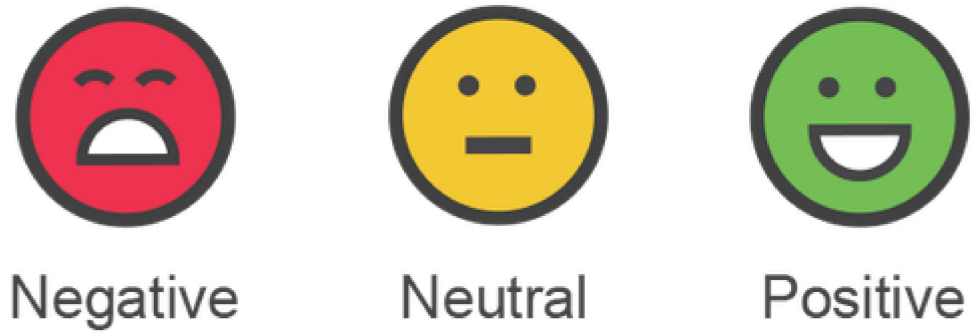
AIM: **Twitter Sentiment Analysis**

NAME: Palak Nath

PRN: 17070124048



IMPORTING THE LIBRARIES

```
import tweepy
from textblob import TextBlob
from wordcloud import WordCloud
import pandas as pd
```

```
import numpy as np
import re
import matplotlib.pyplot as plt
% matplotlib inline
plt.style.use('fivethirtyeight')
from sklearn.datasets import load_digits
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
```

LOADING THE ACCESS KEY INFORMATION

```
from google.colab import files
uploaded=files.upload()
```

Choose Files Access Tokens Twitter.csv

- **Access Tokens Twitter.csv**(application/vnd.ms-excel) - 408 bytes, last modified: 3/24/2021 - 100% done

```
Saving Access Tokens Twitter.csv to Access Tokens Twitter (1).csv
```

```
#Get the data
log=pd.read_csv('Access Tokens Twitter.csv')
```

AUTHENTICATING AND CREATING API OBJECT

```
#Twitter API credentials
consumerKey = log['Value'][0]
consumerSecret = log['Value'][1]
accessToken = log['Value'][3]
accessTokenSecret = log['Value'][4]


#Authentication Object
authenticate = tweepy.OAuthHandler(consumerKey, consumerSecret)

#Set the access token and the access token secret
authenticate.set_access_token(accessToken, accessTokenSecret)
```

```
authenticate.set_access_token(accessToken,accessTokenSecret)

#Create the API objects while passing the auth information
api = tweepy.API(authenticate,wait_on_rate_limit = True)
```

EXTRACTING 100 TWEETS FROM AN ACTIVE TWITTER USER

```
#SELECT THE TWITTER USER
twitter_screen_name = "BillGates" #I am choosing Bill Gates as he is trying to craete a positive impact


#GETTING THE POSTS OF THE TWITTER USER
posts = api.user_timeline(screen_name = twitter_screen_name, count=100, language= "en" ,tweet_mode= "extended" )

#Print 5 recent tweets from this account
print("Show the 5 recent Tweets from this account: \n")
i=1
for tweet in posts[0:5]:
  print(str(i) + ')'+ tweet.full_text + '\n')
  i=i+1
```

```
Show the 5 recent Tweets from this account:

1)RT @WHO: It's #WorldTBDay

Even as we battle #COVID19, we must not ease up the fight against #Tuberculosis, which remains the 🌎🌍🌏's deadlie…

2)I'm answering your questions now on @reddit: https://t.co/HXMsI0lSK1 https://t.co/XG7eAMNrGM

3)In this video, I answered some really good questions, including one about two really important numbers. Come ask me some ques

4)After you finish your pancakes this morning, come ask me anything on @reddit at 11:15 Pacific Time: https://t.co/R3wfDhtqxD h

5)It's deeply unfair that the people who contribute the least to climate change will suffer the worst from its effects: https:/
```

CONVERTING EXTRACTED POSTS INTO A DATAFRAME

```
#Create a Dataframe with a column called Tweets to make the extracted Tweets more accessible

df= pd.DataFrame( [tweet.full_text for tweet in posts] , columns=['Tweets'] )

#Show first 5 columns of this dataframe
df.head()
```

	Tweets
0	RT @WHO: It's #WorldTBDay\n\nEven as we battle...
1	I'm answering your questions now on @reddit: h...
2	In this video, I answered some really good que...
3	After you finish your pancakes this morning, c...
4	It's deeply unfair that the people who contrib...

GETTING DATETIME OF TWEETS

```
#Adding a column called Tweets to show datetime

dt= pd.DataFrame( [tweet.created_at for tweet in posts] , columns=['Datetime'] )
df['Datetime'] = dt
#Show first 5 columns of this dataframe
df.head()
```

	Tweets	Datetime
0	RT @WHO: It's #WorldTBDay\n\nEven as we battle...	2021-03-24 22:00:50

CREANING THE DATA

- Removing '@'
- Removing Hashtags
- Removing Hyperlinks

4	It's deeply unfair that the people who contrib...	2021-03-15 20:28:15

```
#Clean Text Function

def cleanTxt(text):
  text = re.sub(r'@[A-Za-z0-9]+', '', text) #substituting the @ with a empty string
#using r we are telling python it is a raw string containing @ followed by either a number, lowercase or uppercase character

  text = re.sub(r'#', '', text) #substituting the # with a empty string
  text = re.sub(r'RT[\s]', '', text) #removing RT retweet
  text = re.sub(r'https?:\/\/\S+', '', text) #removing any hyperlink

  return text


#Applying the Clean Text Function to the Tweets
df['Tweets'] = df['Tweets'].apply(cleanTxt)

#Show the Cleaned Texts
df
```

	Tweets	Datetime
0	: It's WorldTBDay\n\nEven as we battle COVID19...	2021-03-24 22:00:50
1	I’m answering your questions now on :	2021-03-19 18:14:28
2	In this video, I answered some really good que...	2021-03-19 17:27:50
3	After you finish your pancakes this morning, c...	2021-03-19 15:02:23
4	It’s deeply unfair that the people who contrib...	2021-03-18 20:26:15
...	...	...
95	: “I will continue fighting. I will continue s...	2020-11-25 18:04:32

MOST FREQUENTLY OCCURING WORDS: TOP 30

97	This week, Rashida and I ask a big question th...	2020-11-23 19:41:14

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(stop_words = 'english')
words = cv.fit_transform(df.Tweets)

sum_words = words.sum(axis=0)

words_freq = [(word, sum_words[0, i]) for word, i in cv.vocabulary_.items()]
words_freq = sorted(words_freq, key = lambda x: x[1], reverse = True)

frequency = pd.DataFrame(words_freq, columns=['word', 'freq'])

frequency.head(30).plot(x='word', y='freq', kind='bar', figsize=(12, 7), color = 'tomato')
plt.title("Most Frequently Occuring Words - Top 30")
```

```
Text(0.5, 1.0, 'Most Frequently Occuring Words - Top 30')
```

GET SUBJECTIVITY AND POLARITY

> **Subjectivity** tells about how opinionated the text is

> **Polarity** tells how Positive or Negative the text is

```
#Get Subjectivity function

def getSubjectivity(text):
  return TextBlob(text).sentiment.subjectivity

#create a function to get the polarity
def getPolarity(text):
  return TextBlob(text).sentiment.polarity

#Create two new columns
```

```
#Create two new columns
df['Subjectivity'] = df['Tweets'].apply(getSubjectivity)
df['Polarity'] = df['Tweets'].apply(getPolarity)

