Stevens Institue of Technology

Tesla Twitter Sentiment Analysis

First-year Paper Design
Author: Ruming L*
Advisor: Victor Luo

December 22, 2022

1 Data collection and characteristics

1.1 Raw Tesla data from twitter

Twitter may be the most active social media for tons of information of everything from everywhere. We used twitter API to collect all tweets from 2021-01-01 to 2022-10-31 which contain the keyword 'Tesla'.

All raw data from twitter are saved under 'Tesla/atafolder' with the format 'Tesla_XXXX-XX-XX.csv'.

We include the information about 'tweet.date', 'tweet.id' ,'tweet.content', 'tweet.url', 'tweet.likeCount', 'tweet.retweetCount', 'tweet.replyCount', 'tweet.quoteCount', 'tweet.user.id', 'tweet.user.username', 'tweet.user.displayname', 'tweet.user.rawDescription', 'tweet.user.created', 'tweet.user.verified', 'tweet.user.followersCount', 'tweet.user.friendsCount', 'tweet.user.favouritesCount', 'tweet.user.listedCount', 'tweet.user.statusesCount' and 'tweet.user.mediaCount' in every raw data files.

We show some graphs for the basic characteristics of the raw data.

- Fig 1: Lines plot of TSLA close price & tweets Count.
- Fig 2: Bar plot of the most tweets hour. Users are more active around 16:00 UTC.

*rliu38@stevens.edu

- Fig 3: Correlation matrix of raw data. The highest correlation is 0.34 between 'verifiedUserPercent' v.s. 'absReturn', other pairs don't have too much significant correlation.

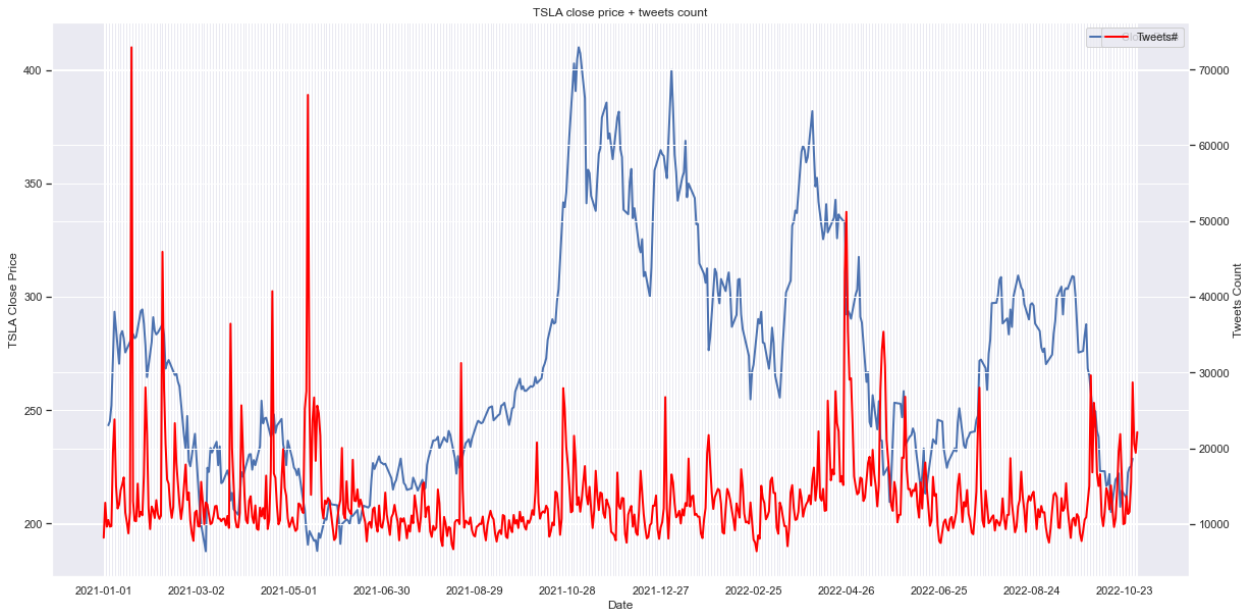

Figure 1: TSLA close price & total tweets count
Generated in 'Tesla/usefulFUnctionTest.ipynb'

1.2 Raw training data from Kaggle [1]

We use the dataset from Kaggle which was crawled and labeled positive/negative. The data provided comes with emoticons, usernames and hashtags which are required to be processed and converted into a standard form. After cleaning, this will become our training dataset of our algorithm.

