```
import pandas as pd
import numpy as np
import spacy
import datetime as dt
import matplotlib.pyplot as plt
from matplotlib import pyplot as plt
from matplotlib import rcParams
import seaborn as sns
import re
from wordcloud import WordCloud
import itertools
import collections
import nltk
import string
from nltk import FreqDist
from sklearn.feature_extraction.text import CountVectorizer
from nltk.util import ngrams
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import gensim
import multiprocessing
from gensim.models import Word2Vec
from multiprocessing import Process
from gensim.models.phrases import Phrases, ENGLISH_CONNECTOR_WORDS
import sklearn
from sklearn.cluster import KMeans
```

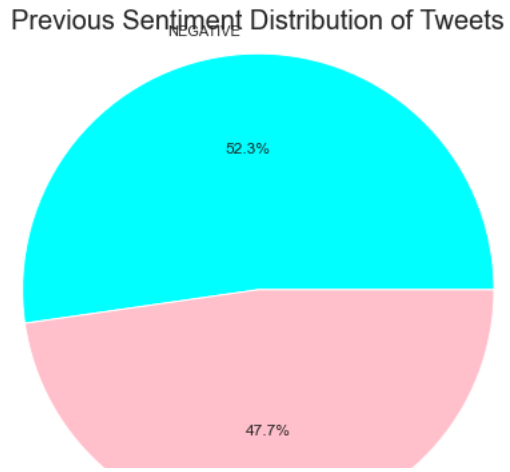
```
cleandf = pd.read_csv('cleandf_bertSentiment.csv')
```

```
cleandf
```

	User	Follows_Count	Language	Date_Tweet	Number_of_Likes	
0	ForestsDAOcom	193	en	2022-09-30 20:15:02+00:00	2	crypto c c exc 1gcx
1	thoughtforfood_	8598	en	2022-09-30 19:07:49+00:00	13	in ha poter ince grow
2	BlociCarbon	143	en	2022-09-30 18:06:44+00:00	1	yo reduc c footprir
3	CHAR_Technology	423	en	2022-09-30 18:02:35+00:00	3	with in up

```
fig = plt.gcf()
fig.set_size_inches(7,7)
colors = ["cyan","pink","yellow"]
cleandf_pie=cleandf["Sentiment"].value_counts().reset_index()
plt.pie(cleandf_pie["Sentiment"],labels=cleandf_pie["index"],radius=2,colors=colors,autopct="%1.1f%%")
plt.axis('equal')
plt.title("Previous Sentiment Distribution of Tweets", fontsize=20)
plt.show()
cleandf_pie
```

```
#plt.savefig("Downloads/sent_dist_words.png")
```




```
#data18 = pd.read_csv('carbonMarket2018_2022.csv')
data18 = pd.read_csv('carbondata_21_221031.csv')
```

```
data18.tail()
```

	User	verified	Date_Created	Follows_Count	Friends_Count	Retweet_Count	Language	Date_Tweet	Number_of_Likes	Sou
355224	SusHealthcare	False	2011-07-05 09:29:14+00:00	7488	4420	4	en	2021-01-01 01:00:22+00:00	5	
355225	LisaKayeCAP	False	2013-08-02 04:16:44+00:00	376	832	1	en	2021-01-01 01:00:03+00:00	1	
355226	thegalonthego	False	2016-04-12 23:51:26+00:00	398	1346	0	en	2021-01-01 00:32:59+00:00	1	T
355227	raularteche	False	2011-11-12 12:30:32+00:00	37	91	2	es	2021-01-01 00:11:37+00:00	3	
355228	raularteche	False	2011-11-12 12:30:32+00:00	37	91	2	es	2021-01-01 00:11:17+00:00	3	

```
len(data18.columns.tolist())
```

```
17
```

```
data18['TweetC'] = data18['Tweet']
data18.head()
```

```
data18.dtypes
```

```
User              object
verified            bool
Date_Created      object
Follows_Count      int64
Friends_Count      int64
Retweet_Count      int64
Language          object
Date_Tweet        object
Number_of_Likes    int64
Source_of_Tweet   object
Tweet_Id           int64
Tweet             object
Hashtags          object
Conversation_Id    int64
In_reply_To       object
Coordinates       object
Place             object
dtype: object
```

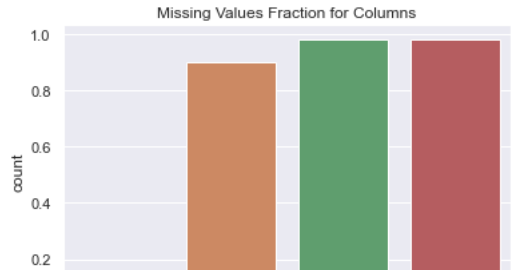
```
data18['Place'].loc[data18['Place'].notnull()]
```

```
40        Place(fullName='Cambridge, England', name='Cam...
426       Place(fullName='Manchester, England', name='Ma...
472       Place(fullName='Sydney, New South Wales', name...
499       Place(fullName='Bracknell, England', name='Bra...
793       Place(fullName='Edinburgh, Scotland', name='Ed...
                              ...
354702    Place(fullName='Western Bay of Plenty District...
354790    Place(fullName='Wezembeek-Oppem, België', name...
354848    Place(fullName='Paynton No. 470, Saskatchewan'...
354917    Place(fullName='Sutton, London', name='Sutton'...
355008    Place(fullName='Oakington, England', name='Oak...
Name: Place, Length: 6739, dtype: object
```

```
# for plotting missing values

def return_missing_values(data_frame):
    missing_values = data_frame.isnull().sum()/len(data_frame)
    missing_values = missing_values[missing_values>0]
    missing_values.sort_values(inplace=True)
    return missing_values

def plot_missing_values(data_frame):
    missing_values = return_missing_values(data_frame)
    missing_values = missing_values.to_frame()
    missing_values.columns = ['count']
    missing_values.index.names = ['Name']
    missing_values['Name'] = missing_values.index
    sns.set(style='darkgrid')
    sns.barplot(x='Name', y='count', data=missing_values)
    plt.xticks(rotation=90)
    plt.title('Missing Values Fraction for Columns')
    plt.show()


#https://github.com/ShilpiParikh/EDA-on-COVID-19-tweets/blob/main/Covid19_tweets_EDA%20.ipynb


return_missing_values(data18)
```

```
Hashtags       0.000028
In_reply_To    0.899307
Coordinates    0.981029
Place          0.981029
dtype: float64
```

```
plot_missing_values(data18)
```




```
# unique values from data
def return_unique_values(data_frame):
    unique_dataframe = pd.DataFrame()
    unique_dataframe['Features'] = data_frame.columns
    uniques = []
    for col in data_frame.columns:
        u = data_frame[col].nunique()
        uniques.append(u)
    unique_dataframe['Uniques'] = uniques
    return unique_dataframe

#https://github.com/ShilpiParikh/EDA-on-COVID-19-tweets/blob/main/Covid19_tweets_EDA%20.ipynb


unidf = return_unique_values(data18)
print(unidf)
```

```
               Features  Uniques
     0              User    66174
     1          verified        2
     2      Date_Created    66168
     3     Follows_Count    16694
     4     Friends_Count     7023
     5     Retweet_Count      281
     6          Language       54
     7        Date_Tweet   348244
     8   Number_of_Likes      586
     9   Source_of_Tweet      460
     10         Tweet_Id   355229
     11            Tweet   346805
     12         Hashtags   185054
     13  Conversation_Id   345559
     14      In_reply_To    15228
     15      Coordinates     1531
     16            Place     1492
```

```
f, ax = plt.subplots(1,1, figsize=(10,5))

sns.barplot(x=unidf['Features'], y=unidf['Uniques'], alpha=0.7)
plt.title('Bar plot for Unique Values in each column')
plt.ylabel('Unique values', fontsize=14)
plt.xlabel('Features', fontsize=14)
plt.xticks(rotation=90)
plt.show()

#https://github.com/ShilpiParikh/EDA-on-COVID-19-tweets/blob/main/Covid19_tweets_EDA%20.ipynb
```


```
places = data18['Place'].loc[data18['Place'].notnull()].tolist()
places[1]
```

```
"Place(fullName='Bracknell, England', name='Bracknell', type='city', country='United Kingdom', countryCode='GB')"
```


```
places = data18['Place'].value_counts(ascending=False).rename_axis('unique_values').to_frame('counts')
places
```

unique_values	counts
Place(fullName='Manhattan, NY', name='Manhattan', type='city', country='United States', countryCode='US')	759
Place(fullName='Glasgow, Scotland', name='Glasgow', type='city', country='United Kingdom', countryCode='GB')	116
Place(fullName='Virginia Water, South East', name='Virginia Water', type='city', country='United Kingdom', countryCode='GB')	95
Place(fullName='Sevenoaks Weald, South East', name='Sevenoaks Weald', type='city', country='United Kingdom', countryCode='GB')	86
Place(fullName='Edinburgh, Scotland', name='Edinburgh', type='city', country='United Kingdom', countryCode='GB')	70
...	...
Place(fullName='Wahoo, NE', name='Wahoo', type='city', country='United States', countryCode='US')	1
Place(fullName='Kota Lama Kanan, Perak', name='Kota Lama Kanan', type='city', country='Malaysia', countryCode='MY')	1
Place(fullName='Kuala Kalumpang, Selangor', name='Kuala Kalumpang', type='city', country='Malaysia', countryCode='MY')	1
Place(fullName='Petaling, Wilayah Persekutuan Kuala Lumpur', name='Petaling', type='city', country='Malaysia', countryCode='MY')	1
Place(fullName='Garston, England', name='Garston', type='city', country='United Kingdom', countryCode='GB')	1

