Sentiment analysis for cryptocurrency market signaling

Raspanti Federico

Bocconi University

Supervisor

Prof. Dirk Hovy

In partial fulfillment of the requirements for the degree of

Bachelor of Science in Economics, Management and Computer Science

Contents

1 Introduction **1**

1.1 Background 2

1.1.1 The Crypto Fear & Greed Index 2

1.1.2 Sentix's Sentiment report 3

2 Methods **6**

2.1 Data collection and structure 6

2.2 Data preparation 8

2.2.1 Parsing the data 8

2.2.2 Extracting price and text data 8

2.2.3 Cleaning text data 10

2.2.4 Dividing text data into instances 11

2.2.5 Evaluating text data 12

2.2.6 Creating the final DataFrames 13

2.3 Analysis methods 13

2.3.1 Plots and correlation 13

2.3.2 Regression 14

2.3.3 Signaling 15

3 Results **16**

3.1 Correlations 16
3.1.1 News 16
3.1.2 Reddit 17
3.1.3 Telegram 17
3.1.4 Twitter 18
3.2 Regressions 19
3.3 Signaling 20

4 Discussion **21**
4.1 Correlation 21
4.2 Regressions 22
4.3 Signaling 23

5 Conclusions **24**

A **26**
A.0.1 News 27
A.0.2 Reddit 28
A.0.3 Telegram 31
A.0.4 Twitter 34

Chapter 1

Introduction

In recent years, cryptocurrencies have been taking over the world both economically and socially. The first one of them, the (in)famous Bitcoin, has seen incredible growth in a short amount of time, going from a market cap of 94 Million$ in 2013 to a market cap of 1.6 Trillion $ at its peak in November 2021[**1**]. Estimates put the number of crypto users at the end of 2021 at around 300 Million.[**2**]
With such figures, it comes to no surprise that the biggest cryptocurrencies users have the power to easily steer the public's opinion and behaviour. For example, shortly after Elon Musk tweeted, on February 24th 2021, a picture alluding to a skyrocketing of Dogecoin's price, an SEC investigation began to asses whether the tweet (and others similar to it) were part of a "pump and dump" fraudolent scheme.

An often underlooked question, however, is whether the opposite dynamic holds. In other words, **can the public's opinion influence the performance of the biggest cryptocurrencies?** Or to be more precise **does there exist a link between the people's sentiment and the performance of the biggest cryptocurrencies?**

1.1 Background

1.1.1 The Crypto Fear & Greed Index

Sentiment analysis has seen more and more use in the crypto market, with indexes such as the **Crypto Fear & Greed Index**(Figure 1.1), that calculates each days' sentiment of the public for the whole market, based on:

1. **Volatility (25%)**: current volatility and max. drawdowns of bitcoin vs. the corresponding average values of the last 30 days and 90 days;
2. **Market Momentum/Volume (25%)**: current volume and market momentum vs. the corresponding average values of the last 30 days and 90 days;
3. **Social Media (15%)**: how fast and how many interactions posts on various hashtags for each coin receive in certain time frames;
4. **Surveys (15%)** (currently paused): weekly crypto polls to ask people how they see the market;
5. **Dominance (10%)**: market cap share of the whole crypto market;
6. **Trends (10%)**: Google Trends data for various Bitcoin related search queries.

This work will mainly focus on points 3 and 4: what's the sentiment that social media show and how popular a coin is. However, the scope of this analysis is both much narrower and broader, as instead of focusing on the whole market, only 3 cryptocurrencies will be considered: **Bitcoin**, the "father of crypto", **Ethereum**, his trusty second in command, and **Ripple**, which at the time of the data collection (2019) was extremely relevant in the market as one of the biggest and most successful altcoins.

1.1.2 Sentix's Sentiment report

Another useful set of tools developed with the purpose of making better investment decision are Sentix's indicators:

1. **Sentix's Bitcoin sentiment Indicator**(Figure 1.2): it reflects the emotions – between greed and fear – of market participants. First, data on investors' opinions is collected through a survey, then the indicator is calculated with the formula

$$\frac{\sum Bullish - \sum Bearish}{\sum SurveyParticipants}$$

   It is useful for short-medium term investment decision, especially to indicate periods when the market is particularly bullish or bearish, corresponding to "extreme" values of the index.

2. **Sentix's Strategic bias on bitcoin Indicator**(Figure 1.3): Similar to the sentiment Indicator, and calculated in the same way, it reflects the 6-months expectation of investors. It's most useful for long term investment decisions. It is supposed to be used in combination with the other two indicators: if, for example, the Strategic bias is high while the sentiment is low, it is usually a signal of a good entry point in the market.

3. **Sentix's Bitcoin Overconfidence index Indicator**(Figure 1.4): it shows how likely it is that at a given point in time the price movement in a market will be perceived as a trend by investors. It's not calculated through survey data, but through price development. The higher the index, the more susceptible investors become to overestimating their own forecasting abilities. In the case of such "overconfidence", the probability of a trend reversal increases, as investors tend to build up excessive positions.

The scope of Sentix's work is closer to the scope of this thesis, but there are still some differences: two of the three aforementioned indicators collect their sentiment data through surveys. While this "classical" approach is surely more precise, it also reduces the amount of data available for analysis. Moreover, by only considering investor's sentiment, a big segment of the public is lost.