2 Data cleaning [2]

The raw training data[3] given is in the form of a comma-separated values files with tweets and their corresponding sentiments. The training dataset is a 'csv' file of type **tweet_id**, **sentiment**, **tweet**, where the **tweet_id** is a unique integer identifying the tweet, sentiment is either 1 (positive) or 0 (negative), and tweet is the tweet enclosed in "".

The dataset is a mixture of words, emoticons, symbols, URLs and references to people. Words and emoticons contribute to predicting the sentiment, but URLs and references to people don't. Therefore, URLs and references can be ignored. The words are also a mixture

[1] Kaggle
[2] Data cleaning script is under Tesla/sentimentAlgorithm/generalDataCleaning.py
[3] This data is saved under 'Tesla/sentimentAlgorithm/training.1600000.processed.noemoticon.csv'

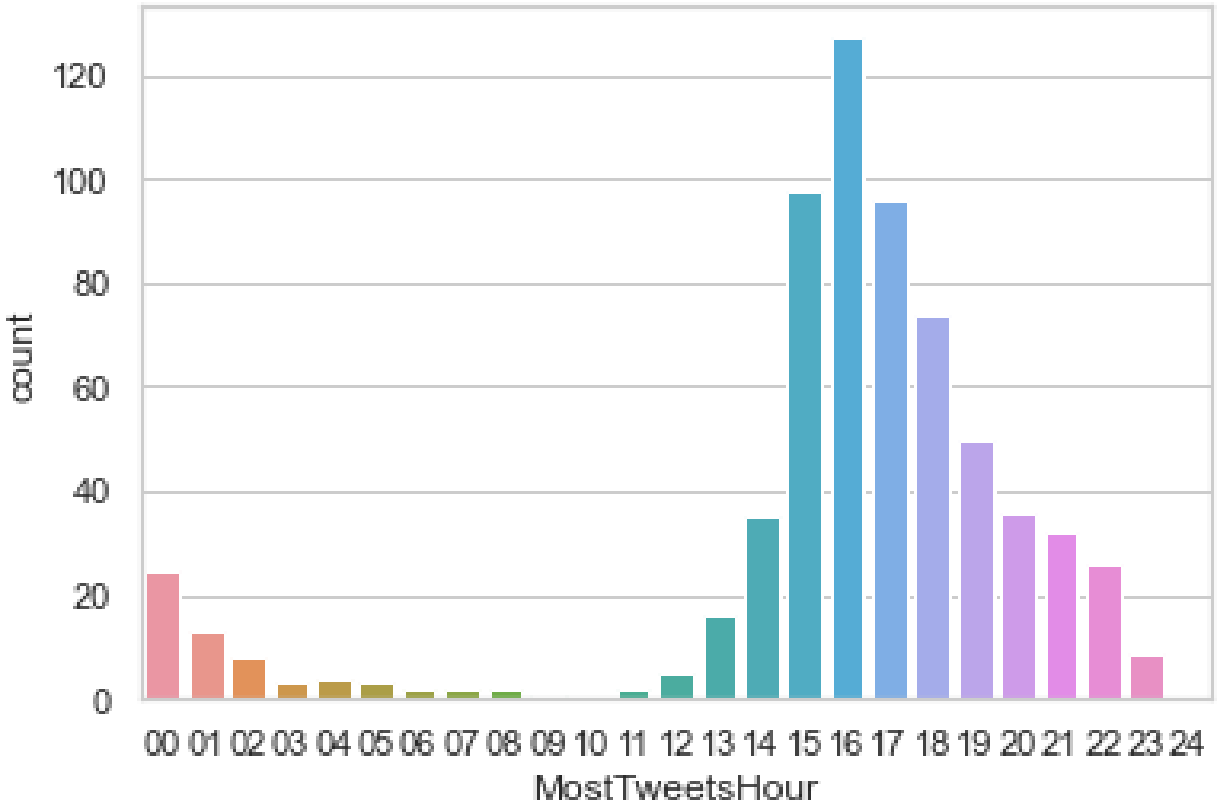

Figure 2: Most tweets hour
Generated in 'Tesla/usefulFUnctionTest.ipynb'

of misspelled words, extra punctuations, and words with many repeated letters. The tweets, therefore, have to be cleaned to standardize the dataset.

General data cleaning procedure:

- Transfer texts to lower cases.
- Replace stop symbol [**'"?!,.():;**] with space.
- Replace two or more spaces with a single space.
- Delete the space at the end of the sentence.

Twitter special data cleaning procedure:

- Replace "www..." or "http..." with "URL".

