Black Swan Impact Predictor

Dhruvi Manish Shah
Software Engineering
San Jose State University
San Jose, CA, USA
dhruvimanish.shah@sjsu.edu

Sandesh Harishchandra Gupta
Software Engineering
San Jose State University
San Jose, CA, USA
sandeshharishchandra.gupta@sjsu.edu

Puneet Tokhi
Software Engineering
San Jose State University
San Jose, CA, USA
puneet.tokhi@sjsu.edu

Hangyuan Xu
Software Engineering
San Jose State University
San Jose, CA, USA
hangyuan.xu@sjsu.edu

***Abstract*—During the course of history, there have been numerous events that have shaken the stock market to its core. What differs all other events from a Black Swan event (BSE) is the element of surprise and its severe negative consequences. The magnitude of a Black Swan event's impact distinguishes itself from all other events because a BSE can't be predicted or forecasted. However, with the recent development in the field of AI and Machine Learning, humans are able to do things that weren't possible in the past. Similarly, this project is an attempt to employ the power of machine learning to build a model by analyzing global headlines and creating an iOS app that predicts whether a global event could be classified as a Black Swan event for the US stock market.**

Keywords—Black Swan events, News Analysis, Stock Market, Machine Learning, Investors

I. INTRODUCTION

The term Black Swan event, which refers to an unpredictable, rare, catastrophic event was coined by Nassim Nicholas Taleb in his 2007 book, "The Black Swan: The Impact of the Highly Improbable." However, the actual term Black Swan originates from the (Western) belief that all swans are white because these were the only ones accounted for.[1]. Over the years, there have been many Black Swan events in the stock market such as the 9/11 terrorist attack, the 2008 housing market crash, etc. The recent example of a black swan event that obliterated the stock market was the COVID-19 pandemic. The pandemic not only crippled the US stock market but also destroyed financial markets around the globe. Many people saw the value of their 401(k) account's dissipate overnight which hurt retirees the most since it was their only source of income. Even though the stock market has recovered now, there could still be a looming threat of another black swan event in the future. Therefore, in this project, our team decided to develop the BlackSwanImpactPredictor iOS app with the mindset to prevent such problems in the future and help both individual and institutional investors protect their portfolios. The idea was also to protect individuals' portfolios from market crashes by giving them a heads up about potential market swings based on global headlines.

II. PROJECT DESCRIPTION

The machine learning model in the backend of the app first takes into account historical news and tries to compute similarities with the current news around the world. It then gets historical stock data for the S&P 500 to compare the stock trend and how the stock reacted to similar news in the past. Finally, it computes all the parameters and generates sentiment based on news, a chart, and an impact score from 1-10 with 10 being the highest impact.".

A. Frontend

The application's frontend was built using the Swift programming language and XCode was used as an integrated development environment. The application used Cocoa Pods, which is a dependency manager to integrate external libraries. In this application, the only external library used was "Alamofire" which was used for networking and is a popular choice because of its low latency. To get the impact score of a particular news, the frontend of the application only takes in the URL as an input parameter and returns the Stock movement chart, News Sentiment, and the Impact Score, as shown in the code snippet in Fig 1.

```
public func getImpactScore()->(imageData: UIImage?,  sentiment: String,
    let json = ["data_type": news_type]
    let jsonData = try? JSONSerialization.data(withJSONObject: json, opt

    let url = URL(string: "http://18.222.25.85:5000/api/process-model")
    var request = URLRequest(url: url)
    request.httpMethod = "POST"
```

Fig. 1. Code snippet of Impact Score function which only takes the backend model URL as an input parameter

In Fig 2, we can see 3 randomly selected news articles that are displayed in table view cells. Each table view cell is clickable and on clicking any of the news articles, the application segues to the next screen. The next screen has an "Analyze" button,

and on clicking the analyze button, the application displays the Stock movement chart pertaining to that specific news, the News Sentiment, and the Impact Score, as shown in Fig 3.

