

# SentimentalCrypto: Classification of Sentiment of the Cryptocurrency market and Clustering analysis

Omkar Bhutra

LIU ID: omkbh878

Course code: 732A92

Statistics and Machine Learning

March 2020

## Abstract

Most Cryptocurrencies use Blockchain technology to record and distribute ledgers of transactions. The largest by market capitalisation are Bitcoin and Ethereum.

There exist over 5000 cryptocurrencies, most of which are widely speculated and traded over a large number of exchanges online. This project aims to build a sentiment classification tool to establish the current sentiment of the cryptocurrency market based on twitter feed. Further, Clustering analysis is performed to better understand the twitter corpus

320K tweets that were segmented by emotions: *anger, fear, greed, hateful, joy, sadness* were used to train our classifiers after preprocessing and classification training was done using Multinomial Naive Bayes, Logistic Regression, Stochastic Gradient Classifier and method with highest accuracy were chosen for further use. Valence Aware Dictionary and sEntiment Reasoner (VADER) was used to generate a polarity score and clustering analysis was performed on twitter corpus using K-Nearest Neighbors.

Finally, the chosen classifier was applied on completely unseen data pulled from Twitter using the Tweepy API to classify market sentiment as on 7th March 2020. Our classifier showed that the market was largely filled with 'Greed' and 'Fear' with a small proportion of 'Hateful'. On 15th March 2020, 'fear' had increased and mixed emotions of 'sadness' and 'joy' were also present.

This project of classification of the current market sentiment was performed during the marketwide sell-off with drop in Bitcoin's price from 9000\$ to 4000\$ amidst the n-covid 19' pandemic where even global stock markets experienced major declines in sentiment.

## 1 Introduction

A Blockchain is a growing list of records called *blocks* that are linked using concepts of cryptography. Such a list is distributed to all users participating in the network and thereby establishing decentralised trust. [1] As everything else in our world we order things and place them in different categories. Bitcoin is the first cryptocurrency that was invented in 2009 by a person who went by the pseudonym 'Satoshi Nakamoto' on Reddit. After the successful launch of the network and with enough network participants.

The task was to build a system for classification of current market sentiment of the cryptocurrencies. The topic of research was to observe what kind of classification models work best for such a corpus. The aim of the study was that this system should be able to aid trader's or investor's in making informed decisions.

After cryptocurrencies such as Bitcoin and Ethereum caught mainstream attention in 2017 and speculation on both its value and technology increased greatly, using a tool to monitor the sentiment of the cryptocurrency market can prove useful in making better investment decisions. Twitter is widely used platform where notable investors and developers of blockchain projects present their work and ideas. Tweeter can sometime's trigger market movements since they are often the first indications of press releases or announcements. Further, Clustering analysis was performed on the corpus of tweets to better understand our dataset and observe if there exists any explainable clusters with common knowledge of the blockchain space.

## 2 Theory

### 2.1 VADER Sentiment Intensity Analysis

Valence Aware Dictionary and sEntiment Reasonor is a lexicon and rule-based sentiment analysis tool that is specifically tuned to sentiment in online social media text. It is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. All lexical features of existing well-established and human validated sentiment lexicons along with additional lexical features that is used to express sentiment in social media text (emojis, acronyms, slang) is used. A wisdom-of-the-crowd approach to establish point estimations of sentiment valance for each of the 9000+ lexical feature candidates and then keeps 7500+ lexical features with mean valance close to 0 and standard deviation less than 2.5 as a human validated gold standard lexicon. Generalizable heuristics of the assessing sentiment in text is identified iteratively and point estimates of sentiment valance on the corpora from separate domains. [2]

*VADER has been found to be quite successful when dealing with social media texts, NY Times editorials, movie reviews, and product reviews* [3]

The valence score of a sentence is calculated by summing up the valence scores of each VADER-dictionary-listed word in the sentence. Cautious readers would probably notice that there is a contradiction: individual words have a valence score between -4 to 4, but the returned valence score of a sentence is between -1 to 1.

They're both true. The valence score of a sentence is the sum of the valence score of each sentiment-bearing word. However, we apply a normalization to the total to map it to a value between -1 to 1. [4]

The normalisation rule is: where  $x$  is the sum of valance scores and  $\alpha$  is the normalization parameter that is set to a constant. 
$$\frac{x}{\sqrt{x^2 + \alpha}}$$

### 2.2 Naive Bayes

Multinomial Naive Bayes build a probabilistic model that uses arbitrary features from documents. When training the model features are extracted from different documents and are given to the model together with the class labels. For each feature and class the probability in formula 1 is calculated where  $feat_i$  is feature  $i$  and  $class_k$  is class  $k$ .

$$p(feat_i|class_k) \tag{1}$$

When predicting the class of a document the same features are extracted and given to formula 2 where there exist  $K$  classes and  $n$  features for each document. The probability of the class is multiplied by all features explained in formula 1. The predicted class  $k$  is the  $k$  that maximizes the expression.

$$\hat{y} = \underset{k \in \{1,2,\dots,K\}}{\operatorname{argmax}} \quad p(\text{class}_k) \prod_{i=1}^n p(\text{feat}_i | \text{class}_k) \quad (2)$$

The strength of the Naive Bayes classifier is that the features are arbitrary. The model determine the importance for all features and let informative features influence the prediction as well as assign uninformative features with low probabilities. [5] [6]

## 2.3 Logistic Regression

Logistic regression (LR) is useful in many areas such as document classification and natural language processing (NLP). It models the conditional probability as:

$$Pw(y = 1|x) = 1/1 + e^{ywTx} \quad (3)$$

where  $x$  is the data,  $y$  is the class label, and  $w$  is the weight vector. Given two-class training data  $(x_i, y_i)_{i=1}^l, x_i \in R^n, y_i \in \{1, -1\}$

Logistic regression minimizes the following regularized negative log-likelihood:

$$P^{LR}(w) = C \sum_{i=1}^l \log(1 + e^{-y_i w^T x_i}) + \frac{1}{2} w^T w \quad (4)$$

[7]

## 2.4 Stochastic Gradient Descent

This estimator implements regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (learning rate). SGD allows minibatch (online/out-of-core) learning. For best results using the default learning rate schedule, the data should have zero mean and unit variance. [8]

This implementation works with data represented as dense or sparse arrays of floating point values for the features. The model it fits can be controlled with the loss parameter; by default, it fits a linear support vector machine (SVM). [9]

## 2.5 Classification report - Accuracy, precision , recall , f1-score and support

Precision and recall ‘zoom in’ on how good a system is at identifying documents of a specific class . Equations 5 and 6 are with respect to the positive class.

