Data Design and Representation

Predicting stock price change from Financial News

Group 8

Mandy Gu

Willa Yu

Mayank Mani

Tejasvini Karunakarbabu

Contents

Executive Summary

The entire worlds stock exchanges have a capitalization of $85 trillion USD, trending up from $25 trillion in 2009 a 320% increase.[1] Stock prices, trends and growth are of keen interest for investors. The volatile nature of the stocks is a major hinderance to prediction. Estimated 93% of Mutual funds underperform in the market [1]. One major factor the influence stock is news and news analytics has gained traction since the advent of natural language processing.

In this project we collected financial news data form Nasdaq, and corresponding stock prices for a select few companies. We used a two-pronged approach in data collection. Firstly, we stored the webpages from historical archives of Nasdaq for extracting the news and article details and Secondly, we used an application programming interface (Alpha Vintage API) to obtain stock prices details for the companies.

In the news data web scraping we obtain information of headlines, the origin of the news, published time, article and the companies mentioned in the article. These are vital inputs to execute Natural Language Processing (NLP) algorithms. The news data is the cause and stock prices are the effect in our modelling, therefore we gathered information such as stock price points (opening, closing and so on), dividend and the corresponding company.

Given that the two data tables were highly structured we chose Structured Query Language (SQL) to communicate with the database. Since the data collection is more for the modelling perspective scalability or atomicity is not a major consideration. This data would be inputs for the machine learning algorithms our company would use to generate market intelligence.

Our unique value proposition is to help our clients manage the stock market volatility by dynamic portfolio management on a daily time period so as to maximize their return on investment.

Introduction

As a fin-tech company, we would like to provide our clients with insights on the impact of news on stocks. By focusing on ten key companies such as Facebook, Apple and Tesla, we predict if the stock price will increase or decrease. We can hedge risks for our clients by advising them to take appropriate measures based on our recommendations. In order to do the analysis, we need to collect the relevant data. This report solely looks at the data and data design aspects which are used as the input for the predictive model used to generate insights of monetary value.

Background:

There is a direct relation between news volume and stock price volatility. Press releases are found to be important events that represent a potential explanation for up to 24% of the major stock price movements [2]. News analytics is a growing field with several applications apart from stock price prediction such as automated detection of insider trading, circuit breakers to halt trading algorithms when news arrive, and stock screening to find interesting assets.

Data Source Introduction:

While News can be obtained through television, newspaper, social media and individual websites of financial news webpages. In this report the News is solely obtained from NASDAQ.

To be able to predict the stock prices based on news sentiment we need two key aspects as inputs.

- News headlines and articles: Source - Nasdaq
- The stock prices on the day after the news: Source - Alpha Vintage API

Web Scraped News Data:

For the sake of simplicity, we picked selected companies before making a generalized model.

We picked 10 of the most prominent stocks from different industries out of the Nasdaq Stock Market, which are: Apple, Amazon, Facebook, Google, Netflix, Adobe, Starbucks, Costco, Tesla, Expedia.

Since these companies are all from the Nasdaq stock market, naturally we select Nasdaq News & Insights session as our news material source: Nasdaq News & Insights session hosts an archive of historical news mostly from famous financial media, though most of the news there are uploaded on weekdays, few on weekends.

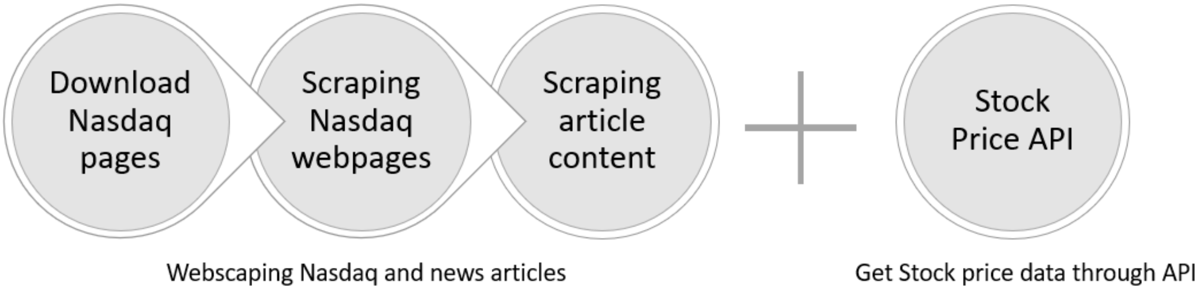
When searching for a particular result, each page will return 10 news headlines with the relative time that it was posted and the origin media if existed. There's also an option to choose how to order the news, in order to gain the news following the time-series order, we chose the 'Most Recent' tab.

