|  |
| --- |
| 投稿類別：資訊類 |
| 篇名：  Python網路爬蟲及數據分析實務－智能診股 |
| 作者：葉胤騋。私立二信高中。資三忠班。  鄭丞佑。私立二信高中。資三忠班。  李彥廷。私立二信高中。資三忠班。 |
| 指導老師：  刑文忠　老師 |

# 前言

1. 研究動機

隨著互聯網的發展，資訊已經成為了現代社會中不可或缺的一部分。而對於投資者而言，如何能夠在短時間內獲取大量的股票數據，並從中挑選出有價值的信息，已經成為了一個重要的問題。本論文旨在通過 Python 網路爬蟲及數據分析實務的研究，探討如何使用程式化方法進行股票數據的獲取和分析。在本論文中，我們選擇了以智能診股為研究對象，進行了一系列的研究和實驗。智能診股是一個可以自動化執行股票選股的系統，可以大大節省投資者的時間和精力。在本論文中，我們嘗試使用 Python 網路爬蟲和數據分析的技術，對股票市場中的數據進行收集和分析，並且建立了一個基於機器學習算法的股票選股模型。

1. 研究目的  
   1. 提供投資者投資的決策參考
   2. 縮短彙整大量資料的時間
   3. 了解如何多重面分析股票
2. 研究方法  
   1. 研究步驟  
      1. 透過Chat GPT 提供建議
      2. 網路查找相關資料
      3. 利用Python 3.11，Replit Online IDE 撰寫程式
      4. （預計）利用Visual Studio Community轉換為人性化視窗
   2. 研究流程甘特圖

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 週次  項目 | 2/17 | 2/24 | 3/3 | 3/10 | 3/17 | 3/24 | 3/31 |  |  |  |  |  |  |  | 負責組員 |
| 主題  擬定 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 全體 |
| 前言  製作 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 全體 |
| 文獻  探討 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 全體 |
| 研究  方法 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 研究分析與結果 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 研究結論與建議 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 參考  文獻 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# 文獻探討

1. Python 程式語言  
     
   Python是一種易於學習，功能強大的程式設計語言。它具有高效的高級數據結構和簡單而有效的面向物件程式設計方法。Python 優雅的語法和動態類型，加上其解釋性，使其成為大多數平臺上許多領域腳本編寫和快速應用程式開發的理想語言。
2. Python網路爬蟲  
     
   網路爬蟲是一種用於在全球資訊網中進行網頁索引的網路機器人。所有類型的搜索引擎都使用網路爬蟲來提供有效的結果。實際上，它收集來自其他網站的所有或某些特定超連結和HTML內容，並以適當的方式預覽它們。當有大量鏈接需要爬取時，即使是最大的爬蟲也會失敗。因此，2000年代初的搜索引擎在提供相關結果方面表現不佳，但現在這個過程已經得到了很大改善，並且立即提供了適當的結果。  
     
   這裡的網路爬蟲是用python3創建的。Python是一種高級編程語言，包括面向對象、命令式、函數式編程和大型標準庫。對於網路爬蟲，使用了兩個標準庫-requests和BeautfulSoup4。requests提供了一種簡單的方法來連接到全球資訊網，而BeautfulSoup4則用於某些特定的字符串操作。
3. Python 數據分析  
     
   數據分析是收集，轉換和組織數據以進行未來預測並做出明智的數據驅動決策的技術。它還有助於找到商業問題的可能解決方案。Python有許多套件可以進行各種數據分析，如使用NumPy分析數值數據、使用Pandas分析表格數據、使用Matplotlib進行數據可視化和探索性數據分析。
4. 分析股票  
     
   股票分析主要分成三種方法，分別是基本面分析、技術面分析、消息面分析。  
   1. 基本面分析：分析公司的基本營運和發展，了解公司的內在價值，主要包括財務分析、管理層分析、行業分析。
   2. 技術面分析：研究過去的股價走勢和股票成交量，預測未來的股價走勢。
   3. 消息面分析：分析與公司有關的消息，包括公司的發展、行業的成長、國家的政策等等，判斷股價會因為這些新聞上升或下跌。
5. 分析公司財務報表  
     