- **Precision** - Precision is the proportion of correctly classified documents among all documents for which the system predicts class .

$$Precision = \frac{\text{no.of Truepositives}}{\text{no.Truepositives} + \text{no.Falsepositives}} \quad (5)$$

When the system predicts class , how often is it correct?

- **Recall** - Recall is the proportion of correctly classified documents among all documents with gold-standard class .

$$Recall = \frac{no.of Truepositives}{no.Truepositives + no.Falsepositives} \quad (6)$$

When the document has class , how often does the system predict it?

- **F1-measure** - A good classifier should balance between precision and recall. The F1-measure is the harmonic mean of the two values:

$$Recall = \frac{2.precision.recall}{precision + recall} \quad (7)$$

[5]

## 2.6 Clustering - k-Means

- **k-means** - The k-means algorithm aims to partition a document collection into clusters, minimising within-cluster variance in distance. distance variance = squared Euclidean distances. Each document, represented by its vector, will be put into the cluster with the nearest centroid (mean).
- **Centroids and medoids** - The centroid of a cluster is the arithmetic mean of the document vectors in the cluster, not necessarily the vector of an actual document. The medoid of a cluster is a vector in the cluster whose average distance to all the other vectors is minimal, not the same as a geometric median

Issues with the k-means algorithm: - The k-means algorithm always converges, but there is no guarantee that it finds a global optimum. Solution: random restarts - The number of clusters needs to be specified in advance, or chosen based on heuristics and cross-validation. Example: elbow method - The k-means algorithm is not good at handling outliers – every document will eventually belong to some cluster. [5]

## 3 Data

### 3.1 Emotions in Cryptocurrency related tweets

The following criterions were considered when looking for a corpus of tweets:

- **Class labels** - Data tagged with a gold standard class label tagged by humans .
- **Reliable** - The compiler behind the dataset should have good score on kaggle.
- **Large corpus** - Covers tweets from a large timeframe to include tweets from varied market conditions, atleast 2 years
- **Volume of data** - To achieve a reliable classification we must have atleast a few hundred thousand tweets.

### 3.2 Training and Testing Twitter data

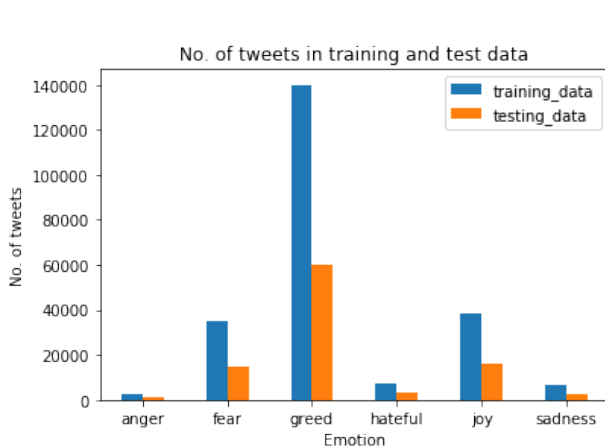
The datasets for each of the 6 emotions are taken and merged. [10] "bitcoin" and "crypto" are the filters used to make 6 datasets by the author, *anger*, *greed*, *fear*, *hateful*, *joy* and *sadness* are the emotions that are used as class labels. This data is merged and a column for class labels is created which are gold standard class labels as they are human verified sentiments in tweets related to the cryptocurrency market. The data is preprocessed and cleaned by the same process as for the training and testing dataset. see 4 for steps in preprocessing.

Table 1 shows a subset of the preprocessed string of textual data from the tweets which is used by the classification models.

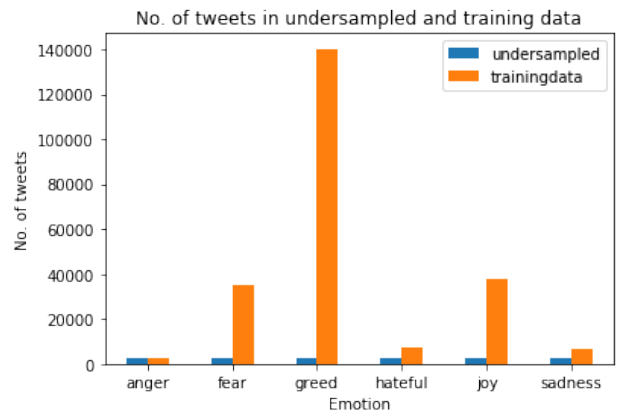
id	Preprocessed text	emotion
100001	digitalvillain tech company still care ai iot ...	greed
100002	commenters saying could pay lower fee wait mis...	greed
100003	senior research scientist robotics startup cyb...	greed
100004	current top dapps volumelastday kyber index for...	greed
100005	cinsbit pleased announce fortem capital token ...	greed
...	...	...
328358	zackvoell digital gold every function gold bet...	sadness
328359	never see adoption principle privacy fungibili...	sadness
328360	blockbits incent target broken economy reward ...	sadness
328361	sad cant proud watching world getting decentra...	sadness
328362	sad thing sheep buy po	sadness

Table 1: Preprocessed string from the tweets

Figure 1 shows the number of tweets by all emotions. This dataset is used for training and testing of the model after randomized shuffling and a split with a ratio of 70/30 for training vs test data. In figure 1a shows the training and test split 1b shows the undersampled data vs the training data , by emotion.



(a) Number of tweets in training and test data



(b) Number of tweets in undersampled and training data

Figure 1: Histograms of number of tweets by emotions

### 3.3 Current Twitter data

Tweepy is used along with the developer's API for twitter to bring current data to classify using our model for the purpose of use in trading and investing. [11] cite4

Future work with this dataset can include this data into the testing of the classifier after labelling this dataset with an emotion class manually. 7

Using the API a search query "bitcoin" and "crypto" is sent with the current date, Retweets are filtered to avoid duplicates in our corpus. The data is compiled by user, location and the tweet text. The data is preprocessed and cleaned by the same process as for the training and testing dataset. see 4 for steps in preprocessing.

