The Pangaia Brand Social Media Analyze

Objective : Use Twitter to perform an analysis of the conversational data collected related to the brand. Construct and critique a semantic model of a social media conversation for the purpose of deducing user opinion, collecting feedback and using it to inform product and or marketing decisions.

Task 1 :

Create a list of 5 possible keywords you could use to identify relevant tweets from Twitter. Demonstrate that there is enough conversational data available on Twitter to perform further analysis.

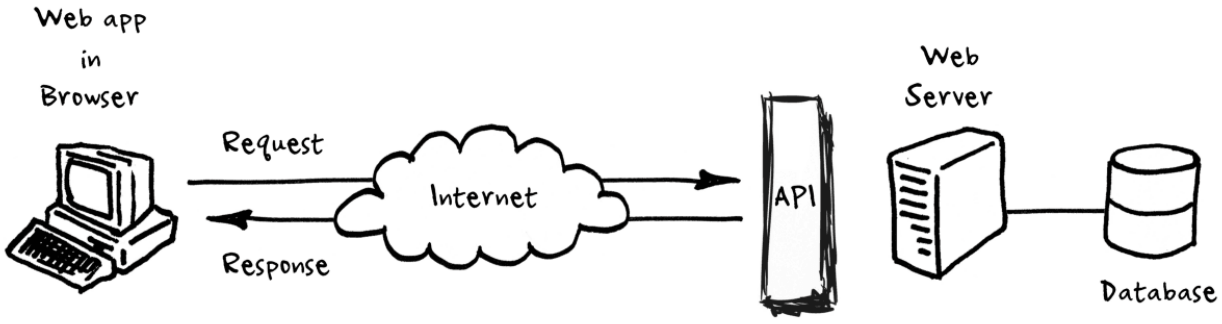
The Keyword which is used to fetch the tweets related to the pangaia brand are following.

1) #PANGAIA
2) @thepangaia
3) Air Pollution ink
4) PANGAIA fashion
5) PANGAIA Collection

Task 2 :

Explain what is meant by an API and compare and contrast the two data collection APIs available on the Twitter platform.

An API is a set of programming code that enables data transmission between one software product and another. It also contains the terms of this data exchange.



There are Two API available for fetching twitter data which are following:

Tweepy: Tweepy is open-sourced, hosted on github and enables Python to communicate with Twitter platform and use its API,Tweets can be customized to have a string which identifies the app which was used.It doesn't reveal user password, making it more secure.
It's easier to manage the permissions, for example a set of tokens and keys can be generated that only allows reading from the timelines, so in case someone obtains those credentials, he/she won't be able to write or send direct messages, minimizing the risk,The application

doesn't reply on a password, so even if the user changes it, the application will still work, One of the main usage cases of tweepy is monitoring for tweets and doing actions when some event happens. The tweepy needs Twitter API so a user has to access the twitter developer account first to get the tokens and keys

Twint : has an advanced tool for Twitter scraping. We can use this tool to scrape any user's tweets without having to use Twitter API.Twitter scraping tool written in Python that allows for scraping Tweets from Twitter profiles, Twint utilizes Twitter's search operators to let you scrape Tweets from specific users, scrape Tweets relating to certain topics, hashtags & trends, or sort out sensitive information from Tweets like email and phone numbers.
Twint has these major benefits**:**

1) Twitter API has restrictions to scrape only the last 3200 Tweets. But Twint can fetch almost all Tweets.
2) Set up is really quick as there is no hassle of setting up Twitter API.
3) Can be used anonymously without Twitter sign-up.
4) It's free!! No pricing limitations.
5) Provides easy to use options to store scraped tweets into different formats — CSV, JSON, SQLite, and Elasticsearch

Task 3 :

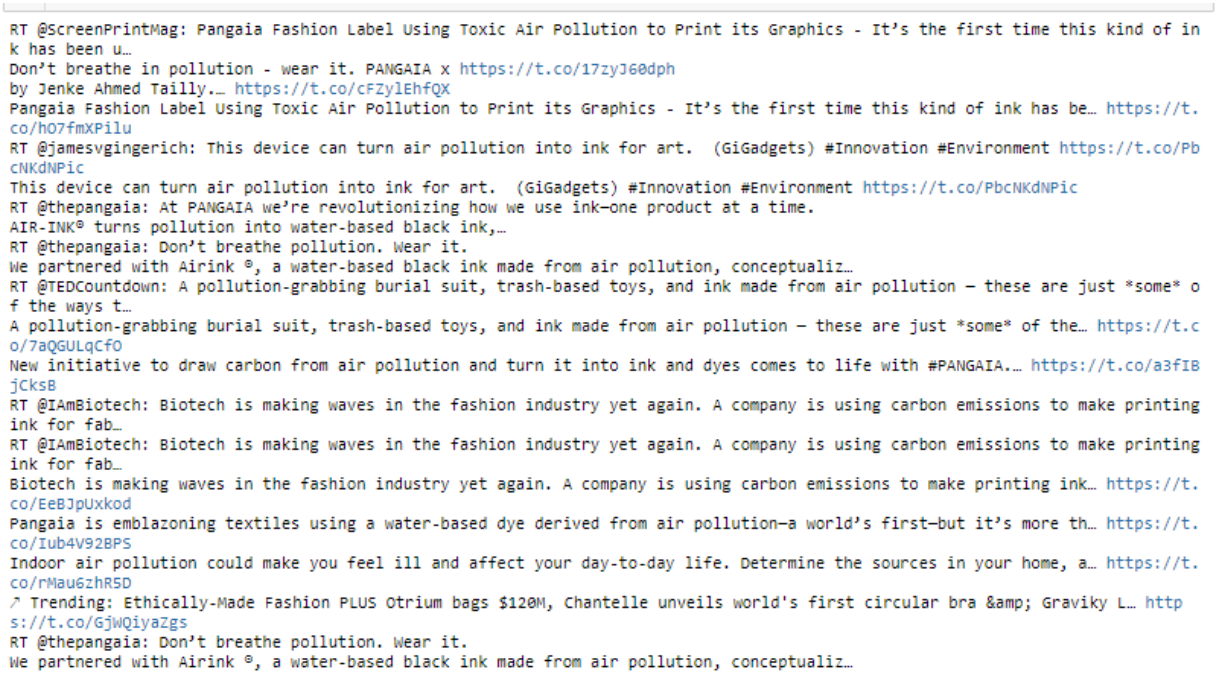
Using your suggested keywords from part a) and your knowledge of Twitter, collect a series of Tweets surrounding The Pangaia and save them to a file. Your collected Tweets should span a minimum ONE week period. Provide evidence of how your data was collected (screenshots, code print outs with relevant comments), the total number of tweets collected and describe key methodological steps.

