CIS 8395 BIG DATA ANALYTICS EXPERIENCE

STOCK PRICE PREDICTION - WALLSTREETBETS

TEAM NEXUS:
ANIKET MAWLANKAR
DIBYASHA MAHAPATRA
DEEPIKA ASHOKKUMAR
YASH KHAIRNAR

Table of Contents

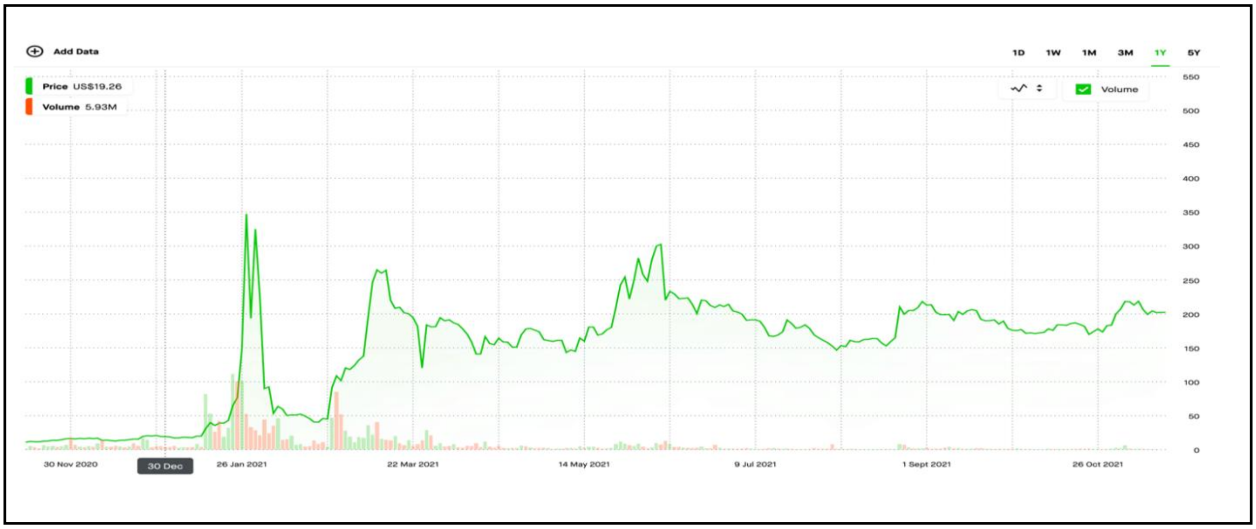
INTRODUCTION AND PURPOSE

Stock movement prediction has been a challenging problem since its inception. Humans have tried to tame the bull throughout history but have never been successful. Stock price prediction is complex because there are too many factors at play, and creating models to consider such variances is almost impossible. However, with the latest advances in Machine Learning and Augmented Intelligence, we now have the ability to process huge amounts of data. With such amounts of data, we can perform predictive and prescriptive analytics, giving us insights leading to solutions and, therefore, actions and recommendations.

Stock price prediction has been a topic of interest among investors and researchers. Among challenges faced in predicting stock, movement in the market is highly stochastic, and we plan on making temporally dependent predictions from chaotic data. We also plan on demonstrating the state-of-the-art performance of our proposed model on a stock movement prediction dataset which we collected from Reddit and Yahoo Finance.

We present a model to predict stock price movement from Reddit's comments and historical stock prices - hourly based. Our primary focus is on meme stock like GameStop Corp. (GME).

The rise of meme stocks has shifted the investor sentiment outside the traditional investing strategies. The rage over meme stocks overtook the market in early 2021, with investors piling into companies gaining recognition over Reddit forums like Wall Street Bets. Since then, meme stocks have only gained momentum.



The story of GameStop as a meme stock began in the third week of January 2021. In just over a week, the stock gained over 2500 percent of its value and tanked again to the former level. One interesting fact about this volatility around GameStop is the emergence of individual retail investors as the backbone of fueling GME with the use of social media. Never have we seen before the trend of Individual Retail Investors outsmarting hedge fund investors at their own game.

There are several reasons associated with this unconventional trend. Among them is the abundance of liquidity in the market due to extra capital from stimulus checks, ease of accessibility to stock investing platforms like Robinhood, Vanguard, Webull etc., prices of these stocks were low and familiarity of stocks like GameStop Corp. In addition to that, social media played a significant role in fueling the momentum by driving attention to these speculative names and bringing meme movement to the retail audience.

With information readily available on the internet and social media, the accessibility and spread of meme stock happened so much faster across geographic boundaries where people, who aren't from the finance and investing domain, get pulled in because of the various trends on social media. As a result, there are more individual retail investors participating in the market for making quick returns, with most of them being novice investors.

Meme Stock

Meme stock is a stock that captures only attention, usually from a younger generation of retail investors on online social media platforms such as Reddit and ends up going viral. Most of these investors do not follow the traditional practices of performing fundamental and technical analyses of the stock before investing. These investors depend on online forums like Reddit and end up going viral once the meme stock gains enough attention, thereby gaining momentum. As a result, most of the individual retail investors end up investing without following the company's security intrinsic value, which is derived from earnings growth, earnings per share, price-to-earnings ratios and dividend yield that leads to the stock price not possibly stable as it would suggest.

Unconventional Trading Practices

With the rise of individual retail investors, it has been observed that the rules of trading are changing.

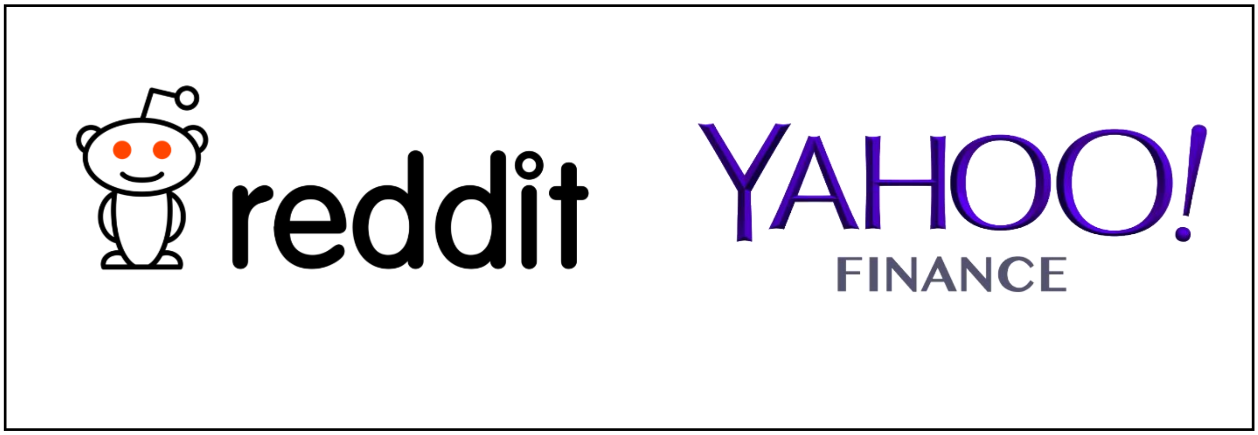
With the meme stock rage, individual retail investors came piling up in the market and soon became a force to be reckoned with. The traditional Hedge Fund investors were caught off guard and were beaten at their own game. Thereby, it is safe to say that the monopoly of the traditional investors, especially the hedge fund investors, has been pierced with the rise of individual retail investors. Individual retail investors now constitute around 25 percent of the total options trading according to the latest estimates.

It has been observed that stocks like GameStop were not traded based on their fundamentals. This led to a chain reaction, where more and more individual retail investors invested in these stocks without doing their homework by directly going against the traditional practices.

The objective is to predict the highly volatile stock – GME by analyzing the sentiment of the comments related to GameStop on the Reddit subreddit – Wall Street Bets combining it with the technical analysis with respect to the historical stock data on an hourly basis.

DATA SOURCING

Each day, various members of wallstreetbet subreddit post different submissions regarding the meme stocks like GME, AMC etc. and people discuss these stocks in the comments of these submissions. So, in order to perform the sentiment analysis among the reddit community, we needed to extract and analyze all these comments from the post. This is where our first data source comes in, that is, wallstreetbet subreddit. For this project, we've decided to focus on one particular stock GME, and try to predict its stock price based on all the comments specific to posts related to GME in the past one year as that's the time period where GME achieved popularity among the wallstreetbet community.

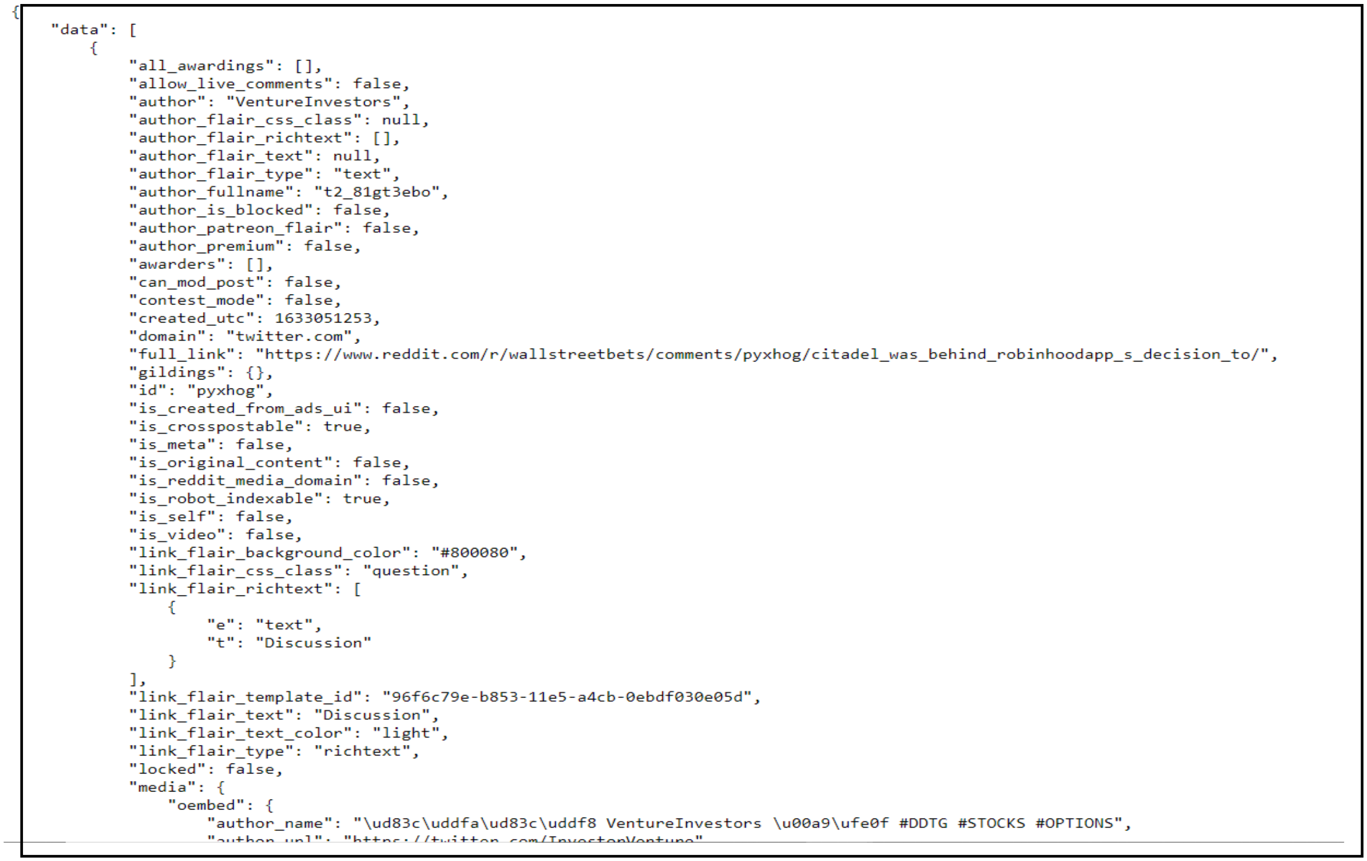


Data Sources

The second source of our dataset is Yahoo finance where we extracted stock price details for GME stock for past one year. We used Databricks platform for data extraction as it is one of the leading cloud-based data engineering tools used for processing and transforming massive quantities of data. It also supports Python language and it's easy to integrate with AWS as well.

