Crypto Knight

The World's Best
Machine Learning
Investment Platform

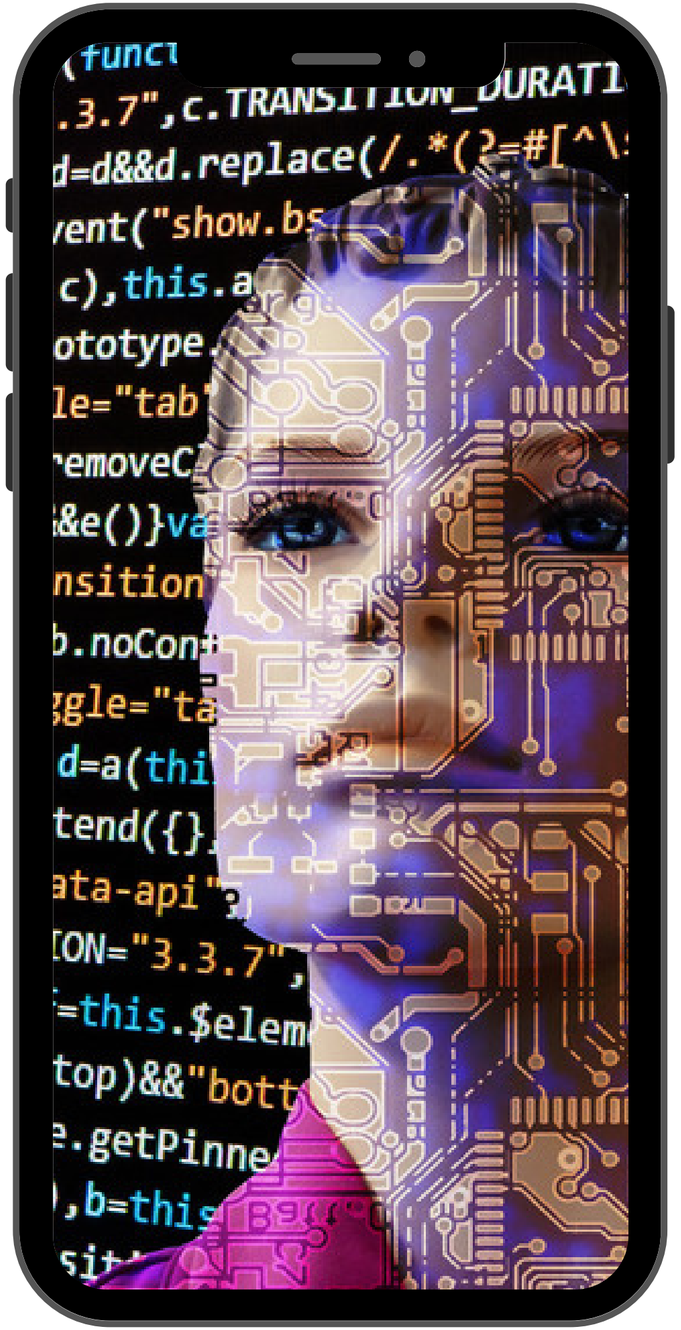
KAI - JAMES - SHANSHAN - ALISHA

In building our Investment Platform, we've explored whether:

(1) You can use a sentiment analyser applied to the general public's twitter activity to predict when to buy and sell cryptocurrency and,

(2) Whether you can use Algorithm Trading to automate the buy and sell process,

All in the efforts to build the World's best Machine Learning Investment Platform that is your CryptoKnight!



Part 1: The Sentiment Analyser

Our hypothesis:

"That you can use a sentiment analyser on public Twitter activity to guide your investment decisions regarding cryptocurreny."

How we tested:

We applied a sentiment analyser using Python library, 'TextBlob' to 50 and 60 days of Tweets about Bitcoin at 100 per day. We then compared this to the price action of Bitcoin to determine whether there was a strong correlation between positive sentiment results and positive price movements.

What model was used to test?

The TextBlob library uses the **Naive Bayes Classification Model** for sentiment analysis.

The pro's of this model outweight the con's.

Pro's

- Works well with little data
- Efficient
- Not biased by outliers
- Works on non-linear problems
- Probabilistic approach

Con's

Assumes that features have the same statistical relevance.

Testing time (the fun part!)

To gather the Twitter data we needed, we had to apply for the Twitter API. This was easy and quick but using it was our first problem.

We struggled to get the search feature working & almost gave up! Till we found a Python library that worked called Tweepy! This allowed us to get the info we wanted but...

