Mohammad Maaz | Mrunalini Devineni
Natural Language Processing for Data Science
Group Final Report

Impact of Emotion on Bitcoin Price

1st May 2021

INTRODUCTION

In our project, we aim to analyze the impact of emotion of cryptocurrency market players on the price of Bitcoin using the sentiment classification abilities of transformer based natural language processing techniques.

GOALS

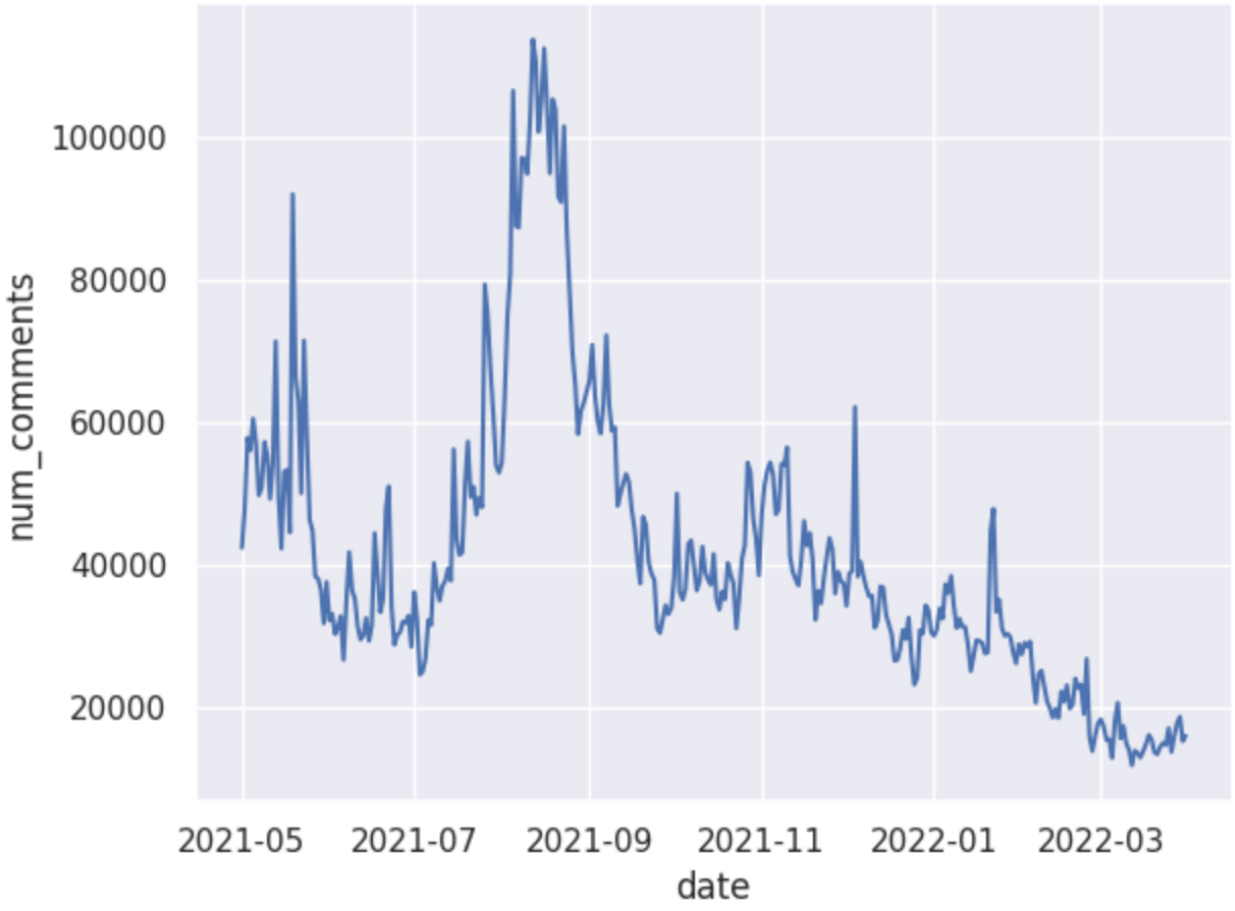
1. Develop a dataset of Reddit comments from the r/cryptocurrency subreddit
2. Fine tune a transformer model on an emotion classification dataset
3. Repurpose the model to predict the emotion exhibited by Reddit comments
4. Investigate trends between certain emotions and the price of Bitcoin

DATA

Reddit Comments

We obtain Reddit comments using PMAW[1], a multithreaded wrapper for PushShift API[2]. PushShift is a RESTful API that gives full functionality of searching Reddit data. We were interested in comments, so we used the /reddit/search/comment endpoint. This API has a requests per minute limit of 60 and data points per request limit of 100. We used PMAW to work on this in a multithreaded way to speed it up. Using this approach, we pulled 16 million comments from the cryptocurrency subreddit from 1st May 2021 to 1st April 2022.