1161 rows × 1 columns

```
data18['Place'].loc[data18['Place'].notnull()]
```

```
472       Place(fullName='Sydney, New South Wales', name...
499       Place(fullName='Bracknell, England', name='Bra...
793       Place(fullName='Edinburgh, Scotland', name='Ed...
895       Place(fullName='Crystal Mini Market', name='Cr...
1009      Place(fullName='Worcester, England', name='Wor...
                              ...
354594    Place(fullName='Garston, England', name='Garst...
354702    Place(fullName='Western Bay of Plenty District...
354790    Place(fullName='Wezembeek-Oppem, België', name...
354848    Place(fullName='Paynton No. 470, Saskatchewan'...
355008    Place(fullName='Oakington, England', name='Oak...
Name: Place, Length: 4779, dtype: object
```

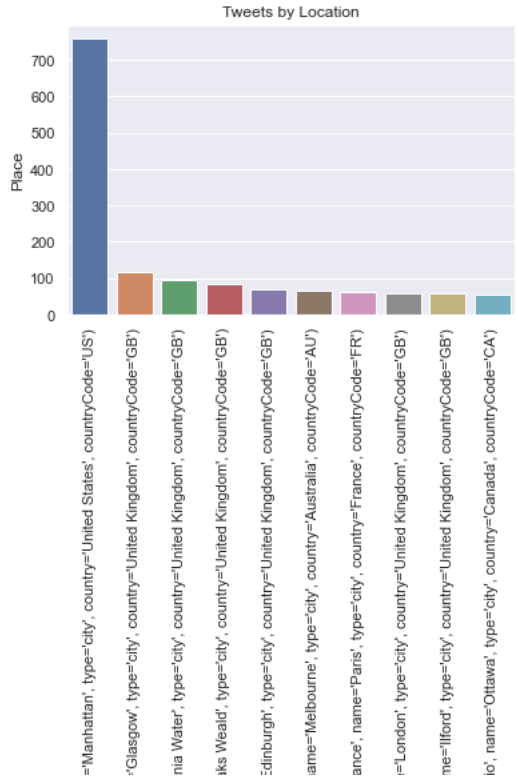
```
data18['Place'].iloc[793]
```

```
nan
```

```
s.split(':')[0]
```

```
sns.barplot(x= data18.Place.value_counts()[:10].index,y=data18.Place.value_counts()[:10]).set(title='Tweets by Location')
plt.xticks(rotation=90)
```

```
(array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9]),
 [Text(0, 0, "Place(fullName='Manhattan, NY', name='Manhattan', type='city', country='United States', countryCode='US')"),
  Text(1, 0, "Place(fullName='Glasgow, Scotland', name='Glasgow', type='city', country='United Kingdom', countryCode='GB')"),
  Text(2, 0, "Place(fullName='Virginia Water, South East', name='Virginia Water', type='city', country='United Kingdom',
countryCode='GB')"),
  Text(3, 0, "Place(fullName='Sevenoaks Weald, South East', name='Sevenoaks Weald', type='city', country='United Kingdom',
countryCode='GB')"),
  Text(4, 0, "Place(fullName='Edinburgh, Scotland', name='Edinburgh', type='city', country='United Kingdom', countryCode='GB')"),
  Text(5, 0, "Place(fullName='Melbourne, Victoria', name='Melbourne', type='city', country='Australia', countryCode='AU')"),
  Text(6, 0, "Place(fullName='Paris, France', name='Paris', type='city', country='France', countryCode='FR')"),
  Text(7, 0, "Place(fullName='London, England', name='London', type='city', country='United Kingdom', countryCode='GB')"),
  Text(8, 0, "Place(fullName='Ilford, London', name='Ilford', type='city', country='United Kingdom', countryCode='GB')"),
  Text(9, 0, "Place(fullName='Ottawa, Ontario', name='Ottawa', type='city', country='Canada', countryCode='CA')")])
```



Pre-processsing

```
data18['Tweet'].iloc[3]
```

```
'Because your children deserve a ‘better’ world..\n\n#skypapers #cop27 #netzero #ClimateEmergency #ClimateScam #climatesame
https://t.co/X4Q5EBtb74'
```

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer

lemma = WordNetLemmatizer()
stop_words = set(stopwords.words('english'))


nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\dantr\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
True
```

```
# the function to clean the tweet and tokenize them
def clean_tweet(tweet):
    if type(tweet) == float:
            return ""

    # turn text into lower
    test = tweet.lower()
    # remove all mentions and hashtags
    test = re.sub("@[A-Za-z0-9_]+","", test)
```

```
        test = re.sub("#[A-Za-z0-9_]+","", test)
        #remove links
        test = re.sub(r"http\S+", "", test)
        test = re.sub(r"www.\S+", "", test)
        #remove punctuation
        test = re.sub('[()!?]', ' ', test)
        test = re.sub('\[.*?\]',' ', test)
        #remove non alphabetical characters
        test = re.sub("[^a-z0-9]"," ", test)
        #remove extra spaces
        test = re.sub(' +', ' ', test)
        # remove single letter words
        test = ' '.join( [w for w in test.split() if len(w)>1] )

        test = ' '.join( [lemma.lemmatize(x) for x in nltk.wordpunct_tokenize(test) if x not in stop_words])
        test =[lemma.lemmatize(x, nltk.corpus.reader.wordnet.VERB) for x in nltk.wordpunct_tokenize(test) if x not in stop_words]

        return test

    # define a function to clean the tweet.
    def clean_tweet2(tweet):

        tweet = tweet.lower()
        tweet = re.sub('https?:\/\/[a-zA-Z0-9@:%._\/+~#=?&;-]*', ' ', tweet)
        tweet = re.sub('\$[a-zA-Z0-9]*', ' ', tweet)
        tweet = re.sub('\@[a-zA-Z0-9]*', ' ', tweet)
        tweet = re.sub('[^a-zA-Z\']', ' ', tweet)
        tweet = ' '.join( [w for w in tweet.split() if len(w)>1] )

        tweet=' '.join([lemma.lemmatize(x) for x in nltk.wordpunct_tokenize(tweet) if x not in stop_words])
        tweet=[lemma.lemmatize(x,nltk.corpus.reader.wordnet.VERB) for x in nltk.wordpunct_tokenize(tweet) if x not in stop_words]
        return tweet


    clean_tweet(data18['Tweet'].iloc[3])

         ['child', 'deserve', 'better', 'world']


    # clean the tweets and create two columns: tokenized tweet and whole tweet
    data18["clean_tweet"]=data18["Tweet"].apply(lambda x:clean_tweet(x))
    data18["cleaned_tweet"]=data18["clean_tweet"].apply(lambda x:' '.join(x))


    # we choose tweets in English and with at least 1 like
    data18 = data18[data18['Language'] == 'en']
    data18 = data18[data18['Number_of_Likes'] >= 1]


    data18['clean_tweet'].iloc[4]

         ['measure',
          'emission',
          'first',
          'step',
          'address',
          'impact',
          'account',
          'service',
          'offer',
          'holistic',
          'approach',
          'address',
          'scope',
          'accordance',
          'protocol',
          'corporate',
          'account',
          'report',
          'standard']


    data18['cleaned_tweet'].iloc[4]

         'measure emission first step address impact account service offer holistic approach address scope accordance protocol corporate account
         report standard'


    data18.shape

         (225097, 20)
```

```
tweets = data18['clean_tweet']
tweets[:10]
```

```
0     [nigeria, pioneer, billion, dollar, worth, vol...
2                             [rare, earth, conference]
16    [patriot, hydrogen, launch, malaysia, first, m...
26          [sgp, patron, help, shape, print, industry]
28    [measure, emission, first, step, address, impa...
29    [article, actually, say, country, 50, billion,...
33    [look, forward, see, system, utilize, east, co...
34    [week, underway, egypt, start, learn, global, ...
35    [mr, goodwin, must, miss, un, wef, globalists,...
36    [build, sustainable, business, strategy, first...
Name: clean_tweet, dtype: object
```

Hashtags

```
# define a function to clean the Hashtags.
def clean_hashtags(hashtags):
    '''
    hashtags: String
              Input Data
    hashtags: String
              Output Data

    func: Convert hashtags to lower case
          Replace ticker symbols with space. The ticker symbols are any stock symbol that starts with $.
          Replace everything not a letter or apostrophe with space
          Removes any spaces or specified characters at the start and end of hashtags.

    '''
    if hashtags:
        hashtags = hashtags.lower()
        hashtags = re.sub('\$[a-zA-Z0-9]*', ' ', hashtags)
        hashtags = re.sub('[^a-zA-Z]', ' ', hashtags)
        hashtags=hashtags.strip()
    return hashtags
```

```
# clean the hashtags
data18["Hashtags"]=data18["Hashtags"].astype(str)
data18["Hashtags"]=data18["Hashtags"].apply(lambda x:clean_hashtags(x))
```

```
data18.head()
```

	User	verified	Date_Created	Follows_Count	Friends_Count	Retweet_Count	Language	Date_Tweet	Number_of_Likes	Source
0	CarbonCredits	False	2017-06-21 17:44:31+00:00	6799	283	0	en	2022-10-31 23:36:00+00:00	5	Twitt
2	M_Costelloe	False	2012-05-03 02:19:44+00:00	604	819	0	en	2022-10-31 23:29:50+00:00	3	Twitte
16	PatriotHydrogen	False	2022-08-15 11:29:38+00:00	7	1	2	en	2022-10-31 23:13:50+00:00	1	Twitt
26	SGPPartnership	False	2012-09-05 15:14:54+00:00	1060	897	1	en	2022-10-31 23:05:05+00:00	1	Sem
28	PeriCarbon	False	2022-08-16 23:58:10+00:00	3	18	0	en	2022-10-31 23:02:17+00:00	2	Twitt

DateColumns: + month, year columns

```
data18['date'] = pd.to_datetime(data18['Date_Tweet'], format='%Y-%m-%d')
data18['month'] = data18['date'].dt.month
data18['year'] = data18['date'].dt.year
```