   一般來說，公司的年度報表、營業報表都指的就是指財務報表。一套完整的財務報表包括資產負債表、綜合損益表、現金流量表、權益變動表(或股東權益變動表)和財務報表附註。財務數據分析中通常以資產負債表、綜合損益表、現金流量表三張表為最基本也是最重要的報表，稱為三大財務報表。  
   1. 資產負債表：主要是告訴投資者在出具報表的時刻公司資產負債情況如何，所以，報表的關鍵是時間，時間對這張報表的影響很大。最重要的一個“勾稽關係”就是資產等於負債加上權益。就是現在擁有的一切，不外乎來源於兩個方面，一個是本來就是自己的，另一個就是借來的，自己有的，再加上借來的，當然就是現在擁有的一切。在會計上，目前我擁有的一切，就叫資產，而借來的錢，就是負債，自己的，就叫權益。這就是資產負債表最重要的內部“勾稽關係”。
   2. 綜合損益表：主要是告訴投資在一段時間裡，公司的損益情況，損益表關鍵一點，就是看這段時間有多長，一般是一個月，一個季度或一年的時間。在損益表裡，最重要的一個【勾稽關係】就是收入減去成本費用，等於利潤。
   3. 現金流量表：主要是告訴投資者，在一段時間裡，這個公司收進了多少現金，付出去了多少現金，還餘下多少現金在銀行裡。這張報表的關鍵也是要看這段時間有多長了，這一點同利益表或損益表一樣。在這張表裡，最重要的一個【勾稽關係】就是流入的現金減去流出的現金，等於剩下的現金。
   4. 財務報表分析指標  
      1. 流動比率：  
           
         流動比率=流動資產/流動負債  
           
         一般認為，流動比率若達到2倍時，是最令人滿意的。若流動比率過低，企業可能面臨著到期償還債務的困難。若流動比率過高，這又意味著企業持有較多的不能營利的閒置流動資產。使用這一指標評價企業流動指標時，應同時結合企業的具體情況。
      2. 速動比率：速動比率又稱酸性實驗比例，是指速動資產占流動負債的比率，它反映企業短期內可變現資產償還短期內到期債務的能力。速動比率是對流動比率的補充。計算公式如下：  
           
         速動比例=速動資產/流動負債  
           
         速動資產是企業在短期內可變現的資產，等於流動資產減去流動速度較慢的存貨的餘額，包括貨幣資金、短期投資和應收帳款等。一般認為速動比率1：1是合理的，速動比率若大於1，企業短期償債能力強，但獲利能力將下降。速動比率若小於1，企業將需要依賴出售存貨或舉借新債來償還到期債務。
      3. 現金比率：現金比率是企業現金同流動負債的比率。這裡說的現金，包括現金和現金等價物。這項比率可顯示企業立即償還到期債務的能力，其計算公式為：  
           
         現金比率=現金/流動負債  
         資產負債率：資產負債率，亦稱負債比率、舉債經營比率，是指負債總額對全部資產總額之比，用來衡量企業利用債權人提供資金進行經營活動的能力，反映債權人發放貸款的安全程度。計算公式為：  
           
         資產負債率=（負債總額/資產總額）X100%  
           
         一般認為，資產負債率應保持在50%左右，這說明企業有較好的償債能力，又充分利用了負債經營能力。
6. C++ 使用者介面(保留)
7. Open AI API  
     
   OpenAI API 可以應用於幾乎任何涉及理解或生成自然語言或代碼的任務。我們提供了一系列不同功率級別的模型，適用於不同的任務，並且能夠微調您自己的定制模型。這些模型可用於從內容生成到語義搜索和分類等所有事物。

