



ETH_USD

108022062 郭銓恩

CONTENTS

01

動機

02

目標

03

概述

04

實際操作

05

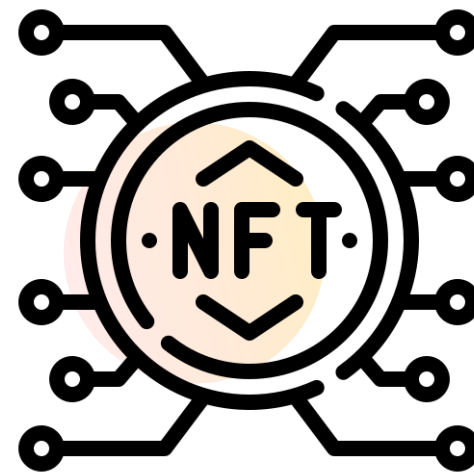
結語

動機
Motivation



NFT

- 最近元宇宙(NFT)的興起，是一種結合加密貨幣及XR的跨時代產品，加密貨幣的表現令人有目共睹，臉書母公司改名為Meta後，元宇宙相關加密貨幣一年漲幅百倍，尤其最有歷史的比特幣，十多年從默默無名到眾人皆知，漲幅早已近萬倍。



ETH

- 以太坊（Ethereum）是一種去中心化且開源有智慧型合約功能的公共區塊鏈平台。以太幣（Ξ）是以太坊的原生加密貨幣。截止 2021 年 12 月，以太幣是市值第二高的加密貨幣，且是被廣泛使用的區塊鏈。

