Philipps Universität Marburg

Speech-driven sentiment analysis on cryptocurrency podcasts

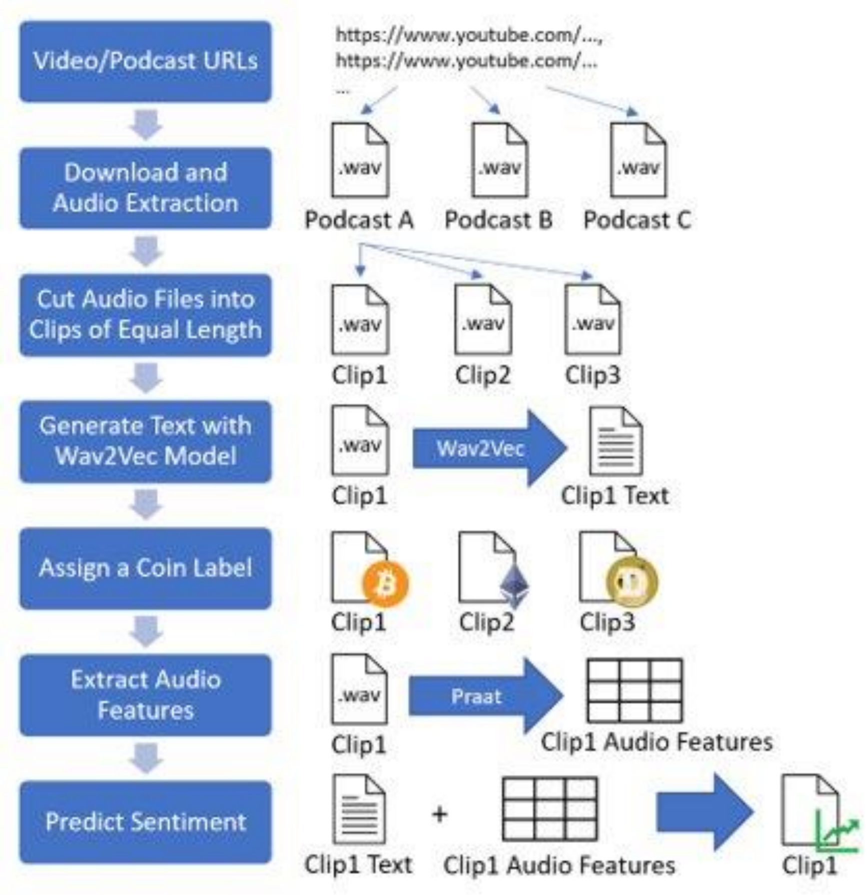
Bashir Hussein, Christian Kujath, Ryan Ford, Tristan Jacobs

Introduction

- Covid pandemic: People forced to stay home
  - Average US-American Income rose due to Covid-relief program [1]
- Social media driven investment boom [2]
  - "classic" stock market: 🚀 GAME, AMC, windeln.de
  - "new" cryptocurrencies: 💎 BTC , ETH, DOGE ⍰
- Podcasts on the rise (NYT: 1619, Joe Rogan, Coronavirus Update) [3]
- Sentiment-driven ETF Investment: non-performer [4]

→ We can do it better!

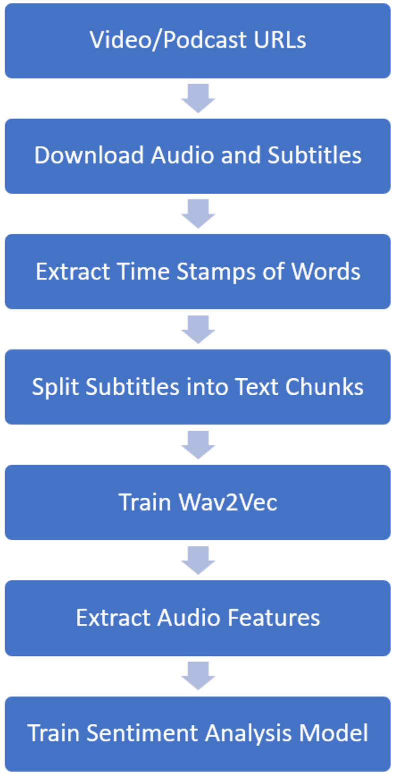
Overview



Our data pipeline: from URLs to sentiments.