[1] https://github.com/mattpodolak/pmaw
[2] https://github.com/pushshift/api



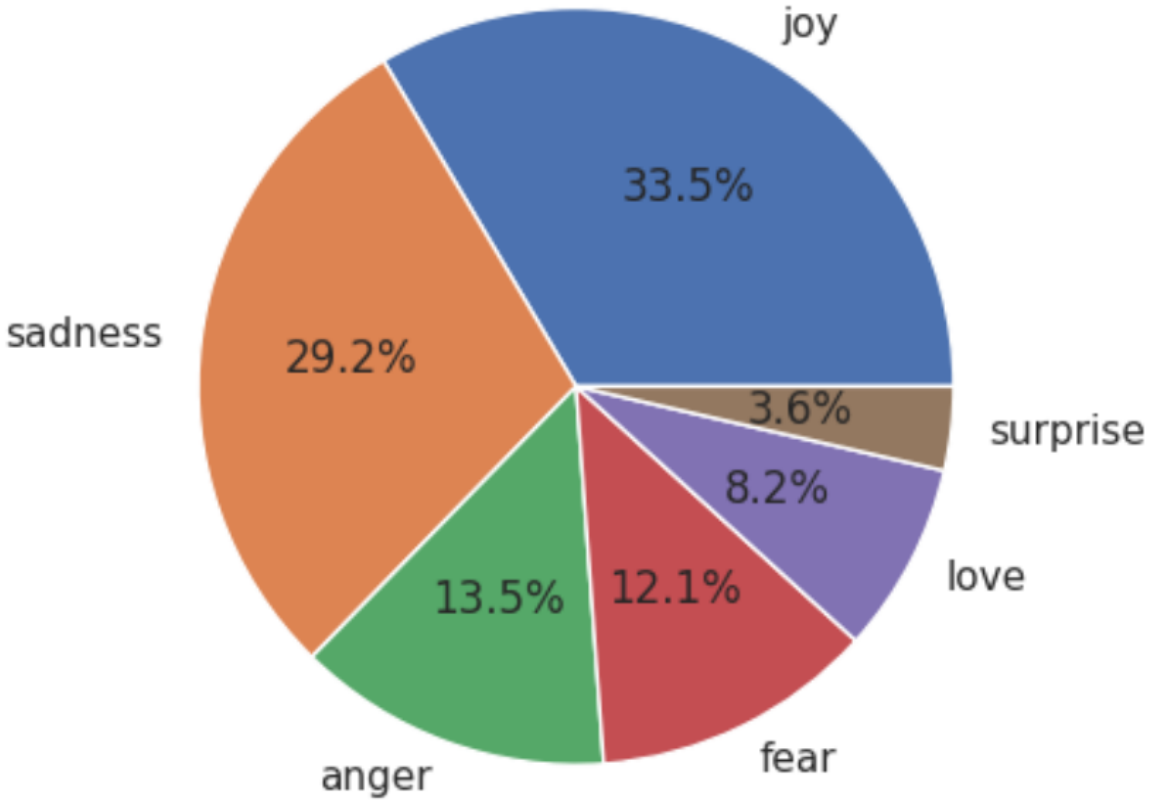
Emotions

The emotion[3] dataset was obtained from Hugging Face. This is a dataset of Twitter messages with six different emotions: anger, fear, joy, love, sadness, and surprise. This is split as follows.

Train	16000
Validation	2000
Test	2000

This dataset was developed in the paper Contextualized Affect Representations for Emotion Recognition[4], which proposed a semi-supervised, graph-based algorithm to produce rich structural descriptors to detect emotion.

[3] https://huggingface.co/datasets/emotion
[4] https://aclanthology.org/D18-1404.pdf



Bitcoin Price

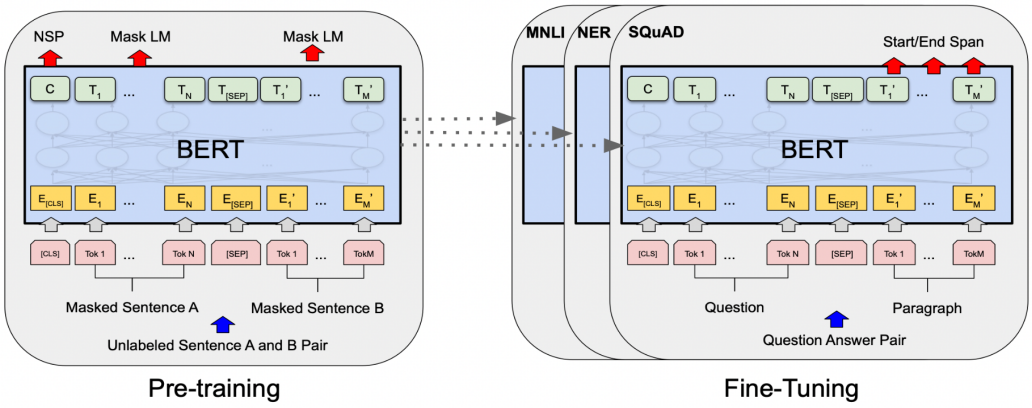
We used the pandas_datareader[5] wrapper for the Yahoo Finance API to obtain the historical price of bitcoin. We used adjusted close price in our analysis, which is the closing price after adjustments for all applicable splits and dividend distributions and thus the best metric for price on a particular day.

MODEL

BERT,[6] short for Bidirectional Encoder Representations from Transformers, is a Machine Learning (ML) model for natural language processing.The most common language tasks performed by it are sentiment analysis and name entity recognition.BERT has revolutionized the NLP space by

[5] https://github.com/pydata/pandas-datareader
[6] https://arxiv.org/pdf/1810.04805.pdf

performing better previous developed models.BERT uses transformer a mechanism that learns contextual relations between words in text. It consists of an encoder that reads the input and decoder that predicts the task. But Bert needs only encoder as its goal is to generate a language model. This is called bidirectional because it reads the entire sequence of words at once.



The model [7]is trained on unlabeled data across many pre-training tasks during pre-training. The BERT model is fine-tuned using labeled data from downstream tasks after it is initialized with the pre-trained parameters. We pertain BERT using two supervised tasks:

1) Masked Language model: Here 15 percent of the words in each word sequence are substituted with a [MASK] token before being fed into BERT. Based on the context provided by the other, non-masked words in the sequence, the model then attempts to predict the original value of the masked words. Further, the BERT loss function considers only predicted masked values.
2) Next Sentence Prediction:The BERT training approach involves feeding the model pairs of sentences and learning to predict whether the second sentence in the pair is the next sentence in the original document.

Fine-tuning is simple since the transformers mechanism allows BERT to mimic a wide range of downstream tasks just by swapping relevant input and outputs. For each task we simply plug in task specific input and output into BERT and fine tune all parameters end to end.

