TABLE OF CONTENTS

1 INTRODUCTION

Knowing what the market is feeling at any point can prove to be a crucial factor in determining the line of action for managers, funds, and traders- both retail and institutional. But we are bombarded with so much information on several different social media and news outlets that it becomes difficult to summarize this information to make decisions in a timely manner. Having a way to aggregate the market's current and historical 'feeling' or 'sentiment' in a quick and efficient manner can be beneficial. For this task, I chose to do a sentiment analysis on the crypto market based on Tweets from users and visualize the results on a flask app. The developed Jupyter notebook pulls data on the crypto market periodically (every 5 minutes) and performs sentiment analysis on the cleaned dataset using some popular sentiment analysis libraries. The aggregate market sentiment and a percentage wise distribution of positive vs negative is created using plotly and hosted on a flask app.

2 LIBRARIES AND TOOLS USED

Library	Purpose
Twint	Twitter Data Collection
vaderSentiment	Sentiment Analysis
tetxtblob	Sentiment Analysis
Pandas	Data Analysis
JupyterDash and Dash	Creating Interactive Visualization Dashboard
Plotly	Data Visualization
Re (regular expressions)	Data Cleaning
datetime	Periodic Execution
Jupyter Notebooks and Python	Overall Development

3 SENTIMENT ANALYSIS USING VADER AND TEXTBLOB

3.1 DATA COLLECTION

The collection of data is done using Twint, an advanced Twitter data scraping library which can scrape tweets without using Twitter's API. I start off by creating a Twint configuration. Then I pass various constraints to a search

i. Keywords: crypto, bitcoin, Ethereum, DeFi

ii. Language: English because we only want to analyse tweets in English
iii. Verified: Search for tweets only from verified accounts

We launch the search and obtain the data.

3.2 Choosing Relevant Parameters

The returned twitter data consists of several columns:

```
Index(['conversation_id', 'created_at', 'date', 'day', 'hashtags', 'hour','id',
'link', 'location', 'name', 'near', 'nlikes', 'nreplies','nretweets', 'place',
'profile_image_url', 'quote_url', 'retweet','search', 'timezone', 'tweet',
'user_id', 'user_id_str', 'username'],dtype='object')
```

The search query returns the above data. For our analysis, the date, number of likes, tweet, and username will be useful. Hence, these columns will be imported into a pandas dataframe for cleaning and further analysis.

3.3 Data Cleaning

The returned twitter data needs to be prepared for sentiment analysis. To do this, the following steps are performed:

i. Dropping duplicates: It is possible that the data that was pulled might contain the same tweets more than once because a tweet could contain more than one keyword specified in the search query above. The data is dropped by checking the datetime column and seeing if more than one tweet was posted at the exact same time. This is more efficient than dropping duplicates by looking at the tweets column and it works because it is highly unlikely that any 2 tweets are posted at the exact same second.
ii. Removing emojis: By entering the tweets into a function deEmojify, which uses regular expressions, all emojis are removed from the tweets data.
iii. Removing https tags and special characters: Sentiment analysis libraries do not respond well to url links and special characters and will return an incorrect score. Hence, these will also be removed using regular expressions.

After performing the above steps, our dataframe is finally ready for sentiment analysis.

3.4 Sentiment Analysis

There are many popular libraries out there that perform sentiment analysis. For this project, I decided to o with 2 options- Vader and TextBlob. The visualization (explained later) will give the user an opportunity to see sentiment scores visualized using 2 different libraries, and hence get a clearer unbiased picture.

i. Vader (Valence Aware Dictionary and Sentiment Reasoner): A lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. It is used for sentiment analysis of text which has both the polarities i.e. positive/negative. VADER is used to quantify how much of positive or negative emotion the text has and also the intensity of emotion.
ii. TextBlob: TextBlob is built upon NLTK and provides an easy to use interface to the NLTK library. It can be used to perform a variety of NLP tasks ranging from parts-of-speech tagging to sentiment analysis, and language translation to text classification.