#SHow the new Dataframe with the new columns
df
```

	Tweets	Datetime	Subjectivity	Polarity
0	: It's WorldTBDay\n\nEven as we battle COVID19...	2021-03-24 22:00:50	0.000000	0.000000
1	I’m answering your questions now on :	2021-03-19 18:14:28	0.000000	0.000000
2	In this video, I answered some really good que...	2021-03-19 17:27:50	0.866667	0.566667
3	After you finish your pancakes this morning, c...	2021-03-19 15:02:23	0.000000	0.000000
4	It’s deeply unfair that the people who contrib...	2021-03-18 20:26:15	0.800000	-0.600000
...	...	...	...	...
95	: “I will continue fighting. I will continue s...	2020-11-25 18:04:32	0.312500	0.062500
96	Today is the 25th anniversary of my first book...	2020-11-24 18:21:39	0.393810	0.067143
97	This week, Rashida and I ask a big question th...	2020-11-23 19:41:14	0.470000	0.200000
98	: How can the fight to endpolio help inform th...	2020-11-22 19:31:40	0.000000	0.000000
99	You can listen to a bonus clip that didn’t mak...	2020-11-20 18:34:27	0.900000	0.500000

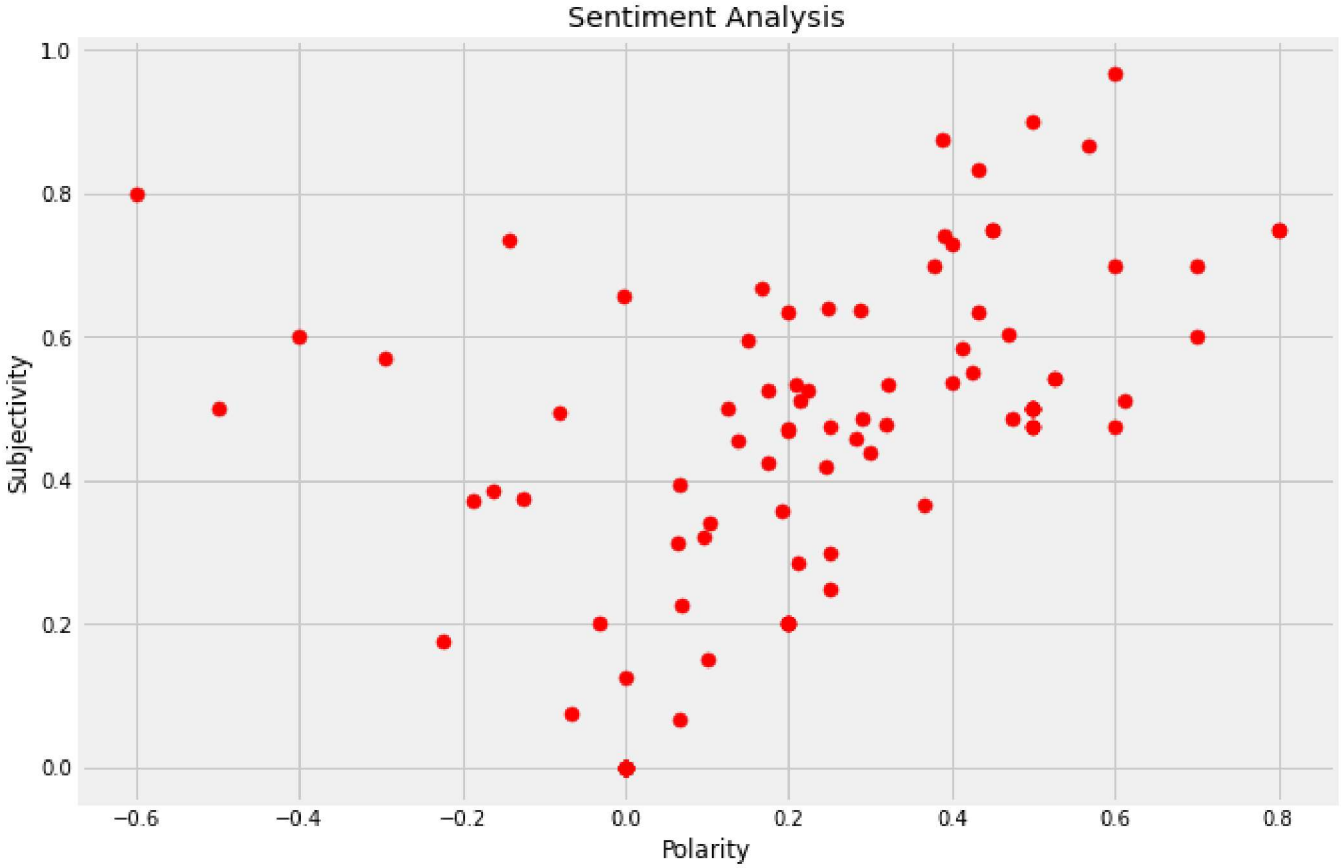
100 rows × 4 columns

SCATTER PLOT

> POLARITY VS SUBJECTIVITY

```
plt.figure(figsize=(10,7))
for i in range(0, df.shape[0]):
  plt.scatter(df['Polarity'][i],df['Subjectivity'][i], color='Red' , s=50)
plt.title('Sentiment Analysis')
```

```
plt.xlabel('Polarity')
plt.ylabel('Subjectivity')
plt.show()
```



FINDING LENGTH OF TWEEETS

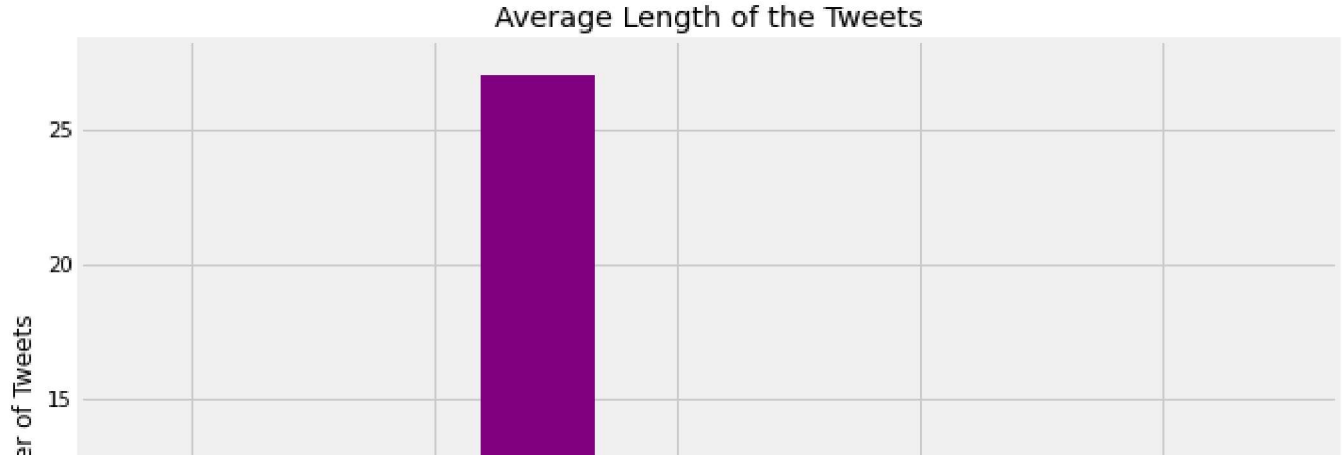
```
# adding a column to represent the length of the tweets

df['len'] = df['Tweets'].str.len()
df.head(10)
```

	Tweets	Datetime	Subjectivity	Polarity	len
0	: It's WorldTBDay\n\nEven as we battle COVID19...	2021-03-24 22:00:50	0.000000	0.000000	130
1	I’m answering your questions now on :	2021-03-19 18:14:28	0.000000	0.000000	39
2	In this video, I answered some really good que...	2021-03-19 17:27:50	0.866667	0.566667	158
3	After you finish your pancakes this morning, c...	2021-03-19 15:02:23	0.000000	0.000000	94
4	It’s deeply unfair that the people who contrib...	2021-03-18 20:26:15	0.800000	-0.600000	119
5	: Over the past few weeks health workers in Et...	2021-03-18 16:17:40	0.175000	-0.225000	130
6	I named my book “How to Avoid a Climate Disast...	2021-03-17 16:20:14	0.500000	0.500000	240
7	Congratulations on this important role. I’m e...	2021-03-11 23:46:36	0.875000	0.387500	210
8	Thank you for a great conversation about clim...	2021-03-11 20:36:16	0.475000	0.500000	100
9	For decades, Australian member Ruth Richar...	2021-03-10 18:58:27	0.000000	0.000000	250

NUMBER OF TWEETS VS THE LENTGH OF TWEETS

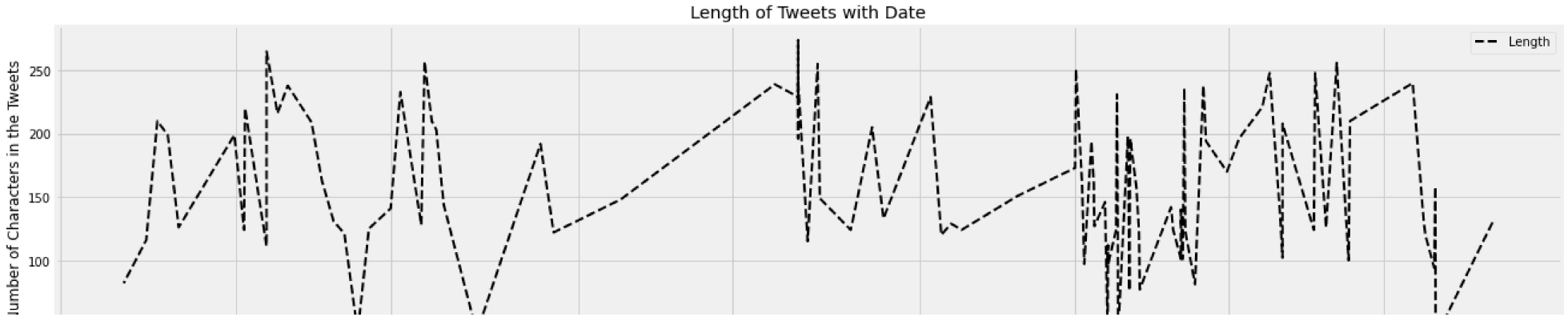
```
length_train = df['Tweets'].str.len().plot.hist(color = 'purple', figsize = (10, 7))
plt.xlabel('Length of the Tweet')
plt.ylabel('Number of Tweets')
plt.title('Average Length of the Tweets')
```

```
Text(0.5, 1.0, 'Average Length of the Tweets')
```



LENGTH OF TWEETS ACCORDING TO THE DATES TWEETED

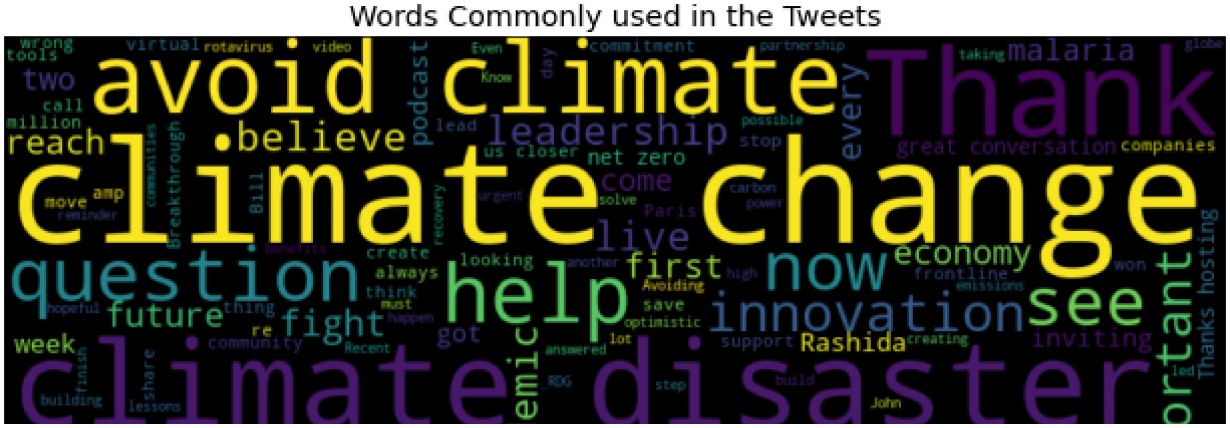
```
plt.rc('font', size=10)
fig, ax = plt.subplots(figsize=(20, 5))