Code snapshots:

In [9]:

```
#Libraries
import os
#Tweepy Library
import tweepy as tw
import pandas as pd

#Twitter API Keys and Tokens
consumer_key= 'PWSW8ixkM13L35pREUbsGuYSG'
consumer_secret= 'VIaYRkNNn9oZmQtENZLvNQzlavBGRimMmOlyTQvvpPnVhbHMYL'
access_token= '826107931846463488-rBaRFGTxHVXyKDMlyh76439HxtiqrK2'
access_token_secret= 'KVyJwx9tMZAIoPSd4edWcPGRIB9o7fCNQUceNktnPabtG'

#Authentication
auth = tw.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tw.API(auth, wait_on_rate_limit=True)
auth = tw.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api = tw.API(auth)

# Define the search term and the date_since date as variables
search_words = "PANGAIA Collection"

#Loop through every tweet then save the features of tweets in a list
for tweet in tw.Cursor(api.search,q=search_words,lang="en").items():
        print(tweet.text)
        #Tweet Level
        tId.append(tweet.id)
        tweetText.append(tweet.text)
        tDate.append(tweet.created_at)
        rc.append(tweet.retweet_count)
        fc.append(tweet.favorite_count)
        favourite.append(tweet.favorited)
        retweeted.append(tweet.retweeted)
        #User Level
        uDate.append(tweet.user.created_at)
        screen_name.append(tweet.user.screen_name)
        sc.append(tweet.user.statuses_count)
        followerCount.append(tweet.user.followers_count)
        friendCount.append(tweet.user.friends_count)
        userFavCount.append(tweet.user.favourites_count)
```

Output Snapshot



The Total number of tweets which are collected over the seven days are 2050 with 13 important features from which there are only 372 unique/distinct tweets.

the features are 'TweetID', 'TweetDate', 'TweetText', 'retweetCount', 'FavouriteCount', 'IsFavourite', 'ISretweetd', 'userDate', 'screen_name', 'status_count', 'follower_count', 'freind_count', 'userFavouriteCount'

methodological steps:

1) First to get the Twitter API from the developer account.
2) In the second step I fetch the pangaia twitter accounts tweets and analyze the major important keywords, then extract those keywords to mine more tweets from twitter related to the pangaia brand.

3) Wite the code for fetching keywords tweets using tweepy library and save the data into list, transform json data into rows and columns using pandas.
4) After building code start fetching the tweets on daily base.

Task 4 :

Using a suitable example, discuss the role of text pre-processing in the context of social media analysis. Identify TWO pre-processing steps relevant to the dataset you created in part c) and apply them to your dataset. In your report you MUST detail the code used to perform each pre-processing step and provide evidence that they have been applied.

With the rise of Social Media, people obtain and share information almost instantly on a 24/7 basis. Many research areas have tried to gain valuable insights from these large volumes of freely available user generated content.extracting meaningful and actionable knowledge from user generated content is a complex endeavor. First, each social media service has its own data collection specificities and constraints, second the volume of messages/posts produced can be overwhelming for automatic processing and mining, and last but not the least, social media texts are usually short, informal, with a lot of abbreviations, jargon, slang and idioms for this purpose we need text preprocessing to analyze each text and its feature to gain insights from the text.

Preprocessing Steps:

Step 1:

1) Removing unwanted hashtags, smile icons, links, RT tags, mentions and converting all the text into lower case.we developed a python script to clean hashtags, mentions and smile icons.

```
[55] # removing everything except alphabets`
     ### Removing un wanted links hastgas ect
     lst=[]
     for i in  (frame['TweetText']):
        text=' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])|(\w+:\/\/\S+)"," ",i).split())
        lst.append(text)
     frame['TweetText']=lst




     ###Cleaning RT
     z = lambda x:re.compile('RT').sub('', x).strip()
     lst=[]
     for i in frame['TweetText']:
       lst.append(z(i))
     frame['TweetText']=lst


     #lower case
     frame['TweetText']=frame['TweetText'].str.lower()
```

Data Before Preprocessing steps 1

As we see in the above picture the tweet text contain mentions, hashtags, icons etc

After Preprocessing step 1

```
'Super impressed with the materials engineering being done by and her team at Amazing',
'vladkis Don t forget to promote the little guys that are on the way up They',
'Nature Tech Sustainable Fashion Our final HealthandWealth session of the day with s an',
'Kicking off EarthDay by highlighting some stand out sustainable brands',
'A collaboration between Media Lab spinoff founded by alum and founded by alum',
'Kicking off EarthDay by highlighting some stand out sustainable brands',
'A collaboration between Media Lab spinoff founded by alum and founded by alum',
'A collaboration between Media Lab spinoff founded by alum and founded by alum',
'A collaboration between Media Lab spinoff founded by alum and founded by a',
'My latest article for explores carbon extraction from air pollution for textile printing a collaboration b',
'The colour range of collections are just',
'Kicking off EarthDay by highlighting some stand out sustainable brands',
'I did not know either of them Thanks for sharing',
'vladkis I ve seen some great strides with brands like and Tons of innovation in a spac',
'Congratulations to Spinoff by our alum this partnership with',
'Meet 1st fashion label to print graphics using ink made out of toxic air pollution',
'SPOTTED in Oxford Street PANGAIAtaxis',
'Kicking off EarthDay by highlighting some stand out sustainable brands',
'Kicking off EarthDay by highlighting some stand out sustainable brands',
'For EarthDay we are showcasing 6 ClimateTech startups that are setting examples in driving change',
'For EarthDay we are showcasing 6 ClimateTech startups that are setting examples in driving change',
'We can all do our bit for EarthDay Great guide courtesy of mentalhealth wellbeing sustainability',
'Kicking off EarthDay by highlighting some stand out sustainable brands',
'Kourtney is wearing here s the link KUWTK',
'Join us and s for our third Health Wealth of America conference with and',
'Join us and s for our third Health Wealth of America conference with',
```

In this picture we saw that the mentions, hashtags and smile icons were removed.

Step 2:

2) In the second step we apply Porter Stemmer, word lemmatizer, tokenizer and bag of words using python Nltk library.

Stemming: Stemming refers to truncating words of their affixes (prefixes or suffixes) to approximate them to their root form. This is often done with a lookup dictionary of prefixes and suffixes, making it computationally fast. However, there is a performance trade-off. In the English language, some affixes change the meaning of the word completely, resulting in inaccurate feature representation.