[7] https://dzone.com/articles/bert-transformers-how-do-they-work

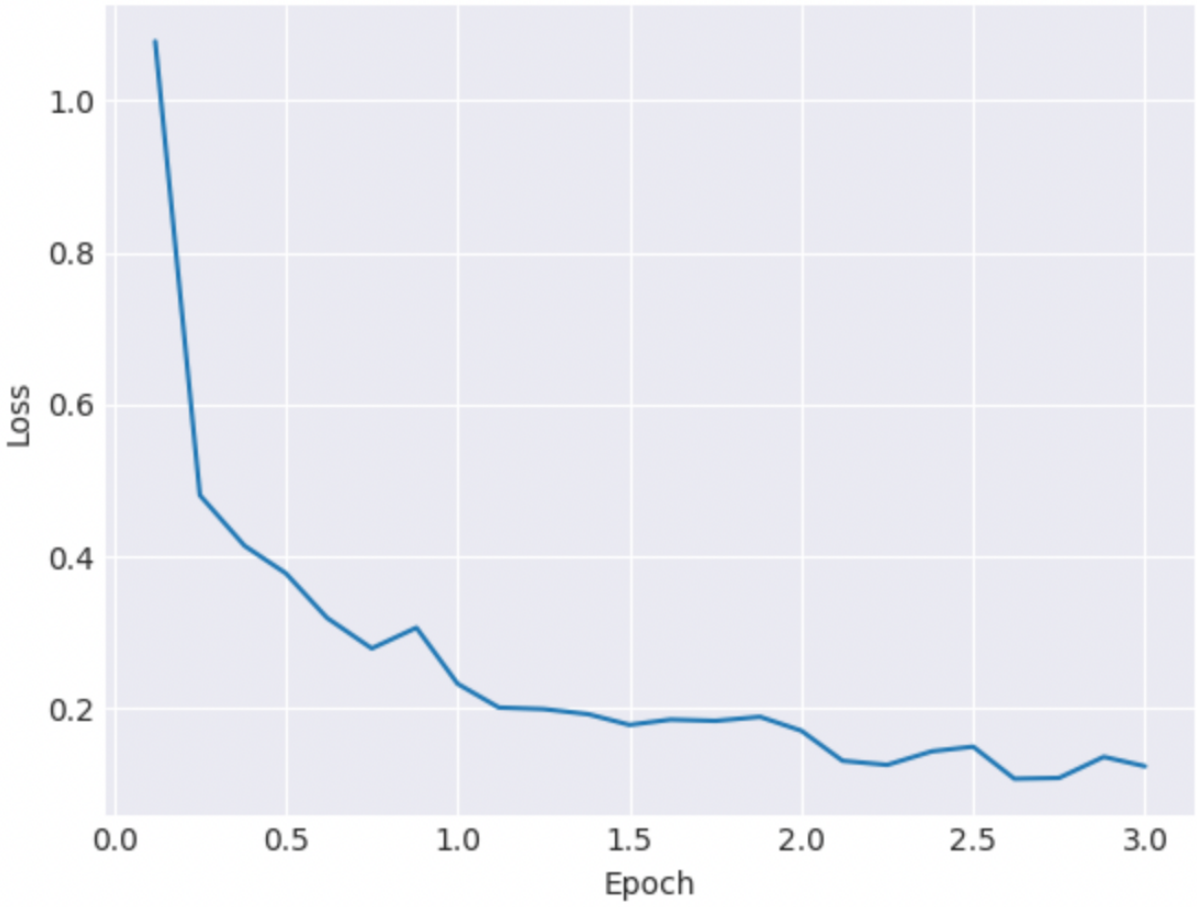
EXPERIMENT

Training BERT

We use the Hugging Face transformer APIs to train our model. We first load the emotions dataset, which has labels and texts for three splits; train, validations and test. We use the BERT tokenizer to convert our text inputs into integer encodings and boolean attention masks. We also use a data collator to collate the encodings and convert them to the same token length. We then load a pre-trained BERT model and train it on our tokenized dataset according to the following hyperparameters.

Hyperparameters	**Values**
Learning rate	2e-5
Batch size	4
num_train_epochs	3
weight_decay	0.01

We monitor the training by plotting the loss on the train set. Since the loss has not plateaued entirely, we can be confident that we have not overfitted our model.



Since our evaluation strategy was epoch, we evaluated on the validation set at the end of every epoch.

Epoch	Eval F1 Score	Eval Loss
0	0.899	0.293
1	0.919	0.203
2	0.921	0.227

Testing BERT

In order to make sure our model has not overfitted, we run it on the unseen test dataset consisting of 2000 data points. The classification performance is as follows.

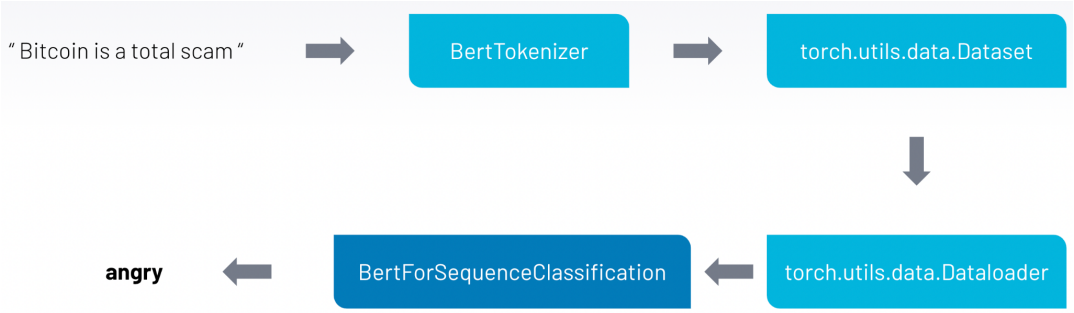
	precision	recall	f1-score	support
sadness	0.95	0.97	0.96	581
joy	0.96	0.94	0.95	695
love	0.80	0.91	0.85	159
anger	0.94	0.89	0.92	275
fear	0.87	0.91	0.89	224
surprise	0.79	0.70	0.74	66
accuracy			0.93	2000
macro avg	0.89	0.89	0.88	2000
weighted avg	0.93	0.93	0.93	2000

The performance on the test set is similar to that on the validation set, leading to the conclusion that our model has not overfitted and is ready for downstream inference.