# Specify how our lines should look
ax.plot(df.Datetime, df.len, color='black', label='Length', linestyle = '--', linewidth=2)

# Same as above
ax.set_xlabel('Date')
ax.set_ylabel('Number of Characters in the Tweets')
ax.set_title('Length of Tweets with Date')
ax.grid(True)
ax.legend(loc='upper right');
```



PLOTTING WORD CLOUD

> The more a word appears in the tweets the bigger it is

```
#Visualizing Word Cloud
allWords = ' '.join([twts for twts in df['Tweets']])
wordCloud = WordCloud(width= 800, height=500, random_state=21, max_font_size =160).generate(allWords)
plt.figure(figsize=(10, 7))
plt.imshow(wordCloud, interpolation = "bilinear")
plt.axis('off')
plt.title('Words Commonly used in the Tweets')
plt.show()
```



OBSERVATION: *We can clearly see the tweets as of 26th March 2021 seem to be using words like WORLD, CLIMATE CHANGE, AVOID, THANK etc*

CLASSIFYING THE TWEETS AS:

- POSITIVE
- NEGATIVE
- NEUTRAL

```
#create a function to compute the negative, neutral and positive analysis

def getAnalysis(score):
  if score < 0:
    return 'Negative'
  elif score == 0:
    return 'Neutral'
  else:
    return 'Positive'

#Create new column of Analysis in the dataframe
df['Analysis'] = df['Polarity'].apply(getAnalysis)
```

```
#show the Dataframe
df
```

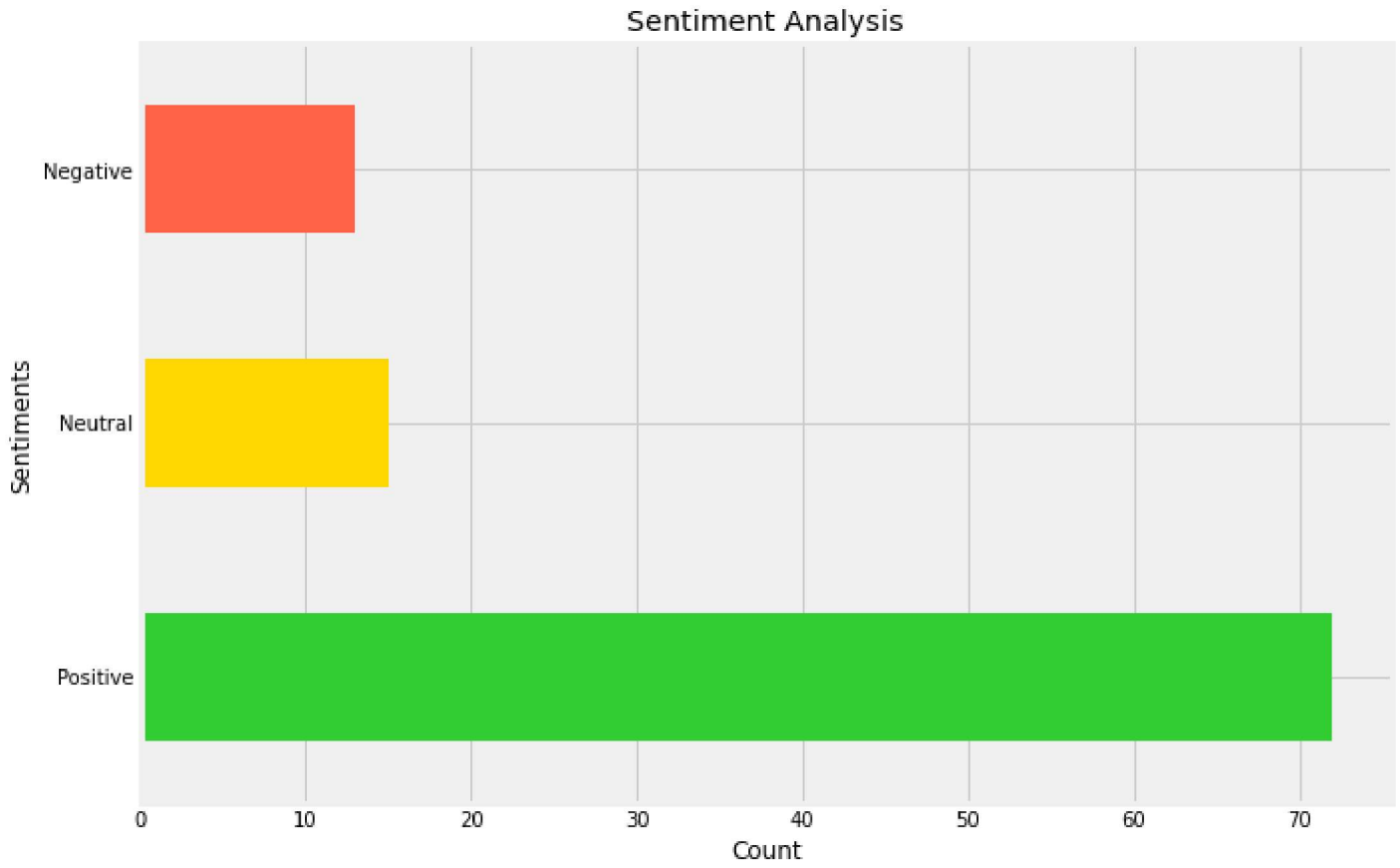
	Tweets	Datetime	Subjectivity	Polarity	len	Analysis
0	: It's WorldTBDay\n\nEven as we battle COVID19...	2021-03-24 22:00:50	0.000000	0.000000	130	Neutral
1	I’m answering your questions now on :	2021-03-19 18:14:28	0.000000	0.000000	39	Neutral
2	In this video, I answered some really good que...	2021-03-19 17:27:50	0.866667	0.566667	158	Positive
3	After you finish your pancakes this morning, c...	2021-03-19 15:02:23	0.000000	0.000000	94	Neutral
4	It’s deeply unfair that the people who contrib...	2021-03-18 20:26:15	0.800000	-0.600000	119	Negative
...	...	...	...	...	...	...
95	: “I will continue fighting. I will continue s...	2020-11-25 18:04:32	0.312500	0.062500	126	Positive
96	Today is the 25th anniversary of my first book...	2020-11-24 18:21:39	0.393810	0.067143	199	Positive
97	This week, Rashida and I ask a big question th...	2020-11-23 19:41:14	0.470000	0.200000	210	Positive
98	: How can the fight to endpolio help inform th...	2020-11-22 19:31:40	0.000000	0.000000	116	Neutral
99	You can listen to a bonus clip that didn’t mak...	2020-11-20 18:34:27	0.900000	0.500000	82	Positive

100 rows × 6 columns

SHOWING THE ANALYSIS COUNT

```
#Show the Value Counts
df['Analysis'].value_counts()

#plot and visualize the counts
plt.figure(figsize=(10,7))
plt.title('Sentiment Analysis')
plt.xlabel('Count')
plt.ylabel('Sentiments')
df['Analysis'].value_counts().plot(kind='barh' , color=['Limegreen','gold','tomato'])
```