  Users often share hyperlinks to other webpages in their tweets. Any particular URL is not important for text classification as it would lead to very sparse features. Therefore, we re- place all the URLs in tweets with the word URL. The regular expression used to match URLs is **((www\.[\S]+)|(https?://[\S]+))**.

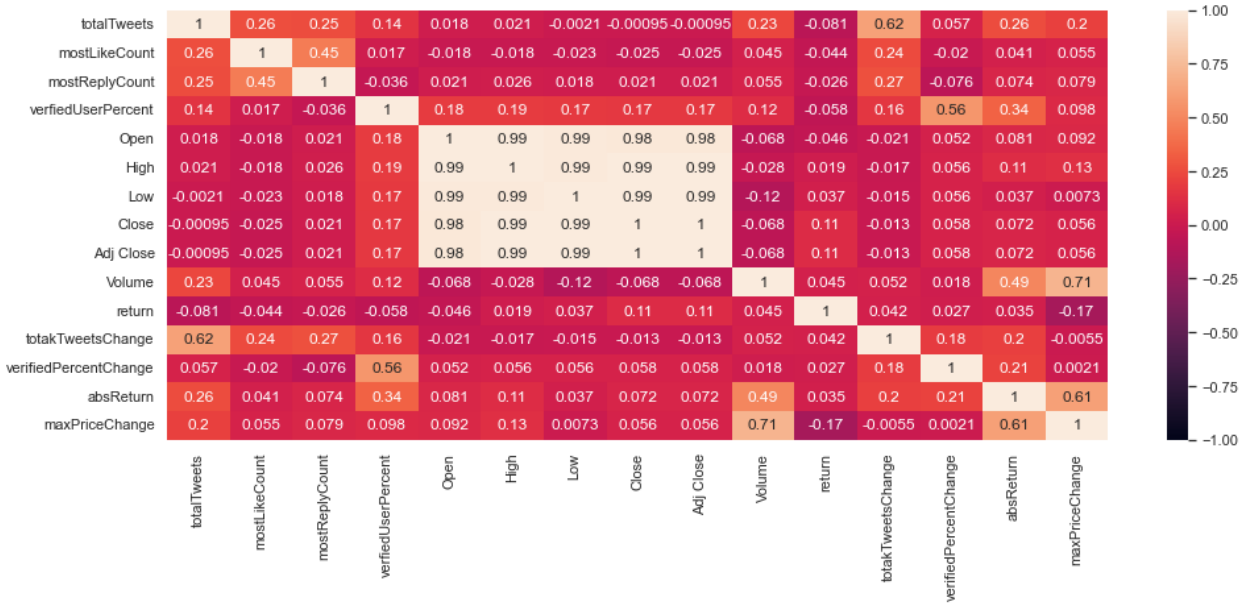

Figure 3: Correlation matrix of raw data
Generated in 'Tesla/usefulFUnctionTest.ipynb'

- Delete the "#" in hash tag. e.g. #tesla => tesla.

  Hashtags are unspaced phrases prefixed by the hash symbol (#) which is frequently used by users to mention a trending topic on twitter. We replace all the hashtags with the words with the hash symbol. For example, #hello is replaced by hello. The regular expression used to match hashtags is **#(\S+)**.

- Replace username with "USER_AT".

  Every twitter user has a handle associated with them. Users often mention other users in their tweets by **@handle**. We replace all user mentions with the word USER_MENTION. The regular expression used to match user mention is **@[\S]+**.

- Emotion symbol to "EMO_POSITIVE"/"EMO_NEGATIVE".

  We replace emotion symbols based on below table.

Emoticon(s)	Type	Regex	Replacement
:), :), :-), (:, (:, (-:, :')	Smile	(:\s?\)|:-\)|\(\s?:|\(-:|:\'\))	EMO_POS
:D, : D, :-D, xD, x-D, XD, X-D	Laugh	(:\s?D|:-D|x-?D|X-?D)	EMO_POS
;-), ;), ;-D, ;D, (;, (-;	Wink	(:\s?\(|:-\(|\)\s?:|\)-:)	EMO_POS
<3, :*	Love	(<3|:*)	EMO_POS
:-(, : (, :(,):,)-:	Sad	(:\s?\(|:-\(|\)\s?:|\)-:)	EMO_NEG
:,(, :'(, :"(	Cry	(:,\(|:\'\(|:"\()	EMO_NEG

- Emoji to "EMO_POSITIVE"/"EMO_NEGATIVE".