```
data18.tail()
```

	User	verified	Date_Created	Follows_Count	Friends_Count	Retweet_Count	Language	Date_Tweet	Number_of_Likes	S
355221	CryptoRiskGroup	False	2020-11-08 04:33:26+00:00	1205	4952	0	en	2021-01-01 02:55:17+00:00	1	
355223	MissionShunya	False	2019-03-31 02:40:21+00:00	318	215	2	en	2021-01-01 01:18:18+00:00	2	
355224	SusHealthcare	False	2011-07-05 09:29:14+00:00	7488	4420	4	en	2021-01-01 01:00:22+00:00	5	
355225	LisaKayeCAP	False	2013-08-02 04:16:44+00:00	376	832	1	en	2021-01-01 01:00:03+00:00	1	
355226	thegalonthego	False	2016-04-12 23:51:26+00:00	398	1346	0	en	2021-01-01 00:32:59+00:00	1	

5 rows × 23 columns

```
list(data18['cleaned_tweet'][(data18['year']==2022)&(data18['month']==9)][:10])
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
```

Turn tweets into embedding vectors

```
!pip install -U gensim
```

```
Collecting gensim
  Downloading gensim-4.2.0-cp39-cp39-win_amd64.whl (23.9 MB)
     ---------------------------------------- 23.9/23.9 MB 11.7 MB/s eta 0:00:00
Collecting Cython==0.29.28
  Downloading Cython-0.29.28-py2.py3-none-any.whl (983 kB)
     ---------------------------------------- 983.8/983.8 kB 15.7 MB/s eta 0:00:00
Requirement already satisfied: scipy>=0.18.1 in c:\users\dantr\anaconda3\lib\site-packages (from gensim) (1.7.1)
Requirement already satisfied: smart-open>=1.8.1 in c:\users\dantr\anaconda3\lib\site-packages (from gensim) (5.2.1)
Requirement already satisfied: numpy>=1.17.0 in c:\users\dantr\anaconda3\lib\site-packages (from gensim) (1.20.3)
Installing collected packages: Cython, gensim
  Attempting uninstall: Cython
    Found existing installation: Cython 0.29.24
    Uninstalling Cython-0.29.24:
      Successfully uninstalled Cython-0.29.24
Successfully installed Cython-0.29.28 gensim-4.2.0
```

```
   3457                 return self._getitem_multilevel(key)
```

```
#Converting the "clean_tweet" column in the format supported by embeddings.
sent = [row for row in data18["clean_tweet"]]
#use Gensim Phrases package to automatically detect common phrases (bigrams) from a list of sentences.
phrases = Phrases(sent, min_count=1, progress_per=50000)
bigram = gensim.models.phrases.Phraser(phrases)
sentences = bigram[sent]
sentences[1]

# https://www.kaggle.com/pierremegret/gensim-word2vec-tutorial
```

```
['rare_earth', 'conference']
```

```
len(sentences)
```

```
225097
```

Word2Vec model

```
#Initializing the word2vec model
w2v_model = Word2Vec(min_count=4,
                     window=5,
                     vector_size =300,
                     sample=1e-5,
                     alpha=0.03,
                     min_alpha=0.0007,
                     negative=20,
                     seed= 42,
                     workers=multiprocessing.cpu_count()-1)


#building vocab of the word2vec model from the custom data
w2v_model.build_vocab(sentences, progress_per=50000)

# https://towardsdatascience.com/unsupervised-sentiment-analysis-a38bf1906483


#training the word2vec model
w2v_model.train(sentences, total_examples=w2v_model.corpus_count, epochs=60, report_delay=1)
```

```
(51360907, 177346140)
```

```
w2v_model.wv.most_similar(positive =["carbon"])
```

```
[('emission', 0.5598576664924622),
 ('farm_productivity', 0.4787425696849823),
 ('greenhouse_gas', 0.4655562937259674),
 ('offset_residual', 0.45276179909706116),
 ('carbon_footprint', 0.44694510102272034),
 ('provider_carbonneutral', 0.4391014277935028),
 ('remove_ghgs', 0.43749967217445374),
```

```
        ('reduce', 0.43422582745552063),
        ('switch_vertua', 0.43141525983810425),
        ('offset_unavoidable', 0.42592522501945496)]
```

```
#saving the word2vec model
w2v_model.save("word2vec.model")
```

```
#Loading the word2vec model
word_vectors = Word2Vec.load("word2vec.model").wv
```

Clustering

```
#Feeding the embeddings to a KMeans model to cluster words into positive, negative, and neutral clusters
model = KMeans(n_clusters=3, max_iter=1000, random_state=42, n_init=50).fit(X=word_vectors.vectors.astype('double'))
```

```
# check what we have in each cluster to label the clusters
word_vectors.similar_by_vector(model.cluster_centers_[0], topn=200, restrict_vocab=None)
```

```
     [('labour_libdems', 0.7927475571632385),
      ('throat', 0.7844064235687256),
      ('kowtow', 0.782042920589447),
      ('anti_fracking', 0.7816827297210693),
      ('cultist', 0.7810706496238708),
      ('dead_water', 0.7708773016929626),
      ('implode', 0.7647237181663513),
      ('pretend_something', 0.7628898024559021),
      ('90min', 0.7591031193733215),
      ('poor_vulnerable', 0.7581508159637451),
      ('total_b', 0.7574341893196106),
      ('unworkable', 0.7560237050056458),
      ('inept', 0.7555508017539978),
      ('wef_agenda', 0.7541081309318542),
      ('massacre', 0.7524727582931519),
      ('foretaste', 0.7514427304267883),
      ('ballot_box', 0.7469227313995361),
      ('religious', 0.7467252612113953),
      ('underlie_cause', 0.7454192042350769),
      ('fcuking', 0.7450405955314636),
      ('crisis_actor', 0.7447817921638489),
      ('wet_dream', 0.7442888021469116),
      ('torture', 0.7437711954116821),
      ('total_disaster', 0.7436293363571167),
      ('totally_wrong', 0.7430939674377441),
      ('globalists_politician', 0.742903470993042),
      ('fetish', 0.7427712678909302),
      ('ridiculous_agenda', 0.7413889765739441),
      ('bitter', 0.7397308349609375),
      ('remoaner', 0.7397273778915405),
      ('certain_aspect', 0.7386876344680786),
      ('embarrassingly', 0.7382276654243469),
      ('suella', 0.7364029288291931),
      ('bilge', 0.7344363331794739),
      ('beyond_belief', 0.7340344190597534),
      ('wokery', 0.7328445911407471),
      ('tiresome', 0.731285035610199),
      ('bankrupt_nation', 0.7307436466217041),
      ('grifter', 0.730520486831665),
      ('technically_illiterate', 0.7303758263587952),
      ('obey', 0.7301363348960876),
      ('sucker', 0.7297070026397705),
      ('derange', 0.7291355729103088),
      ('headlong', 0.7289630770683289),
      ('technocrat', 0.7284268140792847),
      ('load_nonsense', 0.7273764610290527),
      ('politician_party', 0.726965069770813),
      ('oust', 0.7267110347747803),
      ('open_mouth', 0.7264942526817322),
      ('ww2', 0.7261033654212952),
      ('declare_war', 0.7260568141937256),
      ('despise', 0.7259213328361511),
      ('modernity', 0.725555419921875),
      ('incapable', 0.7250706553459167),
      ('myopic', 0.7245981693267822),
      ('bang_money', 0.7245380282402039),
      ('zac', 0.7245115041732788),
      ('disrespectful', 0.724464476108551),
```

```
# Labelling the clusters based on the type of words they carry
positive_cluster_center = model.cluster_centers_[1]
negative_cluster_center = model.cluster_centers_[0]
neutral_cluster_center= model.cluster_centers_[2]


#Creating a DataFrame of words with their embeddings and cluster values
words = pd.DataFrame(word_vectors.index_to_key)
words.columns = ['words']
words['vectors'] = words.words.apply(lambda x: word_vectors[f'{x}'])
words['cluster'] = words.vectors.apply(lambda x: model.predict([np.array(x)]))
words.cluster = words.cluster.apply(lambda x: x[0])

# https://towardsdatascience.com/unsupervised-sentiment-analysis-a38bf1906483


words
```

	words	vectors	cluster
0	amp	[0.07874743, 0.108380824, -0.0063456777, 0.095...	1
1	net_zero	[0.008779592, -0.32702243, 0.18982653, -0.3640...	1
2	emission	[-0.03364873, 0.007788754, -0.056246445, -0.37...	1
3	new	[0.22049752, 0.033542696, -0.17602187, -0.3385...	1
4	need	[-0.10592899, 0.09529903, 0.080500744, -0.0258...	1
...	...	...	...
34572	safer_faster	[0.080763854, 0.14341721, -0.47658885, -0.2650...	1
34573	immediacy	[0.038642377, -0.025591485, -0.21957204, -0.09...	1
34574	multibillion_pound	[-0.1570436, 0.26472902, -0.24957645, -0.18978...	1
34575	hideous	[0.12516734, 0.014523674, -0.5922054, -0.31292...	0
34576	four_woman	[-0.091371, 0.026921628, 0.07694977, 0.0796975...	1

34577 rows × 3 columns

```
#Assigning 1 to positive values, 0 to neutral and -1 for negative values
words['cluster_value'] = [1 if i==1 else 0 if i==2 else -1 for i in words.cluster]
words['closeness_score'] = words.apply(lambda x: 1/(model.transform([x.vectors]).min()), axis=1)


with pd.option_context('display.max_rows', None,):
    print(words[words["cluster_value"]==-1][:300].sort_values("closeness_score"))
```