Figure 2 shows current stream of tweets tagged by 'bitcoin' or 'crypto'



	user	location	text
0	KrakenPrices		 Prices update in \$USD (1 hour):\n\n\$BTC - 91...
1	EmilyHyipNews		TODAY PAYING HYIPS - 7/03/2020! EXWAY and DEXA...
2	coinpricenow		1 Bitcoin ( #BTC )\nDollar: 9119.84\$ \n\n1 Bit...
3	Rakamoto	Worldwide	The #crypto #ecosystem tends to see regulation...
4	Rakamoto	Worldwide	Within the #crypto #community, debates have be...
...	...	...	...
95	cryptoWhisper	somewhere	#Bitcoin #Crypto #BTCUSD \n9600 in coming days...
96	betbybitcoins		07 March 2020 Betting Tips <a href="https://t.co/nnKvbH...">https://t.co/nnKvbH...</a>
97	WiseAnalyze		While H&S still possible, #BTC draws anoth...
98	ElixiumCapital	England, United Kingdom	This Bitcoin chart is insane! Oh, wait... that's...
99	official_ckm	London, United Kingdom	 Did you know what Satoshi Nakamoto means? \n\...
100 rows x 3 columns			

Figure 2: Table of current tweets with no class labels, by user , location and text in the tweets

### 3.4 Code

The complete code for this project can be found at GitHub. [12] Documentation on how to retrieve the corpus and run the system can be found in the repository. Future work and updates on this project can be found on the same github repository.

## 4 Method

- **Dataset** - Tweets were segmented by emotions like *anger*, *fear*, *greed*, *hateful*, *joy* and *sadness*. This data was taken from Kaggle and it consisted around 320,000 tweets in separate datasets for each emotion and it is an unbalanced raw dataset of tweets from which class labels for emotions are generated into a single large dataset. [10] [13]
- **Data preprocessing** - The tweets were preprocessed in the following order: punctuation removal, link/url removal and tokenization using the library 're', stopwords removal and removal of some specific words that to reduce noise using the package 'nltk' [14], Stemming / Lemmatisation. [15] [3]. Wordclouds of the cleaned string of words are produced for each 'emotion' class.
- **Sentiment Intensity** - The VADER algorithm was applied only on the punctuation cleaned tweets and this gives the a polarity index between -1 and 1 for each tweet. -1 representing extreme negative sentiment and +1 representing extreme positive sentiment. This is visualised by a violin plot of the polarity score on the y-axis, named 'compound' the dataset Vs the 6 'emotion' classes on the x-axis. The library 'vaderSentiment' is used and the 'SentimentIntensityAnalyzer' class is initialized and only the punctuation removed tweets[2]
- **Training and Testing data** - The corpus of tweets in the training and test data consisted each tweet is has a class label one of 6 'emotion' types. The corpus was shuffled after importation. The seed for the random state function is initialized with a constant value of 42 to ensure reproducibility.
- **Classification** - Multinomial Naive Bayes, Logistic Regression and Stochastic Gradient Descent Classifier are used during the modelling process. The method which produces the highest accuracy is chosen for further predictions.
- **Prediction of the classifier on current tweets** - 500 Current tweets are streamed through the *tweepy API*, Preprocessed in the same way as the training and test dataset, VADER sentiment is calculated and prediction of the class label is done using the classifier with the highest accuracy.

### 4.1 Wordclouds

Wordclouds are generated for the purpose of visualisation of the preprocessed tweets that are used in the classifiers. The wordclouds for all emotions as well as by each emotion are produced using the library *wordcloud* using the functions *WordCloud*, *ImageColorGenerator*.

- **Maximum words** - 100
- **Interpolation** - bilinear

### 4.2 Sentiment Intensity

Punctuations are removed from all the tweets and this data is used in the VADER Sentiment Intensity analyser.

Table 2 shows a subset of the only the punctuation removed text from tweets which is used by the VADER sentiment intensity analyser.



id	Punctuation removed text	emotion
100001	DigitalVillain Tech companies still care mo...	greed
100002	All the commenters saying he could pay a lower...	greed
100003	Senior Research Scientist Robotics Startup C...	greed
100004	current top dapps volumelastday kyber idex for...	greed
100005	cinsbit pleased announce fortem capital token ...	greed
...	...	...
328358	zackvoell Bitcoin is digital gold it does ever...	sadness
328359	Bitcoin BTC will never see adoption if princip...	sadness
328360	BlockBits Incent Targets Broken Economy With B...	sadness
328361	So sad that I cant be there but so proud to be...	sadness
328362	Sad thing is sheep will buy into this POS ...	sadness

Table 2: Punctuation cleaned from the tweets

The polarity scores vs emotion type for all the tweets are visualised in a violin plot. This includes summary statistics of the data, the box plot, interquartile ranges, density plot.

### 4.3 Baseline

The classifier was initialized with the randomly shuffled corpus and the data was split up into training data and test data. The default values were 70% training data and 30% test data. The baseline accuracy is calculated as the average number of cases where the mode of emotion types of the training data are equal to the emotion types in the test data.

### 4.4 Classification

Multinomial Naive Bayes, Logistic Regression and Stochastic Gradient Descent Classifier are used during the modelling process. The method which produces the highest accuracy is chosen for further use in the study.

- **Naive Bayes** - Multinomial Naive Bayes is applied on the training data in a pipeline with *CountVectorizer* and *MultinomialNB* functions from the *sklearn* library , the overall accuracy, classification report and the confusion matrix are produced. Undersampling without replacement is performed based on the a random choice function from the *numpy* library and a balanced dataset is produced to calculate the undersampled accuracy of the Naive Bayes model. GridSearch Cross Validation is performed on the Naive Bayes model to see if the results are consistent. Other summary statistics such as precision, recall, f1-score and support are also shown in the classification reports.
- **Logistic Regression** - Logistic Regression is applied on the training data in a pipeline with *CountVectorizer* and *LogisticRegression* functions from the *sklearn* library , the overall accuracy, classification report and the confusion matrix are produced.
- **Stochastic Gradient Descent** - *SGDClassifier* is applied on the training data in a pipeline with *CountVectorizer* and *LogisticRegression* functions from the *sklearn* library, lbfgs optimality is used [7] .The overall accuracy, classification report and the confusion matrix are produced. Grid Search Cross Validation is performed for the logistic regression model

## 4.5 Clustering - k-means

Hard clustering is done using k-means from the *sklearn* library. The elbow point is analysed upto  $k = 10$  iteratively. The elbow method lead us to check for  $k = 2$  and  $8$ .

