

Department of Computing

|  |  |
| --- | --- |
| Student ID |  |
| Module Title | Data In Action |
| Module Code | P18732\_100301\_2023-24\_1 |
| Module Lecturer | Mandi Qi |
| Date of Submission |  |
| Actual Word Count |  |

Contents

1. Executive Summary
2. Data collection
   1. Value of Data in Forex Trading
   2. Data Extraction
      1. Fundamental data for analysis
      2. Technical data for analysis
   3. Data Storage
3. Data analysis
4. Data Visualization
5. Data mining for competitors
6. References
7. Individual logs
8. Appendix Code
   1. Select date
   2. Ads
   3. Complete header and details Scrapping
9. Executive Summary

In the dynamic landscape of trading, the abundance of raw data has become both a boon and a challenge for aspiring traders, whether individuals or brokers. The conventional presentation of historical data often falls short of providing actionable insights, necessitating a shift towards comprehensive data processing and extraction techniques. This report delves into the realm of data mining for trading, with a specific focus on the forex market, incorporating news and fundamental information, as well as generating novel insights through feature extraction from technical data.

Data Collection: Our primary data source for news and fundamental information is myfxbook, utilizing web scraping techniques to extract relevant data. This approach ensures a comprehensive collection of real-time, market-moving events that impact the forex market, offering a holistic view for traders.

Data Analysis: To make informed decisions, we employ machine learning algorithms, particularly focusing on time series analysis. By leveraging advanced models, we aim to discern patterns, trends, and potential market movements. This analytical framework empowers traders with predictive capabilities, enhancing their ability to act proactively in the face of market uncertainties.

Data Presentation (Visualization): The extracted insights are presented through a user-friendly interface, utilizing Python's Matplotlib and Plotly Express (px). These visualization tools are instrumental in transforming complex data into clear, actionable information, facilitating a more intuitive understanding of market dynamics.

Competitor Analysis: In addition to individual trading strategies, our report explores the competitive landscape by mining data from various providers. Through classification algorithms, we aim to categorize competitors into specific industries within the trading domain. This intelligence can aid traders in benchmarking their strategies against industry trends and competitor performance.

By synthesizing news, fundamental, and technical data through advanced data mining techniques, this report advocates for a more nuanced and informed approach to trading. The integration of machine learning models, coupled with robust visualization tools, not only empowers traders with predictive capabilities but also facilitates a comprehensive understanding of the competitive landscape. As the forex market continues to evolve, leveraging data mining becomes imperative for staying ahead in the dynamic world of trading.

1. Data collection in Forex Trading
   1. Value of Data

Data collection is a critical process in the field of data mining for trading. It involves the gathering, measuring, and analysis of various types of information using a set of standard validated techniques. The main objective of data collection is to gather information-rich and reliable data and analyze them to make critical decisions. There are two main methods of data collection in research based on the information that is required, namely: Primary Data Collection and Secondary Data Collection. Primary data refers to data collected from first-hand experience directly from the main source, while secondary data refers to data that has been collected in the past. Primary data collection methods are generally regarded as the best kind of data in research. Organizations need to collect data to research the demand, customer preferences, competitors, etc., before creating any new product. Even after the product is launched, many companies continue to collect their customers’ data to get feedback and identify ways to improve their overall customer experience. Data scientists play a crucial role in helping companies not only collect data but also organize it and derive results from it for shareholders to make decisions. (Duggal, N., 2023).

Figure 1 shows some categories of factors that affect the market.

