SEMESTER PROJECT

AUTOMATING REAL-TIME INVESTMENT DATA COLLECTION: STOCKS

DSCI 505 U15: Business Analytics Fundamentals

02

Table of Contents

Presentation Outline

Stock market investments are subject to fluctuations in the financial markets, both up and down. Past performance is no guarantee of future performance. Therefore, real-time monitoring is of utmost importance to reduce risks and increase accuracy, therefore leading to an all-over higher efficiency.

Introduction

Problem

Time Consuming

Poor data collection and quality

Not Flexible or Scalable

Error-Prone

Solution

05

Superior performance

Flexible and easy-to-use data

Scalable

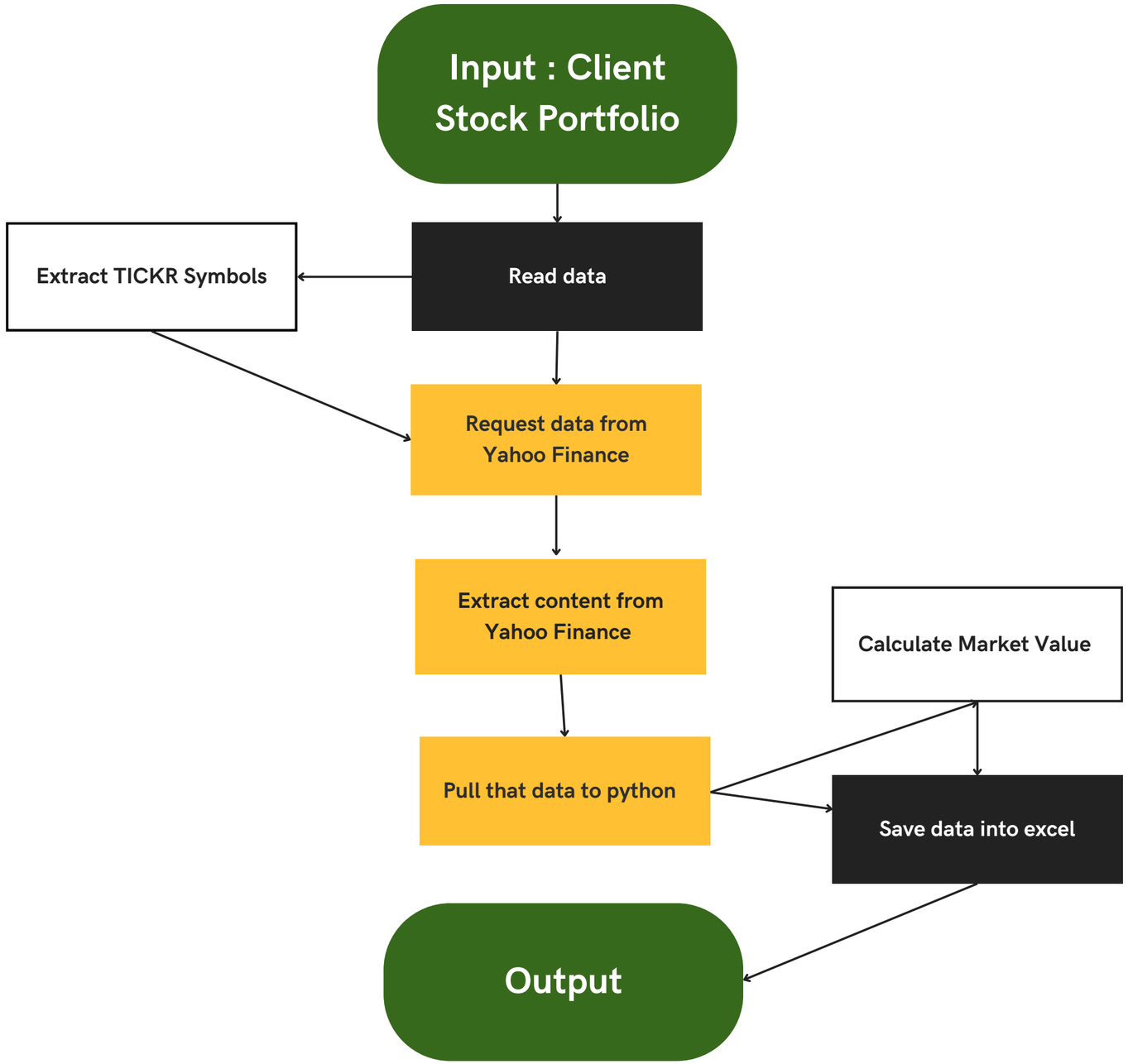
Efficiency

Faster Decision Making

Enhanced data-collection

Flowchart

06



Data Set: Sample Portfolio

Date	Company_Name	Tickr	Number_of_Shares	Purchase_Price	Book_Value
11/01/2022	Apple Inc.	AAPL	7	$ 150.65	$ 1,054.55
11/01/2022	Alphabet Inc.	GOOG	12	$ 90.50	$ 1,086.00
11/01/2022	Microsoft Corporation	MSFT	4	$ 228.17	$ 912.68
11/01/2022	The Walt Disney Company	DIS	10	$ 105.98	$ 1,059.80
11/01/2022	Tesla, Inc.	TSLA	4	$ 227.82	$ 911.28

Reading Data

CODE

```
import pandas as pd

df = pd.read_excel( "/Users/theju/Desktop/USD/Business Analytics Fund./Portfolio.xlsx")
```


Index	Date	Company Name	Tickr	Number of Shares	Purchase Price	Book Value
0	2022-01-11 00:00:00	Apple Inc.	AAPL	7	150.65	1054.55
1	2022-01-11 00:00:00	Alphabet Inc.	GOOG	12	90.5	1086
2	2022-01-11 00:00:00	Microsoft Corporation	MSFT	4	228.17	912.68
3	2022-01-11 00:00:00	The Walt Disney Company	DIS	10	105.98	1059.8
4	2022-01-11 00:00:00	Tesla, Inc.	TSLA	4	227.82	911.28

OUTPUT

09

CODE

```
import requests
from bs4 import BeautifulSoup
from datetime import date
import schedule
import time

mystocks = list(df.Tickr)