We can compute rand indices since we have gold standard class labels. The rand index and adjusted rand score is calculated for both the chosen  $k$  cases. 5 terms with highest centroid values are identified for each cluster in both the cases.

## 5 Results

The results will focus mainly on comparing two different genres or comparing all genres. For all tests the split in training and test is 70% and 30% respectively. Cross validation is 5-fold. All other arguments use the default values. Between different tests the random seed was reset which makes the results reproducible.

Performance for the baseline system is shown in table 3.

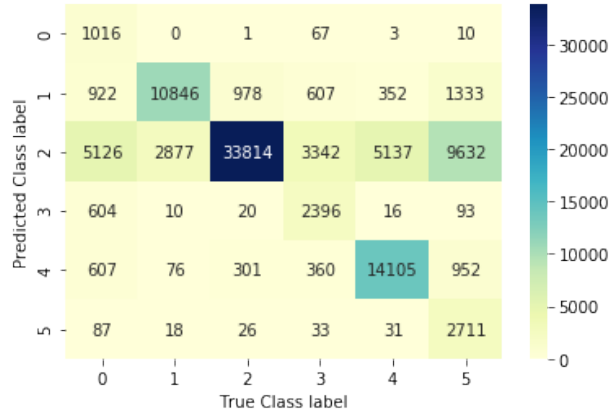
### 5.1 Baseline

Classification report for the model is given in the table 3. A baseline accuracy of 60.83% is achieved.

emotion	precision	recall	f1-score	support
anger	0.12	0.93	0.21	1097
fear	0.78	0.72	0.75	15038
greed	0.96	0.56	0.71	59928
hateful	0.35	0.76	0.48	3139
joy	0.72	0.86	0.78	16401
sadness	0.18	0.93	0.31	2906
accuracy			0.66	98509
macro avg	0.52	0.79	0.54	98509
weighted avg	0.84	0.66	0.70	98509

Table 3: Classification report of the Naive Bayes model

The Confusion Matrix is given in the figure 3 *greed* has the highest number of true positives, followed by *joy* and *fear*



(a) Confusion Matrix for the Balanced dataset

Figure 3: Confusion Matrix

## 5.2 Most common words by emotion

Figure 4 in *a-e* shows the show most common words by emotion *anger*, *fear*, *greed*, *hateful*, *joy* and *sadness*



Figure 4: Wordclouds from the tweets by emotion

## 5.3 VADER

A sentiment intensity score is produced for each tweet based on only punctuation removed tweets without the use of class labels. 4.2



(a) VADER Sentiment Intensity vs Emotion

Figure 5: Violin Plot

anger	fear	greed	hateful	joy	sadness	Summary
-0.30	0.10	0.20	-0.27	0.71	-0.07	mean
-0.49	0.29	0	-0.44	0.765	-0.20	median
-0.55	0.29	-0.003	-0.58	0.62	-0.29	mode (density plot)
-0.65	0	0	-0.63	0.62	-0.45	1st Quantile (box plot)
0	0.296	0.45	0	0.85	0.29	3rd Quantile (box plot)

Table 4: Summary statistics from the violin plot

This score is specifically designed to analyze the sentiment from social media text. Summary stat are given in the table 4 and the plot is shown in figure 5

## 5.4 Naive Bayes

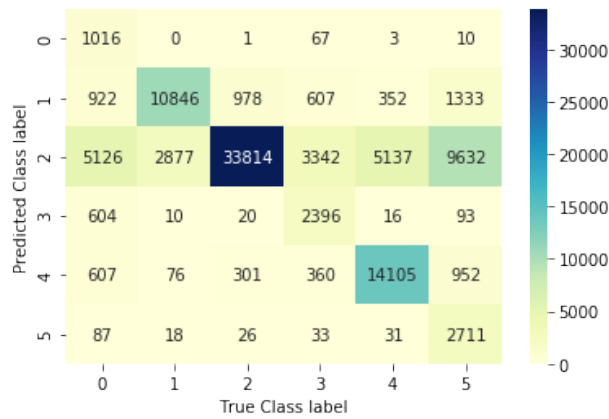
The classifier achieved an accuracy of 85.86%. 5 fold Cross validation, which marginally improve the performance.

Classification report for the model is given in the table 5.

The Confusion Matrix is given in the figure 6 *greed* has the highest number of true positives, followed by *joy* and *fear*

emotion	precision	recall	f1-score	support
anger	0.84	0.08	0.15	1097
fear	0.85	0.71	0.81	15038
greed	0.85	0.96	0.90	59928
hateful	0.79	0.45	0.57	3139
joy	0.85	0.86	0.86	16401
sadness	0.94	0.24	0.38	2906
accuracy			0.86	98509
macro avg	0.87	0.55	0.61	98509
weighted avg	0.86	0.86	0.85	98509

Table 5: Classification report of the Naive Bayes model



(a) Confusion Matrix for the Naive Bayes model

Figure 6: Confusion Matrix

## 5.5 Logistic Regression

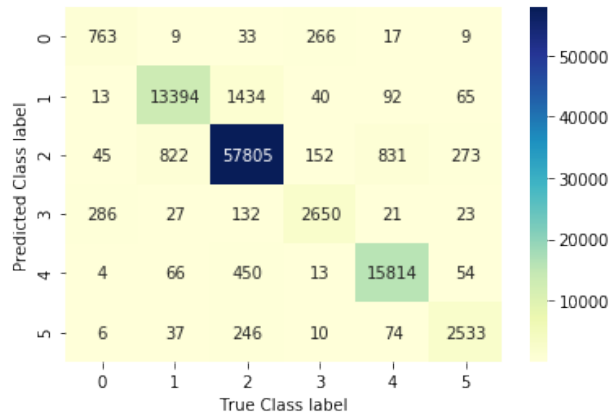
The accuracy of the model achieved is 94.36%.

Classification report for the model is given in the table 6.