Lemmatization: The alternative to stemming is lemmatization, where words are reduced to their lemmas, or root form. This is done using a lookup dictionary of words and their lemmas, hence resulting in it being more computationally expensive. However, performance is often better, since features are represented more accurately.

Bag of words: Bag of words is a way to represent text data numerically. Text data is essentially split into words (or more accurately, tokens), which are features. The frequency of each word in each text data is the corresponding feature values. For example, we might represent "I love cake very very much" as a bag of words dictionary.

'I':1, 'love':1, 'cake':1, 'very':2, 'much':1 }

```
#Downloading NLTK Packagaes
nltk.download('punkt')
nltk.download('wordnet')
#loading Word Net lemmatizer
from nltk.stem import WordNetLemmatizer
def preprocess_news(df):
    """
    Function to take dataframe of tweets and apply three steps
    1) Stemming
    2) Lemmatizing
    3) Tokenization
    """
    corpus=[]
    stem=PorterStemmer()  #stemming
    lem=WordNetLemmatizer() #lemmatizing
    for tweet in frame['TweetText']:
        words=[w for w in nltk.tokenize.word_tokenize(tweet) if (w not in stop_words)]
        words=[lem.lemmatize(w) for w in words if len(w)>2]

        corpus.append(words)
    return corpus
```

Output after this step :

```
 'therefore',
 'decided',
 'make',
 'mini',
 'series'],
['pangaia',
 'launch',
 'first',
 'ever',
 'pop',
 'store',
 'excited',
 'new',
 'experience',
 'london',
 'visit'],
['playful', 'pop', 'pangaia', 'selfridges', 'oxford', 'street', 'colour'],
['people',
 'started',
 'showing',
 'interest',
 'foodforest',
 'project',
 'pangaia',
 'therefore',
 'decided',
 'make'
```

Bag of Words Code

▸ Bag of Word Model

```
#Bag of words Model
dic=gensim.corpora.Dictionary(corpus)
bow_corpus = [dic.doc2bow(doc) for doc in corpus]
bow_corpus
```

```
 (714, 1)],
[(0, 1), (58, 1), (633, 1), (686, 1), (687, 1), (688, 1)],
[(222, 1), (408, 2), (409, 2), (410, 1), (411, 1), (412, 1)],
[(0, 1), (58, 1), (633, 1), (686, 1), (687, 1), (688, 1)],
[(222, 1), (408, 2), (409, 2), (410, 1), (411, 1), (412, 1)],
[(222, 1), (408, 2), (409, 2), (410, 1), (411, 1), (412, 1)],
[(222, 1), (408, 1), (409, 2), (410, 1), (411, 1), (412, 1)],
[(10, 1),
 (11, 1),
 (19, 1),
 (33, 1),
 (142, 1),
 (222, 1),
 (271, 1),
 (715, 1),
 (716, 1),
 (717, 1)],
[(2, 1), (264, 1), (513, 1)],
[(0, 1), (58, 1), (633, 1), (686, 1), (687, 1), (688, 1)],
[(476, 1), (718, 1), (719, 1), (720, 1)],
```

Task 5 :

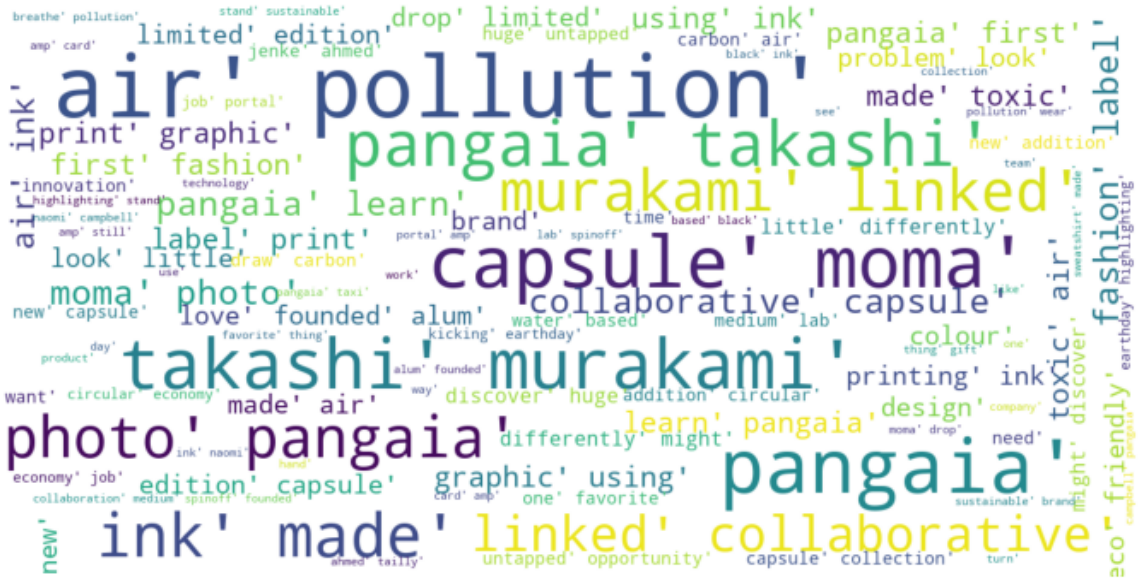
Create a Python program to count the most commonly used words in your dataset and use it to generate a "word cloud". In your report you MUST include a table of the top 10 most commonly used words, your Python code and a screenshot of your word cloud.

Top 10 most commonly used Words
Air Pollution
Fashion
Pangaia
Takashi
Capsule
Moma
Murakami
Linked

Collaborative

Word Cloud is a great way to represent text data. The size and color of each word that appears in the word cloud indicate it's frequency or importance.

