc. Elle nous permet aussi d'en prendre en compte les wallets qui sont perdus. C'est pour ça que le realized cap est la quantité \* le prix auquel on a acheté qui s'applique pour toute l'offre en circulation. Le tableau ci-dessous nous permet de comprendre plus facilement cette métrique.

Tableau 3 : Crypto A circulation

Crypto A circulation			
Date	Quantité	Prix à cette date	Total
22-03-2022	4	10.00\$	40.00\$
23-03-2022	2	12.00\$	24.00\$
24-03-2022	1	5.00\$	5.00\$

(Anushan Alvaipillai, 2022)

Le realized cap pour la crypto A est de :  $(40.00\$ + 24.00\$ + 5.00\$) / 7 = 9.86\$$

Maintenant que nous comprenons en quoi consiste le realized cap, on peut continuer sur le ratio MVRV. Pour obtenir le ratio (MVRV), il faut diviser le market cap par le realized cap. En moyenne ce ratio devrait être à 3.7, si ce ratio y est inférieur, cela veut dire que cette cryptomonnaie est sous-évaluée. Si elle est supérieure, cela montre qu'elle est surévaluée.

### 2.2.3 Analyse Sentimentale

Selon Thami Kabbaj, l'analyse sentimentale est une méthode redoutable qui permettrait d'anticiper certains retournements majeurs d'un actif financier. En effet, à la suite d'une bonne nouvelle sur un actif financier, le cours peut expliquer plusieurs raisonnements

Tableau 4 : Analyse sentimentale - Bonne nouvelle

	Tendance haussier	Tendance baissier
Baisse des cours	Cette bonne nouvelle annonce qu'il y aura un retournement baissier, puisque les vendeurs réussissent à gagner contre les acheteurs malgré cette nouvelle positive sur une tendance haussière. Alors le trader devra être prêt à vendre ses positions et à short le marché	C'est un signal très pessimiste. En effet, malgré cette bonne nouvelle, les opérateurs ne sont pas convaincus. Le cours suivra pendant un bon moment sa tendance baissière.
Stabilisation des cours	Malgré cette nouvelle positive, le cours n'arrive pas à grimper. Le trader devra faire attention à ne pas acheter à ce moment-là et se préparera à un retournement baissier.	Là-aussi, le signal est pessimiste. C'est le même raisonnement que pour la tendance baissière. Le trader devra alors se préparer à short le marché lorsqu'il y aura une cassure de la résistance.

Hausse des cours	Cette nouvelle est saine et que cela sous-entend que le trader pourra encore garder encore ses positions.	C'est un signal qui peut définir un retournement haussier. Le trader pourra préparer un plan d'action pour rentrer dans cet actif
------------------	---	---

(Thami Kabbaj, 2015, p.126)

Lorsqu'il y a une mauvaise nouvelle d'un actif financier le cours peut expliquer plusieurs raisonnements :

Tableau 5 : Analyse sentimentale - Mauvaise nouvelle

	Tendance haussier	Tendance baissier
Baisse des cours	C'est un mouvement normal. Suite à cette mauvaise nouvelle, certains opérateurs sortent du marché par peur. Le trader pourra profiter de rentrer en position ou d'accumuler ses achats.	C'est un mouvement logique. Suite à cette nouvelle, le cours suit sa tendance baissière. Le trader devra garder sa position pour continuer à short le marché.
Stabilisation des cours	Cela montre que, malgré cette mauvaise nouvelle, ceci n'affecte pas les opérateurs. Le prochain mouvement du cours sera un mouvement puissant.	Cette une information combinée avec d'autres éléments peut montrer un retournement haussier qui peut subvenir à tout moment. Le trader devra alors se préparer pour rentrer en position lorsqu'il y a une cassure de la résistance
Hausse des cours	C'est un signal très optimiste (bullish) pour les	Cette mauvaise nouvelle annonce qu'il y aura un

	<p>acheteurs. Cela montre que rien ne pourra arrêter cet actif. Mais le trader devra quand même gérer son risque en plaçant son stop-loss pour minimiser son risque.</p>	<p>retournement haussier, puisque les acheteurs réussissent à gagner contre les vendeurs malgré cette mauvaise nouvelle sur une tendance baissière. Alors le trader devra se préparer à acheter des positions.</p>
--	--	--

(Thami Kabbaj, 2015, p.125)

L'analyse sentimentale illustre parfaitement la phrase suivante : Quand tous les médias disent d'acheter, c'est le moment de vendre. En effet, selon Thami Kabbaj, quand tous les médias disent d'acheter, les personnes qui débutent dans le trading, vont rentrer sur le marché, en espérant à faire du gain. Mais c'est ici que les traders expérimentés vont commencer à vendre leur position pour récupérer des gains, puisqu'ils savent que cette tendance haussière s'écroulera à un moment.

On peut constater l'analyse sentimentale permet d'analyser la psychologie et les émotions des personnes qui sont sur le marché. Grâce à cette méthode, on peut comprendre le résultat du cours d'un actif suite à une nouvelle. Grâce à ceci, on peut préparer un plan d'action pour rentrer sur le marché ou sortir du marché.

## 2.2.4 Analyse Technique

Maintenant qu'on a fini avec l'analyse sentimentale, nous pouvons voir le fonctionnement de l'analyse technique.

### 2.2.4.1 Définition

L'analyse technique consiste à étudier le cours des actifs à l'aide de plusieurs outils pour déduire la direction du cours. Pour ça, je vais vous énumérer quelques outils qui sont connus et qui sont utilisés par les traders qui sont adeptes de l'analyse technique.

### 2.2.4.2 Structure

Pour reconnaître si un actif poursuit sur une tendance haussière ou baissière, il existe un outil qu'on nomme structure qui permet de reconnaître cette tendance. Il se fait grâce à des lignes de tendances qui montre que le cours oscille sur cette ligne. Selon Anthony Busière, il faudrait que la ligne soit touchée au moins 2 fois par les points bas pour que

la tendance haussière soit confirmée et 2 fois par les points hauts pour que la tendance baissière soit confirmée. Dès qu'il y a une cassure de cette ligne, cela indique que la partie adverse est en train de prendre la main et que le cours est prêt à changer de direction.

#### 2.2.4.2.1 *Tendance haussière*

Sur le graphique ci-dessous, on peut voir le cours du bitcoin qui est en tendance haussière depuis novembre 2020 à 20 avril 2021. Cette tendance a été inversé lorsque le cours a cassé la ligne de tendance haussière. Ce qui entraîna la chute du prix.

Figure 4 : Bitcoin - Tendance haussière



(<https://www.tradingview.com/chart/?symbol=BINANCE%3ABTCUSDT>)

#### 2.2.4.2.2 *Tendance Baissière*

Sur le graphique ci-dessous, on peut voir le Bitcoin qui est en tendance baissière sur de moyen terme depuis le 10 novembre 2021 jusqu'à aujourd'hui. L'inversion de cette tendance s'effectuera peut-être lorsqu'il y aura la cassure de cette ligne de tendance baissière.

Figure 5 : Bitcoin - Tendance Baissière



(<https://www.tradingview.com/chart/?symbol=BINANCE%3ABTCUSDT>)

### 2.2.4.2.3 Sans tendance

Il existe aussi un dernier type de cas qui définit la tendance d'un actif qui s'appelle la stagnation. En effet, la stagnation définit que l'actif n'a aucune tendance. Les acheteurs et les vendeurs n'arrivent pas à se décider sur quelle direction va aller le prix. Du coup, le prix de l'actif reste bloqué dans une sorte de rectangle.

Ce rectangle est défini par 2 points hauts qui doivent être touchés par une ligne de tendance en hauts et par 2 points bas qui doivent être touchés par une ligne de tendance en bas.

Lorsqu'il y a une cassure de cette la ligne de tendance par le haut, cela indique que les acheteurs sont en train de prendre la main. Si c'est par le bas, cela indique que ce sont les vendeurs qui prennent la main. Comme on peut le voir sur le graphique ci-dessous, on voit Ethereum qui était en période de stagnation depuis octobre 2018 jusqu'à 20 juillet 2020, qui après avoir cassé la ligne de tendance par le haut, le prix a fortement augmenté.

Figure 6 : Ethereum - sans tendance



(<https://www.tradingview.com/chart/?symbol=BINANCE%3ABTCUSDT>)

### 2.2.4.3 Indicateur technique

En dehors de ces lignes de tendance, il existe d'autres outils comme les indicateurs pour parfaire l'analyse. En effet, les indicateurs techniques sont des formules mathématiques qui sont souvent visualisées en graphique par les traders et qui permettent de voir si le marché est en surachat ou en sous-achat. Ci-dessous, vous retrouverez quelques indicateurs techniques les plus connus, ainsi que la façon dont ils fonctionnent.

### 2.2.4.3.1 RSI (Relative strength Index)

Le RSI est un indicateur qui permet de savoir si le prix de l'actif est en surachat ou sous-achat. Il a un paramètre qui est défini dans sa formule qui est le nombre de périodes. En effet, le nombre de périodes peut être variable selon la périodicité, mais la valeur reste la même. Mais généralement, le nombre de périodes reste à 14. Ci-dessous, vous retrouverez la formule du RSI.

$$RSI = 100 - \left[ \frac{100}{1 + \frac{\text{Nombre moyen de gain sur } x \text{ période}}{\text{Nombre moyen de perte sur } x \text{ période}}} \right]$$

Cette formule nous montre qu'il est facile de calculer le RSI. Le fonctionnement est simple, lorsqu'il y a beaucoup plus de gains sur la période x, le RSI indique qu'il est en surachat. Jason Fernando indique que le RSI doit être supérieur à 70 pour considérer que l'actif est en surachat. S'il y a beaucoup plus de pertes sur x période, cela veut dire que l'actif est en sous-achat. Jason Fernando indique que ce chiffre doit être inférieur à 30 pour considérer la sous-évaluation. Il existe encore plein de thèmes à aborder avec le RSI comme les divergences, mais ce ne pas l'objectif de ce travail de recherche. Il s'agit de juste montrer un petit aperçu sur le trading pour ensuite créer le Bot.

Figure 7 : RSI - Surachat et survente



(<https://www.ig.com/en/trading-strategies/a-trader-s-guide-to-the-relative-strength-index--rsi--190320>)

### 2.2.4.3.2 MACD (Moving Average Convergence Divergence)

Le MACD est un indicateur qui permet de savoir s'il y a une divergence entre 2 Exponential moving average (EMA) qui sont sur des périodes différentes. Exponential moving average se calcule de la manière suivante avec un paramètre qu'il faut renseigner qui est l'unité de temps :

Figure 8 : Calcul EMA

$$EM A_t = \left[ V_t \times \left( \frac{s}{1+d} \right) \right] + EM A_y \times \left[ 1 - \left( \frac{s}{1+d} \right) \right]$$

**where:**

$EM A_t$  = EMA today

$V_t$  = Value today

$EM A_y$  = EMA yesterday

$s$  = Smoothing

$d$  = Number of days

(<https://www.investopedia.com/terms/e/ema.asp>)

Une fois qu'on a calculé le Exponential moving average sur une période de 12 et le exponential moving average sur une période de 26, on peut calculer le MACD qui s'effectue de la manière suivante : 12-Period EMA – 26-Period EMA

À côté de ça, on couple un signal line qui est l'EMA sur une période de 9 avec le MACD. Ce qui nous permet d'avoir un signal acheteur ou vendeur. Par exemple si le MACD est supérieur au signal line, James Chen nous indique que les acheteurs ont pris la main. Si elle est en dessous, cela veut dire que ce sont les vendeurs ont pris la main. Ci-dessous, vous retrouverez un graphique sur le bitcoin qui illustre parfaitement ce que j'ai écrit.

Figure 9 : EMA - prise de positions



(<https://www.tradingview.com/chart/?symbol=BINANCE%3ABTCUSDT>)

### 3. Bot de trading

Dans cette partie, nous allons voir le fonctionnement des différents bots de trading de cryptomonnaie qui existe sur le marché. Mais avant ça, il faut tout de même comprendre ce que c'est un bot de trading.

#### 3.1 Définition

Un Bot de trading est un robot qui est programmé par un programme informatique pour exécuter des trades 24h sur 24, 7 jours sur 7. L'objectif du robot est d'appliquer la stratégie du trader de façon automatique et ininterrompue.

#### 3.2 Pourquoi choisir un Bot de trading

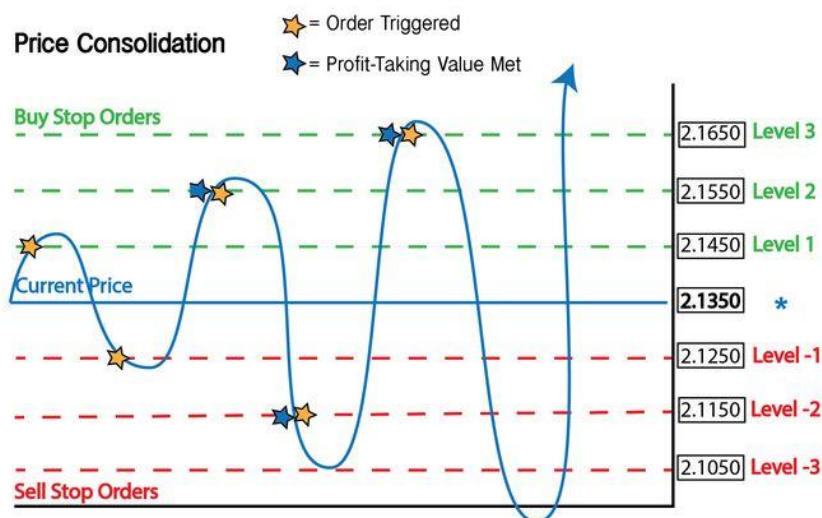
L'avantage du Bot de trading est que le robot tourne sans arrêt, même quand on dort. Aussi, nous n'avons plus besoin de rester sur l'écran pour surveiller le marché pour attendre une opportunité. On peut utiliser ce temps pour faire autre chose. De plus, nous n'avons plus besoin de subir cette anxiété et cette frustration lorsqu'on fait des trades perdants, puisque le Bot n'a pas d'émotions. Il est important de se rappeler qu'aucune stratégie dans le monde a un winrate de 100%. Cela n'existe pas.

#### 3.3 Différents Bots de trading et stratégie

Sur Kucoin, il existe 2 différents types de Bot de trading. La première est le grid trading :

##### 3.3.1 Grid trading

Figure 10 : Stratégie Grid



(<http://www.forex-central.net/grid-trading-strategy.php>)

Le grid trading fonctionne de la manière suivante. La ligne bleue qui est au milieu est le prix de référence. Les 3 lignes rouges sont là où on va acheter de la cryptomonnaie. Les 3 lignes vertes illustrent les prix où on va vendre les cryptomonnaies qu'on a achetées. Le nombre de lignes définit le nombre de parts qu'on alloue pour chaque achat par rapport à notre capital de départ. Ici, dans notre exemple, on allouerait 1/3 de notre capital pour chaque achat. L'objectif de cette stratégie est d'acheter les creux du marché pour les revendre aux prix où se situe sur les lignes vertes. Inconvénient : Si le prix chute, il sera difficile de vendre au plus haut.

### **3.3.2 DCA trading**

La deuxième est le DCA (dollar cost averaging) qui fonctionne de la manière suivante. C'est d'acheter une cryptomonnaie régulièrement malgré la hausse ou la baisse du prix. Pour ça, l'utilisateur paramétera les éléments suivants : la périodicité, un montant fixe de la paire (souvent l'usdt) qu'on souhaite échanger et la cryptomonnaie qu'on souhaite acheter pendant chaque période. Cependant, il n'y a aucune condition de vente. L'objectif de cette stratégie est d'accumuler un maximum les jetons et de les vendre selon l'envie de l'utilisateur. Cette stratégie est intéressante pour les personnes qui croient aux fondamentaux du projet.

### **3.3.3 B-Cube Ai**

B-Cube AI propose à leur client un Bot qui utilise de l'intelligence artificielle. En effet, ce Bot a été entraîné au préalable avec des anciens tweets et anciennes publications de Facebook. Leur objectif est d'analyser les publications concernant les cryptomonnaies en temps réel pour savoir si le cours va monter ou descendre. Si la plupart des gens sont positifs sur une cryptomonnaie, il va acheter cette cryptomonnaie. Si les gens sont négatifs, il va vendre ces cryptomonnaies.

