BAX 422 - DDR - FINAL PROJECT

Group 17

eBay Data Collection: Extracting ebay Stock Information from Various Platforms

Aditya Satpute

Meghana S Kanthadai

Ian Heggen

Executive Summary

The assessment of eBay's future stock performance involves sentiment data collected through the scraping of contents from Business Insider and a notable Reddit thread. This approach is designed to measure investor sentiment towards eBay, a key entity in the e-commerce domain facing stiff competition and evolving market dynamics. By storing the scraped content in MongoDB with detailed posting dates, the analysis intends to correlate public sentiment with stock price movements. Such an examination is crucial for identifying the influence of investor reactions to news on eBay's stock, offering critical insights amidst the challenges of competition and the necessity for ongoing innovation in the e-commerce industry.

Project Background

Business Use Case: eBay aims to boost its stock performance and understand public sentiment by analyzing online discussions, which will help shape better strategies and strengthen confidence in the company. Overall goal is to increase proactive business decisions as it relates to predicting and understanding eBay stock volatility.

Problem Statement: The stock price of eBay, a leading worldwide e-commerce company, has fluctuated in the face of a fast changing industry. Many variables, including competition, shifts in consumer behavior, and conditions in the global economy can be blamed for this volatility. But one area that hasn't been well investigated is how eBay's stock performance is affected by the opinions of the general public expressed in online forums, chats, and articles. One major weakness in eBay's strategic planning and investor relations activities is the absence of thorough insights into the mood and opinions of investors, consumers, and the general public as expressed in these digital platforms.

Social media sites, financial discussion boards, and online forums are excellent places to find raw viewpoints and thoughts regarding eBay and its performance in the market. These online discussions have the power to shape the opinions of both current and prospective investors, influencing their choice to purchase, hold, or sell eBay shares. Nevertheless, eBay does not now have a methodical way to record, examine, and respond to the opinions voiced in these online debates. Without this knowledge, eBay is losing out on important information that may guide tactics to boost investor relations, boost stock performance, and eventually spur business expansion.

Data Sources

Reddit - A subreddit named “Wall Street Bets” - Wall Street Bets - WallStreetBets is a popular subreddit known for its bold stock market discussions and speculative, high-risk trading strategies, often influencing notable fluctuations in stock prices. Inside the Wall Street Bets subreddit, the word “ebay” is typed in the search bar to filter out only posts pertaining to eBay. This subreddit has over 15 million members.

Business Insider - Is a well established and respected media company that also serves as an investing resource hub containing news articles that report and evaluate events and their implications nearly in real time. These articles are likely to influence investing decisions to some extent as the platform has over 100 million unique monthly visitors. In addition to stock articles, the platform also displays and allows the download of stock data going back to any specified date and time in an excel spreadsheet. For this project, these tools were used to search eBay related news and historical stock prices.

Web Scraping Routines

Web scraping routine designed to collect data from eBay-related discussions on the Reddit subreddit WallStreetBets. It's divided into several key parts, including initialization, data extraction, content saving, and data insertion into a database. Similarly, we scraped eBay related articles on Business Insider. These articles go back four months and their content was extracted along with their posting dates. All data was then stored in MongoDB for easy retrieval.

Reddit Scraping General Overview:

The code automates a web browser to access, scroll, and collect data from a specific subreddit search result page on Reddit. It then parses the collected HTML content to extract information about individual posts, including their titles, publication dates, and content, along with comments under these posts. Finally, it saves the collected information into a MongoDB database.