To gain the articles for training the word to vector model, a common NLP technique in machine learning, we also need the news articles, which can be retrieved by clicking into the news headlines. Furthermore, under each article, there are tags containing other companies' symbols that are mentioned in this news. These tags indicated that the same news would appear in the search result of these companies. We collected this information as well since we believed that these tags could be a useful feature to deal with redundant news in the future.

To sum up, from the Nasdaq News & Insight session we collected the news headlines, relative publish time, news origins and the links directing to the news articles, from each news content page, we collected the article and the list of other companies mentioned in this article. In order to gain enough information for sentiment model building and stock price prediction while considering the workload, we planned to collect 500 headlines for each company.

Process flow of data collection

The process of scraping the news data can be broken down into two parts:



Part 1: News data

Download the search pages

Since the search pages of each company are constantly changing as the news is updated every day, it’s essential to download the search result pages before processing and scraping the elements we need. However, Nasdaq is using Ajax to generate their websites dynamically, normal HTTP requests cannot get the elements we want which are all generated with Javascript, so we need to use Selenium to simulate the browser behavior and get that information.

The search result website is fetched from the Nasdaq server with ‘GET’ request, which means that we can generate a list of URLs for the browser simulator to go through automatically. A standard URL link that directs to the first page of ‘apple’ search result, sorted by recency is like this: https://www.nasdaq.com/search?q=apple&page=1&sort_by=recent.

For every company, we just need to change the term following ’q=’ to the symbol or the full name of the company. Similarly, we change the term following ‘page=’ to loop through the search result pages. In each step of the loop, we create a text file to store the page contents using page_source function.

Apart from downloading the result pages to local, another important thing we did is to record the ‘current time’ when each result page is opened and downloaded. Because all the recent headlines only have relative publish time, like ‘2 minutes ago’, ‘1 day ago’, this is essential for calculating the published time of each news headline. We stored the ‘current time’ for each page in a CSV file for next step’s use.

Scrape the downloaded webpages

In each text file of the search result, we scraped the 10 headlines with their URLs, relative published time, and the news origins and stored that in a list. These elements and the CSS selectors we used to find them are as follows:

Elements	CSS selector
Headlines, News URL	find_all(class_ = 'search-result__link')
news origin	find_all(class_ = 'search-result__topic')
relative published time	find_all(class_ = 'search-result__date')

During the loop for reading all the text files and scraping these elements, we also calculated the actual published time for every news using the ‘day_calculator’ function we wrote, which could retain all the formats of returned from the ‘relative published time’ selector to standard ‘mm/dd/yyyy’ time style. Alongside, we attached the weekdays of the published date, so that we can put the difference between weekdays and weekends into consideration in the final model.

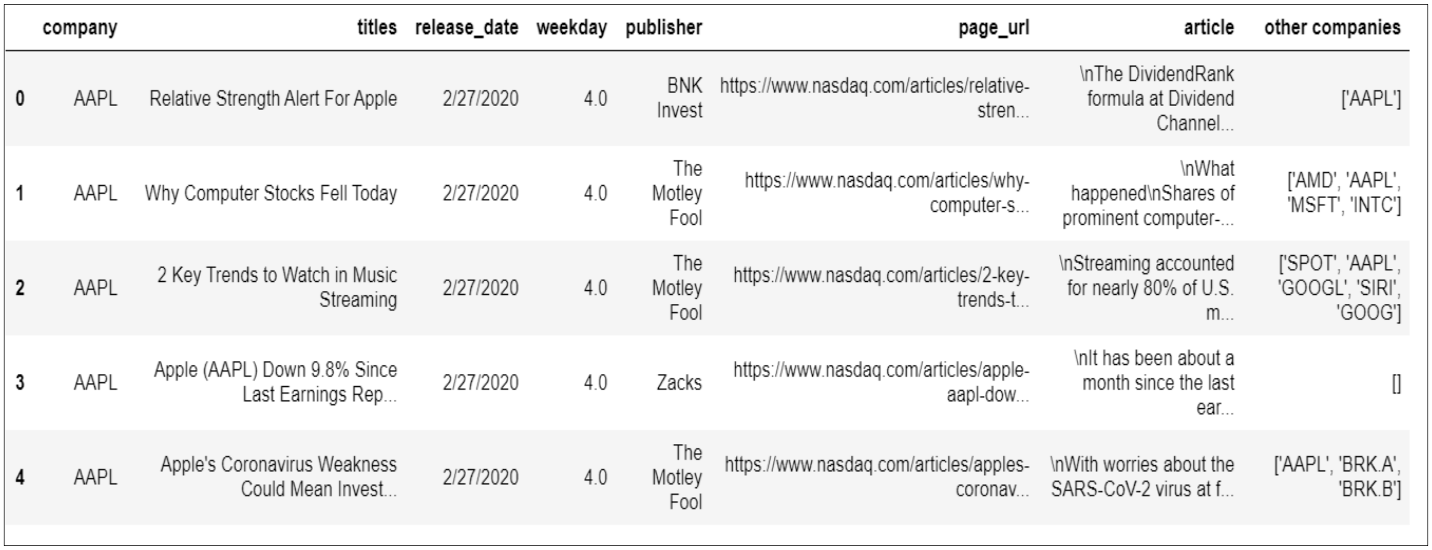
In this process, we also combined and stored the company name with these element lists into a dictionary and stored it in a file named ‘news_full_list.csv’.