# 研究方法

## JSON

JSON代表JavaScript物件表示法。它是一種輕量級的資料交換格式，易於閱讀和撰寫。JSON檔案通常用於在網路應用程式中傳輸資料。

如何使用JSON檔案？

使用JSON檔案存取資料非常簡單。以下是一個範例JSON檔案：

{

"name": "John",

"age": 30,

"city": "New York"

}

您可以使用各種編程語言來讀取和寫入JSON檔案。例如，在Python中，您可以使用以下程式碼來讀取JSON檔案：

import json

# 讀取JSON檔案

with open('example.json', 'r') as f:

data = json.load(f)

# 存取資料

print(data['name'])

print(data['age'])

print(data['city'])

在這個例子中，我們使用了Python的json庫來讀取JSON檔案並存取其資料。

JSON的優點

JSON有許多優點，包括：

易於讀寫和編寫

可以被各種不同的編程語言支援和解析

支援複雜的資料結構，如陣列和巢狀物件

支援標準的Unicode編碼，可以處理多國語言的資料

因為這些優點，JSON已成為在網路應用程式中廣泛使用的資料交換格式。

## 建立 Python 模組

要創建模組儲存函式集，可以使用Python的模組機制。

以下是一個範例程式碼：

# 在 my\_module.py 中定義您的函式集

def add(x, y):

return x + y

def subtract(x, y):

return x - y

# 在其他程式中使用您的函式集

import my\_module

print(my\_module.add(1, 2)) # 輸出 3

print(my\_module.subtract(5, 3)) # 輸出 2

在這個例子中，您可以在 my\_module.py 檔案中定義您的函式集。然後，您可以在其他 Python 程式中使用 import 陳述式來載入您的函式集，並使用它們來完成您的任務。

請注意，您需要確保您的模組檔案位於 Python 模組搜尋路徑中。您可以將模組檔案放在與您的 Python 程式相同的目錄中，或將其放在 Python 的標準模組路徑中。您也可以將模組檔案的路徑添加到 PYTHONPATH 環境變數中，以將其添加到模組搜尋路徑中。

希望這可以幫助您創建您自己的 Python 模組儲存函式集。

## 使用 Beautiful Soup進行財務資訊的爬蟲

Beautiful Soup是一個Python的函式庫，用於從HTML和XML檔案中提取資料。您可以使用它來解析網頁並提取所需的資訊。以下是一個範例程式碼，演示如何使用Beautiful Soup來提取財務資訊。

import requests from bs4 import BeautifulSoup

# 設置要爬取的網址

url = '<https://finance.yahoo.com/quote/TSLA/>'

# 發送請求並獲取響應

response = requests.get(url)

# 使用Beautiful Soup解析頁面

soup = BeautifulSoup(response.text, 'html.parser')

# 找到我們需要的元素

price = soup.find('span', {'class': 'Trsdu(0.3s) Fw(b) Fz(36px) Mb(-4px) D(ib)'})

在這個例子中，我們設置了要爬取的網址，然後使用 requests 函式庫發送請求，獲取網頁響應。接下來，我們使用 BeautifulSoup 函式庫解析頁面。在這裡，我們找到了我們需要的元素，並使用 price.text 輸出所需的資訊。

## 利用 twstock 取得即時股價

以下是使用 twstock 模組來取得股市即時價格的範例程式碼：

import twstock

# 設置要查詢的股票代號

stock\_code = '2330'

# 取得股票即時資訊

stock = twstock.realtime.get(stock\_code)

# 輸出所需的資訊

print('股票名稱：', stock['info']['name'])

print('股票代號：', stock['info']['code'])

print('即時股價：', stock['realtime']['latest\_trade\_price'])

在這個例子中，我們使用了 twstock 模組來取得股票即時資訊。我們設置了要查詢的股票代號，然後使用 twstock.realtime.get() 方法取得股票的即時資訊。在這裡，我們使用 print() 函式輸出所需的資訊，包括股票名稱、股票代號和即時股價。請注意，您需要確保已安裝 twstock 模組才能執行此程式碼。