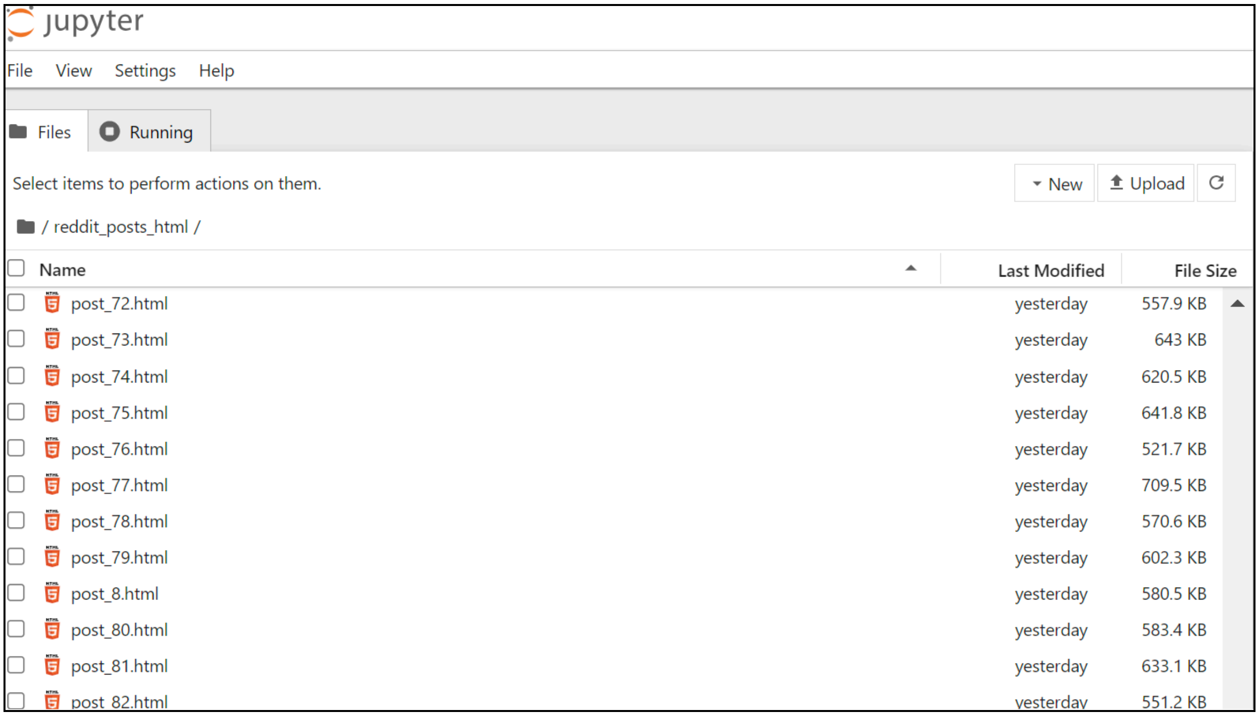
Steps for Web Scraping Reddit:

1. Browser Automation and Data Collection:

   - Initialize a Selenium WebDriver to automate a web browser (Chrome).
   - Navigate to the WallStreetBets subreddit search page for "eBay."
   - Automate scrolling to the bottom of the page to ensure all posts are loaded.
   - Use BeautifulSoup to parse the page source and extract details of each post.

2. Data Extraction and HTML Content Saving:

   - For each post identified, extract the title, date, and link to the post.
   - Navigate to each post's link to load the full post content.
   - Save the HTML content of each post to a local directory for further analysis.



Select items to perform actions on them.

/ reddit_posts_html /

□ Name	Last Modified	File Size
□ post_72.html	yesterday	557.9 KB
□ post_73.html	yesterday	643 KB
□ post_74.html	yesterday	620.5 KB
□ post_75.html	yesterday	641.8 KB
□ post_76.html	yesterday	521.7 KB
□ post_77.html	yesterday	709.5 KB
□ post_78.html	yesterday	570.6 KB
□ post_79.html	yesterday	602.3 KB
□ post_8.html	yesterday	580.5 KB
□ post_80.html	yesterday	583.4 KB
□ post_81.html	yesterday	633.1 KB
□ post_82.html	yesterday	551.2 KB

3. Parsing Saved HTML Files:

- Loop through the saved HTML files, each representing a post.
- Use BeautifulSoup again to parse these files, extracting the post's title, date, and content.
- Additionally, extract comments under each post, focusing on their text content and scores.

4. Database Insertion:

- Establish a connection to MongoDB.
- For each post, insert its details (file name, title, post date, and content) into a designated table in the database.
- For each comment under a post, insert its details (associated post ID, comment text, and score) into another table in the database.
- Ensure transactions are committed to the database to save the inserted data.

5. Cleanup and Closure:

- Close the browser session.
- Close the database connection to ensure no resources are left hanging.

