A Capstone Project Report on

EXTRACTION OF FINANCIAL RATIOS AND ANALYSIS USING WEB SCRAPING AND TEXT ANALYTICS

Submitted to

Praxis Business School, Kolkata

(in fulfillment of the requirements for the award of the degree)

Post Graduate Program

in

Data Science

by

Abhishek Kangale (A19003)
Akashnil Roy (A19007)
Ameya Kanawade (A19009)
Moumita Ghosh (A19021)
Praveen Kumar V (A19024)
Vivek John P (A19040)

Under the guidance of

Prof. Dr. Subhasis Dasgupta

Praxis Business School
CELEBRATE YOUR WORTH

Department of Data Science
Academic Year: 2019 - 20

Acknowledgement

We are profoundly grateful to **Prof. Dr. Subhasis Dasgupta** Head of Machine Learning & Analytics for his expert guidance and continuous encouragement through out to see that this project meets its target since its commencement until completion.

We would like to express deepest appreciation towards **Prof. Dr. Prithwis Mukerjee** and **Prof. Charanpreet Singh**, Founders & Directors, Praxis Business School, Kolkata. **Prof. Dr. Sourav Saha**, Academics Dean, **Prof. Amit Parakh** and **Prof. Dr. Manaswee Kumar Samal**, Head of Finance whose invaluable guidance supported us in completing this project.

Lastly, we would like to express our sincere heartfelt gratitude to all the staff members of Data Science Department who helped us directly or indirectly during this course of work.

Abhishek Kangale
Akashnil Roy
Ameya Kanawade
Moumita Ghosh
Praveen Kumar V
Vivek John P

Date: 17 April 2020

Contents

List of Figures

Abstract

Investors need to gauge the company's performance and financial condition before making investments. But they can't alone depend on financial statements as they don't explain the complete picture of the company. For instance, a company might be clocking higher revenues or margins but it's business might be highly leveraged by debts or not having enough cash to survive. At this point, investors can rely on financial ratios, however, no company readily calculates and publishes its financial ratios.

In our project, we extracted the financial ratios and analysis of the company's performance and outlook through summarizing the chairman's statement and the director's speech. Users can get all this information by searching the company name or the company's NSE/BSE number on the GUI developed by us. We integrated our GUI with our backend python code that can extract all these data from a website by using web scraping and text analytics techniques. Through this project, we want to position ourselves in helping investors find important information for making their best investments.

Chapter 1

Problem Statement

Financial ratios are important indicators of a company's performance and financial condition. These ratios are calculated from the financial reports released by the companies. Analyzing financial reports of a company can be a very tedious job. Because annual reports of many companies are over 100 pages which consist of several financial jargons. If you do not understand what these terms mean, you won't be able to read the reports efficiently.

Nevertheless, several financial ratios have made the life of investors very simple. Now, you do not need to make several calculations and you can just use these financial ratios to understand the gist. Understanding of these ratios helps investors and analysts to communicate and evaluate the strengths and weaknesses of individual companies or industries. It also measures companies' operational efficiency, liquidity, stability and profitability, giving investors more relevant information than raw financial data.
Ratios over the years are used to analyze and to compare trends. Hence, our objective is to web scrape the financial ratios, summarizing the chairman's statement and director's speech so that investor can have an understanding about the company's financial condition as well as the brief of the company's performance from its top management. We also extracted the company's latest news from multiple business media websites summarizing the sentiment towards the company. This would save investors' time in analyzing and deciding whether the company is worth investing or not.

Chapter 2

Roadmap of our Project

We tried implementing our project in two methods. In the first method, we took the company's annual report PDFs as input and extracted text of the chairman's statement, director's speech and also extracted tables of financial statements. However, due to certain challenges faced during this process, we had to try another method of web scraping the financial ratios, chairman's statement, and director's speech from website **Money Control**

2.1 Method 1: Working with PDFs

1. Reading text from company's annual reports which are in PDF format. Used **PYPDF2** to reach pages having chairman's statement and director's speech.

2. Used Tabula and Camelot libraries to extract financial statements from PDF, which are in tabular format.

3. Challenges faced working with company annual report PDFs

   (a) Unable to extract information in proper format from few companies' annual reports, as financial statements are not getting extracted properly.