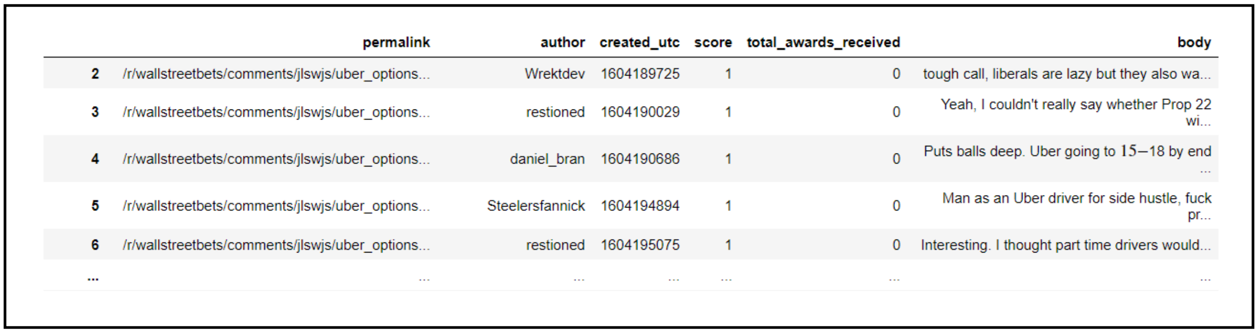
We explored different options that could help us extract this huge dataset from Reddit. There was an R package called RedditExtractoR which uses Reddit API to fetch the posts and comments, but it had limitations of getting data for past one month alone. Therefore, that was out of the option. Then there was PRAW, Python Reddit API Wrapper. Even this package had its limitations as there was no option to input date parameter for the time-period we wanted to extract the data for. Then, we came across pushshift.io Reddit API. This package provided us with exactly what we needed. We were able to input queries like time period of past 1 year, posts containing GME ticker and even filter based on subreddit. There were a couple of challenges though. First was, we had to convert the time period to Unix time. The other was we were limited to just 1000 posts per api request. So, we ended up creating a loop where the code would fetch 1000 posts each time and append it for the last one year. We eventually we managed to extracts close to 1 million comments for the past 1 year alone.

Raw json data from reddit api



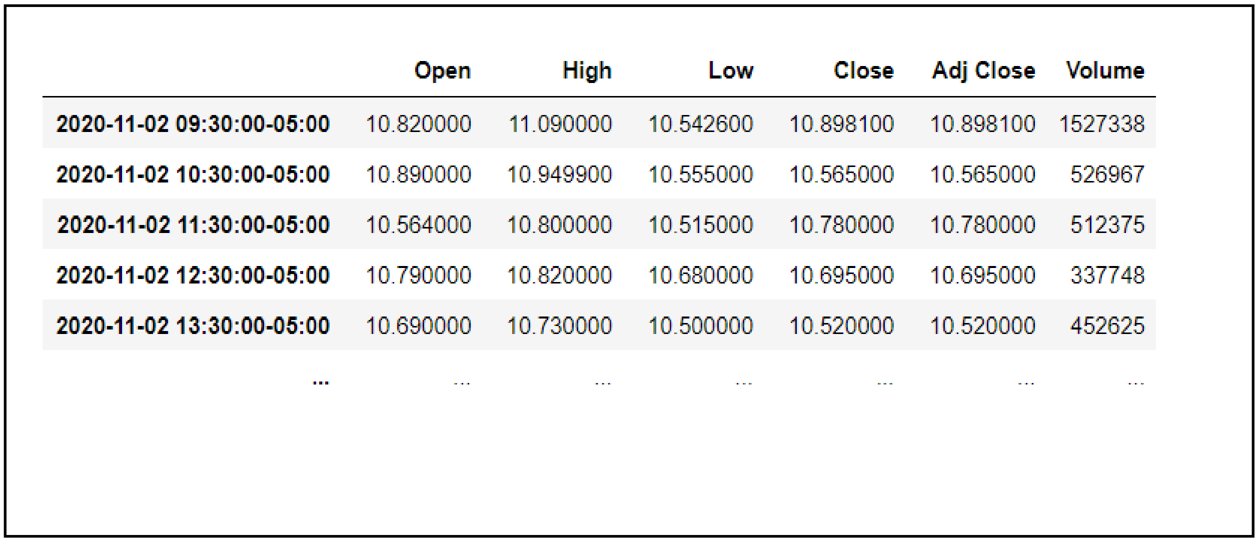
```
{
  "data": [
      {
          "all_awardings": [],
          "allow_live_comments": false,
          "author": "VentureInvestors",
          "author_flair_css_class": null,
          "author_flair_richtext": [],
          "author_flair_text": null,
          "author_flair_type": "text",
          "author_fullname": "t2_81gt3ebo",
          "author_is_blocked": false,
          "author_patreon_flair": false,
          "author_premium": false,
          "awarders": [],
          "can_mod_post": false,
          "contest_mode": false,
          "created_utc": 1633051253,
          "domain": "twitter.com",
          "full_link": "https://www.reddit.com/r/wallstreetbets/comments/pyxhog/citadel_was_behind_robinhoodapp_s_decision_to/",
          "gildings": {},
          "id": "pyxhog",
          "is_created_from_ads_ui": false,
          "is_crosspostable": true,
          "is_meta": false,
          "is_original_content": false,
          "is_reddit_media_domain": false,
          "is_robot_indexable": true,
          "is_self": false,
          "is_video": false,
          "link_flair_background_color": "#800080",
          "link_flair_css_class": "question",
          "link_flair_richtext": [
              {
                  "e": "text",
                  "t": "Discussion"
              }
          ],
          "link_flair_template_id": "96f6c79e-b853-11e5-a4cb-0ebdf030e05d",
          "link_flair_text": "Discussion",
          "link_flair_text_color": "light",
          "link_flair_type": "richtext",
          "locked": false,
          "media": {
              "oembed": {
                  "author_name": "\ud83c\uddfa\ud83c\uddf8 VentureInvestors \u00a9\ufe0f #DDTG #STOCKS #OPTIONS",
                  "author_url": "https://twitter.com/InvestorVenture"
```

json package in Python.



	permalink	author	created_utc	score	total_awards_received	body
2	/r/wallstreetbets/comments/jlswjs/uber_options...	Wrektdev	1604189725	1	0	tough call, liberals are lazy but they also wa...
3	/r/wallstreetbets/comments/jlswjs/uber_options...	restioned	1604190029	1	0	Yeah, I couldn't really say whether Prop 22 wi...
4	/r/wallstreetbets/comments/jlswjs/uber_options...	daniel_bran	1604190686	1	0	Puts balls deep. Uber going to 15−18 by end ...
5	/r/wallstreetbets/comments/jlswjs/uber_options...	Steelersfannick	1604194894	1	0	Man as an Uber driver for side hustle, fuck pr...
6	/r/wallstreetbets/comments/jlswjs/uber_options...	restioned	1604195075	1	0	Interesting. I thought part time drivers would...
...	...	...	...	...	...	...

Reddit data after extracting relevant features from json



	Open	High	Low	Close	Adj Close	Volume
2020-11-02 09:30:00-05:00	10.820000	11.090000	10.542600	10.898100	10.898100	1527338
2020-11-02 10:30:00-05:00	10.890000	10.949900	10.555000	10.565000	10.565000	526967
2020-11-02 11:30:00-05:00	10.564000	10.800000	10.515000	10.780000	10.780000	512375
2020-11-02 12:30:00-05:00	10.790000	10.820000	10.680000	10.695000	10.695000	337748
2020-11-02 13:30:00-05:00	10.690000	10.730000	10.500000	10.520000	10.520000	452625
...	...	...	...	...	...	...

Data from Yahoo Finance - Hourly stock price for past 1 year

For Yahoo finance, we used the yfinance package for Python and passed the relevant parameters to get the data for past one year as shown above. This SP500 index price data contains a couple key columns including open price (OPEN), close price (CLOSE), max price of the day (HIGH), min price of the day (LOW), volume of the day (VOLUME).

Quick Summary:

Reddit Data:

- Features: permalink, author, created_utc, score, total_awards_received, body
- Time range: Nov 1, 2020 – Oct 31, 2021
- Number of comments: 834,237
- Extracted using pushshift.io Reddit API on Python

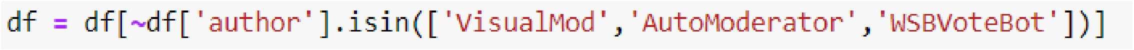
Yahoo Finance Data:

- Features: Open, High, Low, Close, Adj Close, Volume
- 1,751 rows of hourly stock price values
- Extracted using yfinance package on Python

DATA CLEANING

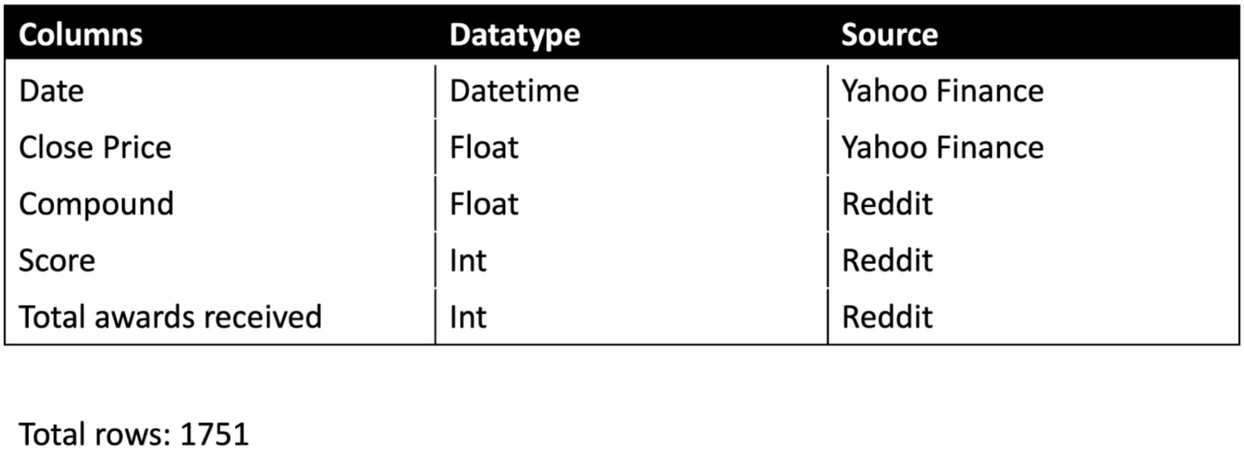
The data that we have extracted needed some cleaning as follows:

- The dataset extracted from reddit also contained comments posted by BOTs. So, we filtered out those comments based on the author names
- The reddit dataset contains some comments where the body was 'deleted' as some of the comments might have been deleted. We filtered out those comments based on the content in 'body' column



```
df = df[~df['author'].isin(['VisualMod','AutoModerator','WSBVoteBot'])]
```

- For Yahoo finance dataset, we know that entries for weekends would be missing as market is closed on the weekends. So, we just used the closing price on Friday to set the stock price for Saturdays and Sundays.