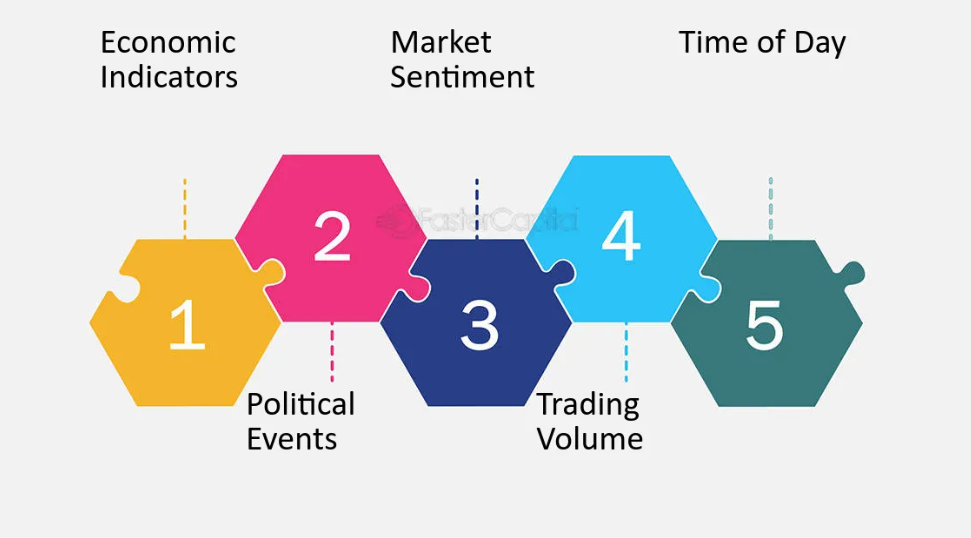


Figure (FasterCapital, 2023).

In the context of trading, fundamental data plays a crucial role in influencing trading decisions. Fundamental data includes economic indicators and news releases that can have a significant impact on the price of a currency. For example, when a major news release, such as non-farm payrolls, is announced, it can cause significant volatility in the EUR/USD currency pair, leading to sharp movements in the price. Fundamental data provides traders with insights into the underlying factors that drive the market and can help them anticipate future price movements. Traders rely on fundamental data to understand the economic conditions that may affect the value of a currency and make trading decisions based on this information. (Hayes, A., 2023; Kelly, L., 2023).

Figure 2 shows how the release of the US Bureau of Labor Statistics had an effect on the EUR/USD pair in the forex market on the 5th of July.



Figure (Tradeciety, 2023)

On the other hand, technical data plays a crucial role in informing trading decisions. Technical data includes indicators such as volume, RSI, MFI, Bollinger Bands, and others. Traders use technical data to analyze the historical price movements and the momentum of a currency, which can help them identify trends and make informed trading decisions. For example, a trader might use the RSI indicator to identify overbought or oversold conditions in the market and make a trading decision based on that information. Technical analysis, which utilizes various technical indicators, chart patterns, and statistical tools, is widely used by traders to forecast future price movements and identify potential entry and exit points for trades. It provides traders with a systematic approach to analyzing the market and can be used in conjunction with other forms of analysis, such as fundamental analysis, to make well-informed trading decisions. (Hayes, A., 2023; Kelly, L., 2023; Tuarob, S., Wettayakorn, P., Phetchai, P. et al., 2021).

Figure 3 shows some price action lines that depict the trend of the market.

