Università di Pisa

Sentiment Analysis from Stock Market tweets

Can Social Media sentiment affect the stock market?

Text Analytics Project

Prof. Lucia Passaro

Students:

Flavio Rossi

Carla Trejo Silva

Marco Vasta

Graziano Amodio

Contents

1. Introduction

It is not absurd to think that the perception of opinion's leaders and the general consensus have an influence in different aspects of society. The ever-increasing presence of social networks has allowed that the effect of these opinions to be noticeable in an immediate or even real time way.

One of the best studied cases of this influence in the field of text analysis is the effect that opinions, news, and forum interactions have on the stock market. A more positive or pessimistic perspective might affect the price of a stock or the volume of transactions in the market on the same or following day. Therefore, the motivation of this project is to determine whether the sentiment perceived from tweets relative to the stock market can produce an impact on it.

This project aims to identify the effects that Twitter user's opinions have on some targeted stock markets using NLP techniques in order to develop a useful model to predict market fluctuations according to opinions from twitter users through a sentiment analysis. This is done by comparing the results of the sentiment analysis obtained with a data-set consisting of the financial data of the same portfolio of SP500 stocks over the same time period.

2. Datasets

2.1 *Tweets* dataset

The data was collected by a team of researchers from the University Institute of Lisbon between 9th April and 16th July 2020 using several S&P500 tags, the references to the top 25 companies in this index, and the Bloomberg tag. The first file (tweets_labelled.csv) contains 5.000 tweets selected by random sampling of the data-set. In this file, 1.279 tweets were annotated in positive, neutral and negative sentiments. The second file (tweets_remaining.csv) contains the remaining 923.674 tweets. Both files are composed of 4 columns:

- **id**: each tweet was identified by a numerical code.
- **created_at**: exact date and time the tweet was published.
- **text**: text of the tweet
- **sentiment**: sentiment associated with the tweet (positive, neutral or negative).

2.2 *S&P500 Prices* dataset

The dataset was extracted from the yahoo finance platform for the time period from April 9, 2020 to July 15, 2020. The dataset consists of (in reference to the SP500):

- **date**: reference date
- **open**: opening price
- **high**: highest price reached in the day
- **low**: lowest price reached in the day
- **close**: close price adjusted for splits
- **adj close**: adjusted close price adjusted for splits and dividend and/or capital gain distributions
- **volume**: transaction volumes

The useful columns for this analysis are the closing and opening price of the stock index. In fact, a new column was added with the difference between the closing and opening price will be created. Now it is possible to note whether the index price increased or decreased on the day considered. Transaction volume is also a column

to be taken into account but it is summarized in the price change.

Next, the daily price change will be plotted with the right scale in order to compare it, at the end of the work, with the daily change in sentiment analysis.

3. Data Understanding

3.1 *Tweets labeled*

As previously stated, the labeled dataset is composed of 5.000 tweets of which 157 are duplicates. The only missing values in the dataset are the values for the *sentiment* column of the 3599 tweets that where not manually annotated.

	id	created_at	text	sentiment
0	77522	2020-04-15 01:03:46+00:00	RT @RobertBeadles: Yo\nEnter to WIN 1,000 Mon...	positive
1	661634	2020-06-25 06:20:06+00:00	#SriLanka surcharge on fuel removed!\n\nThe ...	negative
2	413231	2020-06-04 15:41:45+00:00	Net issuance increases to fund fiscal programs...	positive
3	760262	2020-07-03 19:39:35+00:00	RT @bentboolean: How much of Amazon's traffic ...	positive
4	830153	2020-07-09 14:39:14+00:00	$AMD Ryzen 4000 desktop CPUs looking 'great' a...	positive
5	27027	2020-04-12 21:52:56+00:00	RT @QuantTrend: Reduce your portfolio RISK! GO...	positive

Figure 1: Tweets labeled dataset

As can be seen from the figure 2, the distribution of the target variable in the labeled tweets is well balanced. This means that no further steps will be used for implementing data sampling techniques with the objective of improving the performance of the classifiers.

