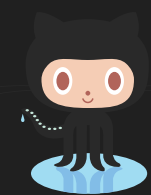




Groupe 8
Grégory Bournassenko
Salohy Rabariniaina



Projet open-source

github.com/excelliarmus/app

2 prérequis

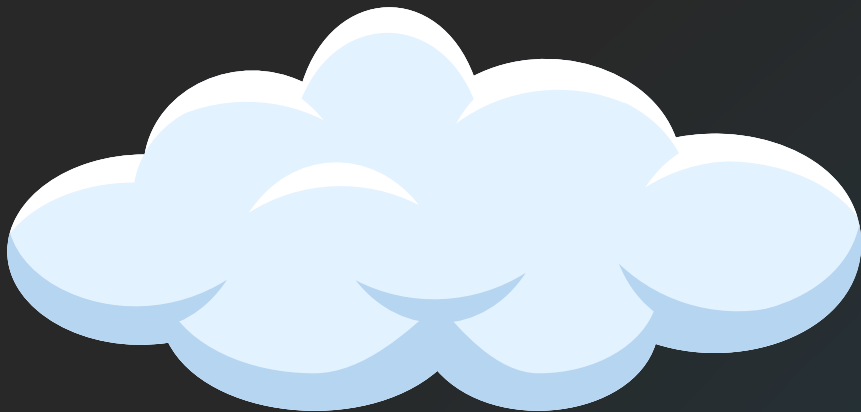
Requêtes HTTP

Machine Learning (KNN)