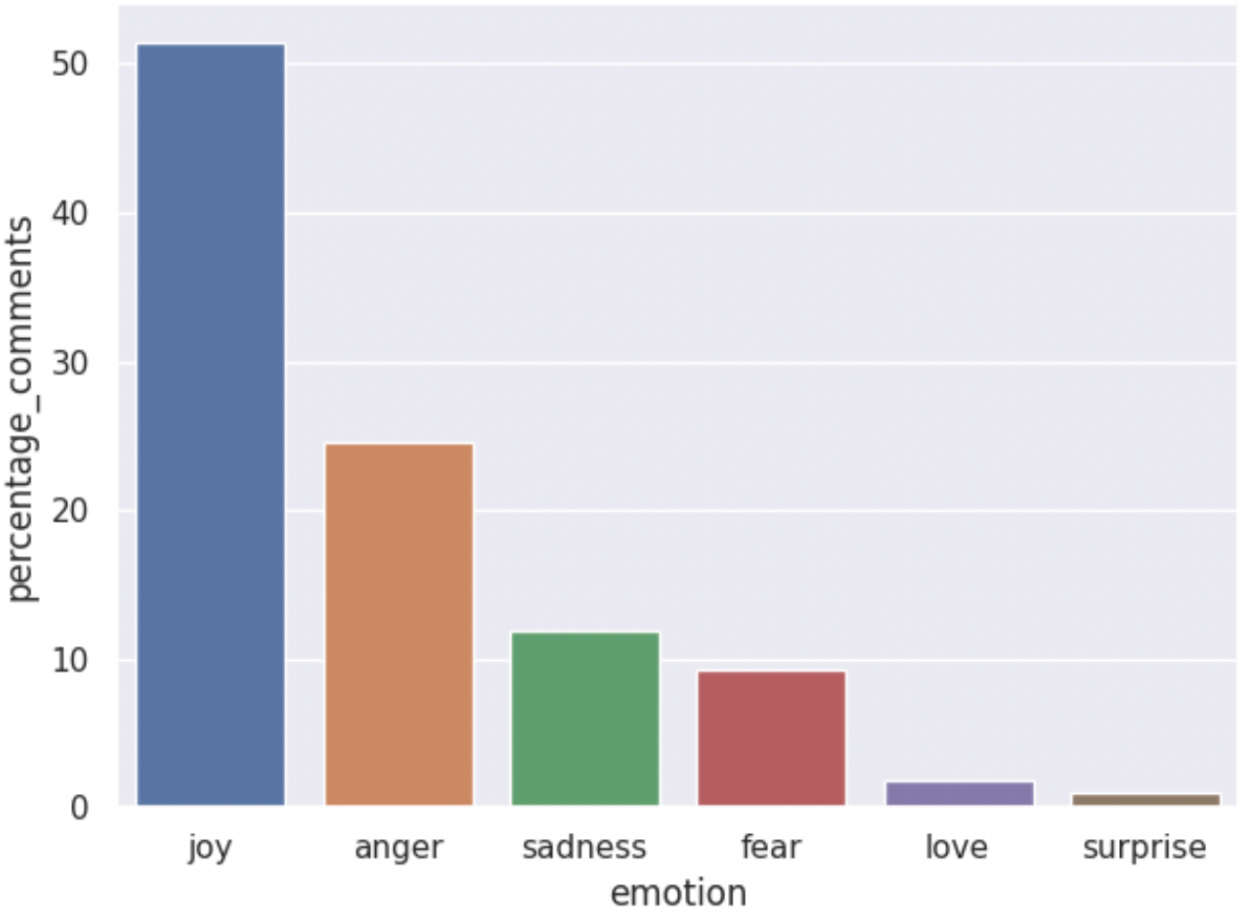
RESULTS

We obtained a random subset of the Reddit comments dataset so that inference could be completed in a reasonable timeframe. Since we are only interested in the percentage of comments exhibiting a particular emotion on a particular day, a random subset provides a good approximation. The subset contained around 500k comments.

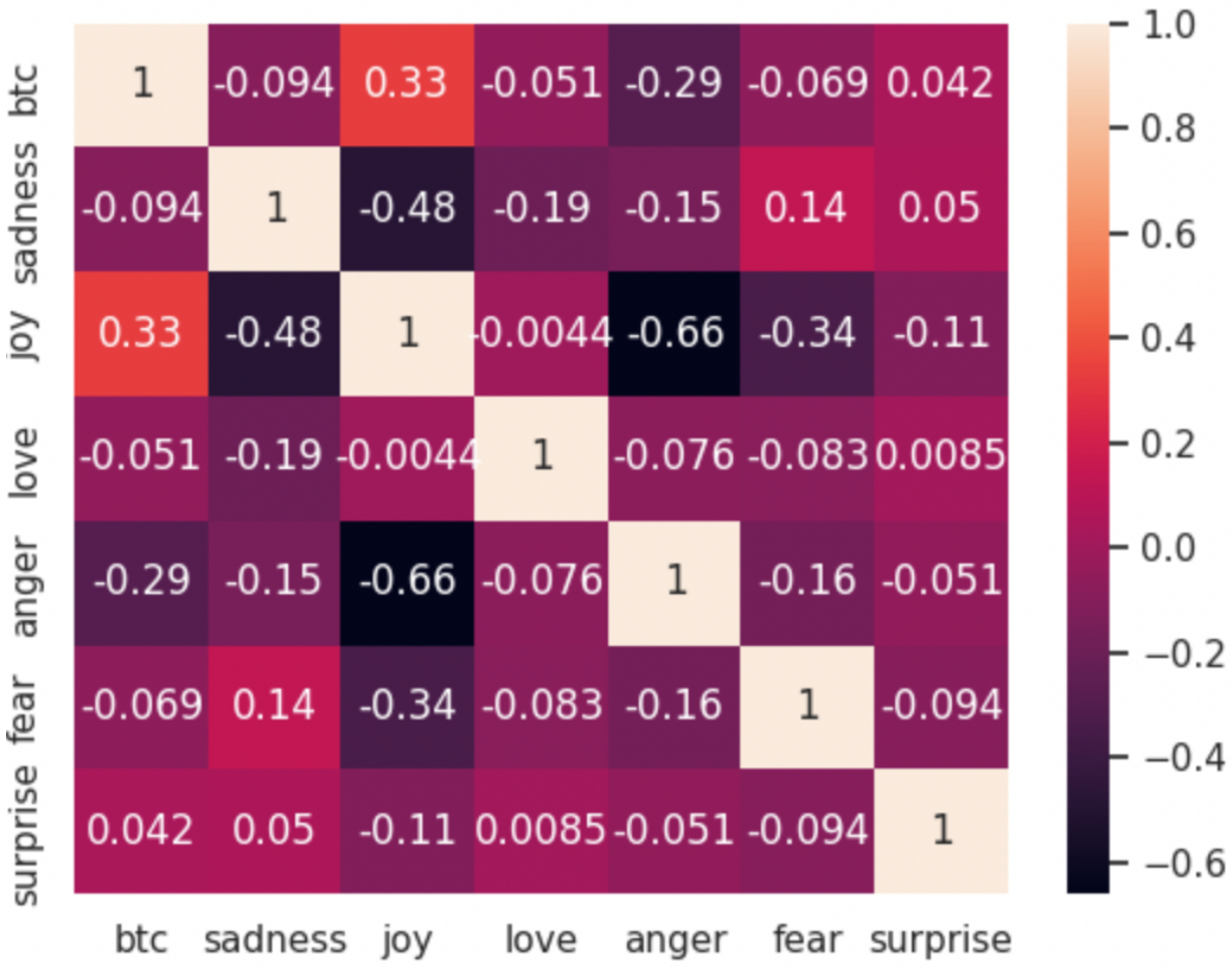
We load the model trained on the emotion dataset and run it on these comments and generate predicted classes for each comment. This was implemented as follows.



