	Unit 12 - Tales from the Crypto  1. Sentiment Analysis  Use the newsapi to pull the latest news articles for Bitcoin and Ethereum and create a DataFrame of sentiment scores for each coin.  Use descriptive statistics to answer the following questions:  1. Which coin had the highest mean positive score?  2. Which coin had the highest negative score?  3. Which coin had the highest positive score?  # Initial imports import os import pandas as pd
[3]:	<pre>from dotenv import load_dotenv import nltk as nltk nltk.download('vader_lexicon') from nltk.sentiment.vader import SentimentIntensityAnalyzer analyzer = SentimentIntensityAnalyzer()  %matplotlib inline  [nltk_data] Downloading package vader_lexicon to [nltk_data] /Users/syedahasan/nltk_data [nltk_data] Package vader_lexicon is already up-to-date!  # Read your api key environment variable load_doteny()</pre>
[3]: [4]:	<pre>load_dotenv() api_key = os.getenv("NEWS_API") type(api_key)</pre>
[8]: [8]:	<pre># Fetch the Bitcoin news articles bitcoin_news = newsapi.get_everything(q='bitcoin', language="en") bitcoin_news.keys()  dict_keys(['status', 'totalResults', 'articles'])  print(f"Total articles about Bitcoin: {bitcoin_news['totalResults']}")  Total articles about Bitcoin: 9561</pre>
13]: 13]:	<pre># Fetch the Ethereum news articles eth_news = newsapi.get_everything(q="Ethereum" , language="en") eth_news.keys()  dict_keys(['status', 'totalResults', 'articles'])  print(f"Total articles about Ethereum: {eth_news['totalResults']}")</pre>
16]:	<pre># Create the Bitcoin sentiment scores DataFrame # classnote 12.2 btc_sentiments = [] for article in bitcoin_news["articles"]:     try:         text = article["content"]         date = article["publishedAt"][:10]         sentiment = analyzer.polarity_scores(text)         compound = sentiment["compound"]</pre>
	<pre>pos = sentiment["pos"] neu = sentiment["neu"] neg = sentiment["neg"] # append each element of each iteration to the btc_sentiments empty list btc_sentiments.append({     "text": text,     "date": date,     "compound": compound,     "positive": pos,     "negative": neg,     "neutral": neu })</pre>
161.	<pre>except AttributeError:     pass  btc_df = pd.DataFrame(btc_sentiments) cols = ["date", "text", "compound", "positive", "negative", "neutral"] btc_df = btc_df[cols]  btc_df.head()  date</pre>
16]:	date         text         compound positive negative neutral           0         2021-06-09         El Salvador's President Nayib Bukele has made         0.8402         0.282         0.000         0.718           1         2021-06-09         El Salvador has become the first country in th         0.1280         0.043         0.000         0.957           2         2021-06-14         It's all about clean energy, it seems. \r\nElo         0.6908         0.169         0.000         0.831           3         2021-07-05         Filed under:\r\nThe supply chain attack has re         -0.5719         0.111         0.184         0.705           4         2021-06-09         image captionThe move means bitcoin will be ac         0.2732         0.060         0.000         0.940
17]:	<pre># Create the Ethereum sentiment scores DataFrame eth_sentiments = [] for article in eth_news["articles"]:     try:         text = article["content"]         date = article["publishedAt"][:10]         sentiment = analyzer.polarity_scores(text)         compound = sentiment["compound"]         pos = sentiment["pos"]         neu = sentiment["neu"]         neg = sentiment["neg"]</pre>
	<pre># append each element of each iteration to the eth_sentiments empty list eth_sentiments.append({     "text": text,     "date": date,     "compound": compound,     "positive": pos,     "negative": neg,     "neutral": neu }) except AttributeError:     pass</pre>
17]:	<pre>eth_df = pd.DataFrame(eth_sentiments) cols = ["date", "text", "compound", "positive", "negative", "neutral"] eth_df = eth_df[cols] eth_df.head()  date</pre>
18]:	2       2021-06-10       This article was translated from our Spanish e       -0.3400       0.000       0.066       0.934         3       2021-07-02       Bitcoin and Ethereum\r\nYuriko Nakao\r\nEther       0.3612       0.110       0.041       0.849         4       2021-07-05       Ether holders have "staked" more than \$13 bill       0.7717       0.194       0.000       0.806         # Describe the Bitcoin Sentiment btc_df.describe()         compound positive negative neutral
18]:	compound         positive         negative         neutral           count         20.000000         20.000000         20.000000           mean         0.136730         0.076250         0.036750         0.88695           std         0.409336         0.070288         0.054092         0.08129           min         -0.612400         0.000000         0.000000         0.70500           25%         -0.050575         0.032250         0.000000         0.84025           50%         0.128000         0.065500         0.000000         0.90500