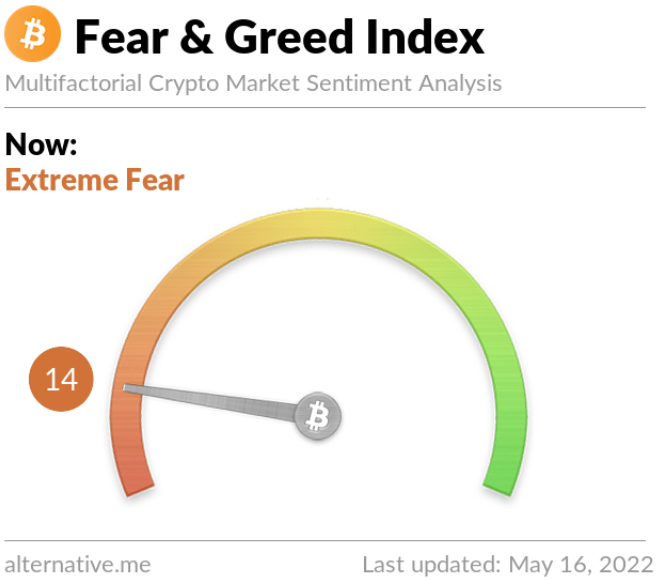


Figure 1.1 Crypto Fear & Greed Index

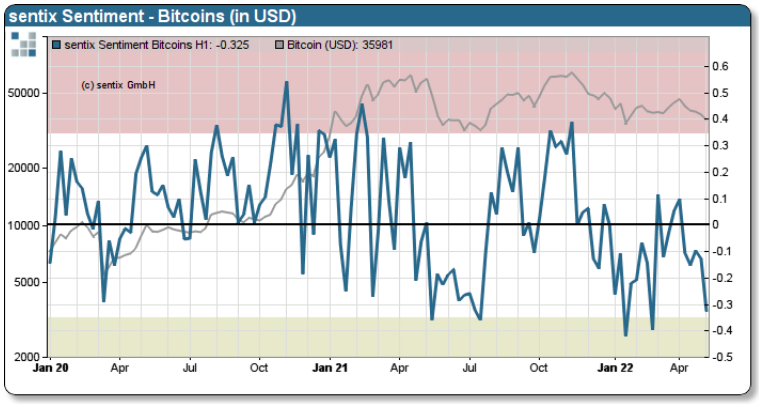


Figure 1.2 Sentix's Bitcoin sentiment

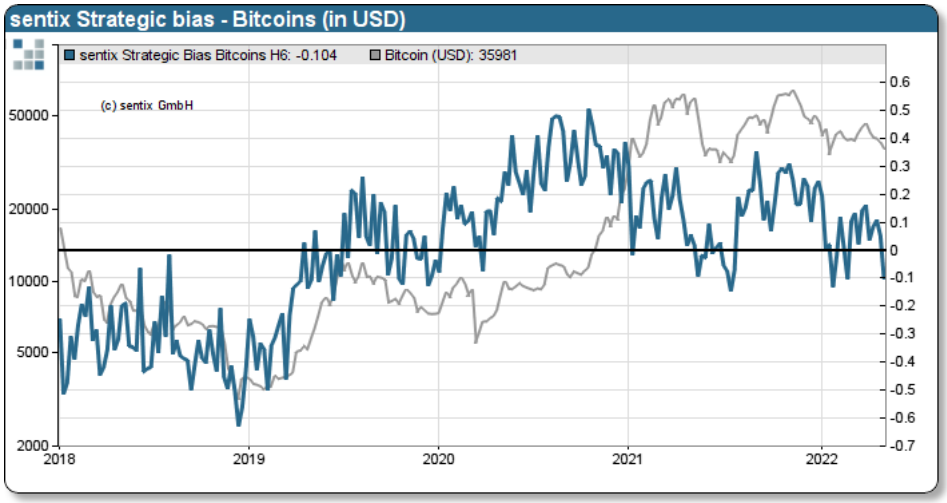


Figure 1.3 Sentix's Strategic bias on bitcoin

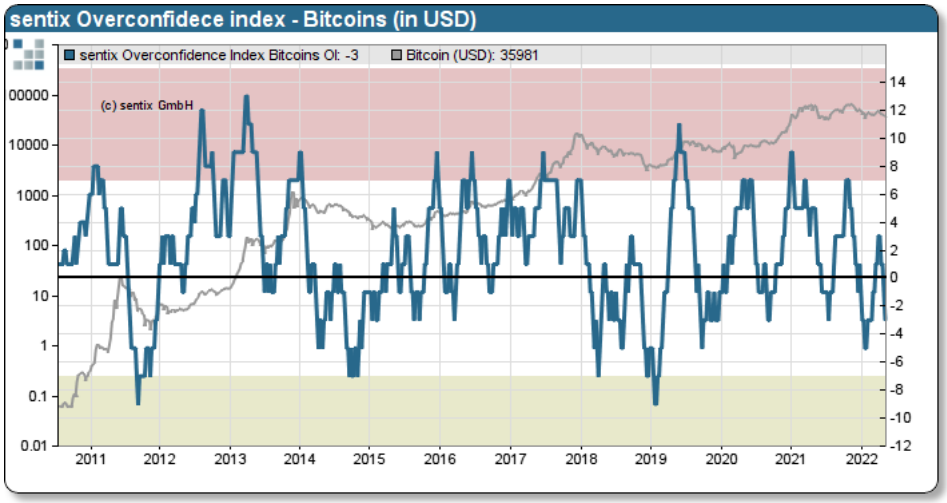


Figure 1.4 Sentix's Bitcoin Overconfidence index

Chapter 2

Methods

2.1 Data collection and structure

The data was collected over the course of 2019 by a script that, every five minutes, recorded various parameters about the data (see below). A common way of saving streaming data is to create JSONL file, to which a line containing data for each window of time is added sequentially in JSON format:

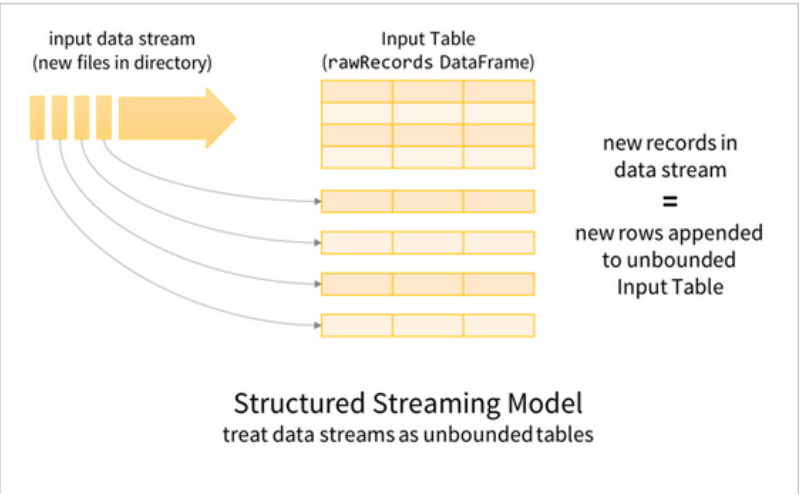


Figure 2.1 A visualization of the collection process

Theoretically this should lead to a single file, but due to occasional failures and restarts of the script the data is divided into 21 files, each containing text and price data collected every 5 minutes for a range of days, for a total of 12GB of data. Also, some periods of time are missing.

The structure of each file is the following:

Filename(contains first date in the file and the number of days for which it recorded the data);

{ **”DATETIME”**: {

”COIN”(BTC, ETH or XRP): {

”price”,

”steps”: {

”price”, **”var100”**, **”logreturn”** }

”TIME”,

”source”(newsapi, twitter, telegram or reddit): {

”categ”, **”source”**, **”mess_type”**, **”text_id”**, **”date”**, **”title_text”**, **”title_unigrams”**, **”title_bigrams”**, **”title_trigrams”**, **”body_text”**, **”title_unigrams”**, **”body_bigrams”**, **”body_trigrams”**}}}}

2.2 Data preparation

Due to the size, complexity and variety of the data, the preparation was broken down in a total of 5 steps, for a total of $\approx$ 850 lines of code:

1. Parsing the JSONL files into JSON files;
2. Extracting text and price data in separate files;
3. Cleaning text data;
4. Dividing text data into instances;
5. Evaluating text data.

2.2.1 Parsing the data

Each of the 21 original files was saved separately as folder containing a collection of JSON files (one per line).
This process led to a total of ≈ 50000 files with the same structure mentioned above, except for the filename, which was changed to ”Json0”, ”Json1”,...,etc.(Figure 2.2)

2.2.2 Extracting price and text data

The price and text data are separated mainly for clarity and to make the texts easier to handle. A simple script iterates over all the files in each folder, saves the price and text data into arrays, and finally merges them all in two DataFrames(Figure 2.3).

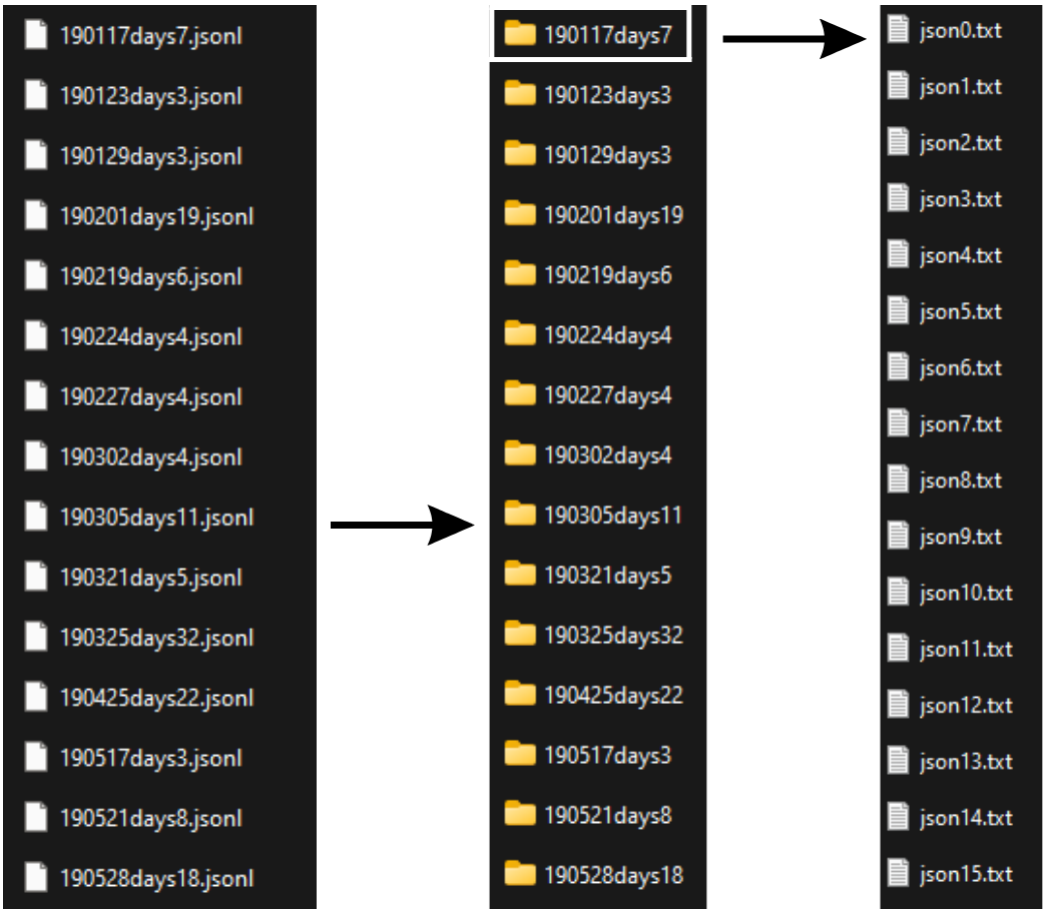


Figure 2.2 Left to Right: JSONL Files, corresponding folders, files in each folder