```
                words                                            vectors  \
1221              ppl  [0.650828, 1.6795492, 0.5562937, 0.10844719, 0...
868             boris  [-0.33551437, 1.7563875, -1.0237154, -0.859760...
1078       wind_solar  [-0.2251152, 1.1444101, -1.4541452, -0.6976538...
1220         morrison  [-0.05148441, 0.33381587, -1.1665766, 0.476167...
1223          cheaper  [-0.51113576, 0.66964525, -0.6279917, -0.42391...
1072              ban  [0.5425604, 1.3014003, 0.054051246, -0.3866854...
1161     electric_car  [0.41359308, 0.3259452, -0.24115357, -0.458324...
1070           winter  [0.7511588, 0.6455518, -1.5948646, -0.05675960...
846         scientist  [0.7114922, 0.28360194, -0.6811638, -0.8574005...
636              tree  [-1.1427224, -0.2907203, -0.48203737, -1.40336...
1185        emergency  [0.65298814, 0.0027287947, -0.055670846, -1.26...
1196         pandemic  [-0.27241492, 0.63738275, -0.733505, -0.812545...
1198       short_term  [0.32726568, 0.4606281, -0.67390007, -0.465343...
786              tory  [0.47428632, 1.0799248, -0.6233574, -0.2511400...
1122           medium  [-0.22258338, 0.28112075, 0.3143846, -0.404343...
854       energy_bill  [0.08509497, 0.7637799, -0.09317464, -0.825925...
994               fly  [-0.74828815, -0.059971966, -0.7648884, -0.231...
1118   prime_minister  [-0.12010259, 0.07675693, 0.19677708, -0.35321...
726          trillion  [0.05023582, 0.13195121, 0.10547822, -0.980591...
1011         domestic  [0.8172309, 0.8306684, -0.9469607, -0.4681458,...
1230             poor  [0.8362255, 0.8339503, -0.43704444, 0.04146762...
1166              bad  [0.029570986, 0.001914841, -0.56363726, 1.0340...
878   energy_security  [0.77840704, 0.4426054, 0.55634964, -0.500262,...
1173          destroy  [-0.066926554, 1.5150987, -1.0248624, -0.03939...
1065             west  [0.011304957, 0.7725545, -1.3494736, -0.186123...
1071        expensive  [-0.12366416, 0.9040988, -0.71317124, -0.45439...
388              wind  [0.07928803, 0.621604, -0.5495421, 0.058392983...
863              burn  [-0.05250529, 1.3068867, -0.8362058, -0.443917...
1190           export  [0.4182591, 0.9580796, -0.8794102, -0.9779354,...
921            profit  [-0.53269714, 1.1597856, -0.4039967, -0.262302...
734              warm  [0.46164885, 0.08521241, -0.31152368, -0.31974...
```

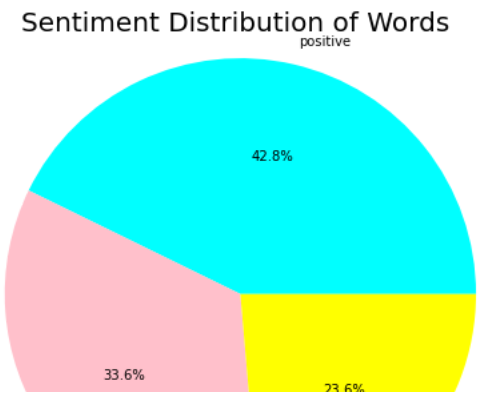
```
999            higher  [0.3986962, 1.0187455, 0.82439286, -0.3402279,...
1028           family  [1.0056014, 1.0647142, -0.87678766, 0.20133601...
780             party  [-0.12115093, 0.75933, -0.14212097, -0.3880774...
717            choice  [0.139697, 0.7030145, 0.9591825, -0.16475895, ...
954              game  [-0.5941067, 0.709968, -0.13659346, 0.20821604...
1169           unless  [-0.72307754, 1.6257223, -0.004336403, -0.4007...
1002        pollution  [0.31224054, 0.5404562, -0.20932539, -0.749528...
928           germany  [0.5684834, 0.34232843, -0.09506084, -0.503937...
779               law  [0.40321925, -0.2523018, 0.81824064, -0.590854...
696                mp  [0.46677804, 0.45457774, 0.40468666, -1.044079...
822           replace  [-0.20928122, 0.8725276, -1.0370166, -0.328547...
818            social  [0.040621083, -0.0670769, -0.32083952, -0.5514...
970             delay  [0.14103231, -0.31716797, 0.36709097, 0.342313...
1148          history  [-0.29704973, -0.07740079, -0.31640613, -0.933...
1241            covid  [-0.5397626, 0.91578174, -0.046005554, -0.1007...
1213        influence  [0.6956324, -0.023763964, -0.35476884, -0.4903...
1251          quickly  [0.614993, 0.4875474, -0.6447686, -0.045345165...
626           oil_gas  [0.26996064, 0.40433812, 0.08269349, -0.434096...
976       global_warm  [0.22388686, -0.7942691, 0.25901756, -0.040212...
982          nonsense  [0.32255217, 1.6221486, -1.1998236, -0.1884068...
767              hand  [-0.45379525, 0.5189935, -0.41419458, 0.031689...
519               tax  [-1.109135, 1.6048301, -0.29975972, -0.5837303...
825        australian  [0.43253633, 0.10921355, 0.23534356, 0.0707043...
601            travel  [-0.90709895, -0.1669, -0.96991354, -0.1734176...
916              rule  [0.008715879, 0.90033776, 0.4638759, -0.307851...
865             worth  [0.22661357, -0.46375704, -0.2959869, -0.70416...
```

```
positive = ['good','better','clean','fantastic','right',"hope", "improve","save", "innovation", "delight", "great"]
neutral = ['nuclear','india','australia','play','data','scotland','canada','job',"race","happens","grocery","person",
          'heat','house','may',"national","state"]
negative= ['risk','waste','carbon_footprint']
for i in positive:
    words.loc[words["words"]==i,"cluster_value"]=1

for i in neutral:
    words.loc[words["words"]==i,"cluster_value"]=0

for i in negative:
    words.loc[words["words"]==i,"cluster_value"]=-1


words[words["words"]=="dangerous"]
```

	words	vectors	cluster	cluster_value	closeness_score
1888	dangerous	[-0.1639378, -0.061654735, -0.93385375, 0.3158...	0	-1	0.09594

Sentiment analysis of words

```
# Plotting pie chart of Sentiment Distribution of words
emotion = {0: "neutral",
           1: "positive",
          -1: "negative"}

words["sentiments"]=words["cluster_value"].map(emotion)


fig = plt.gcf()
fig.set_size_inches(7,7)
colors = ["cyan","pink","yellow"]
df_pie=words["sentiments"].value_counts().reset_index()
plt.pie(df_pie["sentiments"],labels=df_pie["index"],radius=2,colors=colors,autopct="%1.1f%%")
plt.axis('equal')
plt.title("Sentiment Distribution of Words ", fontsize=20)
plt.show()
df_pie

plt.savefig("Downloads/sent_dist_words.png")
```



Out of 19911 unique words and bigram from the dataset:

11621 (33.61%) are Neutral sentiments 14786 (42.76%) are Positive sentiments 8170 (23.63%) are Negative sentiments

It shows that the Neutral and Positive words have larger domination in the dataset

```
<Figure size 432x288 with 0 Axes>
```

Custom sentiment analysis of tweets

```
# creating a dictionary of the word and its cluster value
words_dict = dict(zip(words.words, words.cluster_value))


# define a function to get the sentiment for the entire tweet
def get_sentiments(x,words_dict):
    '''
    x:         List
               Input data: Row of a DataFrame
    sent_dict: Dictionary
               Input: Dictionary of Words: Sentiments
    sentiment: String
               Output: Sentiment of the whole sentence

    Function: Getting sentiments of the entire sentence by averaging out the sentiments of individual words
    '''
    total=0
    count=0
    test=x["clean_tweet"]
    #print(test)
    for t in test:
        if words_dict.get(t):
            total+=int(words_dict.get(t))
            #print('adding', int(words_dict.get(t)))
        count+=1
    if count == 0:
        sentiment = 'no data'
    else:
        avg=total/count
        sentiment=-1 if avg<-0.15 else 1 if avg >0.15 else 0
    return sentiment


#x = data18.iloc[20]

total=0
count=0
#test=data18.iloc[2431]["clean_tweet"]
test=data18.iloc[0]["clean_tweet"]
print(test)
for t in test:
    if words_dict.get(t):
        total+=int(words_dict.get(t))
        print('adding', int(words_dict.get(t)))
    count+=1
if count == 0:
    print('ZERO ERROR')
    sentiment = 'no data'
else:
    avg=total/count
    sentiment=-1 if avg<-0.15 else 1 if avg >0.15 else 0
```

```
print('total:', total)
print('count:', count)
print('average:', avg)
print('sentiment:', sentiment)
```

```
['nigeria', 'pioneer', 'billion', 'dollar', 'worth', 'voluntary', 'carbon', 'market', 'africa', 'africa', 'carbon', 'market', 'initiativ
adding 1
adding 1
adding 1
adding -1
adding 1
adding 1
adding 1
adding 1
adding 1
adding 1
adding 1
adding 1
adding 1
adding 1
adding 1
adding 1
adding 1
adding 1
adding 1
adding 1
adding 1
adding 1
total: 20
count: 24
average: 0.8333333333333334
sentiment: 1
```

```
for i in range(len(data18)):
    x = data18.iloc[i]
    data18['sentiment'][i] = get_sentiments(x, words_dict)
```

```
C:\Users\dantr\AppData\Local\Temp/ipykernel_22536/2313760499.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  data18['sentiment'][i] = get_sentiments(x, words_dict)
C:\Users\dantr\anaconda3\lib\site-packages\pandas\core\indexing.py:1732: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  self._setitem_single_block(indexer, value, name)
```

```
data18.head()
```

User	verified	Date_Created	Follows_Count	Friends_Count	Retweet_Count	Language	Date_Tweet	Number_of_Likes	Source