Data Collection

- Speech to text Model
  - Input: audio clip
  - Label: correct transcript
- Sentiment Model
  - Input: transcript from audio clip
  - Label: bullish, bearish, neutral



Video/Podcast URLs

Download Audio and Subtitles

Extract Time Stamps of Words

Split Subtitles into Text Chunks

Train Wav2Vec

Extract Audio Features

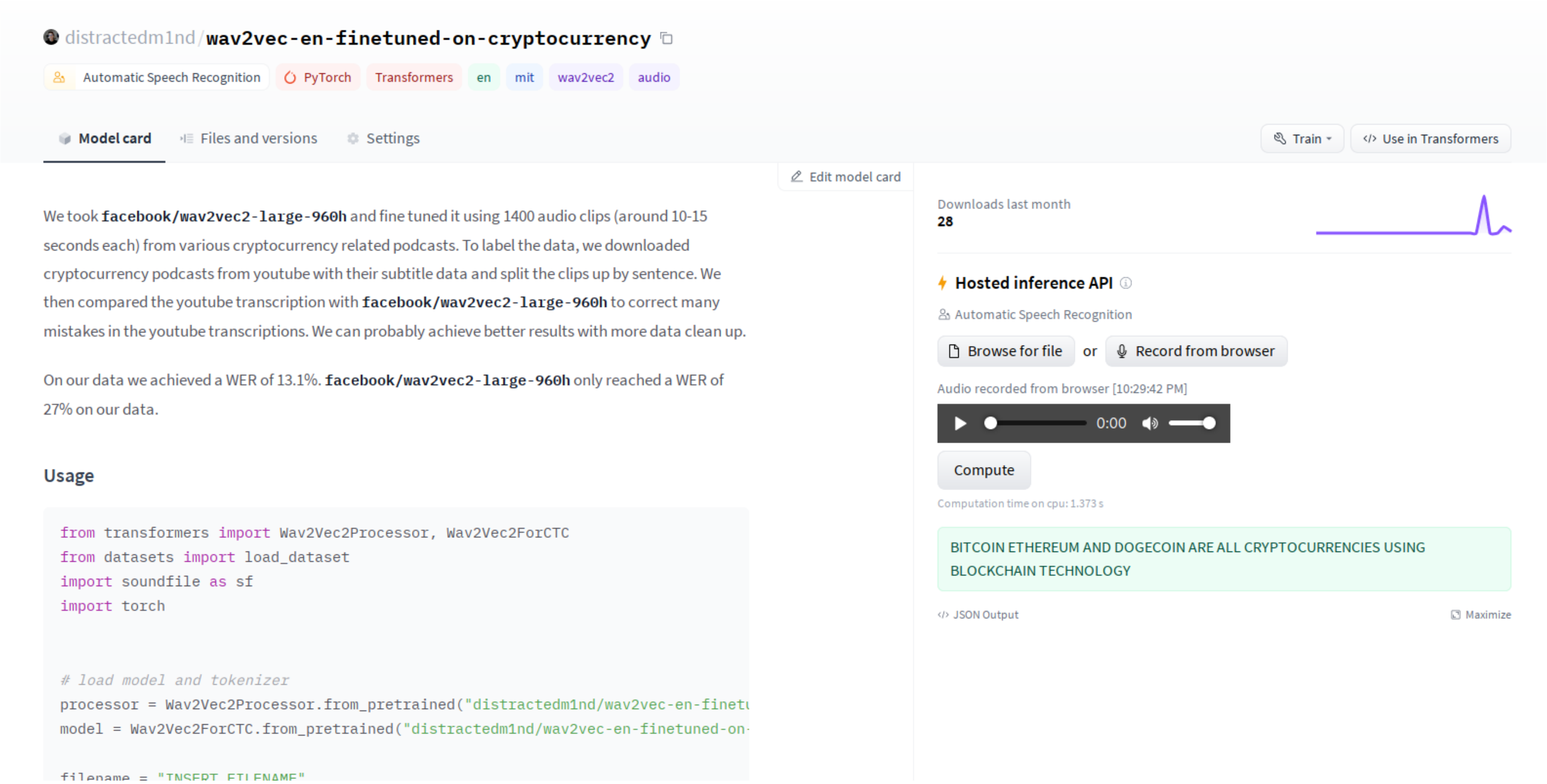
Train Sentiment Analysis Model

Pipeline for data collection

Wav2Vec2

- Wav2Vec2 trained using connectionist temporal classification (CTC)
- Finetuned on pretrained facebook/wav2vec2-large-960h
- Used PyTorch and 🤗 Transformers
- ~1400 clips of length 12-15 seconds for training data (30/70 split), transcriptions manually corrected
- Finetuning: Training a model to learn to align the pretrained representations to new words + update likelihoods of letters appearing together/being skipped
- Our model is available here and is testable in the browser!
- Before: 27% WER (words not mapped successfully).
- After: 13.1% WER. This is competitive, especially for messy speech data

Wav2Vec2 – Available on 🤗