```
plt.show()
```

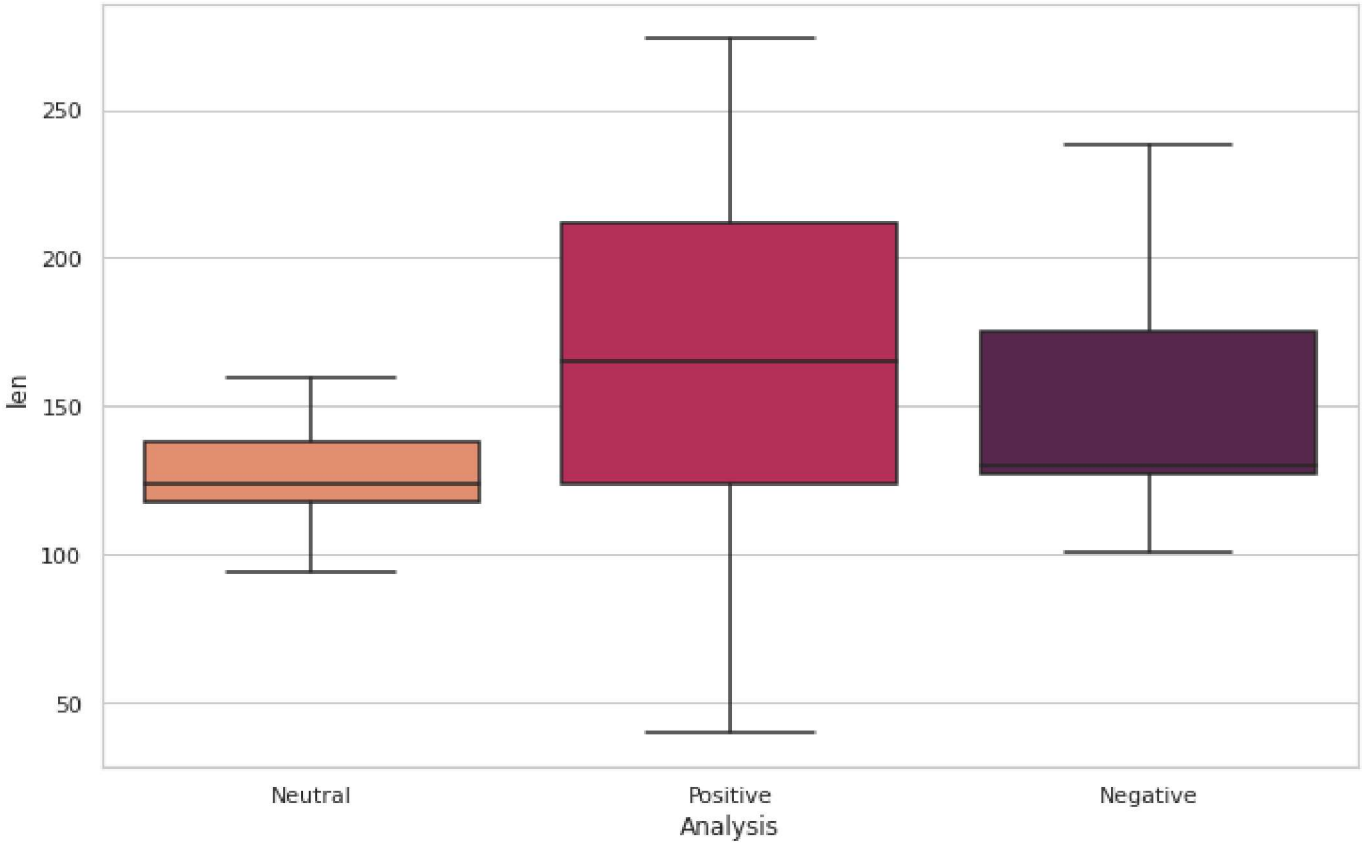


OBSERVATION: *Our Twitter User seems to be a person trying to spread positivity using his tweets as more than 70% of his tweets (as of 26th March 2021) are positive*

SHOWING BOX PLOTS

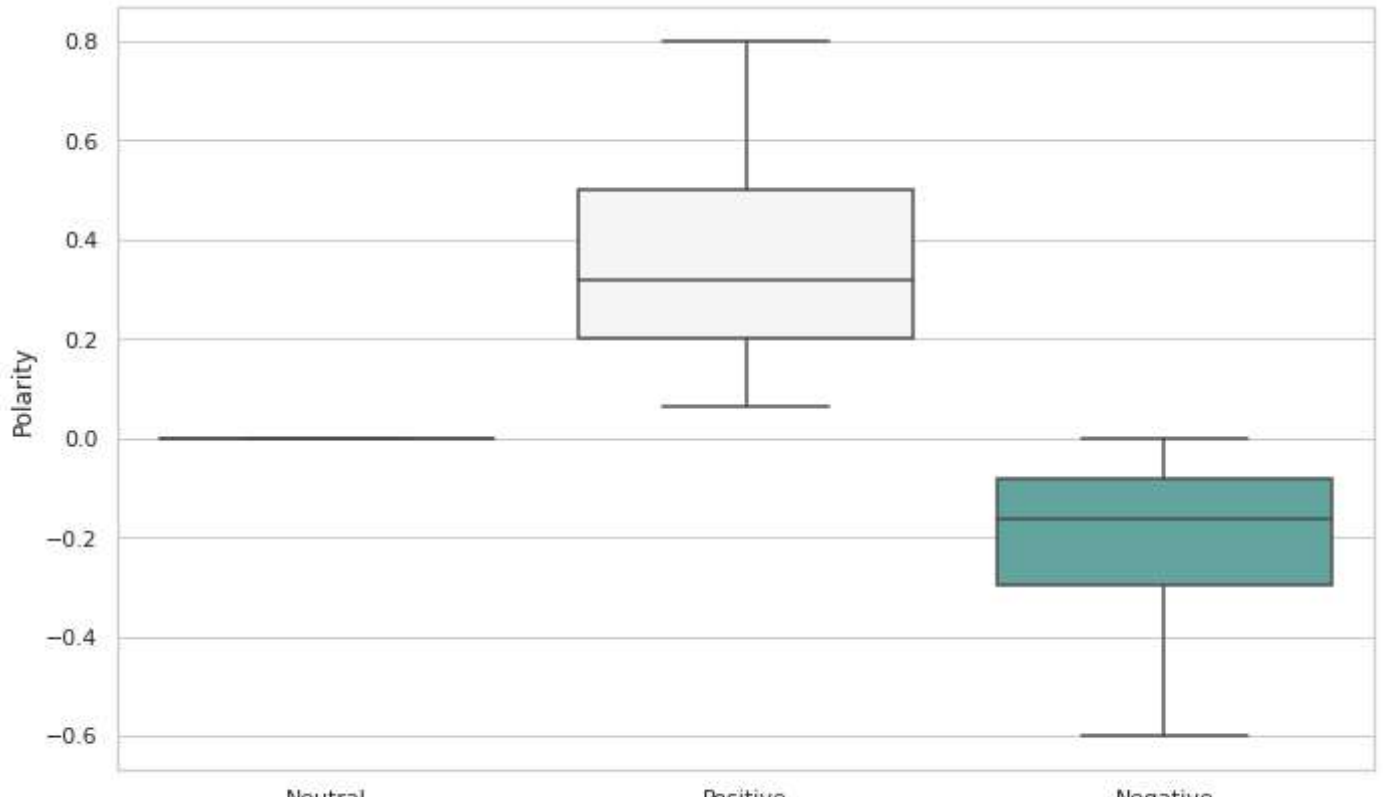
```
#Analysis Labels vs Length
import seaborn as sns
plt.figure(figsize=(10,7))
```

```
sns.set_theme(style="whitegrid")
ax = sns.boxplot(x="Analysis", y="len", data=df, palette="rocket_r", showfliers=False)
```



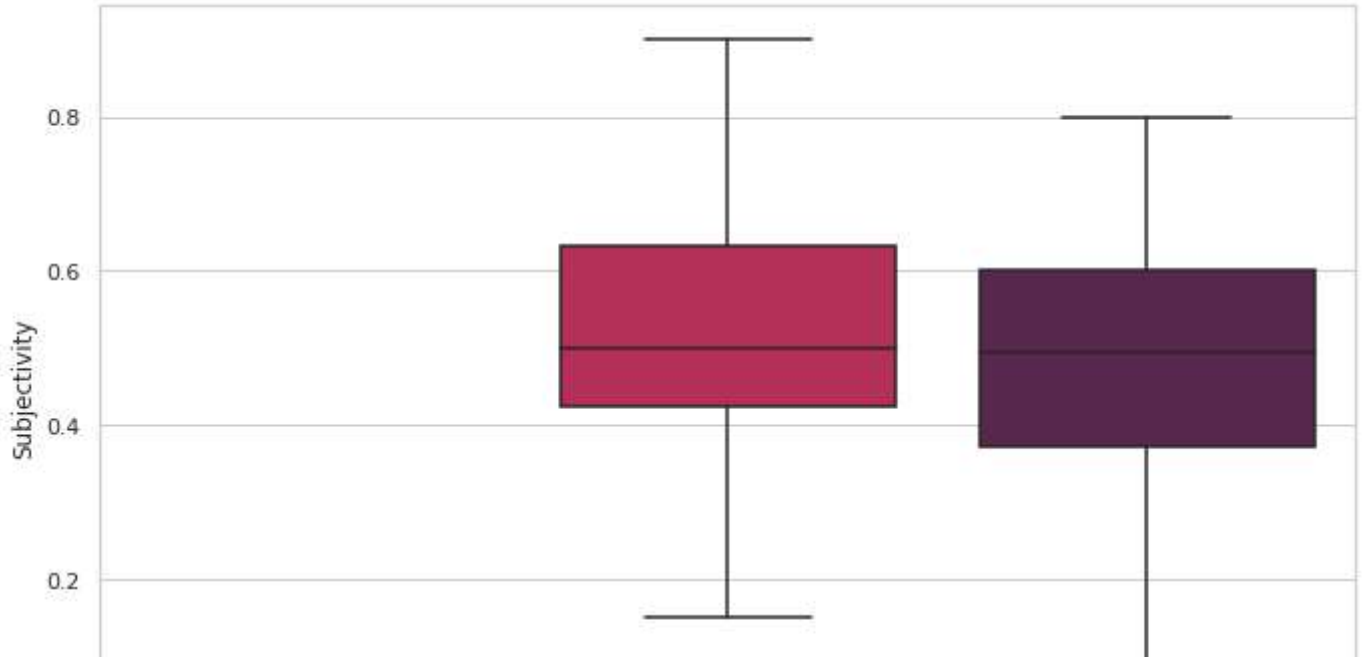
▼ OBSERVATION: *Our Twitter User seems to be a person who promotes Poitivity by longer and more emphasised tweets*

```
#Analysis Labels vs Polarity
sns.set_theme(style="whitegrid")
plt.figure(figsize=(10,7))
ax = sns.boxplot(x="Analysis", y="Polarity",palette="BrBG", data=df)
```



OBSERVATION: *We can clearly see that Neutral texts lie on the edge of 0 whereas positive analysis lie on the positive side and the negative on the negative side*

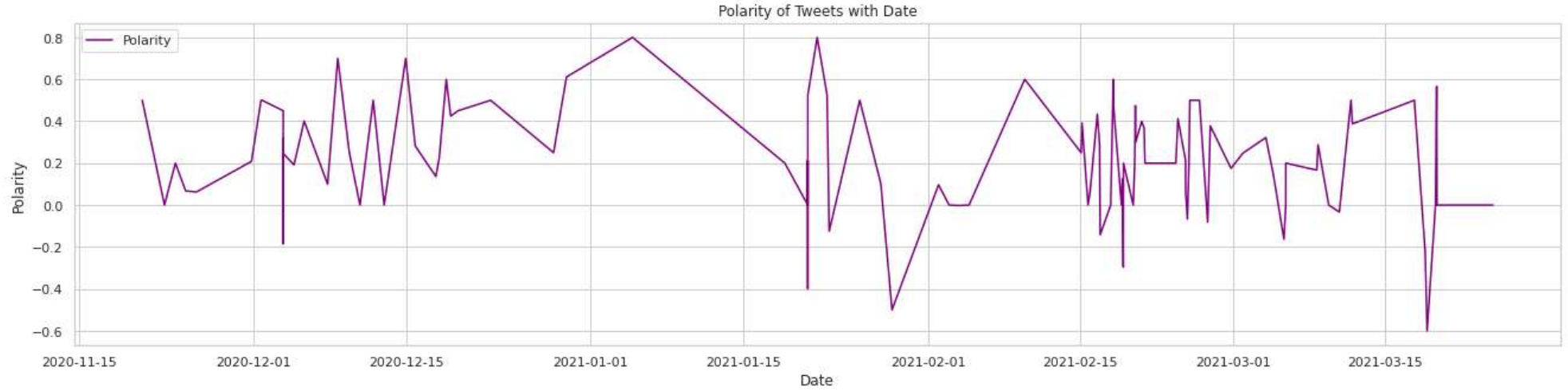
```
#Analysis Labels vs Subjectivity
sns.set_theme(style="whitegrid")
plt.figure(figsize=(10,7))
ax = sns.boxplot(x="Analysis", y="Subjectivity",palette="rocket_r", data=df, showfliers=False)
```



OBSERVATION: *We can clearly see that Neutral texts are not subjective that is they aren't very opioniated*

POLARITY OF TWEETS WITH DATES