```
counts = 0
for i in range(len(data18)):
    test = type(get_sentiments(data18.iloc[i], words_dict))
    if test is str:
        counts+=1
print(counts)
    #print('sentiment for', i, ':', get_sentiments(data18.iloc[i], words_dict))
```

```
180
```

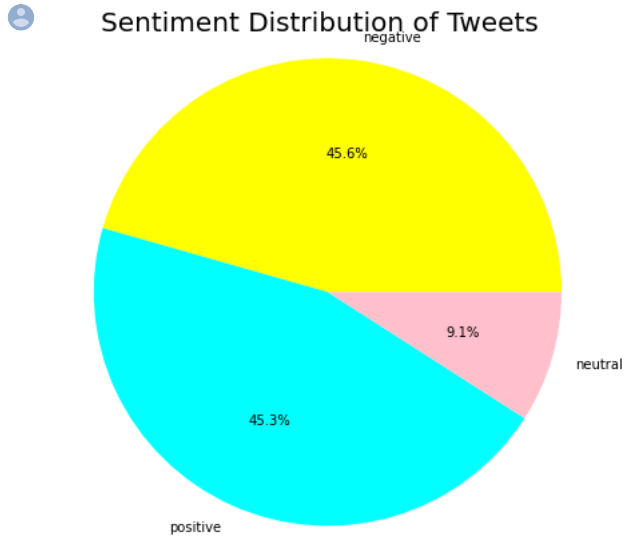
2022-08-15 2022-10-31

```
# checking the value counts of each sentiment
data18["sentiment"].value_counts()
```

```
-1         102577
1          102013
0           20412
no data        95
Name: sentiment, dtype: int64
```

```
# Plotting pie chart of Sentiment Distribution of tweets
emotion = {0: "neutral",
           1: "positive",
          -1: "negative"}

data18["sentiments_val"]=data18["sentiment"].map(emotion)
data_pie=data18["sentiments_val"].value_counts().reset_index()
fig = plt.gcf()
fig.set_size_inches(7,7)
colors = ["yellow","cyan","pink"]
plt.pie(data_pie["sentiments_val"],labels=data_pie["index"],radius=2,autopct="%1.1f%%", colors=colors)
plt.axis('equal')
plt.title("Sentiment Distribution of Tweets ", fontsize=20)
#plt.savefig("images/Sentiment_Distribution.png")
plt.show()
data_pie
```

```
plt.savefig("Downloads/sent_dist_tweets.png")
```




```
<Figure size 432x288 with 0 Axes>
```

Out of 77467 tweets from the dataset:

102577(45.6%) are Negative sentiments 102577(9.1%) are Neutral sentiments 20412(45.3%) are Positive sentiments

```
#data18.to_csv('carbondata_labeled_custom1.csv', index=False)
```

```
data_negative = data18[data18["sentiment"]==-1]
```

```
# checking the cause of negative tweets in 2019
list(data18['cleaned_tweet'][(data18['year']==2021)&(data18['month'].isin([10,11]))])
```

```
['gtr transition operation seek develop project drill project carbon neutral secure full release gtr ax',
 'proud part generation',
 'financial sector make major climate pledge implement target bank take sectoral approach focus explain',
 'contribution pembina institute support electricity grid across canada progressive business coalition strategy cap emission much donate today',
 'oh people like desperate wake pursue fantasy prepare sacrifice disable people elderly',
 'new study mitigation pathway question overshoot',
 'force arbitrary arrest even murder fact continue soar since first incentivized voluntary 1997 abuse rural community',
 'amp concern india underdevelop world develop nation cannot look lens emerge mostly occidental construct global north',
 'great see student work support local business carbon reduction journey',
 'negotiation make clear play role achieve global climate objective need meet robust quantification methodology offer carbon credit st',
 'combat achieve need many target shoot goal remain misunderstand underinvested opportunity energy transition thrill announce investment come',
 'ux future front month contract dip 10c today u 47 15 160 decade low bottom 18 lb end november 2016 celebrate year bull market anniversary',
 'busy day london discuss round inside corridor power westminster give evidence committee single use plastic opportunity greater amp suez uk ciwm',
 'blast catch old friend discus takeaway outcome drive global policy amp pivotal decade climate action listen conversation podcast',
 'money say u 130 trillion commitment make glasgow financial alliance net zero obstacle finance sector must overcome path',
 'spoiler',
 'busy day london discuss round give evidence committee single use plastic opportunity greater amp',
 'excite creative tension two different company two different solution tackle problem mean race say',
 'key find report effect climate change reality pressure set target intensify',
 'stock exchange convener capital service objective think greater objective generation transition net zero julia hoggett ceo london stock exchange lse',
 'half world asset worthless 2036 transition accord new study recent article',
 'miss director innovation michelle xuereb speak topic include thursday panel moderate register',
 'consider allow natural gas energy project label investment move could help shift via',
 'climate council ceo say business lead climate action require big mind shift lion share action take decade',
 'join bobby tudor industry expert discus houston leverage energy leadership accelerate solution maintain global competitiveness world pursue target register',
 'canrea commit work collaboration stakeholder ensure canada implement lowest cost reliable sustainable pathway',
 'producer need double meet zero 2050',
 'india move towards achieve commitment business aid realise mission establish set green pledge write',
 'pledge action next cop26 annual summit flurry new corporate commitment glasgow exception key topic transition commodity drive coal amp clean power',
 'congratulation team successful list london stock exchange aim market prouder gelion play part power transition renewable energy',
 'commercial builder chandos construction announce commitment net zero 2040 tim coldwell president say achieve goal possible without long term partnership commitment',
 'glad part project sparkchange physical carbon etc withhold million tonne co2 permit first four week trade',
 'shameela ebrahim johannesburg stock exchange answer question',
 'day fund bring together potential partner deliver ev concierge hub next night round table drive innovation decarbonisation',
 'work denier support elvaston castle garden trust trust people admit derbyshire cc pls cancel nt sub let know',
 'india commit achieve target 2070 ample time institutionalise green policy pathway use top approach highlight',
 'fund join drive new 3m motor vehicle project bridge skill gap low carbon vehicle mechanic support uk target read',
 'really inspire hear david attenborough final time act business could part answer',
 'panxchange build robust market carbon removal credit derive crop land join webinar learn credit generation',
 'green pivot talk thing renewable amp lot rainham construction amp engineer college',
 'come join u upcoming webinar next tuesday sign require guest speaker time tiffany vas energy amp industry researcher talk industry',
 'call anyone consider career help shape future energy help develop skill study amp practical work',
 'look forward welcome salamanca next year lead way improve air quality ship',
 'worker impact uk transition economy look disconnect fear likely reality find',
 'green queen boutique name people trust feel home bring forward new staple italy way cbd product',
 'support energy worker transition important work glad champion',
 'push forward bc canada thank message',
 'congratulation use tourism leadership amp recovery fund facilitate safe isle skye',
```

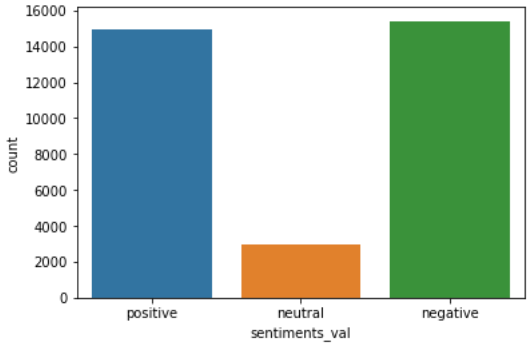
```
# some positive tweets
list(data18[data18["sentiment"]==1]["cleaned_tweet"][300:330])
```