We took **facebook/wav2vec2-large-960h** and fine tuned it using 1400 audio clips (around 10-15 seconds each) from various cryptocurrency related podcasts. To label the data, we downloaded cryptocurrency podcasts from youtube with their subtitle data and split the clips up by sentence. We then compared the youtube transcription with **facebook/wav2vec2-large-960h** to correct many mistakes in the youtube transcriptions. We can probably achieve better results with more data clean up.

On our data we achieved a WER of 13.1%. **facebook/wav2vec2-large-960h** only reached a WER of 27% on our data.

Usage

```
from transformers import Wav2Vec2Processor, Wav2Vec2ForCTC
from datasets import load_dataset
import soundfile as sf
import torch


# load model and tokenizer
processor = Wav2Vec2Processor.from_pretrained("distractedm1nd/wav2vec-en-finetu
model = Wav2Vec2ForCTC.from_pretrained("distractedm1nd/wav2vec-en-finetuned-on-

filename = "INSERT FILENAME"
```

Downloads last month
28
Hosted inference API
Automatic Speech Recognition
Browse for file
or
Record from browser
Audio recorded from browser [10:29:42 PM]
0:00
Compute
Computation time on cpu: 1.373 s

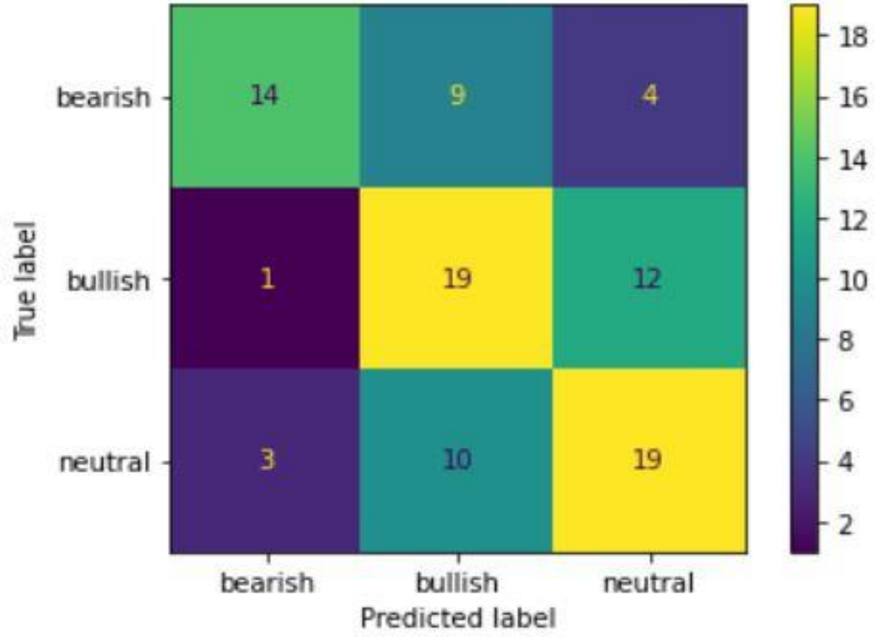
BITCOIN ETHEREUM AND DOGECOIN ARE ALL CRYPTOCURRENCIES USING BLOCKCHAIN TECHNOLOGY