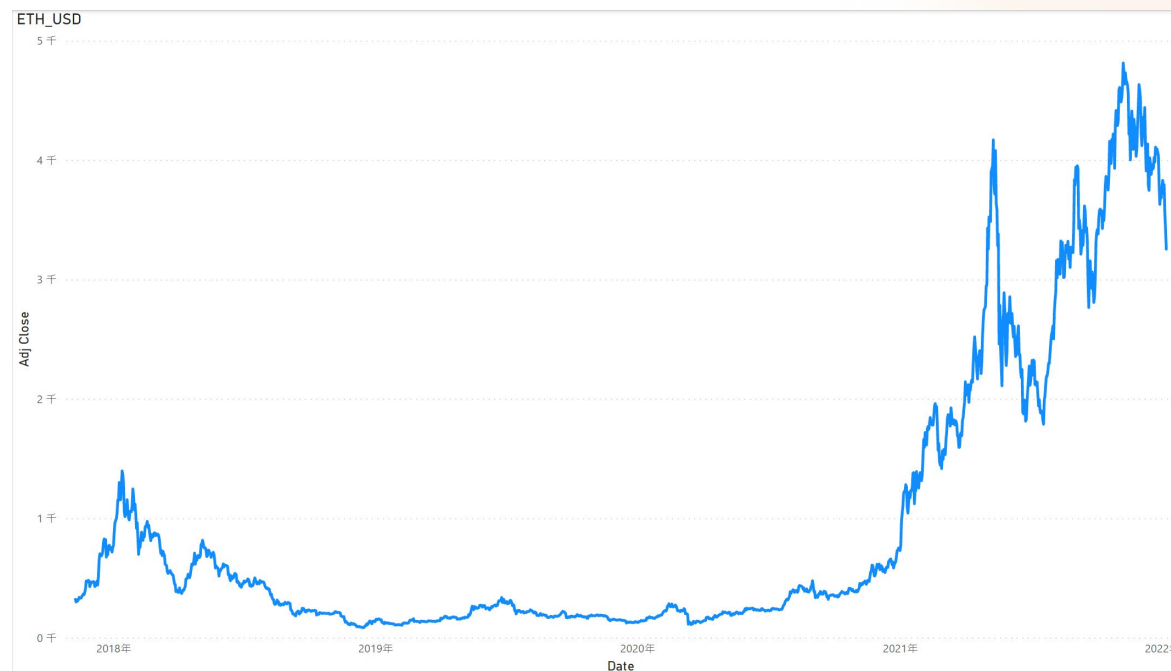


目標
Goal

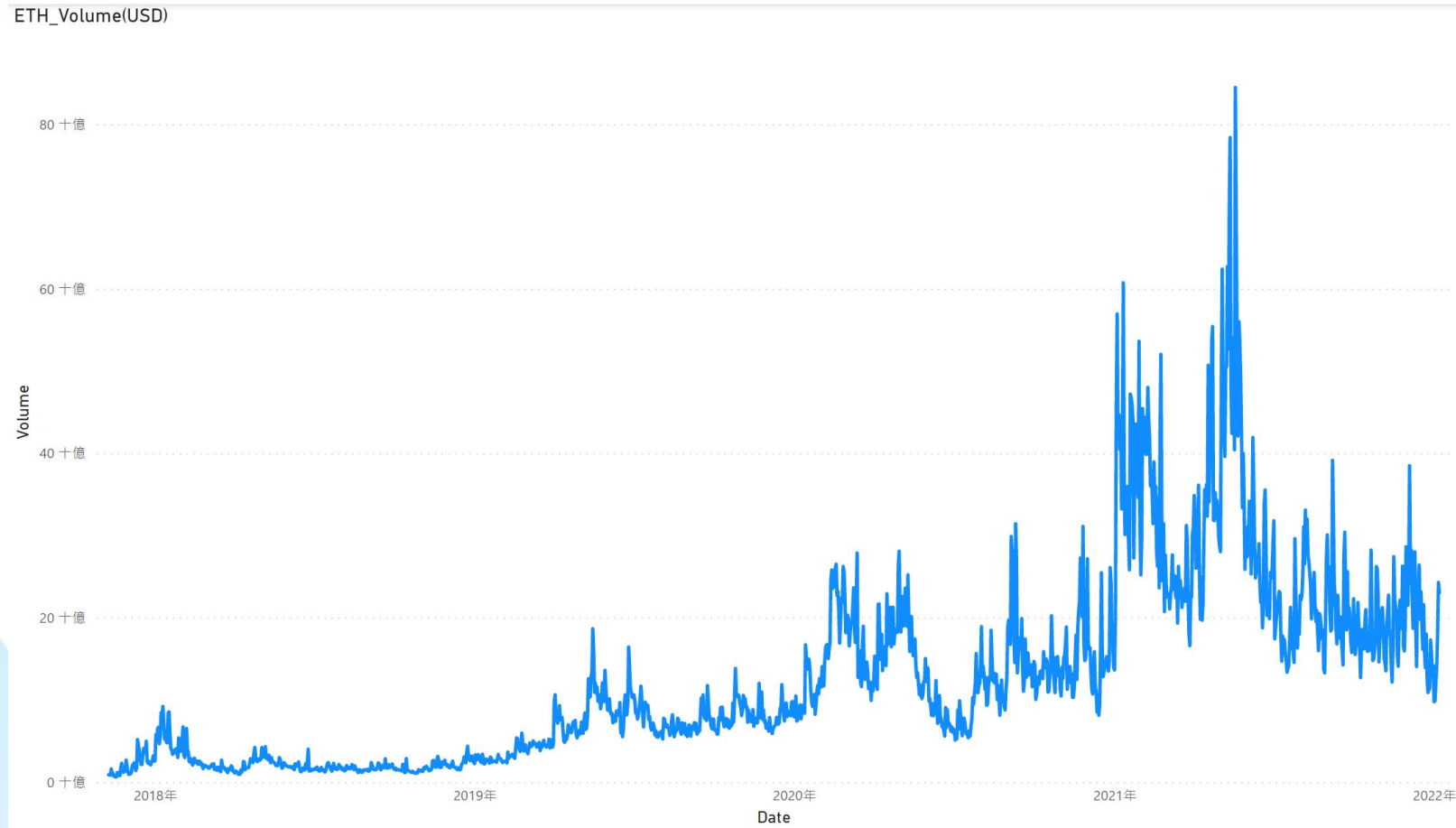


日期

- 希望透過歷史數據，用線性回歸推算未來加密貨幣的走勢。



ETH_Volume(USD)