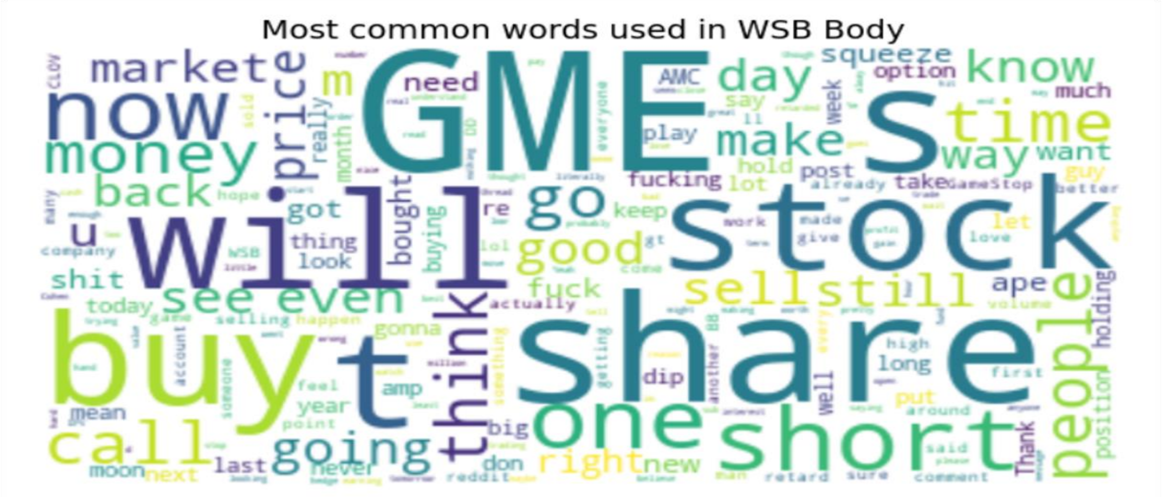
Columns	Datatype	Source
Date	Datetime	Yahoo Finance
Close Price	Float	Yahoo Finance
Compound	Float	Reddit
Score	Int	Reddit
Total awards received	Int	Reddit

Total rows: 1751

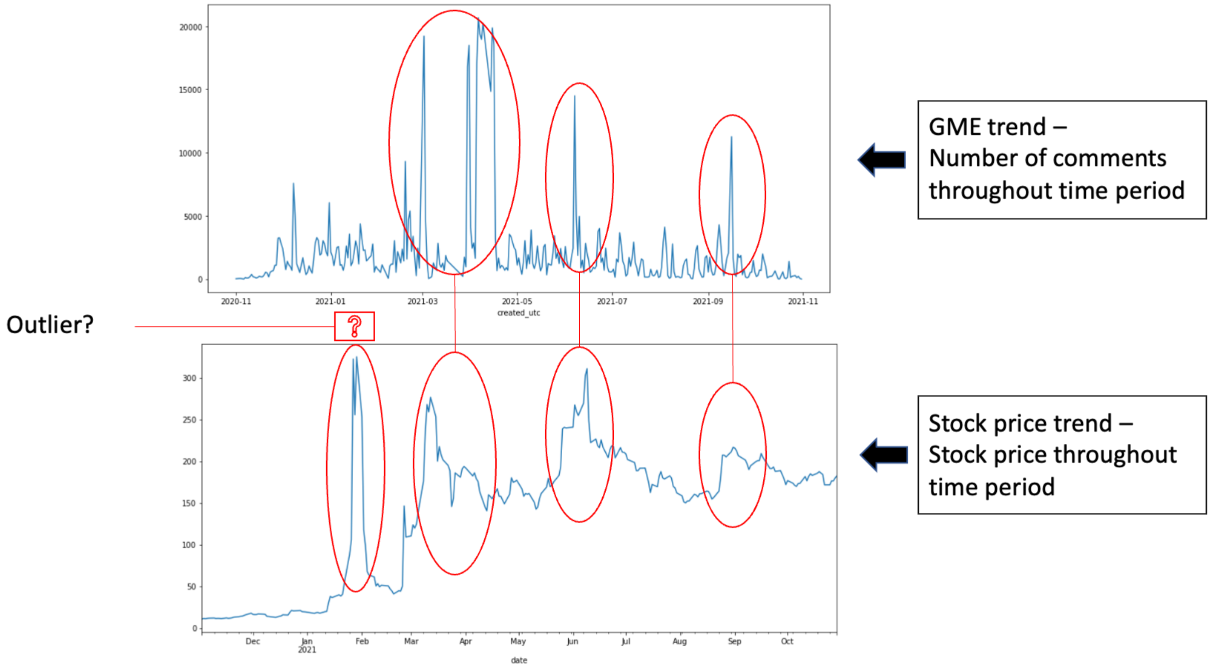
DATA EXPLORATION

Word Cloud

The magnitude of each word represents its frequency or relevance in a word cloud, which is a data visualisation tool for visualising text data.A word cloud can be used to emphasise important textual data points.A word cloud is a basic but effective visual representation object for text processing that displays the most frequently used words in larger, bolder letters and with varied colours. The lesser the importance of a term, the smaller it is.Stopwords are words that have no meaning, such as 'is,' 'are,' 'an,' 'I,' and many others.Wordcloud has a built-in stop-word library that was automatically remove stop words from the text.However, an intriguing aspect of this is that we can use the stopwords.add() function in Python to add our own stop words.Word clouds, also known as tag clouds, are graphical representations of word frequency that give words that appear more frequently in a source text more emphasis. The larger a word becomes in the text, the more frequently it appears. You can determine the large terms and thus the top topics just by glancing at the cloud.



Trend In Reddit Vs Actual Stock Price



GME trend – Number of comments throughout time period

Outlier?

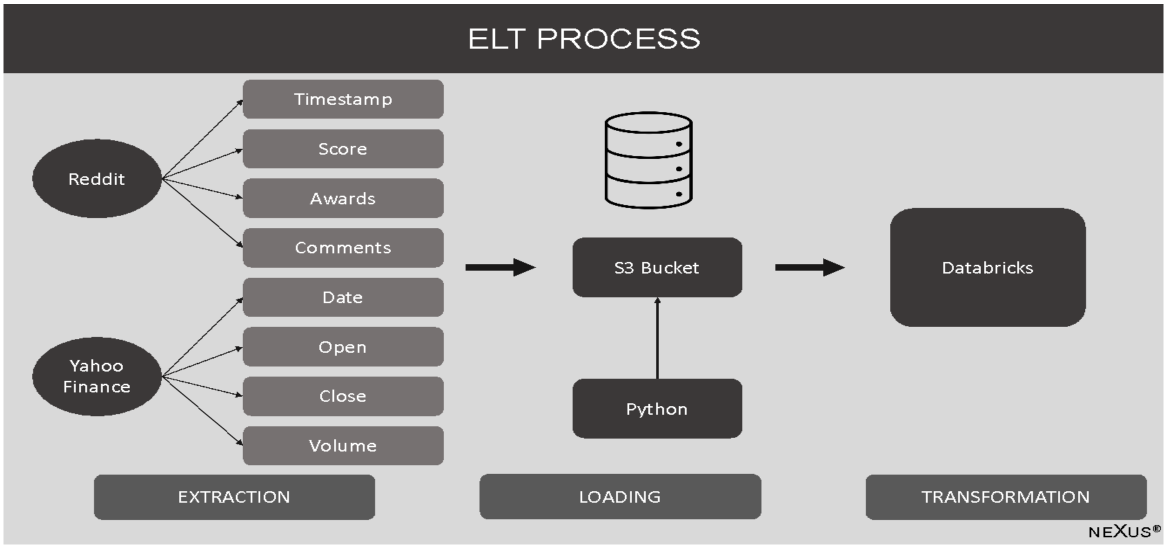
Stock price trend – Stock price throughout time period

EXTRACTION LOADING TRANSFORMATION PROCESS (ELT)

Extraction

For extracting the data from sources like Reddit and Yahoo Finance, we will be using Python.The objective is to understand the sentiment of the Individual Retail Investors on Reddit's subreddit named Wall Street Bets. Wall Street Bets is a community on Reddit with around 11 million subscribers. This community – subreddit was at forefront in fueling the momentum for meme stock volatility. GameStop is one such stock that is highly discussed on Wall Street Bets with around 964,367 comments in a span of a year. We extracted these comments with the use of a Reddit API wrapper named *pushshift.io.*

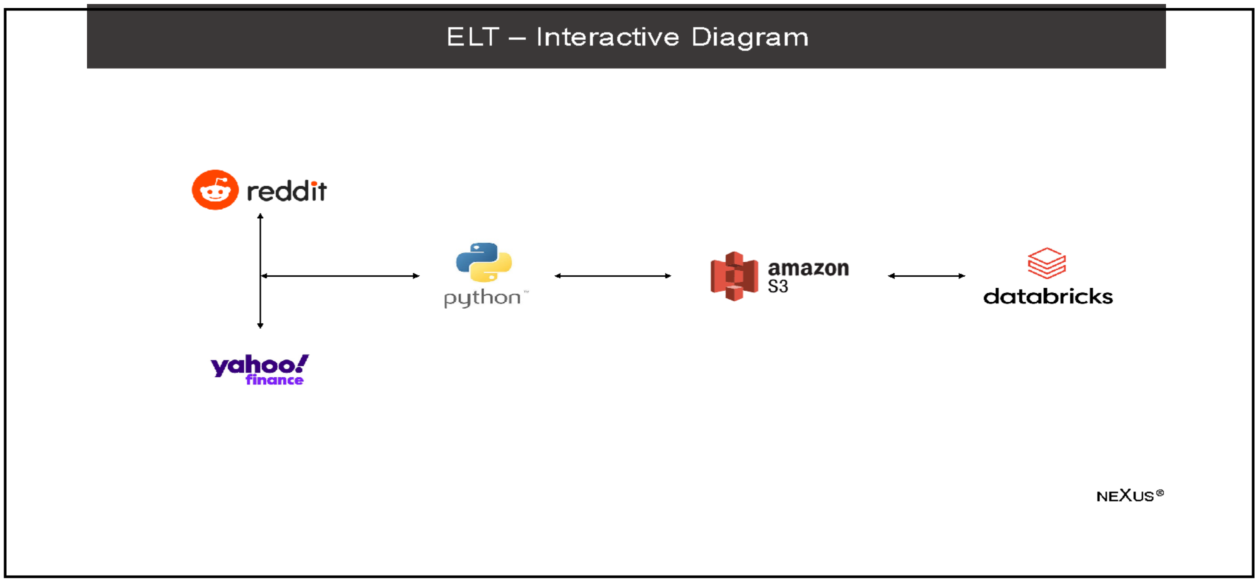
For extracting the historical stock data on an hourly basis, we extracted it using the *yfinance* package using Python. We extracted around 1751 rows.



Loading

For loading the data extracted from Reddit and Yahoo Finance Package, we used the AWS Software Development Kit (SDK) package in python to load the data on the S3 bucket. The objective was to S3 API to load it into one cluster.

We loaded the data on S3 bucket by pulling it from the Reddit API pushshift.io and Yahoo Finance package.



ELT – Interactive Diagram

Transformation

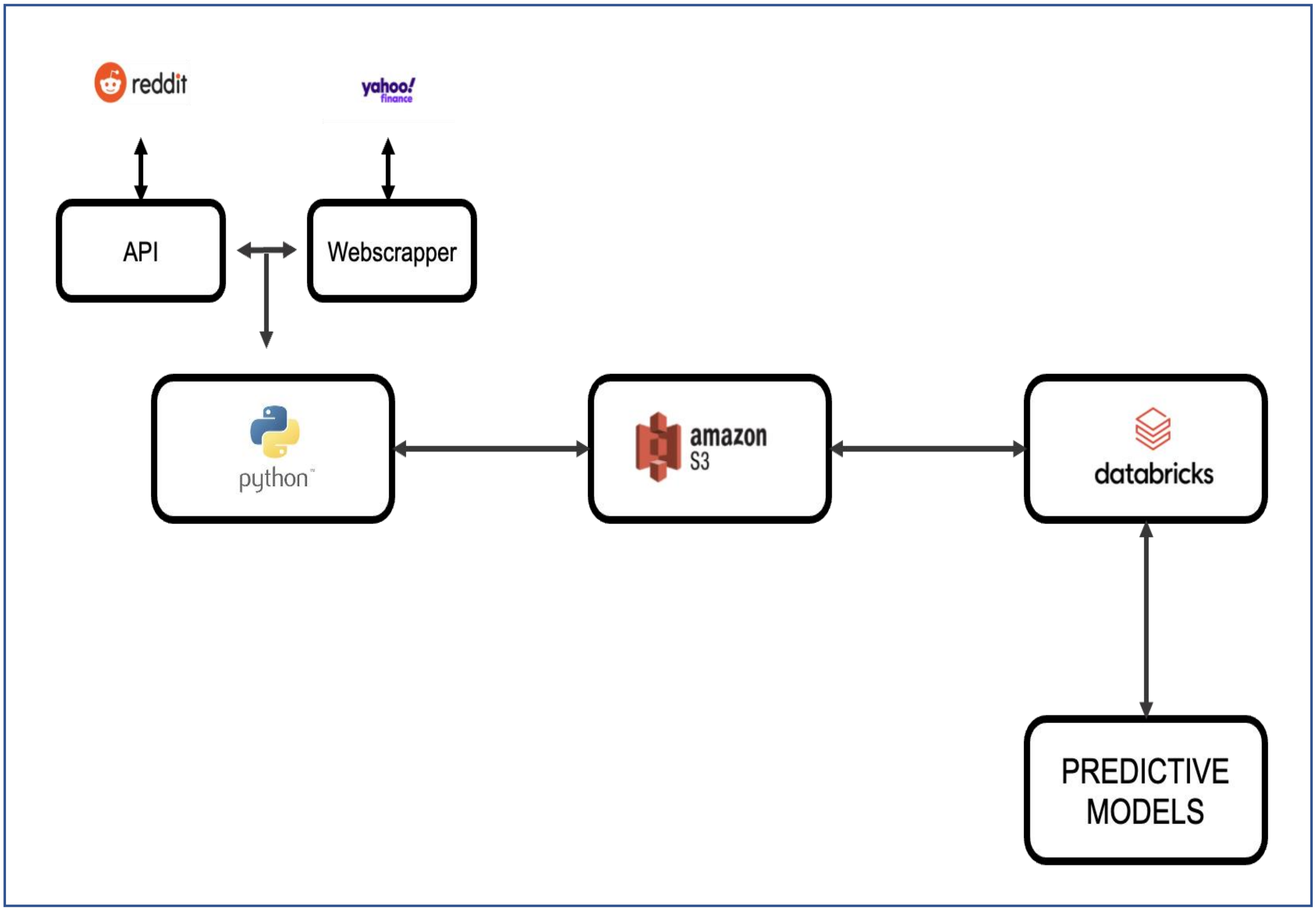
Once the data is uploaded on the S3 bucket, we structured the data in the form of tables. From Reddit we formed the Awards, Timestamp, Score, Comments as columns in a single table and from Yahoo Finance's – historical stock data we formed the Date, Open, Close and Volume as columns in a single table. We formed the relationship between Timestamp and Date that being a primary key.

