Twitter Sentiment Analysis for Stock Price Prediction

Executive Summary

Driven by a curiosity about the financial underpinnings of stock investments, our analysis focused on whether remarks in tweets could influence stock price trends. The sentiment analysis suggested that reviews from the past five days may not significantly correlate with the next day's price changes. However, companies can still leverage this prototype model to explore other social platforms, gaining insights into investor perceptions and opinions about their products. This can enable them to tailor their offerings better to meet their customers' needs.

Project Objective

Our goal is to utilize sentiment analysis on tweets associated with specific stocks, using historical price data and tweets from previous trading days. We aim to categorize these tweets based on their sentiment and subsequently explore potential correlations between the sentiment classifications and movements in stock prices.

Data Description

Our dataset comprises two distinct subsets:

1. Twitter Data: We sourced our Twitter data from Hugging Face, which includes tweets containing references to five major companies: Amazon (AMZN), Apple (AAPL), Tesla (TSLA), Google (GOOG), and Microsoft (MSFT). This dataset spans five years, from January 1, 2015, to December 31, 2019, and contains 120,000 data points. Each record in this dataset consists of three attributes: the tweet's date, the company mentioned, and the tweet content.
2. Stock Price Data: We retrieved stock price information using the Python package Yfinance, which provides access to Yahoo Finance data. For each of the five companies mentioned above, our dataset includes daily closing prices and trading volumes over the same five-year period.

Methodology

1. **Preprocessing**
    a) The initial step in our data preprocessing involved converting the 'post_date' field from string format to a DateTime object to facilitate temporal analysis. During this conversion, we identified and removed data points with unreadable or corrupted date time entries. This cleaning process reduced the original dataset from 120,000 to 93,133 entries. Upon reviewing the cleaned data, we observed the distribution of tweets per company. On average, each company was mentioned in approximately 2 to 24 daily tweets (Please refer to Exhibit 1). This variance indicates the relative volume of social media attention each company received during the study period.
    b) Text Normalization: We would like to ensure that when we train the model, it remains a proper probabilistic model that can accurately reflect the underlying thematic structures of the whole data set. Here, we apply the function “normalize_corpus” in the package “Text_Normalization_Function” to help us deal with common issues such as Stemming and Lemmatization. The result can be referred to Exhibit 2.
2. **LDA**

a) Reasons for LDA:
   - Unsupervised Learning: LDA can automatically discover the topics in a text corpus without prior annotations(labeling) of the texts.
   - Dimensionality Reduction: It reduces the large set of possible words in the texts into smaller ones, which solves the problem of sparsity.
   - Easy Handling of Unseen Data: Once the model is trained, it can determine the topic distribution of new, unseen documents, which provides excellent flexibility when adding new data.

b) LDA Process:
   1. Vectorization: We turn the corpus in each document into the bag-of-words format, which you can refer to in Exhibit 3. Furthermore, we limit our feature numbers and only extract the top 1000 important ones to reduce the sparsity problem.
   2. LDA Model Construction: To simplify our final result, we only allow three distinguished topics, which represent Positive, Negative, and Neutral. Also, in hyperparameters α and β, a smaller α value encourages the model to produce documents that contain a smaller number of topics, with a few topics being dominant within each document, which also applies to β for a few words being dominant within each topic. Therefore, we set the α and β 0.9 to make sure each topic would have a dominant topic, which we can easily observe.
   3. Prediction: After training the LDA model, we can use it to distinguish the dominant topic in each document, and according to the document content in each topic, we can classify each topic as Positive, Negative, or Neutral by our intuition. You can refer to Exhibit 4 to get the whole picture.

c) Linear Regression Process:
   1. Data Preprocessing: First, we primarily aim to use the past five days (day 1 to day 5) tweets to predict whether the stock will rise or go down the next day (day 6). Therefore, we utilize Pandas to help us organize the dataset how we want. The details can be referred to the code I provided in the Appendix, and the result can be seen in Exhibit 5. Also, to Validate the model's accuracy, we randomly split the data into train and test based on the 80/20 rule.
   2. Linear Model Construction and Prediction: Using the LogisticRegression function in Sklearn, we can efficiently train the model and predict the test data to get accuracy.

3. AFINN Lexicon

a. Reasons for Using the AFINN Lexicon:
   - The AFINN lexicon is a list of English terms manually rated for valence with an integer between -5 (negative) and +5 (positive) by Finn Årup Nielsen between 2009 and 2011.
   - Tweets have their nature, so we adopted the AFINN Sentiment Lexicon as it was initially created for microblogs.

b. Lexicon Process

- Get Sentiment Score: The words in the tweets were mapped to the sentiment score calculated by the AFINN Lexicon method to get the sentiment of the tweets. (Exhibit 6)

c. Logistic Regression Process
   i. Split the whole dataset into training data (80%) and testing data (20%)
   ii. Data Preprocessing: As previously mentioned, our goal is primarily to use the tweets from the past five days to predict the stock price. Thus, we used Python's lag feature to collect the sentiment scores for the past five days and set those scores as the independent variables.
   iii. Logistic Model Construction and Prediction: Using the Logistic Regression function in Sklearn, we train the model, predict the test data, and compare the prediction with the actual situation.

