ReadMe

(The work breakdown is present in Section 2, and the error metric is specified in Section 3.)

Presentation of the Project

The goal of the project is to successfully measure the "sentiment" towards Bitcoin and the cryptocurrency ecosystem and use it to identify good zones for selling or buying (tops or bottoms) Bitcoin. It will require us to collect data, specifically YouTube comments, and correctly label them as positive or negative. The percentage of positive comments and the number of comments each day will provide us with information to create our own indicators.

1 Run the Project

Part 1: Setting Up the Environment

Step 1: Run the Python script, `setup_script.py`, in a terminal with the command:

```
python setup_script.py
```

It will automatically create a virtual environment called 'venv' and install the dependencies inside the `requirements.txt` file.

Part 2: Import All the Necessary Datasets

Step 2: Download the daily information about Bitcoin for the last years by running the `Import_bitcoin_data.ipynb` notebook. We use the 'yfinance' library, and the data is saved in a CSV file called `bitcoin_data.csv`.

Step 3: Download all the YouTube comments under the daily videos from the channel *Le Trone Crypto*. Run the notebook `Pull_all_Youtube_comments.ipynb`. Note that it will require a YouTube API key. It will store:

- All video IDs in the file `video_ids.csv`.
- All retrieved comments in `comments_data/combined_data.csv`.

Step 4: Download a labeled dataset of French tweets from Kaggle to fine-tune our model. Run the `Twitter_extract.ipynb` notebook (a `kaggle.json` file is required). The data will be saved in `french_tweets.csv`.

Part 3: Download and Fine-Tune the Model

Step 5: Run the Python script `Tweets_sampling.py` to randomly select 10,000 tweets from `french_tweets.csv` and save them in `sampled_tweets.csv`.

Step 6: Use the CamemBERT model from HuggingFace for labeling YouTube comments. Import and fine-tune it using `Fine-tunning.ipynb`.

Part 4: Label Comments and Create Visualizations

Step 7: Run `Labeling_yb_comments.ipynb`. Note: the process takes around 75 minutes. The labeled data will be saved in `comments_data/labeled_data.csv`.

Step 8: Run `First_visualisations.ipynb` to analyze the results, find correlations, and build the indicator.

2 Work Breakdown

- **Collecting Data:**
  - Choice of data: 10 hours
  - YouTube comments (API): 10 hours
  - Tweets (Kaggle): 2 hours
  - Bitcoin data (Yfinance): 30 minutes
- **Model and Training:**
  - Constructing my own model: 15 hours
  - Preparing data: 5 hours
  - Importing HuggingFace model: 3 hours
  - Training: 12 hours
- **Labeling Data and Visualizations:**
  - Preparing data: 4 hours

  - Analyzing results: 8 hours

- **Creating the Indicator:** 6 hours

3 Discussions

Model and Labeling Quality

The error metric is the accuracy of the predicted label compared to manual labeling. I aimed for 90% accuracy but achieved 86% using a custom model. Switching to CamemBERT improved the accuracy to 97% after fine-tuning.

YouTube Comments: Pros and Cons

Pros:

1. Large historical data since 2019.
2. Daily videos reflect timely sentiment.
3. Consistent format and community.

Cons/Challenges:

1. Comment volume grows over time.
2. Positive bias due to community behavior.
3. Inconsistent posting during vacations.

Choice of Twitter Data for Fine-Tuning

Tweets and YouTube comments are similar enough for classification tasks, and Twitter datasets are more accessible.

Insights That Helped Create the Indicator

1. Most people will lose in the market as it is a zero-sum game. Therefore, the biggest movements will always go against the dominant opinion. The definitive bottoms will be found in periods of fear and disinterest, while the major tops of markets occur when people are euphoric, and many get trapped in the market.
2. The number of comments and the percentage of positive comments are necessary and sufficient to build a reliable indicator. One translates the global interest in the market, and the other reflects the sentiment towards it.

3. Those sentiments are very volatile and won't be so reliable in the short term (daily analysis). Instead, we should focus on using them to predict trends at a larger time scale (weekly, monthly, or annually).

Performances of the Indicator

If we were to develop a strategy where:

- Buy when the indicator score is under -25 and the 7-day average is below the 30-day average,
- Sell when the indicator goes over 50 and the 7-day average exceeds the 30-day average,

we would have:

BOUGHT at the PRICES

```
2022-08-02    22978.117188
2022-09-25    18802.097656
2022-11-12    16799.185547
2022-12-27    16717.173828
2022-12-31    16547.496094
2023-06-06    27238.783203
2023-10-01    27983.750000
```

SOLD at the PRICES

```
2021-11-14    65466.839844
2021-11-17    60368.011719
2023-11-18    36585.703125
```

We identify that:

- The indicator perfectly captured the bottom of the market without sending false buy signals at the wrong time.
- The mean entry price is very low, offering a potential 500% gain with the current Bitcoin price.
- The indicator also perfectly identified the exact top of the market in the previous cycle. However, it gave a false signal of a top in November 2023.

The proposed strategy is very simple and should incorporate other elements, such as price action, before making a financial decision.

Limits

- The indicator has only one cycle worth of data, and we still need confirmation to verify if it is well-tuned for predictions in the new cycle.
- It can give false signals when the market is going up.
- The results given before the beginning of the year 2020 are unreliable because the number of comments was too low.

Further Integration

In the future, I aim to integrate the indicator into TradingView and automate daily data collection.