stockdata = []

def getData (symbol) :

    headers= {'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36'}
    url= f'https://finance.yahoo.com/quote/{symbol}'
    r = requests.get(url, headers=headers)
    soup = BeautifulSoup(r.text, 'html.parser')
    t = time.localtime()
    stock = {
    'Current Date' : date.today(),
    'Current Time' : time.strftime("%H:%M:%S", t) ,
    'Tickr': symbol,
    'Closing_Price': soup.find('div', {'class': 'D(ib) Mend(20px)'}) .find_all('fin-streamer')[0].text,
    'Change': soup.find('div', {'class': 'D(ib) Mend(20px)'}).find_all('fin-streamer')[1].text,
    'Difference': soup.find('div', {'class': 'D(ib) Mend(20px)'}).find_all('fin-streamer')[2].text,
    }
    return stock

for tickr in mystocks:
    stockdata.append(getData(tickr))
    print ("Getting: ", tickr)

print(stockdata)

schedule.every().day.at("16:00").do(getData)
```

OUTPUT

```
Getting:  AAPL
Getting:  GOOG
Getting:  MSFT
Getting:  DIS
Getting:  TSLA
[{'Current Date': datetime.date(2022, 12, 4), 'Current Time': '00:36:14', 'Tickr':
'AAPL', 'Closing_Price': '147.81', 'Change': '-0.50', 'Difference': '(-0.34%)'},
{'Current Date': datetime.date(2022, 12, 4), 'Current Time': '00:36:15', 'Tickr':
'GOOG', 'Closing_Price': '100.83', 'Change': '-0.45', 'Difference': '(-0.44%)'},
{'Current Date': datetime.date(2022, 12, 4), 'Current Time': '00:36:16', 'Tickr':
'MSFT', 'Closing_Price': '255.02', 'Change': '+0.33', 'Difference': '(+0.13%)'},
{'Current Date': datetime.date(2022, 12, 4), 'Current Time': '00:36:18', 'Tickr':
'DIS', 'Closing_Price': '99.43', 'Change': '+0.84', 'Difference': '(+0.85%)'},
{'Current Date': datetime.date(2022, 12, 4), 'Current Time': '00:36:19', 'Tickr':
'TSLA', 'Closing_Price': '194.86', 'Change': '+0.16', 'Difference': '(+0.08%)'}]
```

Web Scraping

...