```
plt.rc('font', size=12)
fig, ax = plt.subplots(figsize=(20, 5))

# Specify how our lines should look
ax.plot(df.Datetime, df.Polarity, color='purple', label='Polarity')

# Same as above
ax.set_xlabel('Date')
ax.set_ylabel('Polarity')
ax.set_title('Polarity of Tweets with Date')
ax.grid(True)
ax.legend(loc='upper left');
```



OBSERVATION: *The time series graph clearly shows that our Twitter User seems to be a person who promotes Poitivity as polarity of most of the psts seem to be above the 0.0 line on the y axis*

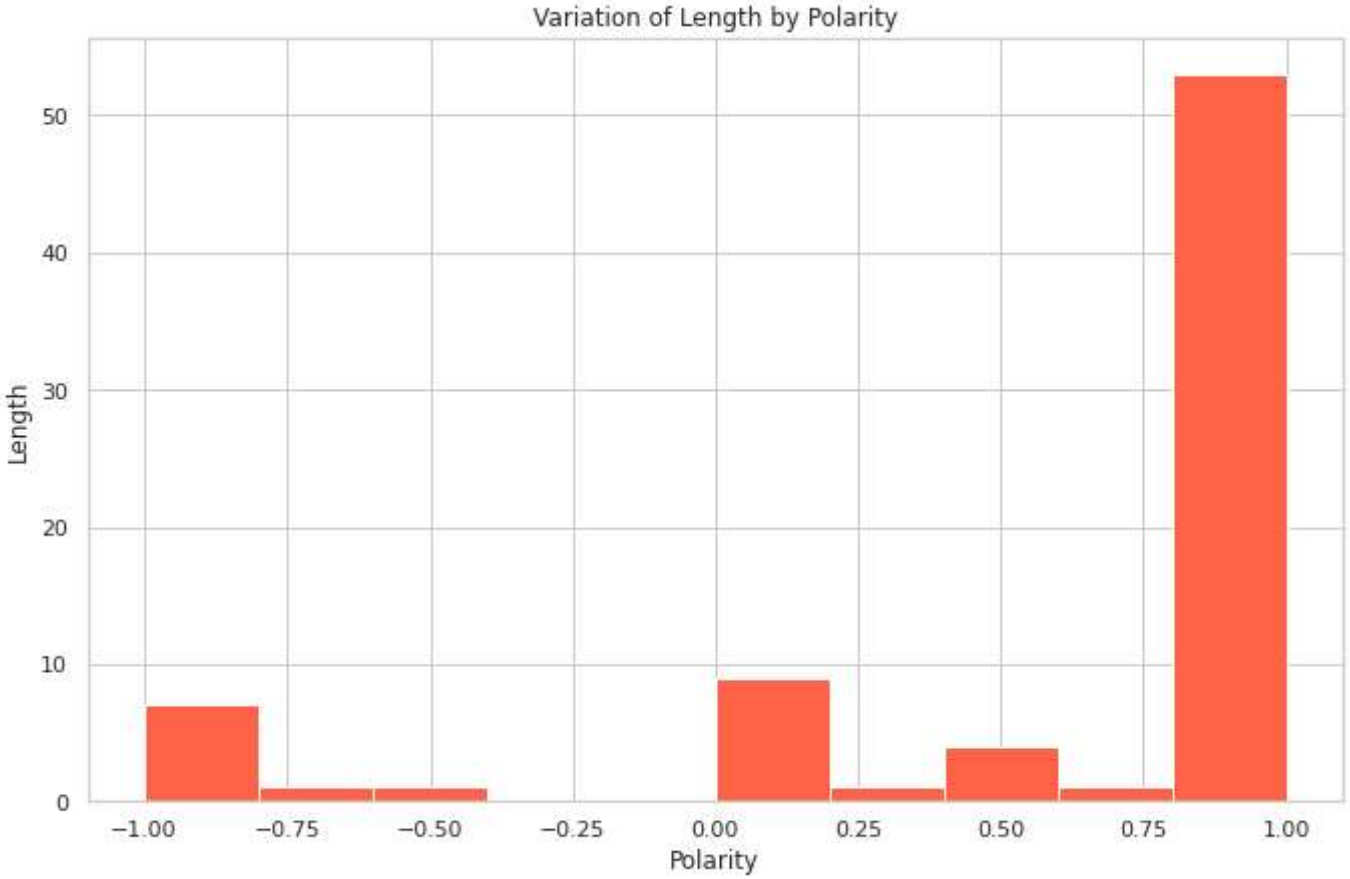
CONVERTING CATEGORICAL VARIABLES INTO NUMERICAL VARIABLES

```
df['label'] = df['Analysis'].replace(['Positive','Neutral','Negative'],['1','0','-1'])
df['label'] = df['label'].astype(str).astype(int)
```

```
df.groupby('label').describe()
```

	Subjectivity								Polarity					
	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%
label														
-1	13.0	0.456899	0.220243	0.075000	0.372619	0.49375	0.600000	0.800000	13.0	-0.216927	0.183814	-0.6000	-0.295833	-0.

```
df.groupby('len').mean()['label'].plot.hist(color = 'tomato', figsize = (10, 7),)
plt.title('Variation of Length by Polarity')
plt.xlabel('Polarity')
plt.ylabel('Length')
plt.show()
```



HEAT MAP SHOWING CORRELATION BETWEEN THE VARIOUS COLUMNS

```
def corr(data):
    correlation = data.corr()
    plt.figure(figsize=(10,7))
    sns.heatmap(correlation, annot=True, cbar=True, cmap="rocket_r")

corr(df)
```

PRINTING POSITIVE TWEETS

```
#print all of the positive tweets

j=1
sortedDF = df.sort_values(by=['Polarity'])
for i in range(0, sortedDF.shape[0]):
  if (sortedDF['Analysis'][i] == 'Positive'):
    print(str(j) + ')'+ sortedDF['Tweets'][i])
    print()
    j = j+1

#This will print the Most Positive Tweets followed by second most positive and so on...
```

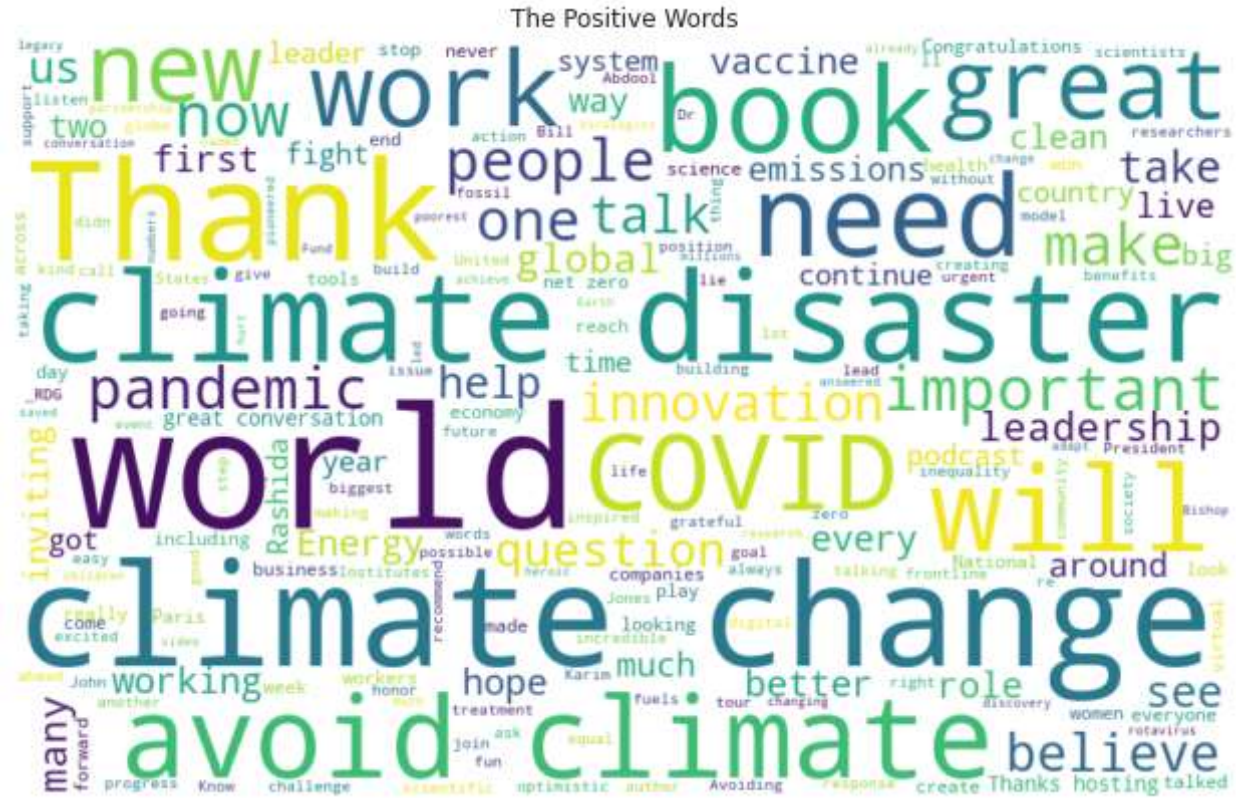
```
46)It’s great to see India’s leadership in scientific innovation and vaccine manufacturing capability as the world works to e

47): The best emergency system is a strong primary health system—powered by community and frontline health workers. Now is t…

48)What does Rashida Jones have on her shopping list? That’s just one of the many questions that didn’t make it into our podc

49)2021 will be better than 2020. Here’s why:

50)Standing up for science has never been more important. Congratulations to Dr. Anthony Fauci and Dr. Salim Abdool Karim on

51)I've known and learned from  for more than 40 years. I'm glad to see  team up with  to mentor and support companies workir

52)This book has nothing to do with viruses or pandemics. But it is surprisingly relevant for these times.  provides a brilli

53)Monoclonal antibodies are one of the most promising treatments we have for COVID-19, thanks in large part to R&amp;amp;D by Ca

54): We’re looking for new ways to advance AlzheimersResearch. If you have an idea for tools, models, or algorithms that will

55)Like many people, I’ve tried to deepen my understanding of systemic racism in recent months. If you’re interested in learr

56)The season finale of our podcast features two incredible people who are using their positions as artists to change the wor

57)Thanks for the great conversation, Carlos.