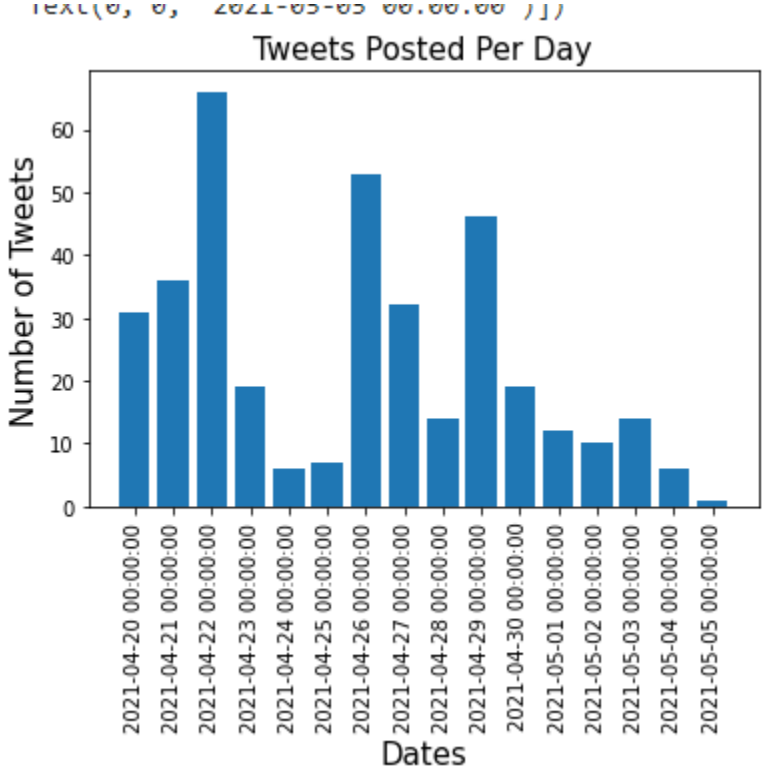
```
from wordcloud import WordCloud, STOPWORDS
stopwords = set(STOPWORDS)
def show_wordcloud(data):
    """
    Function to generate word cloud
    """
     wordcloud = WordCloud(
         background_color='white',
         stopwords=stopwords,
         max_words=100,
         max_font_size=30,
         scale=3,
         random_state=1)

     wordcloud=wordcloud.generate(str(data))
     fig = plt.figure(1, figsize=(12, 12))
     plt.axis('off')
     plt.imshow(wordcloud)
     plt.show()

show_wordcloud(corpus)
```



Task 6 :

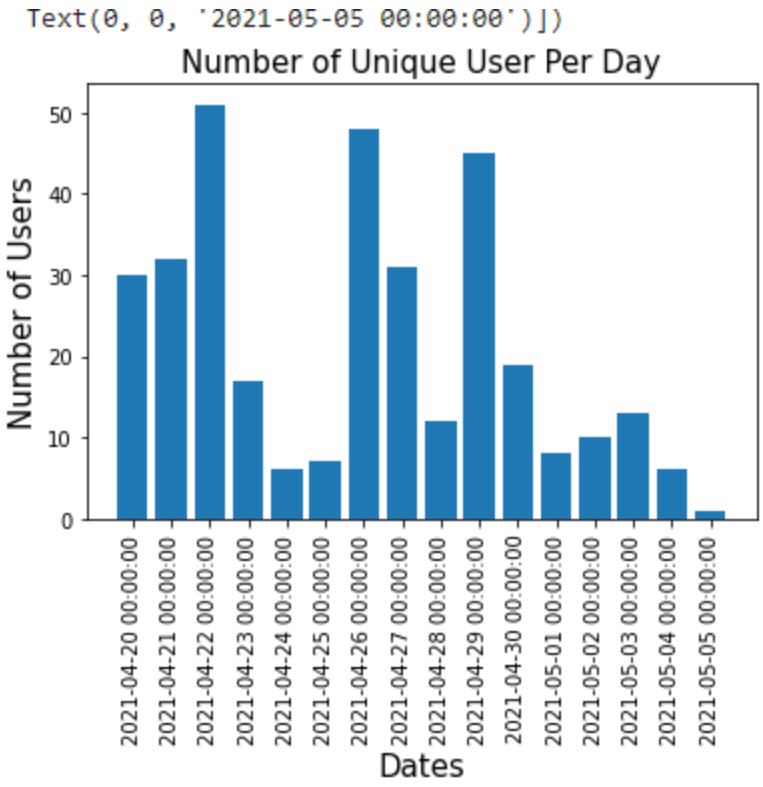
Use your processed data file to produce a series of graphs or charts to summarise the following information. I. The number of tweets posted per day II. The number of unique users per day III. The top 10 most active users over the entire period In your report you MUST detail your processing steps and comment on the results.

The Number of Tweet Posted Per Day

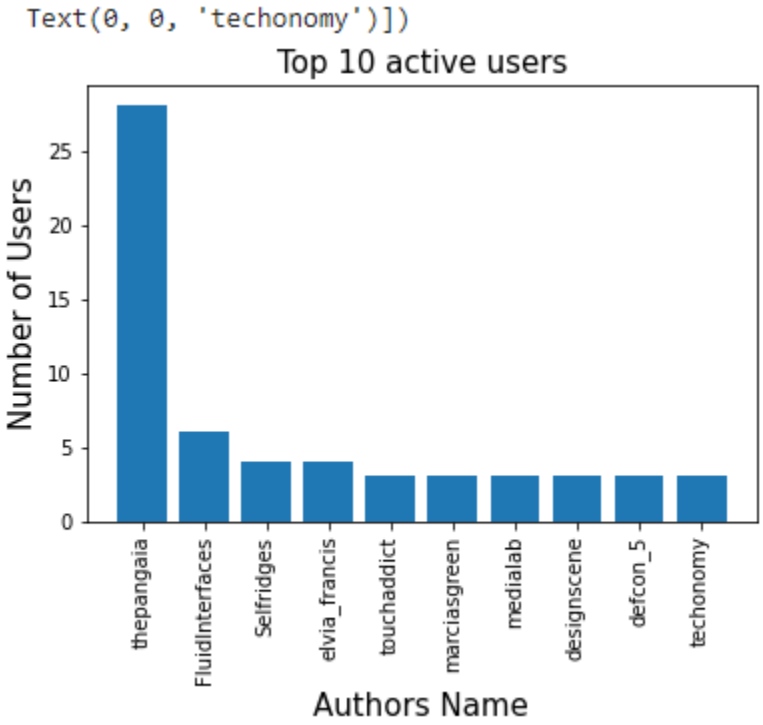


From the above graph we analyze that 22,26,29 april has greater tweets than other days.

The number of unique per day



Top 10 Active users



This are the top ten authors which are the part of spreading the tweets.

Task 7 :

Using a suitable approach, construct a LDA topic model to identify themes of discussion within your dataset. In your report you MUST; - Discuss what is meant by topic modelling and explain how your chosen approach works - Provide details of the steps that you have carried out. - Use any tables, graphs and charts you feel are necessary to illustrate your findings - Provide a critical evaluation of your model and discuss one strength and one weakness

Topic modeling is the process of using unsupervised learning techniques to extract the main topics that occur in a collection of documents.

Latent Dirichlet Allocation (LDA) is an easy to use and efficient model for topic modeling. Each document is represented by the distribution of topics and each topic is represented by the distribution of words.LDA analyses the words in each paper and calculates the joint probability distribution between the observed (words in the paper) and the unobserved (the hidden structure of topics).