Exporting Final Output to Excel

CODE

```
dfoutput = pd.DataFrame (stockdata, columns = ['Current Date', 'Current Time', 'Tickr', 'Closing_Price', 'Change', 'Difference'])
df_join = pd.merge(df, dfoutput, how = 'inner', on='Tickr')
df_join['Closing_Price'] = pd.to_numeric(df_join['Closing_Price'], errors='coerce')
df_join['Change'] = pd.to_numeric(df_join['Change'], errors='coerce')

df_join ['Market Value']= df_join ['Number_of_Shares']* df_join ['Closing_Price']

df_join.to_excel("/Users/theju/Desktop/USD/Business Analytics Fund./Output.xlsx", sheet_name = "Stock Prices")
```

Calculating Current Market Value and exporting the final output to excel file named "Output" with Sheet name "Stock Prices"

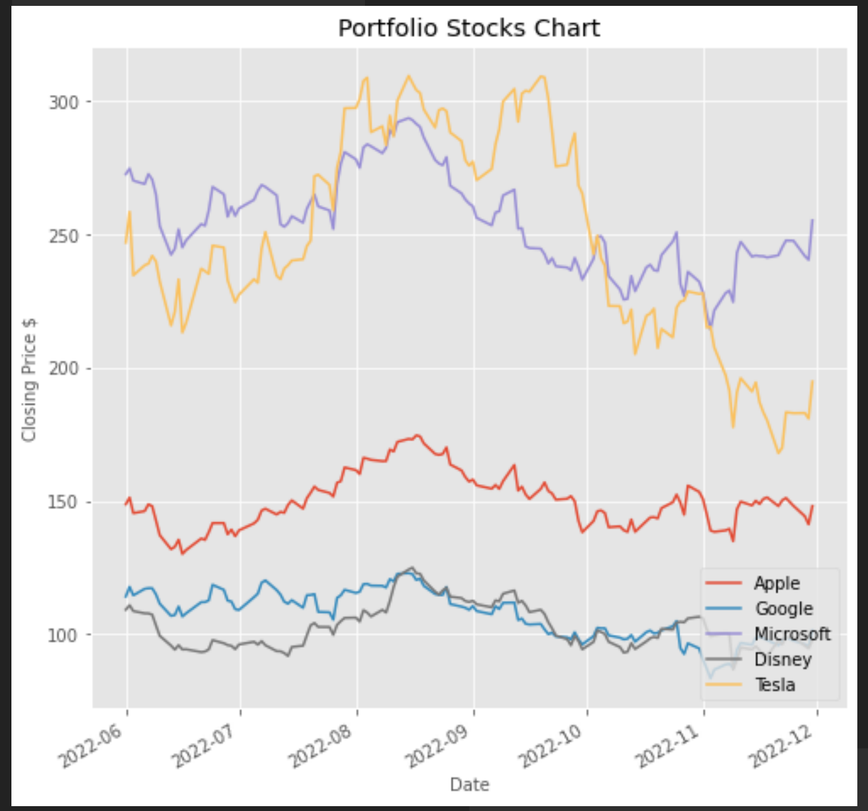
Final Output

Date	Company Name	Tickr	No of Shares	Purchase Price	Book_Value	Current Date	Current Time	Closing Price	Change	Difference	Market Value
2022-01-11	Apple Inc.	AAPL	7	150.65	1054.55	2022-12-06	16:29:46	142.91	-3.72	(-2.54%)	1000.37
2022-01-11	Alphabet Inc.	GOOG	12	90.5	1086	2022-12-06	16:29:47	97.31	-2.56	(-2.56%)	1167.72
2022-01-11	Microsoft Corporation	MSFT	4	228.17	912.68	2022-12-06	16:29:48	245.12	-5.08	(-2.03%)	980.48
2022-01-11	The Walt Disney Company	DIS	10	105.98	1059.8	2022-12-06	16:29:49	92.29	-3.64	(-3.79%)	922.9
2022-01-11	Tesla, Inc.	TSLA	4	227.82	911.28	2022-12-06	16:29:50	179.82	-2.63	(-1.44%)	719.28

