

# **Rapport de stage**

**Lejolviet Noah**

**2025**

**SN<sub>2</sub>**

# Sommaire

## **1 - Introduction**

- **Présentation de « l'entreprise » et des missions**

## **2 - Algorithmes de Trading**

- **Stratégies et technologies utilisées**

## **3 - Backtest des Algorithmes**

- **Objectifs, mise en œuvre et analyse des résultats**

## **4 - Développement d'un Indicateur de Trading**

- **Programme Python pour la collecte de données et affichage en HTML/JS.**

## **5 - Conclusion**

- **Bilan du stage et perspectives d'amélioration**

## **6 - Annexes**

- **Sources et documentation**

## **1- Introduction**

**J'ai effectué ce stage dans mon entreprise après avoir contacté plus d'une trentaine de boîtes mêmes en Angleterre, aucune n'ayant donné une réponse favorable...**

**Si vous souhaitez vérifier je vous donne le nom de certaines boîtes (Wintermute, Oakland Search, BNP, Société Générale, CMB, LSEG...) (preuves appuyables)**

**Parmi mes 3 missions principales, il y avait :**

- Développer un algorithme de Trading Automatique basé sur des conditions spécifiques pour effectuer des ordres sur les marchés financiers**
- Réaliser un script de Backtest permettant de vérifier sur une période d'au moins 1 an auparavant la fiabilité de cette stratégie**
- Réaliser une interface web avec au moins un indicateur et un script python qui récupère la data et la traite en fonction de plusieurs choix de l'utilisateur**

## **2- Algorithme de trading**

**Il y a un algorithme principal que j'ai réalisé en python, ce bot utilise un fichier de config avec des identifiants MetaTraders qui est une application téléphone/pc pour trader sur les marchés financiers. Pendant toute la période de développement un compte de démonstration avec un montant fictif a été utilisé.**

**Dépôt git ([https://gitlab.com/noah\\_ljvt/trading-bots](https://gitlab.com/noah_ljvt/trading-bots))**

**Autre Dépôt plus ancien de premier bot (ne sera pas évoqué par la suite :**

**[https://gitlab.com/noah\\_ljvt/mts\\_basic\\_bot](https://gitlab.com/noah_ljvt/mts_basic_bot))**

**Dans le bot principal nous avons :**

- L'utilisation de Python et principalement de la librairie MetaTraders**
- L'utilisation de l'application de Bureau MetaTraders (Ce bot ne fonctionnera pas sous Linux car la bibliothèque ne fonctionne pas sous Linux, malgré une version qui s'en approche nommée mtslinux mais qui n'a pas fonctionné pour moi après avoir passé pas mal de temps à pourtant essayer de la faire fonctionner)**

## **Fonctionnement**

**Le bot utilise l'application de bureau MetaTraders pour récupérer la data des bougies automatiquement (ouverture, fermeture, haut et bas de chaque session en fonction de la Time Frame (période de temps choisie, 5min, 1h...))**

**Il va ensuite effectuer des calculs pour en sortir des indicateurs, dans mon cas un EMA, un RSI (indicateur développé par la suite dans l'autre partie de projet) et un MACD, je ne vais pas m'attarder sur le côté Trading mais je vais vous mettre des liens si vous souhaitez vous renseigner pour mieux comprendre. En clair les indicateurs en Trading donnent des informations sur le marché a un instant T, si il parait Suracheté/Survenu, si la tendance est Haussière etc..**

**Les liens ci-dessous expliquent d'une manière ludique et simple comment les indicateurs marchent, il y a de quoi complètement se former sur le domaine du Forex et de la Cryptomonnaie avec plus de 700 leçons.**

**EMA :** <https://www.babypips.com/learn/forex/exponential-moving-average>

**MACD :**

<https://www.babypips.com/learn/forex/macd>

**RSI :**

<https://www.babypips.com/learn/forex/relative-strength-index>

**Pour faire simple, quand le Ema 20, plus réactif au marché est au-dessus du Ema 50, 100 et 200, moins réactif au marché dans l'ordre chronologique**

(illustration :



]

, que le RSI est entre 52 et 72 et que le MACD donne un signal haussier car il est supérieur à sa dernière valeur, le bot va acheter le marché avec un risque géré automatiquement à savoir en mettant un Stop Loss pour limiter la perte et un Take Profit pour encaisser les gains bien prédéfinis sur une base de ratio de 1/5, il faut donc au-dessus de 20% de ratio de gains pour être rentable.