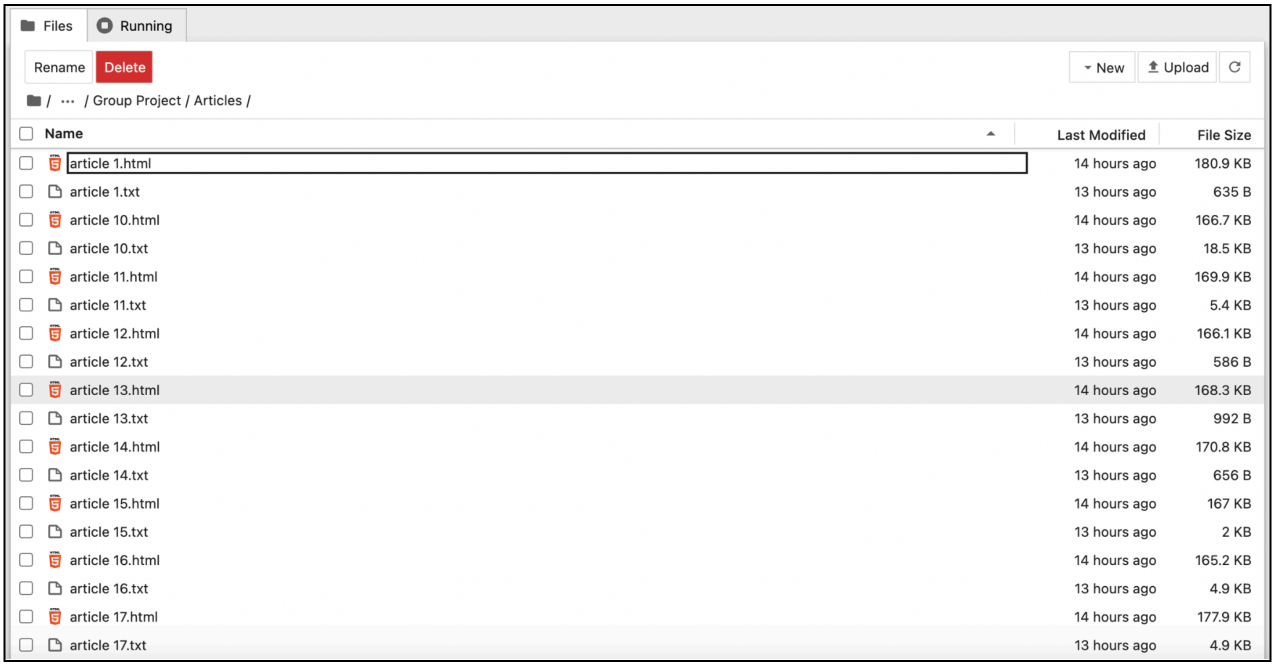
Business Insider Scraping General Overview:

The code automates a web browser to access and collect data from the eBay news section on Business Insider. Popular media sources for stock data typically prevented easy scraping, in this case, each of the 2 pages of news articles needed to be scraped independently along with posting date information. These articles were saved as HTML and then parsed into the text of the article. It also automates the local download of an excel spreadsheet containing all relevant stock data going back 4 months. Finally, it saves the collected information into a MongoDB database.

Steps for Web Scraping Business Insider:

1. Browsing Automation and Data Collection

- Browser is automated using selenium, directed to eBay news pages
- Soup.findall was used to locate each article displayed. Relevant tag was <a> with the class_= ’news-link'
- If else statement was needed for saving the correct article URL. Statement accounted for some hrefs being in URL form already while others needed to be added to a base URL.
- Links were stored, then code was written to download each link as an HTML file to be used later to extract article content in text format.



/ ··· / Group Project / Articles /

□	Name	Last Modified	File Size
□	article 1.html	14 hours ago	180.9 KB
□	article 1.txt	13 hours ago	635 B
□	article 10.html	14 hours ago	166.7 KB
□	article 10.txt	13 hours ago	18.5 KB
□	article 11.html	14 hours ago	169.9 KB
□	article 11.txt	13 hours ago	5.4 KB
□	article 12.html	14 hours ago	166.1 KB
□	article 12.txt	13 hours ago	586 B
□	article 13.html	14 hours ago	168.3 KB
□	article 13.txt	13 hours ago	992 B
□	article 14.html	14 hours ago	170.8 KB
□	article 14.txt	13 hours ago	656 B
□	article 15.html	14 hours ago	167 KB
□	article 15.txt	13 hours ago	2 KB
□	article 16.html	14 hours ago	165.2 KB
□	article 16.txt	13 hours ago	4.9 KB
□	article 17.html	14 hours ago	177.9 KB
□	article 17.txt	13 hours ago	4.9 KB

2. Article Dates

- Process for extracting article dates could not be condensed with scraping for article URLs as the website prevented it.
- Alternate approach was done after links were saved to collect date information.