19]: 19]:	75% 0.369700 0.107250 0.064750 0.94375  max 0.840200 0.282000 0.184000 1.00000  # Describe the Ethereum Sentiment eth_df.describe()  compound positive negative neutral  count 20.000000 20.000000 20.000000 20.000000
	mean         0.063475         0.040500         0.025150         0.934350           std         0.356132         0.058708         0.032347         0.053135           min         -0.381800         0.000000         0.000000         0.806000           25%         -0.273200         0.000000         0.000000         0.920000           50%         0.000000         0.000000         0.936500           75%         0.361200         0.074750         0.060250         0.961000           max         0.771700         0.194000         0.080000         1.000000
	Questions:  Q: Which coin had the highest mean positive score?  A: Bitcoin has the highest mean positive score 0.076250  Q: Which coin had the highest compound score?  A: Bitcoin has the highest coumpund score 0.840200
,	Q. Which coin had the highest positive score?  A: Bitcoin has the highest positive score 0.282000  2. Natural Language Processing
	Tokenizer  In this section, We use NLTK and Python to tokenize the text for each coin. Make sure to:  1. Lowercase each word.  2. Remove Punctuation.  3. Remove Stopwords.  from nltk.tokenize import word tokenize, sent tokenize
49] <b>:</b>	<pre>from nltk.corpus import stopwords from nltk.stem import WordNetLemmatizer, PorterStemmer from string import punctuation import re  # Instantiate the lemmatizer # class note 12.1 lemmatizer = WordNetLemmatizer()</pre>
50]:	<pre># Create a list of stopwords sw = set(stopwords.words('english'))  # Expand the default stopwords list if necessary sw_addon = {"."}  # Complete the tokenizer function def tokenizer(text):     """Tokenizes text."""</pre>
	<pre># Remove the punctuation from text regex = re.compile("[^a-zA-Z ]") re_clean = regex.sub('', text)  # Create a tokenized list of the words words = word_tokenize(re_clean)  # Lemmatize words into root words lem = [lemmatizer.lemmatize(word) for word in words]  # Convert the words to lowercase</pre>
51]:	<pre># Remove the stop words tokens = [word.lower() for word in lem if word.lower() not in sw.union(sw_addon)]  return tokens  # Create a new tokens column for Bitcoin #tokenizer(btc_df['text'][0])  btc_tokens = [] for text in btc_df['text']:</pre>
51]:	tokens = tokenizer(text) btc_tokens.append(tokens) btc_df['tokens'] = btc_tokens btc_df.head()  date text compound positive negative neutral tokens  0 2021-06- El Salvador's President Nayib Bukele has made 0 2021-06- El Salvador has become the first country in 0.1280 0.043 0.000 0.957 [el, salvador, ha, become, first, country]
52]:	th  2021-06- 14 It's all about clean energy, it seems. \r\nElo  3 2021-07- 05 Filed under:\r\nThe supply chain attack has re  4 2021-06- 09 image captionThe move means bitcoin will be ac  10 0.000 0.000 0.937 wor  10 0.000 0.937 wor  10 0.000 0.937 wor  10 0.000 0.931 [clean, energy, seemselon, musk, tesla caused  11 0.184 0.705 [filed, underthe, supply, chain, attack has ha, r  12 0.01-06- 09 image captionThe move means bitcoin will be ac  10 0.000 0.940 [image, captionthe, move, mean bitcoin, accep]
	<pre># Create a new tokens column for Ethereum #tokenizer(eth_df['text'][0])  eth_tokens = [] for text in eth_df['text']:    tokens = tokenizer(text)    eth_tokens.append(tokens) eth_df['tokens'] = eth_tokens eth_df.head()</pre> date text compound positive negative neutral tokens
52]:	datetextcompoundpositivenegativeneutraltokens02021-06- 17TL;DR: The Cryptocurrency with Ethereum and So0.00000.0001.000[tldr, cryptocurrency, ethereum solidity, blo12021-06- 20It wasn't long ago that your average person ha0.29600.0000.0610.939[wasnt, long, ago, average, person clue, nft,22021-06- 10This article was translated from our Spanish e0.34000.0000.0660.934[article, wa, translated, spanish edition, us32021-07- 02Bitcoin and Ethereum\r\nYuriko Nakao\r\nEther overtook0.36120.1100.0410.849[bitcoin, ethereumyuriko, nakaoether overtook
	4 2021-07- Ether holders have "staked" more than \$13 0.7717 0.194 0.000 0.806 [ether, holder, staked, billion, worth crypto  NGrams and Frequency Analysis  In this section you will look at the ngrams and word frequency for each coin.
[53]: [59]:	<pre>1. Use NLTK to produce the n-grams for N = 2. 2. List the top 10 words for each coin.  from collections import Counter from nltk import ngrams  # Generate the Bitcoin N-grams where N=2 btc_ngram_counts = [] for tokens in btc_df['tokens']:</pre>