The Confusion Matrix is given in the figure 7 *greed* has the highest number of true positives, followed by *joy* and *fear*

emotion	precision	recall	f1-score	support
anger	0.68	0.70	0.69	1097
fear	0.93	0.89	0.91	15038
greed	0.96	0.96	0.96	59928
hateful	0.85	0.84	0.85	3139
joy	0.94	0.96	0.95	16401
sadness	0.86	0.87	0.86	2906
accuracy			0.94	98509
macro avg	0.87	0.87	0.87	98509
weighted avg	0.94	0.94	0.94	98509

Table 6: Classification report of the Naive Bayes model



(a) Confusion Matrix for the Logistic Regression model

Figure 7: Confusion Matrix

## 5.6 Stochastic Gradient Descent

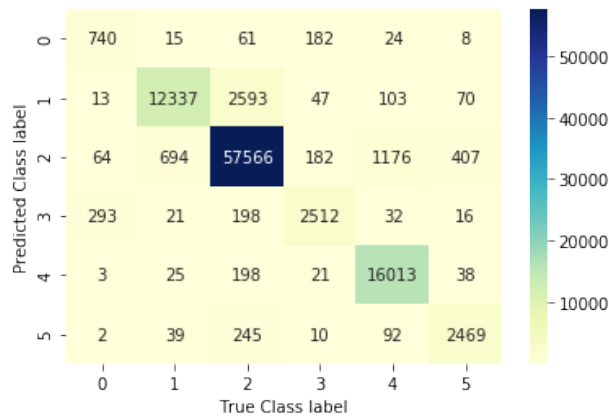
The accuracy of the model achieved is 93.02%.

Classification report for the model is given in the Table 7.

The Confusion Matrix is given in the Figure 8 *greed* has the highest number of true positives, followed by *joy* and *fear*

emotion	precision	recall	f1-score	support
anger	0.66	0.72	0.69	1030
fear	0.94	0.81	0.87	15163
greed	0.95	0.96	0.95	60089
hateful	0.85	0.82	0.83	3072
joy	0.92	0.98	0.95	16298
sadness	0.82	0.87	0.84	2857
accuracy			0.93	98509
macro avg	0.86	0.86	0.86	98509
weighted avg	0.93	0.93	0.93	98509

Table 7: Classification report of the Naive Bayes model



(a) Confusion Matrix using Stochastic Gradient Descent

Figure 8: Confusion Matrix

5-fold cross validation allows for tuning of the model. Best score: 0.929 Best parameters set:  $\alpha = 1e - 05$ , max iterations = 20 ,penalty = 'elasticnet', tfidf idf use = False, vect.max.df = 0.5 ,n-gram range = (1, 2)

## 5.7 Prediction of current sentiment

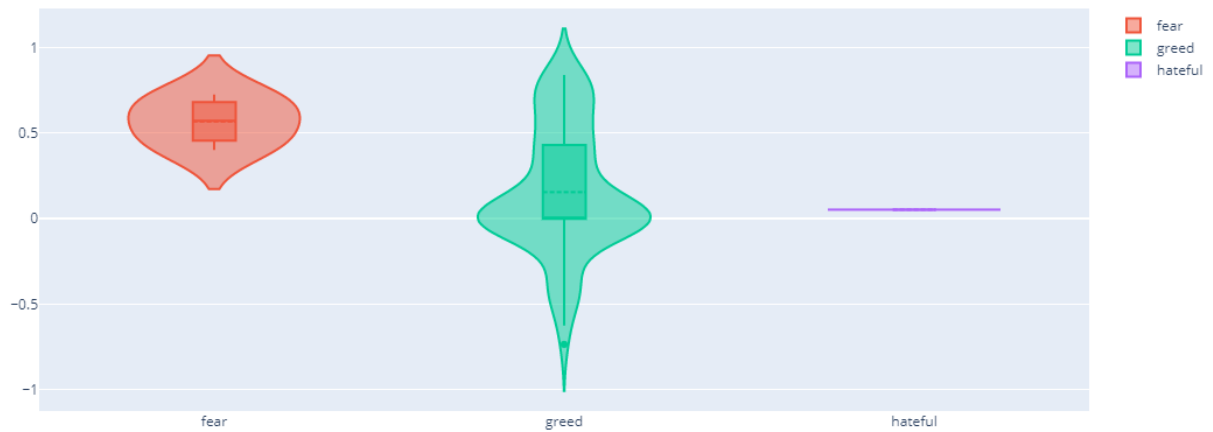
100 tweets filtered by the tags '*Bitcoin*' and '*Crypto*' are streamed using the api. 4 Preprocessing is performed in the same way as training and test dataset. VADER sentiment is calculated and trained Logistic Regression classifier is used to classify the current emotion of the market.

	user	location	text	compound	predicted_sentiment
0	moncef_fahim	New York, NY	want trade pm move webull get free stock free...	0.7845	greed
1	Crypto556		thanks follower make many giveaway future altc...	0.4404	greed
2	Mattsolid87		tried get cash bbt atm yesterday apparently cl...	0.0000	greed
3	cryptzos		middleman support k starting take serious blow...	0.3400	greed
4	DomZilliqa	Republic of Slovenia si	look like well busy week zilliqans update tomo...	0.7269	greed
...	...	...	...	...	...
495	gridnetproject		wouldnt gridnetproject good candidate opened o...	-0.5875	greed
496	Drens	Mars	bonus bet bet bet interwetten bahis iddaa	0.5423	greed
497	Visible_Banking	London	report jpmorgan perspective digitalcurrency	0.0000	greed
498	cryptotothemoo1	South East, England	declining open interest cme future led trader ...	0.4588	greed
499	AdapoolsO		real realdada live stake ada share estimate r...	0.7003	greed