**For page 2 data collection, steps 1 & 2 were repeated with a change in the URL being navigated.

3. Staying Organized

- Article HTMLs for each page was scraped into two separate lists as well as the dates for each posting
- After all content was scraped, page 1 URLs were joined with page 1 article dates. Same was done for page 2.
- Page 1 and page 2 were then joined together

4. Parsing Saved HTML Files and Saving Content as txt File:

- Looped through each HTML
- Parse was done by finding all <p> within each HTML file as trying to establish a point to end was not successful as it was not consistent across all pages.
- Parsed content from each HTML was then saved as a text file

5. Downloading Stock Data:

- Stock data table needed to be located
- Once located, page was scraped and saved the contents of the table going back four months
- All columns in the table remained. This included: date, open, high, low, close, and volume.
- Table was downloaded locally as a csv file as, “daily_stock_data”. See appendix for output.

6. Database Insertion:

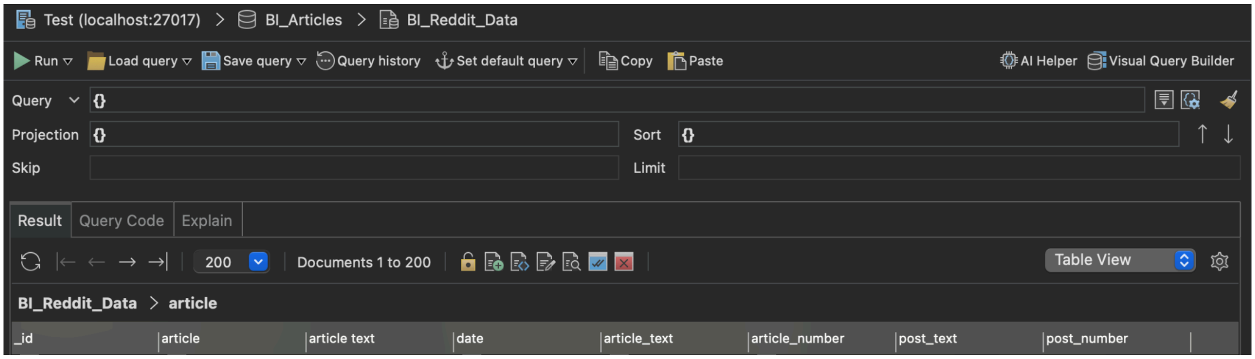
- Establish a connection to MongoDB.
- For each article, insert its details (file name, title, post date, URL, and content) into a designated table in the database.

7. Cleanup and Closure:

- Close the browser session.
- Close the database connection to ensure no resources are left hanging.

Finally, both data sources - Reddit data as well and data from Business Insider were merged on the basis of “date” and stored in MongoDB.

The output is as follows:



Query {}
Projection {}
Sort {}
Skip
Limit

200 Documents 1 to 200

_id	article	article text	date	article_text	article_number	post_text	post_number

Field names: _id, article, article text, date (for articles and Reddit posts), article number, post_text, and post_number.

Database Choices

MongoDB was selected for its dynamic schema, scalability, and superior query performance, making it ideal for analyzing eBay's stock performance data. Its ability to efficiently handle unstructured data, such as text and HTML from diverse online sources, is crucial for our web scraping approach. Notably, MongoDB's disaster recovery, high availability, and flexible data modeling are invaluable for sentiment analysis, which requires adapting to changing data formats. These features enable timely insights into eBay's market sentiment, supporting the project's need for frequent data updates and queries. MongoDB's capacity to accommodate data growth and evolving analysis needs makes it a fitting choice for this application.

Data Wrangling

We utilized advanced web scraping techniques with Python's Selenium for automating browser tasks and BeautifulSoup for extracting HTML content. A key focus was on standardizing date formats across various data sources to ensure an accurate match between sentiment data and stock price fluctuations. The collected information—comprising article titles, posting dates, and textual content—was carefully consolidated and saved in MongoDB. This procedure demanded precise attention to keeping date formats consistent and seamlessly integrating data from disparate sources for orderly storage in MongoDB.

Summary and Conclusion

This project illustrates a comprehensive approach to utilizing web scraping techniques for sentiment analysis, particularly in assessing public perception of eBay's stock performance. By leveraging Python libraries such as Selenium for browser automation and BeautifulSoup for HTML content parsing, the team was able to systematically collect and store data from two significant online platforms: Business Insider and Reddit's WallStreetBets subreddit. This methodological collection spanned four months of content, focusing on articles and discussions related to eBay, which was then meticulously parsed and stored into MongoDB for efficient retrieval and analysis.