概述 OverView



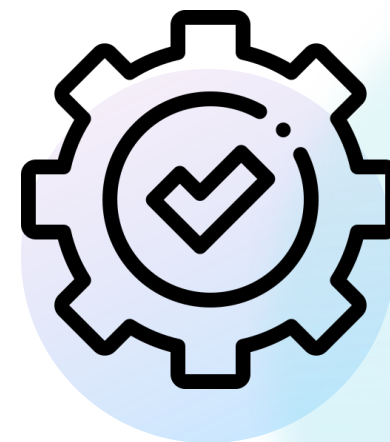
類神經網路

使用平台	Colaboratory PowerBI	 
程式語言	Python	
檔案來源	yfinance中的ETH-USD最大歷史數據(2017/11/09~2022/01/07) 下載為eth_historical_max.csv	
人工神經網路	線性回歸(LinearRegression) 循環神經網路(RNN)	

eth_historical_max.csv

Date	Timestamp	Open	High	Low	Close	Adj Close	Volume
日期	Unix時間 距離1970 年1月1日, 多少日 (86400秒)	當日開盤 價	當日最高 價	當日最低 價	當日收盤 價	當日經過 加權的收 盤價	當日交易 量

實際操作
Oprate



Yfinance下載歷史資料

```
[ ] import numpy as np
    from datetime import date
    import pandas as pd
    import matplotlib.pyplot as plt
    from sklearn.model_selection import train_test_split
    from sklearn.linear_model import LinearRegression
    from google.colab import drive
    from google.colab import files
```

```
[ ] !pip install yfinance
    !pip install yahoofinancials
    import yfinance as yf
    from yahoofinancials import YahooFinancials
```

!安裝yfinance

匯入yfinance命名為yf

```
[ ] eth_usd=yf.Ticker('ETH-USD')
    eth_usd.info
    eth_usd = yf.download('ETH-USD', period="max")
    eth_usd.to_csv('eth_historical_max.csv')
    files.download('eth_historical_max.csv')
```

.info查看'ETH-USD'的相關詳細資訊

.download下載成.csv

- 首先匯入必要套件，從Yahoo Finance 抓取ETH USD所有歷史(max)資料匯出成csv，將csv中的['Date']用excel新增Timestamp欄位。