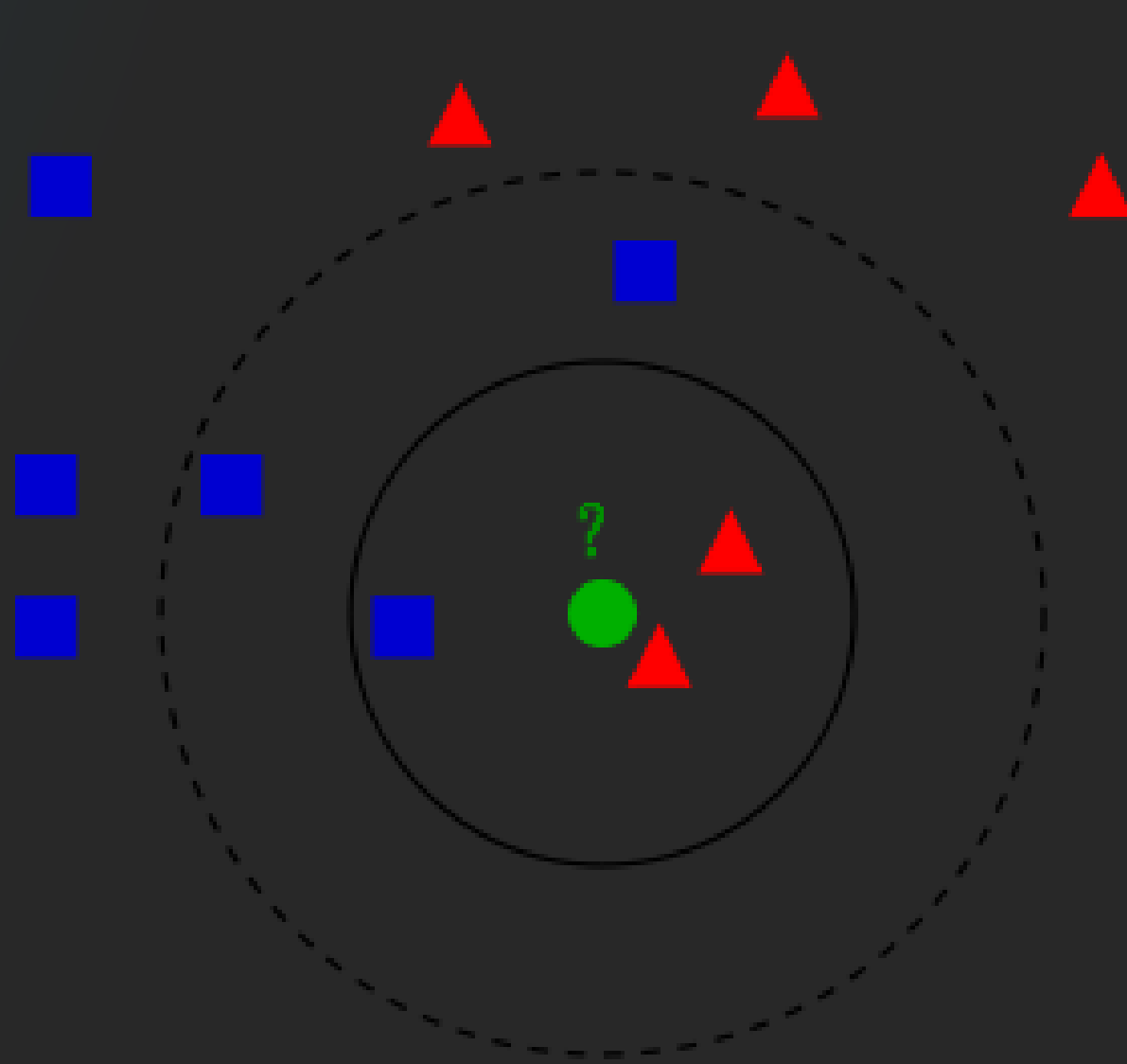
Requête HTTP

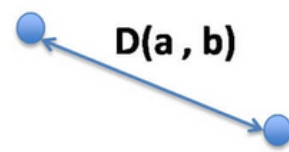
```
Sub writeData1(ticker As String)
  Dim xmlhttp As Object
  Set xmlhttp = CreateObject("MSXML2.ServerXMLHTTP.6.0")
  Dim json As Object
  Dim OHLCopen As Double
  Dim OHLChigh As Double
  Dim OHLClow As Double
  Dim OHLCclose As Double
  On Error GoTo noticker
  url = "https://api.binance.com/api/v3/klines?symbol=" & ticker & "&interval=1m&limit=100"
  xmlhttp.Open "GET", url, False
  xmlhttp.Send
  Set json = JsonConverter.ParseJson(xmlhttp.responseText)
```

JSON	Données brutes	En-têtes
Enregistrer	Copier	Tout réduire
Tout développer	Filtrer le JSON	
▼ 0:		
0:	1681456500000	
1:	"30900.90000000"	
2:	"30907.56000000"	
3:	"30900.00000000"	
4:	"30900.00000000"	
5:	"42.19107000"	
6:	168145659999	
7:	"1303078.77019360"	
8:	818	
9:	"16.24802000"	
10:	"502133.01630380"	
11:	"0"	
▶ 1:	[...]	
▶ 2:	[...]	
▶ 3:	[...]	
▶ 4:	[...]	
▶ 5:	[...]	
▶ 6:	[...]	
▶ 7:	[...]	
▶ 8:	[...]	
▶ 9:	[...]	
▶ 10:	[...]	
▶ 11:	[...]	
▶ 12:	[...]	
▶ 13:	[...]	
▶ 14:	[...]	
▶ 15:	[...]	