59]:	<pre>ngrams_count = dict(Counter(ngrams(tokens, n=2)))   btc_ngram_counts.append(ngrams_count.items()) btc_ngram_counts  [dict_items([(('el', 'salvadors'), 1), (('salvadors', 'president'), 1), (('president', 'nayib'), 1), (('nayi'bukele'), 1), (('bukele', 'ha'), 1), (('ha', 'made'), 1), (('made', 'good'), 1), (('good', 'promise'), 1), (('promise', 'adopt'), 1), (('adopt', 'bitcoin'), 1), (('bitcoin', 'legal'), 1), (('legal', 'tender'), 1), ender', 'officials'), 1), (('officials', 'central'), 1), (('central', 'american'), 1), (('american', 'country'), 1), (('country', 'congress'), 1), (('congress', 'voted'), 1), (('voted', 'accept'), 1), (('accept', 'crocurrency'), 1), (('cryptocurrency', 'majori'), 1), (('majori', 'char'), 1)]),</pre>
	<pre>dict_items([(('el', 'salvador'), 1), (('salvador', 'ha'), 1), (('ha', 'become'), 1), (('become', 'first'),</pre>
	ack'), 1), (('attack', 'ha'), 1), (('ha', 'reached'), 1), (('reached', 'thousand'), 1), (('thousand', 'organ tionsillustration'), 1), (('organizationsillustration', 'alex'), 1), (('alex', 'castro'), 1), (('castro', 'vethree'), 1), (('vergethree', 'day'), 1), (('day', 'ransomware'), 1), (('ransomware', 'attacker'), 1), (('atkarted'), 1), (('started', 'holiday'), 1), (('holiday', 'weekend'), 1), (('weekend', 'comprom'), 1), (('comprom', 'char'), 1)]), dict_items([(('image', 'captionthe'), 1), (('captionthe', 'move'), 1), (('move', 'mean'), 1), (('mean', 'biin'), 1), (('bitcoin', 'accepted'), 1), (('accepted', 'everywhere'), 1), (('everywhere', 'good'), 1), (('good'service'), 1), (('salvador', 'alongside'), 1), (('alongside', 'us'), 1), (('us', 'dollarel'), 1), (('first', 'county', 'locuntry', 'world'), 1), (('world', 'officially'), 1), (('officially', 'classify'), 1), (('classify', 'bitcoin'), 1), (('bitcoin', 'char'), 1)]),
	<pre>dict_items([(('several', 'crypto'), 1), (('crypto', 'fan'), 1), (('fan', 'descended'), 1), (('descended', 'mi'), 1), (('miami', 'florida'), 1), (('florida', 'last'), 1), (('last', 'weekend'), 1), (('weekend', 'larget'), 1), (('largest', 'bitcoin'), 1), (('bitcoin', 'conference'), 1), (('conference', 'history'), 1), (('histy', 'saying'), 1), (('saying', 'theyve'), 1), (('theyve', 'tested'), 1), (('tested', 'positive'), 1), (('posve', 'covid'), 1), (('covid', 'larry'), 1), (('larry', 'cermak'), 1), (('cermak', 'research'), 1), (('research', 'director'), 1), (('director', 'char'), 1)]),     dict_items([(('hello', 'friend'), 1), (('friend', 'welcome'), 1), (('welcome', 'back'), 1), (('back', 'week'), ('week', 'reviewlast'), 1), (('reviewlast', 'week'), 1), (('week', 'wrote'), 1), (('wrote', 'tech'), 1)     (('tech', 'taking'), 1), (('taking', 'disney'), 1), (('disney', 'week'), 1), (('week', 'im'), 1), (('im', 'ting'), 1), (('talking', 'search'), 1), (('search', 'new'), 1), (('new', 'crypto'), 1), (('tec', 'char'), 1)]),</pre>
	<pre>dict_items([(('maryann', 'russonbusiness'), 1), (('russonbusiness', 'reporter'), 1), (('reporter', 'bbc'),   (('bbc', 'newsimage'), 1), (('newsimage', 'copyrightgetty'), 1), (('copyrightgetty', 'imagesformer'), 1), ((     agesformer', 'us'), 1), (('us', 'president'), 1), (('president', 'donald'), 1), (('donald', 'trump'), 1), ((     ump', 'ha'), 1), (('ha', 'told'), 1), (('told', 'fox'), 1), (('fox', 'business'), 1), (('business', 'see'),     (('see', 'bitcoin'), 1), (('bitcoin', 'scam'), 1), (('scam', 'affecting'), 1), (('affecting', 'value'), 1),     (('value', 'us'), 1), (('us', 'dolla'), 1), (('dolla', 'char'), 1)]),     dict_items([(('image', 'copyrightgetty'), 1), (('copyrightgetty', 'imagesthe'), 1), (('imagesthe', 'gang'),     1), (('gang', 'behind'), 1), (('behind', 'colossal'), 1), (('colossal', 'ransomware'), 1), (('ransomware', 'ack'), 1), (('attack', 'ha'), 1), (('ha', 'demanded'), 1), (('demanded', 'paid'), 1), (('paid', 'bitcoin'),     (('bitcoin', 'return'), 1), (('return', 'universal'), 1), (('universal', 'decryptor'), 1), (('decryptor', 'sy'), 1), (('say', 'unlock'), 1), (('unlock', 'file'), 1), (('file', 'char'), 1)]),     diet_items([('say', 'say', 'unlock'), 1), (('say', 'inpo', 'say', 'unlock', 'file'), 1), (('file', 'char'), 1)])</pre>