58)Bill Foege and Viktor Zdanov are phenomenal examples of what it means to harness science for global health. A well-deserve

59)Here are five books that I’d recommend as we wrap up 2020. I hope you find something that helps you—or the book lover in y

60)Are we actually making progress on climate change? Can we really prevent a climate disaster? In this week’s podcast, Rashi
```

```
61)This new quiz from  is a clever way to fight misconceptions and “upgrade your worldview” about the incredible progress the
62)Rashida Jones and I talked to Yuval Noah Harari about COVID conspiracy theories, the role social media plays in spreading
63)I believe we can avoid a climate disaster—if we deploy the clean-energy tools we have now wisely, and if we make big break
64)To reduce duplication, focus the government’s efforts, and get the most innovation out of every dollar of funding, we shou
65)This the most important thing the U.S. can do to lead the world in innovations that will solve climate change.
66)I’m inspired by Quarraisha Abdool Karim and . They are two of the most respected HIV/AIDS researchers in the world. And nc
67): My parents taught me to leave the world better than I found it.
So many people across the country believe in giving bac…
68)I’m a big fan of author _yuval and was excited to talk to him about why humanity is so willing to believe falsehoods and w
69): “I will continue fighting. I will continue supporting other patients. I will never give up.”
Meet Aftab, a multi-drug resi…
70)Today is the 25th anniversary of my first book, The Road Ahead. I made a lot of predictions about technology in the book,
71)This week, Rashida and I ask a big question that has never felt more urgent: is inequality inevitable? We spoke with Compt
72)You can listen to a bonus clip that didn’t make it into our latest episode here:
```

MOST COMMON WORDS IN POSITIVE TWEETS

```
positive_words =' '.join([text for text in df['Tweets'][df['label'] == 1]])

wordcloud = WordCloud(background_color = 'white', width=800, height=500, random_state = 0, max_font_size = 110).generate(positive_wor
plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.title('The Positive Words')
```

```
plt.title('The Positive Words')
plt.show()
```



OBSERVATION: *We can clearly see all the positive tweets as of 26th March 2021 seem to be using words like WORLD, CLIMATE CHANGE, NEED, GREAT, THANK etc*

PRINTING NEGATIVE TWEETS

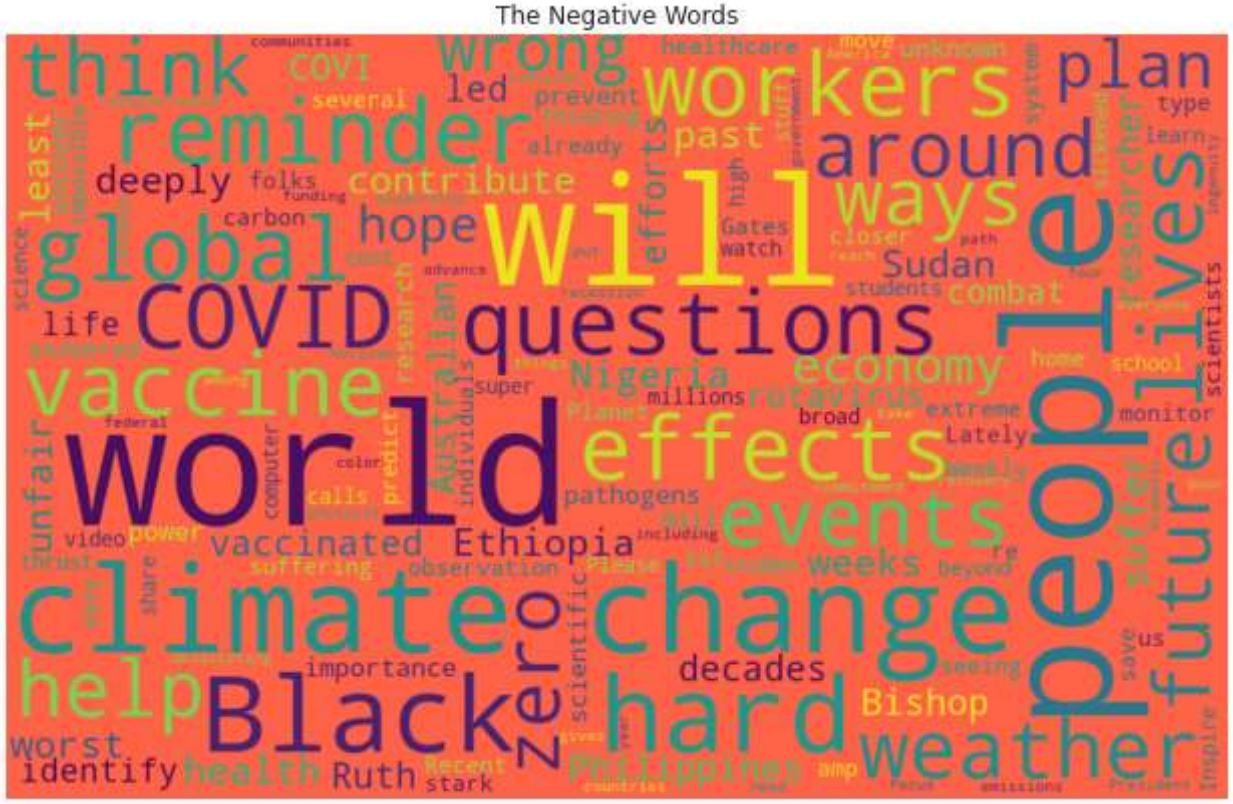
```
#print all of the negative tweets

j=1
sorted_negativeDF = df.sort_values(by=['Polarity'], ascending=False)
```

```
for i in range(0, sorted_negativeDF.shape[0]):
  if (sorted_negativeDF['Analysis'][i] == 'Negative'):
    print(str(j) + ')'+ sorted_negativeDF['Tweets'][i])
    print()
    j = j+1

#This will print the Most Negative Tweets followed by second most negative and so on...
#That is why ascending is False
```

```
1)It’s deeply unfair that the people who contribute the least to climate change will suffer the worst from its effects:

2): Over the past few weeks health workers in Ethiopia ET, Nigeria NG, Sudan SD and the Philippines PH were vaccinated against C

3)For decades, Australian researcher Ruth Bishop led global efforts to identify and combat rotavirus. Her life is a reminder of

4): Black folks have questions about the COVID-19 vaccine. I sat down w/ Black healthcare workers &amp;amp; they answered my questi

5)Recent extreme weather events are a stark reminder that we’re already seeing the effects of climate change here at home and a

6)There are several ways individuals can help move us closer to a zero-carbon future. Here are a few:

7): The Weekly Planet: Lately, Bill Gates has been thinking about what he calls the “hard stuff” of climate change. These hard…

8): "People who think a plan is easy are wrong. People who think a plan is impossible are wrong. It's super hard and very broad

9): Only 3% of Black students learn computer science in high school or beyond. Please watch and share this video. Inspire a stu

10)COVID-19 has cost lives, sickened millions, and thrust the global economy into a devastating recession. But hope is on the h

11)Here are four other ways that America can advance its leadership on climate change this year and put the world on a path to

12)The President’s commitment to reengage with the world gives me hope that the recovery will reach everyone, including communi

13)We need to revolutionize the world’s physical economy—and that will take, among other things, a dramatic infusion of ingenui
```

```
negative_words =' '.join([text for text in df['Tweets'][df['label'] == -1]])

wordcloud = WordCloud(background_color = 'tomato', width=800, height=500, random_state = 0, max_font_size = 110).generate(negative_wo
```