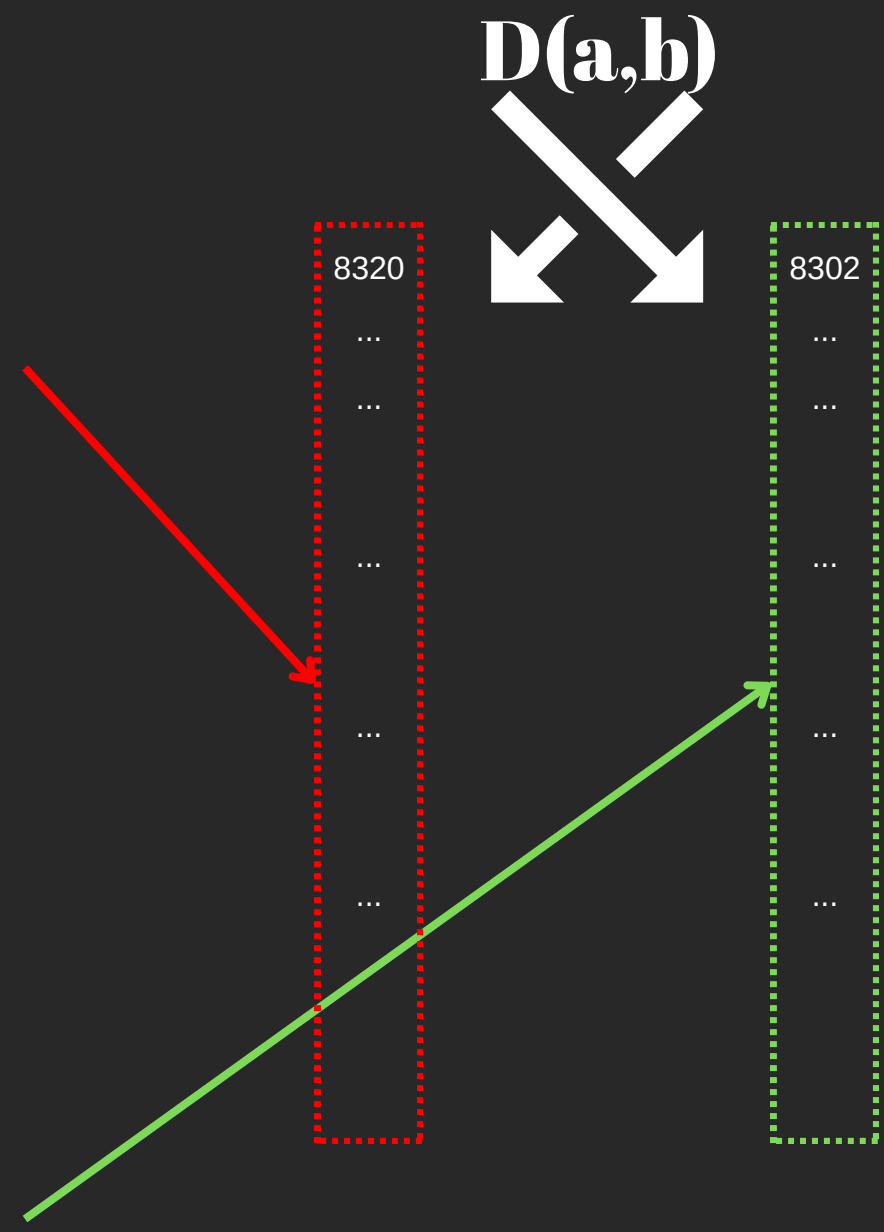


KNN



$$D(a,b) = \sqrt{\sum_{i=1}^n (b_i - a_i)^2}$$


Date	Time	Open	High	Low	Close	Volume	y
20150101	91500	8320	8325	8295.2	8304	129675	1
20150101	91600	8304	8307	8302.9	8307	72425	1
20150101	91700	8306.95	8312.3	8306.95	8312.3	42100	-1
20150101	91800	8312.5	8314.45	8308	8309	46075	...
20150101	91900	8308	8308	8302.25	8304	38675	...
20150101	92000	8305	8305	8300	8301	59550	...
20150101	92100	8300.15	8302.45	8298.4	8302.45	32025	...
20150101	92200	8302	8307.9	8302	8307.15	27425	...
20150101	92300	8307.45	8310.1	8307.45	8309.8	25900	...
20150101	92400	8309.8	8310.65	8309	8310	31800	...
20150101	92500	8309.5	8310	8308.1	8309.05	10075	...
20150101	92600	8309	8309.75	8307.35	8308.75	11825	...
20150101	92700	8308.8	8310.15	8308.05	8310.05	14925	...
20150101	92800	8310.6	8311.7	8309.4	8311	15825	...
20150101	92900	8311.45	8311.45	8309	8309	10000	...
20150101	93000	8309	8309	8307.45	8307.45	11375	...
20150101	93100	8307.45	8308.95	8307	8307.5	8525	...
20150101	93200	8307.15	8309.5	8307.05	8309.5	11000	...
20150101	93300	8309.05	8309.95	8308.55	8309	9125	...
20150101	93400	8309	8309.55	8305.15	8305.15	27050	...
20150101	93500	8305.15	8305.55	8301	8301.3	20850	...
20150101	93600	8302	8303.75	8302	8303	12175	?