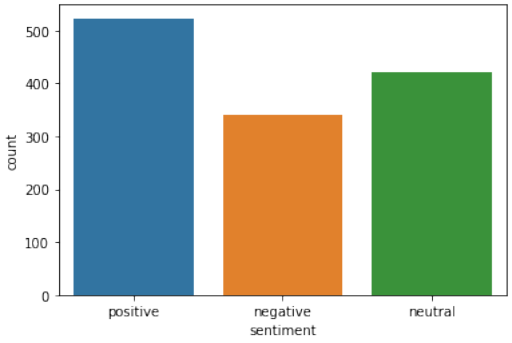


Figure 2: Distribution tweets labeled

Looking at the *text* column, it can be observed that tweets are within 140 characters in length. Values that deviate from this value usually hold links attached to the text.

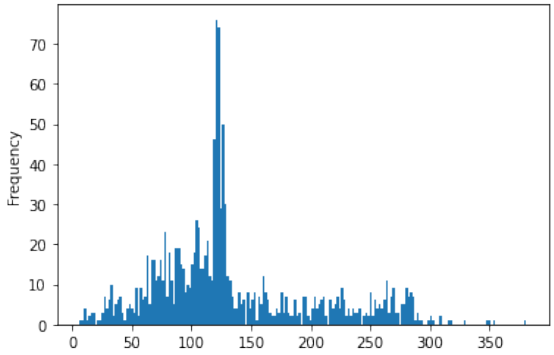


Figure 3: Tweets distribution per length

From the following plot is possible to observe the frequency of the major hashtags:

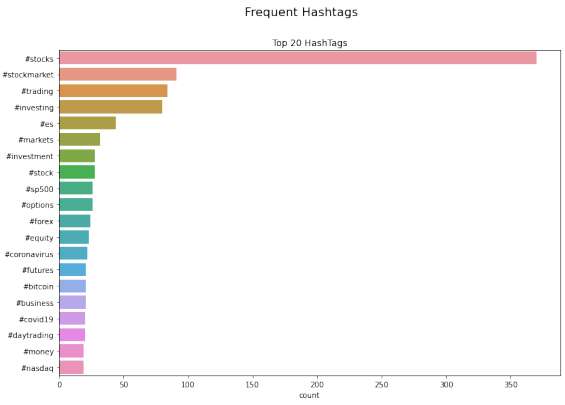


Figure 4: Frequent hashtags

4. Data Pre-processing

4.1 *Data Cleaning*

In this section, the dataset was cleaned from textual characteristics of tweets in order to facilitate the processing of the text. In this stage, the data cleaning involved:

- Remove handles from tweets ("@user");
- Remove of retweets (removing the words "RT" and "retweet");
- Replace of URLs with the code word "link" (then replaced with space);
- Remove special characters and replacement of these with space;
- Replace emojis with their description;
- Change all tweets into lowercase;
- Replace the character "_", added by the elimination of emojis, with the space;
- Remove contractions;
- Return spaces between words;
- Remove punctuation;
- Remove of stopwords;
- Lemmatization;
- Remove of duplicates.

To go more in depth about some of the choices in the data cleaning phase. Regarding the emojis, it was decided not to remove emojis altogether as they contribute to the conveyance of sentiment, but to replace them with a textual description of them using the python module *emoji*.

Another important step in all NLP preprocessing steps is removing contractions in order to better assign a "pos tag" and, then, lemmatize. Contractions are those little literary shortcuts that are taken where instead of "should have" is preferred "should've" or where "do not" quickly becomes "don't". To do this it was decided to use the python module "contractions" and apply a lambda function to the "text" column which will expand any contractions.

Stopwords are typically extremely used words that do not add much meaning to a sentence. In the English language common stopwords include "you, he, she, in, a, has, are, etc.". For these reasons it was decided to remove them using the "stopwords" module from the "nltk.corpus" library, setting the language in English and applying a lambda function to the "text" field.

