CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

Sentiment Analysis to Predict Bitcoin Market Fluctuations

Machine Learning (MDS271) Project Report

Siddhartha Roy, 2148026

Vaibhav Joshi, 2148029

Table of Contents

Abstract

This study examines the impact of sentiment on the fluctuations in the cryptocurrency market and specifically the dominant cryptocurrency i.e, Bitcoin. We examine how the prices fluctuate based on the polarity and subjectivity scores of news headlines extracted from the CNBC website. Furthermore, we also consider the Fear and Greed Index and try to predict the price fluctuations of Bitcoin based on the aforementioned features. We perform binary classification on the target variable column using popular machine learning models such as Random Forest, Decision Tree, Support Vector Machines, and Naive Bayes. We have obtained an accuracy of 75.4% on the training set and 72.11% on the testing set using Random Forest Classifier. As is indicative, when the equity market investors' sentiment is bearish, cryptocurrency prices rise, indicating that cryptocurrency can act as an alternative avenue for investment.

Introduction

In the past decade, another class of financial assets, cryptocurrency, arose into the asset management industry in risky portfolios, because of their low correlations with other major financial assets. As per Coinmarketcap, the cryptocurrency market has evolved dramatically over the past few years, with market capitalization surpassing US$2tn. As of April 21, 2021, the Bitcoin (BTC) market value already surpassed US$1tn with 89% of coins in circulation. CryptoCoinCharts shows 10,125 crypto coins as of April 2021, mostly attributed to factors like BTC open source which allows the continual creation of new cryptocurrencies.

Natural Language Processing (NLP), as an emerging branch of ML, has particularly seen various applications in finance and asset management. Sentiment Analysis is one such area where attempts are made to uncover information from content like financial news headlines. For instance, it was found in previous studies that user comments in online crypto communities can predict changes in BTC prices. Some studies also provide evidence to show that BTC is more of a speculative bubble whose prices and volume of trade depend on people's sentiments about it.

This study that we have conducted aims to explore the unexplored seas of crypto market volatilities using methods of text and sentiment analysis to predict whether the prices of BTC increased or decreased on a given day.

Sentiment Analysis, cryptocurrency, and machine learning

Sentiment analysis is an emerging part of NLP, with evidence ranging from the valence of words/phrases to document level classifications. Using sentiment in crypto predictions is an attractive research topic, due to BTC becoming public on 3 January 2009. Studies like Shah and Zhang (2015) used tools like Bayesian regressions to predict BTC but failed to include sentiment data. In addition to Patel (2015), who used support vector regression, artificial neural
networks, and random forest algorithms, Cocco (2021) analyzed the use of Bayesian
neural networks, long short-term memory neural networks, and support vector regression in predicting BTC prices.

Data Collection

We have performed the web data scraping using the tool Octoparse. It is an effective, newly designed, powerful tool that can scrape data automatically from the HTML elements of the web pages with support for a user-friendly task edit interface.

We have used Octoparse to extract the data regarding the news headlines and the dates they were published on. We have also used the BeautifulSoup4 and requests library in Python 3. x to extract the Fear and Greed index of BTC since 2019 from an interactive chart using .json requests. Both the aforementioned methodologies are discussed below.

Methodology (Octoparse):

- Get links from which data is to be scrapped. For our purposes of extracting news headlines related to BTC, we have used CNBC which is an all-purpose website for Stock market and finance-related news and headlines.
- Getting web page in Octoparse:
    - We enter the web page link in Octoparse.
    - After clicking Start, the tool automatically extracts the specified columns of data from the HTML elements of the pages.
- Creating Pagination
    - Because the WebPage contains many pages, we create Pagination to extract all the web page data:
    - We scroll to the next page by clicking on the next (>) button present on the webpage.
- Create Workflow for Data Extraction
- Building a Loop item - we create a loop inside pagination to extract data from all web pages.
- Run the task.
- After running the task, we can save the file in .csv format and import it into Python as DataFrame for further application.