4 modules

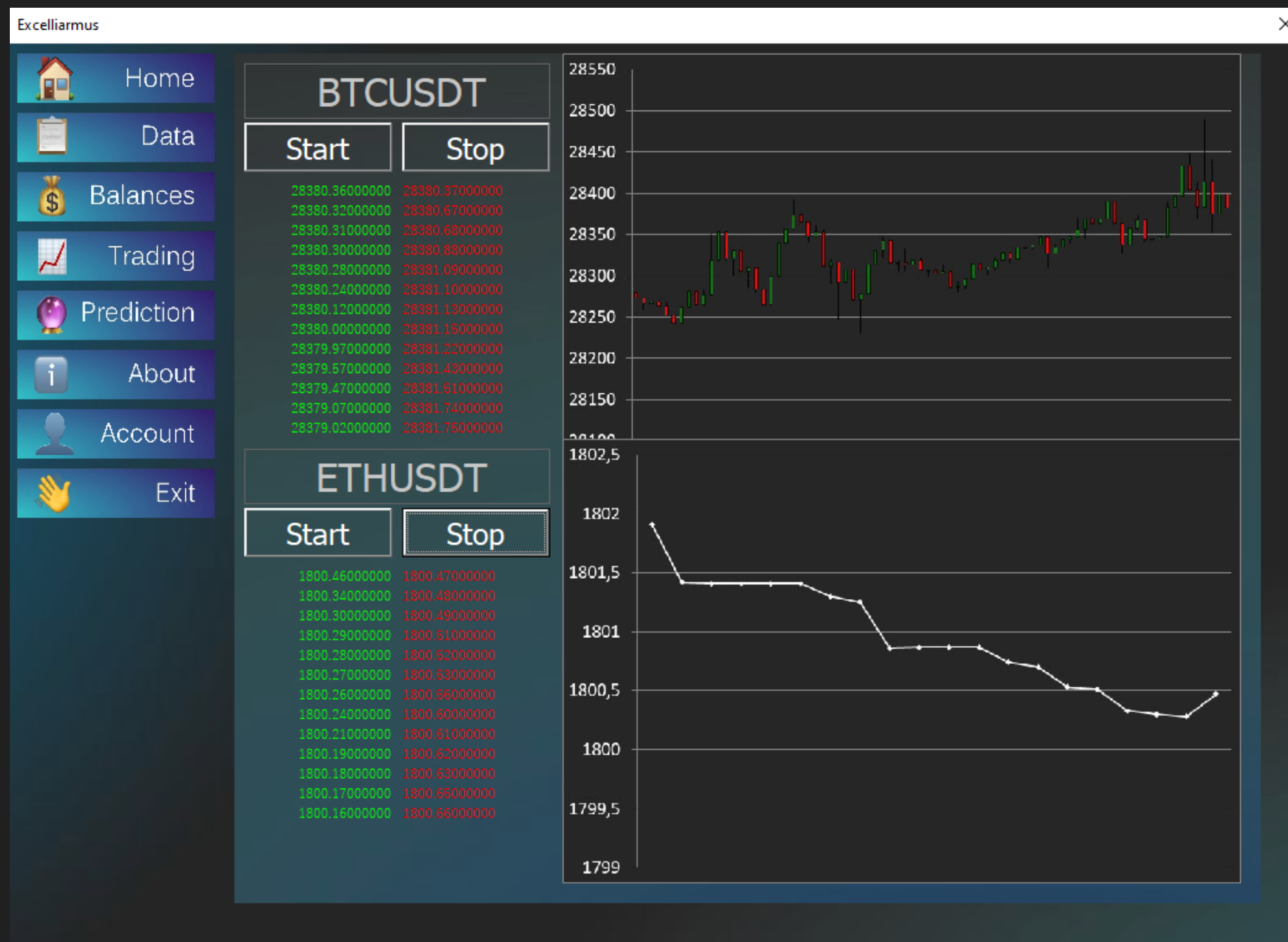
DATA

BALANCES

TRADING

PREDICTION

DATA



- Pour afficher les données en temps réel:
- Rentrer le nom de l'actif au format {BASE}{QUOTE}
 - Cliquer sur "Start"

Ce qui se passe:

- Le bid et le ask sont mis à jour en temps réel
- Les données historiques sont mises à jour en temps réel (graphiques en chandelles et en lignes)

Comment ça marche:

- Le bouton start enclenche une boucle infinie, dans laquelle on télécharge les données via une requête HTTP, puis on affiche les nouvelles données
- Le graphique du bas affiche les 30 derniers prix

BALANCES

Excelliarumus

Home

Data

Balances

Trading

Prediction

About

Account

Exit

PS4SLTYa1ygObExJMV2uq1F0lhJ1G2Oc

KnVZZttcy3aeggb353y52uPQt1OX3Fjge

Get balances

Start stream

Stop stream

Global Stream :
OFF

Overall balance

649618.80786553

\$

1000.000

BNB

326271.1

\$

Start stream

Stop stream

BNB Stream :
OFF

Update

1.037000

BTC

29414.007

\$

Start stream

Stop stream

BTC Stream :
OFF

Update

10000.0

BUSD

9873.187

\$

Start stream

Stop stream

BUSD Stream :
OFF

Update