	Step	BTC_price	BTC_future_prices	BTC_var100	BTC_logreturn	ETH_price	ETH_future_prices	ETH_var100	ETH_logreturn	XRP_price	XRP_future_prices	XRP_var100	XRP_logreturn	Time
0	1	3172.953	3172.941	-0.000	-0.000004	104.426	104.449	0.022	0.000220	0.28	0.280	0.000	0.000000	['2019-01-23-10-15']
1	3	3172.953	3171.763	-0.038	-0.000375	104.426	104.422	-0.004	-0.000038	0.28	0.280	0.000	0.000000	['2019-01-23-10-15']
2	6	3172.953	3168.963	-0.126	-0.001258	104.426	104.263	-0.156	-0.001562	0.28	0.279	-0.357	-0.003578	['2019-01-23-10-15']
3	12	3172.953	3174.127	0.037	0.000370	104.426	104.413	-0.012	-0.000124	0.28	0.280	0.000	0.000000	['2019-01-23-10-15']
4	24	3172.953	3175.823	0.090	0.000904	104.426	104.480	0.052	0.000517	0.28	0.281	0.357	0.003565	['2019-01-23-10-15']

	twitter	twitter_date	news	news_date	reddit	reddit_date	telegram	telegram_date
0	Why CEOs are Naive to Ignore Trade War, Recess...	2019-01-17-17-38-30	NaN	NaN	This chart is both encouraging and discouragin...	2019-01-17-13-36-16	NaN	2019-01-16-15-18-17
1	$pny holding good the 65 sat \nmay be is good ...	2019-01-17-17-38-33	NaN	NaN	It will be more valuable and more stable on th...	2019-01-17-13-38-56	Who makes face like that in this bear trend🤔🤔🤔	2019-01-16-15-19-05
2	I like *BTC above* 3600 and *ETH above* 122, w...	2019-01-17-17-38-41	NaN	NaN	This is the first time I heard about this exch...	2019-01-17-13-45-22	@Crypto_Tonight_Show how are you brother?	2019-01-16-15-19-28
3	@crypto_rand What do you feel about $btc???	2019-01-17-17-38-42	NaN	NaN	>I've been out of the loop of bitcoin for a lo...	2019-01-17-13-48-45	Plz how do I access the bot that gives daily u...	2019-01-16-15-57-00
4	Bitcoin prices steady, holds within medium-ter...	2019-01-17-17-38-43	NaN	NaN	I'll be back in Jan 2020 to scoop some BTC for...	2019-01-17-13-55-25	@coinmarketcapbot	2019-01-16-15-57-25

Figure 2.3 An example of the DataFrames

2.2.3 Cleaning text data

While the price data is now properly prepared for the analysis, the text data still needs work. Specifically, the algorithm deployed in order to evaluate the sentiment of each piece of text has the following requirements:

1. Each word must be a single item in a list, array or DataFrame.
   This is easily implemented with some manipulation of the DataFrame.

2. There must be no special characters.
   This is easily implemented with a filter.

3. There must be no ”stop words”, i.e. words that carry no meaning directly linked to sentiment. Examples of this are the words ”no”, ”if”, ”when”, ”and”, ”the”, etc.
   The nltk package offers a library of English stop words, that is used to filter them out.

4. Each word must be provided not in its entirety, but as its ”lemma”, i.e. a stripped version of the word that still identifies it. For example, break, breaks, broke, broken and breaking all must be converted to ”break”.
   This is implemented thanks to nltk’s ”WordNetLemmatizer”, an algorithm that converts words into their lemmas using by knowing the role that each word plays in a sentence. The role is provided thanks to the POS-Tagger algorithm, implemented through nltk’s function ”pos_tag”.

2.2.4 Dividing text data into instances

Up until now the data has been divided into the original files, but that's not useful for the analysis. In order to make things easier, the texts must be divided by source and date, so that there is one file for each date covered by each source. For example, the texts mined from twitter should be divided into 193 files, one for each day.

This step was implemented with KNIME, a data analytics software that offers the perfect tools for this job.

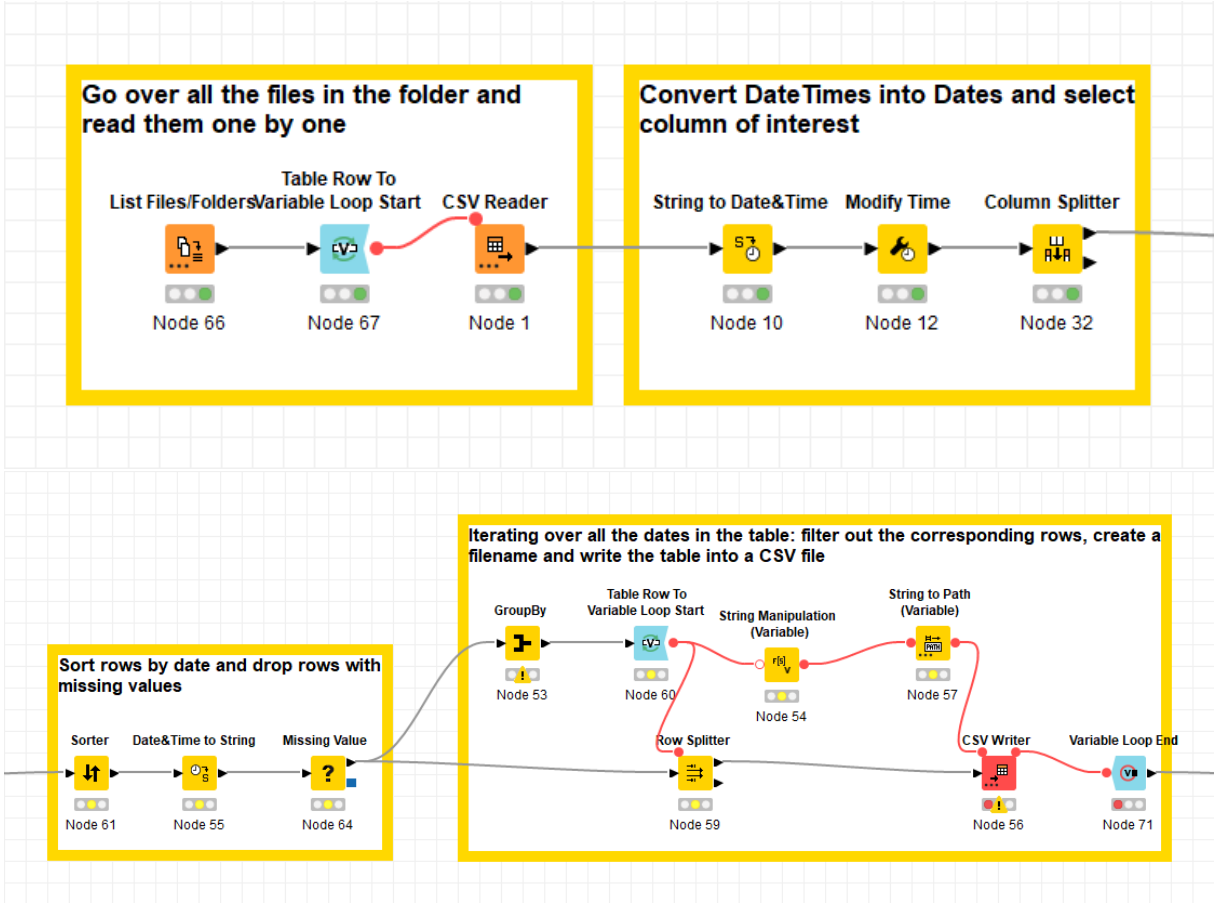


Figure 2.4 Above: the outer loop to read the files
Below: the manipulation component to clean and save the instances

2.2.5 Evaluating text data

The last step of data preparation follows the following process:

1. Each piece of text is passed to the SentimentIntensityAnalyzer algorithm, which calculates how positive, neutral and negative it is.
2. The three values are summarized in the ”compound” parameter, which gives an overall ”positivity value” of the text, between -1 and 1
3. A DataFrame is created by attaching to each piece of text the value calculated above.