Fig. 2. News is displayed in clickable table view cells

Fig. 3. UI Showing Stock movement Chart, News Sentiment, and Impact score

B. *Machine Learning Model*

The application's frontend was built using the Swift programming language and XCode was used as an integrated development environment. The application used Cocoa Pods, which is a dependency manager to integrate external libraries. In this application, the only external library used was "Alamofire" which was used for networking and is a popular choice because of its low latency. To get the impact score of a particular news, the frontend of the application only takes in the URL as an input parameter and returns the Stock movement chart, News Sentiment, and the Impact Score, as shown in the code snippet in Fig 1.

(i) Dataset

The machine learning model uses historical news and S&P 500 Index data to perform the analysis. The historical news data is sourced from multiple news portals like CNBC, The Guardian, BBC, Reuters, Indian Express among others. As of now, the historical news is fetched manually and stored into a CSV file. The news details consist of the news' dates, headlines, and the URL. The stock data for S&P 500 Index is procured from the NASDAQ portal[2]. It contains data of the past 10 years with dates ranging from 11/23/2011 to 11/23/2021. The dataset includes features like Date, Volume, Close price, Open price, High price, and Low price.

(i) Model Implementation

Fig 4 gives an overview of the flow of the ML model. The input to the model is the news URL, whose impact is to be analyzed.

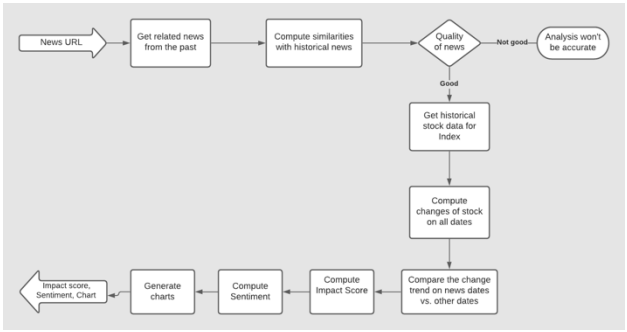

Fig. 4. Flow Diagram of ML model

First, the model gets all the historical news related to that particular news present in the CSV file. After that, the similarity scores are computed between the news and the historical news, using the Dandelion API. For instance, if there are 10 historical news stories, the similarity score is computed between the news to be analyzed and all the 10 historical news. Once the scores are computed, the mean of scores is taken. This gives an idea about the quality of the data. If the historical news is not similar to the current news, the analysis won't be accurate. Mean scores > 0.5 can be considered as a good score to proceed with the analysis.

After that, the model retrieves the historical data of the S&P stock index and we proceed with the analysis. The stock movement is computed for each day by subtracting the closing and opening prices of the stock. The stock change data is segregated into two sets. The first set contains the stock movements on the days of the historical news while the second set contains the movement on other dates. Students' t-test statistics are applied to the two sets to compare the trends of the stock movement. The t-test has a p-value as one of the output variables. A p-value represents the probability that the results from sample data occurred by chance. A p-value of 0.01 would indicate that there is only a 1% probability that the results from an experiment happened by chance. Low p-values, in this case, mean that the stock data movement on historical news dates did not occur by chance and is most likely an impact of the news.

The p-value is converted to a scale of 1-10 to convey the Impact Score. Impact Score indicates the potential impact of the news on the stock market. A score of 1 would mean no impact, and 10 being the most impactful. The sentiment of the news article is computed using the Dandelion API. A negative sentiment score with a high Impact Score would mean that the news could cause the stock market to decline. Whereas, a positive sentiment score with a high Impact score could result in an overall gain in the stock market. The model generates a pictorial chart of the stock market movement on the historical news dates. This would help users to see the patterns in a concise manner in a single image. After all the computations and analysis, the model returns the following three variables (Impact Score, Sentiment, Stock movement chart) as the output which is then received by the frontend.

C. Backend

The backend consists of a Flask server built using Python. It has one endpoint 'process_news' that takes the news URL as an input and sends the URL to the ML model for analysis. The ML model returns Impact Score, News Sentiment, and the Stock Movement Chart which is passed back to the UI.