100.0000

ETH

179303.72

\$

Start stream

Stop stream

ETH Stream:
OFF

Update

500.0000

LTC

40188.015

\$

Start stream

Stop stream

LTC Stream :
OFF

Update

500000.0

TRX

33527.1

\$

Start stream

Stop stream

TRX Stream :
OFF

Update

8861.93

USDT

8764.4516

\$

Start stream

Stop stream

USDT Stream :
OFF

Update

50000.00

XRP

22277.225

\$

Start stream

Stop stream

XRP Stream :
OFF

Update

Pour afficher les données :
Rentrer les clés API
Cliquer sur "Get balances"

- Ce qui se passe:
- Via un requête, on récupère les données relatives aux soldes, puis on calcul leurs équivalents en USD

- Comment ça marche:
- Le bouton "Get balances" envoie un requête avec les données des clés API et reçoit les données du compte
 - Le bouton "Start stream" fait la même chose mais dans une boucle infinie

TRADING

Excellarius

Home

Data

Balances

Trading

Prediction

About

Account

Exit

Market

Limit

Stop Loss

BTCUSDT

0.001

☒ BUY

☐ SELL

Price

100000

Random Trading Bot

Mean Reversion Trading Bot

Margin

0.5

Momentum Trading Bot

Open orders

All orders

Frequency :

2

Sec.