Les mêmes conditions dans l'autre sens effectuent une vente sur les mêmes critères.

L'analyse pour savoir si on achète, vend ou ne faisons rien est basée sur la time frame choisie dans le code, si la time frame est de 15 minutes par exemple, toutes les 15 minutes pile, c'est-à-dire a

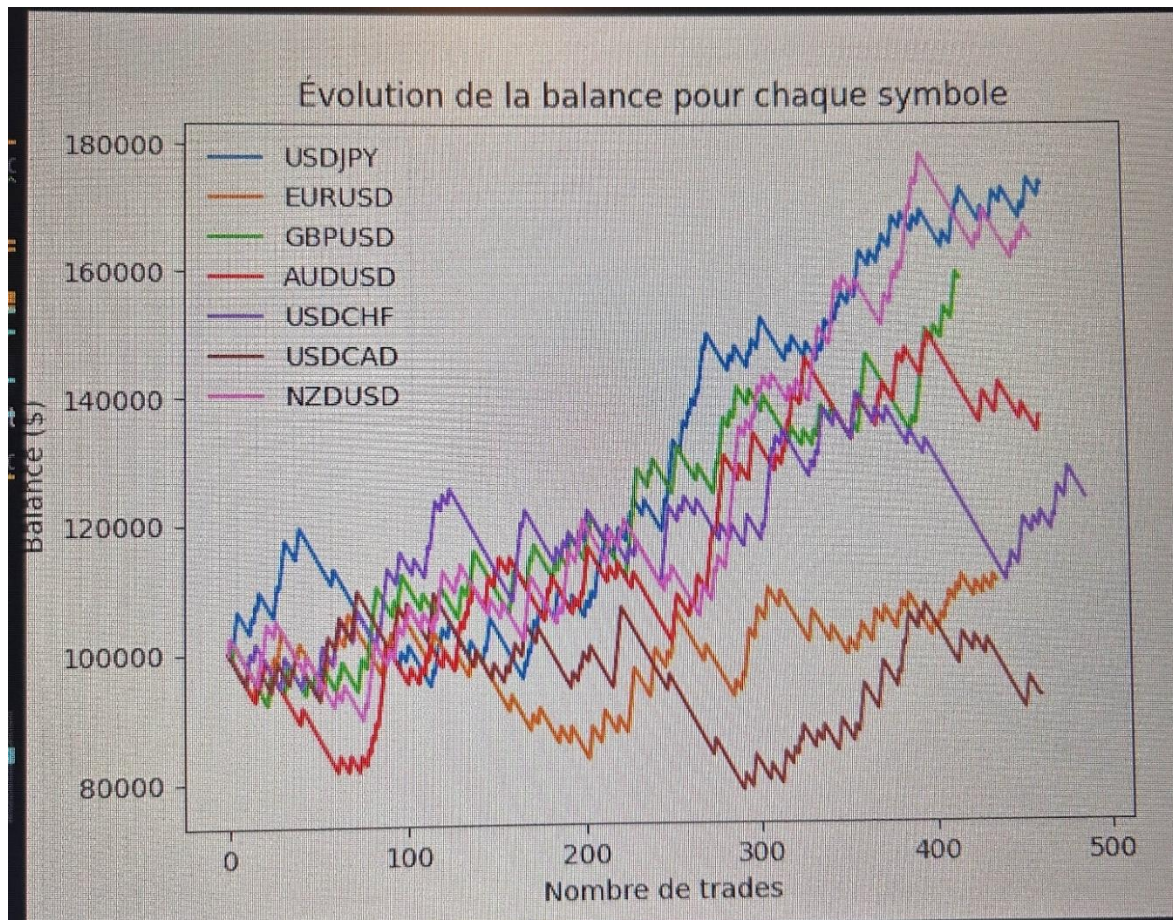
**chaque nouvelle bougie, le script tourne 1 fois et attend les prochaines 15 minutes piles.**

**Exemple :**

**Je lance le bot a 14 :10, en ayant mis la time frame 15min, il va attendre 5 minutes et à 14 :15 :00 il va regarder si il doit effectuer une action avant de ré-attendre 15 minutes tout pile.**