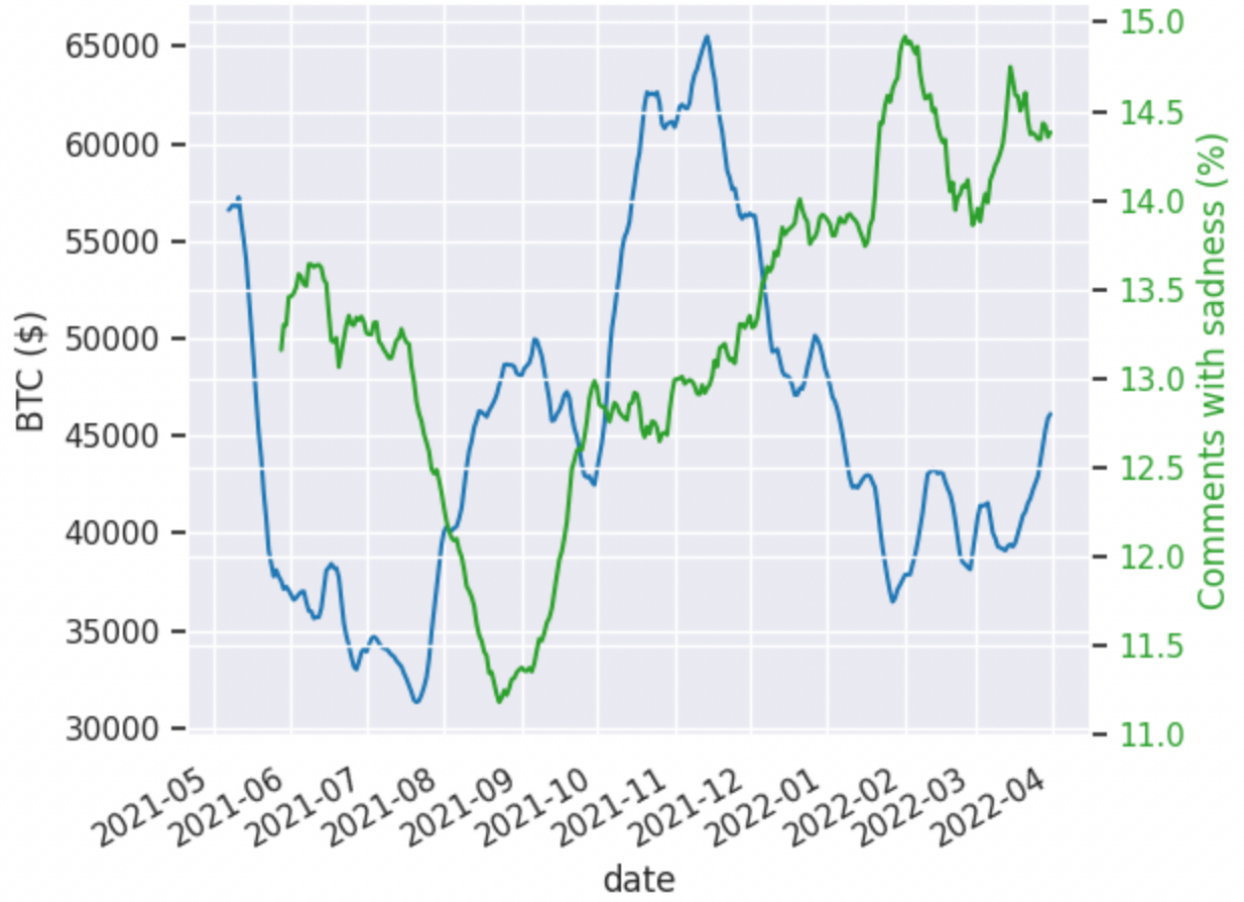
The distribution of predicted emotions is plotted below.