   (b) As there are consolidated and standalone financial statements for a few companies, table extraction tools are detecting multiple instances of statements creating a DataFrame with repeated information that include both the statements.

   (c) As we are working upon Indian companies, we extracted only three years of the annual report as the country implemented new accounting standards based on the International Financial Reporting Standards (IFRS) effective from April 1, 2016. After the implementation of new rules, reporting of

financial statement format has changed which made us ignore earlier annual reports.

2.2 Method 2: Web Scraping & Text Analytics

1. Overcame these challenges by web scraping the financial ratios, director's speech, chairman's statement from the website **Money Control** and fifteen latest news articles about company from a business website.

2. Used text analytics techniques to summarize director's speech and chairman's statement as well as summarized the sentiment towards the company from the extracted articles.

3. Created a GUI showing the financial ratios and graphs representing ratios, summarization of director's speech and chairman's statement, and fifteen news articles about the company as well as the sentiment towards the company from these articles.

Chapter 3

Working with PDFs

3.1 Data Collection

Public companies are intended to keep the company's investors and potential investors up to date on the company's performance. Hence, these companies file earnings reports after the end of their first three quarters, and both quarterly and annual reports after their fiscal year ends.
In our project, input data includes three years annual report PDFs
(2018-19, 2017-18, 2016-17) of top 10 pharmaceutical companies in India comprising 30 annual reports collected from the company's respective websites.

3.2 Reading text from PDF

PDFs are one of the most important and widely used digital media. Most of the organization release their annual reports in PDFs only. We can extract the required text from these PDFs to draw insights about the company, otherwise which is a significant challenge for investors or analysts to go through hundreds of pages of entire annual report. Python has libraries that are well integrated and provide the solution to handle unstructured data sources like PDF and could be used to make it more sensible and useful.

* ***PYPDF2:***

  In our project, we used PyPDF2, a pure-python PDF library capable of splitting, merging, cropping, and transforming the pages of PDF files. It can also add custom data, viewing options, and passwords to PDF files. It can retrieve text and metadata from PDFs as well as merge entire files together.

3.3 Reading tables from PDF

We used two methods to extract tables from PDFs in our project. One method of extraction is through using tabula-py, a python tool and the second method is by using Camelot, a Python library.

3.3.1 Tabula

Tabula is a simple Python wrapper of tabula-java, which can read table of PDF. We can read tables from PDF and convert into Python Pandas's DataFrame. It also enables us to convert a PDF file into CSV/TSV/JSON file.

3.3.2 Camelot

Camelot is a Python library that makes it easy for anyone to extract tables from PDF files. It also exports tables to multiple formats, including CSV, JSON, Excel and HTML.

Both these libraries use Stream and Lattice parsing methods. Stream is used to parsing tables that have whitespaces between cells to simulate a table structure, while Lattice is used to parse tables that have demarcated lines between cells. We found that Camelot works better than Tabula in all Lattice cases. Whereas Tabula does better table detection for Stream cases, but it still fails to give good parsing output, which Camelot solves for with its configuration parameters.

3.4 Challenges in working with PDFs

As we are working on PDFs of multiple Indian companies, we found that each company publishes an annual report in more graphically appealing for investors. Hence, every company uses its format to publish an annual report. Due to such disparity between documents, both the table extraction methods failed to extract financial statements in a structured format, which is very critical in our project as we can get the best results when the tabular data is well structured.
Few Indian companies have standalone and consolidated financial statements. Consolidated financial statements cover all the activities of the entire group as a whole including subsidiaries. Standalone financial statements report these findings as a separate entity. If a company has both these financial statements, table extraction tools are detecting multiple instances of statements creating a dataframe with repeated information that include both the statements, which becomes difficult for us to work upon

as it consumes a lot of time to preprocess the extracted data.
Due to the above challenges, we forgo this method to automate financial ratio calculations from the company's annual report PDFs.

Chapter 4

Web Scraping

4.1 Overview of Web Scraping

Web Scraping also known as web data extraction is an important technique used for extracting unstructured data from the websites and transforming it into structured data such as table format. Web Scraping is also identified as web data scraping, screen scraping or web harvesting. It is a form of data mining, the basic and important goal of this process is to extract data from a website and transform it into comprehensible formats like spreadsheets, database, an API, or a comma-separated values (CSV) file. But, doing web scraping is not a simple task in most cases as the websites come in various forms and types. So, web scraping differs in functionality and features.