```
['net zero 70k 2030 audacious level democratic heist require much advance plan',
 'important metal run short',
 'vote 2022 showcased arable farmer journey inspire u sequester huge amount carbon amp plan capitalise natural capital gain',
 'icymi koda praha join hyundai holtec international support construction smr 160 small modular reactor batch unit',
 'excellent idea get planet 2050 great use game play',
 'icymi lab get 150m upgrade infrastructure boost nuclear research amp development key component support grow u nuclear industry achieve future',
 'new climate news game changer idea water sustainability centre stage ahead major water conference',
 'icymi announce partnership support goal mover small modular reactor deployment',
 'icymi minister wilkinson position sector provider reliable affordable non emit global stage',
 'icymi announce wecan win edge cooperation advance partnership help country meet amp climate goal',
 'one thing story set clear upper bind specify minimum proportion question 1c nation consultation due tonight folk hope do',
 'banker bring message africa risk sound place wail bagpipe egyptian souq via key focus critical task finance adaptation amp climate resilience',
 'watch interview great insight life 19 drive view nuclear lie ahead',
 'well worth read climate scientist view effort combat weak even psychological sociological aspect need pay attention strive',
 'make today day commit good amp buy amp approximately 20k panel save provide better bank rate take',
 'talk plant would harm good people business daft enough buy seem wrong party',
```

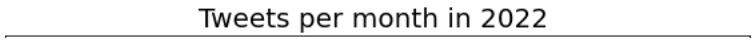
```
 'pal tap acclaim scientist reduce carbon emission',
 'water reduce emission amp make resilient',
 'icymi bill gate terrapower warren buffet pacificorp announce look deploy 345mwe natrium small modular reactor addition first unit
build',
 'guide fix market make house amp',
 'everything learn design wrong need get right take cheer',
 'icymi world largest conference last week washington pledge add 24 gigawatts new nuclear 2050 24',
 'icymi invest nearly billion build new reactor nation first grid scale 300mw small modular reactor 24',
 'new climate news edge wave continental shelf fuel 2021 acapulco bay tsunami sciencedaily',
 'volumetric modular construction arrive hungary company deal bad esg score contact u hello cc visit page',
 'rishi cabal give fuckerty gibet folk want look jacob rees mogg deliver important warn via',
 'draw oecd analysis amp data virtual pavilion bring event finance mitigate adaptation amp register join u 27 oct 18 nov',
 'dr sultan ahmed al jaber meet global energy leader adnoc',
 'new climate news epa award 3m small business continue development innovative environmental technology',
 'uk wealthy power elite plan reverse get rid pretend matter pretend massive matter worship know despise brit']
```

Data Visualization

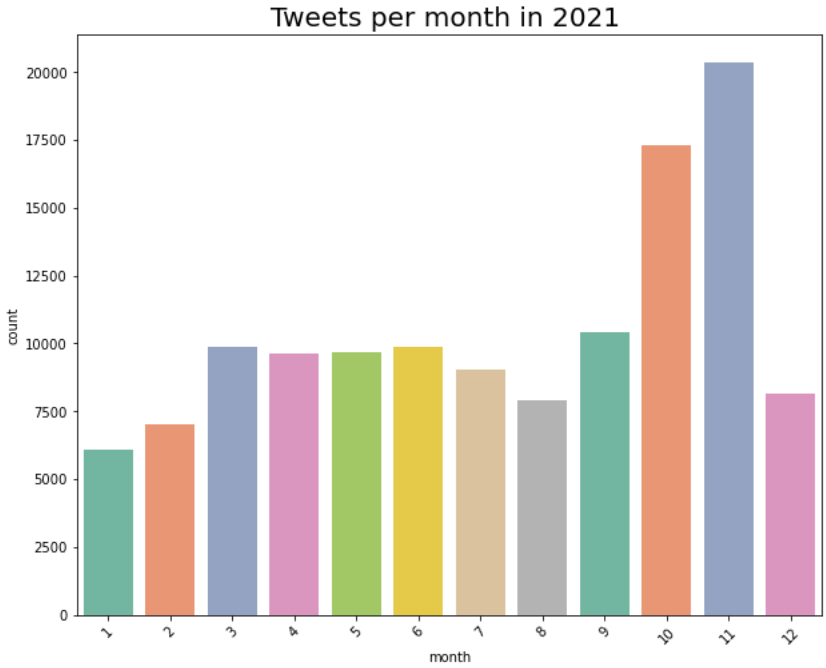
```
#data_list=["carbon","offsets","credit","blockchain"]
data_list=["carbon","offsets","credit","john oliver","oliver"]
pattern="|".join(data_list)
data18_2_sent=data18[(data18["cleaned_tweet"].str.contains(pattern))]
sns.countplot(x=data18_2_sent["sentiments_val"]);
plt.title("Sentiment Distribution of Tweets ", fontsize=20)
#plt.savefig("Downloads/johnoliver_sent2122.png")
```



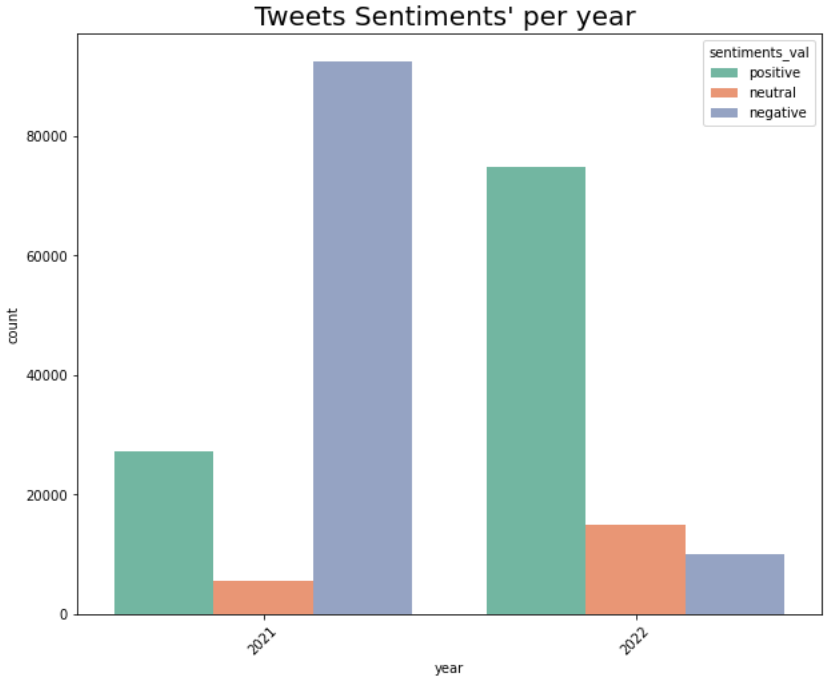

```
# plot Tweets count
plt.subplots(figsize = (10,8))
data22=data18[data18["year"]==2022]
chart = sns.countplot(x="month",data=data22, palette="Set2");
chart.set_xticklabels(chart.get_xticklabels(), rotation=45)
plt.title("Tweets per month in 2022 ", fontsize=20)
plt.savefig("Downloads/num_tweets2022.png")
plt.show();
```



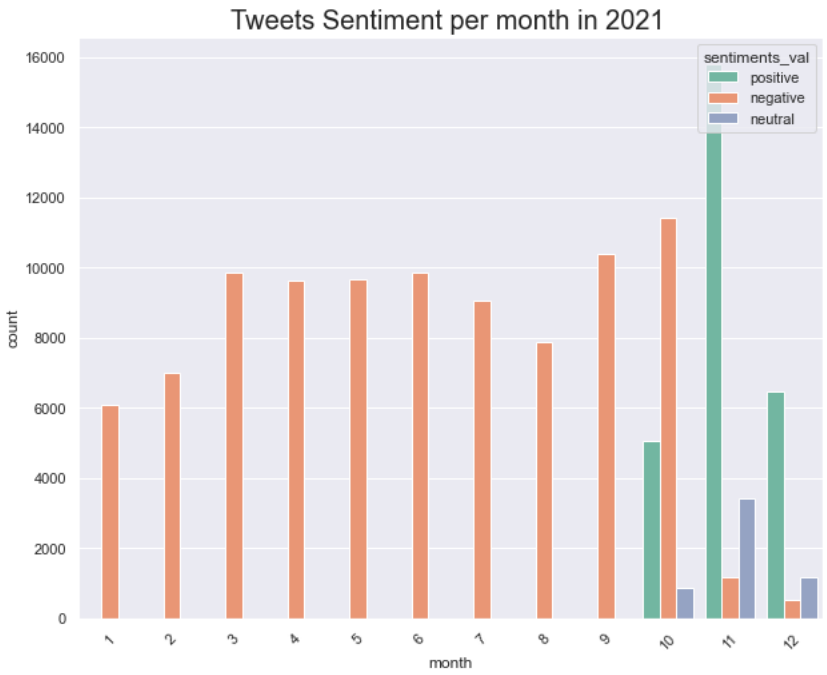

```
# plot Tweets count
plt.subplots(figsize = (10,8))
data21=data18[data18["year"]==2021]
chart = sns.countplot(x="month",data=data21, palette="Set2");
chart.set_xticklabels(chart.get_xticklabels(), rotation=45)
plt.title("Tweets per month in 2021 ", fontsize=20)
plt.savefig("Downloads/num_tweets2021.png")
plt.show();
```



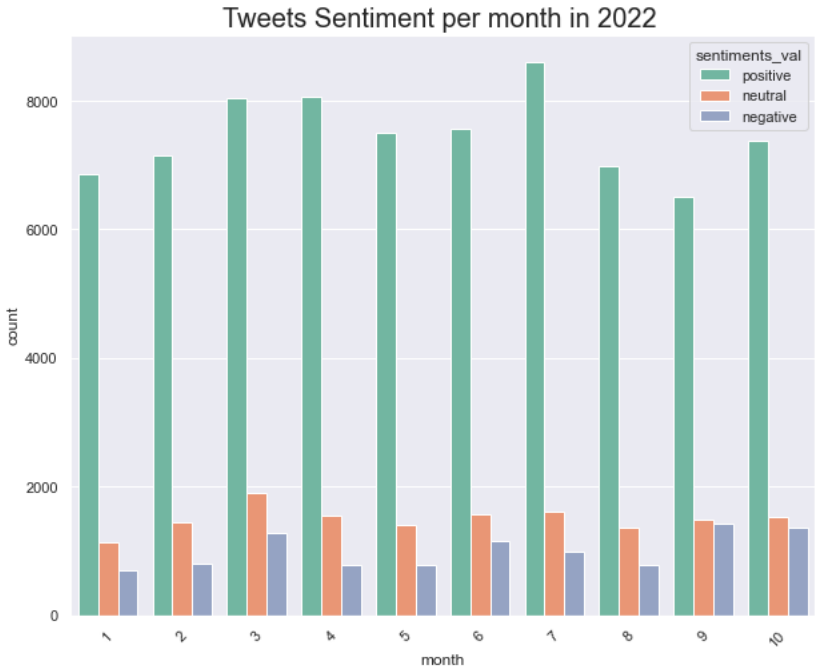

```
# plotting Tweets Sentiments for each year
plt.subplots(figsize = (10,8))
chart = sns.countplot(x="year",data=data18, palette="Set2",hue="sentiments_val");
chart.set_xticklabels(chart.get_xticklabels(), rotation=45)
plt.title("Tweets Sentiments' per year ", fontsize=20)
plt.savefig("Downloads/Tweets_per_year.png")
plt.show();
```