The objective of LDA is to perform dimensionality reduction However, we want to preserve as much of the class discriminatory information as possible. extract the relevant information by reducing the redundancy and minimizing the noise.

Steps to build LDA Model

But before getting into topic modeling we have to pre-process our data a little. We will:

- *tokenize*: the process by which sentences are converted to a list of tokens or words.
- *remove stopwords*
- *lemmatize*: reduces the inflectional forms of each word into a common base or root.
- *convert to the bag of words*: Bag of words is a dictionary where the keys are words(or ngrams/tokens) and values are the number of times each word occurs in the corpus.
- Then we create the LDA Model

The LDA Model: We extract four topics from LDA model

Topic 1:

'0.059*"air" + 0.056*"ink" + 0.046*"pollution" + 0.045*"pangaia" + 0.033*"made" + 0.032*"fashion" + 0.029*"using" + 0.028*"label" + 0.026*"print" + 0.026*"first"'.

Topic 2 :

'0.037*"pollution" + 0.029*"air" + 0.026*"ink" + 0.019*"pangaia" + 0.016*"collaboration" + 0.016*"sustainable" + 0.015*"founded" + 0.015*"alum" + 0.013*"carbon" + 0.013*"lab"

Topic 3:

'0.080*"pangaia" + 0.038*"capsule" + 0.036*"moma" + 0.030*"murakami" + 0.028*"takashi" + 0.022*"learn" + 0.022*"collaborative" + 0.021*"photo" + 0.021*"linked" + 0.016*"amp"

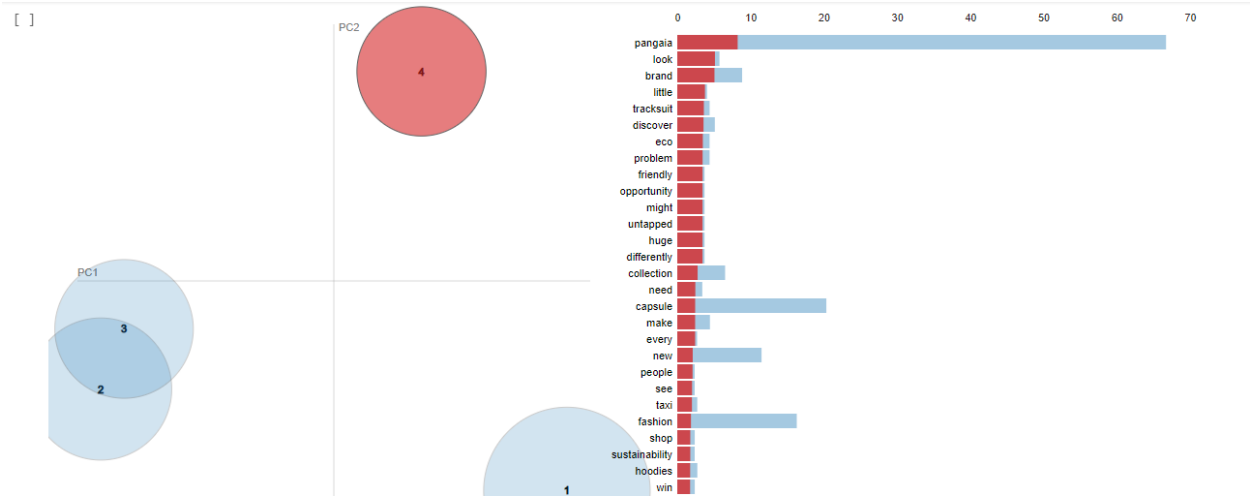
Topic 4 :

'0.030*"pangaia" + 0.019*"look" + 0.018*"brand" + 0.014*"little" + 0.013*"tracksuit" + 0.013*"discover" + 0.013*"eco" + 0.012*"problem" + 0.012*"friendly" + 0.012*"opportunity"

```
lda_model = gensim.models.LdaMulticore(bow_corpus,
                                       num_topics = 4,
                                       id2word = dic,
                                       passes = 10,
                                       workers = 2)
lda_model.show_topics()
```

```
[(0,
  '0.059*"air" + 0.056*"ink" + 0.046*"pollution" + 0.045*"pangaia" + 0.033*"made" + 0.032*"fashion" + 0.029*"using" + 0.028*"label" + 0.026*"print" + 0.026*"first
 (1,
  '0.037*"pollution" + 0.029*"air" + 0.026*"ink" + 0.019*"pangaia" + 0.016*"collaboration" + 0.016*"sustainable" + 0.015*"founded" + 0.015*"alum" + 0.013*"carbon"
 (2,
  '0.080*"pangaia" + 0.038*"capsule" + 0.036*"moma" + 0.030*"murakami" + 0.028*"takashi" + 0.022*"learn" + 0.022*"collaborative" + 0.021*"photo" + 0.021*"linked"
 (3,
  '0.030*"pangaia" + 0.019*"look" + 0.018*"brand" + 0.014*"little" + 0.013*"tracksuit" + 0.013*"discover" + 0.013*"eco" + 0.012*"problem" + 0.012*"friendly" + 0.0
```

Cluster Modeling



In the above graph we analyze that on the left side, the area of each circle represents the importance of the topic relative to the corpus. As there are four topics, we have four circles.

The distance between the center of the circles indicates the similarity between the topics. Here you can see that the topic 3 and topic 4 overlap, this

indicates that the topics are more similar.

On the right side, the histogram of each topic shows the top 30 relevant words. For example, in topic 1 the most relevant words are police, new, may, war, etc

Dominant topic and its percentage contribution in each document

In LDA models, each document is composed of multiple topics. But, typically only one of the topics is dominant.below is the figure which shows the table of Dominant topics.

	Document_No	Dominant_Topic	Topic_Perc_Contrib	Keywords	Text
0	0	3.0	0.9416	pangaia, capsule, moma, murakami, takashi, ink...	eco friendly essential the eco friendly brand ...
1	1	1.0	0.9386	pangaia, amp, ink, air, founded, alum, new, po...	new initiative to draw carbon from air polluti...
2	2	1.0	0.9386	pangaia, amp, ink, air, founded, alum, new, po...	new initiative to draw carbon from air polluti...
3	3	3.0	0.9416	pangaia, capsule, moma, murakami, takashi, ink...	eco friendly essential the eco friendly brand ...
4	4	3.0	0.9416	pangaia, capsule, moma, murakami, takashi, ink...	eco friendly essential the eco friendly brand ...
5	5	3.0	0.9416	pangaia, capsule, moma, murakami, takashi, ink...	eco friendly essential the eco friendly brand ...
6	6	2.0	0.7430	one, favorite, fashion, gift, thing, pollution...	re selling blue pangaia tracksuit in size xs p...
7	7	3.0	0.8895	pangaia, capsule, moma, murakami, takashi, ink...	check out the latest pangaia looks modeled by ...
8	8	3.0	0.9368	pangaia, capsule, moma, murakami, takashi, ink...	s new air ink capsule and the hoodies t shirts...
9	9	0.0	0.9300	pangaia, air, pollution, fashion, print, using...	family can be biological but it s also about a...

