TWITTER SENTIMENT ANALYSIS

Twitter Sentiment Analysis is a data analytics project that involves analyzing a dataset of tweets to determine the sentiment expressed in each tweet—whether it is positive, negative, or neutral. The project aims to gain insights into public opinions, trends, and sentiments shared on Twitter, utilizing data analytics techniques.

Importing required modules :

```
[3]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     import re
     import string
     import nltk
     import warnings
     %matplotlib inline
```

```
[4]: !pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (3.8.1)
Requirement already satisfied: click in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from nltk) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from nltk) (2023.12.25)
Requirement already satisfied: tqdm in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from nltk) (4.66.1)
Requirement already satisfied: colorama in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from click->nltk) (0.4.6)
```

```
[5]: df = pd.read_csv('tweets.csv')
     df.head()
```

Display the given table :

```
[8]:
```

	target	ids	date	flag	users	tweets
0	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
1	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
2	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
3	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....
4	0	1467811372	Mon Apr 06 22:20:00 PDT 2009	NO_QUERY	joy_wolf	@Kwesidei not the whole crew

Data Exploration & Cleaning :

```
df['clean_tweet'] = df['clean_tweet'].str.replace("[^a-zA-Z#]", " ")
df.head()
display(df.head())
```

```
df['clean_tweet'] = np.vectorize(remove_pattern)(df['tweet'],"@[\w]*")
```

```
def remove_pattern(text, pattern):
    r = re.findall(pattern,text)
    for word in r:
        input = re.sub(word, "",text)
    return text
```

```
df['clean_tweet'] = df['clean_tweet'].apply(lambda x: " ".join([w for w in x.split() if len(w)>3]))
df.head()
display(df.head())
```

Exploratory Data Analysis (EDA):

Text Analysis & Preprocessing :

```
# stem the words
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()
tokenized_tweet = tokenized_tweet.apply(lambda sentence: [stemmer.stem(word) for word in sentence])
tokenized_tweet.head()
```

```
# individual words considered as tokens
tokenized_tweet = df['clean_tweet'].apply(lambda x: x.split())
tokenized_tweet.head()
```

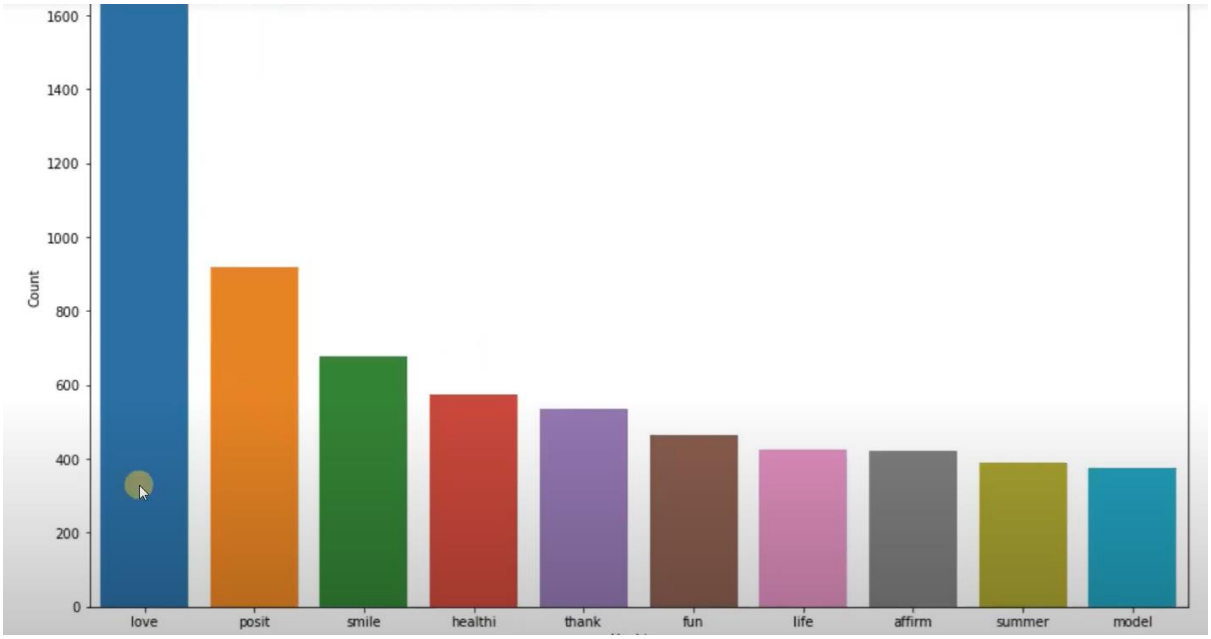
```
# combine words into single sentence
for i in range(len(tokenized_tweet)):
tokenized_tweet[i] = " ".join(tokenized_tweet[i])
df[ (clean_tweet'] = tokenized_tweet
df.head()
```