	<pre>dict_items([(('san', 'salvador'), 1), (('salvador', 'june'), 1), (('june', 'reuters'), 1), (('reuters', 'el 1), (('el', 'salvador'), 1), (('salvador', 'replace'), 1), (('replace', 'us'), 1), (('us', 'dollar'), 1), (('lar', 'bitcoin'), 1), (('bitcoin', 'legal'), 1), (('legal', 'tender'), 1), (('tender', 'finance'), 1), (('fince', 'minister'), 1), (('minister', 'alejandro'), 1), (('alejandro', 'zelaya'), 1), (('zelaya', 'said'), 1) (('said', 'wednesday'), 1), (('wednesday', 'central'), 1), (('central', 'american'), 1), (('american', 'natin'), 1), (('nation', 'char'), 1)]), dict_items([(('london', 'june'), 1), (('june', 'reuters'), 1), (('reuters', 'el'), 1), (('el', 'salvador'), 1), (('salvador', 'determined'), 1), (('determined', 'push'), 1), (('push', 'ahead'), 1), (('ahead', 'making'), (('making', 'bitcoin'), 1), (('bitcoin', 'legal'), 1), (('legal', 'tender'), 1), (('tender', 'process'), 1), (('process', 'bring'), 1), (('bring', 'small'), 1), (('small', 'risk'), 1), (('risk', 'prove'), 1), (('pr', 'leap'), 1), (('leap', 'forward'), 1), (('forward', 'humanity'), 1), (('humanity', 'country'), 1), (('cory', 'char'), 1)]),</pre>
	<pre>dict_items([(('london', 'june'), 1), (('june', 'reuters'), 1), (('reuters', 'el'), 1), (('el', 'salvador'), 1), (('salvador', 'determined'), 1), (('determined', 'push'), 1), (('push', 'ahead'), 1), (('ahead', 'making'), 1), (('making', 'bitcoin'), 1), (('bitcoin', 'legal'), 1), (('legal', 'tender'), 1), (('tender', 'process'), 1), (('process', 'bring'), 1), (('bring', 'small'), 1), (('small', 'risk'), 1), (('risk', 'prove'), 1), (('pe', 'leap'), 1), (('leap', 'forward'), 1), (('forward', 'humanity'), 1), (('humanity', 'country'), 1), (('cory', 'pr'), 1), (('pr', 'char'), 1)]),     dict_items([(('representations', 'virtual'), 1), (('virtual', 'currency'), 1), (('currency', 'bitcoin'), 1)     (('bitcoin', 'stand'), 1), (('stand', 'motherboard'), 1), (('motherboard', 'picture'), 1), (('picture', 'ill ration'), 1), (('illustration', 'taken'), 1), (('taken', 'may'), 1), (('may', 'reutersdado'), 1), (('reuterso', 'ruvicillustrationfile'), 1), (('ruvicillustrationfile', 'photomicrostrategy'), 1), (('photomicrostrategy'), 1), (('photomicrostrategy'), 1), (('inc', 'mstro'), 1), (('staffsan', 'salvador'), 1), (('salvador', 'reuters'), 1), (('ruterso', 'staffsan'), 1), (('staffsan', 'salvador'), 1), (('salvador', 'reuters'), 1), (('ruterso', 'salvador'), 1), (('salvador', 'reuterso'), 1), (('ruterso', 'staffsan'), 1), (('staffsan', 'salvador'), 1), (('salvador', 'reuterso'), 1), (('ruterso', 'salvador'), 1), (('salvador', 'reuterso'), 1), (('ruterso', 'salvador'), 1), (('salvador', 'reuterso'), 1), (('salvador', 'reuterso'), 1), (('salvador'), 1), (('</pre>
	<pre>dict_items([('reuters', 'staffsan'), 1), (('staffsan', 'salvador'), 1), (('salvador', 'reuters'), 1), (('rers', 'el'), 1), (('el', 'salvador'), 1), (('salvador', 'replace'), 1), (('replace', 'us'), 1), (('us', 'dol r'), 1), (('dollar', 'bitcoin'), 1), (('bitcoin', 'legal'), 1), (('legal', 'tender'), 1), (('tender', 'finare'), 1), (('finance', 'minister'), 1), (('minister', 'alejandro'), 1), (('alejandro', 'zelaya'), 1), (('zela', 'said'), 1), (('said', 'wednesday'), 1), (('wednesday', 'central'), 1), (('central', 'american'), 1), (erican', 'char'), 1)]),    dict_items([(('colonial', 'pipeline'), 1), (('pipeline', 'biggest'), 1), (('biggest', 'recent'), 1), (('rect', 'ransomware'), 1), (('ransomware', 'attack'), 1), (('attack', 'targeted'), 1), (('targeted', 'jbs'), 1), (('jbs', 'meat'), 1), (('meat', 'supplier'), 1), (('supplier', 'thats'), 1), (('thats', 'responsible'), 1), (('responsible', 'one'), 1), (('one', 'quarter'), 1), (('quarter', 'meat'), 1), (('meat', 'processing'), 1), (('processing', 'us'), 1), (('us', 'late'), 1), (('late', 'last'), 1), (('last', 'week'), 1), (('week', 'comy'), 1), (('company', 'char'), 1)]),</pre>
	dict_items([(('hello', 'welcome'), 1), (('welcome', 'back'), 1), (('back', 'toequity'), 1), (('toequity', 'hcrunchs'), 1), (('techcrunchs', 'venture'), 1), (('venture', 'capitalfocused'), 1), (('capitalfocused', 'post'), 1), (('podcast', 'unpack'), 1), (('unpack', 'number'), 1), (('number', 'behind'), 1), (('behind', 'heanesthis'), 1), (('headlinesthis', 'equity'), 1), (('equity', 'monday'), 1), (('monday', 'morning'), 1), (('ning', 'coffee'), 1), (('coffee', 'chat'), 1), (('chat', 'ab'), 1), (('ab', 'char'), 1)]), dict_items([(('story', 'originally'), 1), (('originally', 'appeared'), 1), (('appeared', 'marketbeatwhen'), 1), (('marketbeatwhen', 'meme'), 1), (('meme', 'coin'), 1), (('coin', 'surged'), 1), (('surged', 'earlier'), 1), (('earlier', 'year'), 1), (('year', 'remember'), 1), (('remember', 'dogecoin'), 1), (('dogecoin', 'mania', 'mightve'), 1), (('mightve', 'easy'), 1), (('easy', 'dump'), 1), (('dump', 'everything'), 1), (('everything', 'owned'), 1), (('owned', 'stock'), 1), (('stock', 'bond'), 1), (('bond', 'gold'), 1), (('gold'crypto'), 1), (('crypto', 'char'), 1)]),