The challenges came while cleaning the data, for cleaning the data extracted from Reddit, we used stop words. And when dealing with the data on the historical stock data, we dealt with a lot of null values. These null values were mostly weekends and national holidays as the stock market is closed on those days. We eliminated the null values by considering the close price of Friday as the open price on Monday.

All these operations were performed on Databricks.

We cleaned the Reddit comments by using stop words. We could not comb through each of the million comments so we just used a library from the word cloud package.The comments made by the bots were as well removed and any body comments that were deleted or removed were filtered out as well.

TECHNICAL ARCHITECTURE



The architecture has four main parts which connect each other-

- Data Sourcing: Reddit and YahooFinance Package
- Data Storage: AWS Services / Platform (Amazon S3)
- Platform: DataBricks
- Predictive Models

Data has been extracted mainly from two sources:

- Reddit API
- Yahoo Finance Package

DATA STORAGE

When designing an application's architecture, data storage and retrieval are critical concerns. Smaller organizations or applications can benefit from using third-party services like AWS. It reduces the requirement for physical infrastructure while also providing capabilities such as instant scaling in the case of a spike in server load. We looked at a variety of data storage choices and decided on AWS S3 after assessing the benefits and drawbacks of various storage and data retrieval processing options.

Amazon S3

The most popular and widely used storage in AWS is S3, also known as Simple Storage Service. S3 is a worldwide service that may be accessed using the AWS Management panel, a user-friendly online interface. To acquire a deeper understanding, we must examine a few key principles in greater depth. The data is saved as objects within buckets on Amazon S3. A file or metadata about a file can be used to create an object. There are a few steps to storing an item in S3, and you can establish rights for both the bucket and the file when uploading a file. Buckets are used to store the items.

Benefits of using AWS S3 on Databricks:

- Scalability: S3's availability and durability are considerably superior to HDFS, thanks to cross-AZ replication, which automatically replicates across various data centers.

- The Schema: We can infer the schema or explicitly set a pre-defined schema if we have one while establishing a data frame.

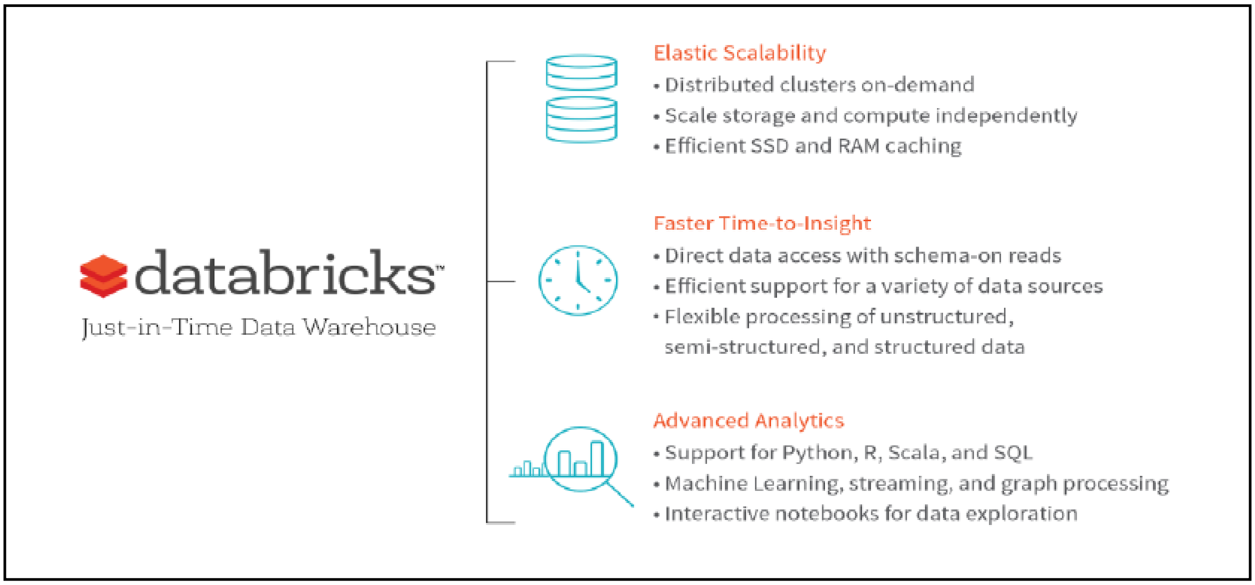
- Safety and security: It is vital that the data is kept secure so IAM roles were used to designate which cluster can access which buckets because keys can appear in logs and table metadata and are thus inherently insecure data.

- Execution: To ensure performance, the scalability and stability of hardware, software, and the network are all checked. S3's low-cost storage facility outperforms EFS/HDFS.

Databricks

Databricks is a platform that enables building, training, and deploying deep learning (DL) models at scale simple. Databricks Runtime ML, a machine learning runtime that provides a ready-to-use environment for machine learning and data science, includes a number of deep learning libraries. Databricks is a web-based data warehousing and machine learning platform developed by the Spark founders. It's a one-stop shop for all data needs, from data storage to data analysis and insights with SparkSQL, building predictive models with SparkML, and active connectivity to visualization tools like PowerBI, Tableau, and Qlikview.

The data engineering team is in charge of various ETL to ensure that data is sourced, cleansed, and quality tested before being stored in data warehouses. ETLs are more efficient, saving time and giving stakeholders a competitive advantage. Databricks is straightforward to integrate with Amazon Web Services, Microsoft Azure, and Google Cloud Platform.



databricks™

Just-in-Time Data Warehouse

Elastic Scalability

- Distributed clusters on-demand
- Scale storage and compute independently
- Efficient SSD and RAM caching

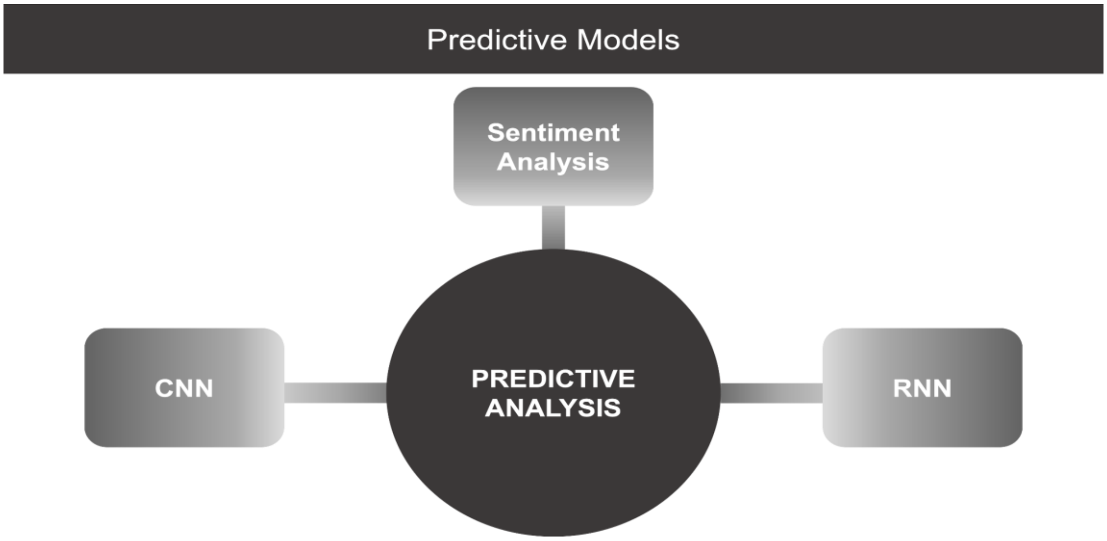
Faster Time-to-Insight

- Direct data access with schema-on reads
- Efficient support for a variety of data sources
- Flexible processing of unstructured, semi-structured, and structured data

Advanced Analytics

- Support for Python, R, Scala, and SQL
- Machine Learning, streaming, and graph processing
- Interactive notebooks for data exploration

PREDICTIVE MODELS



Convolutional neural networks (CNN) and recurrent neural networks (RNN) are two types of neural networks that can be used to solve problems.
(RNN) are utilized to determine which network is the most suitable for training. CNN is a multi-layered, connected feed-forward network that accepts and produces fixed-size inputs. It excels when data is gathered on a regular basis to generalize local trends. On the other hand, RNN can manage any input and output size, as well as connect prior data to the current one. It shines in circumstances involving a series of time series data.
Lengthy short-term memory (LSTM) is a particular RNN that uses numerous interacting layers to allow the algorithm to recall information for long periods of time. This might be useful for predicting stock moves that are comparable.

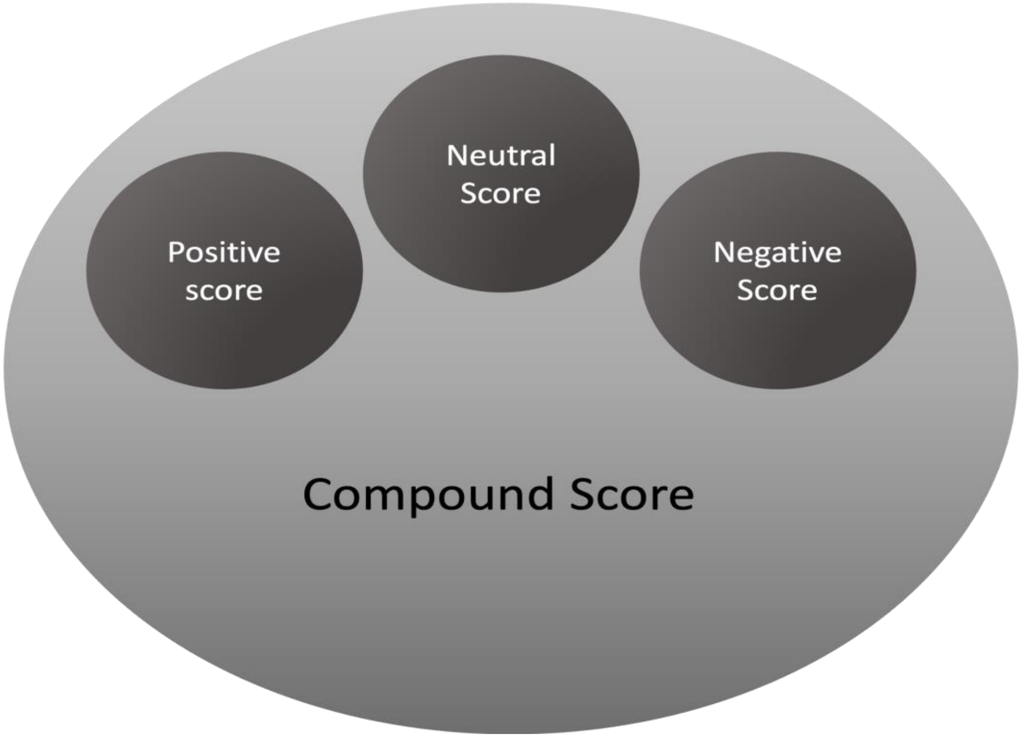
SENTIMENT ANALYSIS

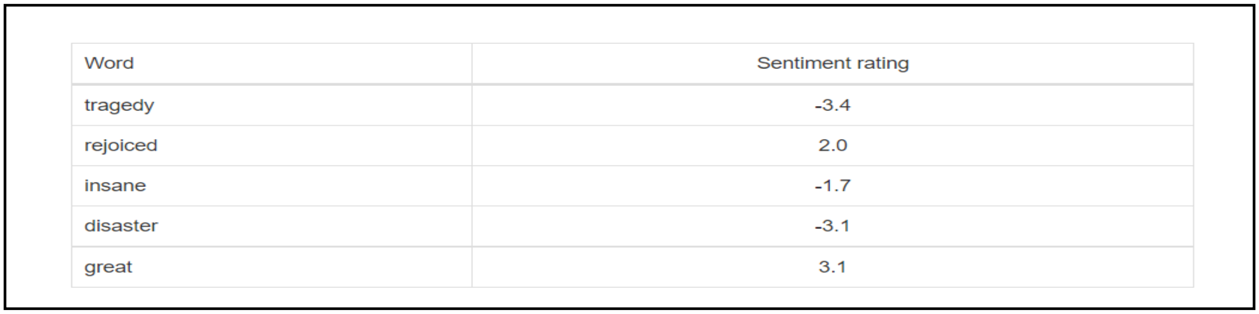
Sentiment analysis is a text analysis technique that finds polarity i.e. a positive or negative opinion in a text. Sentiment analysis tries to quantify a writer's attitude, sentiments, and emotions using a computational approach of subjectivity in a text.