### **3.3.4 Constat**

Comme on peut le voir ci-dessus, les Bots qui sont proposés par kucoin sont assez simple au niveau de la logique. Des simples conditions qui tournent en continu peuvent être établies pour atteindre l'objectif de ces Bots. Pourtant, un grand nombre d'utilisateur utilisent ces Bots pour espérer faire du profit. Cependant, pour le cas de B-Cube Ai, le fonctionnement est beaucoup plus complexe et demande beaucoup plus de connaissances et de ressources matérielles pour atteindre les mêmes objectifs. Cependant, le Bot n'utilise aucune analyse technique, ni d'analyse fondamentale. Il utilise simplement de l'analyse sentimentale.

## 4. Construire son Bot de trading

Maintenant que nous avons vu toutes les informations importantes pour être un bon trader. Nous allons passer à la création de notre propre bot de trading et qui se passe de la manière suivante :

### 4.1 Comment obtenir les cours d'un crypto dans le passé ?

Apprendre du passé, c'est minimiser le risque. C'est pour ça qu'il est important d'avoir les données du passé pour vérifier que notre stratégie fonctionne bien. Ça s'appelle le Backtesting. Je détaillerais le Backtesting sur le point Backtesting Pour obtenir les cours d'une cryptomonnaie dans le passé, on peut le faire de plusieurs manières. Mais la solution qui est la plus facile est de passer par la librairie de CCXT. En effet cette librairie permet d'avoir l'historique des cours des cryptomonnaies sur plusieurs échanges. Il y en a qui sont certifiés « CCXT Certified » comme Binance, FTX, AcendEx, etc. Ces échanges nous suffiront amplement pour simuler notre stratégie.

Pour importer les OHCLV (Open, High, Close, Low, Volume), nous allons créer un fichier Javascript qui va nous permettre d'importer l'historique des données sur les cryptos qu'on souhaite importer de manière asynchrone. Ce qui veut dire que le programme va envoyer toutes les requêtes d'un coup à CCXT, puis il recevra toutes les réponses de CCXT.

L'avantage de l'asynchronicité est que c'est beaucoup plus rapide que la synchronicité. Une fois que le programme a reçu cette donnée, le programme lira une par une la donnée pour la retransmettre sur un fichier CSV.

Ci-dessous, je vais vous expliquer quelques parties de la fonction principale du programme. Cette fonction principale a pour but d'obtenir les données, puis de les retransmettre dans un fichier CSV.

Cette partie va nous servir d'avoir une liste de date depuis la date de début qu'on a mis en paramètre.

Figure 11 : Instanciation du tableau

```
let exchange_name = exchange.name;
console.log(pair_name, exchange_name, timeframe, since_date);
let starting_date = date_to_timestamp(since_date);
let now = current_utc_date();
let tf_array = [starting_date];
let last_tf = starting_date;
let result_ohlcv = [];
let current_request = 0;

while (last_tf < now) {
  last_tf = last_tf + (limit) * tf_ms;
  if (last_tf < now) {
    tf_array.push(last_tf);
  }
}
let total_request = tf_array.length;
```

(Anushan Alvaipillai, 2022)

Cette liste de date va nous permettre de faire une boucle pour obtenir le prix d'une paire à telle date qu'on va ensuite concaténer avec la liste result\_ohlcv. S'il y a une erreur, on réessaye de faire jusqu'à 2 fois. S'il y a une troisième erreur, on indique sur la console.

Figure 12 : Ajouts des OHCVL

```
for (const tf in tf_array) {
  // console.log(tf_array[tf])
  exchange.fetchOHLCV(symbol = pair_name, timeframe = timeframe, since = tf_array[tf], limit = limit).then(resp => {
    result_ohlcv = result_ohlcv.concat(resp);
    current_request++;
  }).catch(err => {
    console.log("Error retrieving candles since", tf_array[tf], exchange_name, pair_name, timeframe);
    exchange.fetchOHLCV(symbol = pair_name, timeframe = timeframe, since = tf_array[tf], limit = limit).then(resp => {
      result_ohlcv = result_ohlcv.concat(resp);
      current_request++;
    }).catch(err2 => {
      console.log("Error retrieving candles since", tf_array[tf], exchange_name, pair_name, timeframe);
      exchange.fetchOHLCV(symbol = pair_name, timeframe = timeframe, since = tf_array[tf], limit = limit).then(resp => {
        result_ohlcv = result_ohlcv.concat(resp);
        current_request++;
      }).catch(err3 => {
        console.log(err2);
        console.log("!! Fatal Error !!", pair_name, timeframe);
        current_request++;
      })
    })
  })
}
```

(Anushan Alvaipillai, 2022)

Une fois que le programme a bien chargé tous les prix d'une paire (exemple → BTC/USDT), le programme va trier par ordre croissant la matrice par date. Puis le programme va paramétrer le csv avec le nom du fichier avec le nom de la paire. Aussi, il lui attribue les titres. Finalement, on ajoute les données du result\_ohlcv, sans les doublons, dans le fichier CSV.

Figure 13 : Exportation fichier CSV

```
const delay = millis => new Promise((resolve, reject) => {
  setTimeout(_ => resolve(), millis);
});
while (current_request < total_request) {
  process.stdout.write(`\rLoading ${current_request}/${total_request} requests | ${result_ohlcv.length} candles loaded`);
  await delay(2000);
}
process.stdout.write(`\rLoading ${current_request}/${total_request} requests | ${result_ohlcv.length} candles loaded`);
result_ohlcv = result_ohlcv.sort(function(a, b) {
  return a[0] - b[0];
});
result_ohlcv = eliminate_double_ts(result_ohlcv);

let file_pair = pair_name.replace('/', '-');
let dirpath = './database/' + exchange_name + '/' + timeframe + '/';
let filepath = dirpath + file_pair + ".csv";

let first_date = timestamp_to_date(result_ohlcv[0][0]);

await fs.promises.mkdir(dirpath, { recursive: true });

const csvWriter = createCsvWriter({
  header: ['date', 'open', 'high', 'low', 'close', 'volume'],
  path: filepath
});

csvWriter.writeRecords(result_ohlcv) // returns a promise
  .then(() => {
    process.stdout.write(`\rSuccessfully downloaded ${result_ohlcv.length} candles since ${first_date} in ${filepath}`);
    return true;
  }).catch(err => {
    console.log(err);
    return false;
  });
```

(Anushan Alvaipillai, 2022)

Le format des données du CSV est le suivant :

- Date : Date du prix → Date
- Open : Prix de l'ouverture → Float
- High: Prix le plus haut → Float
- Low: : Prix le plus bas → Float
- Close : Prix de la cloture → Float
- volume: Nombre de volume → Float

## 4.2 Analyse fondamentale – Bot de trading

Pour sélectionner les cryptos qu'on souhaite pour investir, nous allons utiliser l'analyse fondamentale pour sélectionner les pépites. En effet, nous allons utiliser les mêmes techniques qu'on a détaillées sur le point « analyse fondamentale », mais de manière automatique. Par ailleurs, nous allons utiliser les métriques qui sont générales pour chaque cryptomonnaie, comme ça il n'y aura pas d'incohérence dans le classement. Par exemple, nous n'allons pas utiliser le ratio de l'hashrate, parce que toutes les cryptomonnaies n'ont pas d'hashrate. Les métriques sont les suivantes :

- Ratio NVT
- Ratio MRVR
- Ratio average transaction value
- Ratio active addresses

Pour calculer ces ratios de manière automatique pour chaque cryptomonnaie, nous allons utiliser la plateforme de Messari. Messari propose une api gratuite où on peut accéder à plein d'informations fondamentales sur la cryptomonnaie. Sur « **Annexe 1** », vous retrouverez un exemple du bitcoin avec toutes les informations qui sont proposées par Messari. Cependant, sur certaines cryptomonnaies, les informations manquent. C'est pourquoi, dans le programme que j'expliquerai par la suite, je tiendrai compte du manque d'informations qu'on pourrait avoir sur certaines cryptomonnaies.

Pour accéder à l'api de Messari, il faut s'inscrire sur leur site : <https://messari.io/api>

Puis, on reçoit une clé qui permet d'effectuer 30 requêtes par minute, ce qui est largement suffisant pour ce travail de recherche

La première chose que le programme devait faire, c'est d'aller chercher les cryptomonnaie qui sont dans le top 100 par rapport à la capitalisation boursière.

C'est possible grâce à l'api de coinmarketcap. Pour accéder à l'api, il faut seulement s'inscrire pour recevoir les clés. Une fois que c'est fait, on fait une requête pour obtenir les cryptomonnaies qui sont dans le top 100. On peut évidemment demander plus, juste en ajoutant un « query parameter (?limit=quantite) »

Figure 14 : Récupération des cryptomonnaies du top 100

```
def get_list_crypto_top_100(API_KEY):
    headers = {
        'X-CMC_PRO_API_KEY': API_KEY
    }
    list_symbol_100 = []
    # url = 'https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest?limit=500'
    url = 'https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest'
    r = requests.get(url, headers=headers)
    find_ranks = {}
    if r.status_code == 200:
        data = r.json()
        for d in data['data']:
            symbol = d['symbol']
            find_ranks[symbol] = d['cmc_rank']
            list_symbol_100.append(symbol)
    print(list_symbol_100)
    return list_symbol_100
```

(Anushan Alvaipillai, 2022)

Ensuite pour chacune de ses cryptomonnaies, il va chercher les informations nécessaires grâce à l'API de Messari, puis il va ajouter toutes les informations importantes dans un dictionnaire.

Figure 15 : Récupération des informations fondamentales

```
def get_information_about_coin(coin):
    print(coin)
    information_of_coin = {}
    api_key_messari = "API_KEY"
    url = "https://data.messari.io/api/v1/assets/{}/metrics".format(coin)
    requete = urllib.request.Request(url)
    requete.add_header("x-messari-api-key", api_key_messari)
    data = urllib.request.urlopen(requete).read()
    data_decode = json.loads(data.decode('UTF-8'))
    transaction_count = data_decode["data"]["on_chain_data"]["transfer_count_last_24_hours"]
    transaction_value_usd = data_decode["data"]["on_chain_data"]["average_fee_usd"]
    active_addresses = data_decode["data"]["on_chain_data"]["active_addresses"]
    hash_rate = data_decode["data"]["on_chain_data"]["hash_rate"]
    market_cap = data_decode["data"]["marketcap"]["current_marketcap_usd"]
    circulating_supply = data_decode["data"]["supply"]["circulating"]
    volume_last_24_hours = data_decode["data"]["market_data"]["volume_last_24_hours"]
    realized_marketcap_usd = data_decode["data"]["marketcap"]["realized_marketcap_usd"]
    information_of_coin["transaction_count"] = transaction_count
    information_of_coin["transaction_value_usd"] = transaction_value_usd
    information_of_coin["active_addresses"] = active_addresses
    information_of_coin["hash_rate"] = hash_rate
    information_of_coin["circulating_supply"] = circulating_supply
    information_of_coin["market_cap"] = market_cap
    information_of_coin["volume_last_24_hours"] = volume_last_24_hours
    information_of_coin["realized_marketcap_usd"] = realized_marketcap_usd
    print(information_of_coin)
    return information_of_coin
```

(Anushan Alvaipillai, 2022)

Puis il va établir une évaluation pour chaque cryptomonnaie, en calculant chaque ratio qu'on a évoqué précédemment. Par exemple :

- Si le résultat d'un ratio considère que la cryptomonnaie est sous-évalué, le programme va retourner -1.
- Si le programme considère que la cryptomonnaie est surévalué par rapport au ratio, il va retourner 1.
- S'il manque une information pour faire le ratio, il va retourner 0.
- Sinon, il retourne aussi 0.

Vous pouvez retrouver sur « **Annexe 2** » toutes les fonctions pour calculer les ratios.

Une fois que le programme a récupéré les résultats de chaque ratio, il va retourner la cryptomonnaie avec le résultat final.

Figure 16 : Analyse fondamentale d'une cryptomonnaie

```
def get_final_result_coin (coin, btc_information):
    result = 0
    information_of_coin = get_information_about_coin(coin)
    transaction_count = information_of_coin["transaction_count"]
    transaction_value_usd = information_of_coin["transaction_value_usd"]
    active_addresses = information_of_coin["active_addresses"]
    hash_rate = information_of_coin["hash_rate"]
    circulating_supply = information_of_coin["circulating_supply"]
    market_cap = information_of_coin["market_cap"]
    volume_last_24_hours = information_of_coin["volume_last_24_hours"]
    realized_marketcap_usd = information_of_coin["realized_marketcap_usd"]
    result += get_result_NVT(market_cap, transaction_count)
    result += get_result_MRVR(realized_marketcap_usd, market_cap)
    # result += get_result_ratio_hashrate(market_cap, hash_rate, btc_information)
    result += get_result_ratio_avg_transaction_value(market_cap, transaction_value_usd, btc_information)
    result += get_result_ratio_active_addresses(market_cap, active_addresses, btc_information)
    return {"coin": coin, "result": result}
```

(Anushan Alvaipillai, 2022)

Après avoir obtenu le résultat final pour chaque cryptomonnaie, on peut enfin avoir le classement avec les autres cryptomonnaies qui sont considérées comme sous-évaluées. Pour ça le programme fait un tri sur le résultat de façon ascendante.

Figure 17 : Classement fondamental des cryptomonnaies

```
def get_crypto_fundamental_classement(total_list_crypto):
    classement_decroissant = sorted(total_list_crypto, key=lambda d: d['result'])
    print(classement_decroissant)
    return classement_decroissant.reverse()
```

(Anushan Alvaipillai, 2022)

Voici un aperçu du résultat obtenu, on peut voir qu'il n'y a pas trop de démarcations, puisque Messari ne propose pas toutes les informations pour chaque crypto. Ce qu'il fait que beaucoup de cryptomonnaie ont un résultat de 0, puisqu'il y a aucune information qu'on souhaite avoir cette cryptomonnaie. C'est pourquoi, nous allons utiliser l'analyse sentimentale, pour ajouter des nouveaux résultats pour que le classement soit plus juste.

## 4.3 Analyse Sentimental

Pour faire suite à l'analyse fondamentale, nous allons évaluer le sentiment du marché grâce à Twitter et à google News. En effet, nous allons utiliser une intelligence artificielle qui va nous permettre de savoir si un article ou un tweet est positif ou négatif. Cette intelligence artificielle provient de la librairie « transformers » délivré par Hugging Face. Hugging Face propose gratuitement des intelligences artificielles qui sont déjà entraînées pour diverses situations. Il propose des IA pour la classification d'image, la détection d'objets, la réponse aux questions, etc. Mais nous ce qui nous intéresse, c'est de savoir si un texte est positif ou pas. Pour ça ils ont un modèle « sentiment analysis » qui nous permet d'analyser le sentiment d'un texte.

Mais avant d'analyser le sentiment d'un texte, il nous faut la donnée brute du texte, pour ça nous allons utiliser l'api de twitter et de google news. De ce fait, nous allons procéder aux étapes suivantes :

### 4.3.1 Twitter

Pour accéder à l'api de twitter, il faut tout d'abord s'inscrire sur le lien : <https://developer.twitter.com/en> en cliquant sur le bouton « sign up » pour avoir accès à l'api de base. L'api de base permet de faire beaucoup moins de requêtes que par rapport à l'api dédié pour les étudiants. J'ai fait les démarches pour avoir accès à l'api pour étudiant, mais ils m'ont dit que je n'étais pas éligible à l'api pour étudiant. Sur « **Annexe 3** », vous retrouverez le mail de refus pour obtenir l'api étudiante.

Une fois qu'on a nos clés pour l'api, on peut installer la librairie tweepy qui va nous permettre de faciliter les requêtes api. Une fois que c'est fait, on peut faire un « search\_recent\_tweets » avec en paramètre le mot-clé qui va nous permettre d'avoir accès aux 100 derniers tweets dans le monde qui contiennent le mot-clé. J'aurais aimé avoir plus de tweets récents et ciblé les pays, mais l'api de base ne me le permet pas. Une fois qu'on a les 100 tweets, on met toutes ces données dans un dataframe grâce à « pandas ».

Figure 18 : Obtenir les récents tweets d'une cryptomonnaie

```
client = tweepy.Client(bearer_token=config.BEARER_TOKEN)
query = crypto + " crypto"
response = client.search_recent_tweets(query=query, max_results=100)
ma_list = response.data
df = pd.DataFrame([str(tweet) for tweet in ma_list], columns=["Tweets"])
df.head()
```

(Anushan Alvaipillai, 2022)

La prochaine étape c'est de nettoyer les tweets pour faire en sorte de faciliter la tâche à l'intelligence artificielle. L'objectif est de passer à un tweet qui contient uniquement des mots et des espaces.