Start bot

Bot's status : **OFF**

Stop bot

Place order

Logs

15:58:06 : ▲ BUY 0.001 BTCUSDT @ 28598,1

15:58:10 : Do nothing 😊

15:58:14 : Do nothing 😊

15:58:17 : Do nothing 😊

15:58:20 : ▲ BUY 0.001 BTCUSDT @ 28607,68

15:58:25 : ▼ SELL 0.001 BTCUSDT @ 28607,16

15:58:29 : Do nothing 😊

15:58:33 : Do nothing 😊

15:58:37 : ▼ SELL 0.001 BTCUSDT @ 28600,03

15:58:41 : Do nothing 😊

15:58:45 : Do nothing 😊

15:58:49 : ▲ BUY 0.001 BTCUSDT @ 28619,91

- Pour acheter de la crypto :
- Rentrer les clés API dans "BALANCES"
- Choisir la crypto
- Choisir la quantité
- Choisir le "Order type" (market/limit/stop loss)
- Choisir le "Side" (buy/sell)
- Cliquer sur "Place order"

Ce qui se passe:

- Via un requête, on place un ordre et on vérifie qu'il s'est bien exécuté, puis on affiche les détails dans les logs

Trading bots:

- Random --> Boucle infinie --> Achat/Vente/Rien proba 1/3
- MR --> Achat si supérieur à SMA5 + margin
- Momentum --> Vente si supérieur à SMA5 + margin

PREDICTION

Excelliarumus

Home

Data

Balances

Trading

Prediction

About

Account

Exit

Trading pair

BTCUSDT

Quantity

0.0005

k parameter

3

Candlesticks

1000

Frequency

2

Margin

0.0001

K Nearest Neighbours

16:34:09

HOLD BUY BUY

Most common label

BUY

Start bot

Stop bot

Bot's status : **OFF**

Logs

16:32:57 : Do nothing 😊

16:33:01 : Do nothing 😊

16:33:06 : Do nothing 😊

16:33:17 : ▲ BUY 0.001 BTCUSDT @ 28690,85 (KNN)

16:33:23 : ▲ BUY 0.001 BTCUSDT @ 28700 (KNN)

16:33:28 : ▲ BUY 0.001 BTCUSDT @ 28701,02 (KNN)

16:33:35 : ▲ BUY 0.001 BTCUSDT @ 28701,04 (KNN)

16:33:47 : ▲ BUY 0.0005 BTCUSDT @ 28716,75 (KNN)

16:33:53 : ▲ BUY 0.0005 BTCUSDT @ 28711,46 (KNN)

16:33:59 : ▲ BUY 0.0005 BTCUSDT @ 28711,46 (KNN)

16:34:04 : ▲ BUY 0.0005 BTCUSDT @ 28711,46 (KNN)

16:34:10 : ▲ BUY 0.0005 BTCUSDT @ 28711,46 (KNN)

- Pour afficher prédire + trader en temps réel :
 - Rentrer les clés API dans "BALANCES"
 - Rentrer les infos requises
 - Cliquer sur "Start bot"
- Ce qui se passe:
 - Le module affiche en temps réel les K plus proches voisins, le voisin le plus commun et l'historique des trades, faits en fonction des prédictions
- Comment ça marche:
 - Via un requête, on télécharges les données, puis on discrimine les rendements avec la marge (cf doc)
 - Dans une boucle infinie on prédit le y de la dernière valeur avec un KNN

3 challenges techniques


AUTHENTIFICATION

REQUÊTES HTTP

TRADING BOTS

AUTHENTIFICATION

Excelliarmus

 **Excelliarmus**

E-mail

First name

Last name

Password

Confirm

Sign up

Already registered ?
[Log in](#)

Objectif:

Autoriser l'accès uniquement aux utilisateurs avec une adresse mail valide et confirmée

Solution:

Utilisation d'un service tiers (Supabase) pour gérer les aller-retour avec une base de données, via des requêtes HTTP