MO DATA, MO PROBLEMS?

That is when we encountered our second issue: the free accounts we signed up for only allowed us to make 50 requests per month.

We wanted as much data as possible knowing our model would be more accurate if it had more data than less.

Twitter Search Method

```
searchterms = 'bitcoin'
startdate = dt.datetime(2021, 2, 8)
daterange = 50
yfstartdate = startdate + dt.timedelta(days=1)
yfenddate = startdate + dt.timedelta(days = daterange)

yfstart = yfstartdate.strftime("%Y-%m-%d")
yfend = yfenddate.strftime("%Y-%m-%d")
BTC = yf.Ticker('BTC-USD')
df_BTC = BTC.history(start=yfstart, end=yfend)

daysentiment = []
rawtexttotal = []

for i in range(daterange):

    fDate = startdate.strftime("%Y%m%d%H%M")
    startdate += dt.timedelta(days=1)
    tDate = startdate.strftime("%Y%m%d%H%M")

    results = api.search_full_archive(twitter_sandbox,


    senti = []
    rawtext = []

    for tweet in results[0:100]:
        #print(tweet.text)
        #print(tweet.created_at)
        analysis = TextBlob(tweet.text)
        score = analysis.sentiment.polarity
        #print()
        #print(f"Sentiment= {score}")
        #print()
        senti.append(score)
        rawtext.append(tweet.text)

    daysentiment.append(mean(senti))
    rawtexttotal.append(rawtext)
```

We used the Tweepy Library to search Twitter over a specific sequential date range.

A FOR loop was used, and the last 100 tweets for the day were collected. Sentiment analysis from the TextBlob Library was used to analyse each tweet individually, and polarity was applied to give each Tweet a score between -1.0 and +1.0.

The search and analysis process took around 5 minutes or so.

The mean value of each days sentiment score across the 100 Tweets was calculated and saved.

The daily BTC-USD pricing was collected from Yahoo finance, and the mean Twitter sentiment for each day was added as another column 'Sentiment'.

Twitter Search Method 2

```
searchterms = 'bitcoin'
startdate = dt.datetime(2021, 2, 8)
daterange = 50
yfstartdate = startdate + dt.timedelta(days=1)
yfenddate = startdate + dt.timedelta(days = daterange)

yfstart = yfstartdate.strftime("%Y-%m-%d")
yfend = yfenddate.strftime("%Y-%m-%d")
BTC = yf.Ticker('BTC-USD')
df_BTC = BTC.history(start=yfstart, end=yfend)

daysentiment = []
rawtexttotal = []

for i in range(daterange):

    fDate = startdate.strftime("%Y%m%d%H%M")
    startdate += dt.timedelta(days=1)
    tDate = startdate.strftime("%Y%m%d%H%M")

    results = api.search_full_archive(twitter_sandbox,


    senti = []
    rawtext = []

    for tweet in results[0:100]:
        #print(tweet.text)
        #print(tweet.created_at)
        analysis = TextBlob(tweet.text)
        score = analysis.sentiment.polarity
        #print()
        #print(f"Sentiment= {score}")
        #print()
        senti.append(score)
        rawtext.append(tweet.text)

    daysentiment.append(mean(senti))
    rawtexttotal.append(rawtext)
```

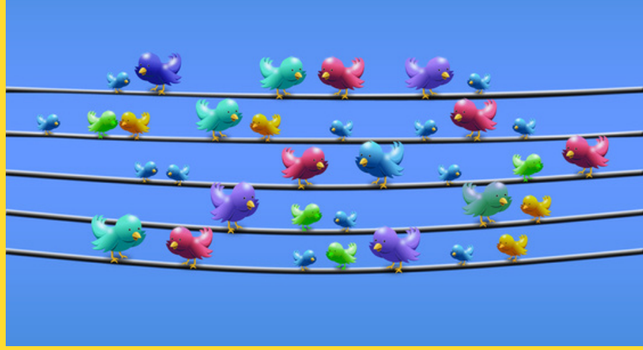
As we ran our FOR loop, we also created a list of strings, for the returned Tweets each day so we could store and analyse them.

At the end of each iteration, the list of Tweets was extended to a master list.

By the end of this process, we had collected 5000 tweets over the 50 days.

This data was then further cleaned of all emoticons and non-English words.