Coin Prediction & Audio Features

- Use simple regex to determine the coin
- Use Praat to extract Audio Feature
    - Pitch 0.05 Quantile, Pitch 0.95 Quantile, Pitch Range (0.05 – 0.95 Quartile), Pitch Stdev, Pitch Median, Jitter, Shimmer, Hammarberg Index
- Used in conjunction with the transcription for the sentiment analysis model

Sentiment Analysis

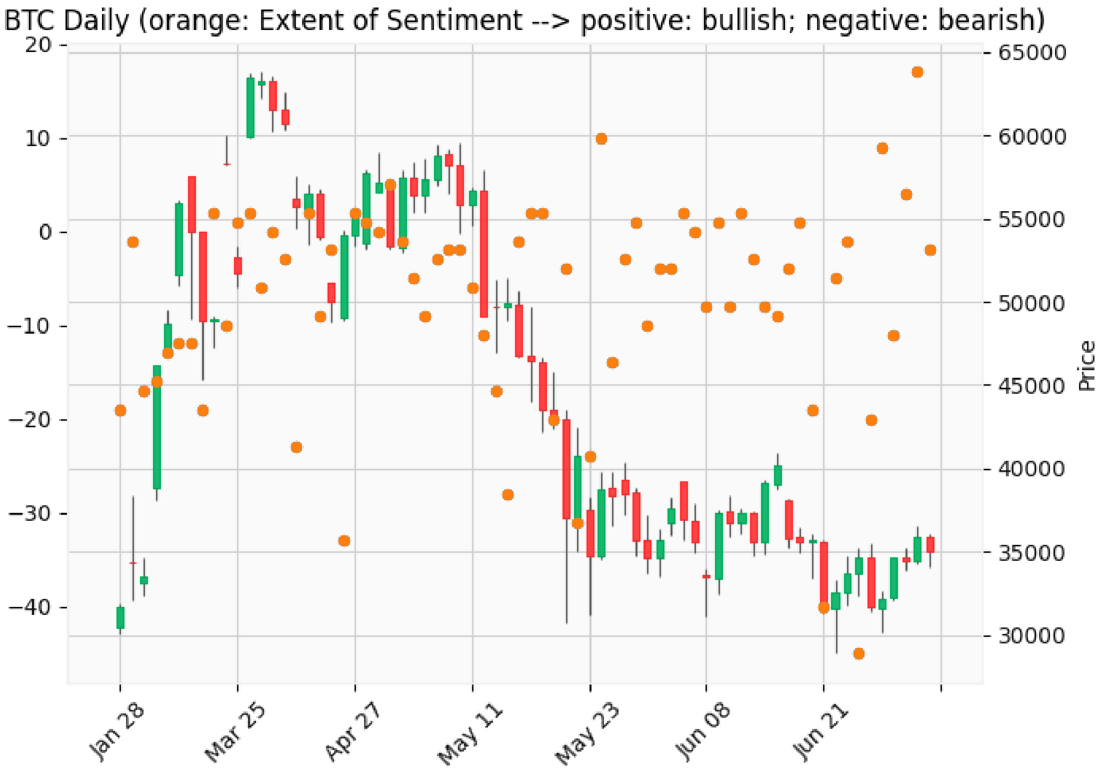
- Used scikit-learn library
- Text represented as Tf-idf
- Audio features concatenated with the Tf-idf vector
- Classifier
  - Multi-layer Perceptron classifier
    - 100 hidden layers
    - ReLU activation
- Achieved 56% accuracy
- Confusion Matrix:

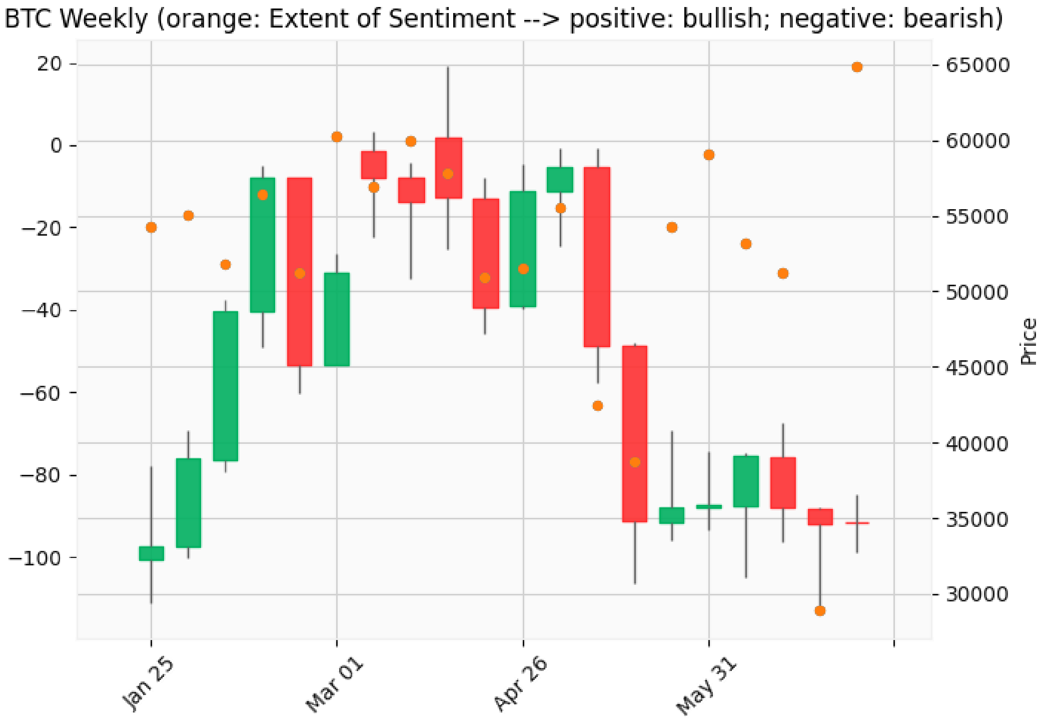


Sentiment Analysis

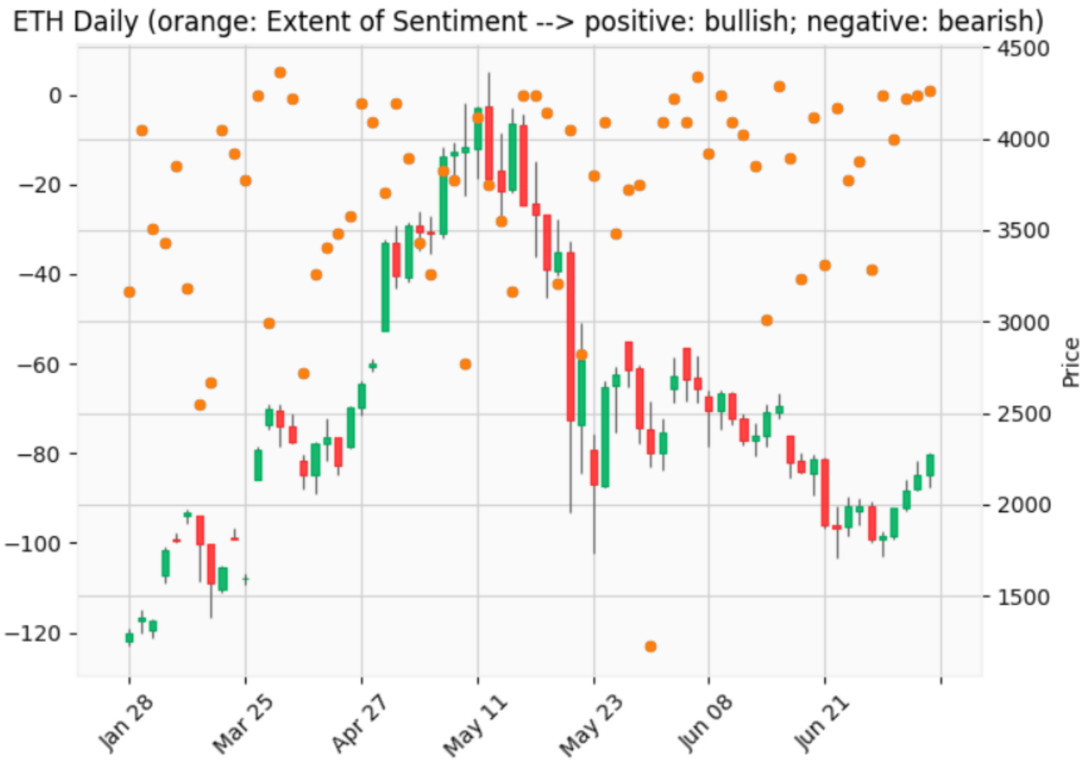
- Problems with the data
  - Texts ended up being too short to independently capture meaningful sentiment (30 words)
    - By the time we noticed this, it was already too late to restart the labelling process
  - Labelling extremely time consuming
  - The data was in fact NOT better than social media, for many reasons
    - Dialogue is messy
    - Tweets contain coherent ideas, discussions have much longer time dependencies
    - Data overwhelmingly positive (very few podcasts are negative about anything)
- Problems with the model
  - Did not experiment enough, only used basic textual representations
  - Each clip treated as individual data point, loses time dependencies