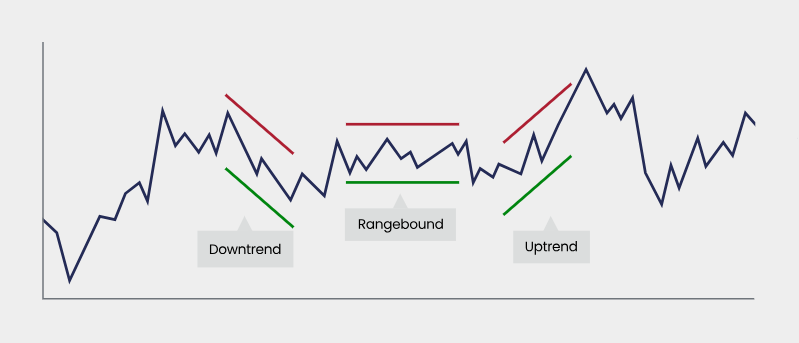


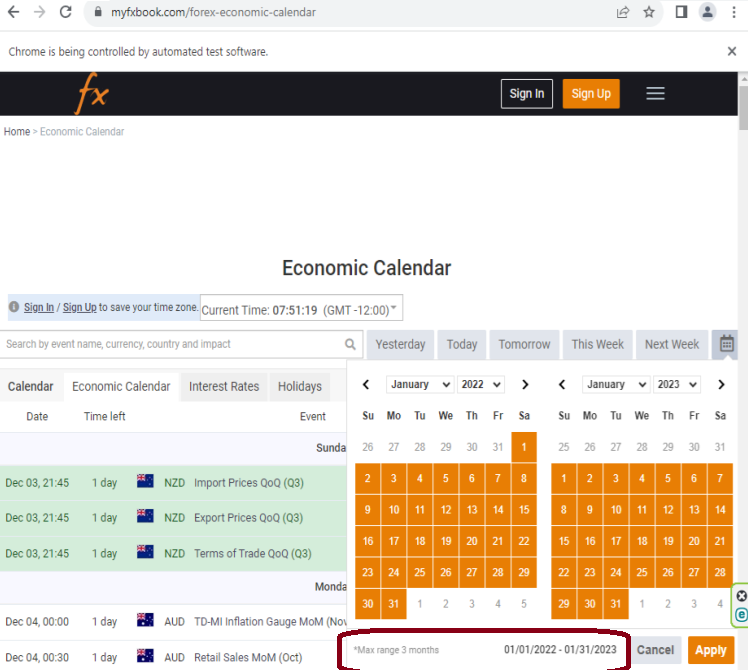
Figure (Forex., 2023)

* 1. Data Extraction:
     1. Fundamental data

In the process of collecting fundamental data, various data sources are available for scraping economic calendar news. However, they often show the same values. Therefore, there is no significant difference between collecting data from investing.uk.com or forexfactory.com. As a result, the decision was made to use myfxbook.com due to the absence of limitations in obtaining HTML from them, the reduced risk of our IP being banned, and the simplicity of the required functions.

The primary challenge stemmed from the limitations imposed by Myfxbook's landing page. This restriction manifested in the presentation of only the current week's header news, prompting the need for a more inclusive approach to gather all relevant information. To address this issue, the team opted for the implementation of Selenium, a powerful web scraping tool.

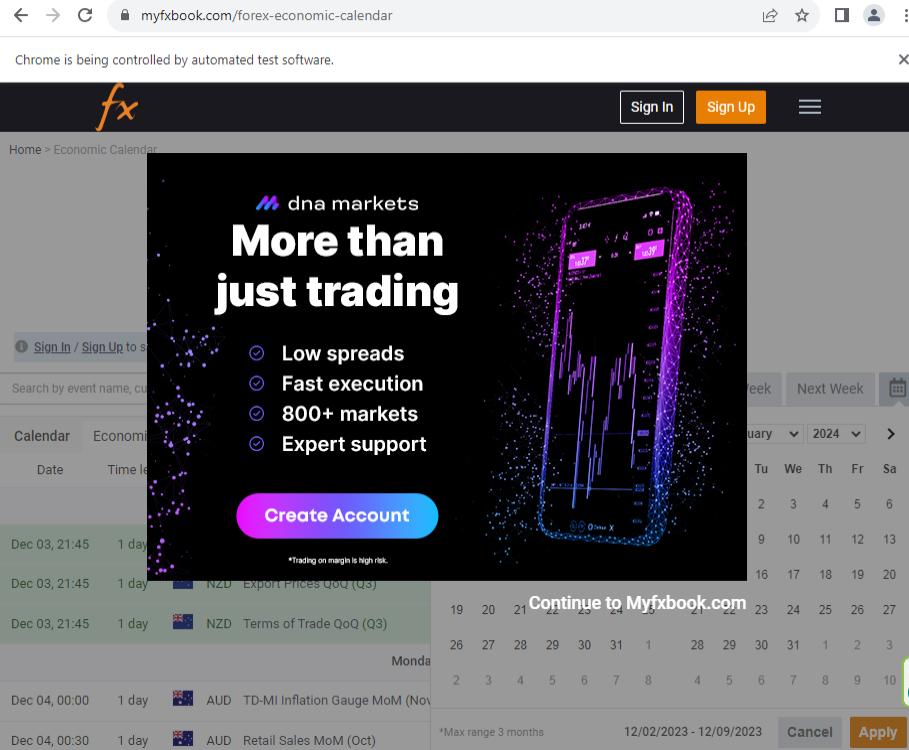
The initial step involved configuring the script to selectively choose the start and end dates for data extraction, focusing on the news headers. However, another obstacle surfaced during this phase. The website's calendar feature posed a constraint, limiting the date range to a maximum of three months (Figure 4). To circumvent the three-month limitation, a two-fold strategy was employed. First, a meticulously crafted list of dates was generated, providing the necessary flexibility for the scraping process. Additionally, a dedicated function was developed to convert standard date formats into the specific format mandated by Myfxbook. (Code related to pick date