Techniques Used:

- **Browser Automation with Selenium:** Enabled accessing dynamically loaded content and navigating through web pages to collect data on eBay stock discussions.
- **HTML Parsing with BeautifulSoup:** Facilitated the extraction of relevant textual information, including article titles, dates, and content, from the raw HTML of web pages.
- **Data Storage with MongoDB:** Offers a structured and scalable way to store the scraped data, allowing for easy retrieval based on specific queries such as article dates.

Business Implications:

- **Enhanced Market Insight:** The ability to analyze sentiment through discussions on social media and financial articles provides eBay with a deeper understanding of public and investor perceptions, potentially revealing the impact of certain events or announcements on stock performance.

- **Strategic Planning and Investor Relations**: By integrating insights from sentiment data, eBay can develop more informed strategies to enhance investor relations, address public concerns proactively, and potentially influence stock performance positively.
- **Mitigating Misinformation:** Identifying and understanding the spread of misinformation through these platforms allows eBay to take timely corrective actions, safeguarding investor confidence and stock stability.

Business questions that are answered through this project:

By analyzing sentiment trends and fluctuations in relation to eBay's stock performance, the company can answer several critical questions using the data scraped:

- How does public sentiment correlate with stock price movements?
- Is there a lag effect that should be considered?
- What topics or events trigger positive or negative sentiment?
- How can eBay improve its investor relations and public image?

The project underscores the value of web scraping and sentiment analysis in gaining actionable insights from online discussions and articles. For eBay, this strategy not only aims to enhance its understanding of market sentiments but also supports making data-driven decisions to improve stock performance and investor relations. The careful coordination of collecting, parsing, and storing data highlights the potential of web scraping in business intelligence and strategy development, especially in the fast-paced e-commerce sector where public perception can significantly impact market performance. This endeavor not only addresses the current lack of systematic sentiment analysis but also sets a precedent for using digital discussions as a valuable data source for strategic planning.

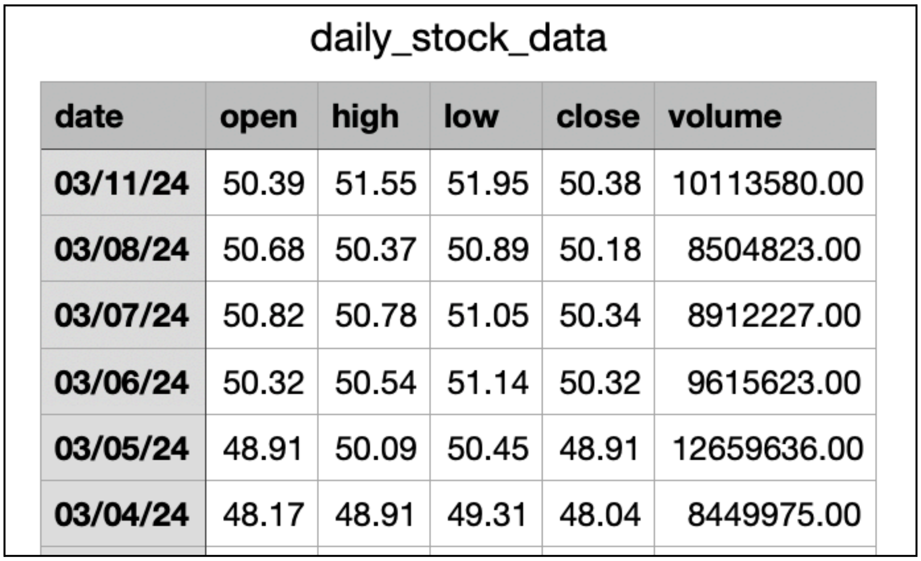
Appendix

Wall Street Bets - https://www.reddit.com/r/wallstreetbets/search/?q=ebay&restrict_sr=1

Business Insider - *eBay stock* - https://markets.businessinsider.com/stocks/ebay-stock

Business Insider - *eBay news* - https://markets.businessinsider.com/news/ebay-stock

Stock Data Output Snippet (goes back 4 months):



daily_stock_data

date	open	high	low	close	volume
03/11/24	50.39	51.55	51.95	50.38	10113580.00
03/08/24	50.68	50.37	50.89	50.18	8504823.00
03/07/24	50.82	50.78	51.05	50.34	8912227.00
03/06/24	50.32	50.54	51.14	50.32	9615623.00
03/05/24	48.91	50.09	50.45	48.91	12659636.00
03/04/24	48.17	48.91	49.31	48.04	8449975.00