Results & Evaluation

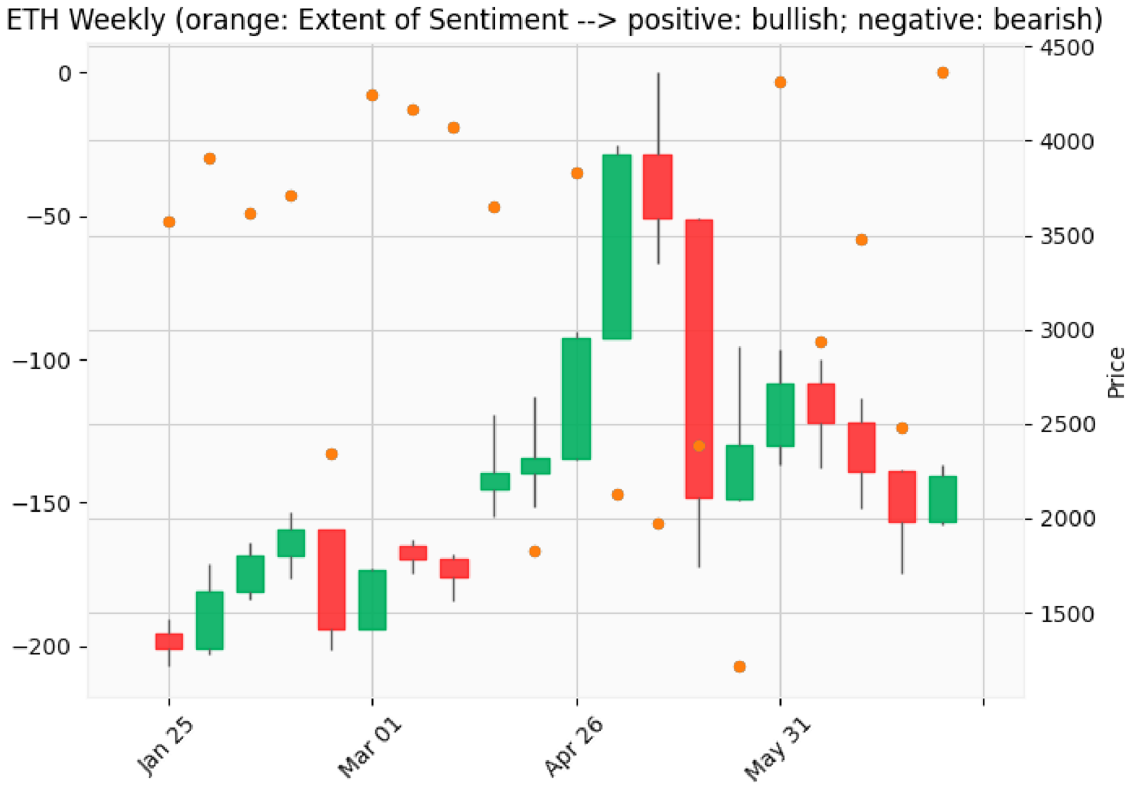




Results & Evaluation

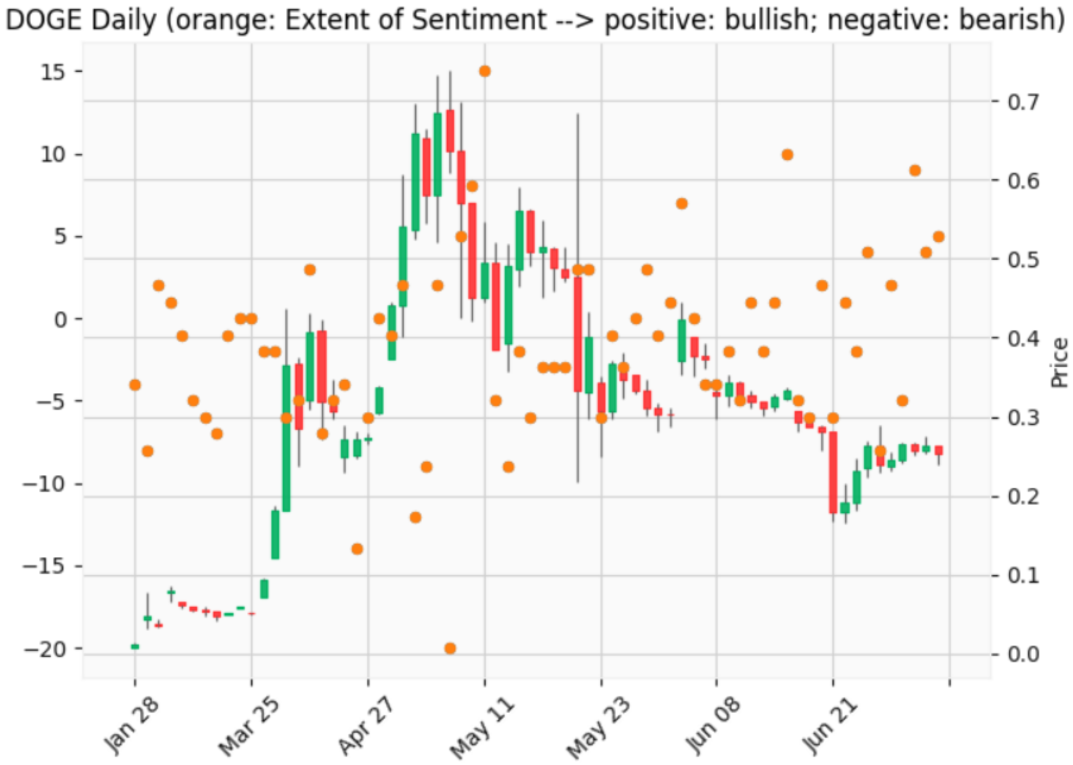


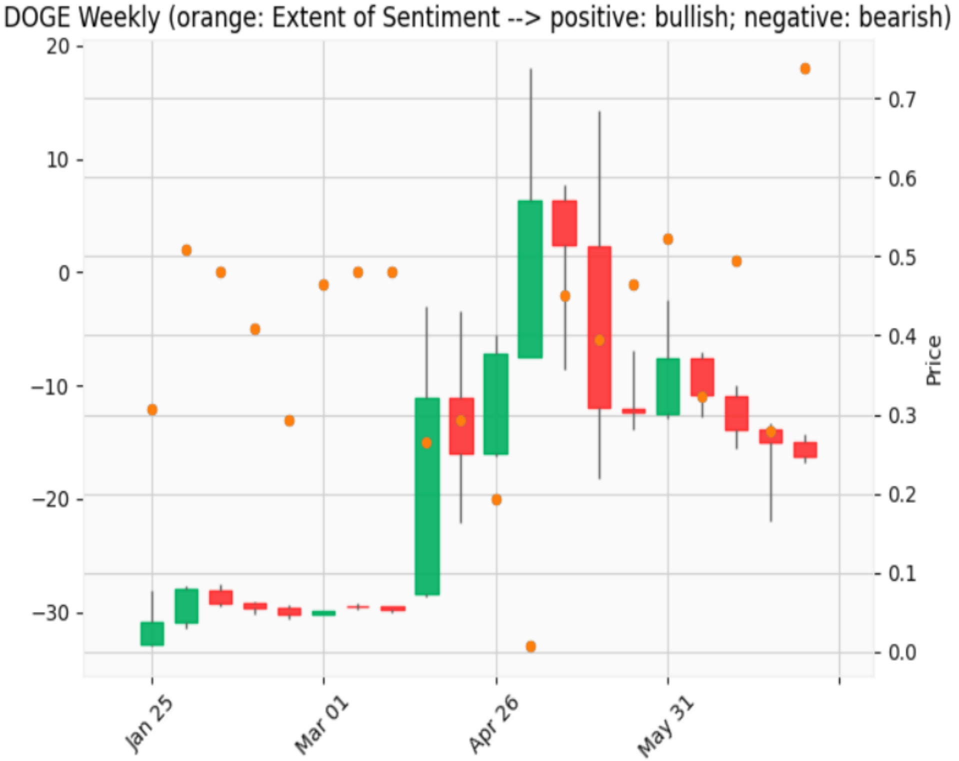
ETH Daily (orange: Extent of Sentiment --> positive: bullish; negative: bearish)



ETH Weekly (orange: Extent of Sentiment --> positive: bullish; negative: bearish)

Results & Evaluation


DOGE Daily (orange: Extent of Sentiment --> positive: bullish; negative: bearish)


DOGE Weekly (orange: Extent of Sentiment --> positive: bullish; negative: bearish)

Future Work

- More Data!
  - Manual dataset labelling surprisingly time consuming
- Tracking time dependencies over audio clips instead of analysing them individually
- Using better textual representations like word embeddings
  - These would have to be self trained, as the space is new and constanly making new words
- Better audio feature extraction methods
  - Also integration with textual representation instead of only concatenation
- Using another model to determine the coin instead of only regex?

Conclusion & Discussion

- Quality of Sentiment Analysis okay-ish
- Prediction efforts did not play out very well (no time series analysis)
- Wav2Vec does really well on just a few hours of training data
- Are podcasts actually a good data source (at all)?
- Are opinion-led Sentiment Analyses applicable to high-volatile financial assets, such as cryptocurrencies?