	<pre>dict_items([(('bitcoin', 'surged'), 1), (('surged', 'wednesday'), 1), (('wednesday', 'adding'), 1), (('adding', 'previous'), 1), (('previous', 'closebitcoin'), 1), (('closebitcoin', 'world'), 1), (('world', 'biggest'), (('biggest', 'bestknown'), 1), (('bestknown', 'cryptocurrency'), 1), (('cryptocurrency', 'year'), 1), (ar', 'low'), 1), (('low', 'jan'), 1), (('jan', 'char'), 1)]), dict_items([(('reuters', 'stafffile'), 1), (('stafffile', 'photo'), 1), (('photo', 'representations'), 1), (('representations', 'bitcoin'), 1), (('bitcoin', 'cryptocurrency'), 1), (('cryptocurrency', 'seen'), 1), (en', 'picture'), 1), (('picture', 'illustration'), 1), (('illustration', 'taken'), 1), (('taken', 'june'), 1), (('june', 'reutersedgar'), 1), (('reutersedgar', 'suillustrationreuters'), 1), (('suillustrationreuters', 'koin'), 1), (('bitcoin', 'dropped'), 1), (('dropped', 'char'), 1)]), dict_items([(('bitcoin', 'dropped'), 1), (('dropped', 'gmt'), 1), (('gmt', 'saturday'), 1), (('saturday', ing'), 1), (('losing', 'previous'), 1), (('previous', 'closebitcoin'), 1), (('closebitcoin', 'world'), 1), orld', 'biggest'), 1), (('biggest', 'bestknown'), 1), (('bestknown', 'cryptocurrency'), 1), (('cryptocurrency'), 1), (</pre>
60]:	<pre>'high'), 1), (('high', 'char'), 1)])]  # Generate the Ethereum N-grams where N=2 eth_ngram_counts = [] for tokens in eth_df['tokens']:     ngrams_count = dict(Counter(ngrams(tokens, n=2)))     eth_ngram_counts.append(ngrams_count.items()) eth_ngram_counts</pre> [dict items([(('tldr', 'cryptocurrency'), 1), (('cryptocurrency', 'ethereum'), 1), (('ethereum', 'solidity'))
60]:	<pre>[dict_items([(('tldr', 'cryptocurrency'), 1), (('cryptocurrency', 'ethereum'), 1), (('ethereum', 'solidity') 1), (('solidity', 'blockchain'), 1), (('blockchain', 'developer'), 1), (('developer', 'bundle'), 1), (('bunde', 'sale'), 1), (('sale', 'june'), 1), (('june', 'saving'), 1), (('saving', 'list'), 1), (('list', 'priceis'), (('priceis', 'everyone'), 1), (('everyone', 'know'), 1), (('know', 'investing'), 1), (('investing', 'crycurrency'), 1), (('cryptocurrency', 'char'), 1)]),     dict_items([(('wasnt', 'long'), 1), (('long', 'ago'), 1), (('ago', 'average'), 1), (('average', 'person'),     (('person', 'clue'), 1), (('clue', 'nft'), 1), (('nft', 'wa'), 1), (('wa', 'nowadays'), 1), (('nowadays', 'tre'), 1), (('theyre', 'making'), 1), (('making', 'headline'), 1), (('headline', 'selling'), 1), (('selling', undred'), 1), (('folk', 'thousand'), 1), (('thousand', 'buck'), 1), (('buck', 'predictably'), 1), (('predictably', 'folk'), 1), (('folk', 'learning'), 1), (('learning', 'nfts'), 1), (('nfts', 'char'), 1)]),     dict_items([(('article', 'wa'), 1), (('wa', 'translated'), 1), (('translated', 'spanish'), 1), (('technology', 'edition'), 1), (('ditousand', 'using', 'ai'), 1), (('ai', 'technology'), 1), (('technology', 'edition'), 1), (('technology', 'edition'</pre>
	rs'), 1), (('errors', 'may'), 1), (('may', 'exist'), 1), (('exist', 'due'), 1), (('due', 'processlast'), 1), (('processlast', 'may'), 1), (('may', 'vitalik'), 1), (('vitalik', 'buterin'), 1), (('buterin', 'became'), 1 (('became', 'youngest'), 1), (('youngest', 'billionaire'), 1), (('billionaire', 'world'), 1), (('world', 'ye'), ('ye', 'char'), 1)]), dict_items([(('bitcoin', 'ethereumyuriko'), 1), (('ethereumyuriko', 'nakaoether'), 1), (('nakaoether', 'oveok'), 1), (('overtook', 'bitcoin'), 1), (('bitcoin', 'number'), 1), (('number', 'active'), 1), (('active', 'ress'), 1), (('demand', 'faltering'), 1), (('network', 'friday'), 1), (('friday', 'sign'), 1), (('sign', 'demand'), 1), (('cryptocurrency', 'world'), 1), (('world', 'biggest'), 1), (('biggest', yptocurrency'), 1), (('cryptocurrency', 'co'), 1), (('co', 'char'), 1)]), dict_items([(('ether', 'holder'), 1), (('holder', 'staked'), 1), (('staked', 'billion'), 1), (('billion', 'th'), 1), (('worth', 'cryptocurrency'), 1), (('cryptocurrency', 'ethereum'), 1), (('ethereum', 'network'), 1), (('network', 'excitement'), 1), (('excitement', 'build'), 1), (('build', 'around'), 1), (('around', 'overhau
	<pre>l'), 1), (('overhaul', 'system'), 1), (('system', 'potential'), 1), (('potential', 'growth'), 1), (('growth' 'staki'), 1), (('staki', 'char'), 1)]), dict_items([(('ether', 'cryptocurrency'), 1), (('cryptocurrency', 'ethereum'), 2), (('ethereum', 'networkph thekgetty'), 1), (('networkphotothekgetty', 'imagesether'), 1), (('imagesether', 'holder'), 1), (('holder', aked'), 1), (('staked', 'billion'), 1), (('billion', 'worth'), 1), (('worth', 'cryptocurrency'), 1), (('ethereum', 'network'), 1), (('network', 'excitement'), 1), (('excitement', 'char'), 1)]), dict_items([(('major', 'upgrade'), 1), (('upgrade', 'ethereum'), 1), (('ethereum', 'network'), 1), (('network', 'could'), 1), (('could', 'help'), 1), (('help', 'turn'), 1), (('turn', 'crypto'), 1), (('crypto', 'staki g'), 1), (('staking', 'billion'), 1), (('billion', 'business'), 1), (('business', 'according'), 1), (('accorg', 'analyst'), 1), (('analyst', 'jpmorganthe'), 1), (('earn', 'cryptocurrency'), 1), (('cryptocurrency', 'r'), 1)]),</pre>