Stemming and Lemmatization are methods that can be greatly useful in text preprocessing for NLP, in fact both of them help to map multiple words to a common root word. That way, these words are treated similarly and the model learns that they can be used in similar contexts. Usually lemmatization is preferred over stemming because it is a contextual analysis of words instead of using a hard-coded rule to chop off suffixes. Due to these reasons, it was decided to use the lemmatization instead of the stemming. First, it was applied to parts of speech tags, in other words, determine the part of speech (ie. noun, verb, adverb, etc.) for each word, using NLTK's word lemmatizer. Fo this, it is needed that the parts of speech tags be converted to wordnet's format. Then, a function which makes the proper conversion and, then, uses the function within a list comprehension to apply the conversion. Finally, the NLTK's word lemmatizer was applied within the list comprehension. The whole process has been enclosed in a pipeline. However, in the previous version the lemmatizer function requires two parameters: the word and its tag (in wordnet form).

At the end of the entire data cleaning phase, the dataset consisted of 4812 unique records, a positive result in order to don't lose too much information.

The following wordscloud is obtained from the most frequent words at the end of the data cleaning process. It is possible to see that, even after the cleaning phase, the most frequent words in the dataset still

continue to be: stock, trade, today, market.

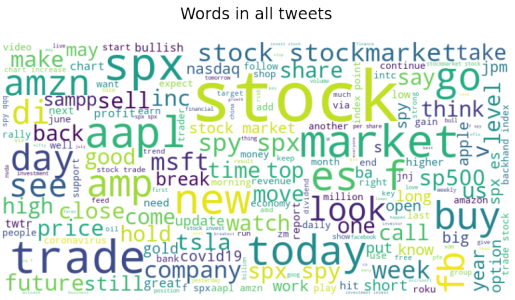


Figure 5: Wordscloud for all tweets

5. Classification

5.1 Data splitting for binary classification

For the classification task, it was decided to try two possible types: binary and multiclass. First, the dataset was evaluated considering only a binary classification, tweets would be labeled with *positive* and *negative* by this analysis; however, this choice would reduce the training dataset to 862 records. The dataset is, then, split with default values test_size = 0.2 and random_state= 25.

5.2 Classifiers for binary classification

In order to learn which classifier might be the best, for the binary classification. It was decided to apply different types of classifiers. Specifically: support vector, decision tree and Naive Bayes classifiers.

A pipeline was used to process the already labeled tweets and the tweets that are to be labelled with the fit function and then applied the predict function to the test array. This pipeline contains:

- **CountVectorize** that converts the collection of text documents to a matrix of token counts for the feature extraction (so in this case the tokenization is per sentence/tweet);
- **SelectKBest(chi2)** for selecting the most relevant features (although the default value of the parameter k is 10, following several iterations, it was decided to set k=20 in order to improve the results of the classifiers);
- **TfidfTransformer** that computes word counts using the previous CountVectorize and then computes the Inverse Document Frequency and only then the TF-IDF scores.

The evaluation for each of these classifiers are presented in the following tables:

Support Vector Machine

	Precision	Recall	F1-Score	Support
Negative	0,72	0,45	0,55	73
Positive	0,69	0,87	0,77	100
Accuracy			0.69	173
Macro avg	0,70	0,66	0,66	173
Weighted avg	0,70	0,69	0,68	173

Table 1: SVC binary classifier evaluation

	Predicted Positive	Predicted Negative
Actual Positive	33 (TP)	40 (FN)
Actual Negative	13 (FP)	87 (TN)

Table 2: SVC binary contingency table

Decision Tree

	Precision	Recall	F1-Score	Support
Negative	0,70	0,45	0,55	73
Positive	0,68	0,86	0,76	100
Accuracy			0.69	173
Macro avg	0,69	0,66	0,66	173
Weighted avg	0,69	0,69	0,67	173

Table 3: DT binary classifier evaluation

	Predicted Positive	Predicted Negative
Actual Positive	33 (TP)	40 (FN)
Actual Negative	13 (FP)	86 (TN)