De passer de ça : "RT @Bitboy\_Crypto: They always say those of us who made money in crypto "got lucky".\n\nLet me ask you:\n\nDO YOU FEEL LUCKY RIGHT NOW?\n\n#bitcoin"

A ça : "De passer à ça"RT They always say those of us who made money in crypto "got lucky"Let me ask you DO YOU FEEL LUCKY RIGHT NOW?"

Pour ça nous allons utiliser une fonction qui va nettoyer le tweet, il va nous permettre globalement de :

- Enlever les mentions
- Enlever les #
- Enlever les RTs
- Enlever les hyper liens
- Enlever les sauts à la ligne
- Enlever les doubles points

Figure 19 : Nettoyer le texte

```
def cleanTxt(text):  
    text = re.sub(r'@[A-Za-z0-9]+', '', text)#enleve les mentions  
    text = re.sub(r'#', '', text)#enleve les #  
    text = re.sub(r'RT[\s]+', '', text)#enleves les rt  
    text = re.sub(r'https?:\/\/\S+', '', text)# enleve les hyper liens  
    text = re.sub(r'\n', '', text)# enleve les sauts à la ligne  
    text = re.sub(r':', '', text)# enleve les doubles points  
    return text
```

(Anushan Alvaipillai, 2022)

### 4.3.2 News api

Pour accéder à l'api gratuite de google news, il faut s'inscrire <https://newsapi.org/s/google-news-api> en cliquant sur le bouton « Get API key ». Une fois qu'on a la clé pour l'api, on peut effectuer une requête qui nous permet d'avoir les 20 articles les plus populaires qui contiennent le mot-clé qu'on a paramétré. Une fois que le programme a les textes, il met tout ça dans un dataframe et qui va ensuite appliquer la fonction « cleanTxt » pour avoir un texte propre.

(Anushan Alvaipillai, 2022)

Figure 20 : Récupérer les articles d'une cryptomonnaie

```
query = crypto + " crypto"
start_date = datetime.today().strftime('%Y-%m-%d')
url = ('https://newsapi.org/v2/everything?'
      'q=' + query + '&'
      'from=' + start_date + '&'
      'sortBy=popularity&'
      'apiKey=' + api_key_google)
response = requests.get(url)
dict_reponses = response.json()
articles = dict_reponses['articles']
```

### 4.3.3 Analyse de texte

Maintenant qu'on a les données brutes, on va pouvoir enfin analyser le sentiment d'un texte, pour ça on va importer le module pipeline « sentiment-analysis » de transformers qui va nous permettre d'avoir une intelligence artificielle pour analyser le sentiment de nos textes.

Figure 21 : Prise de la pipeline - sentiment-analysis

```
from transformers import pipeline
classifier = pipeline('sentiment-analysis')
```

(Anushan Alvaipillai, 2022)

Une fois que ça s'est fait, le processus est très simple, on laisse l'IA faire le travail et on utilise une fonction qui récupère l'analyse grâce au label. Le label indique si le sentiment est positif ou négatif.

Figure 22 : Obtenir le sentiment du texte

```
def getAnalysisTransformers(text):  
    return classifier(text)[0]['label']
```

(Anushan Alvaipillai, 2022)

Une fois que c'est fait, on pourra en tenir compte pour l'analyse sentimentale de la cryptomonnaie.

#### 4.3.4 Analyse sentimental final cryptomonnaie

Une fois qu'on a analysé tous les textes, il faut que les 2 parties (twitter et google news) aient un ratio de label « positive » plus avantageux ou qu'ils aient un ratio négatif d'au moins 80% pour passer à l'étape suivante. Chaque partie compte 50 - 50. Il faut que les 2 parties aient plus de textes (article, tweet) positifs ou un ratio de négatif d'au moins 80%. Parce que comme on l'a vu avec Thami Kabbaj quand les gens sont très pessimistes, c'est le moment où les traders expérimentés rentrent sur le marché. J'ai choisi de faire 50 – 50. Puisque sur twitter, on trouve de tout, mais on récupère beaucoup plus de textes. Alors que sur l'api de google news, on a moins de textes, mais les articles sont plus pertinents.

### 4.4 Analyse technique

Après qu'on ait les 2 analyses qui sont l'analyse sentimentale et l'analyse fondamentale. Il nous reste plus que l'analyse technique. En effet, cette analyse sera notre dernier filtre pour acheter une cryptomonnaie et pour la vendre par la même occasion. Pour ça nous allons utiliser une librairie principale qui est la suivante :

#### 4.4.1 Librairie TA

Cette librairie a besoin des données brutes qui est le DOHLCV (Date, Open, High, Low, Close, Volume) d'une paire pour qu'elle fonctionne. Avec ces données, on peut construire n'importe quel indicateur technique. Cependant, nous n'allons pas coder un par un nous-mêmes. Nous allons plutôt utiliser cette librairie qui va nous permettre d'avoir plus de 130 indicateurs pré-faits que nous utiliserons pour implémenter notre stratégie.

## 4.4.2 Stratégie d'analyse

Une fois qu'on a importé, dans un dataframe, les données de ces 9 cryptomonnaies qui ont été sélectionné par l'analyse fondamentale. Puis être validé par l'analyse fondamentale, nous pouvons procéder à l'analyse technique qui se déroulera de la manière suivante :

Tout d'abord, pour notre stratégie, nous allons utiliser 3 indicateurs techniques qui sont :

### 4.4.2.1 Awesome Ocillator

Awesome Ocillator calcule la différence entre le simple moving average d'une période de 5 à une simple moving average d'une période de 34.

$$\text{MEDIAN price} = (\text{High} + \text{Low}) / 2$$

$$\text{AO} = \text{SMA}(\text{Median Price}, 5) - \text{SMA}(\text{Median Price}, 34)$$

L'Awesome Ocillator permet de déterminer la force d'une partie (acheteur-vendeur) qui est à l'origine du mouvement. Elle va nous permettre de mesurer la tendance du marché sur le moyen et long terme. Si l'Awesome Ocillator est supérieur à 0, c'est les acheteurs qui ont pris la main. S'il est inférieur à 0, ce sont les vendeurs qui ont pris la main.

### 4.4.2.2 Stochastic RSI

Le « Stochastic RSI » est la version stochastique du RSI. En effet, cet indicateur utilise le RSI pour faire son calcul. Le calcul fonctionne de la manière suivante :

$$\text{Stoch RSI} = (\text{RSI actuel} - \text{RSI le plus bas de la période mise en paramètre})$$

Cet indicateur va déterminer la force d'une partie (acheteur-vendeur) sur le court / moyen terme. Lorsque le Stochastic RSI est sous-évalué, il se trouvera généralement inférieur à 20. S'il est surévalué, il se trouvera supérieur à 80.

### 4.4.2.3 William %R

William %R est un indicateur de type Momentum qui va de 0 à -100 et qui mesure la surachat et la survente, généralement sur le court terme. Il est calculé de la manière suivante :

Figure 23 : Formule Willam % R

$$\text{Williams \% R} = \frac{\text{Le plus élevé} - \text{proche}}{\text{Le plus élevé} - \text{Plus bas Bas}}$$

où

Le plus élevé = Prix le plus élevé du lookback période, généralement 14 jours.

proche = Cours de clôture le plus récent.

Plus bas Bas = Prix le plus bas du lookback période, généralement 14 jours.

(<https://www.investopedia.com/terms/w/williamsr.asp>)

Pour déterminer la surachat, il faut que le William % R soit < -85. Pour la survente, il faut que l'indicateur soit supérieur à -10.

Après avoir choisi les indicateurs techniques qu'on souhaite implémenter, on peut les importer dans notre programme grâce à la librairie TA en indiquant les paramètres qu'on souhaite lui passer comme la donnée brute et les paramètres pour la période. Finalement, le programme stocke les résultats dans la bonne ligne du dataframe.

Figure 24 : Placement des indicateurs techniques dans le data frame

```
# -- Indicator variable --
aoParam1 = 6
aoParam2 = 22
stochWindow = 14
willWindow = 14

# -- Drop all columns we do not need --
for df in dfList:
    # -- Drop all columns we do not need --
    df.drop(columns=df.columns.difference(['open','high','low','close','volume']), inplace=True)

    # -- Indicators, you can edit every value --
    df['AO'] = ta.momentum.awesome_oscillator(df['high'],df['low'],window1=aoParam1,window2=aoParam2)

    df['STOCH_RSI'] = ta.momentum.stochrsi(close=df['close'], window=stochWindow)

    df['WillR'] = ta.momentum.williams_r(high=df['high'], low=df['low'], close=df['close'], lbp=willWindow)

print("Indicators loaded 100%")
dfList[0]
```

([https://github.com/CryptoRobotFr/cBot-Project/blob/main/live\\_strategy/big\\_will\\_v2\\_live.py](https://github.com/CryptoRobotFr/cBot-Project/blob/main/live_strategy/big_will_v2_live.py))

### 4.4.3 Condition d'achats et de ventes

Les conditions d'achats et de ventes se présentent de la manière suivante :

Condition d'achat :

Pour provoquer l'achat, il faut que l'Awesome Oscillator soit supérieur ou égal à 0 et que le William %R soit inférieur à -85.

Figure 25 : Condition d'achat technique

```
# -- Condition to BUY market --
def buyCondition(row):
    if (
        row['AO'] >= 0
        and row['WillR'] < -85
    ):
        return True
    else:
        return False
```

([https://github.com/CryptoRobotFr/cBot-Project/blob/main/live\\_strategy/big\\_will\\_v2\\_live.py](https://github.com/CryptoRobotFr/cBot-Project/blob/main/live_strategy/big_will_v2_live.py))

Conditions de vente

Pour provoquer la vente d'une cryptomonnaie, il faut que l'Awesome Oscillator soit inférieur à 0, le Stochastic RSI soit supérieur à 0.2 et que le William %R soit supérieur à -10.

Figure 26 : Condition de vente technique

```
# -- Condition to SELL market --
def sellCondition(row):
    if (
        (row['AO'] < 0
         and row['STOCH_RSI'] > 0.2)
        or row['WillR'] > -10
    ):
        return True
    else:
        return False
```

([https://github.com/CryptoRobotFr/cBot-Project/blob/main/live\\_strategy/big\\_will\\_v2\\_live.py](https://github.com/CryptoRobotFr/cBot-Project/blob/main/live_strategy/big_will_v2_live.py))

#### **4.4.4 Constat**

On peut voir que c'est une stratégie qui est facile à implémenter. C'est une stratégie que j'ai vue sur un github et qui s'appelle la stratégie Big will 2. Cependant chacun peut implémenter sa propre stratégie. Il faudra juste modifier les fonctions que je viens de vous montrer avec ses propres indicateurs et ses propres conditions pour que le Bot se mette à jour.

## 4.5 Exchange FTX

Pour effectuer les achats et les ventes, il nous faut un exchange, pour ça nous allons utiliser FTX comme exchange. L'avantage de FTX est qu'il nous propose une api pour effectuer des achats et des ventes. Aussi, c'est l'exchange qui propose le moins de commission (fees) par rapport à ses concurrents et c'est un exchange qui est fiable. Les commissions pour l'achat au marché sont de 0,075%. Sur « **annexe 4** », vous retrouverez un tableau sur les pourcentages de commissions que prennent les exchanges.

Pour accéder à l'api, il faut tout d'abord s'inscrire sur FTX et remplir le formulaire sur le lien suivant : <https://ftx.com/onboarding/signup>

Une fois qu'on a rempli le formulaire, il faut faire la KYC (Know Your Customer) pour vérifier notre identité.

Ensuite on peut importer la librairie python « SpotFtx », qui va nous permettre d'effectuer les transactions. Pour ça, il faut d'abord s'authentifier à l'aide de la librairie :

Figure 27 : Clés personnelles FTX pour passer les ordres

```
ftx = SpotFtx(  
    apiKey='',  
    secret='',  
    subAccountName='')  
)
```

(Anushan Alvaipillai, 2022)

Après ceci, le programme peut établir un achat grâce la fonction « `convert_price_to_precision` ». Elle se structure de la manière suivante :

```
buyPrice = float(ftx.convert_price_to_precision(symbol, prix_demandé))
```

Pour effectuer les ventes, le programme va utiliser la fonction « `place_market_order` » Elle se structure de la manière suivante :

```
sell = ftx.place_market_order(symbol, 'sell', les_paires(xxx/usdt))
```

## 4.6 Notifications – Telegram

Les notifications des prises de positions par le Bot peuvent être possible grâce à Telegram. En effet, Telegram permet à ses utilisateurs de créer leur propre Bot de message. Pour ça, ils ont mis à disposition le Bot « theBotFather » pour envoyer des messages. Pour le configurer, il faut le faire à la main et cela se passe de la manière suivante :

- Installer Telegram sur le téléphone et s'inscrire.
- Se connecter à Telegram web en scannant le QR code grâce à l'application Telegram depuis le téléphone.
- Cliquer sur le lien suivant : <https://telegram.me/BotFather>, et cliquer sur le open in browser.
- Cliquer sur start et envoyer dans la conversation le message : « /newBot ».
- Choisir un nom pour le Bot qui se termine par Bot et qui n'a pas été pris.
- Exécuter la commande « pip install telegram-send » dans le terminal
- Exécuter la commande telegram-send --configure dans le terminal, puis coller le token envoyé par « BotFather » pour recevoir le lien de notre Bot et le mot de passe.
- Cliquer sur le lien envoyé par « BotFather », cliquer sur open in browser et start, insérer le mot de passe envoyé par « BotFather ».

Félicitation la configuration est terminé.

Figure 28 : Message de félicitations pour avoir terminé la configuration



(<https://web.telegram.org/k/>)

Maintenant, il n'y a plus qu'à configurer notre Bot de trading sur le programme pour qu'il puisse envoyer des messages lors de ses prises des positions. Pour qu'il puisse envoyer des messages, le programme devra utiliser la librairie « telegram-send ». Pour ça, il

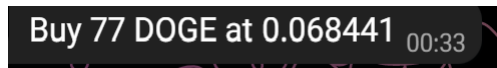
faudra d'abord importer la librairie sur la machine. Puis, faire en sorte que le Bot envoie le bon message. Comme ceci :

```
import telegram_send
telegram_send.send(messages="Message qu'on souhaite envoyé")
```

Voici un aperçu des différentes notifications qu'on va recevoir :

Lorsqu'il y a un achat :

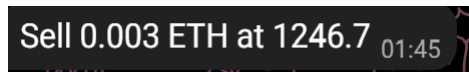
Figure 29 : Bot - Achat



(<https://web.telegram.org/k/>)

Lorsqu'il y a une vente :

Figure 30 : Bot - Vente



(<https://web.telegram.org/k/>)

## 4.7 Résumé fonctionnement du Bot de trading

Maintenant qu'on a vu les 3 parties qui vont structurer notre Bot de trading, nous allons voir comment les parties vont communiquer les unes avec les autres.

Chaque mois, le Bot de sélection va rechercher sur Coinmarketcap, les 100 premières cryptomonnaies qui sont classées au niveau de la capitalisation boursière. Pour chaque cryptomonnaie qui sont dans le top 100, il va rechercher sur Messari toutes les informations nécessaires qui vont lui permettre de faire l'analyse fondamentale. Dès qu'il a reçu les informations, il va établir la phase de test au niveau de l'analyse fondamentale pour déterminer si la cryptomonnaie est sous-évaluée ou surévaluée avec son résultat. Une fois que le Bot a attribué le résultat fondamental pour chaque cryptomonnaie, il va établir un classement pour sélectionner les 9 cryptomonnaies qui sont les plus sous-évaluées. Ensuite, il va placer ces cryptomonnaies dans un fichier texte.

Chaque heure, le Bot de trading va chercher les 9 cryptomonnaies dans le fichier texte. Pour chaque cryptomonnaie, il va rechercher les 100 récents tweets à propos de ce cette cryptomonnaie. Chaque tweet va être nettoyé avant de passer par une analyse sentimentale pour déterminer si ce texte est positif ou non. Une fois que les 100 tweets sont analysés, il va déterminer l'analyse sentimentale finale de ces textes.