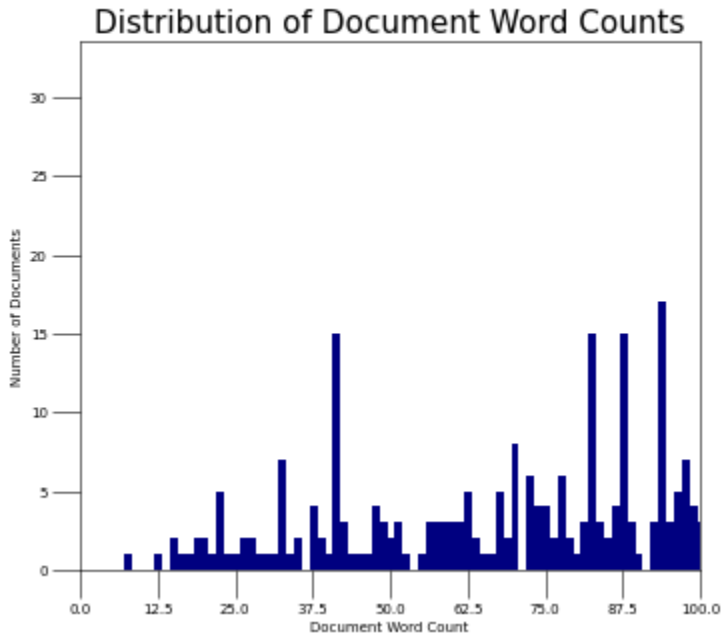
The most representative sentence for each topic

samples of sentences that most represent a given topic.

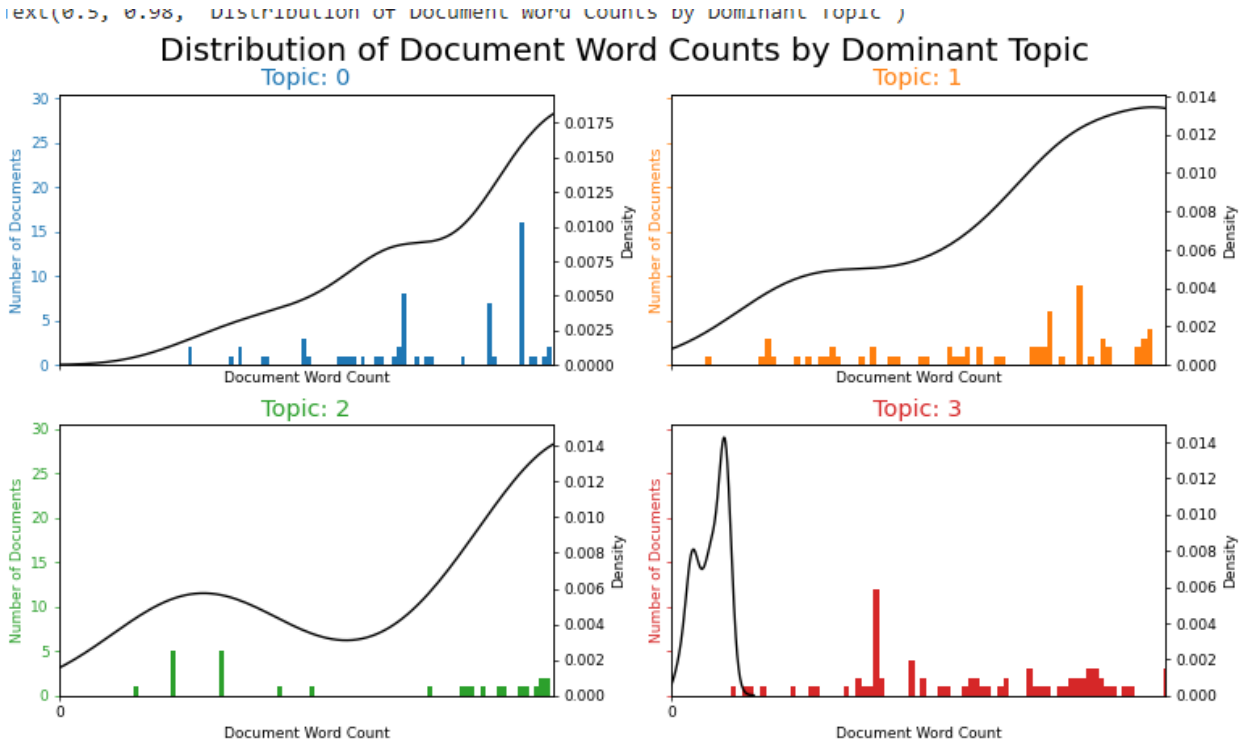
	Topic_Num	Topic_Perc_Contrib	Keywords	Representative Text
0	0.0	0.9495	pangaia, air, pollution, fashion, print, using, made, label, ink, first	137 check out these fashion brands focused on sustainable alternatives pangaia yes friends sep j...
1	1.0	0.9463	pangaia, amp, ink, air, founded, alum, new, pollution, lab, tracksuit	ladygaga was spotted leaving her rome hotel while sporting the alc winona rib knit top 365 organ...
2	2.0	0.9479	one, favorite, fashion, gift, thing, pollution, make, shop, company, day	biotech is making waves in the fashion industry yet again a company is using carbon emissions to...
3	3.0	0.9509	pangaia, capsule, moma, murakami, takashi, ink, learn, air, collaborative, photo	at pangaia we re revolutionizing how we use ink one product at a time air ink turns pollution in...

Frequency Distribution of Word Counts in Documents

When working with a large number of documents, we want to analyze how big the documents are as a whole and by topic. Let's plot the document word counts distribution.