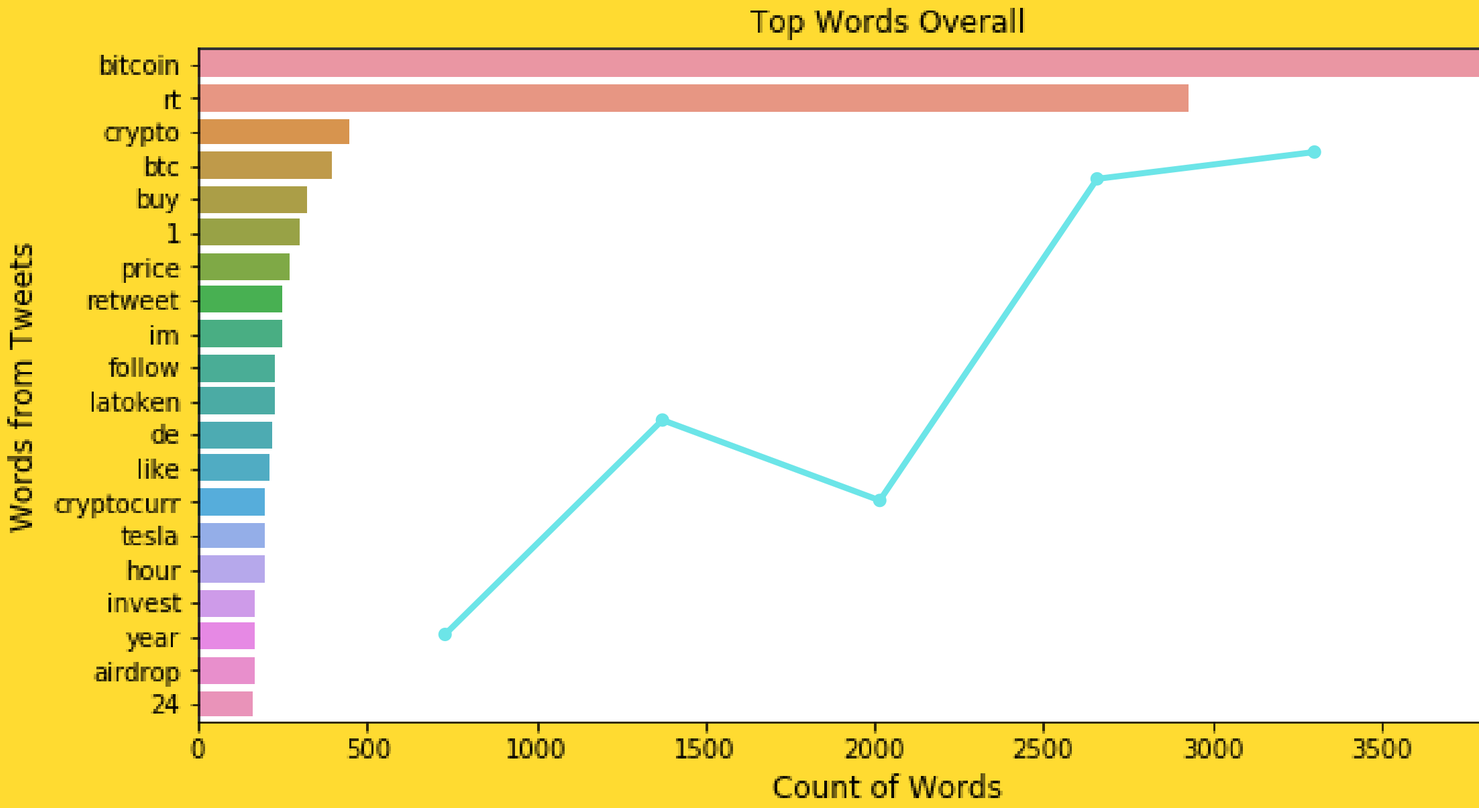
This text data was now ready for word analysis.