### 3- Backtest de l'Algorithme et Résultats

Mon script de backtest principal nommé `backtest_usd_jpy_reel.py` est en réalité le même que le main mais qui au lieu de regarder maintenant si il doit agir il fait comme si il avait commencé il y a environ 1 an et avait tourné pendant toute cette période en simulant. Ce script a été backtesté seulement sur la paire USD/JPY car après avoir fait un script il y a longtemps (j'en ai perdu toute trace de code mais il me reste un screenshot que j'avais pris :



*ce script backtestait pour chaque paire me sortait les résultats et USD/JPY était la plus performante sur la période donnée.*



**Le backtest s'effectue sur une période de 1 an et 3 mois, ajustable plus ou moins jusque 1.5 années max.**

**Cela simule et donne pour chaque trade pris, la date, l'heure, le prix d'entrée, le TP, le SL et si c'est un gain ou une perte, ainsi que la Balance actualisée après le trade.**

**Exemple :**

```
--- Simulation Trade ---  
Action : BUY, Entry : 143.73600, TP : 145.30500, SL : 143.42200  
Trade gagnant ! Gain : 2500, Nouvelle balance : 55000.00  
Signal détecté : BUY pour USDJPY à 2024-01-04 15:30:00  
  
--- Simulation Trade ---  
Action : BUY, Entry : 144.51500, TP : 146.08400, SL : 144.20100  
Trade perdant. Perte : 500, Nouvelle balance : 54500.00  
Signal détecté : BUY pour USDJPY à 2024-01-05 01:30:00
```

**Pour ce script la (backtest\_prop\_reel.py) le résultat est le suivant :**

```
Max balance : 393500.00 /// capital final: 360500.00.
```

**Avec une balance initiale de 50 000\$ il y a 1an et 3mois le script aujourd'hui atteint une balance de 360 000\$ (et maximum 400 000\$) ce qui fait environ un X7 mais il faut un capital de départ important...**

**J'en ai fait certains qui simulent les résultats mais sur les prop firms car les prop firms permettent de gagner plus d'argent avec des capitaux fictifs si les performances sont avec faible pertes et gains réguliers pour faire simple.**

**(Lien pour comprendre les prop firms  
<https://www.cafedutrading.com/trading/prop-firm>)**

**Pour faire simple on va avoir un compte avec 100k faux \$ qui coûte 500\$, le but est de faire 8% et 5% de gains sans**

**perdre 5% par jour et 10% au total, ce qui permet par la suite de reprendre un compte a 100k faux \$ mais chaque gain sera à 80% dans notre poche. Ce qui est intéressant mais la gestion du risque doit être plus stricte.**

**J'ai donc fait aussi un script qui teste avec les règles de prop firms et qui prendrait les profits des 10k \$ de sécurisés.**

**Et qui rachète un compte a 500\$ -> qui donne accès à 100k faux \$ en boucle.**

**Il me met ensuite un 'x' par paiement de 10k\$ qui aurait réussi et la balance finale du compte**

**Exemple d'un compte (le 8, avec un paiement de 10k\$ sorti pour un coût initial de 500\$) :**

**Balance finale du compte 8 : 67500.00 € x**

**Pour ce script la (backtest\_prop\_reel.py) le résultat est le suivant :**

```
Backtest terminé !
Nombre total de challenges effectués : 30
Coût total des challenges : 10350 €
Balance finale du compte 2 : 64500.00 €
Balance finale du compte 8 : 67500.00 € x
Balance finale du compte 10 : 62000.00 €
Balance finale du compte 11 : 94000.00 € xxx
Balance finale du compte 14 : 73000.00 € x
Balance finale du compte 16 : 65000.00 €
Balance finale du compte 18 : 64500.00 €
Balance finale du compte 20 : 60000.00 €
Balance finale du compte 21 : 65000.00 €
Balance finale du compte 25 : 69500.00 € x
Balance finale du compte 29 : 79000.00 € xx
```

**Il y aurait eu 10350\$ de comptes à payer au total mais 80 000\$ de sortis au total ce qui aurait été rentable d'environ 70 000\$.**