```
plt.figure(figsize=(10, 7))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis('off')
plt.title('The Negative Words')
plt.show()
```

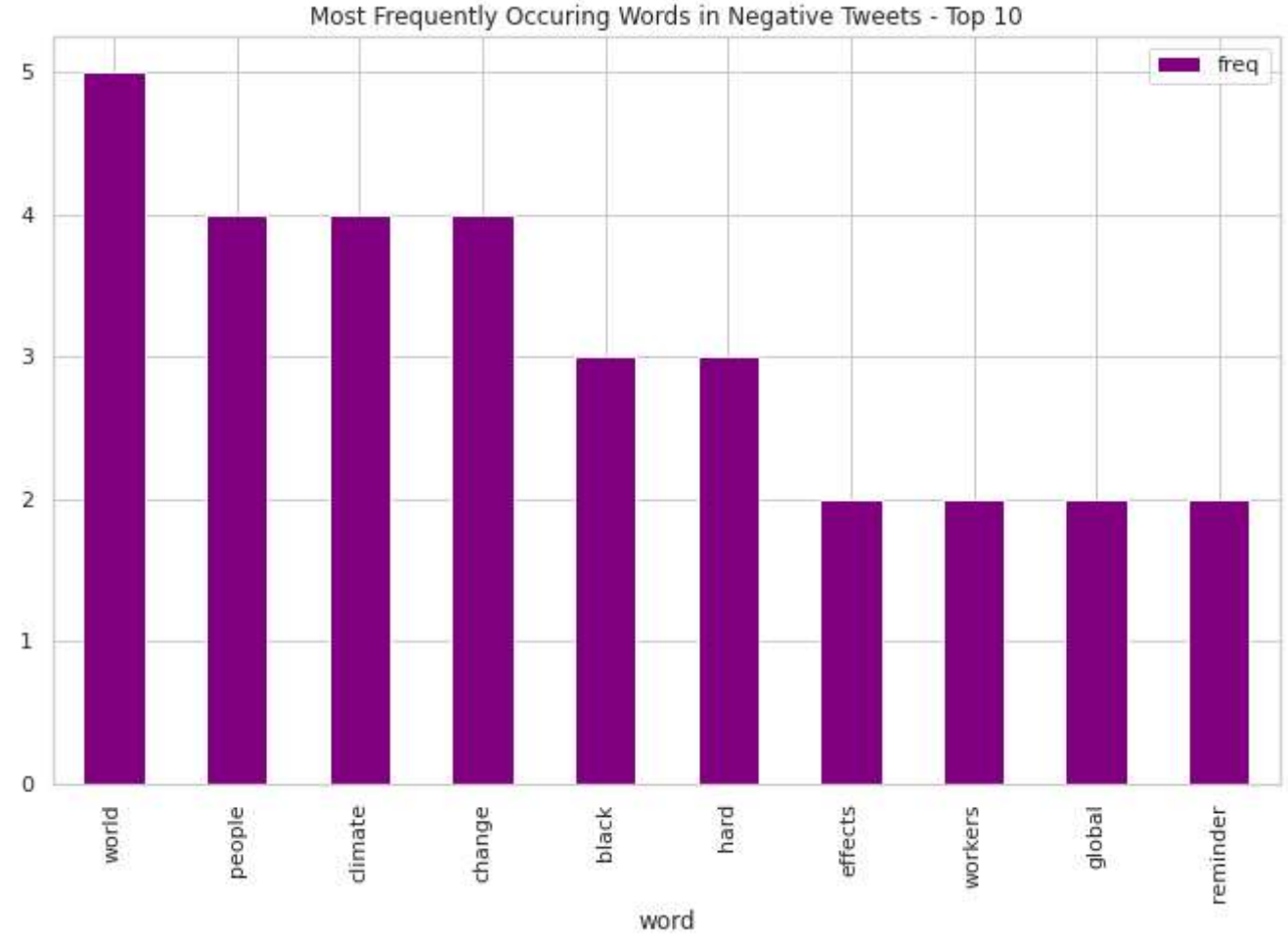


OBSERVATION: *Negative tweets as of 26th March 2021 seem to be using words like WORLD, COVID, CLIMATE, EFFECTS, BLACK,HARD,WRONG etc*

```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(stop_words = 'english')
words = cv.fit_transform(df.Tweets[df['label']==-1])
```

```
sum_words = words.sum(axis=0)

words_freq = [(word, sum_words[0, i]) for word, i in cv.vocabulary_.items()]
words_freq = sorted(words_freq, key = lambda x: x[1], reverse = True)

frequency = pd.DataFrame(words_freq, columns=['word', 'freq'])

frequency.head(10).plot(x='word', y='freq', kind='bar', figsize=(10, 7), color = 'purple')
plt.title("Most Frequently Occuring Words in Negative Tweets - Top 10")
```

```
Text(0.5, 1.0, 'Most Frequently Occuring Words in Negative Tweets - Top 10')
```



FINDING THE PERCENTAGE

```
#Get the percentage of Positive Tweets
ptweets = df[df.Analysis == 'Positive']
ptweets = ptweets['Tweets']

percentage_positive= round(ptweets.shape[0]/ df.shape[0] *100,1)
print('Percentage of Positive Tweets are: '+ str(percentage_positive) + '%')
```

```
     Percentage of Positive Tweets are: 72.0%
```

```
#Get the percentage of Negative Tweets
ptweets = df[df.Analysis == 'Negative']
ptweets = ptweets['Tweets']

percentage_negative= round(ptweets.shape[0]/ df.shape[0] *100,1)
print('Percentage of Neagtive Tweets are: '+ str(percentage_negative) + '%')
```

```
     Percentage of Neagtive Tweets are: 13.0%
```

```
#Get the percentage of Neutral Tweets
ptweets = df[df.Analysis == 'Neutral']
ptweets = ptweets['Tweets']

percentage_negative= round(ptweets.shape[0]/ df.shape[0] *100,1)
print('Percentage of Neutral Tweets are: '+ str(percentage_negative) + '%')
```

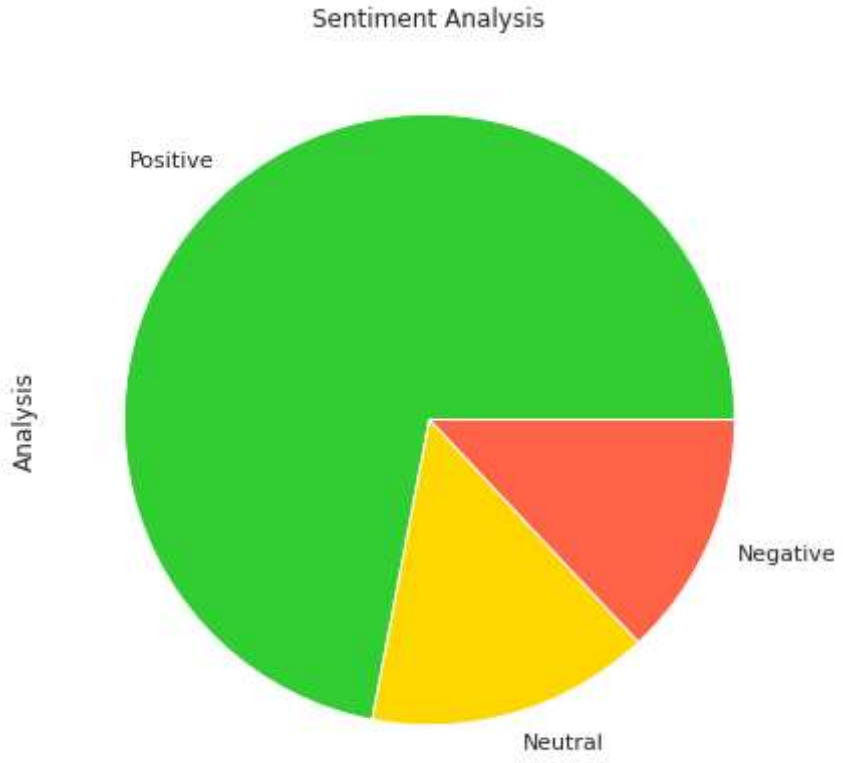
```
     Percentage of Neutral Tweets are: 15.0%
```

PLOTTING SENTIMENT ANAYLSIS RESULTS OF THE LATEST 100 TWEETS

```
#Show the Value Counts
```

```
df['Analysis'].value_counts()

#plot and visualize the counts
plt.figure(figsize=(10, 7))
plt.title('Sentiment Analysis')
colors = ["limegreen", "gold", "tomato"]
df['Analysis'].value_counts().plot(kind='pie', colors=colors)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efe972ef890>
```



K MEANS CLUSTERING

```
#Making a Dataframe with only numerical Columns
df2 = df.drop('Tweets', axis=1)
```

```
df2 = df2.drop('Analysis', axis=1)
df2 = df2.drop('Datetime', axis=1)