DATA COLLECTION

Twitter API + Tweepy python package

- Full archive vs Cursor search
- 400,000 tweets for 24 hours of data!
- Used first 100 tweets, 50 days
- Yahoo Finance API for Bitcoin price data

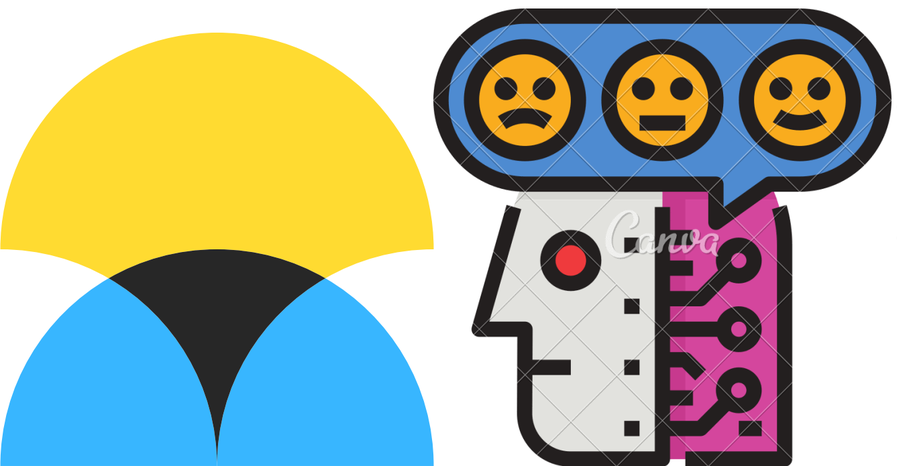


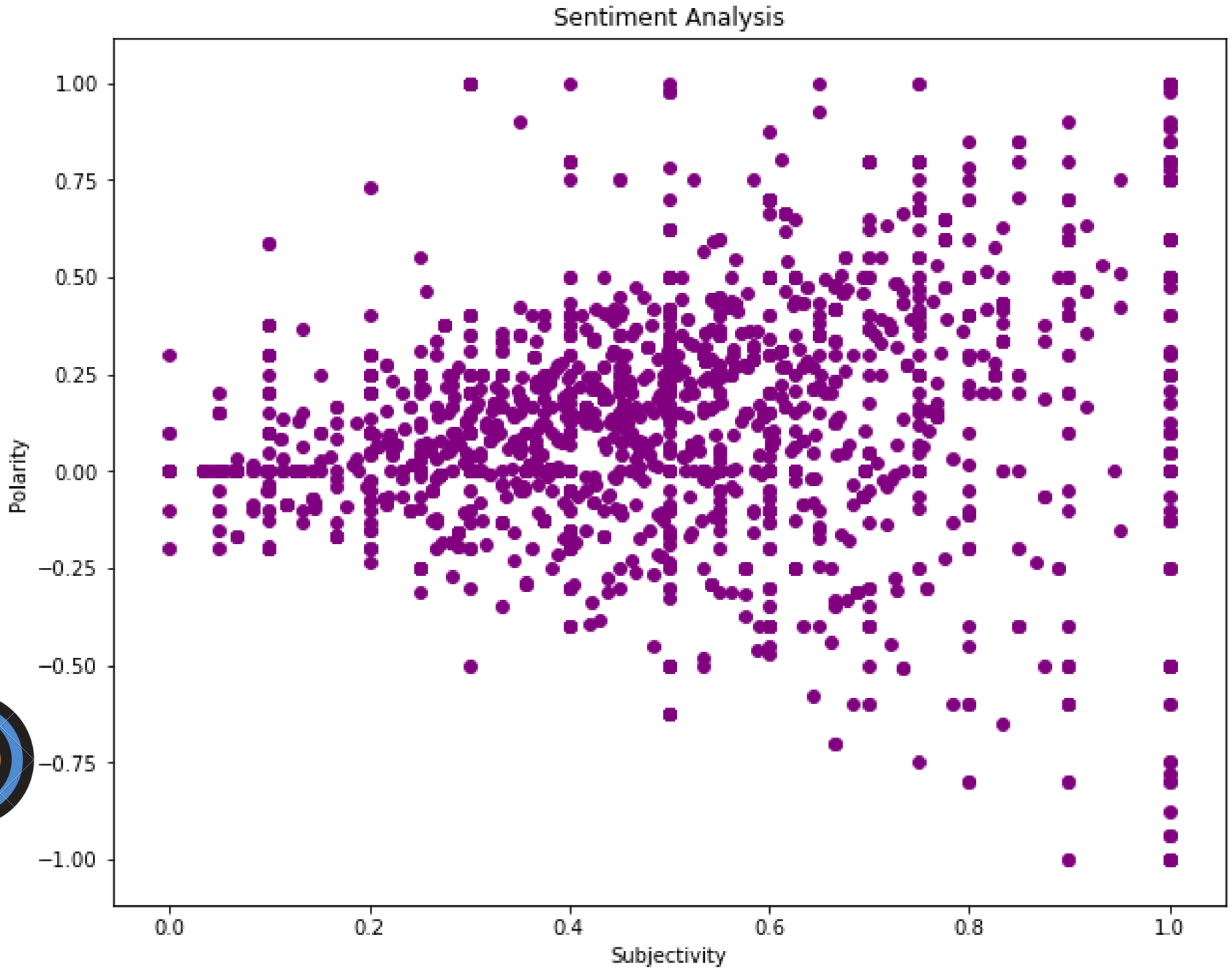
CLEANING DATA

- Tokenization
- Lemmatization
- Removing unwanted text & symbols
- Big string
- Wordclouds created

WHAT IS THE SENTIMENT?