VADER : *Valence Aware Dictionary for Sentiment Reasoning*

It's a text sentiment analysis model that's sensitive to both emotion polarity (positive/negative) and intensity. It's included in the NLTK package and may be used on unlabeled text data. It uses a lexicon to connect lexical information to emotion intensities, which are referred to as sentiment scores. The sentiment score of a text can be estimated by adding the intensity of each word. Words like 'love,' 'enjoy,' 'glad,' and 'like,' for example, all express a happy attitude. VADER is also smart enough to recognize the underlying meaning of certain words, such as "did not love" as a negative statement. It also recognizes the importance of capitalization and punctuation, such as in the phrase "ENJOY." Below is an excerpt from VADER's vocabulary, with more positive words receiving positive evaluations and more adverse words, negative evaluations.

VADER produces four sentiment measurements from these word grading, which you can see underneath. The initial three, +ve, neutral, and -ve, address the extent of the content that falls into those classifications. The compound score is the total amount of the lexicon grades which have been normalized to run between – 1 and 1.





Word	Sentiment rating
tragedy	-3.4
rejoiced	2.0
insane	-1.7
disaster	-3.1
great	3.1

What is a compound score and how is it calculated?

The compound score is the sum of positive, negative & neutral scores which is then normalized between **-1(most extreme negative)** and **+1 (most extreme positive)**. These scores are calculated based on the Valence scores for the words.

What is a Valence Score?

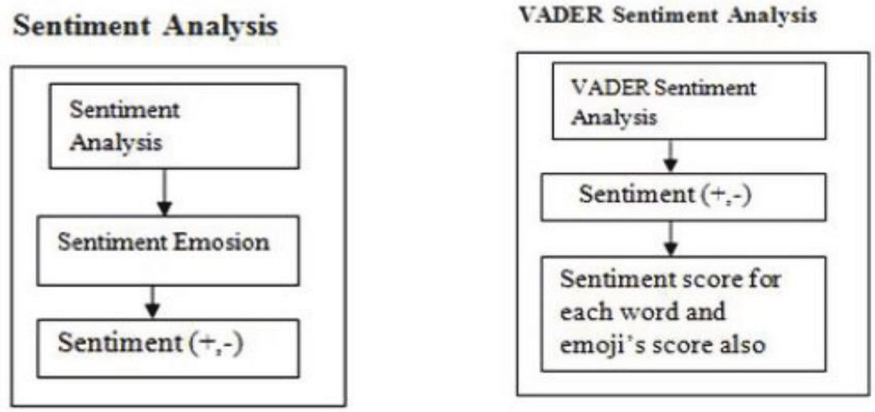
It is a score assigned to the word under consideration by means of observation and experiences rather than pure logic.

Example:

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

sentences = ["VADER is smart, handsome, and funny.",  # positive sentence example

            "VADER is smart, handsome, and funny!",  # punctuation emphasis handled
correctly (sentiment intensity adjusted)

            "VADER is very smart, handsome, and funny.", # booster words handled
correctly (sentiment intensity adjusted)

            "VADER is VERY SMART, handsome, and FUNNY.",  # emphasis for
ALLCAPS handled

            "VADER is VERY SMART, handsome, and FUNNY!!!", # combination of
signals - VADER appropriately adjusts intensity

            "VADER is VERY SMART, uber handsome, and FRIGGIN FUNNY!!!", #
booster words & punctuation make this close to ceiling for score

            "VADER is not smart, handsome, nor funny.",  # negation sentence example

            "The book was good.",  # positive sentence
```

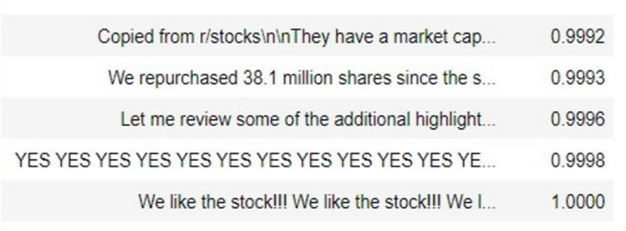
Advantages of using VADER*:*

- It does not require any training data.
- It can very well understand the sentiment of a text containing emoticons, slangs, conjunctions, capital words, punctuations and much more.
- It works well on social media text.
- It works with multiple domains



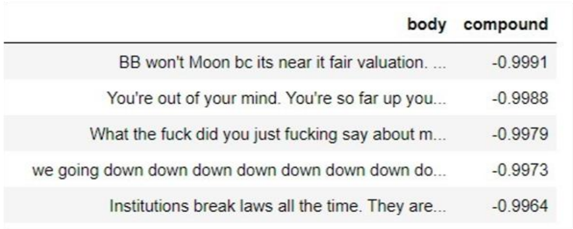
We used the Sentiment Intensity Analyzer library of Vader in order to analyze the sentiments of the Reddit user comments.We applied it to the body comments to obtain scores.We obtained a dictionary of positive,negative,neutral and compound scores.The dictionary was converted to the panda series and then assigned to the columns.The compound scores oscillate between -1 and +1.

Positive Sentiment Examples

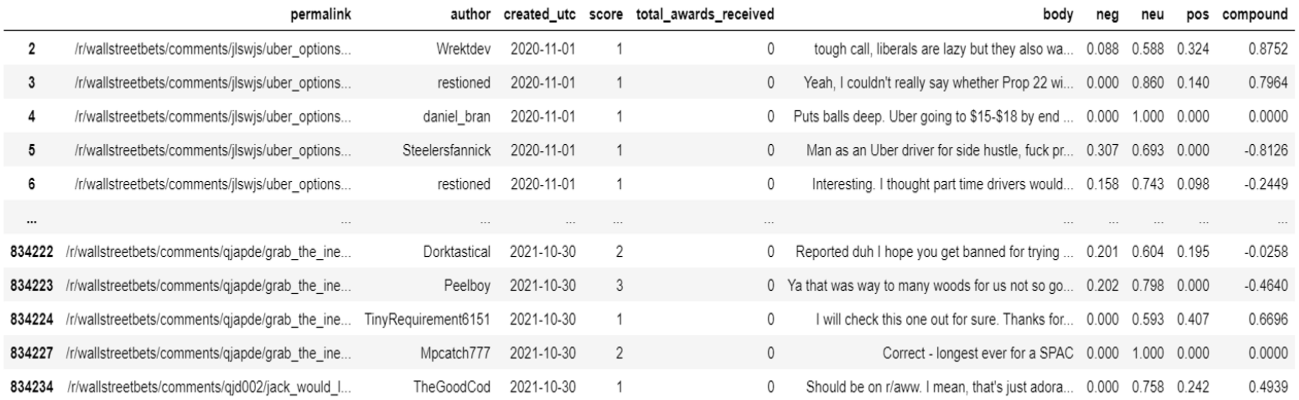


Copied from r/stocks\n\nThey have a market cap...	0.9992
We repurchased 38.1 million shares since the s...	0.9993
Let me review some of the additional highlight...	0.9996
YES YES YES YES YES YES YES YES YES YES YES YE...	0.9998
We like the stock!!! We like the stock!!! We l...	1.0000

Negative Sentiment Examples

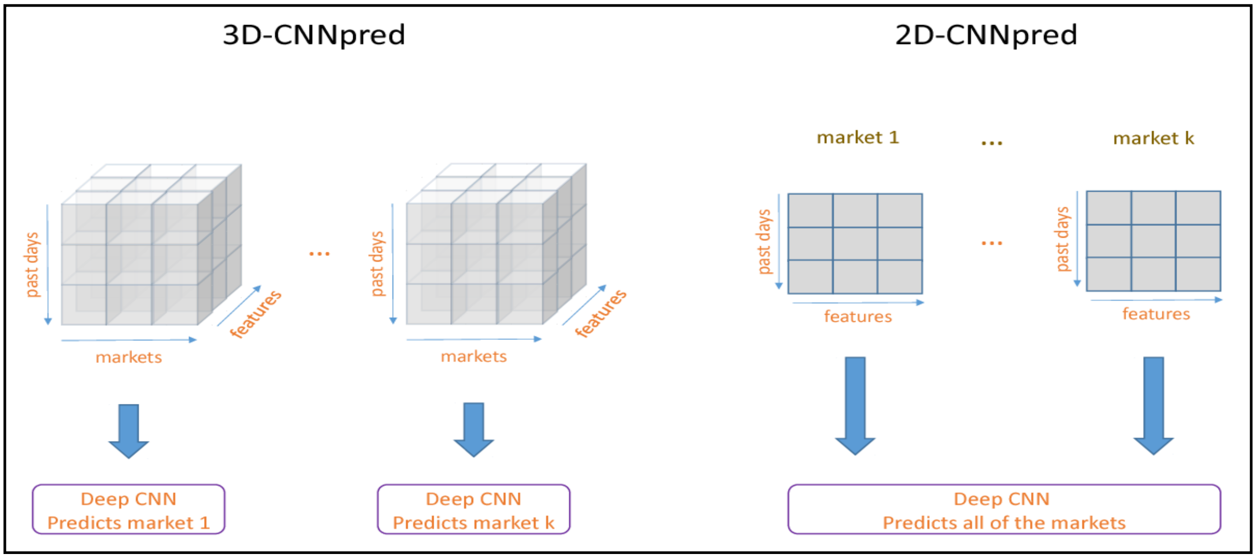


body	compound
BB won't Moon bc its near it fair valuation. ...	-0.9991
You're out of your mind. You're so far up you...	-0.9988
What the fuck did you just fucking say about m...	-0.9979
we going down down down down down down down do...	-0.9973
Institutions break laws all the time. They are...	-0.9964



	permalink	author	created_utc	score	total_awards_received	body	neg	neu	pos	compound
2	/r/wallstreetbets/comments/jlswjs/uber_options...	Wrektdev	2020-11-01	1	0	tough call, liberals are lazy but they also wa...	0.088	0.588	0.324	0.8752
3	/r/wallstreetbets/comments/jlswjs/uber_options...	restioned	2020-11-01	1	0	Yeah, I couldn't really say whether Prop 22 wi...	0.000	0.860	0.140	0.7964
4	/r/wallstreetbets/comments/jlswjs/uber_options...	daniel_bran	2020-11-01	1	0	Puts balls deep. Uber going to $15-$18 by end ...	0.000	1.000	0.000	0.0000
5	/r/wallstreetbets/comments/jlswjs/uber_options...	Steelersfannick	2020-11-01	1	0	Man as an Uber driver for side hustle, fuck pr...	0.307	0.693	0.000	-0.8126
6	/r/wallstreetbets/comments/jlswjs/uber_options...	restioned	2020-11-01	1	0	Interesting. I thought part time drivers would...	0.158	0.743	0.098	-0.2449
...	...	...	...	...	...	...	...	...	...	...
834222	/r/wallstreetbets/comments/qjapde/grab_the_ine...	Dorktastical	2021-10-30	2	0	Reported duh I hope you get banned for trying ...	0.201	0.604	0.195	-0.0258
834223	/r/wallstreetbets/comments/qjapde/grab_the_ine...	Peelboy	2021-10-30	3	0	Ya that was way to many woods for us not so go...	0.202	0.798	0.000	-0.4640
834224	/r/wallstreetbets/comments/qjapde/grab_the_ine...	TinyRequirement6151	2021-10-30	1	0	I will check this one out for sure. Thanks for...	0.000	0.593	0.407	0.6696
834227	/r/wallstreetbets/comments/qjapde/grab_the_ine...	Mpcatch777	2021-10-30	2	0	Correct - longest ever for a SPAC	0.000	1.000	0.000	0.0000
834234	/r/wallstreetbets/comments/qjd002/jack_would_l...	TheGoodCod	2021-10-30	1	0	Should be on r/aww. I mean, that's just adora...	0.000	0.758	0.242	0.4939

CONVOLUTION NEURAL NETWORK



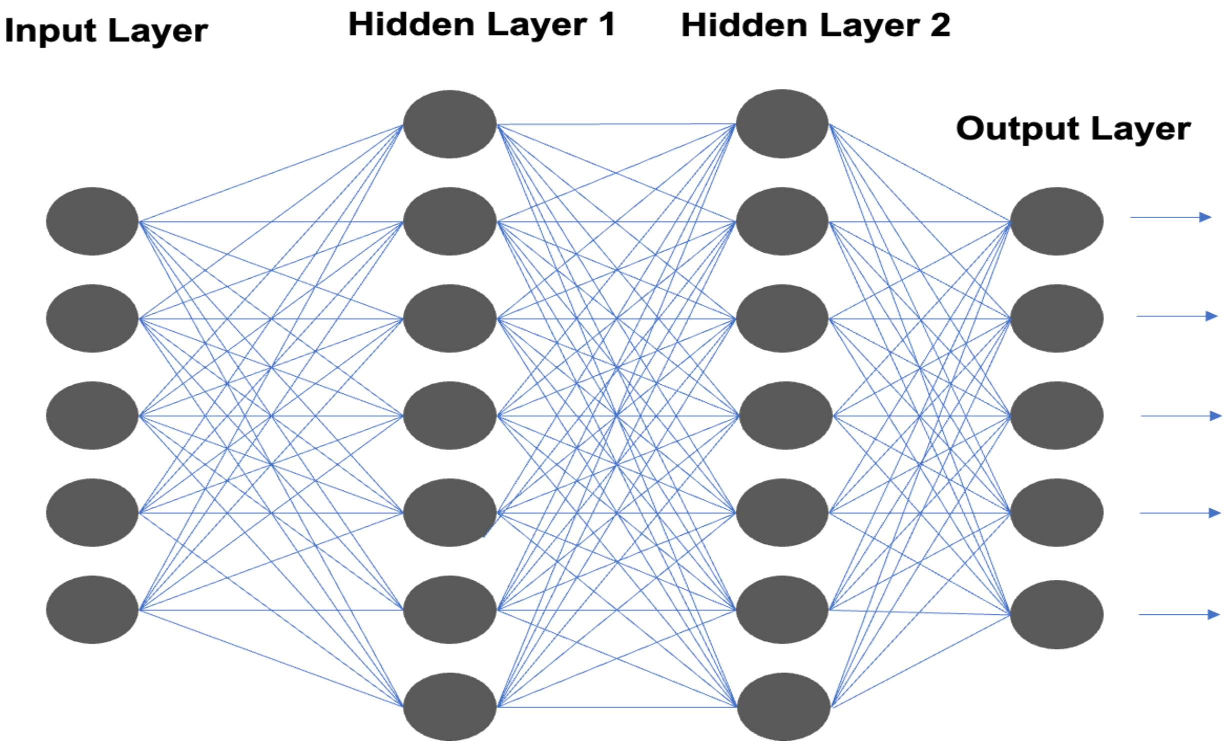
An input layer, an output layer, and hidden layers are all present in CNNs. Convolutional layers, ReLU layers, pooling layers, and completely linked layers are among the hidden layers.

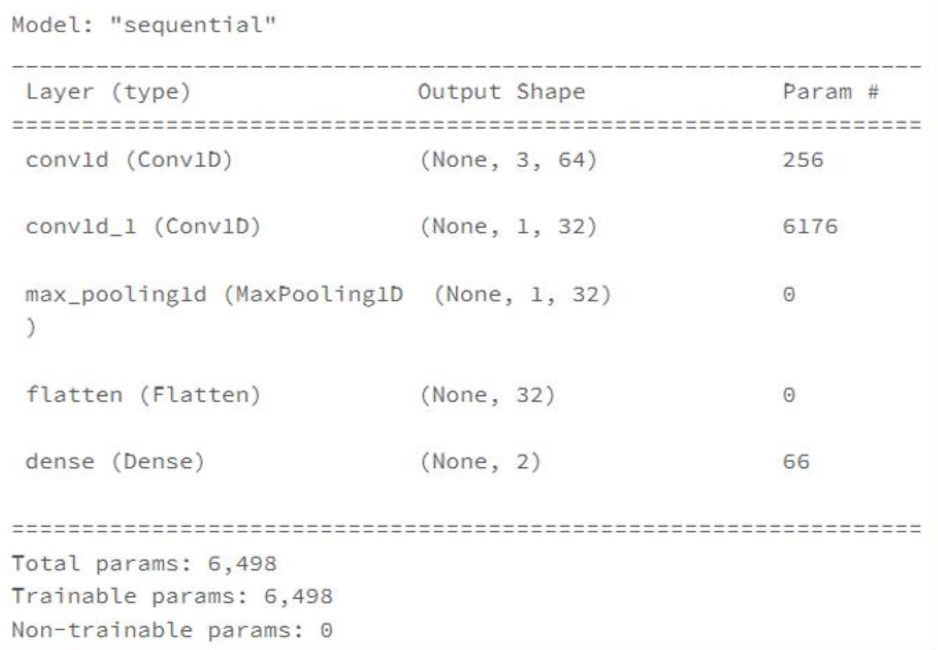
Convolutional Neural Networks (CNNs) are a type of Neural Network that has shown to be particularly useful in fields like image identification and categorization. Apart from powering vision in robots and self-driving cars, Convolutional Neural Networks have been successful at recognizing faces, objects, and traffic signs. The more convolution stages we have, the more complex features our network will be able to learn to recognize in the original dataset.

CNN is a sort of Deep Neural Network (DNN) that is specifically designed for image processing. A neural network takes in data, computes required values in a hidden layer using randomized parameters, and outputs a prediction or classification. The term "deep neural networks" refers to neural networks that have more than one hidden layer. DNN is a catch-all term for all neural network topologies, such as CNN and RNN (Recurrent Neural Network).

A CNN architecture is divided into two components.

- In a process known as Feature Extraction, a convolution tool isolates and identifies the distinct characteristics of an image for analysis.
- A fully connected layer that uses the output of the convolution process to forecast the image's class using the information acquired in previous stages.