At the end ≈ 800 DataFrames that look like in Figure 2.5 are left.

	twitter	evaluation
0	current price 174 79 increase 0 87 market cap ...	0.3182
1	total market cap 194 745 263 506 total 24 hr v...	0.0000
2	episode 95 bitcoin live topic today announces ...	-0.2732
3	investment app abra allow user connect account...	0.2263
4	bitcoin unit price 6 403 33 market cap 113 296...	0.0000

Figure 2.5 The end result of the evaluation process

2.2.6 Creating the final DataFrames

Now that the texts are evaluated, it's possible to create the definitive DataFrames for the analysis:

1. for each coin, the daily mean of the prices is calculated and added to a single array
2. for each text source, the daily mean of the prices is calculated and added to a single array, for both the real sentiments and the simplified ones

All the arrays created this way have their data scaled from 0 to 1, in order to facilitate comparisons

2.3 Analysis methods

2.3.1 Plots and correlation

This part of the analysis is carried out through 3 main functions:

1. **plot_comparison**: A simple function to plot two arrays one on top of the other.
2. **crosscorr**: Uses pandas' "corr" and "shift" methods to compute the correlation between two arrays, with an option to account for lags. It is important to note that pandas calculates the correlation using Pearson's formula:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

3. **plot_crosscorr**: Calculates the cross correlation between two arrays for different possible lag values.
   It returns the lag value leading to the highest correlation, the two plots aligned at the optimal lag and a graph of the correlations vs. lag.

2.3.2 Regression

In order to assess the relationship between the compound sentiment and the price of the coins, one of the simplest and most efficient tools is a linear regression. The script implements the following steps for each crypto-source combination:

1. deletes from the two dataframes, one for the coins and one for the sentiment, any date that does not appear in both, to avoid inconsistencies.

2. fits a linear regression model according to the formula

$$price_i = c + \alpha^I * sentiment_i$$

   where c is a constant and α^I is the coefficient to find.

3. generates dummies for months and days and fits a linear regression model according to the formula

$$price_i = c + \alpha * sentiment_i + \beta_i * M_i + \gamma_i * D_i$$

   where c is a constant, α is the sentiment's coefficient, β is the vector of coefficients of the months' dummy variables and γ is the vector of coefficients of the days' dummy variables.

The parameter used to assess the significance of sentiment on price is the R^2 value of the first regression, which indicates how much the sentiments' coefficient α is able to explain the price.
The second regression is implemented to check the extent to which the price is dependent on time.

2.3.3 Signaling

In order to get a clearer idea of the impact of sentiment on price, two types of signals are generated:

1. the first type of market signals are calculated by comparing the 5 days moving average (5_SMA) and 20 days moving average (20_SMA) of each coins' price. When they cross, a signal is generated following these rules:

   (a) if 5_SMA was above 20_SMA, generate a ”sell” signal

   (b) if 5_SMA was below 20_SMA, generate a ”buy” signal

2. the second type of market signal is generated with the same method as above, but the price is replaced by a new parameter ϕ, where

$$\phi_i = predicted_price_i * (\alpha * R^2) + price * (1 - (\alpha * R^2))$$

The plot function marks buying signals with a green, upwards triangle and selling signals with a red downwards triangle
The idea behind the last formula is to move the price towards what the sentiment would suggest the price should be, by an amount proportional to the power of the sentiment in determining the price.

Chapter 3

Results

3.1 Correlations

3.1.1 News

1. BTC: Lag = 58, Crosscorr = 0.38206042097006776
2. ETH: Lag = 57, Crosscorr = 0.511156721161426
3. XRP: Lag = 57, Crosscorr = 0.429801812867029

This source seems to have delivered satisfying results for what concerns correlation between coins and sentiment, considering that the literature considers 0.5 to be a ”strong” correlation coefficient. It is also quite impressive to notice that optimal lag to obtain the highest correlations is extremely consistent.
However, Figure A.3 shows that this results are only obtained for relatively high lag values of around 60 days over 200. Moreover, the positive lag value seems to suggest that the price *anticipates* the sentiment.

3.1.2 Reddit

1. BTC: Lag = 7, Crosscorr = 0.129335669820196878
2. ETH: Lag = 6, Crosscorr = 0.22700798232688085
3. XRP: Lag = 7, Crosscorr = 0.242227009724228

This sources delivered rather satisfying results. The small difference between optimal lags and their small values are both good signs, together with the correlation coefficients, that especially for ETH and XRP indicate at least a moderate correlation with sentiment.
Figure A.7 shows that the correlation values remain quite consistent in the neighborhood of the optimal, especially for BTC

3.1.3 Telegram

1. BTC: Lag = -45, Crosscorr = 0.5250229444635333
2. ETH: Lag = -45, Crosscorr = 0.4859437349514198
3. XRP: Lag = 16, Crosscorr = 0.3064529539799354

Telegrams results are also quite satisfying, as they show both good correlation coefficients and consistency in optimal lags (Figure A.11 shows that XRPs lag has a local maximum around -45).
Moreover, even though the optimal lags seem to be quite high, it is worth noticing that the correlation coefficients remain consistent for small negative values as well, suggesting that this sentiment may anticipate the price in some way.

3.1.4 Twitter

1. BTC: Lag = 84, Crosscorr = 0.052187660698606794
2. ETH: Lag = -78, Crosscorr = 0.26319474083570676
3. XRP: Lag = -78, Crosscorr = 0.35616363251122485

Finally, twitters results are the less satisfying of all since even though the correlation coefficients, especially those of ETH and XRP, could be considered good, they are only achieved for extreme lags, with insignificant values for small lags.

Before moving on to presenting the regression and signaling results, it is important to notice that the News source contained an extremely small and fragmented amount of data, which makes any result based on it unreliable and in general unlikely to be relevant.
For this reason, the News source was removed from any further analysis.

3.2 Regressions

First Regression			
Coin	Source	R^2	α^I
BTC	Twitter	0.08068201986082657	-0.6911682975053619
	Reddit	2.0595082928176467e-05	-0.021900470665914294
	Telegram	0.2562882448467174	-1.0559873508841764
	News	0.06301987828911737	-0.5952618352767082
ETH	Twitter	0.05519772029159142	-0.4286934467036905
	Reddit	0.005242182619890512	0.26191582105562783
	Telegram	0.06845357597893542	-0.409097124213771
	News	0.03487018147983578	-0.22866873654651726
XRP	Twitter	0.0009870740769555297	-0.05874368859046651
	Reddit	0.025181791472056902	0.5882110666037784
	Telegram	0.010750609628690455	0.1661227539886729
	News	0.024722793798451104	0.07248859347774698

Second Regression			
Coin	Source	R^2	α
BTC	Twitter	0.9457340849343104	0.08567506436415849
	Reddit	0.9456214205861687	0.14711536223473615
	Telegram	0.9449816974390448	0.03246539107029732
	News	0.9925471598017027	-0.038294899568604164
ETH	Twitter	0.8828298365247643	0.10775872544630777
	Reddit	0.8840178743467577	0.249860285511237
	Telegram	0.8813573349237556	0.08014266873236836
	News	0.9293751928295146	0.1222248461846545
XRP	Twitter	0.7830146897957475	0.18040729130970137
	Reddit	0.7853376609813708	0.4006899427501613
	Telegram	0.7769617420951688	0.07698335004146717
	News	0.4721745936370406	0.13161404551249234

The regression results offer an interesting insight: the first regression, only fitted with the compound sentiment, has almost completely insignificant R^2 values, with the exception of Bitcoin-Telegram. This seems to suggest that the sentiment have little to no explanatory power over the price of the coins.

The second regression shows how the price of Bitcoin and Ethereum was almost completely dependent on time properties, which makes sense considering that over the course of 2019 their prices kept rising almost constantly.

3.3 Signaling

As foreshadowed from the previous section, the signals generated using the ϕ variable are in general extremely close to the original prices, as seen in Figures A.6, A.10 and A.14. The exception is again the Bitcoin-Telegram pair.

Chapter 4

Discussion

The results presented in the previous chapter set the base for some interesting considerations.

4.1 Correlation

From the point of view of statistical correlation, Reddit and Telegram were the data sources that delivered the most satisfying results, as the findings were realistic and consistent, albeit average.
Especially impressive is the fact that the highest correlation values are obtained for relatively small lag values, which indicates that

1. The statistics are based on a reasonable amount of data
2. There is some form of linkage between price's movements and sentiment's movements

The quality of this sources' results can be attributed to many reasons, but it's probably mostly due to the platforms' users and structures: at its core, Telegram is an instant

messaging platform, but recent years have seen the rise of thematic groupchats, open to whomever is interested in sharing its passion or ideas on the topic. This is also true for cryptocurrencies, with many crypto-themed channels populated by enthusiasts that seek to improve their investments.
The combination of Telegram's structure and users lead to its data being highly frequent (virtually real time) and most important specialized.
A parallel argument can be made for Reddit's forum structure, based on "subreddits" whose function is of separating discussions on topics. Because of this the text mined from Reddit is also very specialized, but also much less frequent than Telegram's.