Distribution of Words Topic Wise:

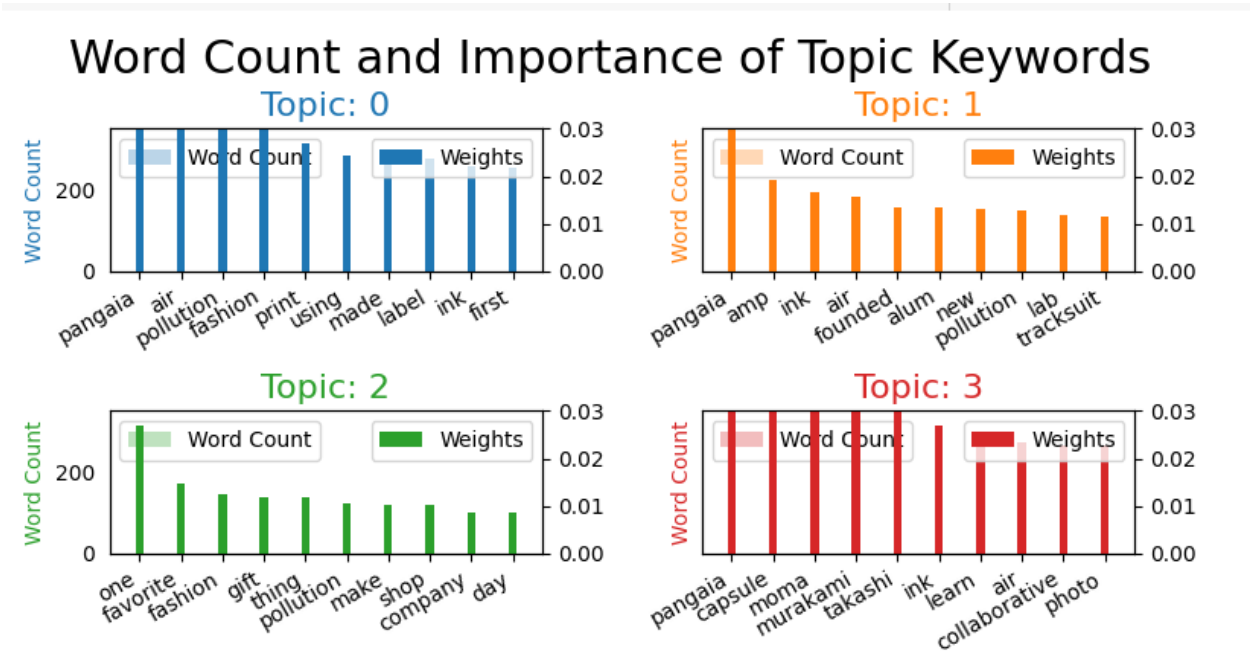


Word Clouds of Top N Keywords in Each Topic:

Though you've already analyzed what are the topic keywords in each topic, a word cloud with the size of the words proportional to the weight is a pleasant sight. The coloring of the topics I've taken here is followed in the subsequent plots as well.

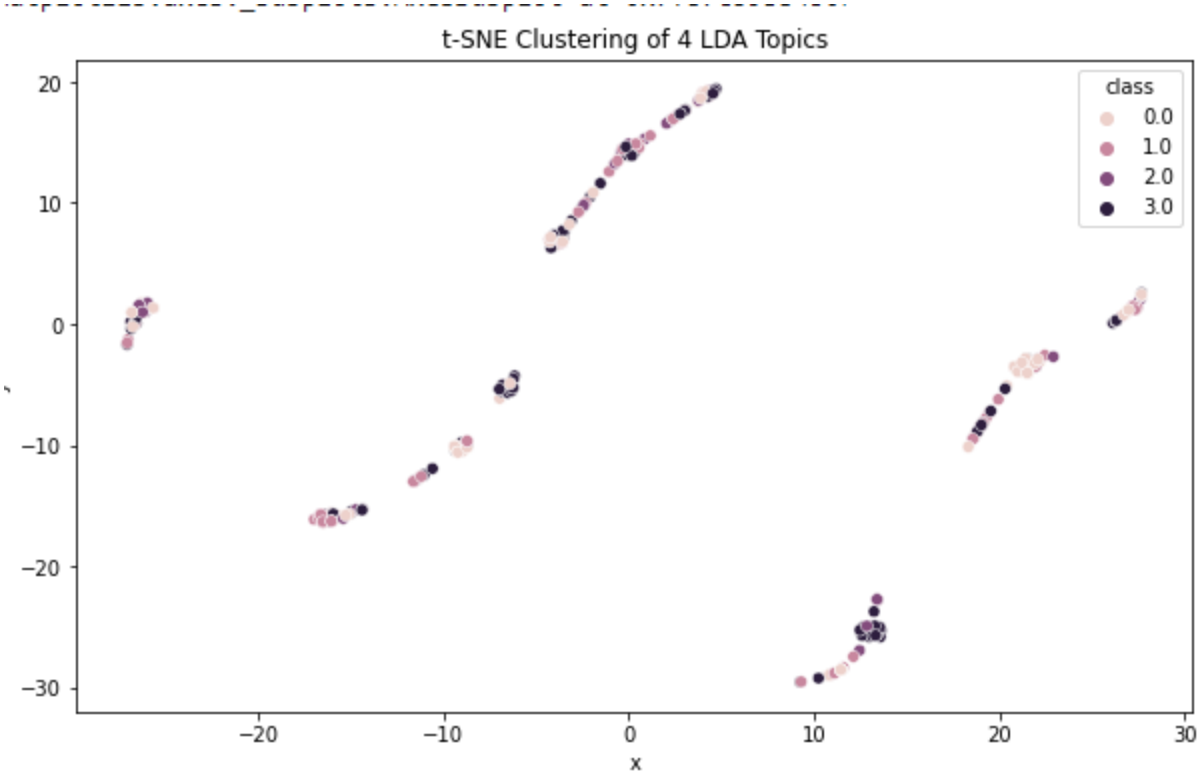
Word Counts of Topic Keywords

When it comes to the keywords in the topics, the importance (weights) of the keywords matters. Along with that, how frequently the words have appeared in the documents is also interesting to look.



t-SNE Clustering:

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets.t-Distributed stochastic neighbor embedding (t-SNE) minimizes the divergence between two distributions: a distribution that measures pairwise similarities of the input objects and a distribution that measures pairwise similarities of the corresponding low-dimensional points in the embedding.We had analyzed our topic modeling using t sne clustering plots.



The strength of the LDA model is that LDA is a probabilistic model with interpretable topics. The weakness of the LDA model is the number of topics is fixed and must be known ahead of time, Uncorrelated topics.