500 rows x 5 columns

(a) VADER sentiment intensity in current stream of tweets

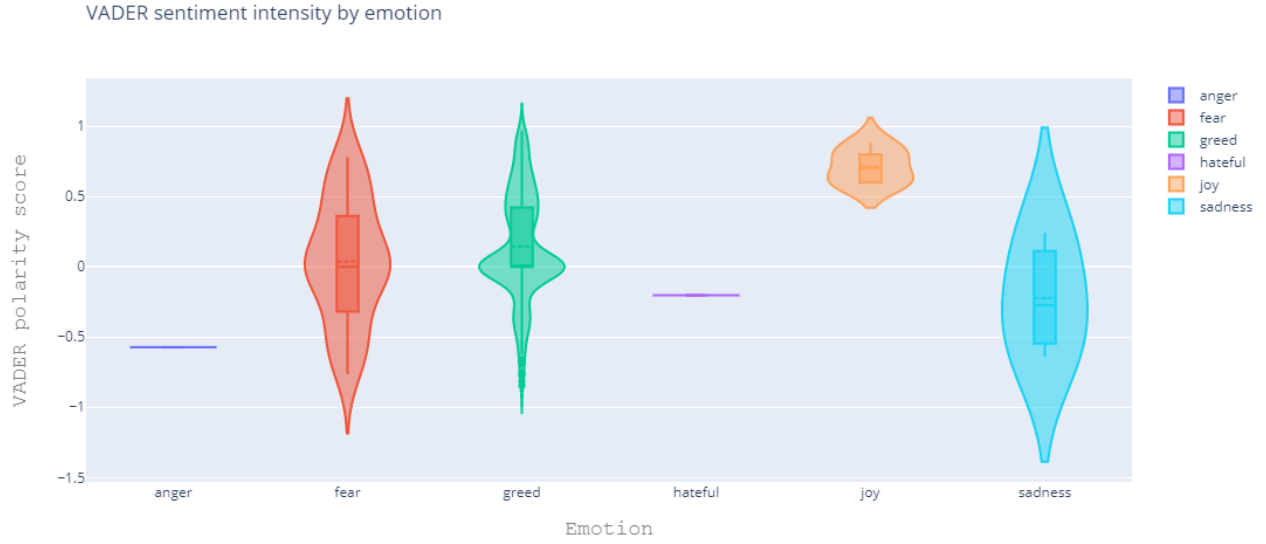
Figure 9: Sentiment intensity



(a) VADER Sentiment Intensity vs Emotion - March 7, 2020

Figure 10: Violin plot





(a) VADER Sentiment Intensity vs Emotion - March 15, 2020

Figure 11: Violin plot

Summary stat are given in the table 8 and the plot is shown in figure 11

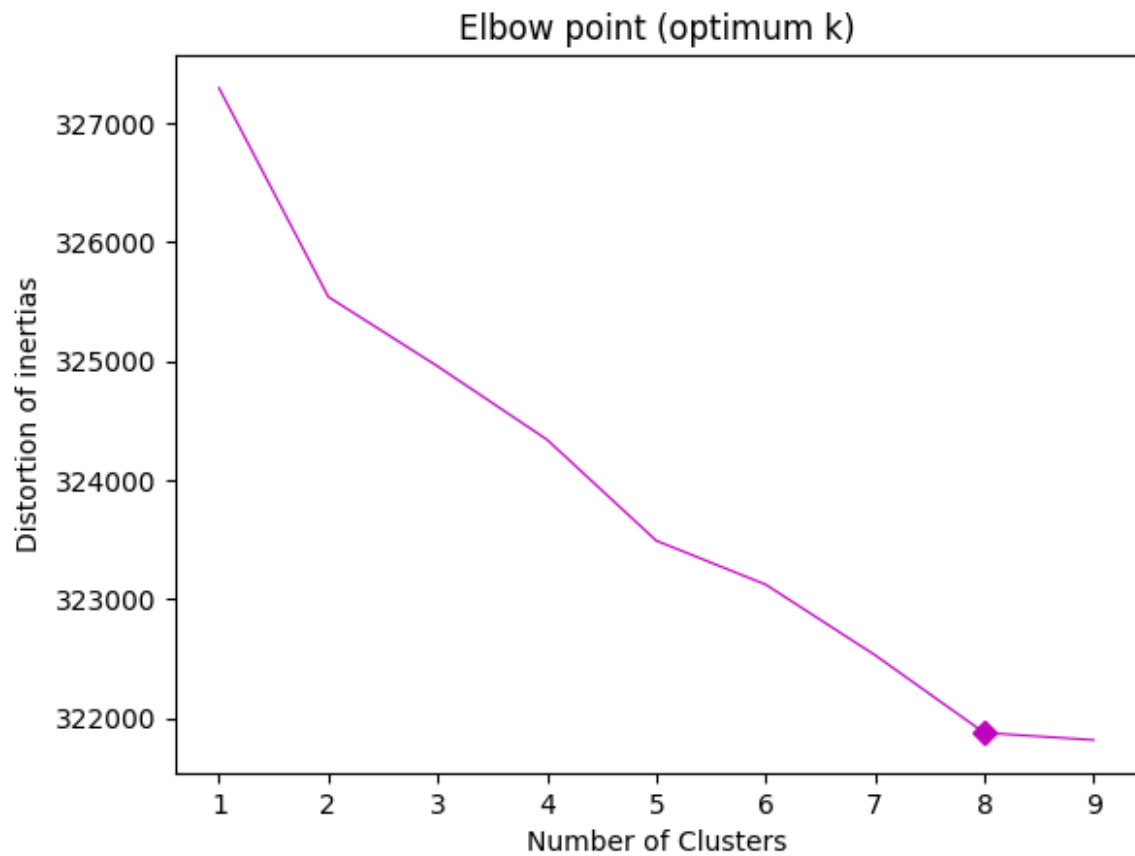
anger	fear	greed	hateful	joy	sadness	Summary
-0.59	0.057	0.14	-0.2	0.71	-0.22	mean
-	0	0		-	-0.70	-0.27
median						
-	0.225	-0.004	-	0.62	-0.31	mode (density plot)
-	-0.32	-0.005	-	0.60	-0.55	1st Quantile (box plot)
-	0.40	0.42	-	0.802	0.11	3rd Quantile (box plot)