  We split emoji sentiment into two groups based on the classification of library **advertools**. The emoji with labels **face-negative**, **face-unwell**, **face-concerned**, **face-sleepy**, **warning** or **face-neutral-skeptical** will be replaced to "EMO_NEG". The

emoji with labels **face-smiling**, **money**, **face-affection**, **award-medal**, will be replaced to "EMO_POS". And for the emoji outside these labels, we will lookup in our own classification dictionary[4].

- Sparse the URL, EMO_POSITIVE... e.g. helloURLbye => hello URL bye.
- Remove hashtags of cryptocurrencies, e.g. $BTC, $tsla, $ETH.

  Tesla has some possible relation with cryptocurrencies. A lot of tweets about Tesla also includes some hashtags of other cryptocurrencies, we will remove these hashtags.

We apply above methods[5] to generate cleaned data. The files **v1_Tesla_XXXX-XX-XX.csv** and **training.1600000.clean.csv** are cleaned by these methods.

3 Training support vector machine (SVM)

Bloomberg sentiment is calculated by SVM, firstly, we will train our own SVM algorithm by the data from Kaggle.

3.1 Tokenlization

We create a frequency distribution of the unigrams present in the dataset and choose top 15000 unigrams for our analysis in Figure 4.

After extracting the unigrams, we represent each tweet as a feature vector in sparse vector representation. The sparse vector representation of each tweet is a 15000×1 dimensions vector. Each unigram is given a unique index depending on its rank. The feature vector for a tweet has a value 1 at the indices of unigrams if they are present in that tweet and 0 if they are not present in the tweet. After the tokenlization process, all cleaned tweets will be converted to a 15000×1 dimensions sparse vectors[6]

3.2 SVM algorithm

SVM, also known as support vector machines, is a non-probabilistic binary linear classifier. For a training set of points (x_i, y_i) where x is the feature vector and y is the class, we want to find the maximum-margin hyperplane that divides the points with $y_i = 1$ and $y_i = 1$. The equation of the hyperplane is as follow

$$\omega \times x - b = 0 \tag{1}$$

We want to maximize the margin, denoted by γ, as follows

$$max_{\gamma,\omega} s.t. \forall i, \gamma \leq y_i(\omega \times x_i + b) \tag{2}$$

to separate different groups points as far as possible.

[4]The dictionary is under 'Tesla/sentimentAlgorithm/emojiSentiment.csv'
[5]This script is under 'Tesla/sentimentAlgorithm/generalDataCleaning.py'
[6]All the tokenlized data is save under 'Tesla/sentimentAlgorithm/training16000CleanVector.csv'