The tweets data will be passed into a sentiment function which will use return 2 sentiment analysis scores for each tweet- one from Vader and the other from TextBlob.

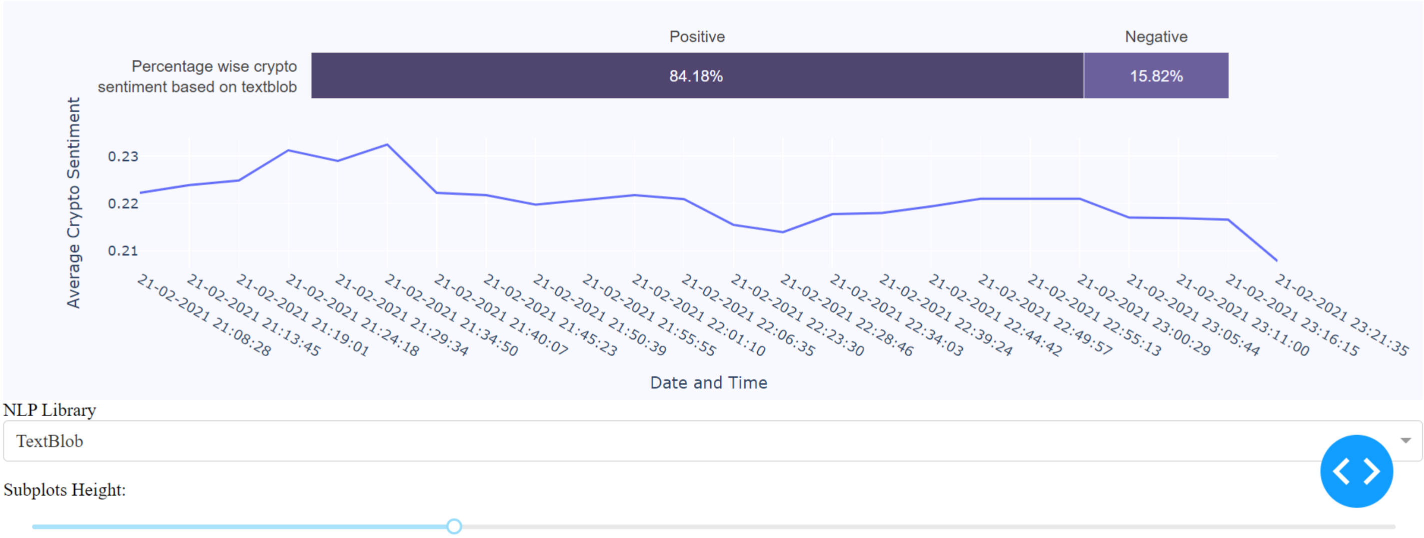
3.5 Visualization

The data will be visualized on a flask app using Dash and plotly. A user will be able to select which sentiment analysis score they wish to view by using a dropdown. They can select from Vader and TextBlob. They will also be able to change the size of the plots if they wish to analyze a plot more closely. The graphs will refresh everytime the data refreshes in the backend (every 5 minutes) and new data becomes available.

The following charts will be displayed for visualization:

i. Percentage of people having a positive sentiment on crypto vs percentage of people having a negative sentiment: This will be displayed using a bar chart and will beneficial if a user wishes to know exactly what percent of the market views crypto positively at that very instant.

ii. Average cumulative crypto score: The average sentiment score for every data pull will be visualized on a line graph. A user can see changing trends in the average sentiment.



NLP Library

TextBlob

Subplots Height:

4 Further Exploration

Data is around us everywhere and the project could be further explored by collecting data from various other social media platforms such as Facebook, LinkedIn, and popular crypto news websites such as CoinDesk, CoinTelegraph. Sentiment analysis could be performed other libraries to give users several different options to choose from. Finally, visualization could be enhanced to generate sentiment based on past day, month, and year, and maybe even on a customized timeline input by the user. Finally, we do not have to stop at crypto. Sentiment analysis can be performed on any kind of data and can be a useful tool in understanding people's sentiment on markets, social issues, politics etc.