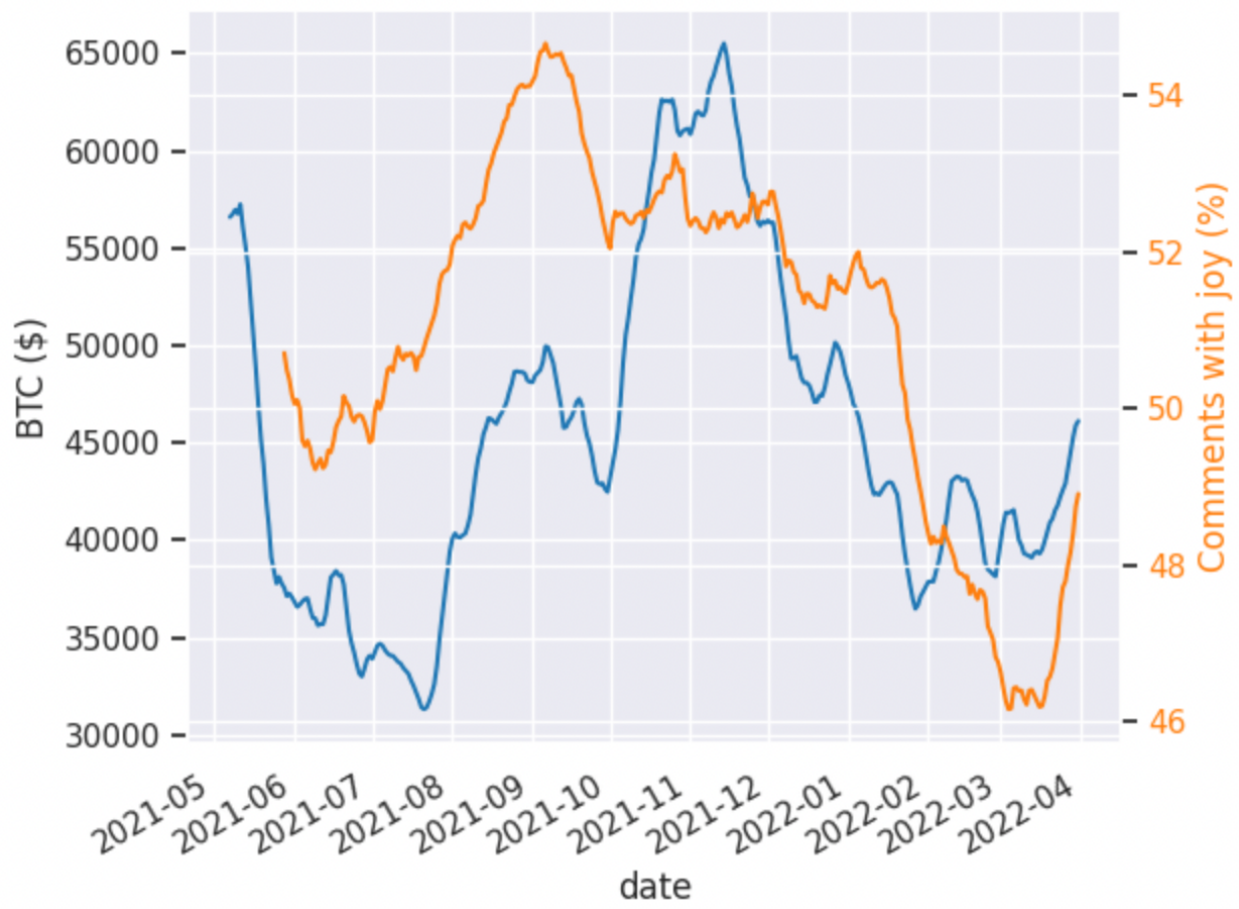
We can see that almost half of the comments are joyous comments, which reflects the general sentiment of the subreddit. We then investigate how this varies with respect to the price of BTC.



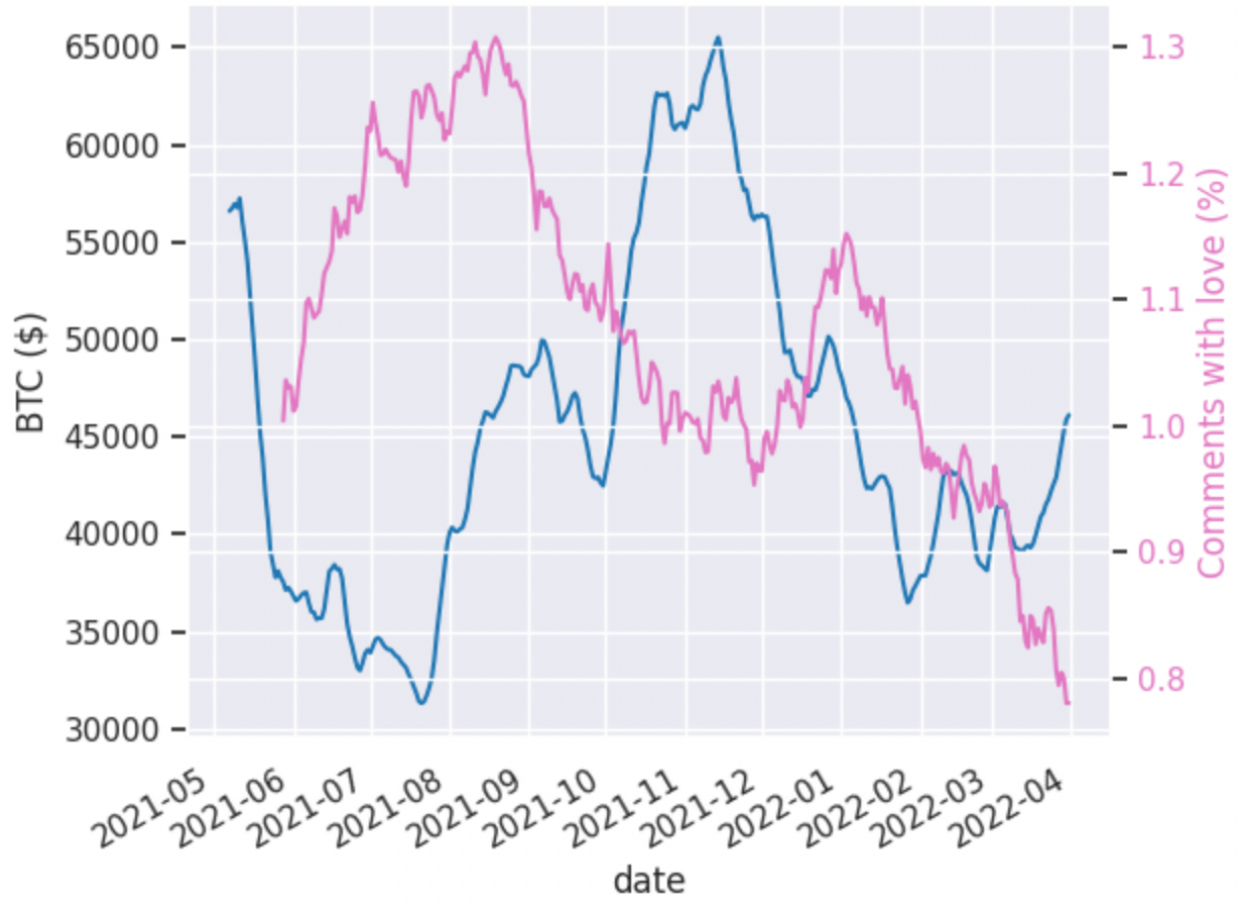
We can see that anger and joy have high magnitudes in their correlations with the price of Bitcoin. It remains to be seen whether this points to causation. To gauge causation, we need to look at this temporally.