Timestamp= (A2-DATE (1970,1,1)) *
86400

panda.csv

```
[ ] import os
drive.mount('/content/gdrive', force_remount=True)
os.chdir("/content/gdrive/MyDrive/亞洲大學/大三/資料科學")

Mounted at /content/gdrive

[ ] !ls -la #顯示該目錄中的所有檔案，用來確認雲端是否掛載成功

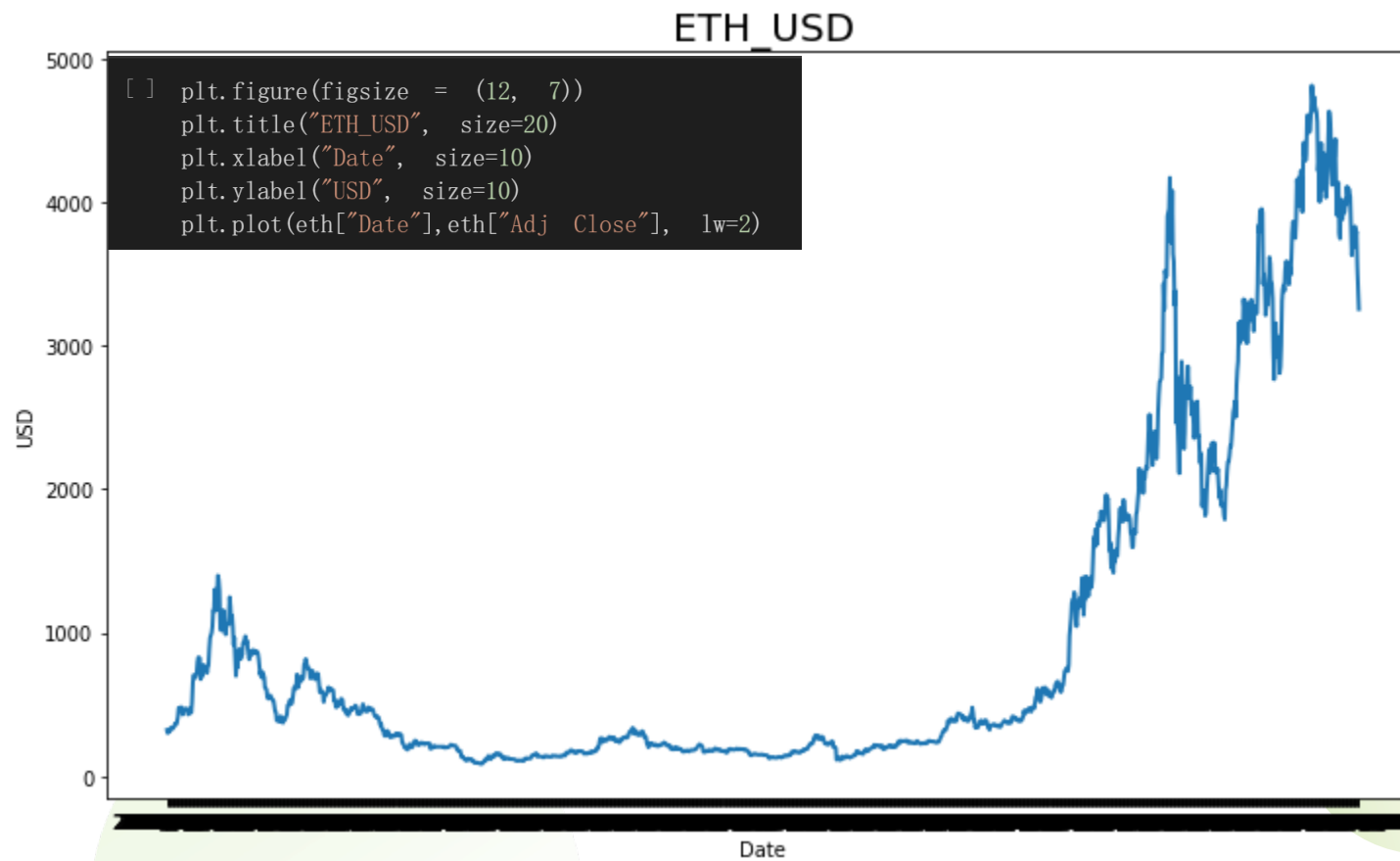
[ ] eth=pd.DataFrame()
eth = pd.read_csv('./eth_historical_max.csv')
eth
```

	Date	Timestamp	Open	High	Low	Close	Adj Close	Volume
0	2017/11/9	1510185600	308.644989	329.451996	307.056000	320.884003	320.884003	893249984
1	2017/11/10	1510272000	320.670990	324.717987	294.541992	299.252991	299.252991	885985984
2	2017/11/11	1510358400	298.585998	319.453003	298.191986	314.681000	314.681000	842300992
3	2017/11/12	1510444800	314.690002	319.153015	298.513000	307.907989	307.907989	1613479936
4	2017/11/13	1510531200	307.024994	328.415008	307.024994	316.716003	316.716003	1041889984
...
1516	2022/1/3	1641168000	3829.535645	3836.198730	3698.047607	3761.380371	3761.380371	12080777893
1517	2022/1/4	1641254400	3761.361572	3876.785156	3723.349854	3794.056641	3794.056641	14030925983
1518	2022/1/5	1641340800	3794.269043	3842.059814	3456.745361	3550.386963	3550.386963	18061338502
1519	2022/1/6	1641427200	3549.708984	3549.708984	3335.376709	3418.408203	3418.408203	24293791313
1520	2022/1/7	1641513600	3420.461914	3420.461914	3149.016602	3253.740723	3253.740723	23039903744