Figure 4 shows the 3-month limitation of selecting a date.

Figure

Another significant challenge involved dealing with advertisements that appeared during the scraping process. To mitigate this issue, we had to check the existence of ads before each page navigation (see Appendix), using the try-catch and click methods in the Selenium library. (Appendix Code relate to click on ads )

Figure 4 illustrates how advertisement pop-ups can prevent web scraping.



Figure

Furthermore, the challenge of data acquisition in the context of scraping Myfxbook arises from the absence of consolidated information on a single page. This necessitates additional steps following the initial capture of news headers. Addressing this obstacle requires a systematic approach wherein the process involves scraping all available headers and subsequently saving them. Subsequently, a thorough exploration of the reference links associated with each news entry becomes imperative. By navigating through these links, one can access comprehensive information, including historical data pertinent to the respective news items. (Appendix Code relate scrapping the hears)

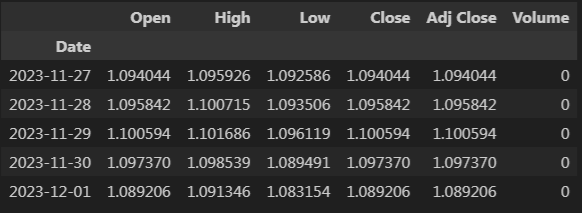
To overcome this challenge, first and foremost, all headers are scraped and stored for reference. Subsequently, a meticulous examination of the associated reference links is conducted. This involves traversing through the linked pages to gather a more in-depth understanding of the news entries and to ensure access to historical data. By adopting this comprehensive approach, the data mining process becomes more effective and yields a more complete dataset for analysis in the context of trading. (Appendix Code related to details)

* 1. Technical data

When capturing technical data, various time frames can be considered. For instance, daily data can be collected, with each row representing an open, high, low, and close candle (Figure 6), which encapsulates a day's trading activity. This approach allows for the aggregation of data at a level that is suitable for the intended analysis, without delving into more granular time frames, such as minutes.

One popular tool for collecting financial data, including forex data, is the yfinance library. This Python library provides a convenient interface for accessing and retrieving historical market data, including stock and forex prices. By using the yfinance library (Appendix Code Get EUR/USD Data), researchers and traders can easily retrieve the necessary data for their analysis and back testing. The library offers a range of functionalities for fetching data, such as stock price history, and more. It’s simple and intuitive syntax makes it a preferred choice for many practitioners in the field of algorithmic trading and financial research.

Figure 6 depicts a sample of data collected from yfinance for the EUR/USD pair.

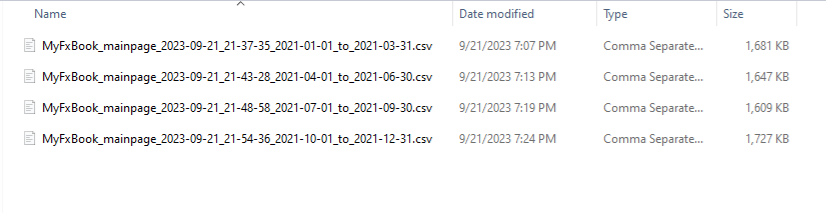


Figure

* 1. Data Storage:

In the beginning phase of the project the focus was on capturing a robust dataset comprising both fundamental and technical data. The team systematically identified and gathered relevant resources, extracting valuable information to create a comprehensive dataset. All the data was organized and structured using Python's pandas library, initially stored in a DataFrame. Recognizing the impermanence of Python data structures, particularly DataFrames, a pivotal decision was made to ensure the durability and long-term storage of the amassed data. For the preliminary stages and structural validation, the researcher employed the widely used CSV file format. This allowed for a quick and straightforward assessment of the data's format and integrity. (Figure 7)

Figure 7 displays the four-season data for the year 2021 in the explorer from the fundamental web scraping section 2.2.1.