Methodology(bs4, requests, and .json):

- We have imported libraries such as requests, BeautifulSoup4, DateTime, and pandas.
- We have initialized headers for our URL requests to make the HTTP requests behave as humanly as possible.
- Next, we try to get the .json element from a webpage that contains the Fear and Greed index data in an interactive chart. We used the chart here with the time as “all-time”.
- We pass the request URL as an argument to our requests. get().json() function and extract the timestamp and fear and greed index in a Python dictionary.
- Store the obtained dictionary as a pandas DataFrame and convert the timestamps into pandas Datetime format using the DateTime.fromtimestamp() function.
- We have obtained the data in our required format. Finally, we join these data without obtaining news headlines data using “inner join” method.

Code:

```
url = "https://api.btctools.io/api/fear-greed-chart?period=all"
```

```
response = session.get(url).json()
```

Data Preparation

Libraries used:

- **Pandas** - for data manipulation and analysis on data structures and numerical tables.
- **NLTK** - Natural Language Toolkit; for the suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.
- **Regular Expressions** - library with functionality and tools to process a sequence of characters that specifies a search pattern in the text.
- **Contractions** - for expanding contractions in textual data.
- **String** - for enabling removal of punctuation functionality from textual data.
- **Numpy** - library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays
- **Scipy. stats** - This module contains a large number of probability distributions, summary and frequency statistics, correlation functions and statistical tests, masked statistics, kernel density estimation, quasi-Monte Carlo functionality, and more.

Methodology:

We have performed the necessary textual data preprocessing such as string lowering, expanding contractions, stripping texts of whitespaces and unnecessary characters, tokenizing text into words (tokens) using word_tokenize from nltk, removal of stopwords using list comprehension, and finally lemmatizing strings (sophisticated text stemming method).
The prepared text has been stored in a separate column in the DataFrame. Most necessary sentiment analysis methods will be applied to the preprocessing column of text.

Preprocessing user-defined function:

```
def text_prep(x: str) -> list:
    corp = str(x).lower()
    corp = contractions.fix(corp)
    corp = re.sub('[^a-zA-Z]+',' ', corp).strip()
    tokens = word_tokenize(corp)
    words = [t for t in tokens if t not in stop_words]
    lemmatize = [lemma.lemmatize(w) for w in words]
    return lemmatize
```

VADER (Valence Aware Dictionary for sEntiment Reasoning)

VADER (Valence Aware Dictionary for Sentiment Reasoning) is a model utilized for text sentiment analysis that is sensitive to both polarity (optimistic/pessimistic) and intensity (strength) of emotion. It is accessible in the NLTK bundle and can be applied directly to unlabeled text information.
VADER sentiment analysis depends on a dictionary that maps lexical elements to feeling powers known as sentiment scores. The sentiment score of a text can be obtained by summarizing the intensity of each word in the text.

Utilizing VADER, we acquire the polarity score for the textual information and store it in a different segment in our DataFrame.

VADER (Polarity) code:

```
from nltk.sentiment.Vader import SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()

train_sentiments = []

for i in df['prep_text']:
   train_sentiments.append(sid.polarity_scores(i).get('compound'))

train_sentiments = np.asarray(train_sentiments)
df['polarity'] = pd.Series(data=train_sentiments)
```