The final output is written to an excel file named "Output" with Sheet name "Stock Prices"

Plotting to Analyze

OUTPUT



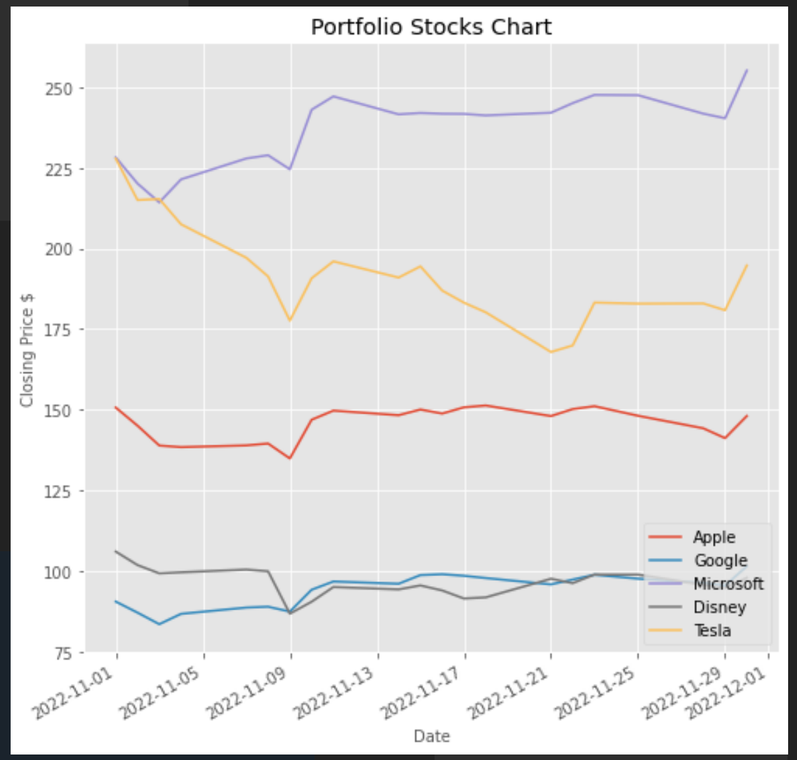
CODE

```
import datetime as dt
from matplotlib import pyplot as plt
from matplotlib import style
from pandas_datareader import data as pd

start = dt.datetime(2022, 1, 1)
end = dt. datetime (2022 ,12, 30)

Apple = pd.DataReader("AAPL", "yahoo", start, end)
Google = pd.DataReader("GOOG", "yahoo", start, end)
Microsoft = pd.DataReader("MSFT", "yahoo", start, end)
Disney = pd.DataReader("DIS", "yahoo", start, end)
Tesla = pd.DataReader("TSLA", "yahoo", start, end)

style.use('ggplot')
Apple['Close'].plot(figsize = (8,8), label= "Apple")
Google['Close'].plot(figsize = (8,8), label= "Google")
Microsoft['Close'].plot(figsize = (8,8), label= "Microsoft")
Disney['Close'].plot(figsize = (8,8), label= "Disney")
Tesla['Close'].plot(figsize = (8,8), label= "Tesla")
plt.title('Portfolio Stocks Chart')
plt.legend (loc= 'lower right')
plt.ylabel("Closing Price $", fontsize =10)
plt.xlabel("Date", fontsize =10)
# plt.grid(color= 'black', linestyle= '--', linewidth = 1)
plt.show()
```



OUTPUT

Thank you