Get and scrape the article and other mentioned companies in the news

Firstly, we read the ‘news_full_list.csv’ line by line, and use Selenium to go through the URL for each news. We used *“driver.find_elements_by_class_name('body__content')”* to scrape the articles and *“driver..find_elements_by_class_name(‘topics-in-this-story__symbol’)”* to scrape the other companies that are mentioned in this news.

In this part, we directly used Selenium to scrape without downloading the original webpages is because we believe that these 'historical' news webpages are less prone to change over time.

After collecting all the articles and 'other companies' for 5000 URLs, we appended the new data into a single table. The below figure shows the format of the data.

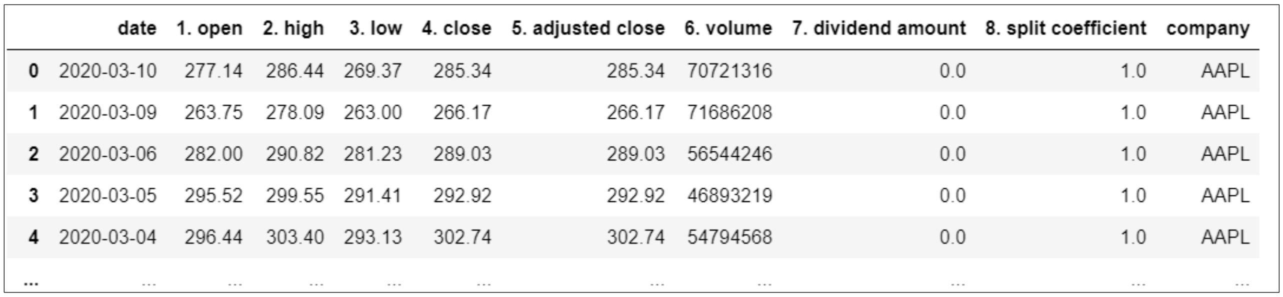


	company	titles	release_date	weekday	publisher	page_url	article	other companies
0	AAPL	Relative Strength Alert For Apple	2/27/2020	4.0	BNK Invest	https://www.nasdaq.com/articles/relative-stren...	\nThe DividendRank formula at Dividend Channel...	['AAPL']
1	AAPL	Why Computer Stocks Fell Today	2/27/2020	4.0	The Motley Fool	https://www.nasdaq.com/articles/why-computer-s...	\nWhat happened\nShares of prominent computer-...	['AMD', 'AAPL', 'MSFT', 'INTC']
2	AAPL	2 Key Trends to Watch in Music Streaming	2/27/2020	4.0	The Motley Fool	https://www.nasdaq.com/articles/2-key-trends-t...	\nStreaming accounted for nearly 80% of U.S. m...	['SPOT', 'AAPL', 'GOOGL', 'SIRI', 'GOOG']
3	AAPL	Apple (AAPL) Down 9.8% Since Last Earnings Rep...	2/27/2020	4.0	Zacks	https://www.nasdaq.com/articles/apple-aapl-dow...	\nIt has been about a month since the last ear...	[]
4	AAPL	Apple's Coronavirus Weakness Could Mean Invest...	2/27/2020	4.0	The Motley Fool	https://www.nasdaq.com/articles/apples-coronav...	\nWith worries about the SARS-CoV-2 virus at f...	['AAPL', 'BRK.A', 'BRK.B']

Part 2: Stock Price Data:

Apart from the original material for sentiment analysis, we also need to collect stock prices over time as the final model's dependent variable via financial APIs.