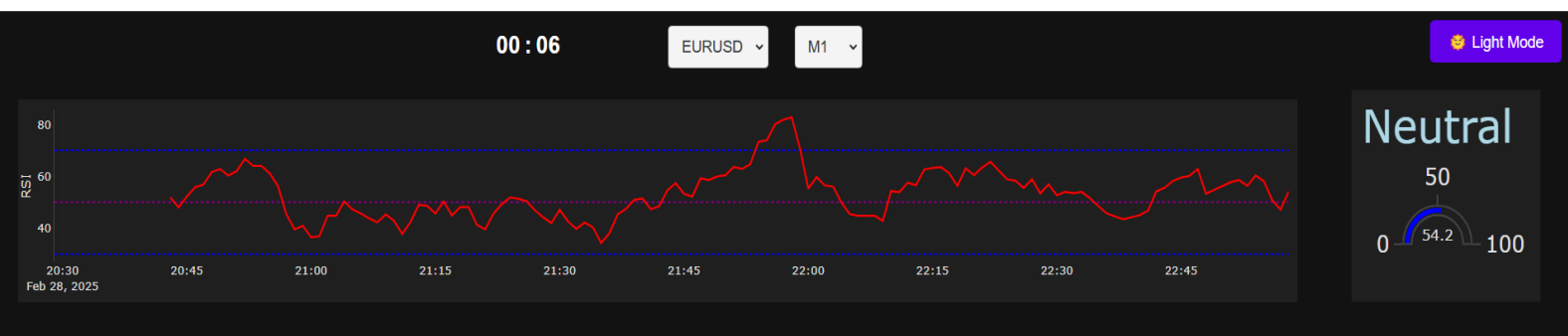
## 4- Développement d'un indicateur de Trading

(Dépôt git : [https://gitlab.com/noah\\_ljvt/trading\\_indicator](https://gitlab.com/noah_ljvt/trading_indicator))

Cet indicateur de Trading est composé de 2 parties majeures :

- Le script pour le fonctionnel en Python
- Le script pour l'affichage et la gestion dynamique des données en HTML/JS

Le site web ressemble à cela :



Il est possible de choisir une paire de monnaie et une time frame ainsi qu'un mode sombre ou non.

La data choisie sur la page web va ensuite envoyer au script Python les données à changer pour réexécuter le script. Si on change la TimeFrame ou la paire, on va renvoyer au script qui va se réexécuter et récupérer toujours via l'application de bureau MT5 la data des bougies (ouverture, fermeture, haut et bas de chaque session)

Le calcul de la formule du RSI pour mesurer la puissance du mouvement va s'effectuer pour avoir la donnée de l'indicateur RSI et en utilisant PlotLy on va afficher le graphique dynamiquement. Toujours avec PlotLy on affiche à droite un index qui prend juste la data du RSI à l'instant T et si c'est au-dessus de 70 on va avoir en rouge OverBought

**pour suracheté et donc il faudrait plutôt vendre et en dessous de 30 l'inverse.**

***L'objectif par la suite est de mettre à disposition un grand nombre d'indicateurs ainsi que le graphique complet de la paire avec combiné un bot qui tourne tout seul mais qui affiche sur le graphique les trades qu'il prend ce qui visuellement est plus parlant.***

## **5- Conclusion**

**Ce stage m'a permis de développer des compétences en trading algorithmique et en création d'outils financiers.**

**J'ai appris à concevoir un algorithme de trading automatique avec des indicateurs techniques (EMA, RSI, MACD) et des mécanismes de gestion des risques (*Stop Loss, Take Profit*).**

**Le backtest a montré un potentiel de rentabilité et l'importance du capital de départ.**

**J'ai aussi travaillé sur un indicateur de trading et une interface web en Python, HTML et JavaScript pour une analyse technique en temps réel. Cette expérience m'a fait comprendre les défis du trading automatisé et m'a permis d'acquérir des compétences réutilisables, bien que des améliorations soient possibles surtout dans l'optimisation des stratégies et l'interface utilisateur.**

## 6 – Annexes

- **BabyPips** : <https://www.babypips.com/>
- **TradeOxy** : Build your own trading bot (m'a servi pour le premier prototype dont j'ai dis ne plus parler par la suite)  
<https://www.youtube.com/watch?v=EkP7iAZoMEw&list=PLe6oBag-fEVclVn5A8kugGrHS37fOym91>
- **StackOverflow** pour MT5linux même si le problème n'a pas été résolu
- **PLotly Graph**  
<https://plotly.com/javascript/uirevision/>
- **Plotly Index**  
<https://plotly.com/javascript/indicator/>
- **TradingView Desktop** pour la vérification de la concordance de mon indicateur RSI avec l'indicateur RSI de l'application.