REQUÊTES HTTP

```
7 ' function takes a string to hash, a secret key to hash with HMAC SHA256 algorithm, and returns an hexadecimal string
8 ' inspired from internet
9 Public Function Hex_HMACSHA256(ByVal sTextToHash As String, ByVal sSharedSecretKey As String)
10     Dim asc As Object, enc As Object
11     Dim TextToHash() As Byte
12     Dim SharedSecretKey() As Byte
13     Set asc = CreateObject("System.Text.UTF8Encoding")
14     Set enc = CreateObject("System.Security.Cryptography.HMACSHA256")
15     TextToHash = asc.GetBytes_4(sTextToHash)
16     SharedSecretKey = asc.GetBytes_4(sSharedSecretKey)
17     enc.Key = SharedSecretKey
18     Dim bytes() As Byte
19     bytes = enc.ComputeHash_2((TextToHash))
20     ' tried to encode bytes directly to HEX but not working so have to do bytes > b64 > b16
21     Hex_HMACSHA256 = LCase(Base64To16(EncodeBase64(bytes)))
22     Set asc = Nothing
23     Set enc = Nothing
24 End Function
25
26 ' function takes an array of bytes and returns b64 encoded string
27 ' inspired from internet
28 Private Function EncodeBase64(ByRef arrData() As Byte) As String
29     'Inside the VBE, Go to Tools -> References, then Select Microsoft XML, v6.0
30     '(or whatever your latest is. This will give you access to the XML Object Library.)
31     Dim objXML As MSXML2.DOMDocument60
32     Dim objNode As MSXML2.IXMLDOMElement
33     Set objXML = New MSXML2.DOMDocument60
34     ' byte array to base64
35     Set objNode = objXML.createElement("b64")
36     objNode.DataType = "bin.base64"
37     objNode.nodeTypedValue = arrData
38     EncodeBase64 = objNode.Text
39     Set objNode = Nothing
40     Set objXML = Nothing
41 End Function
```

Problème:

Binance demande des requêtes signées avec l'algorithme de cryptage HMAC SHA256 à partir des clés API

Solution:

Recherche sur internet pour crypter en SHA256

Ensuite encrypter les bytes en B64

Ensuite encrypter la B64 en B16

Puis utiliser la B16 dans les requêtes

Convertir les bytes en B16 directement en marche pas, on ne sait pas pourquoi

TRADING BOTS

```
Do Until Not isMRBotOn
    If current_price < downLimit Then
        signal = placeMarketOrder(UserForm1.inputBalances1, UserForm1.inputBalances2, "BUY", ticker, qt)
        If signal = "success" Then
            addLog (get_time_for_logs & " : " & ChrW(9650) & " BUY " & qt & " " & ticker & " @ " & ModDate)
        Else
            addLog (signal)
        End If
    ElseIf current_price > upLimit Then
        signal = placeMarketOrder(UserForm1.inputBalances1, UserForm1.inputBalances2, "SELL", ticker, qt)
        If signal = "success" Then
            addLog (get_time_for_logs & " : " & ChrW(9660) & " SELL " & qt & " " & ticker & " @ " & ModDate)
        Else
            addLog (signal)
        End If
    Else
        addLog (get_time_for_logs & " : Do nothing " & ChrW(9787))
    End If
    ' *NOT MANDATORY* : updating the all the balances
    Call ModBalances.UpdateBalances(UserForm1.inputBalances1, UserForm1.inputBalances2)
    Application.Wait (Now + TimeValue("00:00:" & frequence))
    DoEvents
Loop
```

Objectif:

Avoir un programme qui tourne tout seul temps qu'il est allumé et qui trade selon certaines conditions et qui affiche l'historique des trades en temps réel

Solution:

Utiliser des booléens pour signifier l'état d'un bot, puis faire une boucle infinie tant que le booléen est vrai, et dans cette boucle placer les conditions de trade, vérifier que les trades se passent bien (auquel cas éteindre le bot) et afficher l'historique des trades en temps réel (et mettre à jour les soldes en temps réel)

DIFFICULTÉS

Gestion des virgules (format anglais vs français)

- Documentation Binance mal fournie
- Pas de multithreading
- Pas de librairie de *Machine Learning*
- Pas de Dataframes, Pandas, Numpy etc
- Requêtes HTTP impossibles sur MacOS
- Autres problèmes de compatibilité
- ...