- Visualisation of Sentiment
- 5000 dots/tweets
- Dots above 0 line represent positive statements
- Surprisingly, most of the dots lie exactly on 0, representing neutral statements
- Many dots with subjectivity = 1
- High subjective tweets move markets?



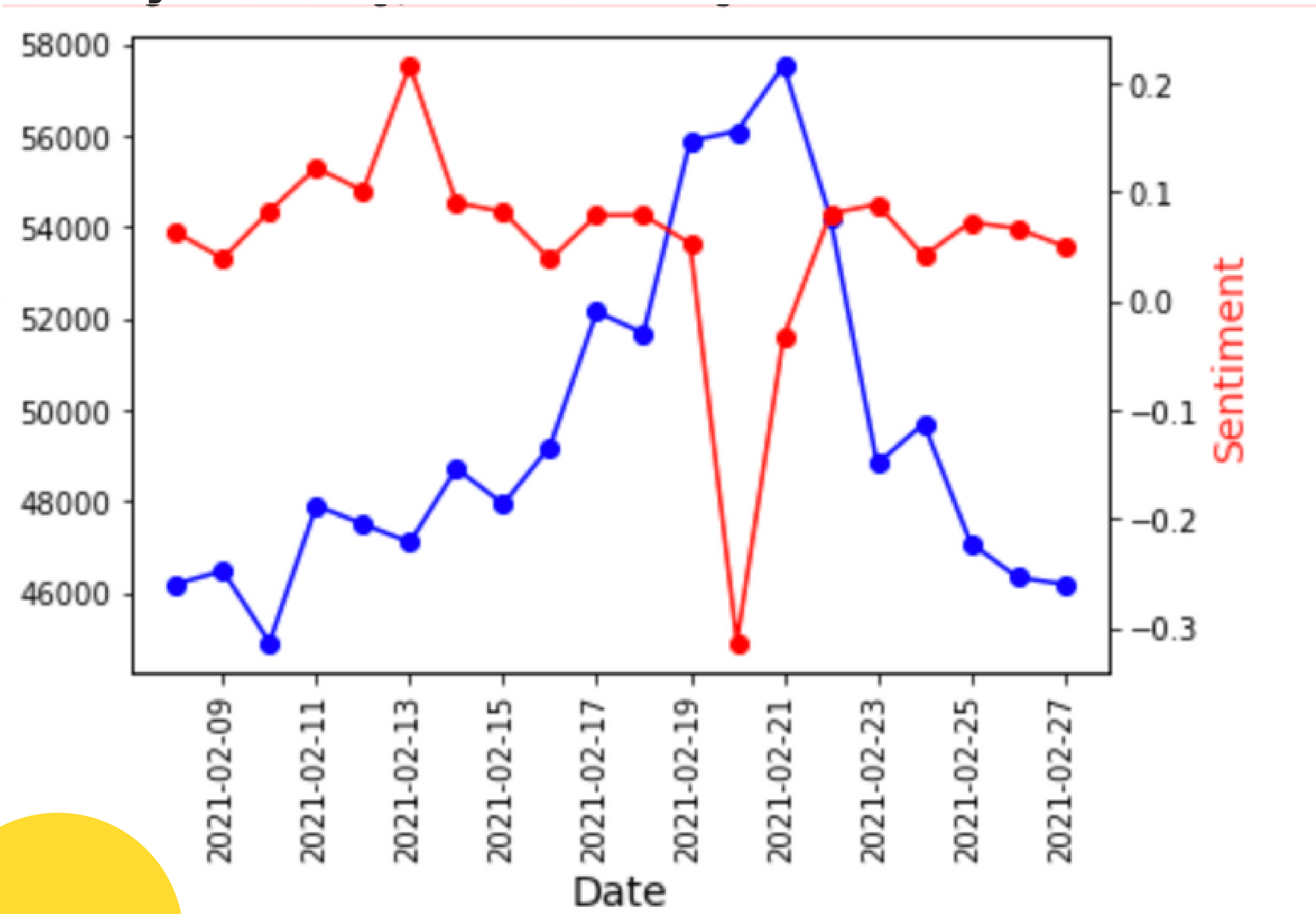


Did it work?