As for the reasons behind Twitter's unsatisfying results, they can be attributed to the extreme lack of specialization, which is not compensated, but possibly aggravated, by it's close-to-real-time frequency: since Twitter groups together the opinions expressed by both crypto enthusiast and by newcomers, the data mined from it are very likely to be too noisy and heterogeneous to be useful.

4.2 Regressions

Comparing the two regressions, it seems that sentiment has very little explanatory power over price. Despite this being unfortunate for this analysis' sake, it is perfectly consistent with reality, at least in regards to "big" cryptocurrencies such as the ones taken into consideration: in 2017, an article from Bloomberg titled "The Bitcoin Whales: 1,000 People Who Own 40 Percent of the Market" showed how "The top 100 bitcoin addresses control 17.3 percent of all the issued currency [...]. With ether, a rival to bitcoin, the top 100 addresses control 40 percent of the supply, and with coins such as Gnosis, Qtum, and Storj, top holders control more than 90 percent."[**3**]; in 2021, a Wall Street Journal article titled "Bitcoin's 'One Percent' Controls Lion's Share of the Cryp-

tocurrency's Wealth" discussed the fact that about 0.01% of bitcoin holders controlled about 27% of the currency in circulation [**4**].
It thus makes perfect sense to consider the possibility that people's opinion may not play a significant role in determining cryptocurrencies' prices, which are likely to be influenced by more "classical" parameters, such as market volume, time-dependent parameters, support levels etc.

4.3 Signaling

The results of this final analysis are one more confirmation of the weakness of sentiment as a price determiner: due to the R^2 values being very close to 0, the ϕ are virtually identical to the original prices.
Specifically, with an average R^2 of ≈ 0.05, the impact is about 5%. Excluding the Bitcoin-Telegram combination, with an R^2 of ≈ 0.25, the average impact drops down to $\approx 3\%$.
This means that whatever signal generated with the ϕ will be practically the same as the ones generated with the original prices, as shown in Figures A.6, A.10 and A.14.
As for the unexpectedly good results of the Bitcoin-Telegram pair, they can be attributed to the specialization and frequency factors discussed at the end of section 4.1.

Chapter 5

Conclusions

To sum up, the problem of understanding whether public sentiment is able to significantly influence the price of cryptocurrencies proved to be fairly complex, and it's difficult to obtain a clear cut answer to the initial question. Despite this, this analysis offered some interesting insights on the issue, which could help set the basis for future work.
First, it showed that a link between sentiment and price *does* exist, under certain conditions.
Second, it showed that these conditions depend on many variables that were ignored by this analysis, including but not limited to the differences between platforms, weight of big holders' opinions vs weight of small holders' opinions and time-dependency.
Including this variables in the model would most likely improve the results to a certain degree, but collecting the relevant data and modeling it properly would require not only more time but also more sophisticated models.
Ii is however worth mentioning that this analysis aimed not only at generating signals based on sentiment, but also at providing clear and understandable reasons behind the generation process. The deployment of more sophisticated models, such as deep neural networks, would probably lead to better results at the cost of its characteristic "black box effect", making it way harder to understand how the variables relate to each other.

Bibliography

[1] https://www.coinbase.com/price/bitcoin.

[2] https://triple-a.io/crypto-ownership/

[3] Vigna, P., The Wall Street Journal, *Bitcoin's 'One Percent' Controls Lion's Share of the Cryptocurrency's Wealth*, Dec, 2021.

[4] Olga Kharif, Bloomberg, *The Bitcoin Whales: 1,000 People Who Own 40 Percent of the Market*Dec 2017.

Appendix A

A.0.1 News

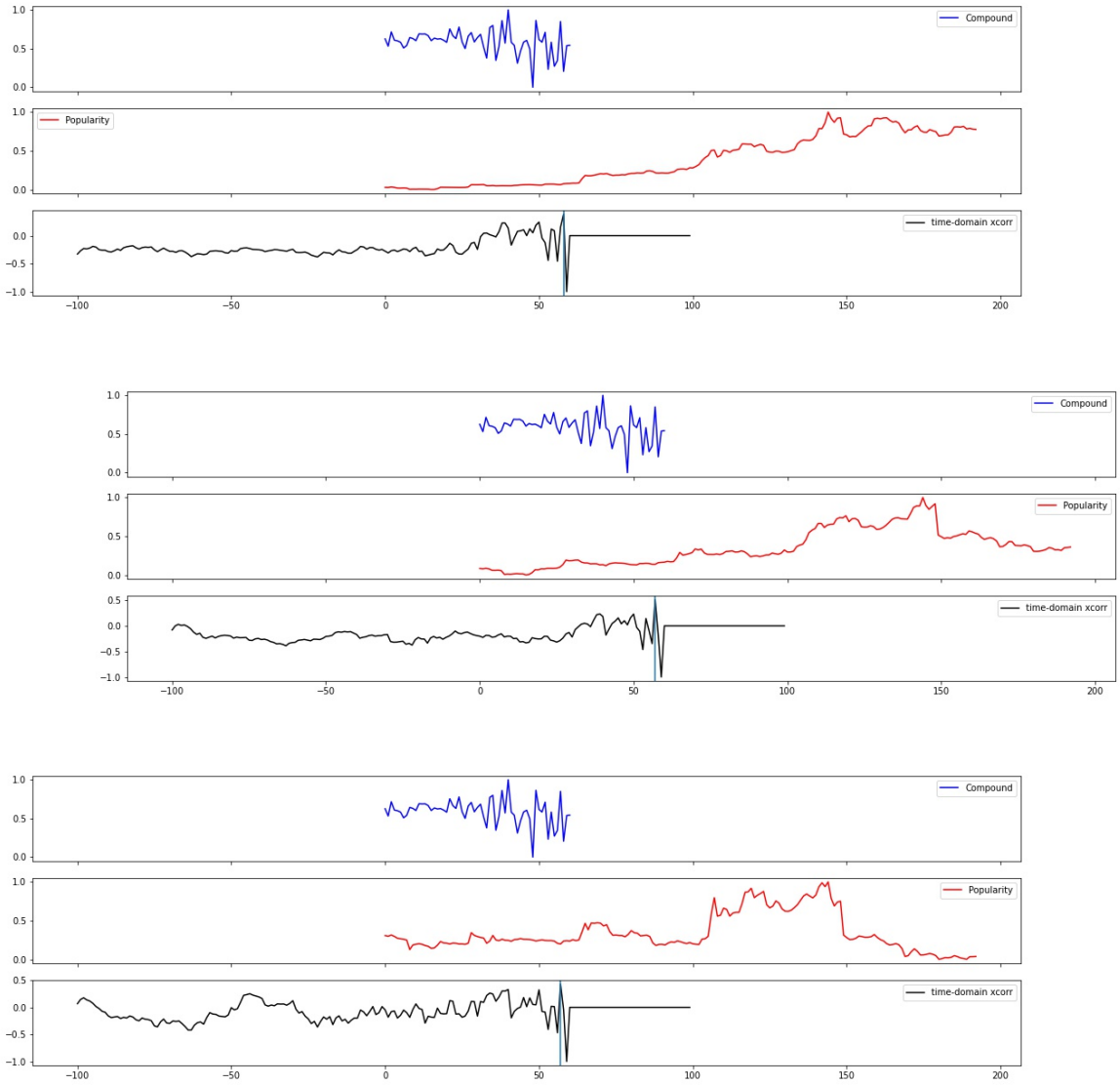


Figure A.1 News Compound vs, from top to bottom:

1. BTC: Lag = 58, Crosscorr = 0.38206042097006776
2. ETH: Lag = 57, Crosscorr = 0.511156721161426
3. XRP: Lag = 57, Crosscorr = 0.429801812867029