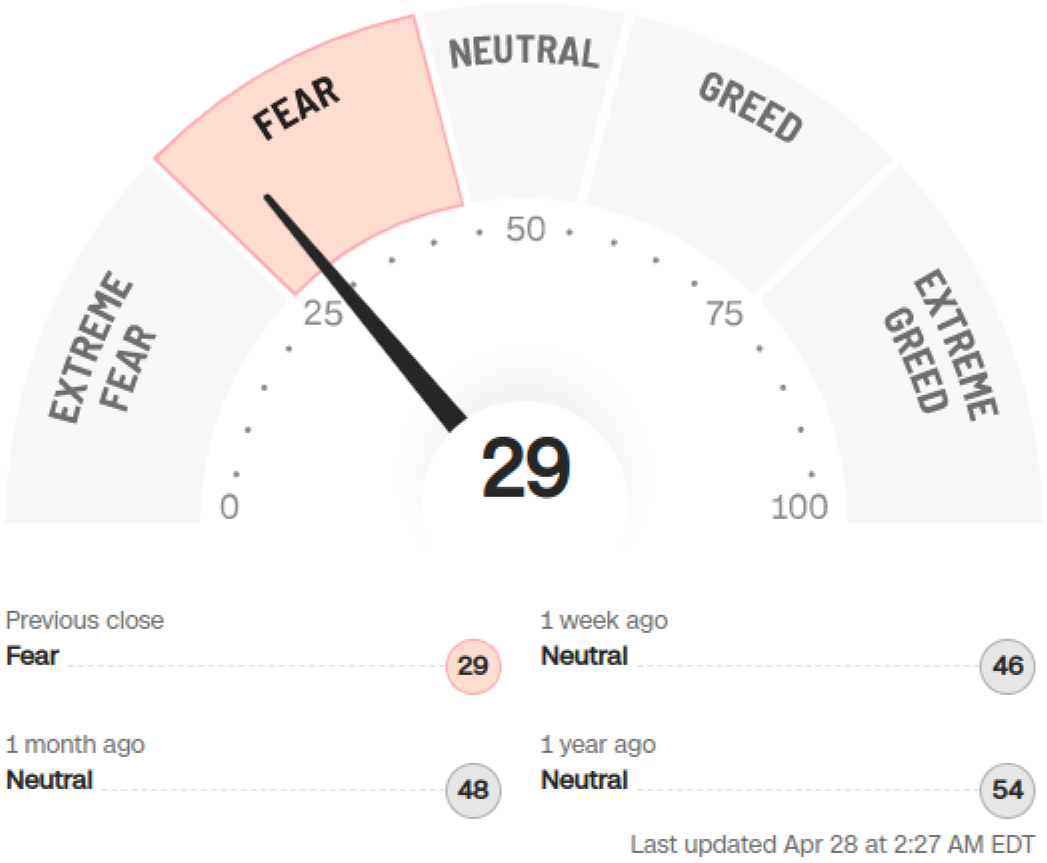
Polarity Score

For ascertaining the polarity of a text, the polarity score of each word of the text, if present in the dictionary, is added to get an 'overall polarity score'. For instance, in the event that a vocabulary matches a word set apart as certain in the dictionary, the absolute polarity score of the text is increased.
We have found the polarity score using the get ‘compound’ function in VADER to get the overall polarity.

Polarity Score = (Positive Score – Negative Score)/ ((Positive Score + Negative Score) + 0.000001)

Fear and Greed Index

The Fear & Greed Index is a compilation of seven different indicators that measure some aspects of stock market behavior. They are market momentum, stock price strength, stock price breadth, put and call options, junk bond demand, market volatility, and haven demand. The index tracks how much these individual indicators deviate from their averages compared to how much they normally diverge. The index gives each indicator equal weighting in calculating a score from 0 to 100, with 100 representing maximum greediness and 0 signaling maximum fear.



Previous close
Fear 29

1 week ago
Neutral 46

1 month ago
Neutral 48

1 year ago
Neutral 54

Last updated Apr 28 at 2:27 AM EDT

Model Building

We have resorted to classic ML models such as Logistic Regression, Decision Trees, Random Forest, Support Vector Machines, and Gaussian Naive Bayes. First, we have performed the train-test split on the entire dataset of 769 rows into a training set and validation set in the ratio of 3:1. Once we have the train and validation sets, we have reshaped them into suitable shapes to be used to train the defined models with the independent variables Polarity and Fear and Greed Index and the dependent/target variable as Price which is either 0 or 1 (0 signifying price decreased from the previous day and 1 signifying the price increased from the previous day). This can be done using the np.reshape(-1, 1) function.