Table 4: DT binary contingency table

Naive Bayes

	Precision	Recall	F1-Score	Support
Negative	0,72	0,45	0,55	73
Positive	0,69	0,87	0,77	100
Accuracy			0.69	173
Macro avg	0,70	0,66	0,66	173
Weighted avg	0,70	0,69	0,68	173

Table 5: NB binary classifier evaluation

	Predicted Positive	Predicted Negative
Actual Positive	33 (TP)	40 (FN)
Actual Negative	13 (FP)	87 (TN)

Table 6: NB binary contingency table

5.3 Data splitting for multiclass classification

In order to apply multiclass classifiers, the dataset must use all tweets labeled with a positive, neutral or negative *sentiment* value. The dataset is split again with default values test_size = 0.2 and random_state = 25. No data sampling technique was used since the target variable has a balanced distribution. Thus, the training set results in 1023 tweets, instead the test set consists of 256 tweets.

5.4 Classifiers for multiclass classification

It was decided to use the same pipeline implemented for the binary classification. However, following several iterations, it was decided to set the 'k' parameter for feature selection equal to 40 instead of 20, in order to improve the results obtained by the classifiers. The results of the evaluation obtained are as follows:

Support Vector Classifier

	Precision	Recall	F1-Score	Support
Negative	0,82	0,37	0,51	73
Neutral	0,42	0,86	0,56	76
Positive	0,64	0,40	0,49	107
Accuracy			0.53	256
Macro avg	0,63	0,54	0,52	256
Weighted avg	0,63	0,53	0,52	256

Table 7: SVC multi classifier evaluation

	Predicted Positive	Predicted Neutral	Predicted Negative
Actual Positive	27 (TP)	31 (FN)	15 (FN)
Actual Neutral	2 (FP)	65 (TN)	9 (FN)
Actual Negative	4 (FP)	60 (TN)	43 (FN)

Table 8: SVC multi contingency table

OneVsOne (SVC) Classifier

	Precision	Recall	F1-Score	Support
Negative	0,76	0,36	0,49	73
Neutral	0,39	0,84	0,53	76
Positive	0,59	0,31	0,40	107
Accuracy			0.48	256
Macro avg	0,58	0,50	0,47	256
Weighted avg	0,58	0,48	0,46	256

Table 9: OVO SVC multi classifier evaluation

	Predicted Positive	Predicted Neutral	Predicted Negative
Actual Positive	26 (TP)	35 (FN)	12 (FN)
Actual Neutral	1 (FP)	64 (TN)	11 (FN)
Actual Negative	7 (FP)	67 (TN)	33 (FN)

Table 10: OVO SVC multi contingency table

Decision Tree

	Precision	Recall	F1-Score	Support
Negative	0,86	0,33	0,47	73
Neutral	0,39	0,91	0,55	76
Positive	0,65	0,32	0,43	107
Accuracy			0.50	256
Macro avg	0,63	0,52	0,48	256
Weighted avg	0,63	0,50	0,48	256

Table 11: DT multi classifier evaluation

	Predicted Positive	Predicted Neutral	Predicted Negative
Actual Positive	24 (TP)	37 (FN)	12 (FN)
Actual Neutral	1 (FP)	69 (TN)	6 (FN)
Actual Negative	4 (FP)	69 (TN)	34 (FN)

Table 12: DT multi contingency table

Naive Bayes

	Precision	Recall	F1-Score	Support
Negative	0,76	0,34	0,47	73
Neutral	0,40	0,11	0,17	76
Positive	0,47	0,89	0,61	107
Accuracy			0.50	256
Macro avg	0,54	0,45	0,42	256
Weighted avg	0,53	0,50	0,44	256

Table 13: NB multi classifier evaluation

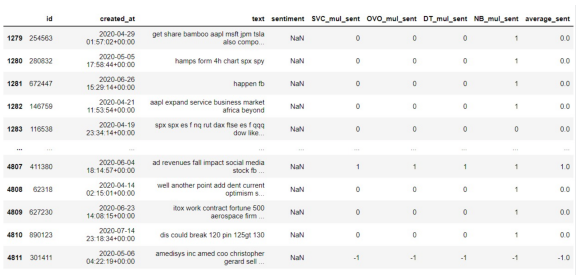
	Predicted Positive	Predicted Neutral	Predicted Negative
Actual Positive	25 (TP)	5 (FN)	43 (FN)
Actual Neutral	3 (FP)	8 (TN)	65 (FN)
Actual Negative	5 (FP)	7 (TN)	95 (FN)