La même chose pour l'api de google news, pour chaque cryptomonnaie, il va rechercher les 20 articles les plus pertinents qui parle de cette cryptomonnaie. Chaque article va être nettoyé avant de passer par une analyse sentimentale pour déterminer si ce texte est positif ou non. Une fois que les 20 articles sont analysés, il va déterminer l'analyse sentimentale finale de ces textes.

Puis, pour chaque cryptomonnaie, il va rechercher les 210 derniers historiques OHCLV de la cryptomonnaie sur un Exchange grâce à CCXT. Il va établir les indicateurs techniques grâce aux OHCLV, puis va les stocker dans un dataframe. Ensuite, il va voir pour déclencher un achat ou une vente.

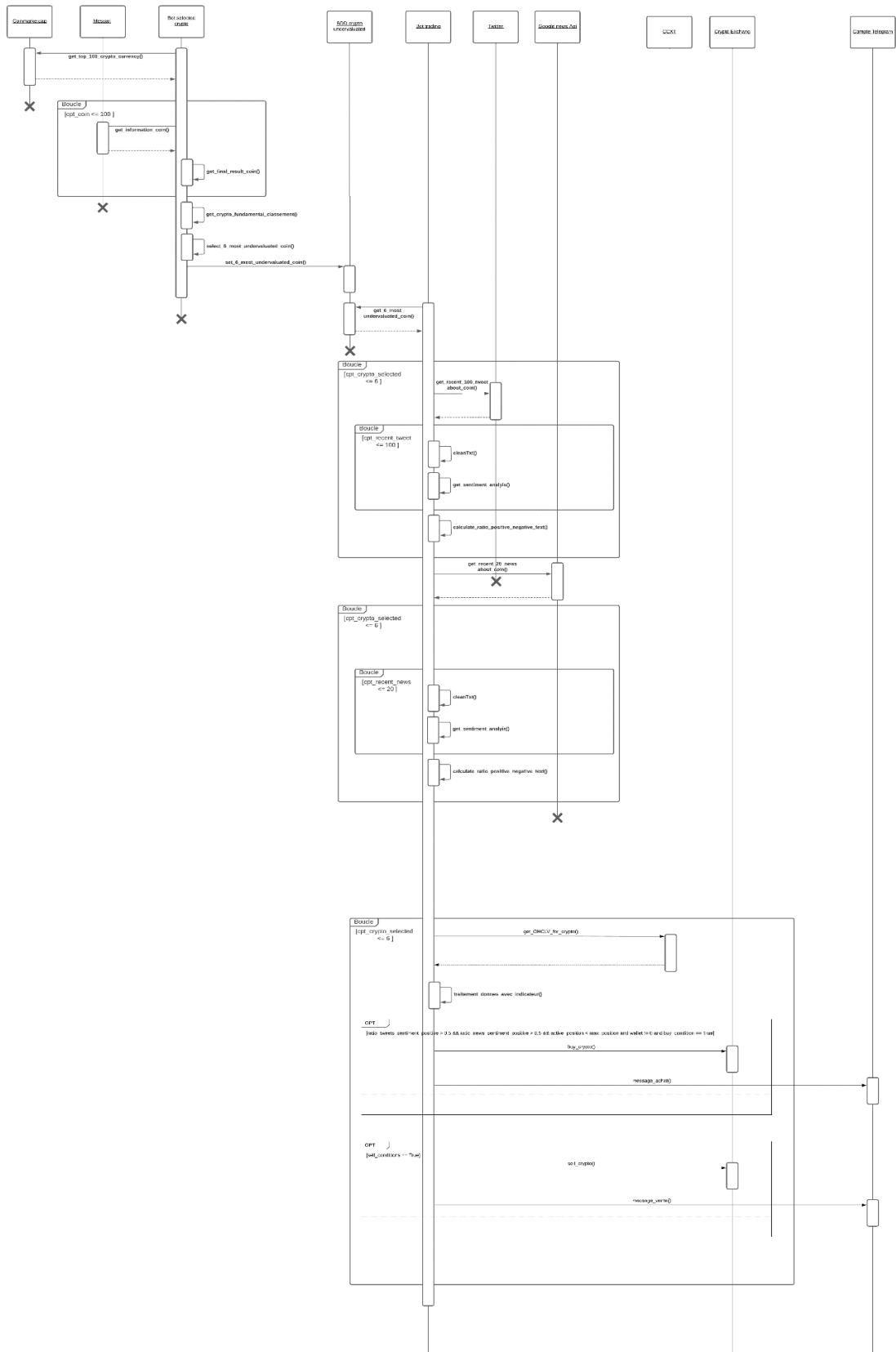
Pour déclencher l'achat, il faut que l'analyse sentimentale de twitter soit validée ET l'analyse sentimentale de Google News soit validée ET la position active soit inférieur à la position maximum ET le portefeuille ne soit pas égal à 0 ET la condition d'achat technique soit validée.

Pour déclencher la vente, il faut que le cryptomonnaie soit dans la liste de cryptomonnaies achetées ET la condition de vente technique soit validée.

Lorsqu'il y a un achat ou une vente, il enverra immédiatement un message Telegram à l'utilisateur pour signaler qu'il y a eu une prise de position. Comme ça l'utilisateur pourra tout de même surveiller d'un œil externe les prises de position par du Bot de trading.

Voilà, l'aperçu global du fonctionnement de ces 2 Bots, ci-dessous vous trouverez un diagramme de séquence qui récapitule tout ce que je viens de dire.

Figure 31 : Diagramme de séquence - Bot



(Anushan Alvaipillai, 2022)

## 4.8 Serveur AWS

Pour lancer notre Bot, il nous faut un serveur. Pour ça, nous allons utiliser un serveur Ubuntu qui sera hébergé sur AWS. L'avantage de AWS est que ce petit serveur sera presque gratuit pendant un an. Pour mettre en place le serveur, voici les étapes qu'il faut mettre en place.

S'inscrire ou se connecter sur le lien suivant : <https://aws.amazon.com/fr/>

Taper sur la barre de recherche EC2 et cliquer sur le premier lien

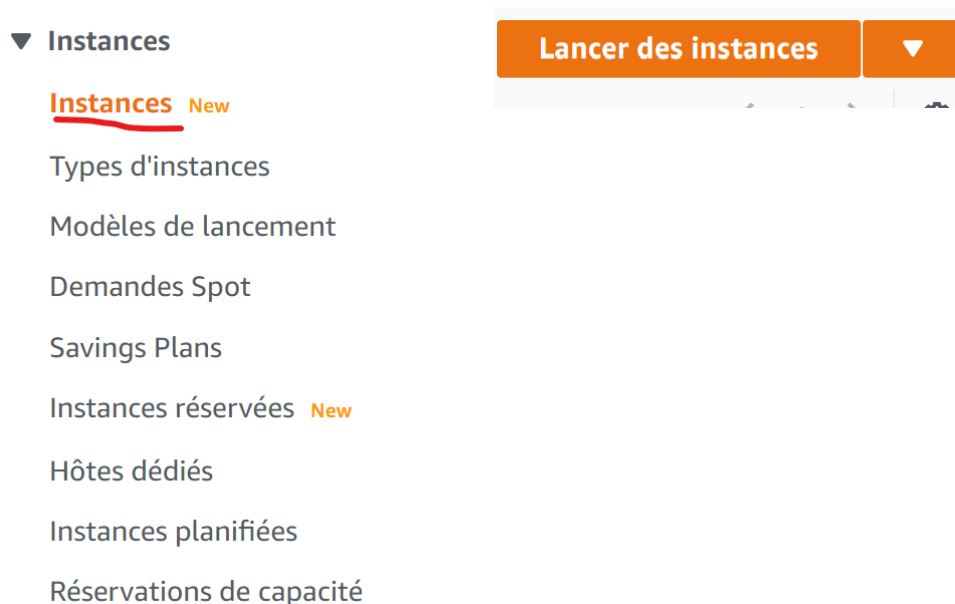
Figure 32 : AWS - EC2



(<https://aws.amazon.com/fr/>)

Cliquer sur « instances » sur le menu de navigation à gauche et cliquer sur le bouton « lancer des instances »

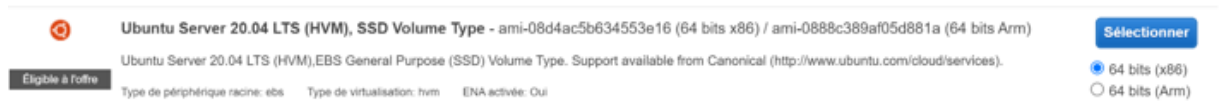
Figure 33 : AWS - Instances



(<https://aws.amazon.com/fr/>)

Sélectionner le serveur Ubuntu 20.04 LTS en cliquant sur le bouton « sélectionner »

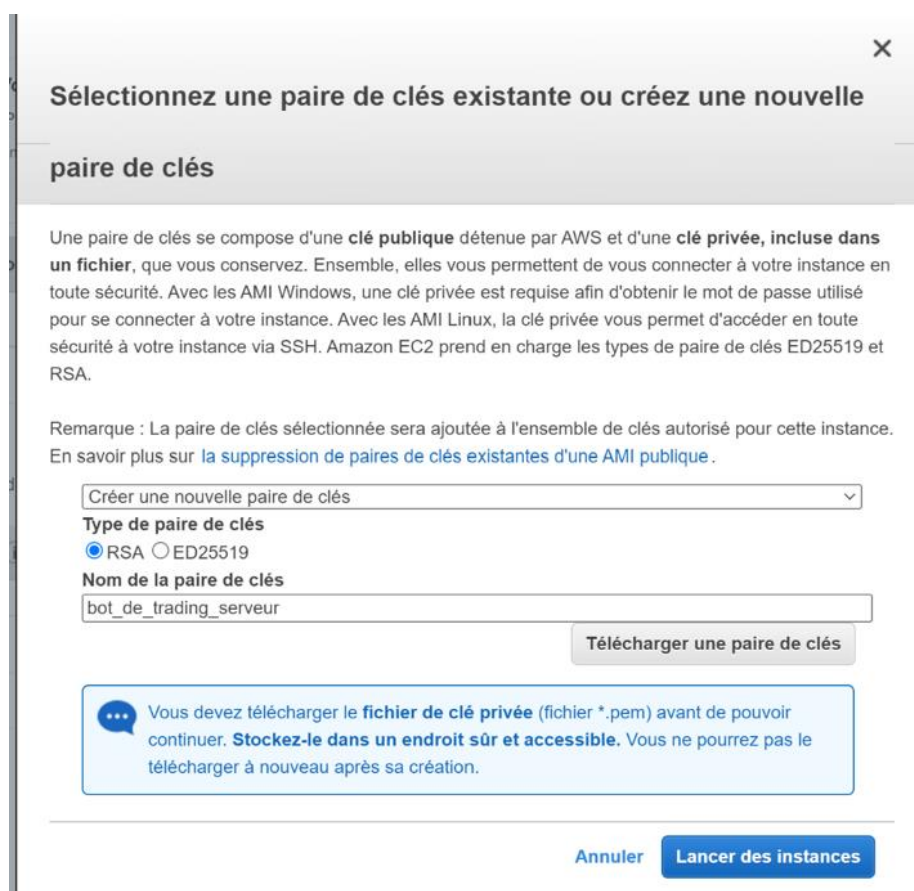
Figure 34 : AWS - serveur Ubuntu 20.04



(<https://aws.amazon.com/fr/>)

Cliquer sur le bouton « vérifier et lancer » et cliquer sur le bouton « lancer »

Figure 35 : AWS - Paire de clés



(<https://aws.amazon.com/fr/>)

Créer une nouvelle paire de clés en ajoutant le nom de la paire et la télécharger.

Une fois que le téléchargement est fait, cliquer sur « Lancer des instances » qui va lancer l'instance. Puis cliquer sur le bouton « afficher l'instance » pour voir si tout s'est passé. Pour ça il faut que le serveur en cours d'exécution.

Figure 36 : AWS - Serveur en cours d'exécution



Après avoir exécuté le serveur, il faut que nous puissions nous connecter à notre serveur web en machine local. Pour ça, nous aurons besoin d'effectuer les étapes suivantes :

Installer Putty en passant par le lien suivant :

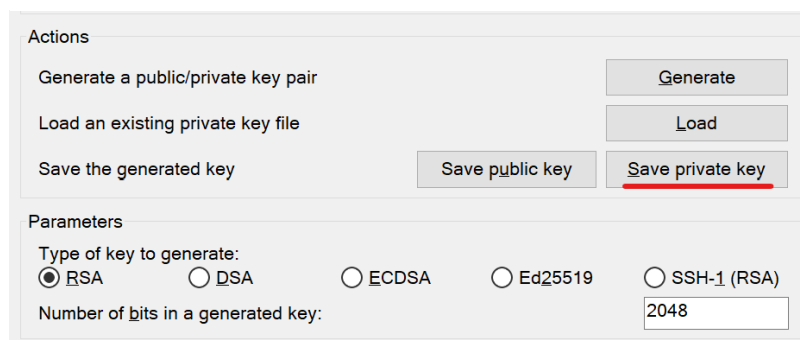
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Ouvrir « puttyGen » depuis votre machine locale.

Cliquer sur « load » et charger la paire de clés que vous avez téléchargé depuis AWS.

Une fois que le fichier a été chargé, on peut sauvegarder notre clé qui va nous permettre de nous authentifier au server depuis notre machine locale.

Figure 37 : Putty - Sauvegarde clé privée



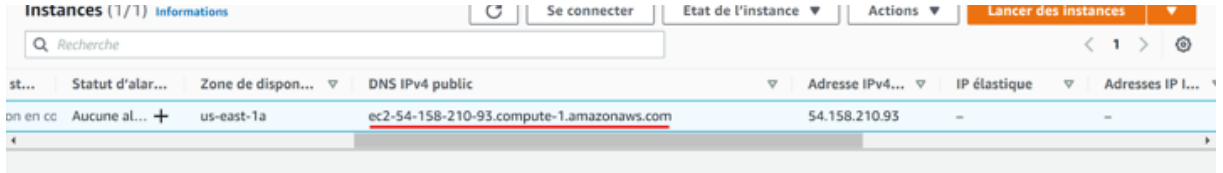
(Putty)

Après avoir généré les clés, on peut enfin nous connecter à notre serveur AWS. Pour ça, il faut procéder aux étapes suivantes :

Ouvrir Putty depuis votre machine locale

Aller sur votre navigateur pour regarder votre instance et regarder la colonne « DNS IPv4 public » qui sera le dns de l'hébergeur.