The accuracy scores of different models are obtained as follows:

1	Random Forest Classifier	76.10	74.74
2	Support Vector Machine	59.05	61.558
3	Gaussian Naive Bayes	57.82	59.47
4	Decision Tree Classifier	76.10	71.58
5	Logistic Regression	58.88	60.53

Hyperparameter Optimization

Pruning

The first step to hyperparameter optimization that we have resorted to is called pruning the decision trees and random forest classifiers for the best value of “ccp_alpha”. Decision Trees are infamous as they can cling too much to the data they are trained on. This leads to poor deployment because it cannot deal with new sets of values. Thus, pruning is done to overcome the problem of overfitting.

When the alpha values are set to zero (default), both the decision tree and the random forest overfits, As the alpha value is slightly increased, more of the tree is pruned thus creating a decision tree and random forest that generalizes better. This is especially of importance to us as our data set is of relatively smaller size. Thus, we need our model to generalize results and predictions better in order to be able to predict unknown results with maximum accuracy.

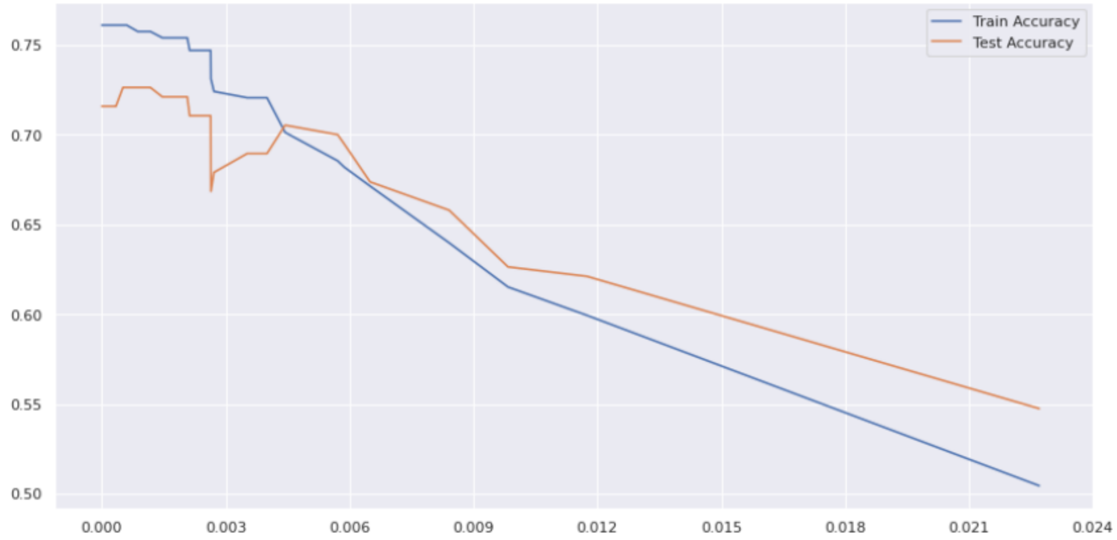


Fig: Pruning Decision Tree

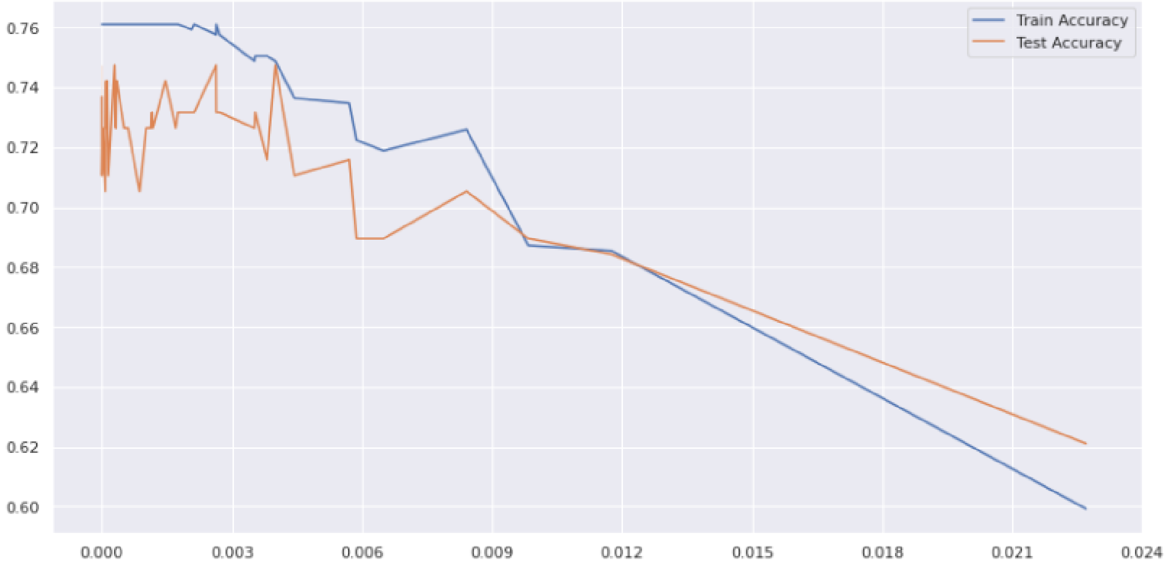


Fig: Pruning Random Forest

GridSearchCV

We run 4 models through our GridSearchCV: Support Vector Machine, Logistic Regression, Decision Tree, Random Forest. We define a dictionary with all the necessary hyperparameters for the above models. We run the cross-validation for 100 iterations and obtain the best estimators as the model RandomForestClassifier(n_estimators=100, criterion='entropy', ccp_alpha=0.0035).

Validation on Test Set

We have validated on the Training and Testing Set for both the Decision Tree and Random Forest model and they yielded test scores of 71.58% and 72.11%, respectively.