Table 14: NB multi contingency table

5.5 Analysis and handling of the results

As can be seen in the figures above, the best-performing results in terms of F1-Score come from the binary classifiers: 0.69 for the SVC, the DT and the NB, although these deliver a trivial and not very significant result. This is easily guessed since, excluding the neutral class, which incidentally represented the minority class among the three, the test set consisted of only 172 tweets. However, the neutral class turns out to be of great importance for the purposes of analysis, it was decided to take into account only the results from the multiclass classifiers. In this case, however, performance is lowered, reaching the following F1-Score levels: 0.53 for SVC, 0.48 for SVC(OvO), 0.50 for DT and for NB. The last one, however, due to the intrinsic nature of the classifier unsuitable for multiclass datasets, has too low capacity to correctly predict the neutral class and it was decided not to take it into consideration for future analyses.

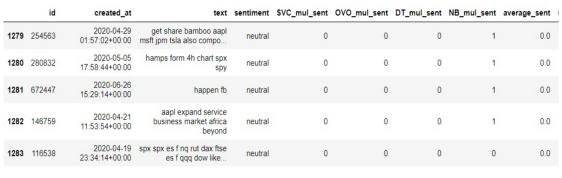
In fact, during the next step, the *sentiment* values predicted by the classifiers above will be transformed into integers in order to calculate the average for each tweet. More precisely, these values will be assigned: 1 for each positive tweet, 0 for neutral tweets, -1 for negative tweets.



	id	created_at	text	sentiment	SVC_mul_sent	OVO_mul_sent	DT_mul_sent	NB_mul_sent	average_sent
1279	254563	2020-04-29 01:57:02+00:00	get share bamboo aapl msft jpm tsla also compo...	NaN	0	0	0	1	0.0
1280	280832	2020-05-05 17:58:44+00:00	hamps form 4h chart spx spy	NaN	0	0	0	1	0.0
1281	672447	2020-06-26 15:29:14+00:00	happen fb	NaN	0	0	0	1	0.0
1282	146759	2020-04-21 11:53:54+00:00	aapl expand service business market africa beyond	NaN	0	0	0	1	0.0
1283	116538	2020-04-19 23:34:14+00:00	spx spx es f nq rut dax ftse es f qqq dow like...	NaN	0	0	0	0	0.0
...	...	...	...	...	...	...	...	...	...
4807	411380	2020-06-04 18:14:57+00:00	ad revenues fall impact social media stock fb ...	NaN	1	1	1	1	1.0
4808	62318	2020-04-14 02:15:01+00:00	well another point add dent current optimism s...	NaN	0	0	0	1	0.0
4809	627230	2020-06-23 14:08:15+00:00	itox work contract fortune 500 aerospace firm ...	NaN	0	0	0	1	0.0
4810	890123	2020-07-14 23:18:34+00:00	dis could break 120 pin 125gt 130	NaN	0	0	0	1	0.0
4811	301411	2020-05-06 04:22:19+00:00	amedisys inc amed coo christopher gerard sell ...	NaN	-1	-1	-1	-1	-1.0

Figure 6: Predicted sentiment converted in integer

Next, the average of the sentiment determined by each classifier is determined in the column *average_sent* (average sentiment). If this value is over zero (0), then the *sentiment* column is filled with the value 'positive'. Similarly, in the case that the average sentiment is less than zero (0), the sentiment is determined as negative. Otherwise, an average sentiment equal to zero ($\bar{0}$) will make the sentiment of the tweet neutral.