Table 8: Summary statistics from the violin plot

predicted sentiment	count
anger	1
fear	40
greed	451
hateful	1
joy	5
sadness	2

Table 9: Count of predicted classes on current tweets

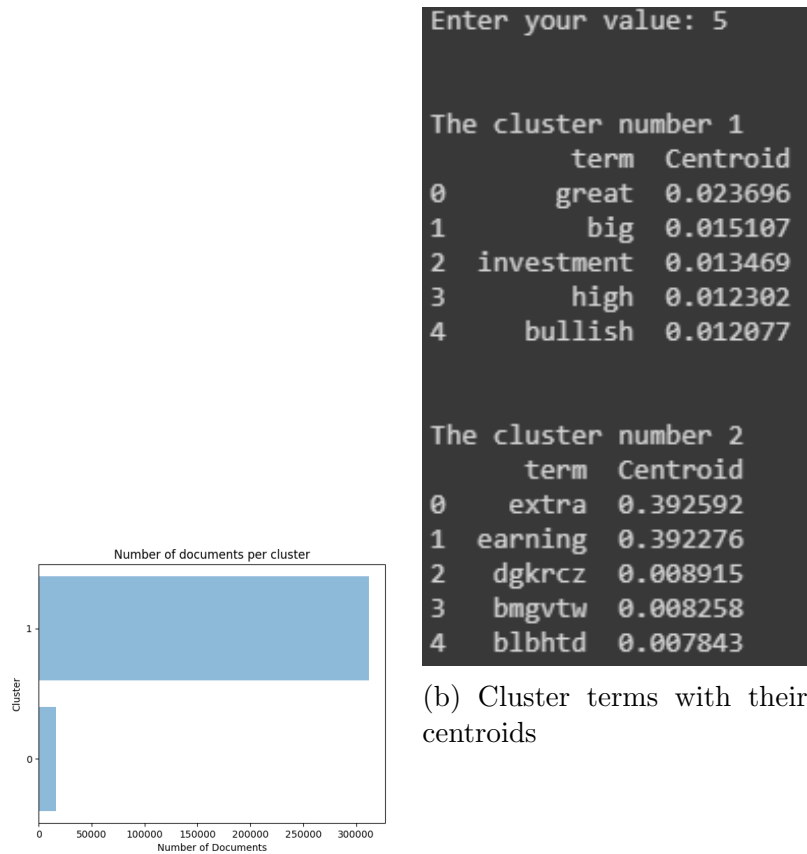
## 5.8 Clustering - k-means



(a) Elbow method to find optimum k

Figure 12: Elbow point

Three possible values for k can be chosen. 2, 5 and 8. All are attempted but 8 provides with coherent terms in its clusters.



(a) Number of terms in clusters

```
Enter your value: 5
```

The cluster number 1

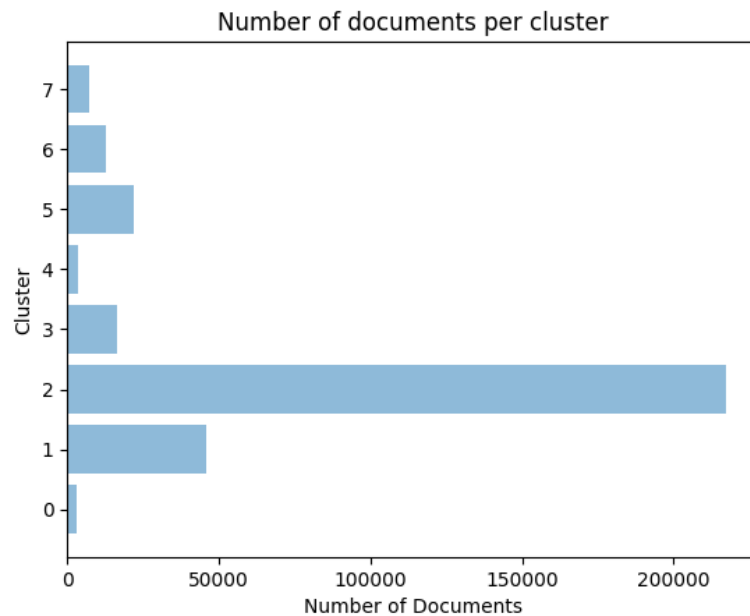
	term	Centroid
0	great	0.023696
1	big	0.015107
2	investment	0.013469
3	high	0.012302
4	bullish	0.012077

The cluster number 2

	term	Centroid
0	extra	0.392592
1	earning	0.392276
2	dgkrcz	0.008915
3	bmgvtw	0.008258
4	blbhtd	0.007843

(b) Cluster terms with their centroids

Figure 13:  $k = 2$



(a) Number of terms in clusters

The cluster number 1		
	term	Centroid
0	tron	0.208328
1	trx	0.140114
2	sun	0.104703
3	justin	0.104064
4	btt	0.034168

The cluster number 2		
	term	Centroid
0	great	0.155437
1	good	0.015950
2	future	0.013456
3	idea	0.013038
4	news	0.012944

The cluster number 3		
	term	Centroid
0	big	0.020329
1	high	0.014746
2	pump	0.013516
3	price	0.010538
4	profit	0.009615

The cluster number 4		
	term	Centroid
0	extra	0.392616
1	earning	0.392275
2	dgkrcz	0.008915
3	bmgvtw	0.008258
4	blbhtd	0.007843

(a) Cluster terms with their centroids

The cluster number 5		
	term	Centroid
0	hour	0.271445
1	change	0.229521
2	average	0.176746
3	overview	0.174964
4	hodling	0.152745

The cluster number 6		
	term	Centroid
0	investment	0.188003
1	ideafex	0.020215
2	money	0.019429
3	startup	0.018228
4	ico	0.016801

The cluster number 7		
	term	Centroid
0	bullish	0.284586
1	price	0.044098
2	june	0.033582
3	april	0.032607
4	analysis	0.020273

The cluster number 8		
	term	Centroid
0	united	0.116090
1	state	0.115973
2	developer	0.097108
3	engineer	0.076366
4	job	0.058155

(b) Cluster terms with their centroids

## 6 Discussion

Wordclouds by emotion type provide us with a good first look into the data. tweets labelled with *anger* have common words such as (*hate,suck,mad,angry*), with *fear* have common words such as (*short,panic,dump,doubt*) (In our data, 'short' would usually refer a short position that traders take when they bet on the price decreasing), with *greed* have common words such as (*pump,big,high,buy*), with *hateful* have common words such as (*hate,shit,bullshit,horrible*), with *joy* have common words such as (*great,happy,thank,announcement*), with *sadness* have common words such as (*low,sad,broken,sell*)

The results of the baseline system are found in table 3. Using all emotions, the baseline system have the accuracy 60.83%. Undersampling resulted in a balanced dataset, when tested with the Naive Bayes classifier produced a 67.37% accuracy.

The performance of Stochastic Gradient Descent was very close to that of Logistic Regression with 93.02% accuracy with precision, recall and f1-score's only marginally lower. The highest accuracy is achieved with the Logistic Regression Classifier of 94.26% and hence that is used to classify the current market sentiment. The lowest precision, recall and f1-score was produced for the class *anger* and the highest for class (*greed*). The bulk of the classified is emotion type (*greed*). The logistic regression method and the SGD classifier method are marginally different in their performance parameters , this can be seen in the confusion matrices in the figure ?? and figure 8.