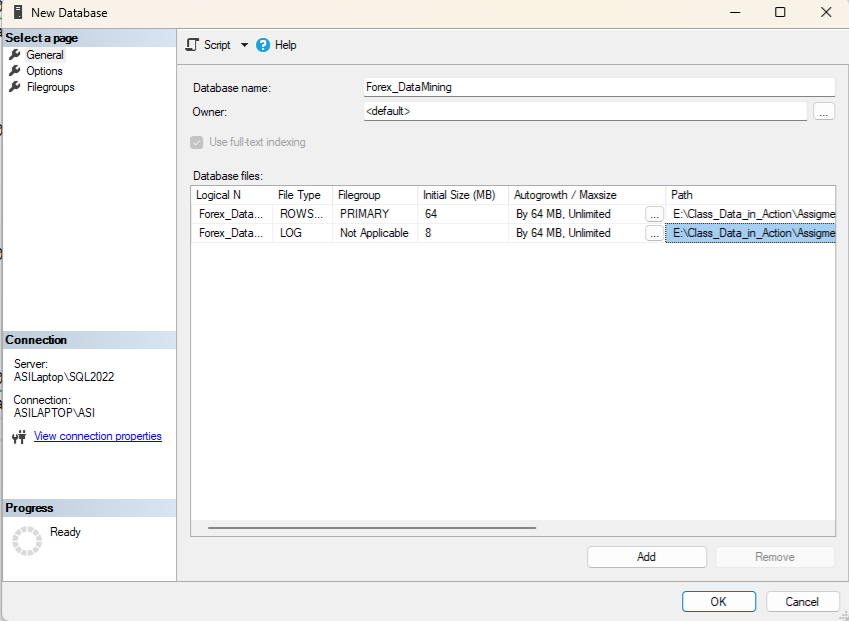


Figure

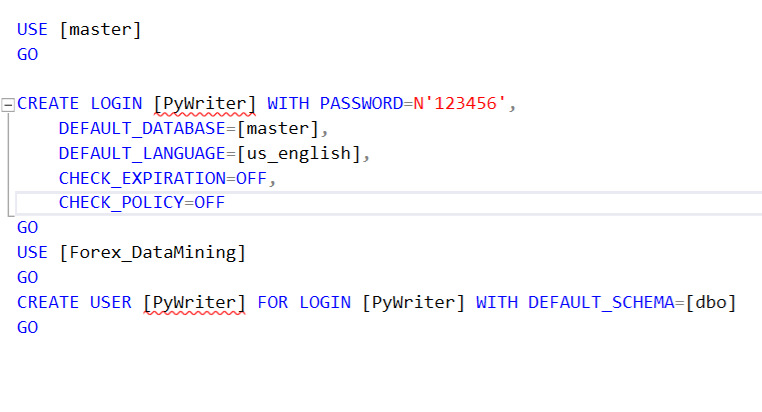
As the research progressed into the analytics phase, necessitating more complex queries and efficient data retrieval, a more durable and sophisticated storage solution became imperative. To meet this requirement, the team opted for SQL Server, a robust relational database management system. SQL Server offered a structured and tabular format, aligning with industry standards for storing financial and trading data securely.

The adoption of SQL Server not only ensured data durability but also facilitated seamless integration with analytical tools, enhancing the overall efficiency of the research process. This strategic choice allowed for the creation of a scalable and efficient system that could accommodate the growing volume of data and support advanced analytics for trading insights.

The focus is on data storing. The initial step involves creating a database in SQL Server (Figure 8) and separating users with writing access (Figure 9) for connection from Python



Figure



Figure

Creating a database in SQL Server and establishing a connection from Python are essential steps in data storing for data mining. The connection allows Python to interact with the SQL Server database, enabling the insertion of data into the tables. Asanka, D. (2022).