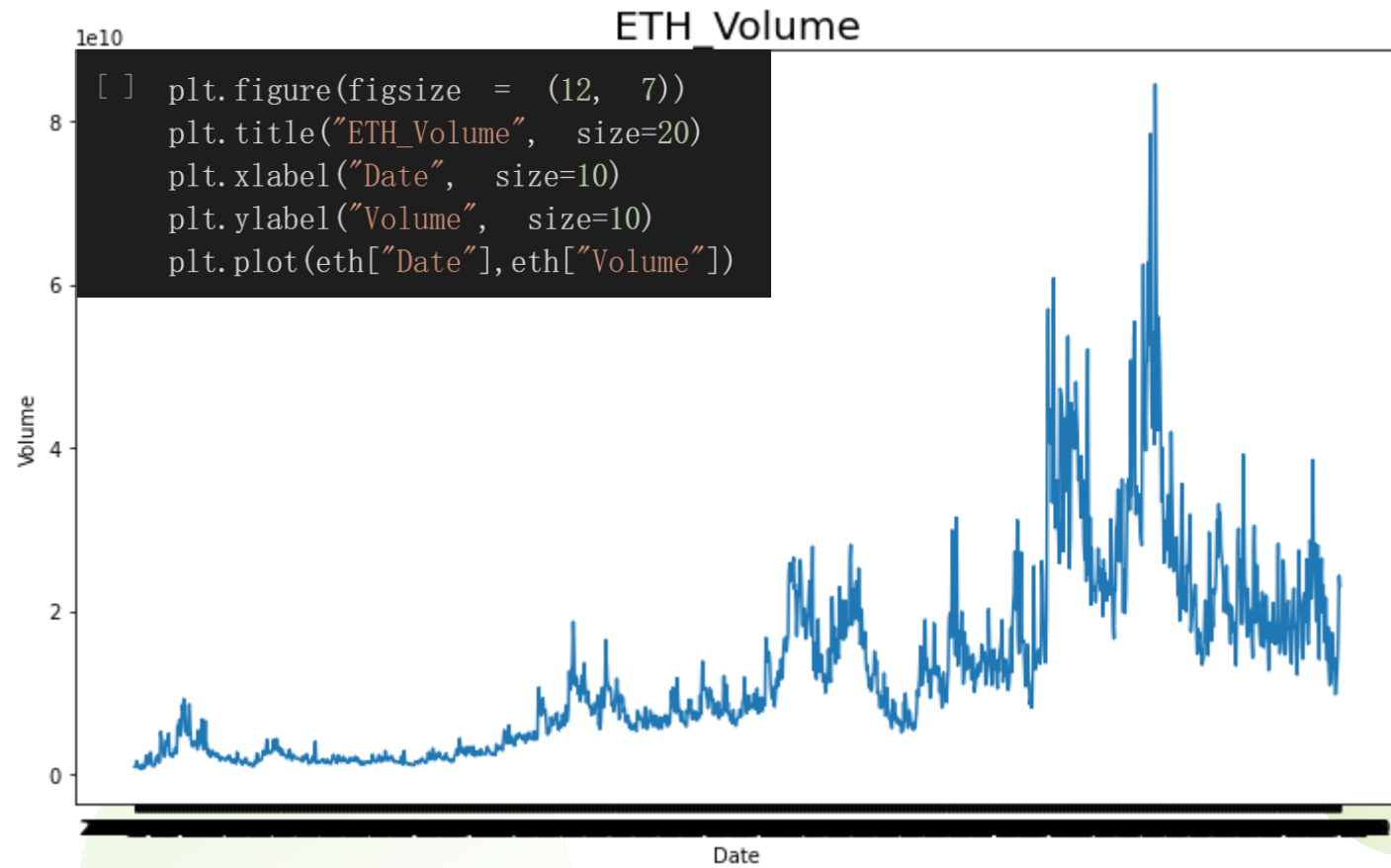
1521 rows × 8 columns

- 掛載Google Drive並匯入修改完的eth_historical_max.csv。

ETH_USD



ETH_Volume (USD)



過去(隨機訓練資料):過去(隨機測試資料):現在(預測未來)=7:3:3

```
[12] eth_a=eth[:int(np.round(len(eth)*0.7))] #過去
      eth_b=eth[len(eth_a):] #現在
      eth_a.shape

(1065, 8)

[13] eth_b.shape

(456, 8)

[14] required_features = ['Timestamp','Open', 'High', 'Low', 'Volume']
      # 'Timestamp','Open', 'High', 'Low', 'Volume'
      output_label = 'Adj Close'
```