We first tried the sentiment analyser on 20 days of Twitter data, which showed that there was a strong correlation, and that there appeared to be a 1 day leading signal.

You will see that the sentiment in red spikes up, and a day later, the bitcoin price in the blue rises as well.

It can also be seen that after the large downswing in sentiment, the price reverses course.

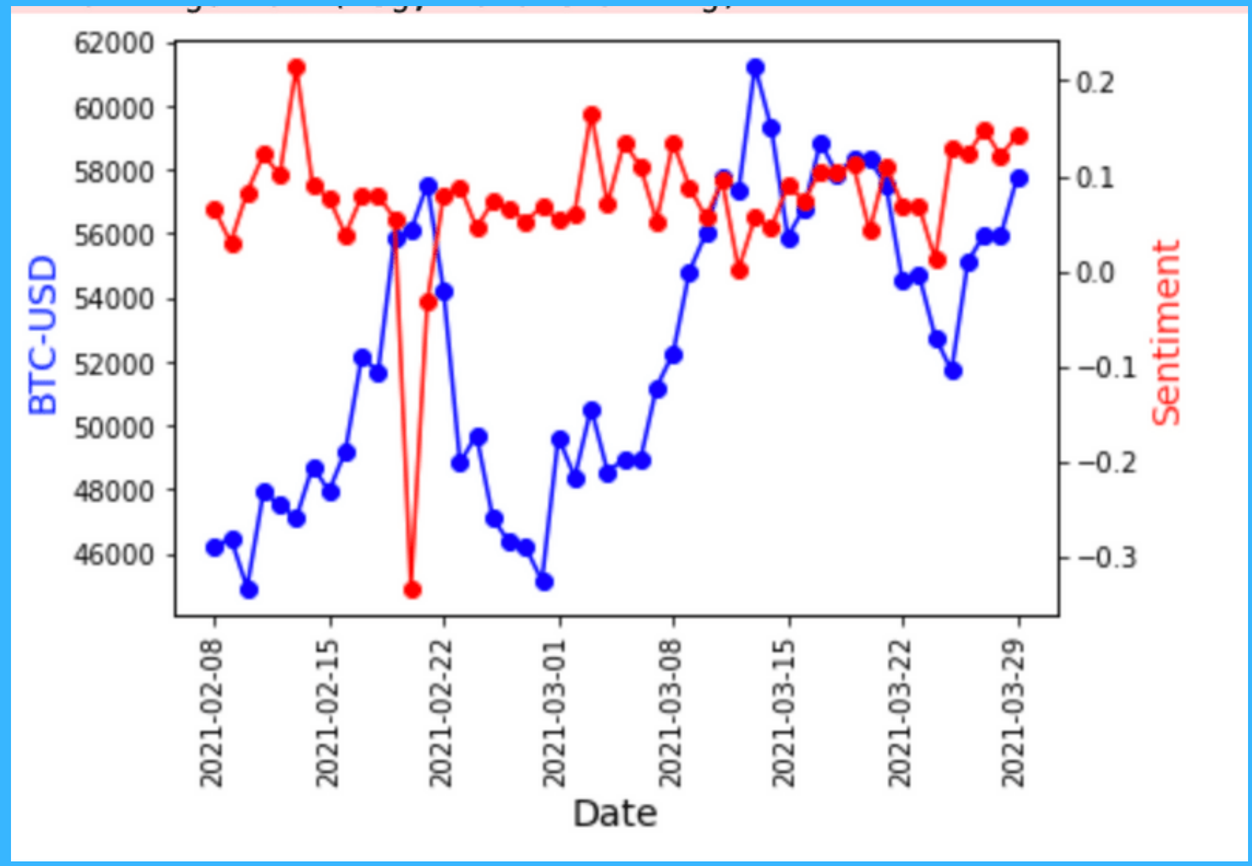


50 Days Of Twitter

This next chart plots the 50 days of sentiment data we collected.