Task 8:

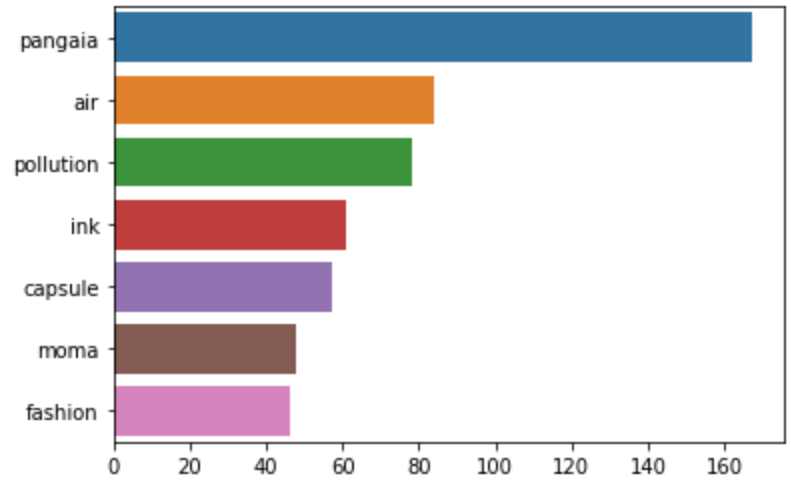
Apply noun phrase recognition to your dataset and identify the top five most mentioned noun phrases. Construct a sentiment model for each of your identified noun phrases and compare and contrast the differences in both polarity and sentiment, In your report you MUST; - Discuss what is meant by sentiment modelling Provide details of the steps that you have carried out to build and evaluate your models. Use any tables, graphs and charts you feel are necessary to illustrate your findings. Provide a critical evaluation of your models and discuss one strength and one weakness.

Noun Phrase recognition :

Noun phrase recognition is the part of speech tagging method that assigns part of speech labels to words in a sentence. There are eight main parts of speech.

The top five noun which recognize are the following

<matplotlib.axes._subplots.AxesSubplot at 0x7f87b9fd21d0>



Sentiment Modeling:

Sentiment analysis is basically the process of determining the attitude or the emotion of the writer, i.e., whether it is positive or negative or neutral.

It is essentially a multiclass text classification text where the given input text is classified into positive, neutral, or negative sentiment.

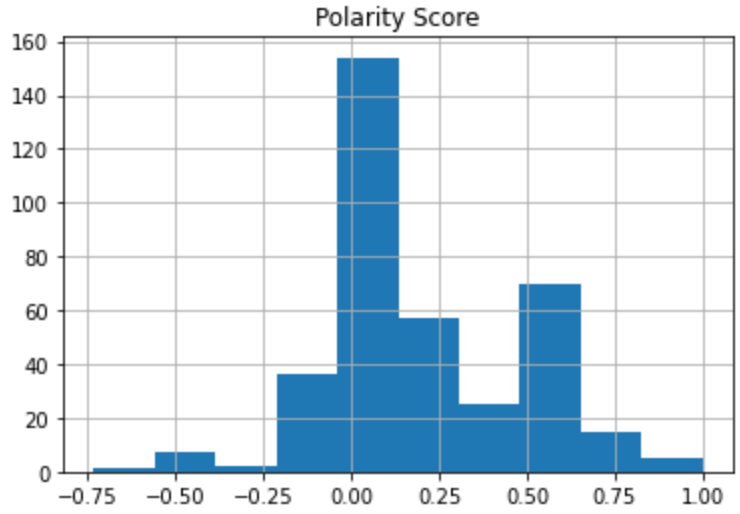
Rule-based sentiment analysis is one of the very basic approaches to calculate text sentiments.

We use Textblob and Vader Sentiment python library to extract the sentiments from the tweets.

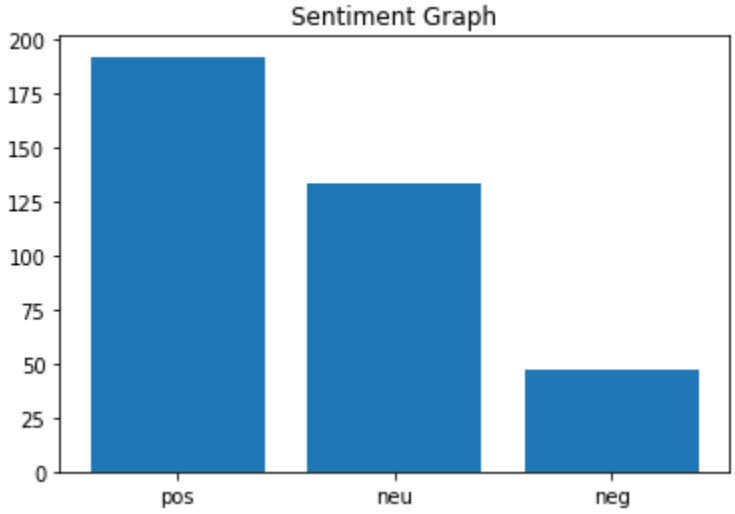
Textblob return two properties for the given input polarity and subjectivity.

Vader Sentiment uses a list of lexical features (e.g. word) which are labeled as positive or negative according to their semantic orientation to calculate the text sentiment, it returns a probability to a given input.

The strength of Textblob and Vader sentiment is that it is faster than NLTK and the weakness is that It does not provide features like dependency parsing, word vectors etc. which is provided by spacy.



From the above figure we analyze that the polarity mainly ranges between -0.75 to 1. majority of data lies between 0.00 to o.50 this mean majority tweets are positive.



From the above figure we analyze that the tweet related to the pangaia brand is very positive, people have a good image in their mind for the pangaia brand.

Code for the Sentiment Modeling

```
[336] from textblob import TextBlob
      import matplotlib.pyplot as plt
      from nltk.sentiment.vader import SentimentIntensityAnalyzer
      import nltk

      def sentiment_vader(text, sid):
          ss = sid.polarity_scores(text)
          ss.pop('compound')
          return max(ss, key=ss.get)

      def sentiment_textblob(text):
              x = TextBlob(text).sentiment.polarity

              if x<0:
                  return 'neg'
              elif x==0:
                  return 'neu'
              else:
                  return 'pos'
```

```
    def plot_sentiment_barchart(text, method='TextBlob'):
        if method == 'TextBlob':
            sentiment = text.map(lambda x: sentiment_textblob(x))
        elif method == 'Vader':
            nltk.download('vader_lexicon')
            sid = SentimentIntensityAnalyzer()
            sentiment = text.map(lambda x: sentiment_vader(x, sid=sid))
        else:
            raise ValueError('Textblob or Vader')
        plt.title('Sentiment Graph')
        plt.bar(sentiment.value_counts().index,
                sentiment.value_counts())
```