	id	created_at	text	sentiment	SVC_mul_sent	OVO_mul_sent	DT_mul_sent	NB_mul_sent	average_sent
1279	254563	2020-04-29 01:57:02+00:00	get share bamboo aapl msft jpm tsla also compo...	neutral	0	0	0	1	0.0
1280	280832	2020-05-05 17:58:44+00:00	hamps form 4h chart spx spy	neutral	0	0	0	1	0.0
1281	672447	2020-06-26 15:29:14+00:00	happen fb	neutral	0	0	0	1	0.0
1282	146759	2020-04-21 11:53:54+00:00	aapl expand service business market africa beyond	neutral	0	0	0	1	0.0
1283	116538	2020-04-19 23:34:14+00:00	spx spx es f nq rut dax ftse es f qqq dow like...	neutral	0	0	0	0	0.0

Figure 7: Table with "sentiment" column filled with labels

In order to assess the actual goodness of our classification results, it was decided to compare them with the 'sentiment polarity' values of 'TextBlob'. The comparison returned encouraging results. It is

possible to see, in the figure below, that the values returned by sentiment polarity do not deviate much from those of the average calculated on sentiment.

	id	created_at	text	sentiment	SVC_mul_sent	OVO_mul_sent	DT_mul_sent	NB_mul_sent	average_sent	sent_polarity
1279	254563	2020-04-29 01:57:02+00:00	get share bamboo aapl msft jpm tsla also compo...	neutral	0	0	0	1	0.0	-0.166667
1280	280832	2020-05-05 17:58:44+00:00	hamps form 4h chart spx spy	neutral	0	0	0	1	0.0	0.000000
1281	672447	2020-06-26 15:29:14+00:00	happen fb	neutral	0	0	0	1	0.0	0.000000
1282	146759	2020-04-21 11:53:54+00:00	aapl expand service business market africa beyond	neutral	0	0	0	1	0.0	0.000000
1283	116538	2020-04-19 23:34:14+00:00	spx spx es f nq rut dax ftse es f qqq dow like...	neutral	0	0	0	0	0.0	0.000000

Figure 8: Comparison of sentiments predicted by classifiers with sentiment polarity by TextBlob

As a final step, the dataset is merged (4812 tweets) by assigning the values of the average calculated on sentiment to the 1259 tweets that were already initially labeled.

	id	created_at	text	sentiment	average_sent	SVC_mul_sent	OVO_mul_sent	DT_mul_sent	NB_mul_sent	sent_polarity
0	77522	2020-04-15 01:03:46+00:00	yo collision enter win 1000 monarch tokens che...	positive	1.0	NaN	NaN	NaN	NaN	NaN
1	661634	2020-06-25 06:20:06+00:00	srilanka surcharge fuel remove fuel pump chart...	negative	-1.0	NaN	NaN	NaN	NaN	NaN
2	413231	2020-05-04 15:41:45+00:00	net issuance increase fund fiscal program gt y...	positive	1.0	NaN	NaN	NaN	NaN	NaN
3	760262	2020-07-03 19:39:35+00:00	much amazons traffic serve lastly help us find...	positive	1.0	NaN	NaN	NaN	NaN	NaN
4	830153	2020-07-09 14:39:14+00:00	amd ryzen 4000 desktop cpus look 'great' track...	positive	1.0	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...
4807	411380	2020-05-04 18:14:57+00:00	ad revenues fall impact social media stock fb ...	positive	1.0	1	1	1	1	0.439683
4808	62318	2020-04-14 02:15:01+00:00	well another point add dent current optimism s...	neutral	0.0	0	0	0	1	0.125000
4809	627230	2020-06-23 14:08:15+00:00	itox work contract fortune 500 aerospace firm ...	neutral	0.0	0	0	0	1	-0.200000
4810	890123	2020-07-14 23:18:34+00:00	dis could break 120 pin 125pt 130	neutral	0.0	0	0	0	1	0.000000
4811	301411	2020-05-06 04:22:19+00:00	amedisys inc amed coo christopher gerard sell ...	negative	-1.0	-1	-1	-1	-1	0.000000

Figure 9: Merged dataset with predicted tweets and labeled tweets

In the figure below, it is possible to see the new distribution of the target variable.