Figure 38 : AWS - DNS IPv4 public

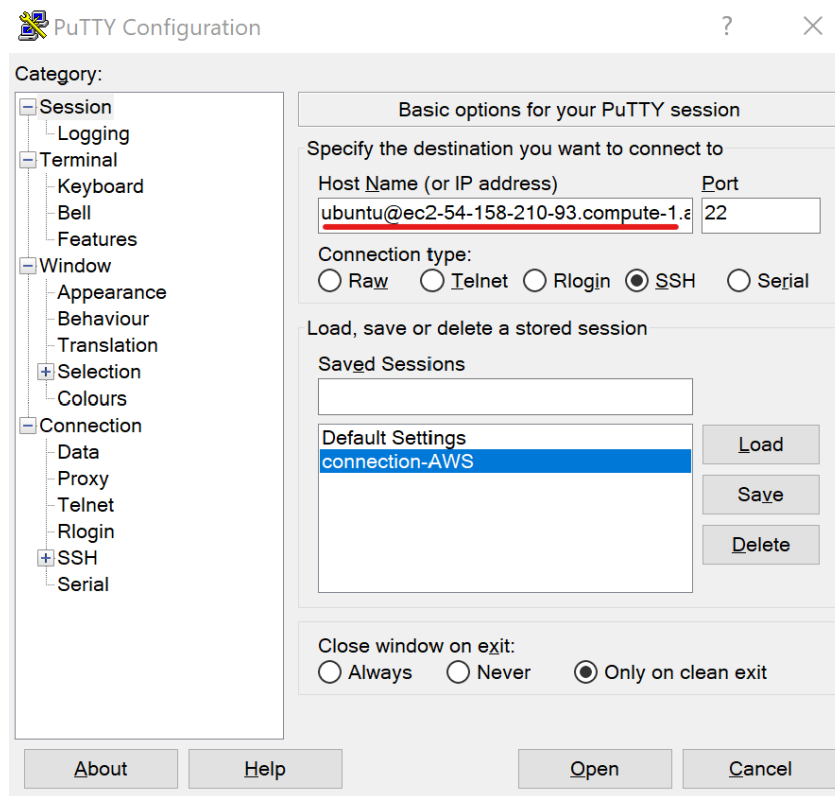


st...	Statut d'alar...	Zone de dispon...	DNS IPv4 public	Adresse IPv4...	IP élastique	Adresses IP L...
on en cc	Aucune al...	us-east-1a	<u>ec2-54-158-210-93.compute-1.amazonaws.com</u>	54.158.210.93	-	-

(<https://aws.amazon.com/fr/>)

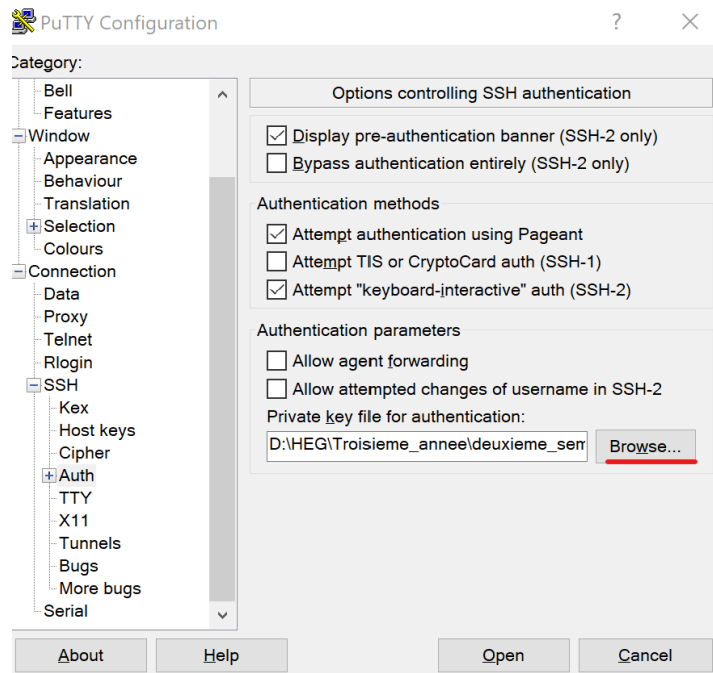
Ajouter sur Putty dans host : « ubuntu@ » suivi le dns de l'hébergeur.

Figure 39 : Putty - Paramétrage du Host Name



(Putty)

Figure 40 : Importation clé privée



(Putty)

Cliquer SSH sur le menu à gauche et cliquer sur Auth. Puis, cliquer sur le bouton « Browse.. » pour aller chercher la clé qu'on vient de générer.

Cliquer sur le bouton « open » qui va permettre de faire une connexion SSH depuis le serveur. Voilà, on a maintenant réussi à faire une connexion SSH entre le serveur et la machine locale.

Figure 41 : Connexion SSH réussie

```
System load: 0.0          Processes: 102
Usage of /: 19.2% of 7.58GB  Users logged in: 0
Memory usage: 21%         IPv4 address for eth0: 172.31.84.194
Swap usage: 0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-84-194:~$
```

(Putty)

Une fois que la connexion SSH est réussie, nous pouvons configurer correctement le serveur. Pour ça, il faut exécuter les requêtes suivantes.

Sudo apt-get update : cette commande va nous permettre d'installer toutes les mises à jour nécessaires pour le bon fonctionnement du serveur.

Sudo apt install python3-pip : cette commande va nous permettre d'installer pip et qui va nous permettre d'installer toutes les librairies sur Python

Après avoir copié-collé les fichiers pour notre programme dans un dossier. On peut enfin planifier un planificateur qui va exécuter le Bot de trading chaque heure et le Bot fondamentales pour sélectionner les cryptomonnaie chaque mois. Les prises de position seront inscrites dans fichier log.

Pour ça, il faut créer un fichier log à l'aide de la commande nano. Exemple : nano historique.log

Ensuite il faut exécuter la commande « crontab -e » pour créer un planificateur de tâche.

Figure 42 : Crontab - planificateur de tâches

```
## Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
0 * * * * python3 main.py >> cronlog.log
0 0 1 * * python3 fundamental_analysis_crypto.py
```

(Putty)

Puis, il faut lui dire dans le fichier, ce qu'il doit faire et quand il doit le faire. Ci-dessus, nous pouvons voir, sur la première ligne blanche, que le « 0 \* \* \* \* » correspond à chaque heure. Chaque heure, le serveur va exécuter le programme python « main » et stocker les prints dans le fichier log.

Sur la deuxième ligne blanche, on peut constater que le « 0 0 1 \* \* » correspond au premier de chaque mois. Chaque premier du mois, le serveur va exécuter le programme python « fundamental\_analysis\_crypto ».

Maintenant que notre robot est terminé, nous pouvons enfin passer à la phase de simulation.

## 5. Backtesting

Le backtesting vise à évaluer la performance d'une stratégie dans le temps. En effet, elle permet de savoir si cette stratégie est rentable ou pas. Une grande partie des traders le font à la main. Ils vont d'abord définir la période où ils vont effectuer la simulation. Puis ils vont mettre les résultats pour chaque trade dans un tableau. Pour ensuite faire une analyse de leur résultat final pour savoir s'ils sont en positif ou négatif.

Nous aussi, on va faire un backtesting pour notre stratégie pour savoir si notre stratégie est rentable ou pas. Pour ça, nous allons créer un script qui va le faire de manière automatisée. Ce script reprend la même structure que notre Bot, mais avec une spécificité supplémentaire. C'est qu'il va prendre les données du passé. Cependant, nous allons le faire sur l'analyse technique avec les 10 cryptomonnaies actuelles qui sont les plus sous-évaluées actuellement selon notre analyse fondamentale. Dans notre backtesting, nous ne pouvons pas utiliser l'analyse sentimentale qu'on souhaite entreprendre pour le Bot de trading, puisque l'api de base de Twitter et de google news, ne nous permet pas de remonter aux anciens tweets ou anciens articles.

Par conséquent, on devra utiliser une autre stratégie sentimentale qui est le fear and greed de Bitcoin. Cette stratégie a pour but de valider l'achat lorsque le fear and greed est inférieur à 20 ou supérieur à 40. Si elle est inférieure à 20, cela montre que beaucoup de gens panique et que c'est une bonne opportunité pour investir. Si elle est supérieure à 40, cela montre que la confiance est installée et que les gens plutôt positifs à l'achat pour le monde de la cryptomonnaie. Le fear and greed sur Bitcoin est basé sur les points suivants :

- Volatilité (25%)
- Momentum du marché (25%)
- Médias sociaux (15%)
- Enquêtes (15 %)
- Dominance (10%)
- Tendances (10%)

Comme ça on aura tout de même une analyse sentimentale sur le marché de la cryptomonnaie dans notre backtesting, puisque Bitcoin est celui qui mène la danse dans le marché de la cryptomonnaie concernant le prix. Cependant, notre backtesting ne sera pas 100% juste, mais nous donnera tout de même une idée des résultats donnés par l'analyse technique.

Les cryptos qui ont été sélectionnées pour le 20 juin 2022 par notre Bot d'analyse fondamentale sont les suivantes :

ETH, ADA, XRP, DOGE, LTC, XLM, BCH, ALGO, ETC

## 5.1 Résultat 2022

Après avoir récupéré ces différents cryptomonnaies, le programme de notre backtest nous a donné le résultat suivant :

On peut le voir sur le résultat ci-dessous, sur une période qui va du 1<sup>er</sup> janvier 2022 jusqu'au 20 juin 2022, avec un capital de départ de 1000 \$, le Bot a réalisé un bénéfice de 29\$.

Figure 43 : Résultat 2022 avec le Fear and Gread

```
Trading Bot sur : 9 cryptomonnaies | Plage de temps : 1h
Période : [2022-01-01 00:00:00] -> [2022-06-20 14:00:00]
Capital de départ : 999.7666666666667 $

----- Informations générales -----
Capital final : 1028.81 $
Performance vs US Dollar : 2.9 %
Buy and Hold Performance : -69.73 %
Meilleur trade : +2.09 %, the 2022-05-31 22:00:00
Pire trade : -2.07 %, the 2022-06-02 04:00:00
Commissions totales : 29.0 $

----- Informations trades -----
Nombre total de trades : 61
Nombre total de trades positifs : 40
Nombre total de trades négatifs : 21
Trades win rate ratio : 65.57 %
Performance moyenne des trades : 0.07 %
Performance moyenne des trades positifs : 0.46 %
Performance moyenne des trades négatifs : -0.66 %
```

(Anushan Alvaipillai, 2022)

Aussi on peut constater que si le Bot avait acheté ces cryptomonnaies et les avait conservé jusqu'au 20 juin 2022, il aurait fait une perte de -69.73% par rapport à son capital de départ. On peut voir aussi que, le meilleur trade et le pire trade, ont un peu près les mêmes ratios. Les 61 trades effectués coûteraient 29\$ en matière de commissions. Ses commissions sont déjà déduites lorsqu'on voit le capital final. On peut voir aussi que le programme a beaucoup plus de trades positifs que de trades négatifs. La performance moyenne des trades est positive.

## 5.2 Résultat 2022 sans l'analyse sentimentale (fear and gread)

Figure 44 : Résultat 2022 sans le Fear and Gread

```
Trading Bot sur : 9 cryptomonnaies | Plage de temps : 1h
Période : [2022-01-01 00:00:00] -> [2022-06-20 14:00:00]
Capital de départ : 999.766666666667 $

----- Informations générales -----
Capital final : 929.07 $
Performance vs US Dollar : -7.07 %
Buy and Hold Performance : -69.73 %
Meilleur trade : +3.43 %, the 2022-03-22 22:00:00
Pire trade : -2.88 %, the 2022-05-05 20:00:00
Commissions totales : 57.67 $

----- Informations trades -----
Nombre total de trades : 126
Nombre total de trades positifs : 78
Nombre total de trades négatifs : 48
Trades win rate ratio : 61.9 %
Performance moyenne des trades : -0.03 %
Performance moyenne des trades positifs: 0.47 %
Performance moyenne des trades négatifs : -0.84 %
```

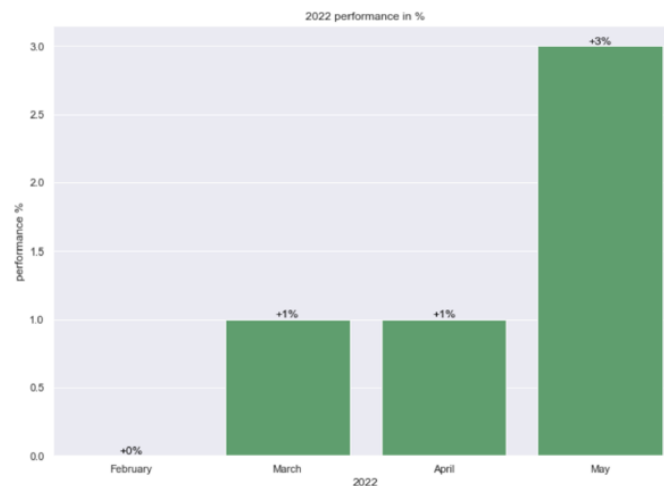
(Anushan Alvaipillai, 2022)

On peut voir ci-dessus, que sans le fear and gread, on peut voir qu'on est en perte de 80.93\$. En effet, le Bot effectue plus de trades, puisqu'il a moins de conditions. Ce qui implique une augmentation des frais de commissions. On peut voir aussi que la performance moyenne des trades négatifs est presque 2 fois plus grande que les trades positifs. Néanmoins, cette stratégie est tout de même meilleure que d'acheter et de garder pendant cette période.

### 5.3 Performance durant chaque mois

Ci-dessous, on peut voir, en comparant avec la capitalisation boursière du marché total de la cryptomonnaie, que le Bot avec le fear and greed a tout de même réussi à être positif tout en long du mois, malgré que le marché de la cryptomonnaie soit en négatif à partir du mois de mars.

Figure 46 : Performance sur chaque mois



(Anushan Alvaipillai, 2022)

Figure 45 : Graphique capitalisation boursière totale cryptomonnaie



(<https://fr.tradingview.com/chart/?symbol=CRYPTOCAP%3ATOTAL>)

## 6. Conclusion

Ce travail m'a permis d'apprendre énormément sur le trading. Ainsi que le fonctionnement du marché boursier. J'ai compris l'importance d'avoir une stratégie et de pouvoir gérer ses émotions. Le Bot qu'on a vu dans ce travail permet de remplir ces 2 objectifs. Il m'a fait prendre conscience l'importance du « Backtesting » qui permet de vérifier si une stratégie est rentable.

Dans ce travail, j'ai pris connaissance qu'il existait énormément d'outil qui facilitait le développement de notre idée. Il m'a permis de prendre conscience qu'on pouvait faire plein de choses avec un simple programme.

Cependant le programme n'est pas encore à la hauteur d'un trader expérimenté qui sait gérer ses émotions. Un trader qui connaît parfaitement son terrain sait s'adapter en utilisant une stratégie bien conçue pour chaque situation. Alors que le robot qu'on a vu continuera d'utiliser la même stratégie dans n'importe quelle situation.

C'est pourquoi, les points à améliorer sur le robot sont les suivants. Premièrement, c'est d'avoir une stratégie différente selon la tendance. Deuxièmement, c'est d'instaurer les longs et les shorts. Troisièmement, c'est de trouver une solution pour obtenir plus d'articles et de tweet pour avoir une analyse sentimentale plus fiable.

Ainsi, on a pu voir dans ce travail, un Bot de trading qui permettait d'établir les 3 types d'analyses sur une cryptomonnaie pour ensuite prendre une position. On a vu aussi comment le Bot pouvait communiquer avec nous et comment on pouvait l'héberger.

## Bibliographie

- La Rédaction JDN, 2019, JDN, Trading : définition, risques, apprendre les bases...[en ligne], [sans date], Dernière modification de cette page le 08 mars 2019. [Consulté le 01 avril 2022], Disponible à l'adresse : <https://www.journaldunet.fr/business/dictionnaire-comptable-et-fiscal/1198619-trading-definition/>
- The Organic Chemistry Tutor, 2020, YouTube, Fundamental Analysis [en ligne], [5 juin 2020], [Consulté le 12 avril 2022], Disponible à l'adresse : <https://www.youtube.com/watch?v=9wsr8ITCGro&t=555s>
- Coin Metrics, 2018, CoinMetrics, INTRODUCING REALIZED CAPITALIZATION [en ligne] [14 décembre 2018], [Consulté le 18 avril 2022], Disponible à l'adresse : <https://coinmetrics.io/realized-capitalization/>
- Kabba Thami, 2015, l'art du trading, 3<sup>e</sup> éd., Paris, Eyrolles, 2015, pp 33 -38, pp 117-128, 147-158, [consulté le 28 avril 2022]
- Dagade Pushpak, 2020, Python Algorithmic Trading Cookbook, 1e ed. Birmingham, Packt Publishing, pp 29-35, pp 80-84, pp 173-176, [consulté le 29 avril 2022]
- Part Time Larry, 2020, Candlestick Pattern Recognition with Python and TA-Lib [en ligne], [02 août 2020], [Consulté le 29 avril 2022], Disponible à l'adresse : <https://www.youtube.com/watch?v=QGkf2-caXmc&list=LL&index=42>
- Auteurs spécialisés Ooreka, 2022, oorekaArgent, robot de trading [en ligne], [sans date] Dernière modification de cette page en juin 2022, [Consulté le 01 mai 2022], Disponible à l'adresse : [https://epargne.ooreka.fr/astuce/voir/600037/robot-de-trading#:~:text=Robot%20de%20trading&text=Un%20robot%20de%20trading%20\(ausi,le%20march%C3%A9%20des%20options%20binaires](https://epargne.ooreka.fr/astuce/voir/600037/robot-de-trading#:~:text=Robot%20de%20trading&text=Un%20robot%20de%20trading%20(ausi,le%20march%C3%A9%20des%20options%20binaires)
- Compuenter Science, 2020, YouTube, Twitter Sentiment Analysis Using Python [en ligne], [04 février 2020], [Consulté le 02 mai 2022], Disponible à l'adresse : <https://www.youtube.com/watch?v=ujld4ipkBio&list=LL&index=2&t=2144s>
- Jason Fernando, 2022, investopedia, Relative Strength Index (RSI) [en ligne], [sans date], Dernière modification de cette page le 19 février 2022, [Consulté le 05 mai 2022], Disponible à l'adresse : <https://www.investopedia.com/terms/r/rsi.asp>
- Jason Fernando, 2022, investopedia, Moving Average Convergence Divergence (MACD) [en ligne], [sans date], Dernière modification de cette page le 01 mars 2022, [Consulté le 05 mai 2022], Disponible à l'adresse : <https://www.investopedia.com/terms/m/macd.asp>
- ZB Campus, [sans date], zonebours, savoir investir – l'analyse technique [en ligne], [sans date], [Consulté le 05 mai 2022], Disponible à l'adresse : [https://www.zonebourse.com/formation/espace\\_pedagogique/L-analyse-technique-111/](https://www.zonebourse.com/formation/espace_pedagogique/L-analyse-technique-111/)
- Cory Mitchell, 2021, investopedia, Williams %R Definition and Uses [en ligne], [sans date], Dernière modification de cette page le 23 août 2021 [Consulté le 05 mai 2022], Disponible à l'adresse : <https://www.investopedia.com/terms/w/williamsr.asp>
- Admirals, 2021, Admiral Markets, Profiter de l'élan du marché grâce à l'Indicateur Awesome Oscillator [en ligne], [22 juillet 2021], [Consulté le 05 mai 2022], Disponible à l'adresse : <https://admiralmarkets.com/fr/formation/articles/indicateurs-forex/indicateur-awesome>