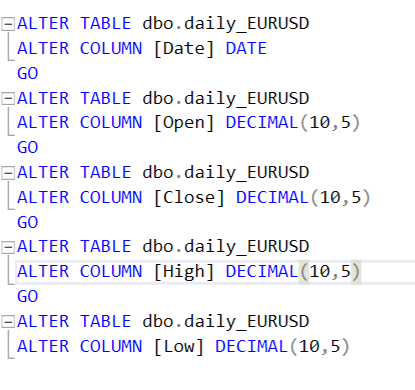
Subsequently, a connection is established from Python using SQLAlchemy (Code 1), and the data is inserted into a table. It is important to note that data types are not predefined in Python, whereas they are crucial for SQL Server in terms of precision and performance.



Code

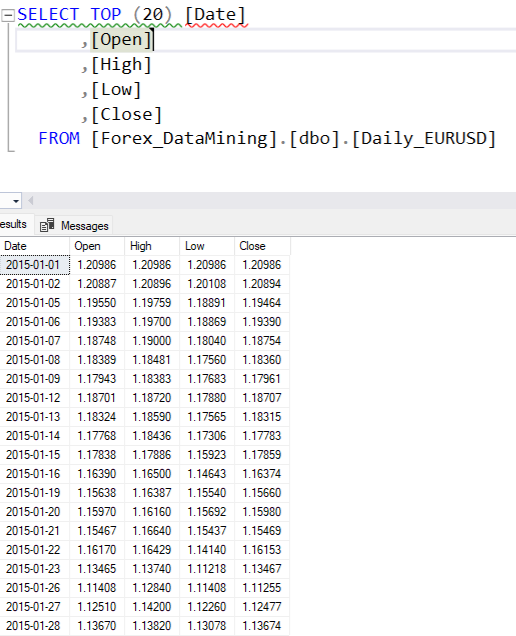
Finally, changes are made to the structure of the table to ensure it aligns with the requirements of the data mining process. (Figure 10)

Unlike Python, SQL Server requires explicit declaration of data types, which is crucial for ensuring data precision and performance. By defining appropriate data types, the integrity of the data is maintained, and storage and retrieval operations are optimized.



Figure

Figure 11 shows the EURUSD data stored in the database and a preview of the top 20 rows from the table 'daily\_EURUSD.



Figure

1. Data analysis (E)
2. Data Visualization (C)
3. Data mining for competitors (Team)
4. References

Asanka, D. (2022). Introduction to SQL Server Data Mining. Available at: <https://www.sqlshack.com/introduction-to-sql-server-data-mining/> (accessed 03 Dec 2023)

Duggal, N. (2023),” Effective Data Collection Methods: Techniques and Use Cases Explained”. Available at: <https://www.simplilearn.com/data-collection-methods-article> .(accessed 30 Nov 2023).

Twin, A. (2023), What Is Data Mining? How It Works, Benefits, Techniques, and Examples. Available at: <https://www.investopedia.com/terms/d/datamining.asp> .(accessed 30 Nov 2023).

Norm, H. (2023), Data Mining Shapes the New Age of Trading at: <https://www.linkedin.com/pulse/data-mining-shapes-new-age-trading-norm-hart> (accessed 30 Nov 2023).

Hayes, A. (2023). Understanding Trend Analysis and Trend Trading Strategies. at: <https://www.investopedia.com/terms/t/trendanalysis.asp> (accessed 30 Nov 2023).

Kelly, L. (2023). What is Trading Data? Definition, Uses, Datasets & Vendors. at: <https://datarade.ai/data-categories/trading-data> (accessed 30 Nov 2023).

Tuarob, S., Wettayakorn, P., Phetchai, P. et al. (2021), Davis: a unified solution for data collection, analysis, and visualization in real-time stock market prediction. Financ Innov 7, 56. Available at: <https://doi.org/10.1186/s40854-021-00269-7> (accessed 30 Nov 2023).

FasterCapital. (2023). Forex trading sessions: Optimal times for trading the Ngultrum.Available at: <https://fastercapital.com/content/Forex-trading-sessions--Optimal-times-for-trading-the-Ngultrum.html> (accessed 30 Nov 2023).