Result

1. LDA: In the end, we can observe that our prediction is 0.54, which has much room for improvement. Also, we can explain why the accuracy rate is low by creating a cross table. Here, we found out that our model easily misclassified "Down" to "Up" (false negative), which might be improved by adjusting our threshold. (Exhibit 7)
2. AFINN Lexicon: The accuracy of our prediction is 0.51. By creating a cross table, we found out that our model easily misclassified "Down" to "Up" (false negative), the same as the result of our LDA method. Since AFINN's last update was 13 years ago, it might fail to map accurate sentiment scores to the tweets. (Exhibit 8)

Conclusion

1. How can Companies utilize:
   - Informing Trading Strategies: Identifying correlations between tweet sentiments and stock price movements can help organizations refine their trading strategies, allowing for more informed decisions on buying, selling, or holding stocks based on diverse platform insights.
   - Risk Management: Analyzing sentiment around stocks aids in risk assessment. Persistent negative sentiment might indicate underlying problems or potential volatility guiding investment decisions.
   - Enhanced Customer Insights: Sentiment analysis from tweets offers valuable insights into customer perceptions and opinions about products or services. By tracking this sentiment, organizations can pinpoint areas for improvement, address customer concerns, and adapt their marketing and communication strategies effectively.
2. Shortcomes and Improvement:
   - Sample Bias: Twitter users represent a specific demographic and may not encompass the full spectrum of stock market participants. This limitation could skew the perceived sentiment or popularity of a given stock.
   - Model Hyperparameters Adjustment: We should try more combinations of α, β, and Topics to get higher precision. Here, we only try a few because of the time limitation.

Appendix

Exhibit 1

ticker_symbol	averge tweets per day
AAPL	15.036786
AMZN	7.618270
GOOG	4.821138
MSFT	2.676543
TSLA	24.746260

Exhibit 2

```
['aapl continue sell premarket news',
 'uber self driving car kill pedestrian maker self driving autonomous car liable accident auto maker want take liability
customer auto insurer gm f tsla tm car aapl goog',
 'tim cook might good time use open market buyback actually defend b trade war nonsense option idiot load crash aapl wartime
take offense']
```

Exhibit 3

	1st	2nd	5g	aal	aapl	aapls	able	absolutely	acb	account	...	year	yes	yesterday	yet	yhoo	ym	yoy	yr	ytd	zone
0	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

Exhibit 4

	Topic_0	Topic_1	Topic_2	dominant_topic
Doc_0	0.3074	0.4959	0.1967	neutral
Doc_1	0.6107	0.1590	0.2303	negative
Doc_2	0.0754	0.3052	0.6194	positive
Doc_3	0.4163	0.0554	0.5282	positive
Doc_4	0.2424	0.2158	0.5418	positive

Exhibit 5

	start_date	end_date	positive	neutral	negative	Date	Close_diff
0	2015-01-01	2015-01-05	12	3	15	2015-01-06	Up
1	2015-01-02	2015-01-06	16	5	24	2015-01-07	Up
2	2015-01-03	2015-01-07	16	5	21	2015-01-08	Up
3	2015-01-04	2015-01-08	28	6	32	2015-01-09	Up
6	2015-01-07	2015-01-11	23	5	31	2015-01-12	Down

Exhibit 6

	index	post_date	tweet	ticker_symbol	sentiment_score
0	0	2019/3/7	There we go $TSLA	TSLA	0.0
1	1	2016/11/28	rubicon59: equityaddict Should I compare to $a...	AMZN	0.0
2	2	2018/4/28	After #Flipkart buy is $WMTeying$EBAY which ...	AMZN	0.0
3	3	2019/7/16	Tesla "we were able to cut costs by firing eve...	TSLA	-3.0
4	4	2018/5/2	$TSLA - A call for the history books - this ti...	TSLA	-3.0

Exhibit 7

Predicted:	Down	Up	All
True:			
Down	6	105	111
Up	9	132	141
All	15	237	252

Exhibit 8

Predicted:	0	1	All
True:			
0	129	390	519
1	124	417	541
All	253	807	1060

References

data source:
https://huggingface.co/datasets/emad12/stock_tweets_sentiment/viewer/default/train?p=959
Preprocess:
https://github.com/adrianaXZ/XZblog/blob/master/preprocess.ipynb
LDA Code:
https://github.com/johnny880624/Tweets_to_Trade/blob/main/LDA_Kuan.ipynb
Lexicon Code:
https://github.com/BosanHsu/WashU-Text-Mining/blob/main/Lexicon.ipynb
Training and Testing Dataset:
https://github.com/johnny880624/Tweets_to_Trade/blob/main/train.csv
https://github.com/johnny880624/Tweets_to_Trade/blob/main/test.csv
Apple Stock Price Dataset:
https://github.com/johnny880624/Tweets_to_Trade/blob/main/appl_stock_price.csv