Adam Hayes, 2021, investopedia, Stochastic RSI -StochRSI Definition [en ligne], [sans date], Dernière modification de cette page le 02 août 2021, [Consulté le 05 mai 2022], Disponible à l'adresse : <https://www.investopedia.com/terms/s/stochrsi.asp>

L'investisseur inspiré, [sans date], Placements en Directs, Qu'est-ce que l'analyse fondamentale ? [en ligne], [sans date], [Consulté avril 12 mai 2022], Disponible à l'adresse : <https://www6.royalbank.com/fr/di/hubs/investing-academy/chapter/fundamental-analysis/jx5cjc6/jxy2wuu2#:~:text=L'analyse%20fondamentale%20%C3%A9tudie%20es.du%20choix%20de%20vos%20titres.>

Crypto Robot, 2021, YouTube, La v2 de la stratégie du BIG WILL ?! Installation live sur FTX de votre BOT DE TRADING ! [en ligne], [21 décembre 2021], [Consulté le 15 mai 2022], Disponible à l'adresse : <https://www.youtube.com/watch?v=cFIENgZukkY&list=LL&index=7&t=814s>

CryptoRobotFr, 2021, Github, big\_will\_v2\_live.py [en ligne], [12 décembre 2021], Dernière modification de cette page le 26 décembre 2021, [Consulté le 15 mai 2022], Disponible à l'adresse : [https://github.com/CryptoRobotFr/cBot-Project/blob/main/live\\_strategy/big\\_will\\_v2\\_live.py](https://github.com/CryptoRobotFr/cBot-Project/blob/main/live_strategy/big_will_v2_live.py)

Academy Binance, 2020, A Guide to Cryptocurrency Fundamental Analysis [en ligne], [21 septembre 2020], Dernière modification de cette page le 23 avril 2022, [Consulté le 28 mai 2022], Disponible à l'adresse : <https://academy.binance.com/en/articles/a-guide-to-cryptocurrency-fundamental-analysis>

Bussière Anthony, 2019, le guide complet du trading, 2<sup>e</sup> ed., Malakoff, Maxima, pp 85-92, [consulté le 29 mai 2022]

Made-Up masters, 2020, Medium, How to Write a Telegram Bot to Send Messages with Python [en ligne], [28 juillet 2020], [Consulté le 15 juin 2022], Disponible à l'adresse : <https://medium.com/@robertbracco1/how-to-write-a-telegram-Bot-to-send-messages-with-python-bcdf45d0a580>

Cronitor, [sans date], crontab, [sans date], [Consulté le 22 juin 2022], Disponible à l'adresse : <https://crontab.guru/>

Dale Gilham, 2021, wealthwithin, Trading the Stock Market – Why Most Traders Fail [en ligne], [sans date], Dernière modification de cette page le 10 février 2021, [Consulté le 23 juin 2022], Disponible à l'adresse : <https://www.wealthwithin.com.au/learning-centre/share-trading-tips/trading-the-stock-market#:~:text=Anyone%20who%20starts%20down%20the,per%20cent%20make%20money%20consistently.>

## Annexe 1 : Informations proposées par Messari – Bitcoin

```
{"status": {"elapsed": 1, "timestamp": "2022-06-26T09:58:53.814226587Z"}, "data": {"id": "1e31218a-e44e-4285-820c-8282ee222035", "serial_id": 6057, "symbol": "BTC", "name": "Bitcoin", "slug": "bitcoin", "contract_addresses": null, "_internal_temp_agora_id": "9793eae6-f374-46b4-8764-c2d224429791", "market_data": {"price_usd": 21396.44479815818, "price_btc": 1, "price_eth": 17.27403516763708, "volume_last_24_hours": 3039053795.9445786, "real_volume_last_24_hours": 2085665855.0926657, "volume_last_24_hours_overstatement_multiple": 1.457114421528253, "percent_change_usd_last_1_hour": -0.12857155090504005, "percent_change_btc_last_1_hour": 0, "percent_change_eth_last_1_hour": 0.2032700592180085, "percent_change_usd_last_24_hours": -0.08118114551387484, "percent_change_btc_last_24_hours": 0.07404917714161582, "percent_change_eth_last_24_hours": -0.6046217668546761, "ohlcv_last_1_hour": {"open": 21426.483909294424, "high": 21427.103700279775, "low": 21421.12385413961, "close": 21423.989954308174, "volume": 680914.3166271511}, "ohlcv_last_24_hour": {"open": 21428.606117825777, "high": 21478.961238301585, "low": 21382.07713163743, "close": 21413.828789668012, "volume": 40032201.12635783}, "last_trade_at": "2022-06-26T09:58:53Z"}, "marketcap": {"rank": 1, "marketcap_dominance_percent": 42.32086112618945, "current_marketcap_usd": 408398637479.5364, "y_2050_marketcap_usd": 449293253283.76953, "y_plus10_marketcap_usd": 442939785666.16956, "liquid_marketcap_usd": 408629889852.547, "volume_turnover_last_24_hours_percent": 0.5103323455943136, "realized_marketcap_usd": 429505184011.16205, "outstanding_marketcap_usd": 409551146129.82904}, "supply": {"y_2050": 20986335.65, "y_plus10": 20689567.33, "liquid": 19086963.72, "circulating": 19076162, "y_2050_issued_percent": 90.94948274116639, "annual_inflation_percent": 1.761171682051078, "stock_to_flow": 56.78038150348807, "y_plus10_issued_percent": 92.25404966455622, "supply_revived_90d": 19076979.6784845}, "blockchain_stats_24_hours": {"count_of_active_addresses": 742895, "transaction_volume": 22193695708.377415, "adjusted_transaction_volume": 7766199891.287385, "adjusted_nvt": 52.735076596379, "median_tx_value": 71.70426614315589, "median_tx_fee": 0.4901222742659428, "count_of_tx": 217926, "count_of_payments": 668284, "new_issuance": 20260748.854072865, "average_difficulty": 29570168636357.99, "kilobyte
```

```
es_added": null,"count_of_blocks_added": 151},"market_data_liquidity":
  {"clearing_prices_to_sell": null,"marketcap": null,"asset_bid_depth":
    null,"usd_bid_depth": null,"updated_at": "0001-01-
01T00:00:00Z"},"all_time_high": {"price": 68721.93482129039,"at": "202
1-11-
10T14:00:00Z"},"days_since": 34,"percent_down": 31.129753778232587,"bre
akeven_multiple": 1.4520058441201504},"cycle_low": {"price": 17664.958
902932794,"at": "2022-06-
18T20:15:00Z"},"percent_up": 21.22126256533331,"days_since": 7},"token_
sale_stats": {"sale_proceeds_usd": null,"sale_start_date": null,"sale_
end_date": null,"roi_since_sale_usd_percent": null,"roi_since_sale_btc_
percent": null,"roi_since_sale_eth_percent": null},"mining_stats": {"
mining_algo": "SHA-256","network_hash_rate": "206845
PH/s","available_on_nicehash_percent": 0.08719448904480616,"1_hour_att
ack_cost": null,"24_hours_attack_cost": null,"attack_appeal": null,"ha
sh_rate": 221961099.20676234,"hash_rate_30d_average": 213615325.548054
1,"mining_revenue_per_hash_usd": 0.000001071452,"mining_revenue_per_ha
sh_native_units": 5e-
11,"mining_revenue_per_hash_per_second_usd": 0.092573487009,"mining_re
venue_per_hash_per_second_native_units": 0.000004312093,"mining_revenu
e_from_fees_percent_last_24_hours": 1.3965743083,"mining_revenue_nativ
e": 957.116848,"mining_revenue_usd": null,"mining_revenue_total": 4397
6294920.85188,"average_difficulty": 29570168636357.99},"developer_acti
vity": {"stars": 64958,"watchers": 3931,"commits_last_3_months": null,
"commits_last_1_year": null,"lines_added_last_3_months": null,"lines_a
dded_last_1_year": null,"lines_deleted_last_3_months": null,"lines_del
eted_last_1_year": null},"roi_data": {"percent_change_last_1_week": 4.
263784619880319,"percent_change_last_1_month": -
25.08608444962012,"percent_change_last_3_months": -
54.5527557427939,"percent_change_last_1_year": -
33.65608049018777,"percent_change_btc_last_1_week": 0,"percent_change_
btc_last_1_month": 0,"percent_change_btc_last_3_months": 0,"percent_ch
ange_btc_last_1_year": 0,"percent_change_eth_last_1_week": -
5.536832754862781,"percent_change_eth_last_1_month": 3.905162505853881
2,"percent_change_eth_last_3_months": 21.822946014927275,"percent_chan
ge_eth_last_1_year": -
2.3669713118472413,"percent_change_month_to_date": -
28.082933618787635,"percent_change_quarter_to_date": -
53.728276518362094,"percent_change_year_to_date": -
55.12356134455426},"roi_by_year": {"2021_usd_percent": 57.162219730209
```

57,"2020\_usd\_percent": 303.2837217865457,"2019\_usd\_percent": 86.68855147566646,"2018\_usd\_percent": -72.45868543938644,"2017\_usd\_percent": 1294.846750497474,"2016\_usd\_percent": 123.12968437572151,"2015\_usd\_percent": 36.81170933790877,"2014\_usd\_percent": -57.34911749493351,"2013\_usd\_percent": 5394.202568976737,"2012\_usd\_percent": 156.47080487351116,"2011\_usd\_percent": 1474.0066666666667},"risk\_metrics": {"sharpe\_ratios": {"last\_30\_days": -3.7887290763586683,"last\_90\_days": -3.590613347550442,"last\_1\_year": -0.4367929706269955,"last\_3\_years": 0.6030733213754257},"volatility\_stats": {"volatility\_last\_30\_days": 0.9020240802056272,"volatility\_last\_90\_days": 0.7414148589434231,"volatility\_last\_1\_year": 0.6949099923426819,"volatility\_last\_3\_years": 0.7698959127003936}},"misc\_data": {"private\_market\_price\_usd": null,"vladimir\_club\_cost": null,"btc\_current\_normalized\_supply\_price\_usd": null,"btc\_y2050\_normalized\_supply\_price\_usd": null,"asset\_created\_at": null,"asset\_age\_days": null,"categories": ["Payments"],"sectors": ["Currencies"],"tags": null},"reddit": {"active\_user\_count": null,"subscribers": null},"on\_chain\_data": {"addresses\_count": 42385115,"active\_addresses": 742895,"active\_addresses\_received\_count": 555241,"active\_addresses\_sent\_count": 455764,"addresses\_balance\_greater\_0\_001\_native\_units\_count": 20836838,"addresses\_balance\_greater\_0\_01\_native\_units\_count": 10301798,"addresses\_balance\_greater\_0\_1\_native\_units\_count": 3629575,"addresses\_balance\_greater\_1\_usd\_count": 34757805,"addresses\_balance\_greater\_1\_native\_units\_count": 870386,"addresses\_balance\_greater\_10\_usd\_count": 24394207,"addresses\_balance\_greater\_10\_native\_units\_count": 148166,"addresses\_balance\_greater\_100\_usd\_count": 13693466,"addresses\_balance\_greater\_100\_native\_units\_count": 15873,"addresses\_balance\_greater\_100k\_usd\_count": 253129,"addresses\_balance\_greater\_100k\_native\_units\_count": 5,"addresses\_balance\_greater\_10k\_usd\_count": 1479082,"addresses\_balance\_greater\_10k\_native\_units\_count": 98,"addresses\_balance\_greater\_10m\_usd\_count": 4920,"addresses\_balance\_greater\_1k\_usd\_count": 5250027,"addresses\_balance\_greater\_1k\_native\_units\_count": 2207,"addresses\_balance\_greater\_1m\_usd\_count": 65017,"addresses\_balance\_greater\_1m\_native\_units\_count": 0,"average\_block\_gas\_limit": null,"block\_height": 742350,"average\_block\_interval": 574.49333333,"blocks\_added\_last\_24\_hours": 151,"block\_size\_bytes\_total": 163961722,"block\_size\_bytes\_average": 1085839.2185430464,"uncle\_blocks\_count\_last\_24\_hours": null,"uncle\_rewards\_last\_24\_hours\_usd": null,"uncle\_rewards\_last\_24\_hours\_native\_units": null,"mining\_revenue\_from\_uncle\_blocks\_percent\_last\_24\_hours": null,"block\_weight": 477981028,"averag

e\_block\_weight": 3165437.27152318,"total\_fees\_last\_24\_hours": 13.366848,"total\_fees\_last\_24\_hours\_usd": 286964.0797865602,"hash\_rate": 221961099.20676234,"issuance\_last\_24\_hours\_native\_units": 943.75,"issuance\_rate\_daily": 0.0049470619,"issuance\_total\_usd": 20260748.854072865,"issuance\_total\_native\_units": 943.75,"new\_issuance": 20260748.854072865,"issuance\_rate": 1.8056775935,"realized\_marketcap\_usd": 429505184011.16205,"adjusted\_nvt": 52.735076596379,"adjusted\_nvt\_90d\_moving\_average": 22.565087559848642,"adjusted\_rvt": 55.304420440299,"adjusted\_rvt\_90d\_moving\_average": 23.664497526637213,"txn\_count\_last\_24\_hours": 217926,"txn\_contracts\_calls\_count\_last\_24\_hours": null,"txn\_contracts\_calls\_success\_count\_last\_24\_hours": null,"txn\_contracts\_creation\_count\_last\_24\_hours": null,"txn\_contracts\_destruction\_count\_last\_24\_hours": null,"txn\_contracts\_count\_last\_24\_hours": null,"txn\_erc20\_count\_last\_24\_hours": null,"txn\_erc721\_count\_last\_24\_hours": null,"txn\_per\_second\_count": 2.522291666667,"txn\_token\_count\_last\_24\_hours": null,"average\_fee\_usd": 1.316868169227899,"average\_fee\_native\_units": 0.00006134,"median\_fee\_usd": 0.4901222742659428,"median\_fee\_native\_units": 0.00002283,"average\_txn\_gas\_used": null,"txn\_gas\_limit\_last\_24\_hours": null,"average\_txn\_gas\_limit": null,"txn\_gas\_used\_last\_24\_hours": null,"transfer\_count\_last\_24\_hours": 668284,"transfer\_erc\_20\_count\_last\_24\_hours": null,"transfer\_erc721\_count\_last\_24\_hours": null,"txn\_volume\_last\_24\_hours\_usd": 22193695708.377415,"adjusted\_txn\_volume\_last\_24\_hours\_usd": 7766199891.287385,"adjusted\_txn\_volume\_last\_24\_hours\_native\_units": 361751.2462245,"txn\_volume\_last\_24\_hours\_native\_units": 1033787.07646192,"average\_transfer\_value\_usd": 33209.976160403414,"average\_transfer\_value\_native\_units": 1.546927767928,"median\_transfer\_value\_usd": 71.70426614315589,"median\_transfer\_value\_native\_units": 0.00334,"average\_utxo\_age": 1162.70261314,"value\_weighted\_average\_utxo\_age": 1284.28728553,"median\_utxo\_age": 957,"utxo\_count\_last\_24\_hours": 82749719,"utxo\_in\_loss\_count": 30767681,"utxo\_in\_profit\_count": 51982038},"exchange\_flows": {"flow\_in\_exchange\_native\_units": 9356.22521924,"flow\_in\_exchange\_usd": 200862653.65739283,"flow\_in\_exchange\_native\_units\_inclusive": 11304.06649991,"flow\_in\_exchange\_usd\_inclusive": 242679578.68547285,"flow\_out\_exchange\_native\_units": 11465.91416404,"flow\_out\_exchange\_usd": 246154179.87811527,"flow\_out\_exchange\_native\_units\_inclusive": 13413.75544471,"flow\_out\_exchange\_usd\_inclusive": 287971104.9061953,"flow\_in\_binance\_usd": 94369031.6296406,"flow\_in\_binance\_native\_units": 4395.72960713,"flow\_in\_bitfinex\_usd": 16973064.271361377,"flow\_in\_bitfinex\_native\_units": 790.6089514,"flow\_in\_bitmex\_usd": 1314116.0308979491,"flow\_in\_bitmex\_native\_units": 61.21180481,"flow\_in\_bitstamp\_usd": 4637139.3696