Figure 4.1: Basic Web Scraping Architecture

As we know, web scraping is a technique used to extract information from web pages based on script routines. Web pages are documents written in Hypertext Markup Language (HTML), and more recently XHTML which is based on XML.
Web documents are represented by a tree-structured called the Document Object Model, or simply the DOM tree and the goal of HTML is to specify the format of text displayed by Web browsers.

4.2 Web Scraping Technique

* ***Document Object Model (DOM) Parsing***

We used DOM parsing technique for extracting information from webpages. In DOM parsing, by embedding a full-fledged web browser, such as the Chrome browser control, programs can retrieve the dynamic content generated by client-side scripts. These browser controls also parse web pages into a DOM tree, based on which programs can retrieve parts of the pages.

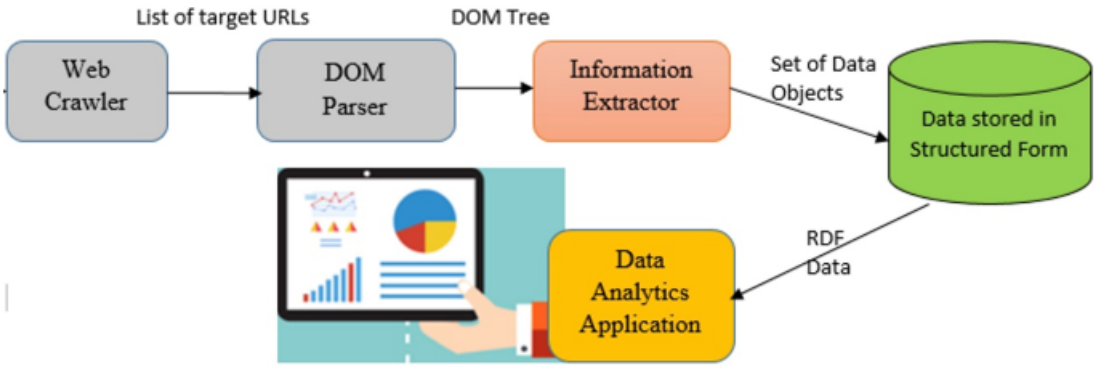


Figure 4.2: Document Object Model Parsing

* ***Web Scraping Tool: Selenium Web driver***

We used Selenium, a popular tool for automating browsers. It's primarily used for testing in the industry but is also very handy for web scraping. Selenium requires a driver to interface with our chosen browser. We worked with Chrome browser, so Chrome requires ChromeDriver, which needs to be installed before we start scraping. The Selenium web driver speaks directly to the browser using the browser's engine to control it. In case, if we used the Mozilla browser for web scraping information, we could have used gecko web driver.

Chapter 5

Text Analytics

5.1 Overview of Text Summarization

Text Summarization is a process of extracting or collecting important information from the original text and presents that information in the form of summary. With text mining, the information to be extracted is clearly and explicitly stated in the text.

In general, there are two different approaches for automatic text summarization: extraction and abstraction. Both approaches are used for summarizing text either for a single document or for multi documents. Abstractive summarization methods consist of producing original text in a new way. Whereas Extractive summarization methods work by identifying important sections of the text and generating them in same words, thus, these methods depend only on the extraction of sentences from the original text.

Extractive summarization methods oftentimes give better results than automatic abstractive summarization methods. Because, abstractive summarization methods cope with problems including inference and natural language generation, a semantic representation which is relatively harder than data-driven approaches such as sentence extraction. There is no completely developed abstractive summarization method currently. Existing abstractive summarizers often rely on an extractive pre-processing component to produce the abstract of the text.