A.0.2 Reddit

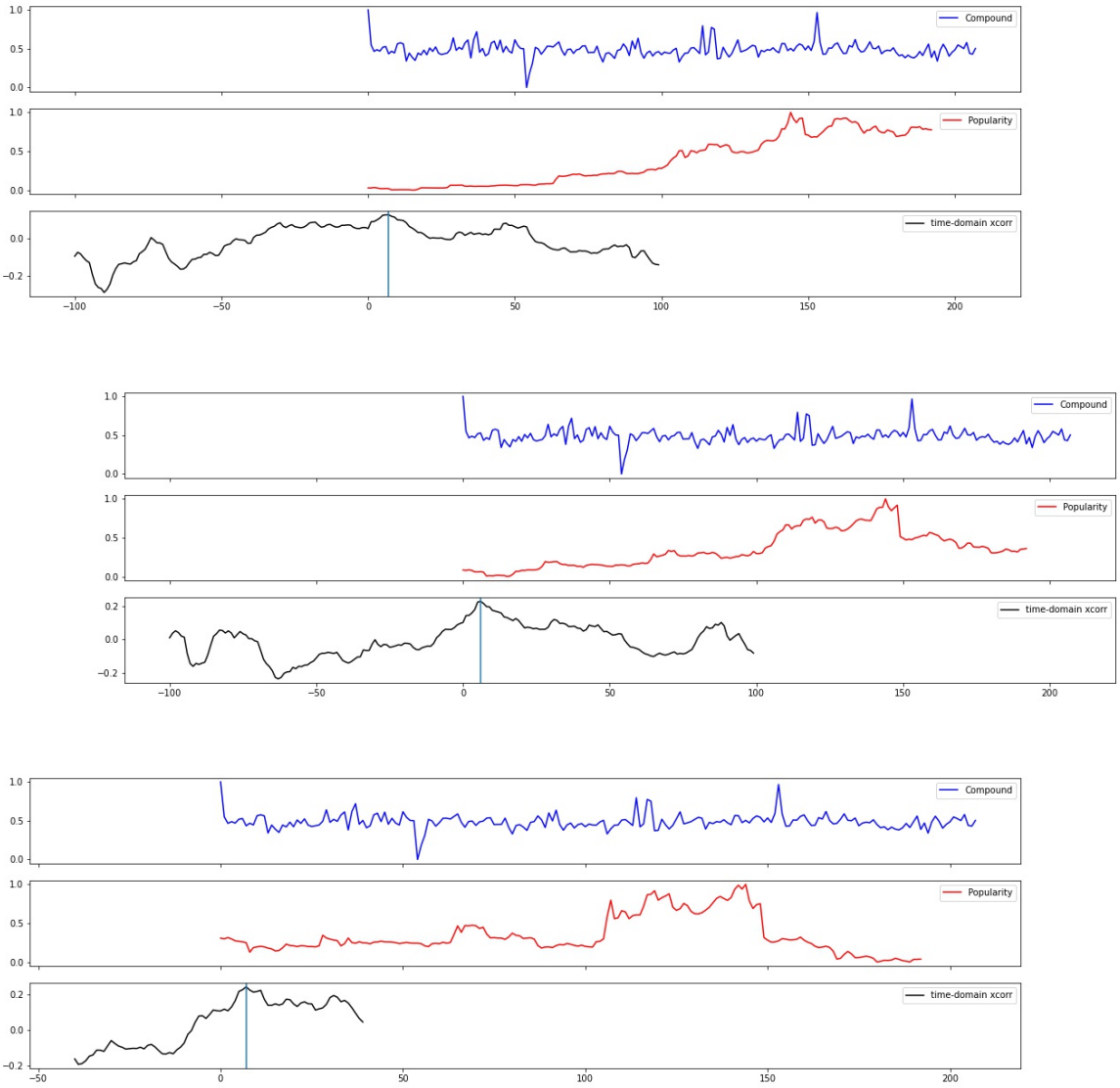


Figure A.2 Reddit Compound vs, from top to bottom:

1. BTC: Lag = 7, Crosscorr = 0.12933566982019687
2. ETH: Lag = 6, Crosscorr = 0.22700798232688085
3. XRP: Lag = 7, Crosscorr = 0.2422270097242288



Figure A.3 Predicted values from the reddit source for, from top to bottom, Bitcoin, Ethereum and Ripple.
In orange, prediction with date dummies; in green, prediction without date dummies.

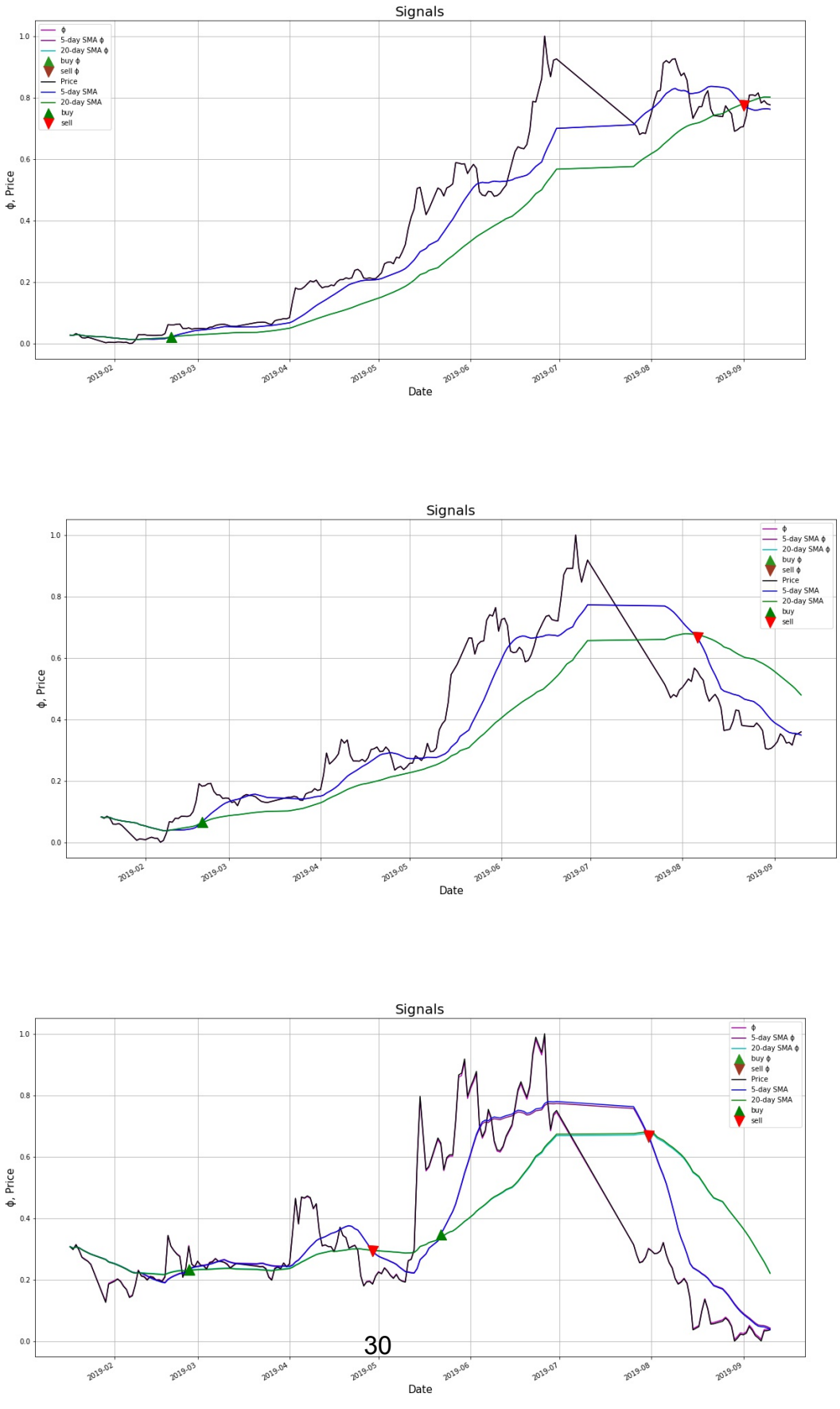


Figure A.4 Signals generated for, from top to bottom, BTC, ETH, XRP. ϕ is calculated using sentiment scores from Reddit