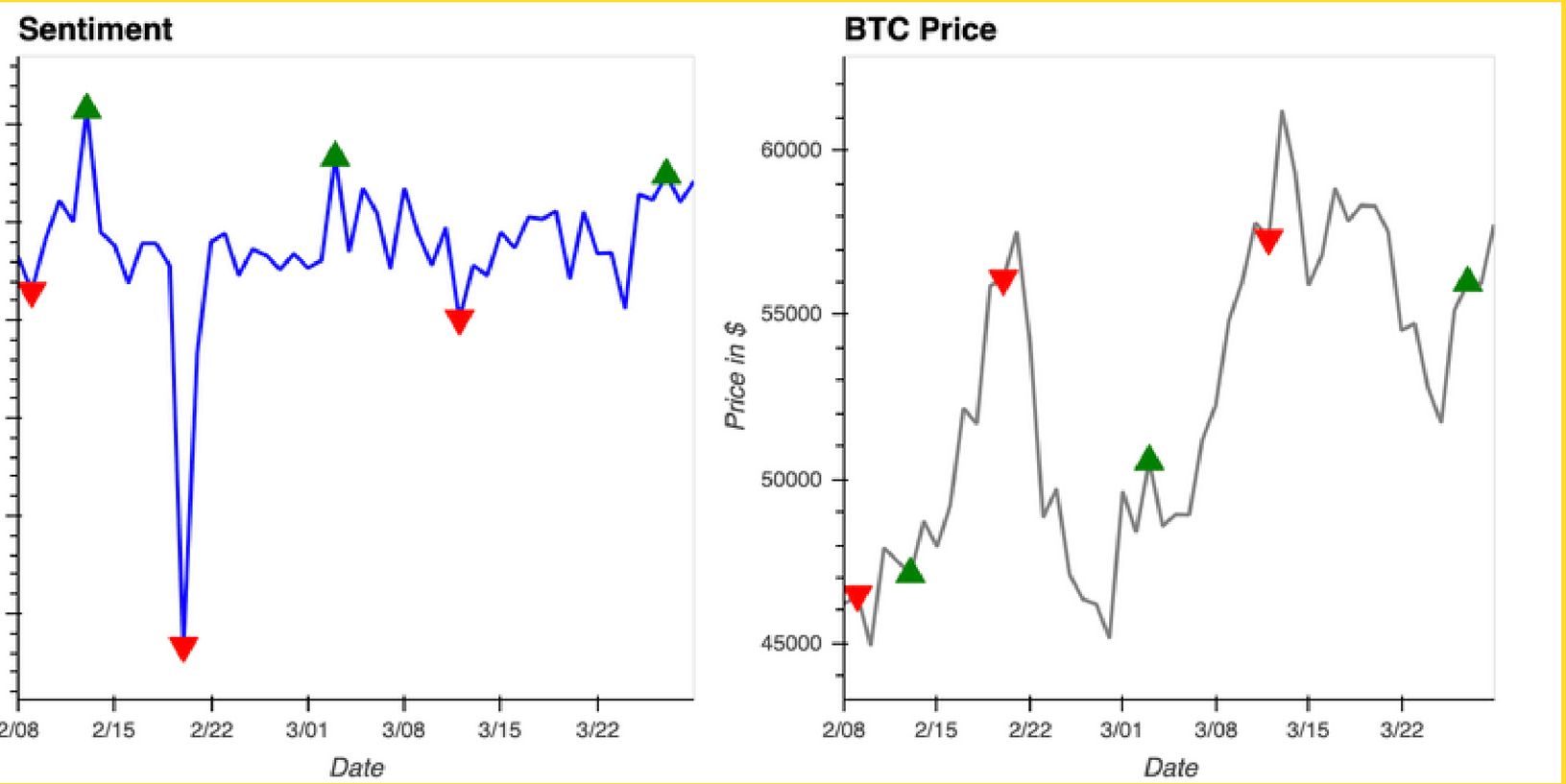
We've concluded from this that:

- There is definitely a leading signal of anywhere from 1 -2 days.
- Our precision would improve with more data (probably costing an arm and leg to get from Twitter)
- There definitely seems to be a visual correlation between the sentiment, and the price action of Bitcoin.
- We decided to test this theory by creating a basic algo-trader, and give ourselves a metric to measure and compare with.



Part 2: The Algo Trader

How did it do?