```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 3, 64)             256

 conv1d_1 (Conv1D)           (None, 1, 32)             6176

 max_pooling1d (MaxPooling1D  (None, 1, 32)            0
 )

 flatten (Flatten)           (None, 32)                0

 dense (Dense)               (None, 2)                 66

=================================================================
Total params: 6,498
Trainable params: 6,498
Non-trainable params: 0
```

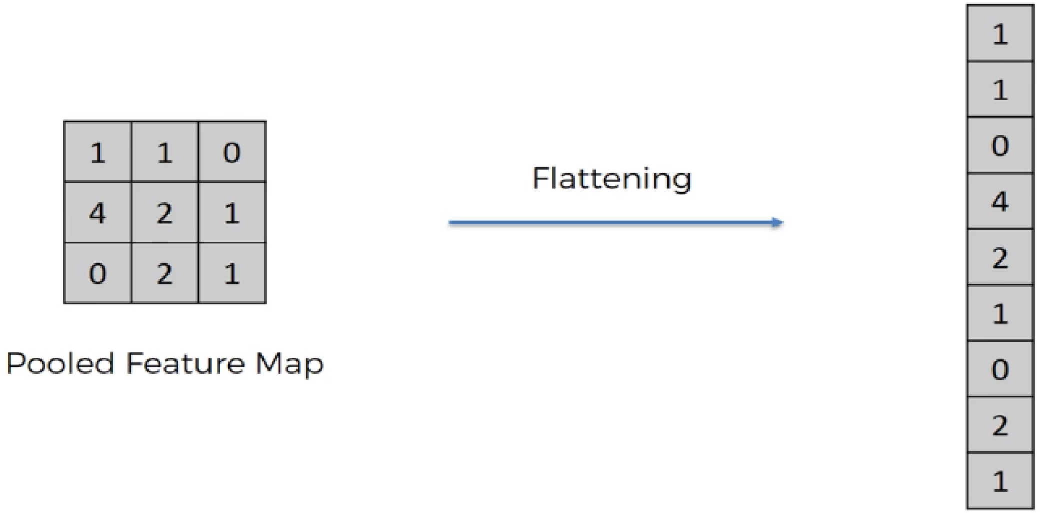
1. Convolutional Layer

This is the initial layer that extracts the different features from the input photos. The convolution mathematical operation is done between the input image and a filter of a specific size MxM in this layer. The dot product between the filter and the sections of the input image with regard to the size of the filter is taken by sliding the filter across the input image (MxM).The Feature map is the result, and it contains information about the image such as its corners and edges. This feature map is then supplied to further layers, which learn a variety of other features from the input image.A Pooling Layer is usually applied after a Convolutional Layer. This layer's major goal is to lower the size of the convolved feature map in order to reduce computational expenses.This is accomplished by reducing the connections between layers and operating independently on each feature map. There are numerous sorts of Pooling operations, depending on the mechanism used.

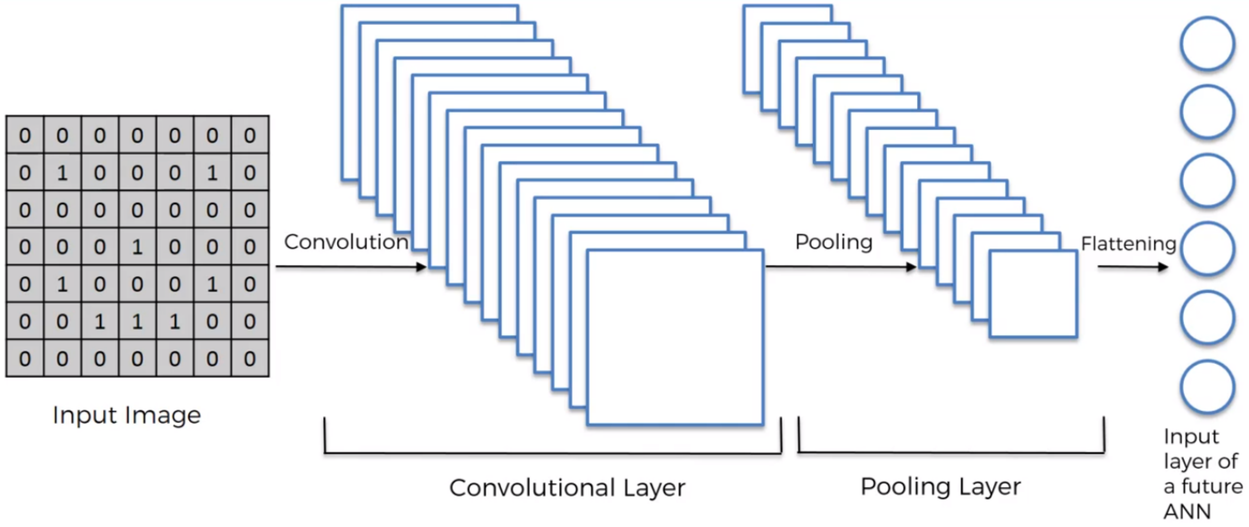
2. Max Pooling

The largest element is obtained from the feature map in Max Pooling. The average of the elements in a predefined sized Image segment is calculated using Average Pooling. Sum Pooling calculates the total sum of the components in the predefined section. Between the Convolutional Layer and the FC Layer, the Pooling Layer normally acts as a link.

3.Flatten Layer



1	1	0
4	2	1
0	2	1

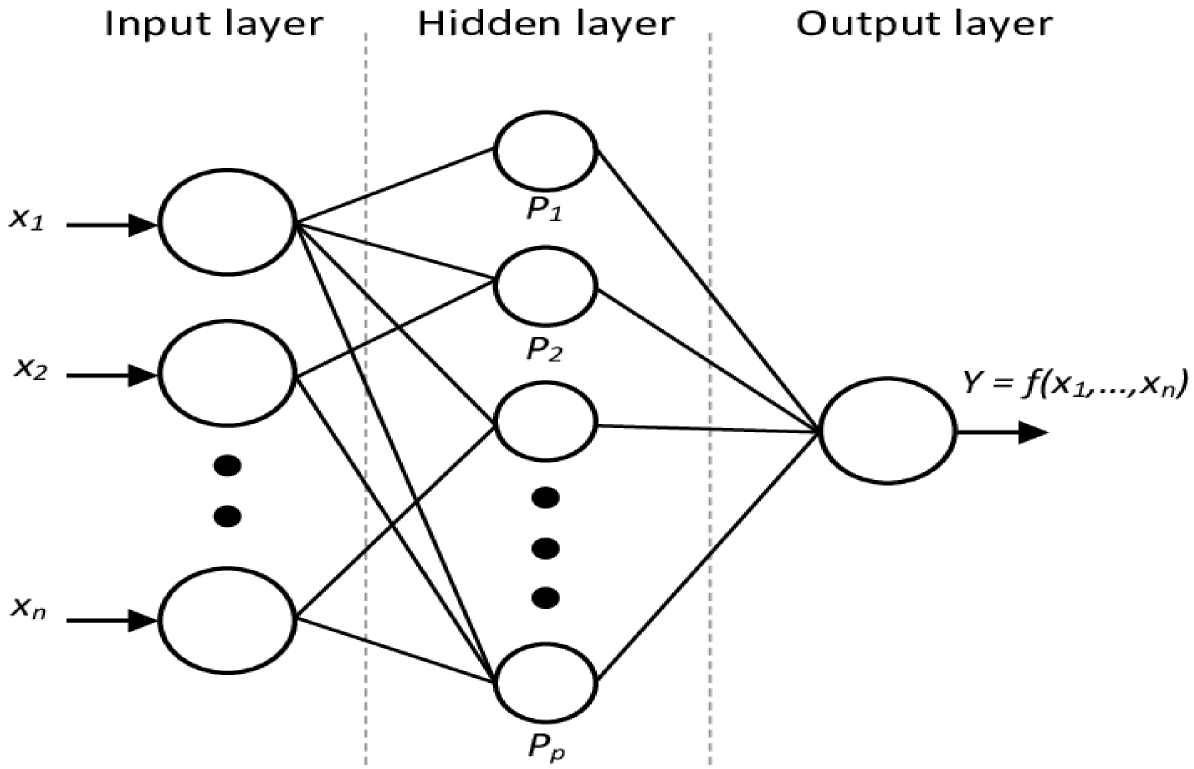


0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