5.1.1 Extractive Summarization

The common methods for extractive summarization are Term Frequency/Inverse Document Frequency (TF/IDF) method, graph-theoretic approach, cluster-based method, machine learning approach, artificial neural networks, Latent Semantic Analysis (LSA) method, fuzzy logic, query-based, concept-obtained text summarization, using a regres-

sion for estimating feature weights, multilingual, Maximal Marginal Relevance (MMR), topic-driven summarization, centroid-based summarization, etc.
To better understand how extractive summarization systems work, we describe three fairly independent steps which all summarizers perform:

Step 1: Construct an intermediate representation of the document which expresses the main aspects of the text.
Step 2: Score the words, sentences, and graphs based on the representation.
Step 3: Select a summary comprising of a relevant number of sentences.

* ***Intermediate Representation***

  In this step, some pre-processing such as tokenization, noise removal, stop word removal, stemming, frequency computation, sentence splitting, etc are applied. There are two types of approaches based on the representation: indicator representation and topic representation. Topic representation methods transform the text into an intermediate representation and interpret the topics in the text. Indicator representation methods describe every sentence as a list of features of importance such as position in the document, sentence length, having certain phrases, etc.

* ***Sentence Score***

  When the intermediate representation is generated, we assign an importance score.

  - Word scoring - assign scores to the most important words
  - Sentence scoring - verifying sentences features such as its position in the document, similarity to the title, etc.
  - Graph scoring - analysing the relationship between sentences

* ***Summary Sentences Selection***

  In this step, the summarizer systems use a specific sorting order to select the most important sentences to produce a summary. Some methods use greedy algorithms to select the most important sentences and some methods convert the selection of sentences into an optimization problem where a collection of sentences are chosen, considering the constraint that it should maximize the overall importance and coherency and minimize the redundancy.

5.2 Text Summarization using Gensim

We used Gensim library for text summarization in our project. Gensim is an open-source python library for unsupervised topic modelling and document indexing for the target audience containing information retrieval (IR) and natural language processing (NLP) community.
This library can automatically summarize the given text, by extracting one or more important sentences from the text. This summarizer is based on the "TextRank" algorithm and was later improved upon "BM25 ranking function".

* ***TextRank Algorithm:***

    - In pre-processing the text, it removes stop words and stem the remaining words.
    - Create a graph where vertices are sentences.
    - Connects every sentence to every other sentence by an edge.
      The weight of the edge depends on how similar the two sentences are.
    - Runs the PageRank algorithm on the graph.
    - Pick the vertices(sentences) with the highest PageRank score.
    - In the original TextRank, the weights of an edge between two sentences are the percentage of words appearing in both of them. Gensim's TextRank uses Okapi BM25 function to check how similar the sentences are.

5.3 Keyword Extraction using Gensim

We also extracted the most important keywords from the chairman's statement and the director's speech using Gensim library. Keyword extraction works in the same way as summary extraction (i.e. sentence extraction), in this method the algorithm tries to find words that are most important or seem representative of the entire text. The keywords extracted are not always single words; in the case of multi-word keywords, these keywords are typically all nouns.

Chapter 6

Sentiment Analysis

Sentiment Analysis is the process of determining whether a piece of an article/writing is positive, negative, or neutral. It is also known as opinion mining, deriving the attitude or opinion of an author.

6.1 Why Sentiment Analysis?

We did sentiment analysis on the company's 15 latest news articles extracted from a business news website to determine the author's opinion on the company. By determining the opinion on the company, investors could decide whether to invest in a company or not based on the latest news about the company. This feature can be more useful to short-term investors who do intra-day trading.

6.2 Sentiment Analysis Technique

* ***VADER Sentiment Analysis***

  VADER (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of A sentiment lexicon is a list of lexical features

  (e.g., words) which are generally labelled according to their semantic orientation as either positive or negative.

  VADER not only tells about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is.

  The Compound score is a metric that calculates the sum of all the lexicon ratings which have been normalized between

  -1 (most extreme negative) and +1 (most extreme positive).

* ***Scores***

  Positive Sentiment: (compound score $\leq$ 0.05)

  Neutral Sentiment: (compound score $>$ -0.05) and (compound score $<$ 0.05)

  Negative Sentiment: (compound score $\leq$ -0.05)

Chapter 7

InvestEazy GUI (Graphical User Interface)

User searches a company name, for example, Aurobindo Pharma Ltd. or its BSE|NSE number in the InvestEazy search bar and clicks on search button.