A.0.3 Telegram

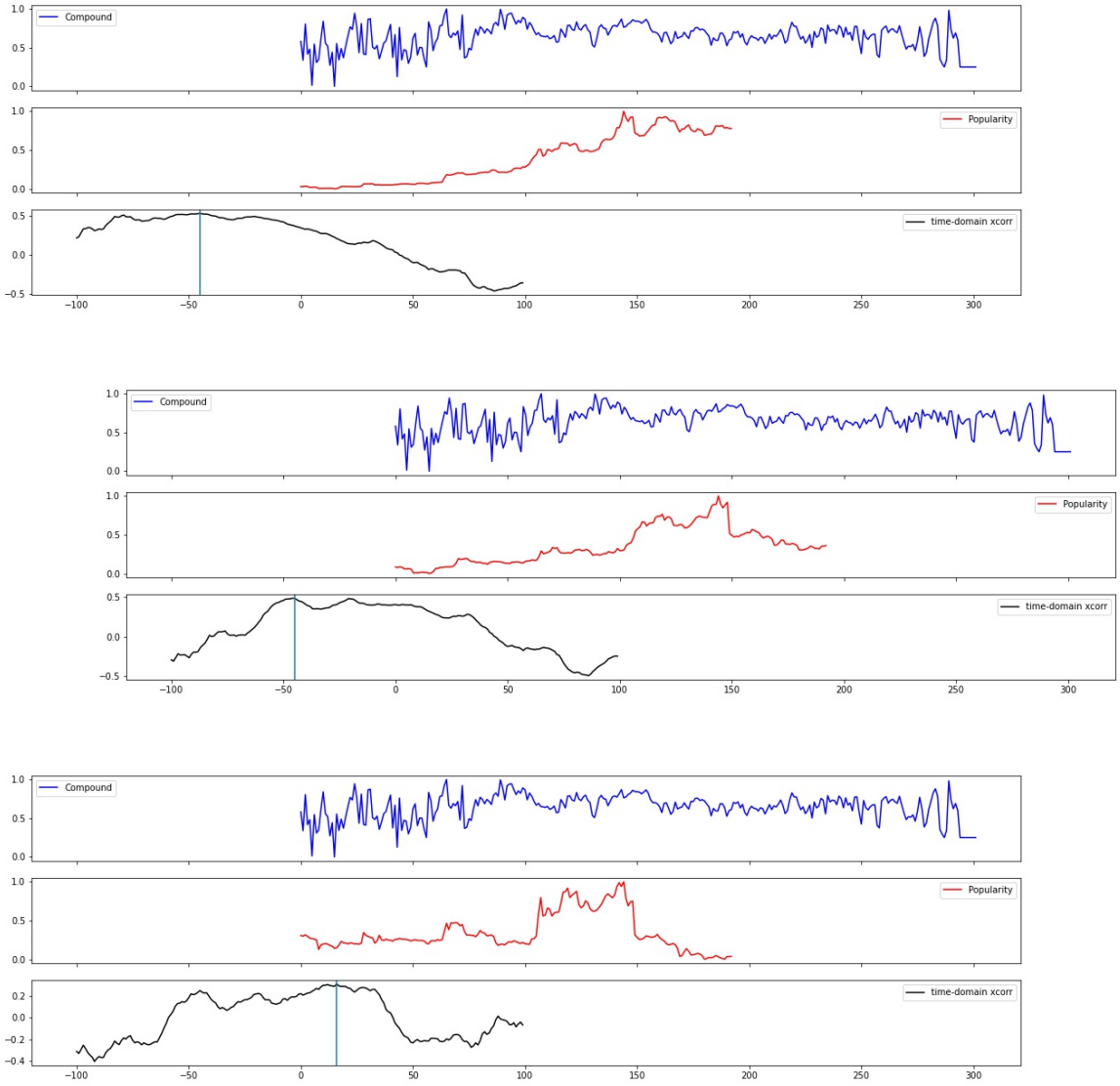
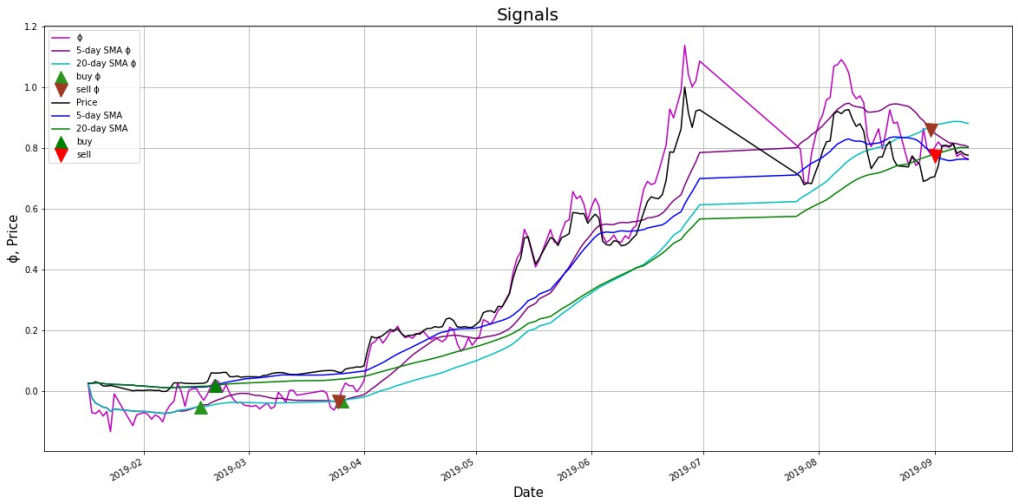


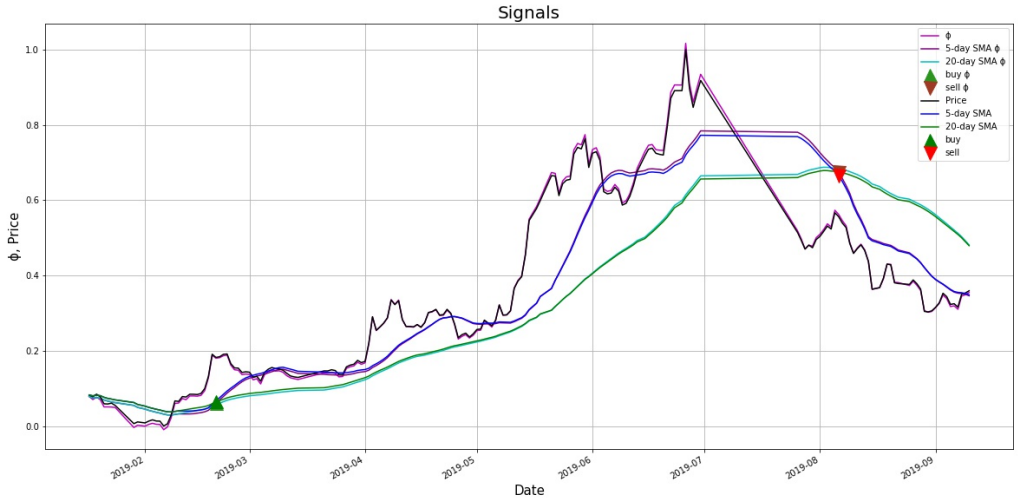
Figure A.5 Telegram Compound vs, from top to bottom:

1. BTC: Lag = -45, Crosscorr = 0.5250229444635333
2. ETH: Lag = -45, Crosscorr = 0.4859437349514198
3. XRP: Lag = 16, Crosscorr = 0.3064529539799354



Figure A.6 Predicted values from the telegram source for, from top to bottom, Bitcoin, Ethereum and Ripple.
In orange, prediction with date dummies; in green, prediction without date dummies.





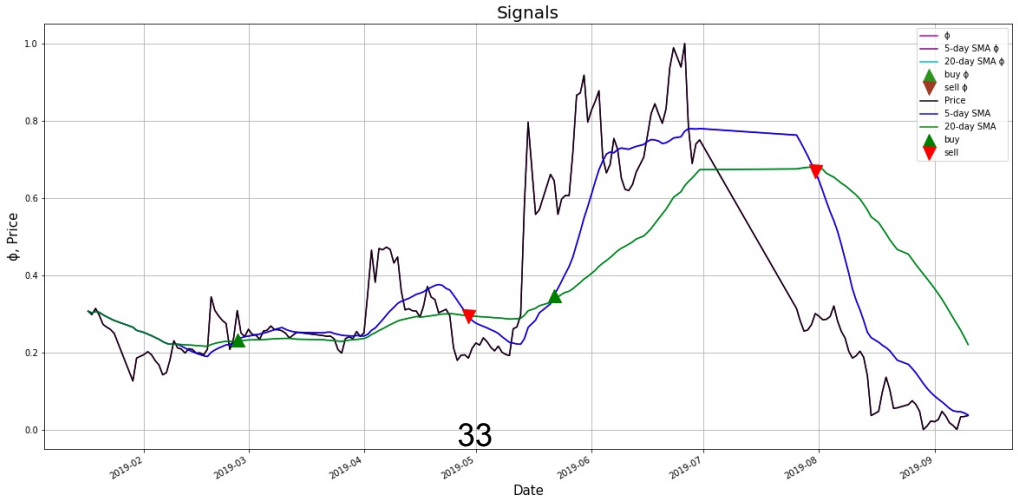


Figure A.7 Signals generated for, from top to bottom, BTC, ETH, XRP. ϕ is calculated using sentiment scores from Telegram