	<pre>dict_items([(('sir', 'tim'), 1), (('tim', 'bernerslee'), 1), (('bernerslee', 'credited'), 1), (('credited', nventor'), 1), (('inventor', 'world'), 1), (('world', 'wide'), 1), (('wide', 'web'), 1), (('web', 'selling') 1), (('selling', 'source'), 1), (('source', 'code'), 1), (('code', 'invention'), 1), (('invention', 'nft'), (('nft', 'sothebys'), 1), (('sothebys', 'auctionthe'), 1), (('auctionthe', 'nft'), 1), (('nft', 'nonfungible 1), (('nonfungible', 'token'), 1), (('token', 'type'), 1), (('type', 'cryp'), 1), (('cryp', 'char'), 1)]), dict_items([(('two', 'ethereumbased'), 1), (('ethereumbased', 'protocol'), 1), (('protocol', 'keep'), 1), (ep', 'network'), 1), (('network', 'nucypher'), 1), (('nucypher', 'getting'), 1), (('getting', 'upgradesinto 1), (('upgradesinto', 'single'), 1), (('single', 'network'), 1), (('network', 'last'), 1), (('last', 'week') 1), (('week', 'token'), 1), (('token', 'holder'), 1), (('holder', 'project'), 1), (('project', 'voted'), 1)] dict_items([(('youve', 'likely'), 1), (('likely', 'seen'), 1), (('seen', 'headline'), 1), (('headline', 'sunding'), 1), (('surrounding', 'pretty'), 1), (('pretty', 'major'), 1), (('major', 'decline'), 1), (('decline'), 1), (('sale', 'market'), 1), (('market', 'recent'), 1), (('recent', 'peak'), 1)</pre>
	<pre>(('peak', 'report'), 1), (('report', 'indicating'), 1), (('indicating', 'much'), 1), (('much', 'percent'), 1) (('percent', 'decline'), 1), (('decline', 'recent'), 1), (('recent', 'week'), 1), (('week', 'whi'), 1), (('week', 'whi'), 1), (('week', 'whi'), 1), (('week', 'char'), 1)]), dict_items([(('ambition', 'crypto'), 1), (('crypto', 'investor'), 1), (('investor', 'swelled'), 1), (('swelled', 'even'), 1), (('even', 'faster'), 1), (('faster', 'market'), 1), (('market', 'ha'), 1), (('ha', 'recent', '(char'), 'month'), 1), (('month', 'institutional'), 1), (('institutional', 'player'), 1), (('player', ountain'), 1), (('mountain', 'blockchain'), 1), (('blockchain', 'data'), 1), (('data', 'try'), 1), (('try', ke'), 1), (('make', 'sense'), 1), (('sense', 'without'), 1), (('without', 'par'), 1), (('par', 'char'), 1)]) dict_items([(('bitcoin', 'surged'), 1), (('surged', 'wednesday'), 1), (('wednesday', 'adding'), 1), (('adding', 'previous'), 1), (('previous', 'closebitcoin'), 1), (('closebitcoin', 'world'), 1), (('world', 'biggest', 'bestknown'), 1), (('bestknown', 'cryptocurrency'), 1), (('cryptocurrency', 'year'), 1), (ar', 'low'), 1), (('low', 'jan'), 1), (('jan', 'char'), 1)]),</pre>
	<pre>dict_items([(('bitcoin', 'dropped'), 1), (('dropped', 'gmt'), 1), (('gmt', 'saturday'), 1), (('saturday', 'ing'), 1), (('losing', 'previous'), 1), (('previous', 'closebitcoin'), 1), (('closebitcoin', 'world'), 1), orld', 'biggest'), 1), (('biggest', 'bestknown'), 1), (('bestknown', 'cryptocurrency'), 1), (('cryptocurrency'), 1), (('cryptocurrency'), 1), (('ethereum', 'solidity'), 1), (('solidity', 'cryptocurrency'), 1), (('cryptocurrency', 'ethereum'), 1), (('developer', 'bundle'), 1), (('bundle', 'explains'), 1), (('explains', 'entire'), 1), (('entire', 'crypto'), 1), (('crypto', 'phenomenon'), 1), (('phenomenon', 'also'), 1), (('also', 'explores'), 1), (('explores', 'creating'), 1), (('creating', 'real'), 1), (('real', 'world'), 1), (('world', 'cryptocurrencyif'), 1), (('cryptocurrencyif', 'knowledge'), 1), (('kledge', 'char'), 1)]), dict_items([(('ina', 'fassbenderafp'), 1), (('fassbenderafp', 'via'), 1), (('via', 'getty'), 1), (('getty', magesinvestors'), 1), (('imagesinvestors', 'pulled'), 1), (('pulled', 'money'), 1), (('money', 'digital'), 1)</pre>
	<pre>(('digital', 'asset'), 1), (('asset', 'third'), 1), (('third', 'week'), 1), (('week', 'running'), 1), (('rung', 'making'), 1), (('making', 'longest'), 1), (('longest', 'series'), 1), (('series', 'consecutive'), 1), onsecutive', 'outflow'), 1), (('outflow', 'since'), 1), (('since', 'february'), 1), (('february', 'coinsh'), 1), (('coinsh', 'char'), 1)]),    dict_items([(('bitcoin', 'surged'), 1), (('surged', 'gmt'), 1), (('gmt', 'sunday'), 1), (('sunday', 'adding'), (('adding', 'previous'), 1), (('previous', 'closebitcoin'), 1), (('closebitcoin', 'world'), 1), (('world', 'biggest'), 1), (('biggest', 'bestknown'), 1), (('bestknown', 'cryptocurrency'), 1), (('cryptocurrency', 'yer'), 1), (('year', 'low'), 1), (('low', 'char'), 1)]),    dict_items([(('reuters', 'staffjune'), 1), (('staffjune', 'reuters'), 1), (('reuters', 'bitcoin'), 1), (('soing', 'dipped'), 1), (('dipped', 'gmt'), 1), (('gmt', 'friday'), 1), (('friday', 'losing'), 1), (('losing', evious'), 1), (('previous', 'closebitcoin'), 1), (('closebitcoin', 'world'), 1), (('world', 'biggest'), 1), (('biggest', 'bestknown'), 1), (('bestknown', 'cryptocurrency'), 1), (('cryptocurrency', 'char'), 1)]),</pre>