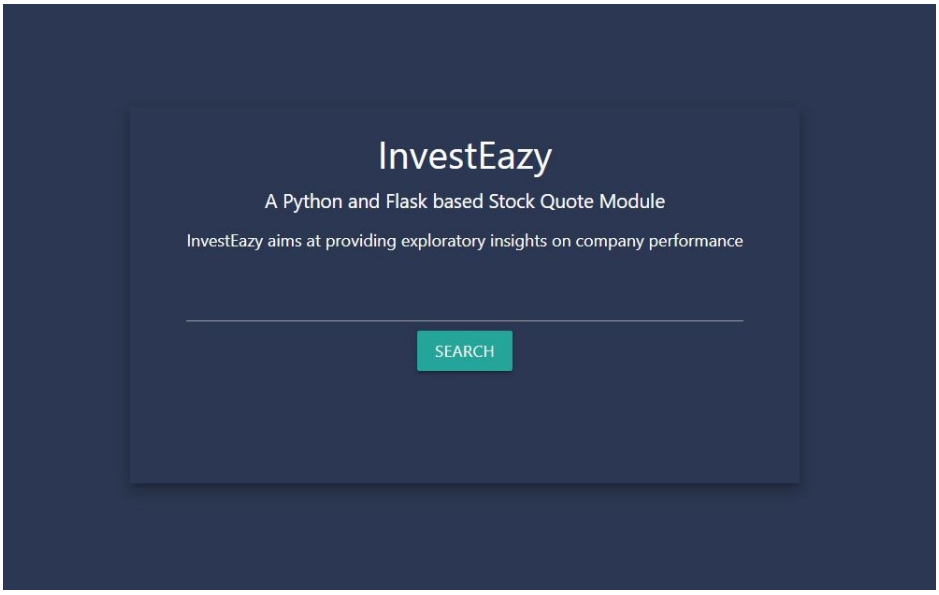
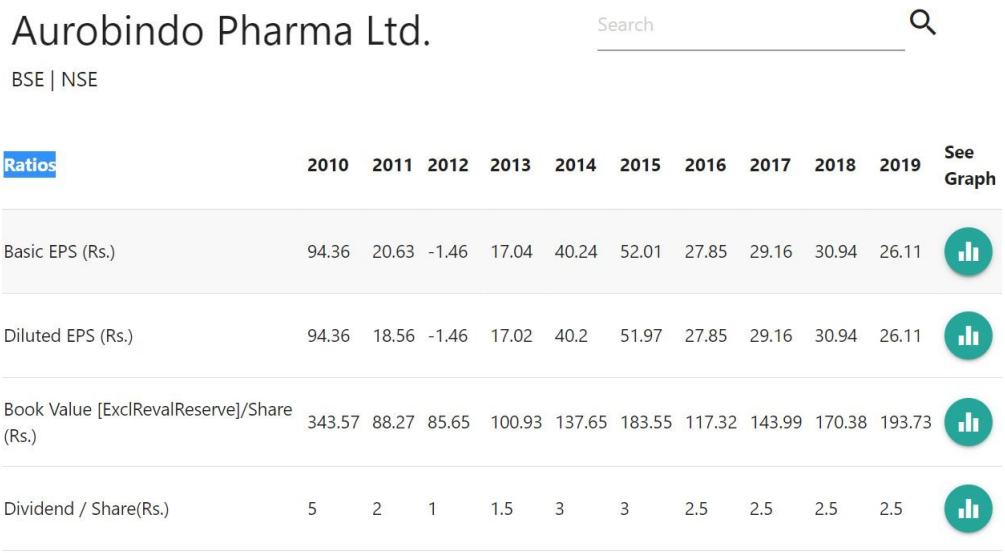


Figure 7.1: InvestEazy Homepage

Then, website redirects to a webpage containing the handpicked 15 financial ratios of the company spanning over the 10 years and their graphs.
Please note that we can extract the handpicked financial ratios of all industries excluding banking companies as the financial ratios of banks differ from other industries.