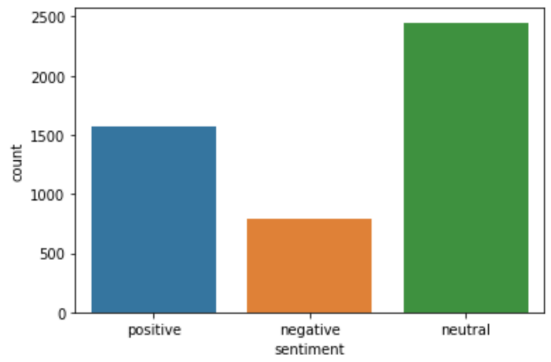


Figure 10: Target variable distribution

5.6 Comparison with "SP500 Prices" dataset

As a final step and a decisive point for the purposes of our work, the predicted 'sentiment' values were compared with the price difference values between daily close and open. As already mentioned, this last column of values was specially created by calculating the difference between the values in the 'Close' column and those in the 'Open' column.

	Date	Open	High	Low	Close	Adj Close	Volume	Dif_CloseOpen
0	2020-07-15	3225.98	3238.28	3200.76	3226.56	3226.56	4.686830e+09	0.58
1	2020-07-14	3141.11	3200.95	3127.66	3197.52	3197.52	4.507140e+09	56.41
2	2020-07-13	3205.08	3235.32	3149.43	3155.22	3155.22	4.902440e+09	-49.86
3	2020-07-10	3152.47	3186.82	3136.22	3185.04	3185.04	4.524190e+09	32.57
4	2020-07-09	3176.17	3179.78	3115.70	3152.05	3152.05	4.843650e+09	-24.12

Figure 11: S&P500 dataset with the change in open and closed prices

In order to better understand the value of the newly created 'Dif_CloseOpen' column, it was decided to briefly focus on the values of the 'Volume' column. It is possible to see that a "Volume" value (number of securities traded during a certain period, one day in the case of our dataset) that is down compared to the previous day is associated with a negative difference between close and open (of the same previous day). On the other hand, an increase in 'Volume' compared to the previous day is associated with a positive difference between close and open (of the same previous day). In order to compare the values of the 'Volume' and 'Dif_CloseOpen' columns, a StandardScaler was applied.

Figure 12

Finally, again following the application of a StandardScaler, the values of the column 'Dif_CloseOpen' were compared with the values of the average sentiment calculated for each tweet.

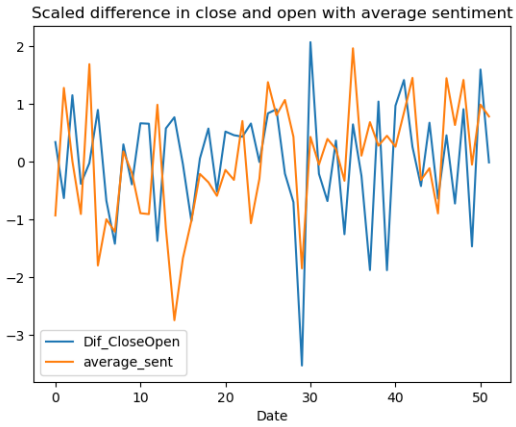


Figure 13

	Dif_CloseOpen	average_sent
2020-04-09	12.83	0.067633
2020-04-13	-20.83	0.240196
2020-04-14	40.96	0.141176
2020-04-15	-12.28	0.069597
2020-04-16	0.21	0.272300

Figure 14: Comparison between price change with the average of the predicted sentiments

6. Conclusion

The purpose of our analysis was to find out whether the opinion of users on social networks, and their level of confidence concerning stock market issues, could in any way influence the actual price trend. The final comparison between the average value of sentiment per day and the value of the corresponding difference between the daily close and open, showed fairly satisfactory levels of relationship between the two variables. Although the results of the sentiment classification per tweet were not highly performing, and therefore not entirely reliable for the purposes of the final goal, it is assumed that the volatility of the market, influenced by numerous other external and internal factors, makes it really difficult to understand which causes determine price trends more heavily.