特徵線性回歸分析

線性回歸output

```
[ ] x_train, x_test, y_train, y_test = train_test_split(
    eth_a[required_features],
    eth_a[output_label],
    test_size = 0.3
)
```

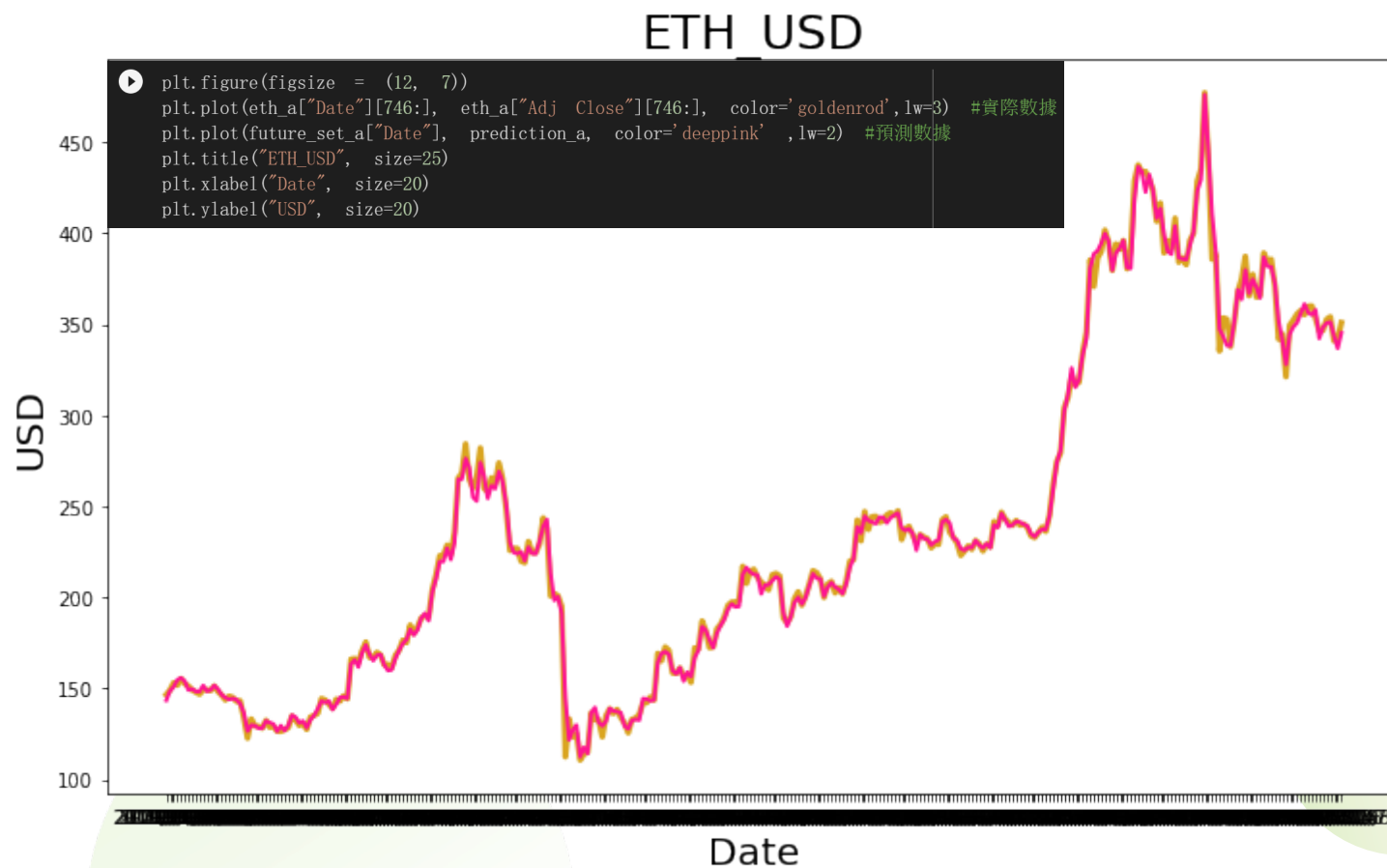
隨機切割訓練資料及測試資料, test_size=0.3為train 70%,test 30%