	<pre>dict_items([(('reuters', 'staffjune'), 1), (('staffjune', 'reuters'), 1), (('reuters', 'bitcoin'), 1), (('staffjune', 'rose'), 1), (('rose', 'gmt'), 1), (('gmt', 'sunday'), 1), (('sunday', 'adding'), 1), (('adding', 'preus'), 1), (('previous', 'closebitcoin'), 1), (('closebitcoin', 'world'), 1), (('world', 'biggest'), 1), (('staffjune', 'bestknown'), 1), (('staffjune', 'reuters'), 1), (('reuters', 'bitcoin'), 1), (('staffjune', 'reuters'), 1), (('reuters', 'bitcoin'), 1), (('staffjune', 'sunday'), 1), (('sunday', 'adding'), 1), (('adding', evious'), 1), (('previous', 'closebitcoin'), 1), (('closebitcoin', 'world'), 1), (('world', 'biggest'), 1), (('biggest', 'bestknown'), 1), (('bestknown', 'cryptocurrency'), 1), (('cryptocurrency', 'char'), 1)]), dict_items([(('bitcoin', 'rose'), 1), (('rose', 'gmt'), 1), (('gmt', 'sunday'), 1), (('sunday', 'adding'), (('adding', 'previous'), 1), (('previous', 'closebitcoin'), 1), (('closebitcoin', 'world'), 1), (('world', 'gest'), 1), (('biggest', 'bestknown'), 1), (('bestknown', 'cryptocurrency'), 1), (('cryptocurrency', 'year')</pre>
56]: 61]:	<pre>1), (('year', 'low'), 1), (('low', 'char'), 1)])]  # Function token_count generates the top 10 words for a given coin  def token_count(tokens, N=3):     """Returns the top N tokens from the frequency count"""     return Counter(tokens).most_common(N)  # Use token_count to get the top 10 words for Bitcoin  btc_wordmap = {}  for btg_token_in_btg_df[!tokens!];</pre>
	<pre>for btc_token in btc_df['tokens']:     top = dict(token_count(btc_token, 10))     for key in top:         if(key in btc_wordmap):             btc_wordmap[key] += top[key]         else:             btc_wordmap[key] = top[key]  btc_word_list = list(btc_wordmap.items()) x = sorted(btc_word_list, key=lambda x:x[1], reverse=True) btc_top_10 = []</pre> <pre>for i in range(0.10):</pre>
[61]:	<pre>for i in range(0,10):     btc_top_10.append(x[i]) btc_top_10  [('bitcoin', 13),     ('salvador', 7),     ('el', 6),     ('reuters', 6),     ('ha', 4),     ('cryptocurrency', 4),     ('us', 4),</pre>
62]:	<pre>('cryptocurrency', 4), ('us', 4), ('world', 3), ('attack', 3), ('week', 3)]  # Use token_count to get the top 10 words for Ethereum eth_wordmap = {} for eth_token in eth_df['tokens']:     top = dict(token_count(eth_token, 10))     for key in top:         if (key in eth_wordmap):             eth_wordmap[key] += top[key]</pre>
	<pre>eth_wordmap[key] += top[key] else:</pre>
62]:	<pre>[('bitcoin', 9),     ('world', 8),     ('cryptocurrency', 7),     ('previous', 7),     ('closebitcoin', 7),     ('ethereum', 6),     ('network', 6),     ('gmt', 6),     ('reuters', 6),     ('adding', 5)]</pre>
	Word Clouds In this section, you will generate word clouds for each coin to summarize the news for each coin  from wordcloud import WordCloud import matplotlib.pyplot as plt plt.style.use('seaborn-whitegrid') import matplotlib as mpl
64]:	<pre>mpl.rcParams['figure.figsize'] = [20.0, 10.0]  # Generate the Bitcoin word cloud text_list = [] for btc_tokens in btc_df['tokens']:     text_list.append(" ".join(btc_tokens)) text = "" for i in text_list:     text += i wc = WordCloud().generate(text)</pre>
64]:	<pre>wc = WordCloud().generate(text) plt.imshow(wc)  </pre> <pre> <pre></pre></pre>
	cryptocess  spressed legal tender  nayib  replace  surged legal tender  replace  replace
	150 150 175 150 175 175 175 176 177 177 178 179 179 179 179 179 179 179 179 179 179
65]:	<pre># Generate the Ethereum word cloud text_list = [] for eth_tokens in eth_df['tokens']:     text_list.append(" ".join(eth_tokens)) text = "" for i in text_list:     text += i wc = WordCloud().generate(text) plt.imshow(wc)</pre>
65]:	<pre><matplotlib.image.axesimage 0x7fb0adc40b80="" at=""></matplotlib.image.axesimage></pre> <pre> control of the co</pre>
	nft succent losing sunday adding reiday charbitcoin sunday adding
	network biggest best known of the state of t
	3. Named Entity Recognition  In this section, you will build a named entity recognition model for both Bitcoin and Ethereum, then visualize the tags using SpaCy.  import spacy from spacy import displacy  # Download the language model for SpaCy
75] <b>:</b>	<pre># Download the language model for SpaCy # !python -m spacy download en_core_web_sm  # Load the spaCy model nlp = spacy.load('en_core_web_sm')</pre> Bitcoin NER
76]: 76]:	<pre># Concatenate all of the Bitcoin text together btc_text = "" for text in btc_df['text']:     btc_text += text btc_text  'El Salvador\'s President Nayib Bukele has made good on his promise to adopt Bitcoin as legal tender. Offici in the Central American country\'s congress voted to accept the cryptocurrency by a majori [+1414 chars]El vador has become the first country in the world to recognize the cryptocurrency bitcoin as legal currency, a</pre>