Tradeciety., (2023), THE MOST IMPORTANT FOREX NEWS.Available at: <https://tradeciety.com/important-forex-news> (accessed 30 Nov 2023).

Forex. (2023), Understanding technical analysis.Available at: <https://www.forex.com/en-uk/trading-academy/courses/technical-analysis/uk-introduction/> (accessed 30 Nov 2023).

1. Appendix

Code related to pick date

############################################# click on calendar button

calendar\_btn=driver.find\_elements(by=By.XPATH, value='//button[@id="calendarCustomBtn"]')[0]

# calendar\_btn=driver.find\_element(By.XPATH, "//button[contains(text(), 'Click Me')]")

calendar\_btn.click()

timesleep\_for\_datepicker=1

############################################# Select start Year

combo\_box\_start\_year=driver.find\_elements(by=By.XPATH, value='//select[@class="yearselect"]')[0]

# Create a new instance of the Select class

select\_start\_year = Select(combo\_box\_start\_year)

time.sleep(timesleep\_for\_datepicker)

# Create a new instance of the Select class

select\_start\_year.select\_by\_value(str(start\_year))

############################################# Select start Month

combo\_box\_start\_month=driver.find\_elements(by=By.XPATH, value='//select[@class="monthselect"]')[0]

# Create a new instance of the Select class

select\_start\_month = Select(combo\_box\_start\_month)

time.sleep(timesleep\_for\_datepicker)

# Create a new instance of the Select class

select\_start\_month.select\_by\_visible\_text(start\_month)

############################################# Select first day of selected month

start\_date\_picker\_value=f'//td[text()="{start\_day}"]'

start\_date\_picker=driver.find\_elements(by=By.XPATH, value=start\_date\_picker\_value)[0]

# click on start date

start\_date\_picker.click()

time.sleep(timesleep\_for\_datepicker)

# wait 5 second

driver.implicitly\_wait(5)

############################################# Select end Year

combo\_box\_end\_year=driver.find\_elements(by=By.XPATH, value='//select[@class="yearselect"]')[1]

# Create a new instance of the Select class

select\_end\_year = Select(combo\_box\_end\_year)

time.sleep(timesleep\_for\_datepicker)

# Create a new instance of the Select class

select\_end\_year.select\_by\_value(str(end\_year))

############################################# Select end Month

combo\_box\_end\_month=driver.find\_elements(by=By.XPATH, value='//select[@class="monthselect"]')[1]

# Create a new instance of the Select class

select\_end\_month = Select(combo\_box\_end\_month)

time.sleep(timesleep\_for\_datepicker)

# Create a new instance of the Select class

select\_end\_month.select\_by\_visible\_text(end\_month)

############################################# Select last day of given duration month

end\_date\_picker\_value=f'//td[text()="{end\_day}"]'

end\_date\_picker=driver.find\_elements(by=By.XPATH, value=end\_date\_picker\_value)[-1]

time.sleep(timesleep\_for\_datepicker)

# click on start date

end\_date\_picker.click()

############################################## Click on apply

apply\_btn=driver.find\_elements(by=By.XPATH, value='//button[@type="button" and text()="Apply"]')[0]

apply\_btn.click()

(Appendix Code relate to click on ads )

############################################# click on continue to MyFxBook (skip the add)

try:

    # continuetosite=driver.find\_elements(by=By.XPATH, value='//a[@class="bold color-white" and @data-dismiss="modal"]')[0]

    continuetosite=driver.find\_elements(by=By.XPATH, value='//\*[@id="popupAdContainer"]'+

                                                            '//div[@class="continue-text"]'+

                                                            '/button[@class="no-padding hover-underline no-background no-border bold color-white"]')[0]

    continuetosite.click()

except:

    print("did not find ads(continuetosite)")

(Appendix Code Get EUR/USD Data )

# Set the ticker as 'EURUSD=X'

forex\_data = yf.download('EURUSD=X', start='2015-01-01', end='2023-12-31')

# Set the index to a datetime object

forex\_data.index = pd.to\_datetime(forex\_data.index)

# Display the last five rows

forex\_data.tail()