```
# plotting Tweets Sentiments for each year
plt.subplots(figsize = (10,8))
chart = sns.countplot(x="month",data=data21, palette="Set2",hue="sentiments_val");
chart.set_xticklabels(chart.get_xticklabels(), rotation=45)
plt.title("Tweets Sentiment per month in 2021 ", fontsize=20)
#plt.savefig("Downloads/Tweets_per_year.png")
plt.show();
```

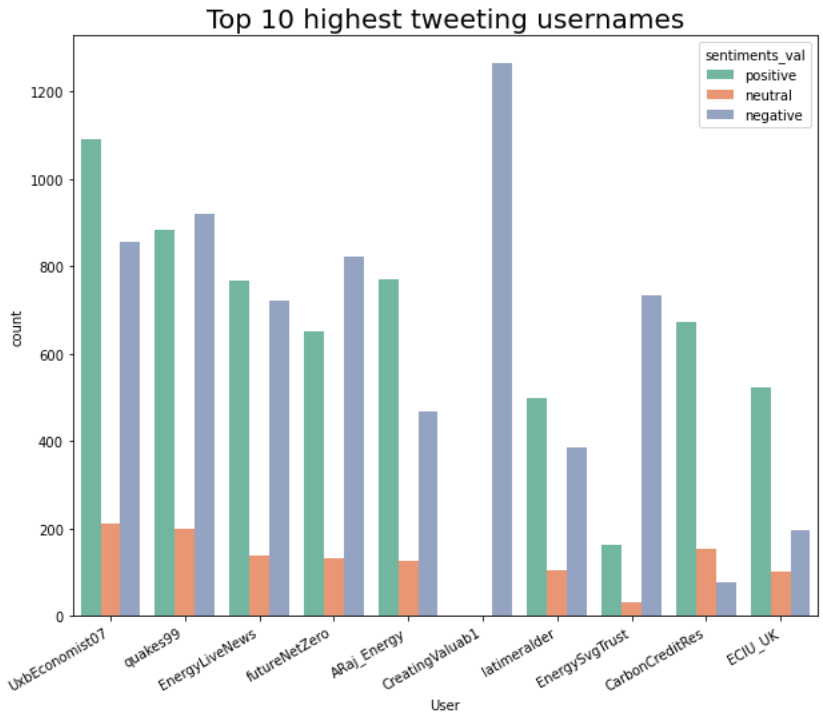



```
# plotting Tweets Sentiments for each year
plt.subplots(figsize = (10,8))
chart = sns.countplot(x="month",data=data22, palette="Set2",hue="sentiments_val");
chart.set_xticklabels(chart.get_xticklabels(), rotation=45)
plt.title("Tweets Sentiment per month in 2022 ", fontsize=20)
#plt.savefig("Downloads/Tweets_per_year.png")
plt.show();
```

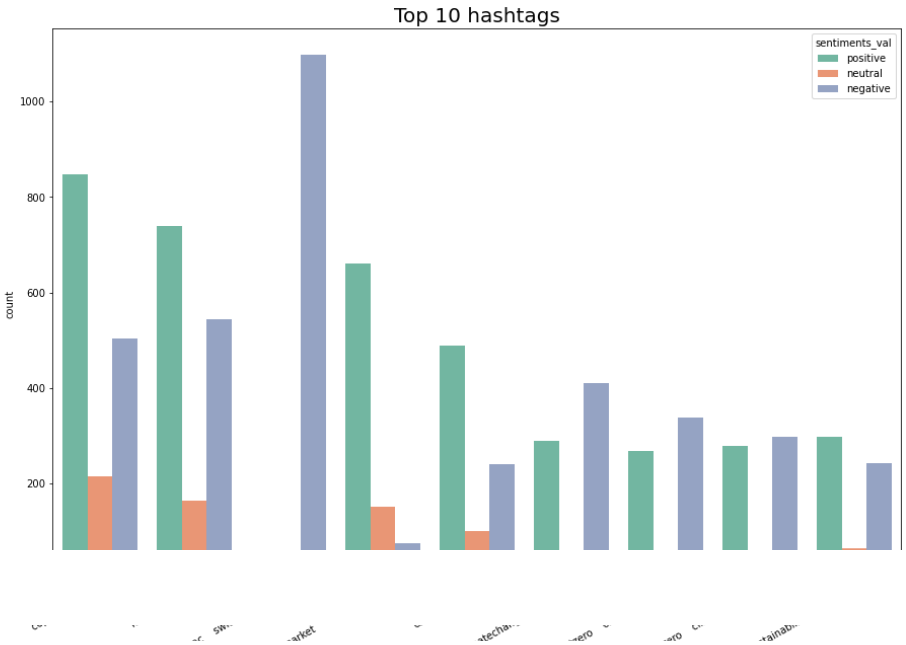



```
# Top 10 highest tweeting usernames
plt.subplots(figsize = (10,8))
plt.title("Top 10 highest tweeting usernames", fontsize=20)
chart=sns.countplot(x="User",hue="sentiments_val",data=data18,palette="Set2",
```

```
                    order= data18["User"].value_counts().iloc[:10].index);
chart.set_xticklabels(chart.get_xticklabels(), rotation=30, horizontalalignment='right');

plt.savefig("Downloads/top10_usernames_dist.png")
```




```
# plotting Top 10 hashtag
plt.subplots(figsize = (15,10))
plt.title("Top 10 hashtags", fontsize=20)
chart=sns.countplot(x="Hashtags",hue="sentiments_val",data=data18,palette="Set2",
                    order= data18["Hashtags"].value_counts().iloc[1:10].index);
chart.set_xticklabels(chart.get_xticklabels(), rotation=30, horizontalalignment='right');

plt.savefig("Downloads/top10_hashtags_dist.png")
```



Wordcloud

```
def create_wordcloud(text):
    words=' '.join([words for words in text])
    wordcloud = WordCloud(max_font_size=50, max_words=100, background_color="white").generate(words)
    plt.figure(figsize=(10, 7))
    plt.imshow(wordcloud, interpolation="bilinear")
    plt.axis('off')
    plt.show()
```

```
#wordcloud for all tweets in 2022
create_wordcloud(data22["cleaned_tweet"].values)

plt.savefig("Downloads/wordcloud_22_1.png")
```


```
<Figure size 432x288 with 0 Axes>
```

```
#wordcloud for positive tweets
create_wordcloud(data22[data22["sentiment"]==1]["cleaned_tweet"].values)

plt.savefig("Downloads/wordcloud_22_pos1.png")
```


```
#wordcloud for positive tweets
create_wordcloud(data22[data22["sentiment"]==-1]["cleaned_tweet"].values)

plt.savefig("Downloads/wordcloud_22_neg1.png")
```


```
<Figure size 432x288 with 0 Axes>
```

Sentiment Curve for 2022

```
data_list=["carbon","offsets","credit","blockchain","carbon credits", "carbon offsets"]
pattern="|".join(data_list)
data22_sent=data18[(data18["cleaned_tweet"].str.contains(pattern))]

data22_sent=data22_sent[data22_sent["year"]==2022]


len(data22_sent)
```

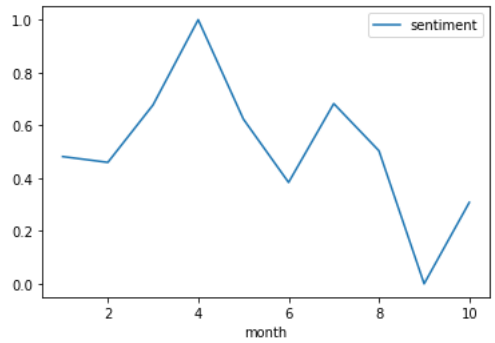
```
15007
```

```
data22_sent = data22_sent[data22_sent["sentiment"]!='no data']


data22_sent_gp=data22_sent.groupby(['month'])["sentiment"].sum()


data22_sent_gp=data22_sent_gp.reset_index()
data22_sent_gp
```

	month	sentiment
0	1	958
1	2	949
2	3	1038
3	4	1170
4	5	1016
5	6	918
6	7	1040
7	8	967
8	9	761
9	10	887

```
from sklearn import preprocessing

X = preprocessing.MinMaxScaler()
scaled_sent22= pd.DataFrame(X.fit_transform(data22_sent_gp.iloc[:,1:]),columns=data22_sent_gp.columns[1:])
scaled_sent22["month"]=data22_sent_gp["month"]
```

```
scaled_sent22.set_index('month').plot();

plt.savefig("Downloads/22_month_sent.png")
```




```
data22_sent=data18[(data18["cleaned_tweet"].str.contains(pattern))]
data21_sent=data22_sent[data22_sent["year"]==2021]


len(data21_sent)
```

```
18367
```

```
data21_sent = data21_sent[data21_sent["sentiment"]!='no data']
data21_sent_gp=data21_sent.groupby(['month'])["sentiment"].sum()

data21_sent_gp=data21_sent_gp.reset_index()
data21_sent_gp
```

	month	sentiment
0	1	-966
1	2	-1061
2	3	-1460
3	4	-1364
4	5	-1362
5	6	-1528
6	7	-1503
7	8	-1225
8	9	-1503
9	10	-1172
10	11	2039
11	12	947

```
from sklearn import preprocessing

X = preprocessing.MinMaxScaler()
scaled_sent21= pd.DataFrame(X.fit_transform(data21_sent_gp.iloc[:,1:]),columns=data21_sent_gp.columns[1:])
scaled_sent21["month"]=data21_sent_gp["month"]


scaled_sent21.set_index('month').plot();

plt.savefig("Downloads/21_month_sent.png")
```

data22

	User	verified	Date_Created	Follows_Count	Friends_Count	Retweet_Count	Language	Date_Tweet	Number_of_Likes	S
0	CarbonCredits	False	2017-06-21 17:44:31+00:00	6799	283	0	en	2022-10-31 23:36:00+00:00	5	
2	M_Costelloe	False	2012-05-03 02:19:44+00:00	604	819	0	en	2022-10-31 23:29:50+00:00	3	
16	PatriotHydrogen	False	2022-08-15 11:29:38+00:00	7	1	2	en	2022-10-31 23:13:50+00:00	1	
26	SGPPartnership	False	2012-09-05 15:14:54+00:00	1060	897	1	en	2022-10-31 23:05:05+00:00	1	
28	PeriCarbon	False	2022-08-16 23:58:10+00:00	3	18	0	en	2022-10-31 23:02:17+00:00	2	
...	...	...	...	...	...	...	...	...	...	
170710	RealJohnWynne	False	2010-11-26 16:57:20+00:00	1086	1428	0	en	2022-01-01 00:43:01+00:00	1	
170711	equitableenergy	False	2014-08-30 17:02:31+00:00	448	2616	1	en	2022-01-01 00:38:18+00:00	3	
170712	stratandbiz	False	2011-08-31 14:52:45+00:00	154372	4000	1	en	2022-01-01 00:30:06+00:00	3	
170715	Cmh176Hughes	False	2013-01-10 01:09:09+00:00	281	728	1	en	2022-01-01 00:21:35+00:00	1	
170716	ProfDaveWorsley	False	2010-04-07 08:47:24+00:00	851	753	1	en	2022-01-01 00:18:22+00:00	3	

99799 rows × 25 columns

EDA of results