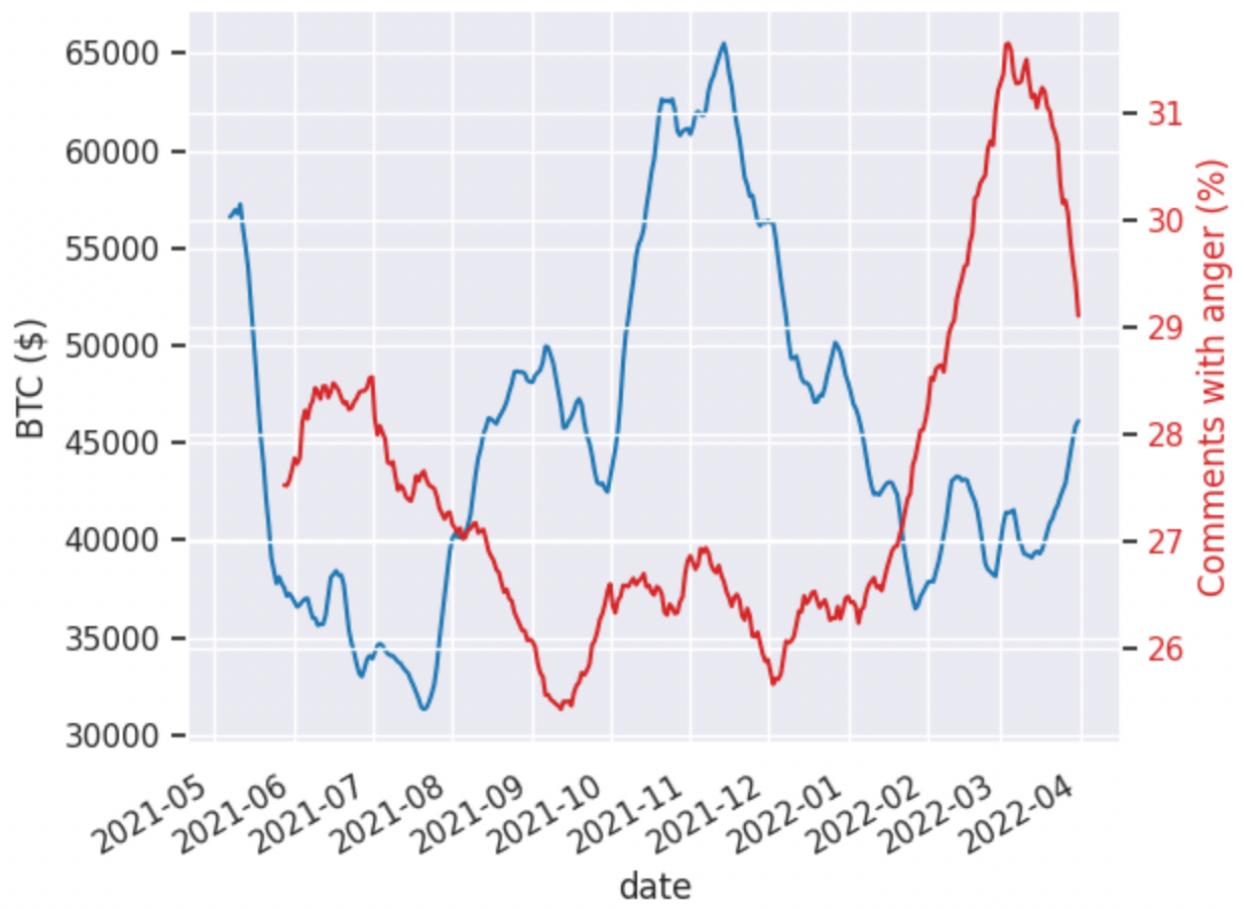
We can see that sadness saw an all time low during the Bitcoin price increase of September 2021. However, it has increased since then.



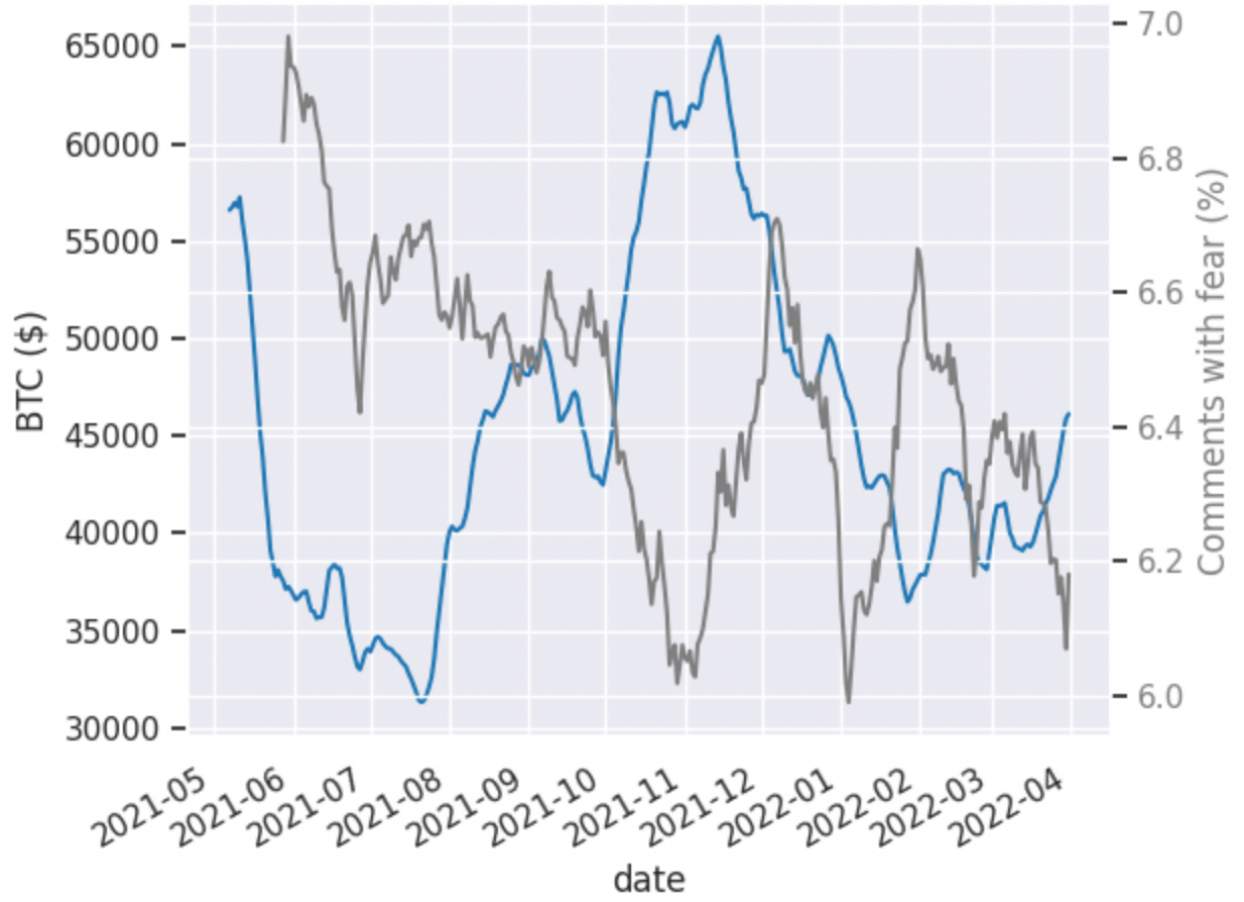
We can see that joy is very highly correlated to the Bitcoin price. An interesting thing to note is that it lags price increases. There was an uptick in joyous comments as early as July 2021 even though the price boom happened in September of the same year. Furthermore, the percentage of joyous comments dropped in September 2021 even though the crash happened in December 2021.