K-Fold Cross Validation

We have also finally performed manual K-Fold Cross Validation (with 100 CVs) for both the models and the Decision Tree yielded a 67.0% score whereas the Random Forest model yielded a 67.86% score.

Conclusion

The shortcomings in the cryptocurrency market persuaded this review to investigate the social part of their price discovery process. In our study, we have used an independently web scraped text data set to capture various sentiment analysis measures such as polarity, subjectivity, and market sentiment measures such as the Fear and Greed Index. The result of the study found that we can predict the fluctuations (whether the price of BTC rose or fell from the previous) day based on the market sentiment measures mentioned above with an accuracy of at least 70%. This proves that Bitcoin sentiment has a positive impact on Bitcoin returns supporting our hypothesis that behavioral aspects play a significant role in the prices of BTC.

References

[1] Anamika, Madhumita Chakraborty, and Sowmya Subramaniam; "Does Sentiment Impact Cryptocurrency?", 2021.

[2] C.J. Hutto Eric Gilbert: "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text",2014.

[3] Ikhlaas Gurrib (2018). Are key market players in currency derivatives markets affected by financial conditions?., 2018

[4] Clayton R. Fink, Danielle S. Chou, Jonathon J. Kopecky, and Ashley J. Llorens: "Coarse- and Fine-Grained Sentiment Analysis of Social Media Text",2011

[5] Fan Fanga, Carmine Ventrea, Michail Basiosb, Hoiliong Kongb, Leslie Kanthanb, David Martinez-Regob, Fan Wub, and Lingbo Lib:" Cryptocurrency Trading: A Comprehensive Survey",2020

[6] Ikhlaas Gurrib, Firuz Kamalov:" Predicting bitcoin price movements using sentiment analysis: a machine learning approach",2021

[7] Jethin Abraham, Daniel Higdon, John Nelson, Juan Ibarra:" Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis",2018