```
#data22.to_csv('carbondata_labeled_custom22.csv', index=False)
#data21.to_csv('carbondata_labeled_custom21.csv', index=False)

data22 = pd.read_csv('carbondata_labeled_custom22.csv')
data21 = pd.read_csv('carbondata_labeled_custom21.csv')
```

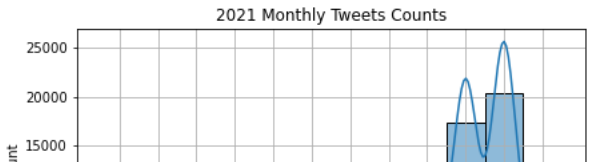
```
C:\Users\dantr\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3444: DtypeWarning: Columns (23) have mixed types.Specify dt
  exec(code_obj, self.user_global_ns, self.user_ns)
```

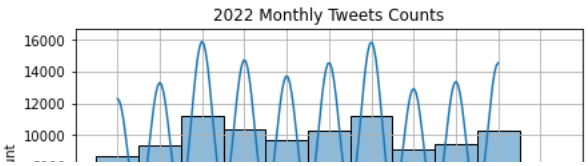
```
data22
```

	User	verified	Date_Created	Follows_Count	Friends_Count	Retweet_Count	Language	Date_Tweet	Number_of_Likes	So
0	CarbonCredits	False	2017-06-21 17:44:31+00:00	6799	283	0	en	2022-10-31 23:36:00+00:00	5	
1	M_Costelloe	False	2012-05-03 02:19:44+00:00	604	819	0	en	2022-10-31 23:29:50+00:00	3	T
2	PatriotHydrogen	False	2022-08-15 11:29:38+00:00	7	1	2	en	2022-10-31 23:13:50+00:00	1	
3	SGPPartnership	False	2012-09-05 15:14:54+00:00	1060	897	1	en	2022-10-31 23:05:05+00:00	1	
4	PeriCarbon	False	2022-08-16 23:58:10+00:00	3	18	0	en	2022-10-31 23:02:17+00:00	2	
...	...	...	...	...	...	...	...	...	...	...
99794	RealJohnWynne	False	2010-11-26 16:57:20+00:00	1086	1428	0	en	2022-01-01 00:43:01+00:00	1	
99795	equitableenergy	False	2014-08-30 17:02:31+00:00	448	2616	1	en	2022-01-01 00:38:18+00:00	3	
99796	stratandbiz	False	2011-08-31 14:52:45+00:00	154372	4000	1	en	2022-01-01 00:30:06+00:00	3	
99797	Cmh176Hughes	False	2013-01-10 01:09:09+00:00	281	728	1	en	2022-01-01 00:21:35+00:00	1	
99798	ProfDaveWorsley	False	2010-04-07 08:47:24+00:00	851	753	1	en	2022-01-01 00:18:22+00:00	3	Tv

99799 rows × 25 columns

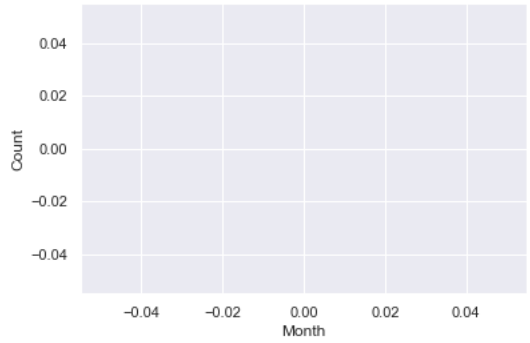
```
plt.figure(figsize=(15, 8))

plt.subplot(221)
sns.histplot(x=data21.month,stat='count',binwidth=1,kde='true',discrete=True)
plt.title('2021 Monthly Tweets Counts')
plt.xticks(np.arange(1,13,1))
plt.grid()

plt.subplot(222)
sns.histplot(x=data22.month,stat='count',binwidth=1,kde='true',discrete=True)
plt.title('2022 Monthly Tweets Counts')
plt.xticks(np.arange(1,13,1))
plt.grid()
```






```
#ax=plt.subplot(221)
sns.lineplot(data21.month.value_counts())
ax.set_xlabel("Month")
ax.set_ylabel('Count')
```

```
C:\Users\dantr\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. Fr
  warnings.warn(
Text(0, 0.5, 'Count')
```




```
print(data21['TweetC'][data21['month']==9][:10])
```

```
45856    Well done to the @AusHydCouncil releasing its ...
45857    A friend in Melbourne told me they won't buy a...
45858    I was on the phone with Paul Polman the night ...
45859    Celebrating @BAFTA @WeAreALBERT 10th anniversa...
45860    #NetZero #renewable dependence &gt; European e...
45861    WSP were engaged by the ACT Government to deli...
45862    "We highlight three ‘bugs’ in the current syst...
45863    “The world’s biggest carbon-sucking machine is...
45864    A growing number of countries and companies ha...
45865    There's no time to waste #ClimateAction #Clima...
Name: TweetC, dtype: object
```

```
print(data21['clean_tweet'][data21['month']==9][:10])
```

```
45856    ['well', 'do', 'release', 'first', 'white', 'p...
45857    ['friend', 'melbourne', 'tell', 'buy', 'apartm...
45858    ['phone', 'paul', 'polman', 'night', 'appoint'...
45859    ['celebrate', '10th', 'anniversary', 'much', '...
45860    ['dependence', 'gt', 'european', 'energy', 'cr...
45861    ['wsp', 'engage', 'act', 'government', 'delive...
45862    ['highlight', 'three', 'bug', 'current', 'syst...
45863    ['world', 'biggest', 'carbon', 'suck', 'machin...
45864    ['grow', 'number', 'country', 'company', 'pled...
45865                                 ['time', 'waste']
Name: clean_tweet, dtype: object
```

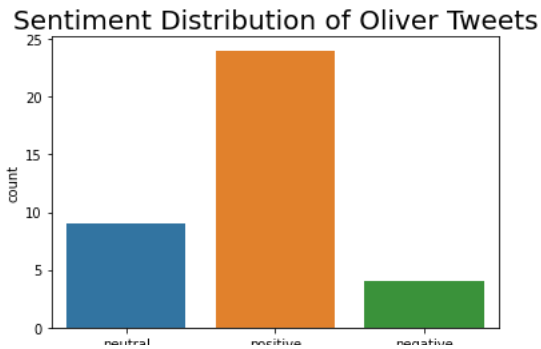
```
data22 = data22[data22['cleaned_tweet'].notna()]
```

```
len(data22)
```

```
99691
```

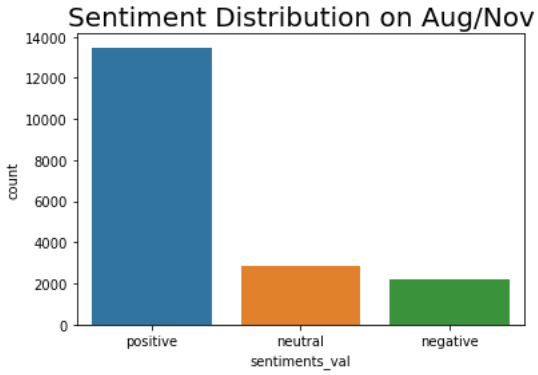
```
data_list=["john oliver","oliver"]
pattern="|".join(data_list)

data22_oliver = data22[data22['month'].isin([8,9])]
data22_oliver=data22_oliver[(data22_oliver["cleaned_tweet"].str.contains(pattern))]
sns.countplot(x=data22_oliver["sentiments_val"]);

plt.title("Sentiment Distribution of Oliver Tweets", fontsize=20)
```

```
Text(0.5, 1.0, 'Sentiment Distribution of Oliver Tweets')
```




```
data22_fall = data22[data22['month'].isin([8,9])]
sns.countplot(x=data22_fall["sentiments_val"]);
plt.title("Sentiment Distribution on Aug/Nov", fontsize=20)
```

```
Text(0.5, 1.0, 'Sentiment Distribution on Aug/Nov')
```




```
data21[['Date_Tweet','TweetC']][(data21['month'].isin([11,12]))]
```

	Date_Tweet	TweetC
0	2021-12-31 22:54:09+00:00	I'm so glad that in the meantime US Defence Ch...
1	2021-12-31 22:53:48+00:00	It's time to normalise #NetZero carbon sustain...
2	2021-12-31 22:49:18+00:00	Happy new year 2022 to the lovers and defender...
3	2021-12-31 22:44:49+00:00	#netzero? #ESG goals? Behind the headlines and...
4	2021-12-31 22:15:10+00:00	How can executives drive industries and organi...
...	...	...
28523	2021-11-01 00:28:31+00:00	We must have to remember that, our response to...
28524	2021-11-01 00:28:02+00:00	GETAnalysis: The decades of endless #HollowPro...
28525	2021-11-01 00:25:03+00:00	✈️🌎 The world's leaders are gathering to decla...
28526	2021-11-01 00:11:18+00:00	The potential of nature-based solutions for cl...
28527	2021-11-01 00:06:00+00:00	@AlboMP @Bowenchris Will you rule out any futu...

28528 rows × 2 columns

Unsupported Cell Type. Double-Click to inspect/edit the content.