- eth_a取前70%數據
- eth_b取後30%數據
- 目前訓練的特徵資料為
['Timestamp' , 'Open' ,
 'High' , 'Low' ,
 'Volume '], 輸出['Adj
Close']
- 透過train_test_split隨機切割
70%(x_train,y_train),
30%(x_test,y_test)

LinearRegression預測結果

```
[18] model.score(x_test, y_test)
```

```
0.9983694299276583
```

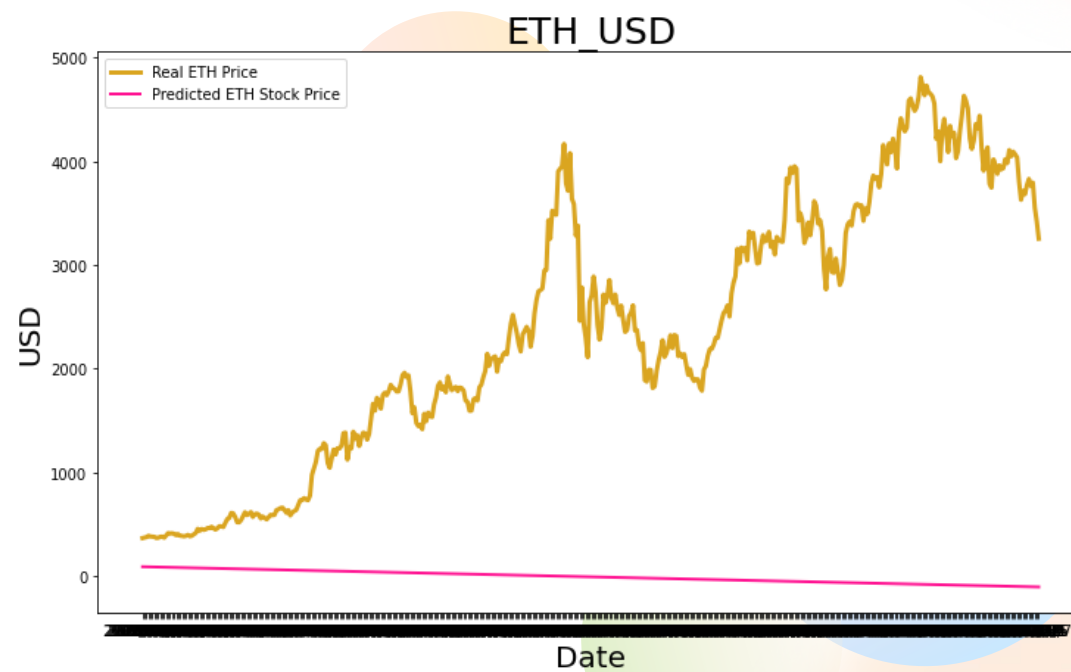


可以看見若是將所有數據帶入線性回歸預測，準確度高達99.8%。

僅透過['Timestamp']預測

```
[130] model=LinearRegression()  
      model.fit(x_train,y_train)  
      model.score(x_test, y_test)  
  
      0.32625267157452853
```