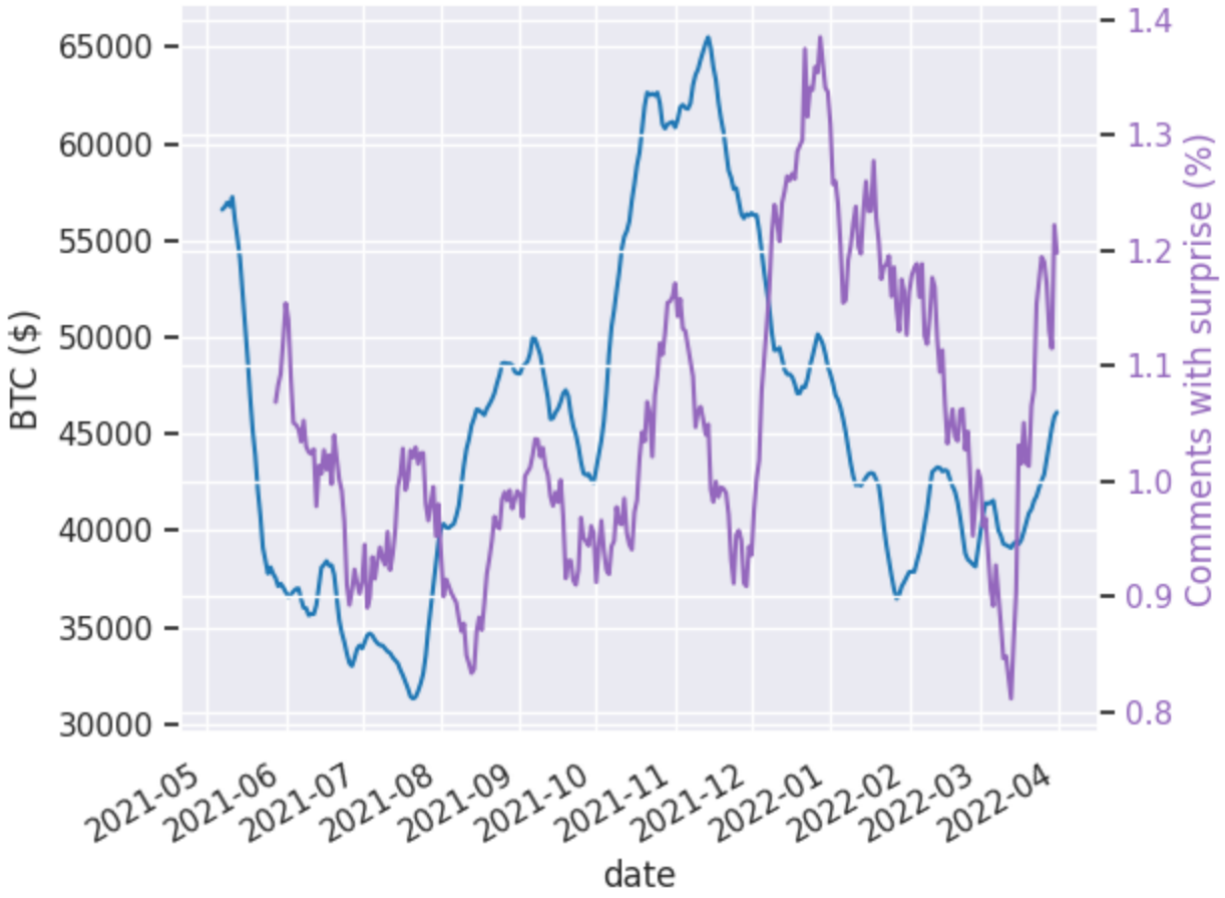
Love is not an emotion that generally drives market activity, thus there are no insights here.



Angry comments are very interesting because they are inversely correlated with the price drops. This is expected because people tend to take out their anger on the same forum that encouraged them to invest. The crests for the graph of Bitcoin price and the troughs for the graph of percentage of angry comments align.



According to most stock market specialists, fear is the biggest reason for fluctuations in a stock or commodity. This, however, is not very true in the case of Bitcoin because the percentage of afraid comments does not show any observable trend with the price of Bitcoin.



Surprise is a reactionary emotion and not very useful for our analysis.

CONCLUSION

In our project, we use the zero-shot classification power of transformer-based models to predict the emotion of Reddit comments, despite the fact that the model was trained on an entirely different dataset. The correlation between the percentage of comments exhibiting certain emotions and the price of Bitcoin indicates that these predictions were reliable. Furthermore, our work also shows that cryptocurrency market analysts can harness NLP techniques to advise investing, because cryptocurrency prices are more volatile than traditional stock prices.

REFERENCES

All references are provided as footnotes.