# 研究分析與結果

class log:

def \_\_init\_\_(self):

self.time = ctime()

self.fd = open("log.txt", "a", encoding="utf-8")

def write(self, msg):

print(msg)

self.fd.write(f"[PRINT] [{self.time}] {msg}\n")

def error(self, msg):

self.write("Something went wrong,please check the log file.")

self.fd.write(f"[ERROR] [{self.time}] {msg}\n")

def info(self, msg):

self.fd.write(f"[INFO] [{self.time}] {msg}\n")

def debug(self, msg):

self.fd.write(f"[DEBUG] [{self.time}] {msg}\n")

def warning(self, msg):

self.fd.write(f"[WARNING] [{self.time}] {msg}\n")

def \_\_del\_\_(self):

self.fd.close()

global logger

logger=log()

def load\_data\_config():

logger=log()

"""載入初值設定

Returns:

dict: 初值設定，以字典形式儲存

"""

logger.info("Loading Config.json...")

fd = open("Config.json", 'r', encoding="utf-8")

data = fd.read()

setting = json.loads(data)

fd.close()

logger.info("Config.json loaded successfully.")

return setting

def get\_web\_content(URL):

"""

擷取網頁資料

Args:

URL (str): 輸入網址

Returns:

Response: 回傳網頁資料

"""

Content = requests.get(URL)

return Content.text

def get\_stockweb\_info(code): # get stock info on tw.stock.yahoo

"""

擷取tw.stock.yahoo資料

Args:

code (str): 輸入股票代碼，作為網址的一部份

Returns:

str: 回傳該頁面資料

"""

stockweb = get\_web\_content(f'https://tw.stock.yahoo.com/quote/{code}')

return stockweb

def htmlize(web): # Turn content into html format

"""

格式化HTML

Args:

web (str): 網頁原始碼

Returns:

str: 格式化後的網頁原始碼

"""

return BeautifulSoup(web, 'html.parser')

def get\_elm(web, elm): # find element from html

return htmlize(web).find(elm)

def get\_realtime\_price(id): # get realtime info by entering stock code

return twstock.realtime.get(id)

def randstock(): # generate random stock code

code = random.randint(1011, 9999)

code = str(code)

return code

def getstockinfo(code):

stock = yf.Ticker(code)

info = stock.info

return (info)

def update(code:str):

try:

data = getstockinfo(f"{code}.tw")

except requests.exceptions.HTTPError as e:

print(e)

return None

S = json.dumps(data)

with open(f"StockLibrary/{code}.json", "w",encoding="utf-8") as fd:

fd.write(S)

print(f"{code}.json Updated.")

def importinfo():

N = {}

L = glob.glob(os.path.join("StockLibrary", "\*"))

for i in L:

code = i[13:17]

update(code)

try:

with open(i, 'r', encoding='utf-8') as f:

S = f.read()

# print(S)

C = json.loads(S)

D = C["longName"]

# print(type(D))

N[i] = D

G = json.dumps(N)

with open("StockLibrary/Name.json", "w") as k:

k.write(G)

except KeyError:

continue

# 研究結論與建議

# 參考文獻

The Python Tutorial：  
<https://docs.python.org/3/tutorial/index.html>   
  
Build a simple python web crawler：  
<https://dev.to/pranay749254/build-a-simple-python-web-crawler#:~:text=%EE%80%80Python%20Web%20Crawler%EE%80%81%20The%20%EE%80%80web%20crawler%EE%80%81%20here%20is,two%20standard%20library%20are%20used%20-%20requests%20.>   
  
Data Analysis with Python：  
  
<https://www.geeksforgeeks.org/data-analysis-with-python/>   
  
[股票分析方法]基本面、技術面、消息面｜投資新手必讀：  
  
<https://gofire.today/stock-analysis/>   
  
10分鐘看懂財務報表，只需掌握一個公式！：  
  
<https://www.finereport.com/tw/knowledge/finereport/caiwu-baobiao.html>