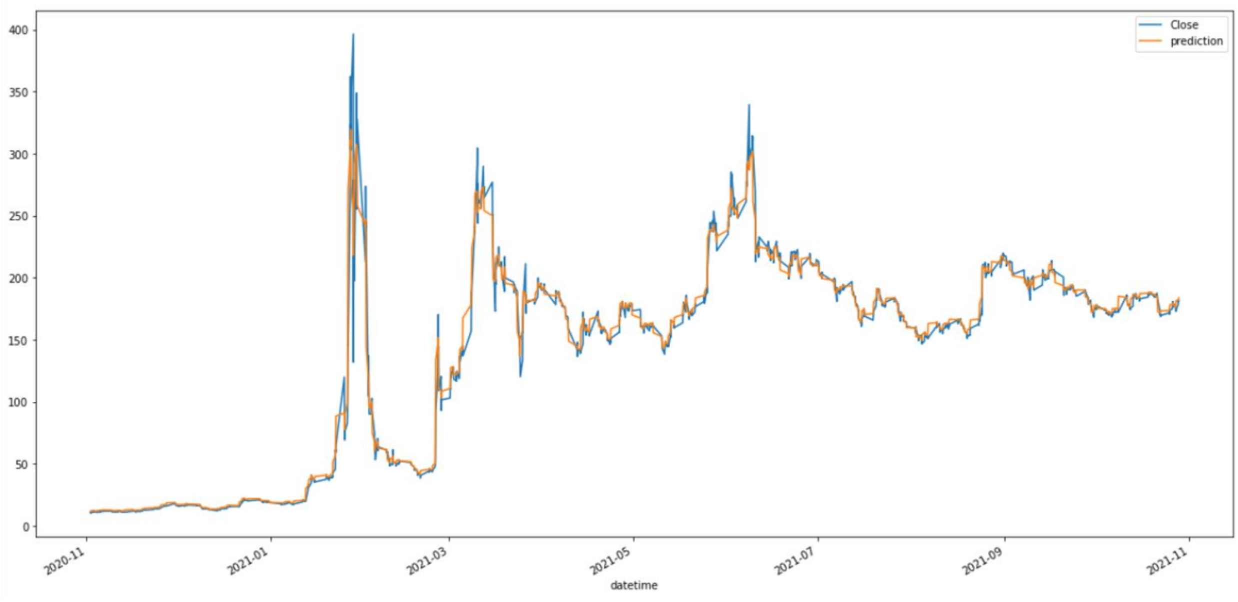
We're going to flatten it out and make a column out of it. Simply take the numbers one by one and place them in this single lengthy column.The goal is to later feed this information into an artificial neural network for processing.When you have a lot of pooling layers, or pooling layers with a lot of pooled feature maps, you flatten them. As a result, you place them one by one in this single lengthy column. For an artificial neural network, you get a single large vector of inputs.So, to summarise, we now have an input image. We apply a convolution layer, then pooling, and finally flatten everything into a large vector that will be our artificial neural network's input layer.

4. Dense Layer

A dense layer in a neural network is one that is deeply connected to the layer before it, meaning that the layer's neurons are connected to every neuron in the layer before it. In artificial neural network networks, this layer is the most widely utilised layer.In a model, the dense layer's neuron receives input from every neuron in the preceding layer, and the dense layer's neurons conduct matrix-vector multiplication.

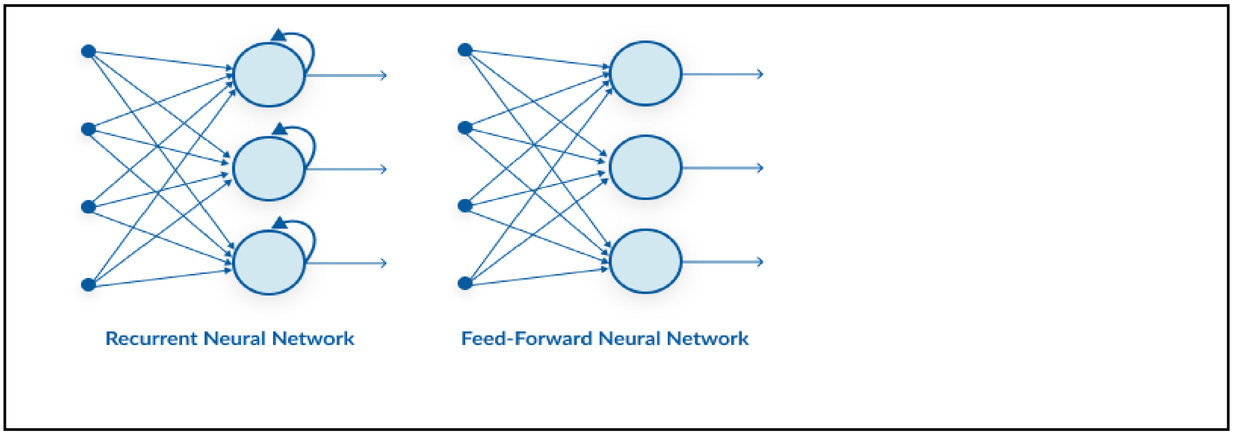


CNN Model prediction vs. actual close price on current data

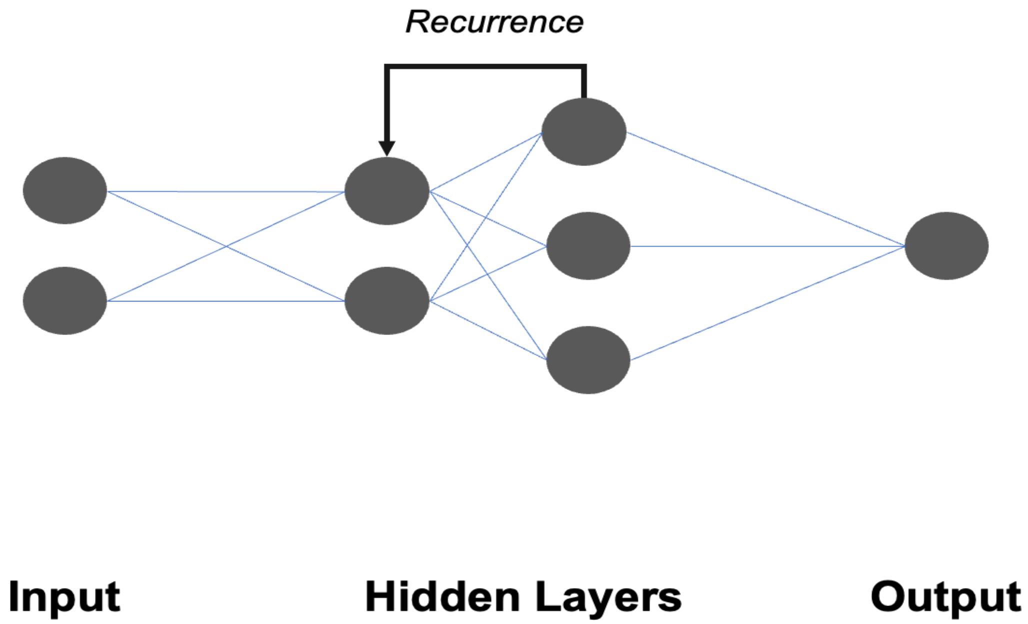


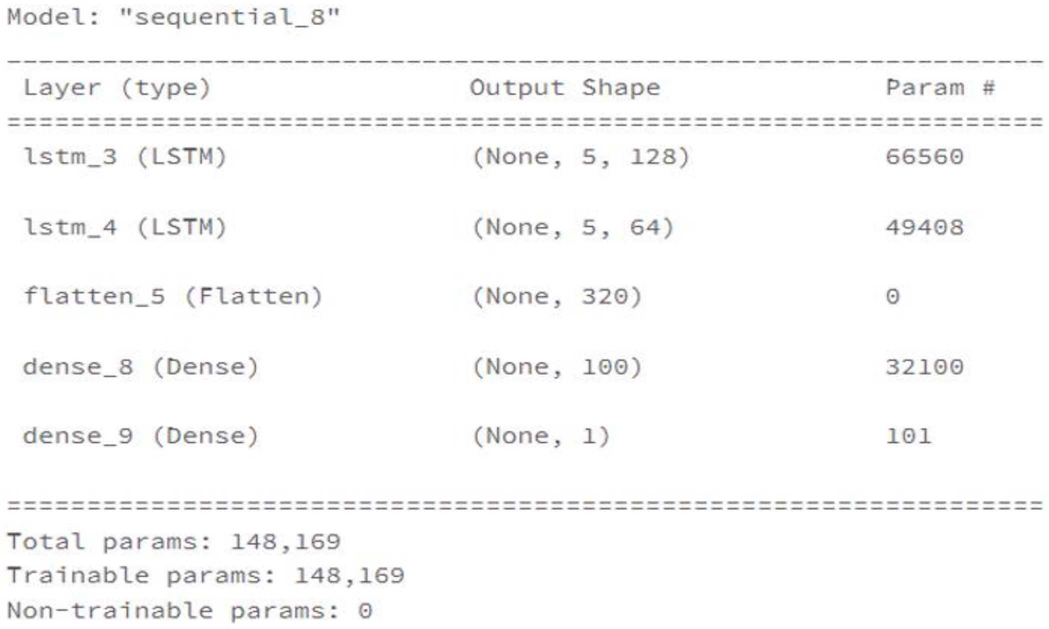
Loss: 0.0732

RECURRENT NEURAL NETWORK:



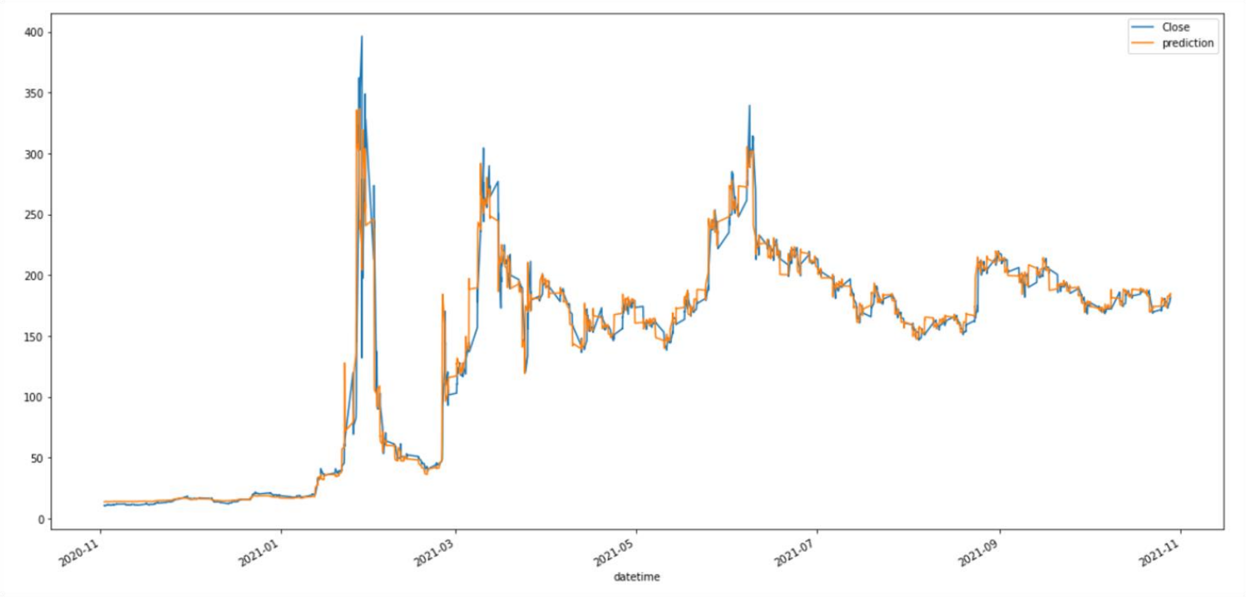
RNNs are popular models that have showed promise in a variety of machine learning tasks. RNNs were created to address the traditional assumption that all inputs (and outputs) are independent of one another. To boost the performance of this Neural Network Model, RNNs uses sequential information in each task. Recurrent neural networks (RNNs) are so named because they complete the same task for each element of a sequence, with the result being dependent on the prior computations. Another way to conceive of RNNs is that they have a "memory" that stores information about previous calculations.