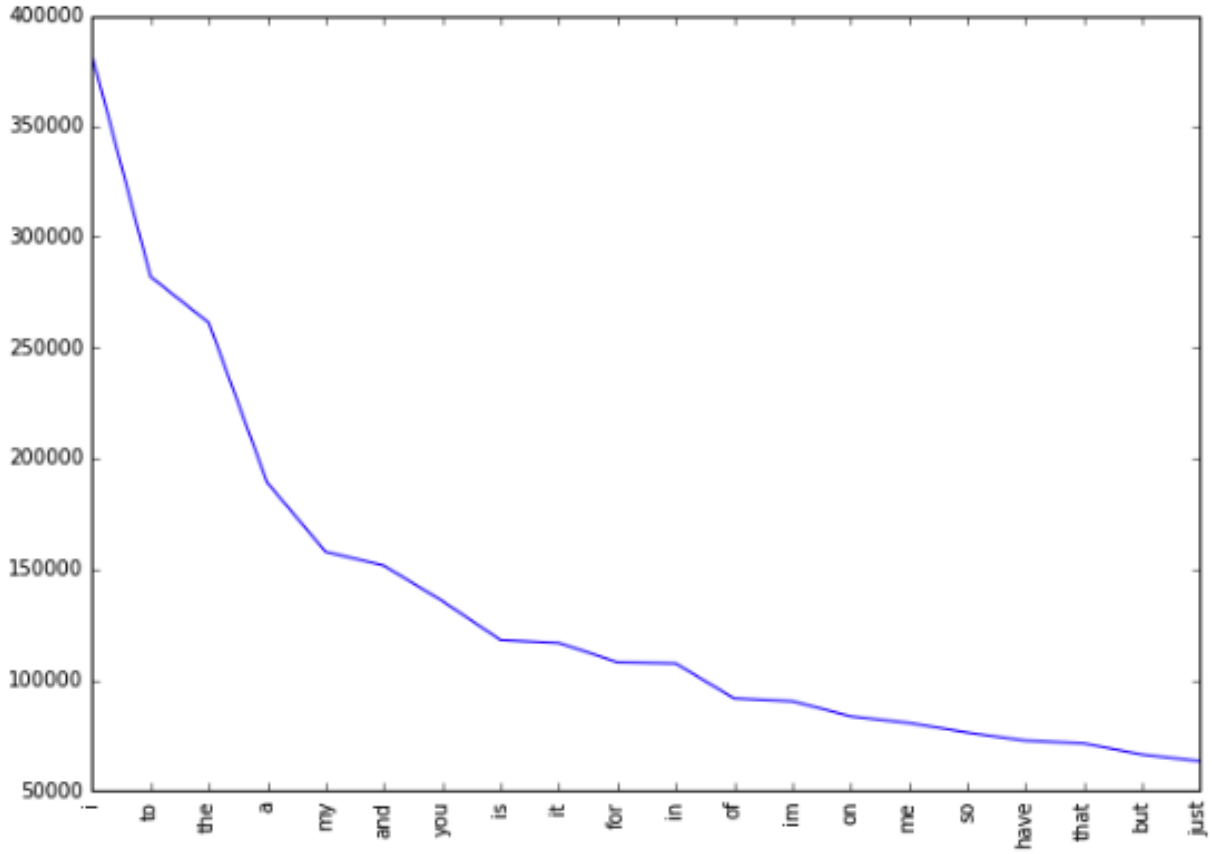

Figure 4: Top 20 unigrams

3.3 SVM training

As we mentioned before, we will use **raining16000CleanVector.csv** to train our model. Furthermore, we ramdomly split this data into two parts. 80% of them are used to train the data, the rest of them are used for model validation. Validation results for different kernal functions in svm:

- The precision of RBF kernal function is 76.5%.
- The precision of linear kernal function is 75.1%.
- The precision of polynomial kernal function is 68.3%.

3.4 Apply SVM algorithm to Tesla tweets

From section 3.3 we notice that the RBF kernal is a reasonable model to predict the sentiment of data from Kaggle. Now we want to see whether the SVM can explain the stock return of Tesla.

We use our trained SVM to calculate the sentiment of tweets about Tesla. Due to the laptop performance, now we can only do sample survey, the number of samples on each

day is 1000. The sentiment prediction of each day is saved under **v2_Tesla_XXXX-XX-XX.csv**.

These files also include the sentiment prediction of "Emoji model" and the combination. We show the correlation of our sentiment prediction and stock return are not significant.

1. All (sampling) users sentiment correlation: 2. Verfied users sentiment correlation:

combine Model mean	−0.025169
combine Model std	0.054496
svm mean	−0.036343
svm std	0.063239
Open	−0.040779
High	0.023351
Low	0.044117
Close	0.113161
Adj Close	0.113161
Volume	0.027073
return	1.000000

3. Bloomberg sentiment correlation:

	Open	High	Low	Close	Adj Close	Volume	return
Open	1.000000	0.995418	0.993225	0.986125	0.986125	-0.025678	-0.035002
High	0.995418	1.000000	0.993328	0.993789	0.993789	0.012531	0.025265
Low	0.993225	0.993328	1.000000	0.994297	0.994297	-0.070696	0.036919
Close	0.986125	0.993789	0.994297	1.000000	1.000000	-0.021227	0.102813
Adj Close	0.986125	0.993789	0.994297	1.000000	1.000000	-0.021227	0.102813
Volume	-0.025678	0.012531	-0.070696	-0.021227	-0.021227	1.000000	0.082850
return	-0.035002	0.025265	0.036919	0.102813	0.102813	0.082850	1.000000
TWITTER_SENTIMENT_DAILY_AVG	-0.158504	-0.159136	-0.161768	-0.158232	-0.158232	0.089403	0.038141
TWITTER_NEUTRAL_SENTIMENT_CNT	-0.021693	-0.032936	-0.022166	-0.029792	-0.029792	-0.075851	-0.010227
TWITTER_NEG_SENTIMENT_COUNT	0.110304	0.101461	0.112123	0.104067	0.104067	-0.122643	-0.013058
TWITTER_POS_SENTIMENT_COUNT	-0.035044	-0.043685	-0.036197	-0.041183	-0.041183	-0.032609	0.009744
TWITTER_PUBLICATION_COUNT	0.005212	-0.006709	0.005031	-0.003263	-0.003263	-0.093635	-0.001826

As the tables show, we didn't find a strong correlation between our sentiment scores with stock daily returns. What's worse, the Bloomberg's sentiment scores also don't have significant correlation with stock daily return.

3.5 Add control variables to our model