The API we are using is Alpha Vantage. Via their APIs, we can collect up to 100 business days' stock prices for each company and the market, and the data frame of stock prices is as follows:



	date	1. open	2. high	3. low	4. close	5. adjusted close	6. volume	7. dividend amount	8. split coefficient	company
0	2020-03-10	277.14	286.44	269.37	285.34	285.34	70721316	0.0	1.0	AAPL
1	2020-03-09	263.75	278.09	263.00	266.17	266.17	71686208	0.0	1.0	AAPL
2	2020-03-06	282.00	290.82	281.23	289.03	289.03	56544246	0.0	1.0	AAPL
3	2020-03-05	295.52	299.55	291.41	292.92	292.92	46893219	0.0	1.0	AAPL
4	2020-03-04	296.44	303.40	293.13	302.74	302.74	54794568	0.0	1.0	AAPL
...	...	...	...	...	...	...	...	...	...	...

The Data Design

The major datasets that need to be stored are the web-scraped news and stock prices from API. Since these two data sources are highly structured and related, we decide to use SQL to host these two data sources.

For the NASDAQ news database, the columns and the according data types are as follows:

Key	Column Name	Data Type	Nullable	Reason
CPK	company	VARCHAR (7)	No	The maximum length of company or symbol's name is 7
CPK	titles	VARCHAR (251)	No	The Primary Key's length must be assigned when created. The maximum length of title is 251
	release_date	DATE	Yes	Release_date follows date format
	day_of_week	INT	Yes	The range of day_of_week is [1,7]
	publisher	VARCHAR (20)	Yes	The maximum length of published doesn't exceed 20
CPK	page_url	VARCHAR (124)	No	The Primary Key's length must be assigned when created. The maximum length of page_url is 124
	article	LONGTEXT	Yes	The lengths of articles are too long to be stored in any VARCHAR
	other_company_1	VARCHAR (7)	Yes	The maximum length of company or symbol's name is 7
	other_company_2			
	other_company_3			
	other_company_4			

In order to make sure that every news is unique, we set company, title and page_url as the primary keys, and we dropped the redundant news that has the same title and appears in the same company's search result. We also spotted the original list of 'other companies' in the data frame, since the form of text list is hard to process in MySQL. Because the average number of ‘other companies’ is approximately 3.5, we kept only the first four ‘other companies’ in the database for data readability.

For the stock prices, the columns and the according data types are as follows:

Key	Column Name	Data Type	Nullable	Reason
CPK	date_of_price	DATE	No	Release_date follows MySQL's standard date format
	open_price	DECIMAL	Yes	Use decimal to store finite decimals are more precise
	high_price			
	low_price			
	close_price			
	adjusted_price			
	volume	INT	Yes	The range of the volume falls into the range of INT
	dividend_amount	DECIMAL	Yes	Use decimal to store finite decimals are more precise
	split_coef			
CPK	company	VARCHAR(7)	No	The maximum length of company or symbol's name is 7

We considered creating foreign keys for these two databases, however, the candidate for foreign keys can only be the ‘company’ and ‘date_of_price’ column. For the company column, stock prices database contains one more value of ‘NDAQ’ than in the news database, and the date column in news database is Nullable. So, we cannot create foreign keys here.

Summary and Conclusions

Our database will be an input for machine learning models which will use sophisticated machine learning algorithms using text analytics, semantic analytics, support vector machines and so on.

News analytics with the data we have collected can generate several solutions few of them are listed below.

- Predict if the stock price will increase or decrease with respect to market
- Predict the percentage change of increase and decrease with respect to the market
- Analyze the stock price fluctuation with respect to time since the news release
- Assign of relevance of news based on the impact on the stock prices

The objective of these solutions can create multiple business strategies such as return strategies, risk management and algorithmic trading.

References:

1) https://www.liberatedstocktrader.com/stock-market-statistics/

2) https://pdfs.semanticscholar.org/e659/16b906f4c5b001e2de551481fa955d3f0747.pdf