III. Architecture Diagram

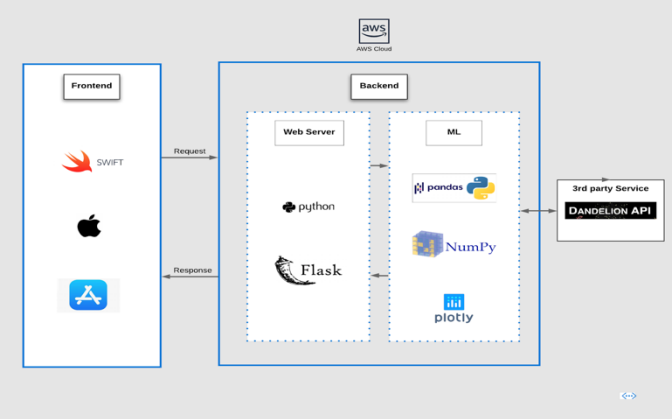

Fig. 5. Diagram showing the technology stack of the application

The architecture diagram demonstrates the technology stack of the application. The frontend is an iOS application that is created using Swift. The backend is divided into two parts web server and the machine learning model. The web server is built using Python in Flask and the server helps the UI communicate with the machine learning model which calculates the similarity and returns a sentiment and impact score for a particular news article. The backend is hosted on AWS and interacts with a third-party service Dandelion API, which is used to calculate the similarity score.

IV. Future Improvements

In the future, with enough demand, we could expose the news impact analytic service as a public API(SaaS) so that it could benefit both financial news portals and financial institutions. We could also add the service as a premium feature in the existing news applications. Once we have collected enough data and high-quality articles, we can launch an automated prediction service. This service would be able to automatically send push notifications to the users about the impact a news would have on the stock market, thus avoiding potential losses. Furthermore, as the data set grows, we can also improve the accuracy of the model for stock price prediction by collecting high-quality historical news articles.

V. Conclusion

The application helps the users to analyze the news impact by providing them an Impact Score, Sentiment, and the Historical Stock Movement chart which would help them make informed investment decisions. In conclusion, the BlackSwanImpactPredictor iOS app can help protect individuals' portfolios from market crashes by preparing investors against potential market swings based on global headlines.in the existing news applications. Once we have collected enough data and high-quality articles, we can launch an automated prediction service. This service would be able to automatically send push notifications to the users about the impact a news would have on the stock market, thus avoiding potential losses. Furthermore, as the data set grows, we can also improve the accuracy of the model for stock price prediction by collecting high-quality historical news articles.

VI. Acknowledgement

Thanks to Professor Rakesh Ranjan for his guidance and constructive feedback throughout the development of this project. His insights and exposure have enabled us to understand the software development process in a more practical and impactful way.

VII. Deliverables

A. Github Repository

BlackSwanImpactPredictor Repository

B. Presentation

BlackSwanImpactPredictor Presentation

C. API definition

API definition attributes

D. Link of Running Application

http://18.222.219.39:5000/api/process-model

References

[1] Taleb, N. N. (2007, April 22). *'The Black Swan: The Impact of the highly improbable'*. The New York Times. Retrieved December 8, 2021, from https://www.nytimes.com/2007/04/22/books/chapters/0422-1st-tale.html.

[2] https://www.nasdaq.com/market-activity/funds-and-etfs/spy/historical

[3] Khan, Wasiat & Ghazanfar, Mustansar ali & Azam, Muhammad Awais & Karami, Amin & Alyoubi, Khaled & Alfakeeh, Ahmed. (2020). Stock market prediction using machine learning classifiers and social media, news. Journal of Ambient Intelligence and Humanized Computing. 10.1007/s12652-020-01839-w.

[4] Usmani S, Shamsi JA. News sensitive stock market prediction: literature review and suggestions. PeerJ Comput Sci. 2021;7:e490. Published 2021 May 4. doi:10.7717/peerj-cs.490

[5] Yinghao Ren, Fangqing Liao, Yongjing Gong, Impact of News on the Trend of Stock Price Change: an Analysis based on the Deep Bidirectional LSTM Model, Procedia Computer Science, Volume 174, 2020

[6] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.