A.0.4 Twitter

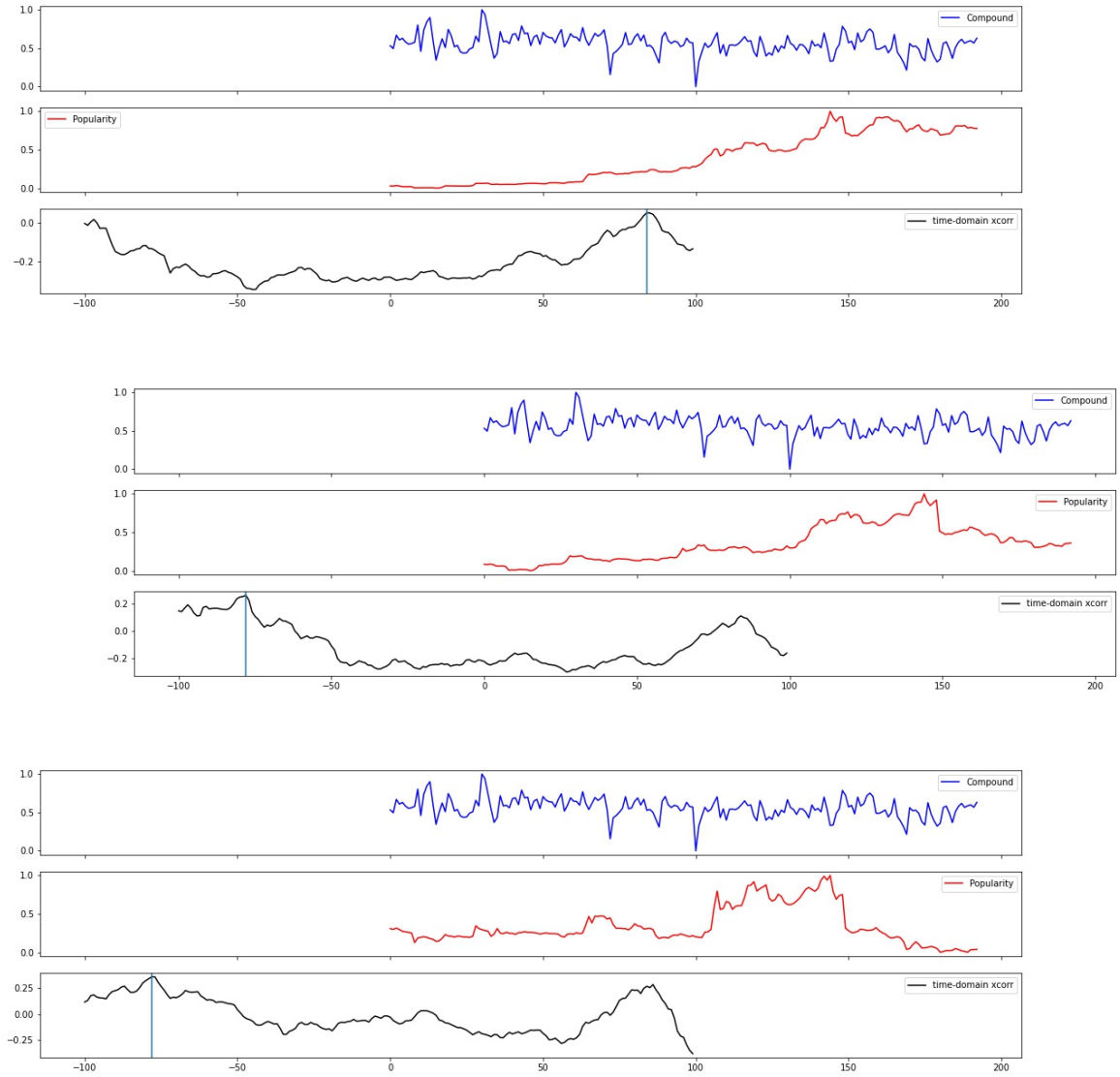


Figure A.8 Twitter Compound vs, from top to bottom:

1. BTC: Lag = 84, Crosscorr = 0.052187660698606794
2. ETH: Lag = -78, Crosscorr = 0.26319474083570676
3. XRP: Lag = -78, Crosscorr = 0.35616363251122485



Figure A.9 Predicted values from the twitter source for, from top to bottom, Bitcoin, Ethereum and Ripple.
In orange, prediction with date dummies; in green, prediction without date dummies.

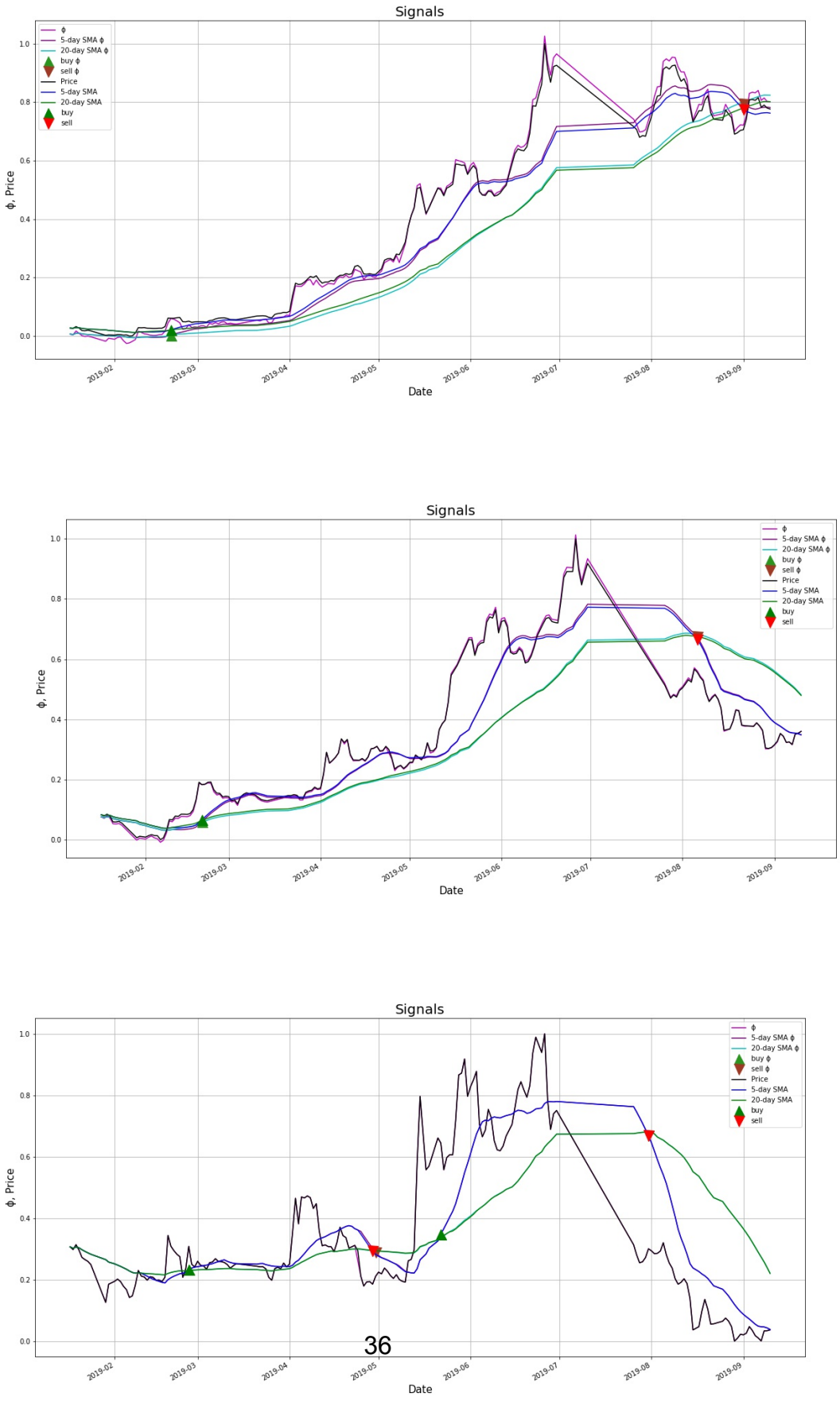


Figure A.10 Signals generated for, from top to bottom, BTC, ETH, XRP. ϕ is calculated using sentiment scores from Twitter