Ratios	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	See Graph
Basic EPS (Rs.)	94.36	20.63	-1.46	17.04	40.24	52.01	27.85	29.16	30.94	26.11	
Diluted EPS (Rs.)	94.36	18.56	-1.46	17.02	40.2	51.97	27.85	29.16	30.94	26.11	
Book Value [ExclRevalReserve]/Share (Rs.)	343.57	88.27	85.65	100.93	137.65	183.55	117.32	143.99	170.38	193.73	
Dividend / Share(Rs.)	5	2	1	1.5	3	3	2.5	2.5	2.5	2.5	

Figure 7.2: Finacial Ratios

In the same webpage, if the user scrolls down, he/she can find the Chairman's statement summary as well as the keywords used by the chairman. Click on the read more button to get the full Chairman's statement and click on the show less button to get the Chairman's statement summary.
Similarly, in the tab next to the Chairman's statement, user can see the Director's speech summary as well as the keywords used by the Director in the same tab. Click on the read more button to get the full Director's speech and click on the show less button to get the Director's speech summary.

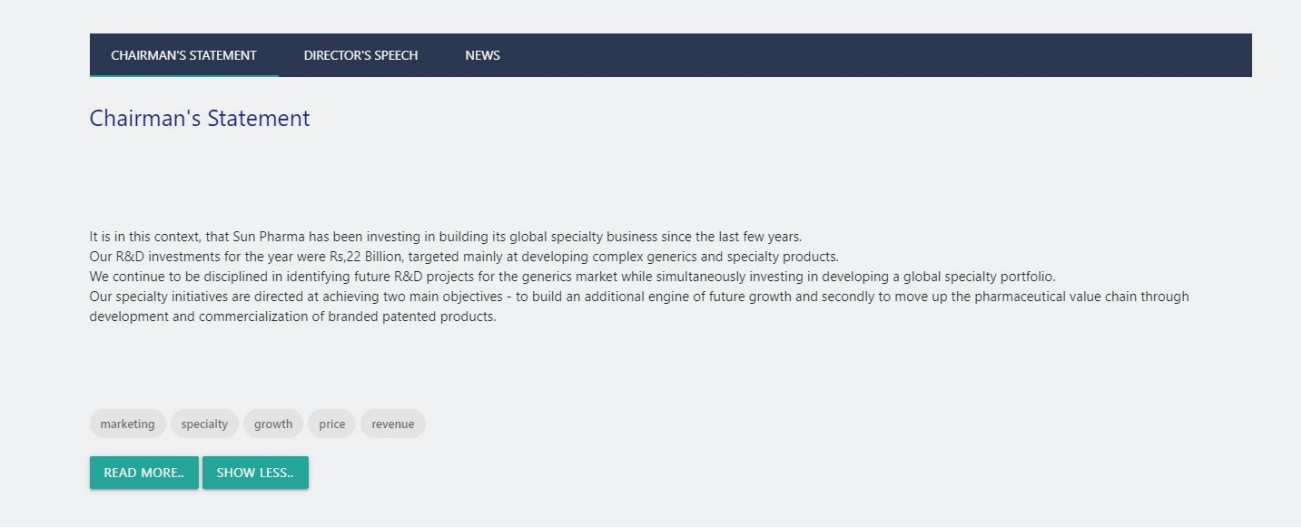


Figure 7.3: Chairman's Statement Summary and Keywords

User can also find the News tab in which 15 latest news articles of the company are extracted from a business website. Clicking on any news article, redirects to the original webpage of the article. A Donut chart representing the sentiment towards the company which is summarized from the extracted news articles.