	vador has become the first country in the world to recognize the cryptocurrency by a majori [+1414 chars]EI vador has become the first country in the world to recognize the cryptocurrency bitcoin as legal currency, a rding to President Nayib Bukele in a tweet on Wednesday. Citizens will be able to [+3840 chars]It\'s all a t clean energy, it seems.\xa0\r\nElon Musk and Tesla caused ripples in the cryptocurrency market in the past w months, first by announcing that Tesla had bought \$1.5 billion worth of bitcoi [+2508 chars]Filed under:\The supply chain attack has reached over a thousand organizations.\r\nIllustration by Alex Castro / The Verg \nThree days after ransomware attackers started the holiday weekend by comprom [+3376 chars]image captionThree over means bitcoin will be accepted everywhere for goods and services, alongside the US dollar\r\nEl Salvador s become the first country in the world to officially classify Bitcoin a [+2233 chars]Several crypto fans to descended on Miami, Florida, last weekend for the largest bitcoin conference in history are now saying theyw ested positive for covid-19. \r\nLarry Cermak, research director [+2389 chars]Hello friends, and welcome to Week in Review!\r\nLast week, I wrote about tech taking on Disney. This week, I'm talking about the search or a new crypto messiah.\r\nIf youre reading this on the Tec [+7741 chars]By Mary-Ann RussonBusiness report BBC News\r\nimage copyrightGetty Images\r\nFormer US President Donald Trump has told Fox Business that he search or the search of the se
	BBC News\r\nimage copyrightGetty Images\r\nFormer US President Donald Trump has told Fox Business that he se Bitcoin as a "scam" affecting the value of the US dolla [+3638 chars]image copyrightGetty Images\r\nThe gar ehind a "colossal" ransomware attack has demanded \$70m (£50.5m) paid in Bitcoin in return for a "universal of yptor" that it says will unlock the files of all [+4140 chars]SAN SALVADOR, June 16 (Reuters) - El Salvador ill not replace the U.S. dollar with bitcoin as the legal tender, Finance Minister Alejandro Zelaya said on nesday, as the Central American nation so [+1266 chars]LONDON, June 23 (Reuters) - El Salvador is determined o push ahead with making bitcoin legal tender, a process that will bring only small risks and prove a "leap ward for humanity", the country\'s [+2764 chars]LONDON, June 23 (Reuters) - El Salvador is determined to push ahead with making bitcoin legal tender, a process that will bring only small risks and prove a leap forward humanity, the countrys Pr [+2648 chars]Representations of the virtual currency Bitcoin stand on a motherbor in this picture illustration taken May 20, 2021. REUTERS/Dado Ruvic/Illustration/File PhotoMicroStrategy Inc. STR.O), a major b [+954 chars]By Reuters Staff\r\nSAN SALVADOR (Reuters) - El Salvador will not replace the
	STR.O), a major b [+954 chars]By Reuters Staff\r\nSAN SALVADOR (Reuters) - El Salvador will not replace the S. dollar with bitcoin as the legal tender, Finance Minister Alejandro Zelaya said on Wednesday, as the Cent American [+1110 chars]Other than Colonial Pipeline, the biggest recent ransomware attack targeted JBS, a magnifier that\'s responsible for up to one quarter of all meat processing in the US. Late last week the compute the numbers behind the headlines.\r\nThis is Equity Monday, our morning coffee chat with you that is all ab 2045 chars]This story originally appeared on MarketBeatWhen meme coins surged earlier this year (remember Do oin mania?) it might\'ve been easy to dump everything you owned - stocks, bonds, gold - for crypto [+6151 ars]Bitcoin surged 8.54 % to \$36,265 on Wednesday, adding \$2,853.31 to its previous close.\r\nBitcoin, the way Reuters Staff\r\nFILE PHOTO: Representations of the Bitcoin cryptocurrency are seen in this picture illustion taken June 7, 2021. REUTERS/Edgar Su/Illustration\r\n(Reuters) - Bitcoin dropped 5.71% [+336 chars]Bitcoin surged 8.57 % [+336 chars]Bitcoin cryptocurrency are seen in this picture illustion taken June 7, 2021. REUTERS/Edgar Su/Illustration\r\n(Reuters) - Bitcoin dropped 5.71% [+336 chars]Bitcoin cryptocurrency are seen in this picture illustion taken June 7, 2021.
	n dropped 5.71% to \$35,210 at 0600 GMT on Saturday, losing \$2,131.11 from its previous close.\r\nBitcoin, the orld\'s biggest and best-known cryptocurrency, is down 45.7% from its 2021 high of \$6 [+215 chars]'  # Run the NER processor on all of the text doc = nlp(btc_text)  # Add a title to the document doc.user_data["title"] = "Bitcoin text analysis"
77]: 78]:	
77]: 78]:	<pre># Render the visualization displacy.render(doc, style='ent')</pre>