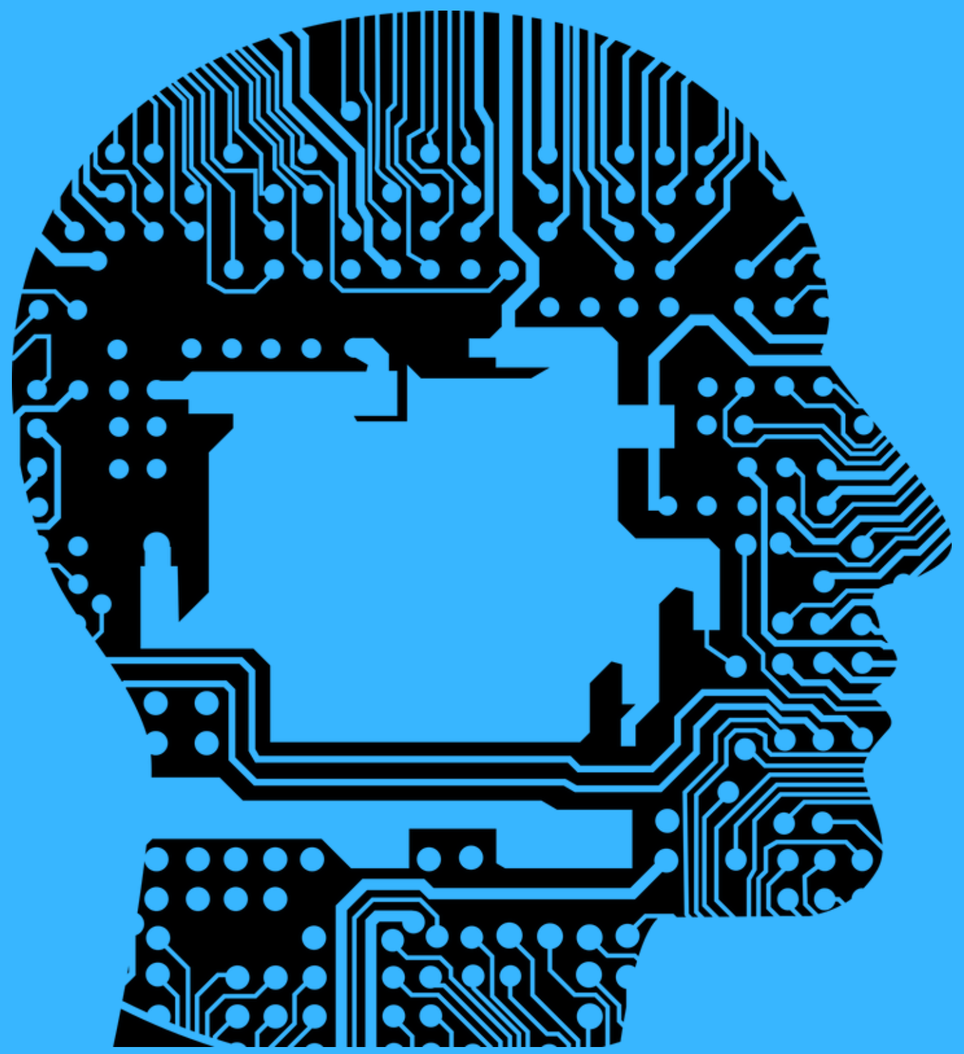
- We used our 50 days of bitcoin data cross-referenced with the sentiment analysis.
- Cleaned the data (removed columns)
- Created the signals
- Sentiment > 0.13 --> Buying Signal
- Sentiment < 0.03 --> Selling Signal
- Generated a return of 54.7% over 50 days!
- We believe that the bias in buy/sell signals is created by the bias of positive tweets vs negative tweets (35% vs 15%)

Postmortem

Our research indicates that here are approximately 400,000 Tweets per day that reference our keyword 'Bitcoin'. Having access to only 100 tweets per day would have a significant impact on the precision of our model, due to the small dataset, but we feel that the results look promising and worth further investigation with a larger data set. The best way would be to pay for greater Twitter API search access and increase the number of Tweets per day being analysed.

We also potentially 'overfitted' the algo-trading parameters to our data set for maximum profit over that time period - the system now needs to be backtested on more historical data to see if the system is robust, accurate and still profitable.

Our sentiment analyser struggled to analyse some human speech such as foreign languages and emoticons. We feel with a bigger datset and further data cleaning before analysis, more accurate results could be obtained by ignoring irrelevant terms within the tweets.



“

THANKS FOR LISTENING! ANY QUESTIONS?

— From the CryptoKnights team:
Shanshan, Kai, James, Alisha