```
Model: "sequential_8"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_3 (LSTM)               (None, 5, 128)            66560

 lstm_4 (LSTM)               (None, 5, 64)             49408

 flatten_5 (Flatten)         (None, 320)               0

 dense_8 (Dense)             (None, 100)               32100

 dense_9 (Dense)             (None, 1)                 101

=================================================================
Total params: 148,169
Trainable params: 148,169
Non-trainable params: 0
```

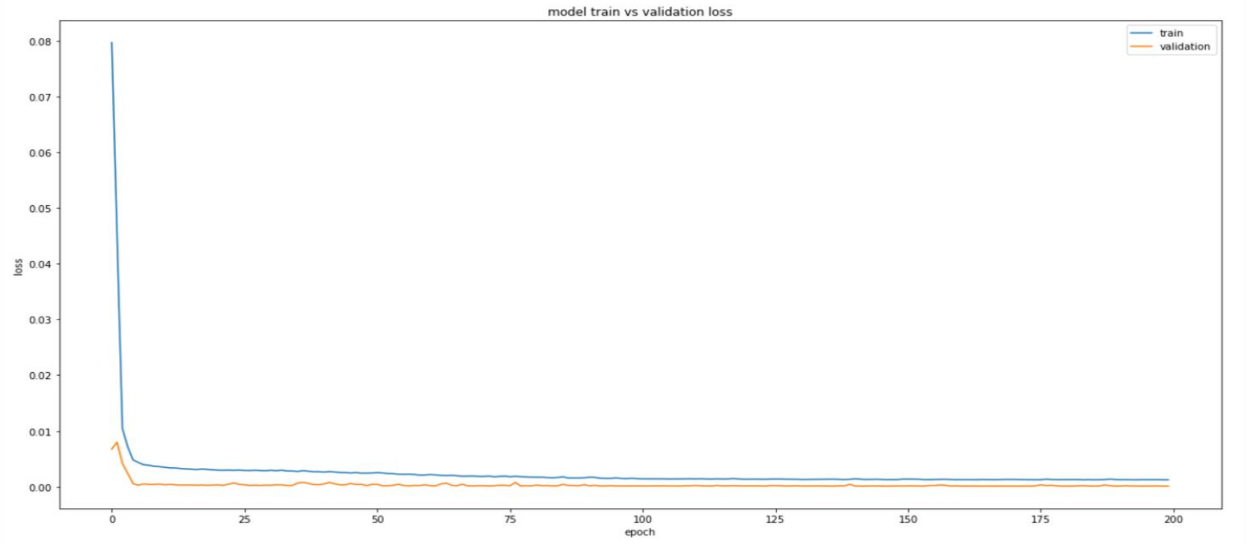
RNN-LSTM Model prediction vs. actual close price on current data



Loss: 0.0012

We calculated the Mean squared error (MSE) which is the most commonly used loss function for regression. The loss is the mean overseen data of the squared differences between true and predicted values. It is calculated to check if our trained model has any outlier with errors. We calculated the MSE to check which model was better than the other and as we see the loss of RNN is lesser than CNN, we conclude that RNN is the better model for our problem statement.

RNN-LSTM Model fit – Underfit/Overfit check



In order to check if the model is overfit/underfit we plotted the training and validation loss which decreased and stabilized around 0 – Good fit

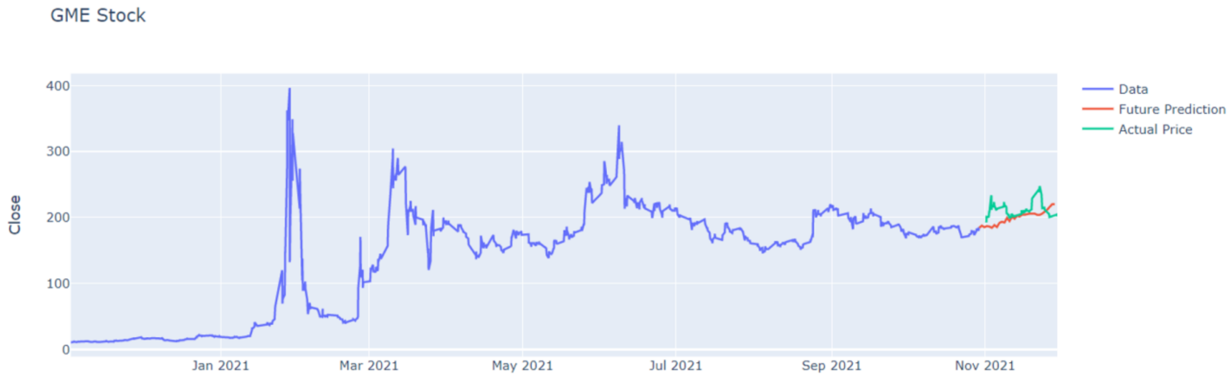
Long Short Term Memory Network:

The LSTM stands for Long Short Term Memory Network and is a type of RNN. LSTM is capable of learning long-term dependencies, to be more specific. A chain of repeating neural network modules exists in all recurrent neural networks. This repeating module in conventional RNNs will have a simpler structure than LSTM. The LSTM has a chain-like structure, but the repeating module has a completely different structure. Instead of a single neural network layer, there are many layers. In an LSTM, there are normally four layers, each of which interacts in a unique way. In LSTM, having the cell state and gate is critical. It has the power to delete or add information to the cell state in an LSTM model, and this interaction is strictly governed by gates.

MULTI-STEP PREDICTION

The challenge of anticipating a sequence of values in a time series is known as multistep-ahead prediction. Multi-stage prediction is a common strategy that involves applying a predictive model step by step and using the anticipated value of the current time step to determine the value of the next time step. The independent value prediction and parameter prediction methodologies are discussed in this study. The first method creates a separate model for each forecast step based on historical data. The second method involves fitting a parametric function to the time series and developing models to forecast the function's parameters. Multi-step time series forecasting is the process of predicting numerous time steps into the future. For multi-step forecasting, there are four basic methodologies to consider.

We created an array and calculated up to 15 days at a time. We considered X-test as the validation date and Y-test as the training data. Once normalized and converted by the multi-step prediction, it was passed on to CNN.
Stock price prediction for one month in future from current data using look_back = 15



Model predicted growth in stock price for Nov 2021; actual stock price roughly increased during the time period

Hyper-Parameter Tuning:

Multiple trials are done in a single training job to perform hyperparameter optimization. Each trial is a complete run of your training application with values for your selected hyperparameters set within the limitations you define. The AI Platform Training service maintains track of each trial's findings and adjusts for future trials. When the work is completed, you will receive a summary of all the trials as well as the most effective value configuration based on the criteria you select. You specify a single goal variable, also known as the hyperparameter metric, and hyperparameter tuning optimizes it. A popular statistic is the model's accuracy, as calculated by an evaluation pass. You can specify whether you want to tweak your model to maximize or minimize your measure, and the metric must be a numeric value.

CHALLENGES

- Predictive models like CNN and RNN are time consuming, so stock prediction on a huge dataset on a real time basis is difficult on conventional machines.
- Identification of external factors like economic, political, social and psychological is in itself a challenging task.
- Implementing predictive models without accommodating external factors like economic, social and psychological is a huge obstacle in predicting stock accuracy based solely on predictive models, especially for highly volatile stocks.

LEARNINGS

•CNNs are commonly used in solving problems related to spatial data, such as images.
•RNNs are better suited to analyzing temporal, sequential data, such as text or videos.
•Incorporating and Identification, external factors impacting the stock is a challenging task.
•Understanding the equilibrium between financial concepts and computational intelligence.

FUTURE ASPECTS

•To Identify and incorporate external factors impacting the stock price.
•To further augment the prediction capabilities on a real time basis.
•Inclusion of more predictive models like DNN.

CONCLUSION – By Aniket

While we tried to predict stock price for GME using r/wallstreetbets comments for past one year, we noticed that the trend of GME amongst reddit community is fairly correlated with the jumps in stock prices throughout the year. The time period when GME was discussed the most amongst reddit community (i.e, number of comments were the most), the stock price of GME jumped as well. Also, when looked at the word-cloud in data exploration part of the project, we see that people discuss a lot about the stock in general. Buy, call, short, money, sell, price, market, were few of the highly used words throughout all the comments. We can deduce that the reddit community is highly engaged in discussing/ buying and selling of the stocks. Therefore, we believe these discussions of meme stocks do have influence over the stock prices.

VADER is a good tool to perform sentiment analysis on social media data as it scores strength of emotions along with the polarity of a sentence. Using the scores from sentiment analysis, we were able to fit a CNN and RNN model using data for past one year. As expected, we found that RNN-LSTM performed better. And although the model was able to predict the future trend of the stock, it didn’t accurately predict the stock price for future one month. We believe the model could have performed better if we had more than 1 year of data to train the model on a longer period of time. But since meme stocks are relatively new and got popular within last year, we were limited to 1 years data. Maybe this project can be extended in future to continue studying the impact of sentiments amongst social media on stock price over the period of few more years. One of approach can be to collate data from different social media platform, like twitter, and perform in-depth sentiment analysis. Including daily news about the stock and performing NLP can help in enhancing the models too. And finally, including fundamental as well as technical analysis of the stock along with sentiment analysis might help in predicting the stocks better.

REFERENCES

1. Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. Proceedings of the International AAAI Conference on Web and Social Media, 8(1), 216-225.

2. https://aws.amazon.com/blogs/storage/migrating-and-managing-large-datasets-on-amazon-s3/

3. https://docs.databricks.com/notebooks/notebooks-use.html

4. https://medium.com/hands-on-data-science/lstms-or-cnns-for-predicting-stock-prices-2974c0c8c4ef

5. https://finance.yahoo.com

6. https://www.reddit.com/r/wallstreetbets/

7. https://www.wsj.com