**82759**, "flow\_in\_bitstamp\_native\_units": **215.9989402**, "flow\_in\_bittrex\_usd": **5548895.551911825**, "flow\_in\_bittrex\_native\_units": **258.4687375**, "flow\_in\_gemini\_usd": **2913805.323558393**, "flow\_in\_gemini\_native\_units": **135.72567302**, "flow\_in\_huobi\_usd": **23666473.535773747**, "flow\_in\_huobi\_native\_units": **1102.38938157**, "flow\_in\_kraken\_usd": **34445541.50891187**, "flow\_in\_kraken\_native\_units": **1604.48066521**, "flow\_in\_poloniex\_usd": **1507688.2986522357**, "flow\_in\_poloniex\_native\_units": **70.2284423**, "flow\_net\_binance\_usd": **4888703.52858356**, "flow\_net\_binance\_native\_units": **227.71685234**, "flow\_net\_bitfinex\_usd": **-18087133.137554847**, "flow\_net\_bitfinex\_native\_units": **-842.50251664**, "flow\_net\_bitmex\_usd": **-8736330.458093094**, "flow\_net\_bitmex\_native\_units": **-406.9401348**, "flow\_net\_bitstamp\_usd": **-860524.8737156346**, "flow\_net\_bitstamp\_native\_units": **-40.08343203**, "flow\_net\_bittrex\_usd": **-5970099.9528235365**, "flow\_net\_bittrex\_native\_units": **-278.08852827**, "flow\_net\_gemini\_usd": **-16741293.93537048**, "flow\_net\_gemini\_native\_units": **-779.81303975**, "flow\_net\_huobi\_usd": **-3382709.713590355**, "flow\_net\_huobi\_native\_units": **-157.56733945**, "flow\_net\_kraken\_usd": **-6915802.116906431**, "flow\_net\_kraken\_native\_units": **-322.13953664**, "flow\_net\_poloniex\_usd": **579429.3175268096**, "flow\_net\_poloniex\_native\_units": **26.98994111**, "flow\_out\_binance\_usd": **89636721.39080209**, "flow\_out\_binance\_native\_units": **4175.29758756**, "flow\_out\_bitfinex\_usd": **35060197.40891622**, "flow\_out\_bitfinex\_native\_units": **1633.11146804**, "flow\_out\_bitmex\_usd": **10050446.488991043**, "flow\_out\_bitmex\_native\_units": **468.15193961**, "flow\_out\_bitstamp\_usd": **6498482.5947404625**, "flow\_out\_bitstamp\_native\_units": **302.70070435**, "flow\_out\_bittrex\_usd": **12244833.755374897**, "flow\_out\_bittrex\_native\_units": **570.36696619**, "flow\_out\_gemini\_usd": **19820850.82465882**, "flow\_out\_gemini\_native\_units": **923.25945603**, "flow\_out\_huobi\_usd": **27059932.94513072**, "flow\_out\_huobi\_native\_units": **1260.45744414**, "flow\_out\_kraken\_usd": **48718829.6701498**, "flow\_out\_kraken\_native\_units": **2269.33346997**, "flow\_out\_poloniex\_usd": **928705.2051462886**, "flow\_out\_poloniex\_native\_units": **43.2592864**, "supply\_exchange\_native\_units": **1601652.47078366**, "supply\_exchange\_usd": **34384824860.45352**, "supply\_binance\_usd": **13739478572.831951**, "supply\_binance\_native\_units": **639987.84035584**, "supply\_bitfinex\_usd": **5807392801.479236**, "supply\_bitfinex\_native\_units": **270509.59448097**, "supply\_bitmex\_usd": **1754794236.8335319**, "supply\_bitmex\_native\_units": **81738.68956866**, "supply\_bitsta

mp\_usd": 903702147.3717782,"supply\_bitstamp\_native\_units": 42094.63863  
971,"supply\_bittrex\_usd": 2927100066.7358856,"supply\_bittrex\_native\_un  
its": 136344.94499085,"supply\_gemini\_usd": 4272320977.0448275,"supply\_  
gemini\_native\_units": 199005.62171351,"supply\_huobi\_usd": 383718135.70  
443124,"supply\_huobi\_native\_units": 17873.67254682,"supply\_kraken\_usd"  
: 2440387575.5581484,"supply\_kraken\_native\_units": 113673.77341386,"su  
pply\_poloniex\_usd": 466416440.51718575,"supply\_poloniex\_native\_units":  
21725.77721132},"miner\_flows": {"supply\_1hop\_miners\_usd": 55008164400  
.64765,"supply\_1hop\_miners\_native\_units": 2562292.02222579,"supply\_min  
ers\_usd": 38704580700.36332,"supply\_miners\_native\_units": 1802867.6185  
1969},"supply\_activity": {"supply\_active\_10y": 16558319.58220977,"supp  
ly\_active\_180d": 4726108.09532285,"supply\_active\_1d": 330879.22292725,  
"supply\_active\_1y": 6681815.42357099,"supply\_active\_1y\_percent": 35.02  
55414441,"supply\_active\_2y": 10545398.88907075,"supply\_active\_30d": 21  
33516.39893527,"supply\_active\_3y": 11824036.66767733,"supply\_active\_4y  
": 13129252.46010115,"supply\_active\_5y": 14537889.44163878,"supply\_act  
ive\_7d": 986514.26014328,"supply\_active\_90d": 3282466.37840322,"supply  
\_active\_ever": 17301214.36962235,"outstanding": 10377.16010271,"supply  
\_revived\_1y": 9475.29020828,"supply\_revived\_2y": 8996.25877794,"supply  
\_revived\_30d": 13212.3206333,"supply\_revived\_3y": 8837.66482022,"suppl  
y\_revived\_4y": 8727.55238604,"supply\_revived\_5y": 69.40038786,"supply\_  
revived\_7d": 23127.28310007,"supply\_revived\_90d": 11410.61351907},"sup  
ply\_distribution": {"supply\_in\_addresses\_balance\_greater\_0\_001\_native\_  
units": 19069927.81698027,"supply\_in\_addresses\_balance\_greater\_0\_01\_na  
tive\_units": 19030083.10468412,"supply\_in\_addresses\_balance\_greater\_0\_  
1\_native\_units": 18812443.3113975,"supply\_in\_addresses\_balance\_greater\_  
\_1\_usd": 19074240.74907869,"supply\_in\_addresses\_balance\_greater\_10\_usd  
": 19072425.17123784,"supply\_in\_addresses\_balance\_greater\_100\_usd": 19  
053657.62315925,"supply\_in\_addresses\_balance\_greater\_100k\_usd": 168219  
76.84582093,"supply\_in\_addresses\_balance\_greater\_100k\_native\_units": 7  
77451.90268887,"supply\_in\_addresses\_balance\_greater\_100\_native\_units":  
11862096.42335457,"supply\_in\_addresses\_balance\_greater\_10k\_usd": 1836  
9608.83166206,"supply\_in\_addresses\_balance\_greater\_10k\_native\_units":  
2956241.75187983,"supply\_in\_addresses\_balance\_greater\_10m\_usd": 983763  
2.46917643,"supply\_in\_addresses\_balance\_greater\_10\_native\_units": 1613  
3717.28899907,"supply\_in\_addresses\_balance\_greater\_1k\_usd": 18923602.1  
638359,"supply\_in\_addresses\_balance\_greater\_1k\_native\_units": 8001764.  
95289323,"supply\_in\_addresses\_balance\_greater\_1m\_usd": 14556055.068724  
61,"supply\_in\_addresses\_balance\_greater\_1m\_native\_units": 0,"supply\_in\_  
\_addresses\_balance\_greater\_1\_native\_units": 17958329.30395899,"supply\_

```
in_contracts_usd": null,"supply_in_contracts_native_units": null,"supply_shielded": null,"supply_in_top_100_addresses": 2976147.72292183,"supply_in_top_10_percent_addresses": 18865689.82822652,"supply_in_top_1_percent_addresses": 17355024.92143969,"supply_in_utxo_in_loss": 8381586.17875128,"supply_in_utxo_in_profit": 10695393.49973322},"alert_messages": null}}
```

## Annexe 2 : Bot fondamental

```
from six.moves import urllib
import json
import time
from pycoingecko import CoinGeckoAPI
import requests

def get_result_NVT(market_cap, transaction_count):
    if market_cap is None or transaction_count is None:
        return 0
    NVT = market_cap / transaction_count
    if NVT < 90:
        return -1
    elif NVT > 90:
        return 1
    return 0

def get_result_MRVR(realized_cap, market_cap):
    if realized_cap is None or market_cap is None:
        return 0
    MRVR = market_cap / realized_cap
    if MRVR < 3.7:
        return -1
    elif MRVR > 3.7:
        return 1
    return 0

def get_result_ratio_hashrate(market_cap, hashrate, btc_information):
    if market_cap is None or hashrate is None:
        return 0
    hashrate_ratio_btc = btc_information["hash_rate"] /
    btc_information["market_cap"]
    hashrate_ratio_specific_coin = hashrate / market_cap
    if hashrate_ratio_specific_coin > hashrate_ratio_btc:
        return -1
    elif hashrate_ratio_specific_coin < hashrate_ratio_btc:
        return 1
    return 0

def get_result_ratio_avg_transaction_value (market_cap,
transaction_value_usd, btc_information):
    if market_cap is None and transaction_value_usd is None:
        return 0
```

```

ratio_avg_transaction_value_btc =
btc_information["transaction_value_usd"] / btc_information["market_cap"]
ratio_avg_transaction_value_specific_coin =
ratio_avg_transaction_value_btc / market_cap
    if ratio_avg_transaction_value_specific_coin >
ratio_avg_transaction_value_btc:
        return -1
    elif ratio_avg_transaction_value_specific_coin <
ratio_avg_transaction_value_specific_coin:
        return 1
    return 0

def get_result_ratio_active_addresses (market_cap, active_addresses,
btc_information):
    if market_cap is None and active_addresses is None:
        return 0
    ratio_active_addresses_value_btc =
btc_information["active_addresses"] / btc_information["market_cap"]
    ratio_active_addresses_value_specific_coin = active_addresses /
market_cap
    if ratio_active_addresses_value_specific_coin >
ratio_active_addresses_value_btc:
        return -1
    elif ratio_active_addresses_value_specific_coin <
ratio_active_addresses_value_btc:
        return 1
    return 0

def get_information_about_coin(coin):
    print(coin)
    information_of_coin = {}
    api_key_messari = ""
    url = "https://data.messari.io/api/v1/assets/{}/metrics".format(coin)
    requete = urllib.request.Request(url)
    requete.add_header("x-messari-api-key", api_key_messari)
    data = urllib.request.urlopen(requete).read()
    data_decode = json.loads(data.decode('UTF-8'))
    transaction_count =
data_decode["data"]["on_chain_data"]["transfer_count_last_24_hours"]
    transaction_value_usd =
data_decode["data"]["on_chain_data"]["average_fee_usd"]
    active_addresses =
data_decode["data"]["on_chain_data"]["active_addresses"]
    hash_rate = data_decode["data"]["on_chain_data"]["hash_rate"]
    # pour la team regarder dans la description du projet pour voir
    combien de fois apparait le mot team.

```

```

    market_cap =
data_decode["data"]["marketcap"]["current_marketcap_usd"]
    circulating_supply = data_decode["data"]["supply"]["circulating"]
    volume_last_24_hours =
data_decode["data"]["market_data"]["volume_last_24_hours"]
    realized_marketcap_usd =
data_decode["data"]["marketcap"]["realized_marketcap_usd"]
    information_of_coin["transaction_count"] = transaction_count
    information_of_coin["transaction_value_usd"] = transaction_value_usd
    information_of_coin["active_addresses"] = active_addresses
    information_of_coin["hash_rate"] = hash_rate
    information_of_coin["circulating_supply"] = circulating_supply
    information_of_coin["market_cap"] = market_cap
    information_of_coin["volume_last_24_hours"] = volume_last_24_hours
    information_of_coin["realized_marketcap_usd"] =
realized_marketcap_usd
    print(information_of_coin)
    return information_of_coin

def get_final_result_coin (coin, btc_information):
    result = 0
    information_of_coin = get_information_about_coin(coin)
    transaction_count = information_of_coin["transaction_count"]
    transaction_value_usd = information_of_coin["transaction_value_usd"]
    active_addresses = information_of_coin["active_addresses"]
    hash_rate = information_of_coin["hash_rate"]
    circulating_supply = information_of_coin["circulating_supply"]
    market_cap = information_of_coin["market_cap"]
    volume_last_24_hours = information_of_coin["volume_last_24_hours"]
    realized_marketcap_usd =
information_of_coin["realized_marketcap_usd"]
    result += get_result_NVT(market_cap, transaction_count)
    result += get_result_MRVR(realized_marketcap_usd, market_cap)
    # result += get_result_ratio_hashrate(market_cap, hash_rate,
btc_information)
    result += get_result_ratio_avg_transaction_value(market_cap,
transaction_value_usd, btc_information)
    result += get_result_ratio_active_addresses(market_cap,
active_addresses, btc_information)
    return {"coin": coin, "result": result}

def best_fundamentals_crypto_save_db(list_dict):
    max_ranking = 10
    list_max = list_dict[0:max_ranking-1]

```

```

str = ""
for i in range(len(list_max)):
    str += list_max[i]["coin"] + ","

with open('10_best_crypto_fundamentals.txt', 'w') as f:
    f.write(str)

def get_list_crypto_top_100(API_KEY):
    headers = {
        'X-CMC_PRO_API_KEY': API_KEY
    }
    list_symbol_100 = []
    # url = 'https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest?limit=500'
    url = 'https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest'
    r = requests.get(url, headers=headers)
    find_ranks = {}
    if r.status_code == 200:
        data = r.json()
        for d in data['data']:
            symbol = d['symbol']
            find_ranks[symbol] = d['cmc_rank']
            list_symbol_100.append(symbol)
    return list_symbol_100

def get_crypto_fundamental_classement(total_list_crypto):
    classement_croissant = sorted(total_list_crypto, key=lambda d:
d['result'])
    return classement_croissant

if __name__ == '__main__':
    #coingecko
    cg = CoinGeckoAPI()
    coin_list = cg.get_coins_list()

    #coinmarketcap
    API_KEY_COINMARKETCAP = ''
    list_crypto_top_100 = get_list_crypto_top_100(API_KEY_COINMARKETCAP)

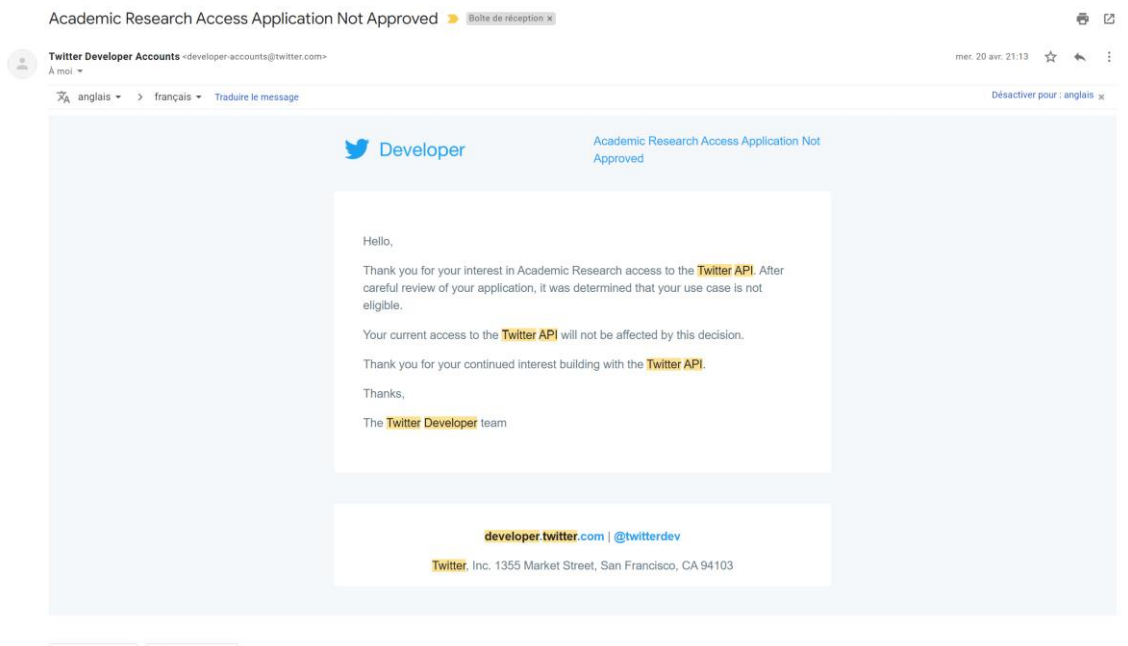
    #messari
    api_key_messari = ""
    cpt = 1
    btc_information = get_information_about_coin("BTC")