ss = StandardScaler()
ss.fit_transform(df2)
```

```
  5.78729018e-01],
[-5.11639438e-01,  1.02016031e-01,  5.60946283e-01,
  5.78729018e-01],
[ 2.86721216e-01,  1.03318527e+00, -2.01077668e+00,
  5.78729018e-01],
[ 1.28467203e+00,  8.46951422e-01, -2.67528227e-01,
  5.78729018e-01],
[ 4.86311379e-01,  7.53834498e-01,  7.50805025e-01,
  5.78729018e-01],
[ 2.14956274e+00,  1.40565297e+00,  8.71624224e-01,
  5.78729018e-01],
[ 3.89367585e-01,  9.17656311e-04,  1.68283885e+00,
  5.78729018e-01],
[ 1.05275613e-01, -3.21242714e-01, -5.43686397e-01,
  5.78729018e-01],
[ 1.19065478e-01,  2.21205693e-01,  1.26860159e+00,
  5.78729018e-01],
[ 1.08508187e+00,  1.77812066e+00, -3.19307884e-01,
  5.78729018e-01],
[-1.70918042e+00, -8.29153208e-01, -5.95466054e-01,
 -8.32805172e-01],
[ 1.86926134e-01,  1.03318527e+00, -2.01077668e+00,
  5.78729018e-01],
[-1.70918042e+00, -8.29153208e-01, -6.64505597e-01,
 -8.32805172e-01],
[-7.11229602e-01,  1.02016031e-01, -4.91906740e-01,
  5.78729018e-01],
[ 6.85901543e-01,  1.77812066e+00,  6.04095997e-02,
  5.78729018e-01],
[-1.11040993e+00, -4.56685512e-01,  8.54364339e-01,
  5.78729018e-01],
[ 1.20201348e+00,  6.64479874e-01,  1.35490102e+00,
  5.78729018e-01],
[-2.78784247e-01, -1.15256792e-01,  9.75183538e-01,
  5.78729018e-01],
```

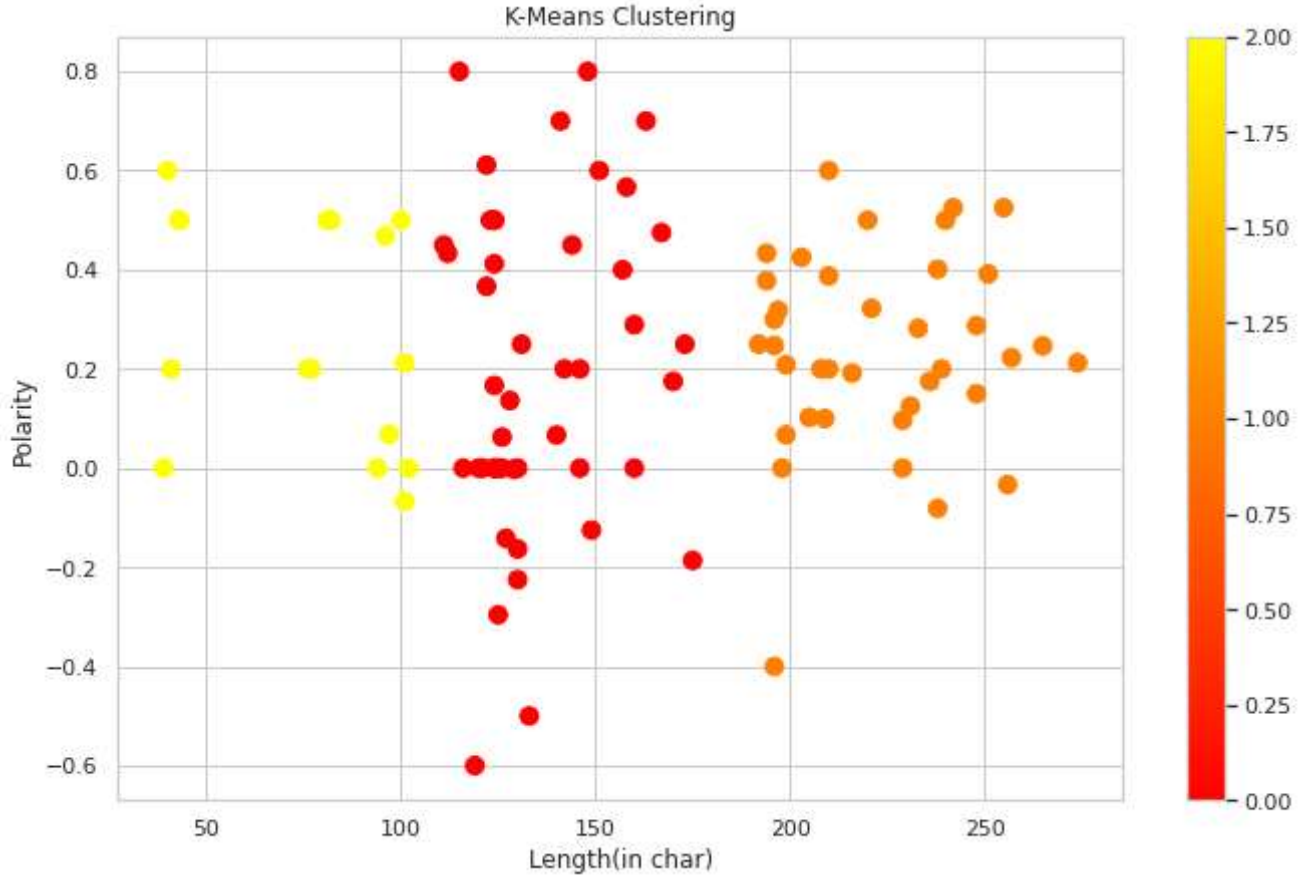
```
       [-3.40486898e-02,  8.87136131e-02,  1.82091793e+00,
         5.78729018e-01],
       [ 1.95998414e-01,  3.55971278e-01,  6.47245711e-01,
         5.78729018e-01],
       [-2.21758486e-01, -1.52235697e+00,  2.67528227e-01,
        -2.24433936e+00],
       [ 1.28467203e+00,  8.46951422e-01, -8.37104453e-01,
         5.78729018e-01],
       [ 2.86721216e-01,  1.03318527e+00,  1.04422308e+00,
         5.78729018e-01],
       [ 2.86721216e-01,  1.03318527e+00, -6.12725940e-01,
         5.78729018e-01],
       [ 4.19781325e-01, -5.31788423e-02,  6.81765482e-01,
         5.78729018e-01],
       [-4.61741897e-01, -5.96360898e-01, -5.78206169e-01,
         5.78729018e-01],
       [-1.37170274e-01, -5.79067755e-01,  6.81765482e-01,
         5.78729018e-01],
       [ 1.66967118e-01, -8.42178169e-02,  8.71624224e-01,
         5.78729018e-01],
       [-1.70918042e+00, -8.29153208e-01, -7.50805025e-01,
        -8.32805172e-01],
       [ 1.88344252e+00,  1.03318527e+00, -1.33764114e+00,
         5.78729018e-01]])
```

```
#K means Clustering
def doKmeans(X, nclust=3):
    model = KMeans(nclust)
    model.fit(X)
    clust_labels = model.predict(X)
    cent = model.cluster_centers_
    return (clust_labels, cent)

clust_labels, cent = doKmeans(df2, 3)
kmeans = pd.DataFrame(clust_labels)

df2.insert((df2.shape[1]),'kmeans',kmeans)
```

PLOTTING THE CLUSTERS OBTAINED USING K MEANS

```
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111)
scatter = ax.scatter(df2['len'],df2['Polarity'],
                     c=kmeans[0],s=80, cmap="autumn")
ax.set_title('K-Means Clustering')
ax.set_xlabel('Length(in char)')
ax.set_ylabel('Polarity')
plt.colorbar(scatter)
```

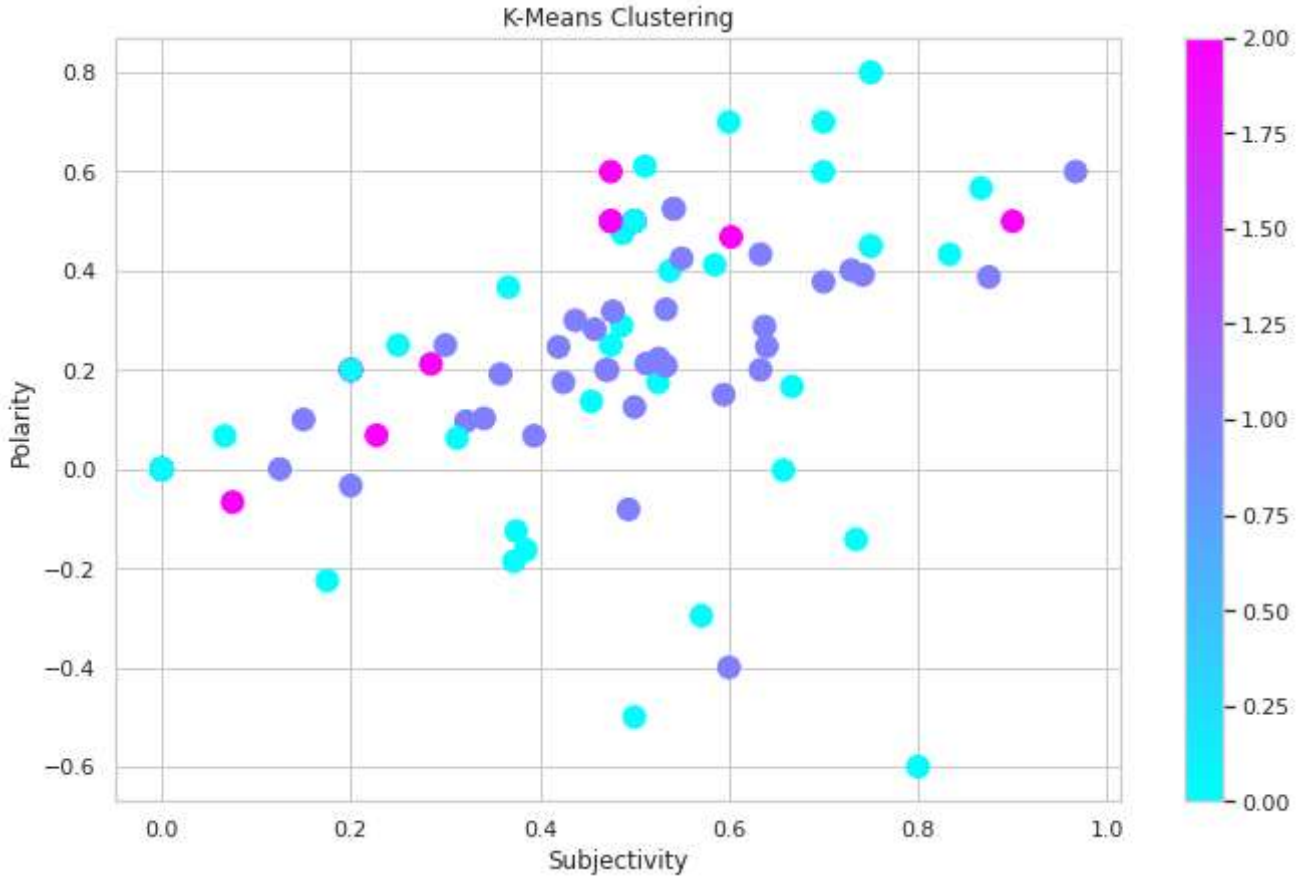
```
<matplotlib.colorbar.Colorbar at 0x7efe97302110>
```




```
#Plot the clusters obtained using k means
```

```
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111)
scatter = ax.scatter(df2['Subjectivity'],df2['Polarity'],
                     c=kmeans[0],s=120, cmap="cool")
ax.set_title('K-Means Clustering')
ax.set_xlabel('Subjectivity')
ax.set_ylabel('Polarity')
plt.colorbar(scatter)
```

```
<matplotlib.colorbar.Colorbar at 0x7efe9729d610>
```



END OF ASSIGNMENT

------ PALAK NATH ------

✓ 0s completed at 12:06 PM