The dataset is relatively unique and while comparing to results from other studies where VADER is used, it is noted that, other studies create *silver standard* class labels as compared to the human tagged class labels in the dataset used in this study. In the studies with the application of VADER , [16] Rochoz calculates the sentiment intensity by each cryptocurrency and studies how the index correlates with the price of that cryptocurrency. [17] J.Thesken uses the output of the sentiment index as inputs to a linear regression model. Similarly, In the paper by E.Linguist [18] the author has used bigrams, trigrams to identify suspicious bot tweets and remove them as part of preprocessing and performed used VADER sentiment to classify social media data related to cryptocurrencies in either positive, neutral or negative sentiment class.

It is observed in the figure 15 , the clustering with  $k = 8$  provides a distinct coherent cluster's. The first cluster's terms include *tron, trx, sun, justin, btt*. The terms include the name of the CEO, the trading tickers of the coins the subsidiary company owns (Tron and Bittorrent). The other cluster has terms related to positive news, increase in price, extra earnings, long term investment, raising capital via initial offerings / startup , analysis and final cluster has terms linked to employment in the cryptocurrency industry.

## 7 Conclusion

The VADER sentiment intensity is a good measure of sentiment from the tweets related to cryptocurrencies as the score's are indicative of the emotion class even though they are independently produced.

Logistic Regression and Stochastic Gradient Descent are both equally good classifiers in this problem case as they have marginal differences in performance statistics.

The cryptocurrency market is generally defined by *greed* and it is defined by high volatility and speculation.

Based on the results and subsequent analysis, it can be concluded that the cryptocurrencies market has negative sentiment and is defined by the entry of mixed emotions of *sadness and joy* on March 15, 2020 after these emotions being absent as on March 7, 2020. This is marked by the decrease in the price of Bitcoin from 8000\$ to 4000\$ in this timeframe.

A realtime indicator could be made as the classification runs every few minutes. This can aid a trader in making better decisions. Twitter does not allow for date manipulation while streaming tweets and this prevented me from performing a backtest and comparative analysis on the performance of *SentimentalCrypto* with respect to market conditions such as price of cryptocurrencies and trading volumes. I would keep in mind if I would work on a trading project again. A solution to this would be to acquire a large database of all tweets from the date we would like and not perform search queries for use in classification. Further work on this project can be to use the current twitter data into the training and testing process of the classifier.

Initially, Clustering by k-means showed less than optimal results. This was due to links still remaining in the corpus after the preprocessing. After this issue was resolved,  $k = 8$  provided distinct clusters but only a few clusters corresponded to sentiment of the market. Suggesting that clusters in the twitter corpus related to cryptocurrencies have some clusters of sentiment while others related to technology focus of the coins such as privacy , proof of stake, proof of work and other

## References

- [1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system.” <http://www.bitcoin.org/bitcoin.pdf>, 2009. [Online; accessed 2020-03-13].
- [2] C. H. E. Gilbert, “Vader: A parsimonious rule-based model for sentiment analysis of social media text,” in *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*. Available at (20/04/16) <http://comp.social.gatech.edu/papers/icwsml4.vader.hutto.pdf>, 2014.
- [3] P. Pandey, “Simplifying Sentiment Analysis using VADER in Python (on Social Media Text).” <https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f>. [Online; accessed 2020-02-18].
- [4] Hutto, “Twitter Emotion cryptocurrency.” <https://github.com/cjhutto/vaderSentimentb>, 2019. [Online; accessed 2020-03-02].
- [5] M. Kuhlmann, “Lectures: 732a92 text mining,” 2020.
- [6] I. Rish, “An empirical study of the naive bayes classifier,” in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, pp. 41–46, IBM, 2001.
- [7] H.-F. Yu, F.-L. Huang, and C.-J. Lin, “Dual coordinate descent methods for logistic regression and maximum entropy models,” *Mach. Learn.*, vol. 85, no. 1-2, pp. 41–75, 2011.
- [8] B. Zadrozny and C. Elkan, “Transforming classifier scores into accurate multiclass probability estimates,” in *KDD*, pp. 694–699, ACM, 2002.
- [9] T. Zhang, F. Damerau, and D. Johnson, “Text chunking based on a generalization of winnow,” *J. Mach. Learn. Res.*, vol. 2, pp. 615–637, 2002.
- [10] H. Zolkepli, “Twitter Emotion cryptocurrency.” <https://www.kaggle.com/huseinzol05/twitter-emotion-cryptocurrency/metadata>, 2019. [Online; accessed 2020-03-02].
- [11] J. Roesslein, “Tweepy Documentation.” <http://docs.tweepy.org/en/latest/>. [Online; accessed 2020-03-02].
- [12] O. Bhutra, “Repository for the project SentimentalCrypto.” <https://github.com/obhutara/SentimentalCrypto>, 2020. [Online; accessed 2020-03-13].
- [13] O. Bhutra, “Twitter Emotion cryptocurrency.” <https://www.github.com/obhutara/SentimentalCrypto/Data>, 2020. [Online; accessed 2020-03-02].
- [14] N. L. Toolkit, “Natural Language Toolkit.” <http://www.nltk.org/>, 2017. [Online; accessed 2020-03-01].
- [15] F. Sun, A. Belatreche, S. Coleman, T. McGinnity, and Y. Li, “Pre-processing online financial text for sentiment classification: A natural language processing approach,” 03 2014.
- [16] S. Richoz, “Twitter Sentiment And Cryptocurrencies.” [https://github.com/Drabble/TwitterSentimentAndCryptocurrencies/blob/master/02\\_CleanedTweetsIntoMultipleFiles.ipynb](https://github.com/Drabble/TwitterSentimentAndCryptocurrencies/blob/master/02_CleanedTweetsIntoMultipleFiles.ipynb). [Online; accessed 2019-12-25].

- [17] J. Thesken, “Building an Altcoin Market Sentiment Monitor.” <https://towardsdatascience.com/building-an-altcoin-market-sentiment-monitor-99226a6f03f6>, 2018. [Online; accessed 2019-12-25].
- [18] J. L. Evita Stenquist, “Building an Altcoin Market Sentiment Monitor.”