```

```
time.sleep(2)
my_total_list = []
for coin in list_crypto_top_100:
    try:
        if "USD" not in coin:
            my_total_list.append(get_final_result_coin(coin,
btc_information))
            time.sleep(3)
            cpt += 1
    except Exception:
        pass

classement_final = get_crypto_fundamental_classement(my_total_list)
print(classement_final)
best_fundamentals_crypto_save_db(classement_final)
```

## Annexe 3 : Refus pour l'API étudiante de twitter



## Annexe 4 : Tableau des commissions

Exchange	Trading Fees			Funding Fees		Discounts	
	Maker	Taker	Spread	Deposits	Withdrawals	Exchange Token Discount	Volume Discount
<a href="#">Bibox</a>	0.075%	0.15%	No	No	Yes	Yes	No
<a href="#">Binance.us</a>	0.1%	0.1%	No	No	No	Yes - 25%	Yes
<a href="#">Binance.com</a>	0.1%	0.1%	No	No	Yes	Yes - 25%	Yes
<a href="#">Bitfinex</a>	0.1%	0.2%	No	No	Yes	No	Yes
<a href="#">Bitstamp</a>	0.5%		No	No	Yes	No	Yes
<a href="#">Bittrex</a>	0.35%	0.35%	No	No	Yes	No	Yes
<a href="#">Bitmex</a>	0.025%	0.075%	No	No	No	No	Yes
<a href="#">Bitpanda.pro</a>	0.1%	0.15%	No	No	No	No	Yes
<a href="#">BlockFi</a>	No trading fees			No	Yes	No	N/A
<a href="#">Blockchain.com</a>	0.2%	0.4%	No	No	No	No	Yes
<a href="#">BTC Markets</a>	0.05%	0.2%	No	No	Yes (AUD Free)	No	Yes
<a href="#">Bybit.com</a>	-0.025% (Rebate)	0.075%	No	No	No	No	No
<a href="#">Cash App</a>	Service fee + additional fee (Exact fees are not disclosed on the website)						
<a href="#">CEX.IO</a>	0.16%	0.25%	No	No	Yes	No	Yes
<a href="#">Coinbase</a>	N/A	The greater of flat fee (\$1.49, \$1.99 & \$2.99) or 1.49%	-0.50% fiat 1.00% crypto	No	No	No	Yes
<a href="#">Coinbase Pro</a>	0.5%	0.5%	No	No	No	No	Yes
<a href="#">Coinmama</a>	Buy: <a href="#">XBX Index</a> average + 2% (standard deviation) Sell: <a href="#">XBX Index</a> average + 2% (standard deviation) + 0.1% - 0.9% sell fee					No	No
<a href="#">Coinspot</a>	0.1%	0.1%	No	Yes	Yes (AUD Free)	No	No
<a href="#">crypto.com</a>	0.1%	0.16%	No	No	Yes	No	Yes
<a href="#">dYdX</a>	0%	Higher of 0.3% or Gas cost	No	No	No	No	No
<a href="#">eToro</a>	No trading fees		Yes - 0.75%	No	No	No	No
<a href="#">FTX</a>	0.02%	0.07%	No	No	No	Yes	Yes
<a href="#">Gate.io</a>	0.2%	0.2%	No	No	Yes	Yes	Yes

<a href="#">Gemini</a>	The greater of flat fee (\$0.99, \$1.49, \$1.99 & \$2.99) or 1.49%		No	No	No	No	Yes
<a href="#">HitBTC</a>	0.1%	0.25%	No	No	No	No	Yes
<a href="#">Huboi</a>	0.2%	0.2%	No	No	No	Yes	Yes
<a href="#">IDEX</a>	0.1%	0.2%	No	No	No	No	No
<a href="#">Independent Reserve</a>	Trade commission - 0.5%			Yes	Yes (AUD Free)	No	Yes
<a href="#">Kraken</a>	0.16%	0.26%	No	No	No	No	Yes
<a href="#">Kucoin</a>	0.1%	0.1%	No	No	Yes	No	Yes
<a href="#">Liquid</a>	0.29%	0.29%	No	No	Yes	Yes	Yes
<a href="#">Luno</a>	0%	0.1%	No	No	No	No	Yes
<a href="#">OKEx</a>	0.1%	0.15%	No	No	No	No	Yes
<a href="#">PayPal</a>	0.50 Fixed fee or variable rate of 2.3%		Yes	No	No	No	Yes
<a href="#">Poloniex</a>	0.125%	0.125%	No	No	Yes	No	Yes
<a href="#">River</a>	Buy & Sell: Fixed fee (\$0.99 - \$2.99) or (1.5% - 1.3%)		Yes	No	No	No	Yes
<a href="#">Robinhood</a>	No trading fees			No	N/A	No	No
<a href="#">Shakepay</a>	No trading fees			No	No	No	No
<a href="#">swyftx</a>	0.6% + spread		Yes - 0.41%	No	No	No	Yes
<a href="#">Uniswap</a>	0.3% for swaps			No	No	No	No
<a href="#">Uphold</a>	N/A	N/A	Yes (0.8% - 1.2% for BTC, ETH & XRP)  0.2% for EUR, USD and GBP.	No	No	No	No

(<https://medium.com/cointracker/2019-cryptocurrency-exchange-fee-survey-49db9b38cb5b>)

## Annexe 5 : Bot de trading

```
from pip import main
from regex import F
import tweepy
import config
from textblob import TextBlob
import pandas as pd
import numpy as np
import re
from transformers import pipeline
import requests
from datetime import datetime
from spot_ftx import SpotFtx
import ta
import ccxt
import time
import telegram_send

def cleanTxt(text):
    text = re.sub(r'@[A-Za-z0-9]+', '', text)#enleve les mentions
    text = re.sub(r'#', '', text)#enleve les #
    text = re.sub(r'RT[\s]+', '', text)#enleves les rt
    text = re.sub(r'https?:/\s+', '', text)# enleve les hyper liens
    text = re.sub(r'\n', '', text)# enleve les sauts à la ligne
    text = re.sub(r':', '', text)# enleve les doubles points
    return text

def getSubjectivity(text):
    return TextBlob(text).sentiment.subjectivity

def getPolarity(text):
    return TextBlob(text).sentiment.polarity

def getAnalysisTransformers(text):
    classfier = pipeline('sentiment-analysis')
    return classfier(text)[0]['label']

def getAnalysis(score):
    if score < 0:
        return 'Negative'
    elif score == 0:
        return 'Neutral'
    else:
        return 'Positive'
```

```

def getFinalSentimentResultTwitter(crypto):
    client = tweepy.Client(bearer_token=config.BEARER_TOKEN)
    query = crypto + " crypto"
    response = client.search_recent_tweets(query=query, max_results=100)
    ma_list = response.data
    df = pd.DataFrame([str(tweet) for tweet in ma_list],
columns=["Tweets"])
    df.head()
    df['Tweets']=df['Tweets'].apply(cleanTxt)
    # df['Subjectivity'] = df['Tweets'].apply(getSubjectivity)
    # df['Polarity'] = df['Tweets'].apply(getPolarity)
    # print(df)
    # df['Analysis'] = df['Polarity'].apply(getAnalysis)
    df['Analysis_transformers'] =
df['Tweets'].apply(getAnalysisTransformers)
    cpt_positive = 0
    cpt_negatif = 0
    sortedDF = df.sort_values(by=['Analysis_transformers'])
    for i in range(0, sortedDF.shape[0]):
        if sortedDF['Analysis_transformers'][i] == 'POSITIVE':
            cpt_positive += 1
        elif sortedDF['Analysis_transformers'][i] == 'NEGATIVE':
            cpt_negatif += 1
    print("cpt positive :", cpt_positive)
    print("cpt négatif :", cpt_negatif)

    ratio = cpt_negatif / 100
    if cpt_positive > cpt_negatif or ratio > 0.80:
        return True
    return False

def getFinalSentimentResultGoogleNews(crypto):

    api_key_google = ""
    query = crypto + " crypto"
    start_date = datetime.today().strftime('%Y-%m-%d')
    url = ('https://newsapi.org/v2/everything?'
        'q=' + query + '&'
        'from=' + start_date + '&'
        'sortBy=popularity&'
        'apiKey=' + api_key_google)
    response = requests.get(url)
    dict_reponses = response.json()
    articles = dict_reponses['articles']

```

```

for i in articles:
    print(i['description'])

df = pd.DataFrame([article['description'] for article in articles],
columns=["Article"])
df.head()

df['Article']=df['Article'].apply(cleanTxt)
df['Subjectivity'] = df['Article'].apply(getSubjectivity)
df['Polarity'] = df['Article'].apply(getPolarity)
df['Analysis'] = df['Polarity'].apply(getAnalysis)
df['Analysis_transformers'] =
df['Article'].apply(getAnalysisTransformers)
print(df)

cpt_positive = 0
cpt_negatif = 0

sortedDF = df.sort_values(by=['Analysis_transformers'])
for i in range(0, sortedDF.shape[0]):
    if sortedDF['Analysis_transformers'][i] == 'POSITIVE':
        cpt_positive += 1
    elif sortedDF['Analysis_transformers'][i] == 'NEGATIVE':
        cpt_negatif += 1

print("cpt positive :", cpt_positive)
print("cpt négatif :", cpt_negatif)

ratio = cpt_negatif / 100
if cpt_positive > cpt_negatif or ratio > 0.80:
    return True
return False

def read_db_file():
    with open('10_best_crypto_fundamentals.txt', 'r') as f:
        line = f.readline()
        tab = line.split(",")
        for i in range(len(tab)):
            tab[i] = tab[i] + "/USD"
    print(tab[:len(tab)-1])
    return tab[:len(tab)-1]

if __name__ == '__main__':
    now = datetime.now()

```

```

print(now.strftime("%d-%m %H:%M:%S"))

ftx = SpotFtx(
    apiKey='',
    secret='',
    subAccountName='t'
)

pairList = read_db_file()

timeframe = '1h'

# -- Indicator variable --
aoParam1 = 6
aoParam2 = 22
stochWindow = 14
willWindow = 14

# -- Hyper parameters --
maxOpenPosition = 3
stochOverBought = 0.8
stochOverSold = 0.2
willOverSold = -85
willOverBought = -10
TpPct = 0.15

dfList = {}
for pair in pairList:
    # print(pair)
    df = ftx.get_last_historical(pair, timeframe, 210)
    dfList[pair.replace('/USD','')] = df

for coin in dfList:
    # -- Drop all columns we do not need --
    dfList[coin].drop(columns=dfList[coin].columns.difference(['open',
    'high', 'low', 'close', 'volume']), inplace=True)

    # -- Indicators, you can edit every value --
    dfList[coin]['AO'] =
ta.momentum.awesome_oscillator(dfList[coin]['high'],dfList[coin]['low'],w
indow1=aoParam1,window2=aoParam2)
    dfList[coin]['STOCH_RSI'] =
ta.momentum.stochrsi(close=dfList[coin]['close'], window=stochWindow)
    dfList[coin]['WillR'] =
ta.momentum.williams_r(high=dfList[coin]['high'],
low=dfList[coin]['low'], close=dfList[coin]['close'], lbp=willWindow)

```

```

dfList[coin]['EMA100']
=ta.trend.ema_indicator(close=dfList[coin]['close'], window=100)
dfList[coin]['EMA200']
=ta.trend.ema_indicator(close=dfList[coin]['close'], window=200)

print("Data and Indicators loaded 100%")

# -- Condition to BUY market --
def buyCondition(row, name_crypto, previousRow=None):
    if (
        row['AO'] >= 0
        and previousRow['AO'] > row['AO']
        and row['WillR'] < willOverSold
        and row['EMA100'] > row['EMA200']
        and getFinalSentimentResultGoogleNews(name_crypto)
        and getFinalSentimentResultTwitter(name_crypto)
    ):
        return True
    else:
        return False
# -- Condition to SELL market --
def sellCondition(row):
    if (
        (row['AO'] < 0
        and row['STOCH_RSI'] > stochOverSold)
        or row['WillR'] > willOverBought
    ):
        return True
    else:
        return False
coinBalance = ftx.get_all_balance()
coinInUsd = ftx.get_all_balance_in_usd()
usdBalance = coinBalance['USD']
del coinBalance['USD']
del coinInUsd['USD']
totalBalanceInUsd = usdBalance + sum(coinInUsd.values())
coinPositionList = []
for coin in coinInUsd:
    if coinInUsd[coin] > 0.05 * totalBalanceInUsd:
        coinPositionList.append(coin)
openPositions = len(coinPositionList)

#Sell
for coin in coinPositionList:
    if "USD" not in coin and sellCondition(dfList[coin].iloc[-2])
== True:

```

```

        openPositions -= 1
        symbol = coin+'/USD'
        cancel = ftx.cancel_all_open_order(symbol)
        time.sleep(1)
        sellPrice = float(ftx.convert_price_to_precision(symbol,
ftx.get_bid_ask_price(symbol)['ask']))
        sell =
ftx.place_market_order(symbol,'sell',coinBalance[coin])
        print(cancel)
        print("Sell", coinBalance[coin], coin, sell)
        telegram_send.send(messages=["Sell " +
str(coinBalance[coin]) + " " + coin + " at " + str(sellPrice)])
        else:
            print("Keep",coin)

#Buy
    if openPositions < maxOpenPosition:
        for coin in dfList:
            if coin not in coinPositionList:
                if buyCondition(dfList[coin].iloc[-2], coin,
dfList[coin].iloc[-3]) == True and openPositions < maxOpenPosition:
                    time.sleep(1)
                    usdBalance = ftx.get_balance_of_one_coin('USD')
                    symbol = coin+'/USD'

                    buyPrice =
float(ftx.convert_price_to_precision(symbol,
ftx.get_bid_ask_price(symbol)['ask']))
                    tpPrice =
float(ftx.convert_price_to_precision(symbol, buyPrice + TpPct *
buyPrice))

                    buyQuantityInUsd = usdBalance * 1/(maxOpenPosition-
openPositions)

                    if openPositions == maxOpenPosition - 1:
                        buyQuantityInUsd = 0.95 * buyQuantityInUsd

                    buyAmount = ftx.convert_amount_to_precision(symbol,
buyQuantityInUsd/buyPrice)

                    buy = ftx.place_market_order(symbol,'buy',buyAmount)
                    time.sleep(2)
                    tp =
ftx.place_limit_order(symbol,'sell',buyAmount,tpPrice)
                    try:
                        tp["id"]
                    except:

```

```
        time.sleep(2)
        tp =
ftx.place_limit_order(symbol,'sell',buyAmount,tpPrice)
        pass
        print("Buy",buyAmount,coin,'at',buyPrice,buy)
        print("Place",buyAmount,coin,"TP at",tpPrice, tp)
        telegram_send.send(messages=["Buy " + str(buyAmount)
+ " " + str(coin) + ' at ' + str(buyPrice)])

        openPositions += 1
```