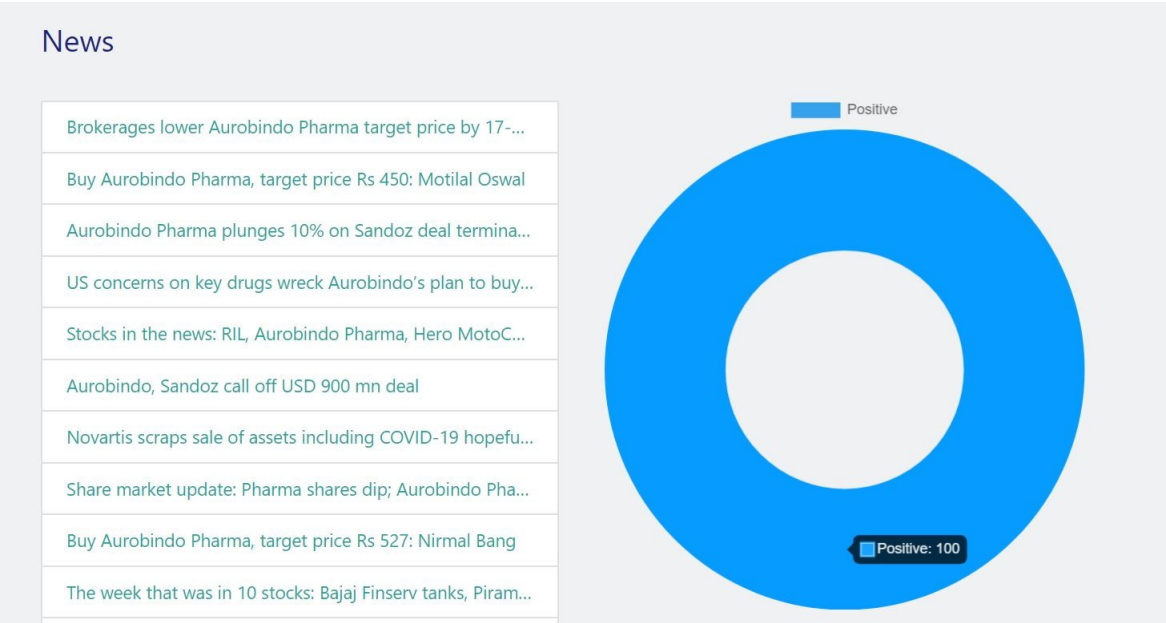


Figure 7.4: News Articles and its Sentiment

Chapter 8

Future Scope & Conclusion

Future Scope

We hope, we can improve InvestEazy on certain aspects in the future.

- Accommodate financial ratios for banking institutions as the financial ratios for banking industry differ from other industries
- Show current and historical data of the company's stock price
- Search with a company name or BSE number
- Add sentiment for every news article extracted along with the overall sentiment as we are showing now
- Extract news articles from three different websites instead of one website to remove the biases in articles
- Make a database of companies, stock price, director's & chairman's statements, news articles which will be updated in regular intervals which will act as the source for our webpage

Conclusion

In this project, we extracted financial ratios and summarized the chairman's statement and the director's speech, also extracted the company's latest 15 news articles and summarized the sentiment towards the company. To achieve this, we used web scraping, extractive text summarization technique, and Vader sentiment analysis. We showed all this information in a single webpage which could save investors' time and help them make best investments.

Chapter 9

References

1. https://tradebrains.in/tag/most-important-financial-ratios-to-analyze-a-company/
2. https://smallbusiness.chron.com/advantages-financial-ratios-3973.html
3. https://pypi.org/project/tabula-py/
4. https://towardsdatascience.com/pdf-preprocessing-with-python-19829752af9f
5. https://github.com/chezou/tabula-py/blob/master/examples/tabula_example.ipynb
6. https://camelot-py.readthedocs.io/en/master/
7. https://github.com/atlanhq/camelot/wiki/Comparison-with-other-PDF-Table-Extraction-libraries-and-tools
8. https://camelot-py.readthedocs.io/en/master/user/how-it-works.html
9. https://www.parsehub.com/blog/what-is-web-scraping/
10. https://www.w3.org/TR/WD-DOM/introduction.html
11. https://www.guru99.com/locators-in-selenium-ide.html
12. https://www.pluralsight.com/guides/web-scraping-with-selenium
13. https://www.ijarcce.com/upload/2016/march-16/IJARCCE%2040.pdf
14. https://arxiv.org/pdf/1707.02268.pdf
15. https://www.researchgate.net/publication/325115666_A_Comprehensive_Survey_on_Extractive_Text_Summarization_Techniques
16. https://pypi.org/project/gensim/
17. https://github.com/RaRe-Technologies/gensim/#documentation
18. https://rare-technologies.com/text-summarization-with-gensim/

19. https://medium.com/the-artificial-impostor/use-textrank-to-extract-most-important-sentences-in-article-b8efc7e70b4
20. https://radimrehurek.com/gensim/auto_examples/tutorials/run_summarization.html#sphx-glr-auto-examples-tutorials-run-summarization-py
21. https://www.geeksforgeeks.org/python-sentiment-analysis-using-vader/