準確率不到50%，甚至在預測
eth_b(後30%)數據中出現了負
數的價格。



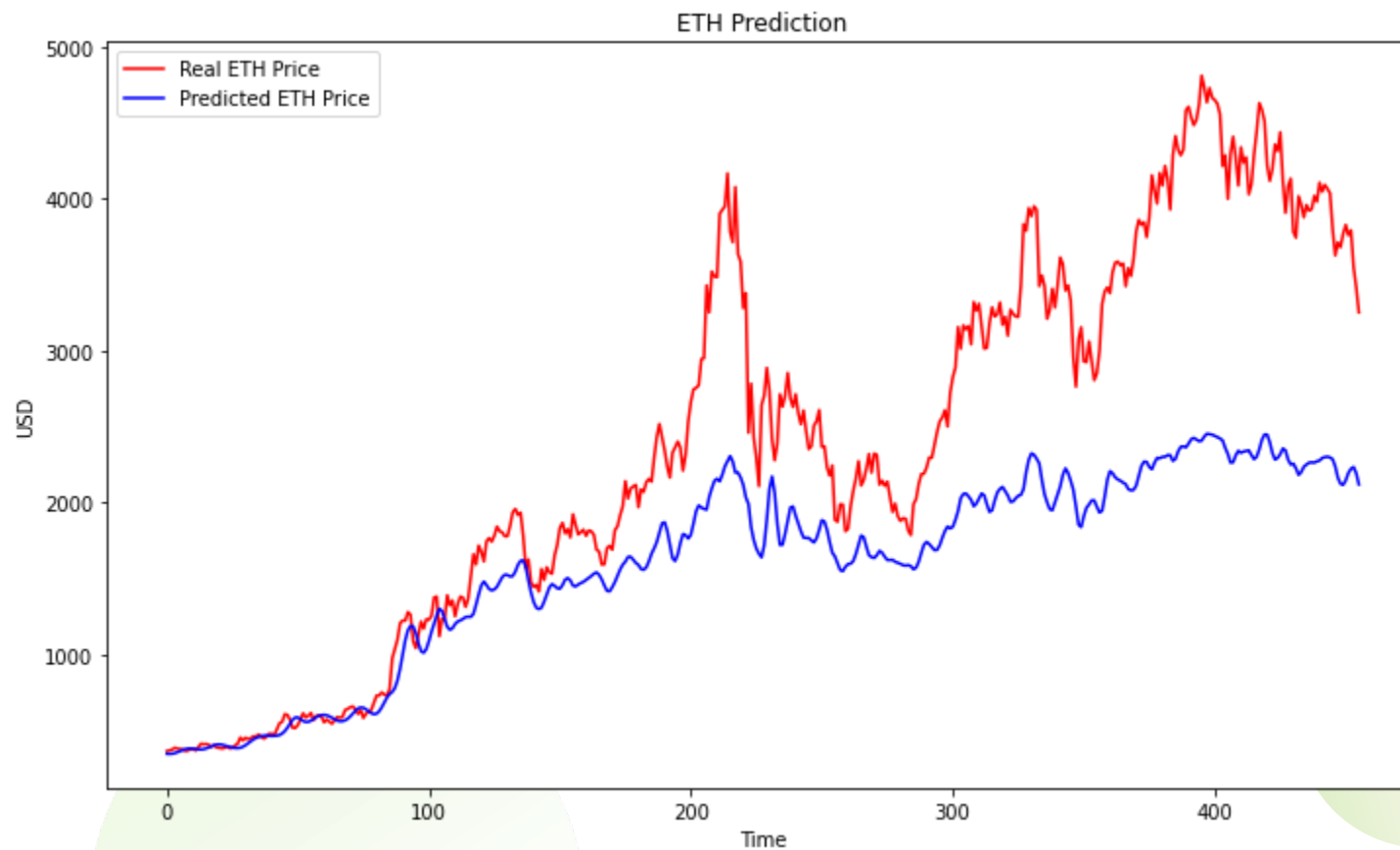
LinearRegression預測總結

- 接著嘗試['Date' , 'Volume']及單獨['Volume']進行預測，準確度都無法高於50%，除非透過['High'],['Low'],['Open']其中之一，準確度才能維持98%以上。
- 雖然線性回歸是個簡單易用的訓練模型，但若想要透過單一日期的數據做為預測極為容易失真。
- 所以線性回歸透過日期預測價格宣告**失敗**。

RNN預測

```
[17] network.compile(loss='mean_squared_error', optimizer='adam', metrics='accuracy')
network.fit(x_train, y_train, epochs=100, batch_size=128) #accuracy 訓練準確率

Epoch 100/100
8/8 [=====] - 2s 226ms/step - loss: 0.0012 - accuracy: 0.0020
<keras.callbacks.History at 0x7f34724fe350>
```



RNN預測準確率雖然僅0.2%，且預測數值多數偏低，但意外地透過以每60天的迴圈訓練，非常貼近實際走勢。

結語
Ending



結語

- 儘管實驗結果不盡人意，但也算在我的意料之內，如果透過簡單的類神經網路即能預測未來加密貨幣資料，就不可能會出現貧窮的問題，影響加密貨幣走勢更多成分應該是時事新聞。