Word Frequency Analysis:

```
!pip install wordcloud
```

```
Collecting wordcloud
  Downloading wordcloud-1.9.3-cp38-cp38-win_amd64.whl.metadata (3.5 kB)
Requirement already satisfied: numpy>=1.6.1 in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from wordcloud) (1.
Requirement already satisfied: pillow in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from wordcloud) (10.2.0)
Requirement already satisfied: matplotlib in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from wordcloud) (3.7.
Requirement already satisfied: contourpy>=1.0.1 in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from matplotlib
Requirement already satisfied: cycler>=0.10 in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from matplotlib->wo
Requirement already satisfied: fonttools>=4.22.0 in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from matplotli
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from matplotli
Requirement already satisfied: packaging>=20.0 in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from matplotlib-
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from matplotlib
Requirement already satisfied: python-dateutil>=2.7 in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from matplo
Requirement already satisfied: importlib-resources>=3.2.0 in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from
Requirement already satisfied: zipp>=3.1.0 in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from importlib-resou
d) (3.17.0)
Requirement already satisfied: six>=1.5 in c:\users\lenovo\anaconda3\envs\py\lib\site-packages (from python-dateutil>=2
0)
Downloading wordcloud-1.9.3-cp38-cp38-win_amd64.whl (300 kB)
   ---------------------------------------- 0.0/300.7 kB ? eta -:--:--
   ---------------------------------------  297.0/300.7 kB 9.2 MB/s eta 0:00:01
   ---------------------------------------- 300.7/300.7 kB 4.6 MB/s eta 0:00:00
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.3
```

```
# visualize the frequent words
all_words = "".join([sentence for sentence in df['tweets']])
from wordcloud import WordCloud
wordcloud = WordCloud(width=800, height=500, random_state=42, max_font_size=100).generate(all_words)
# plot the graph
plt.figure(figsize= (15,8))
plt.imshow(wordcloud, interpolation= 'bilinear')
plt.axis('off')
plt.show()
```



Sentiment Distribution:

```
# extract hashtags from non-racist/sexist tweets
ht_positive = hashtag_extract(df['clean_tweet'][df['label']==0]
# extract hashtags from racist/sexist tweets
ht_negative = hashtag_extract(df['clean_tweet'][df['label']==1]
```

```
# unnest List
ht_positive = sum (ht_positive, [])
ht_negative = sum(ht_negative, [])
ht_positive[:5]

freq = nltk.FreqDist(ht_positive)
d = pd.DataFrame ({'Hashtag':list(freq.keys()),
                   'Count': list(freq.values())})
```

```
# extract the hashtag
def hashtag_extract(tweets):
    hashtags = []
    # Loop words in the tweet
    for tweet in tweets:
        ht = re.findall(r"#(\w+)",tweet)
        hashtags.append(ht)
     return hashtags
```

Feature Importance:

```
# feature extraction
from sklearn.feature_extraction.text import CountVectorizer
bow_vectorizer = CountVectorizer (max_df=0.90, min_df=2, max_features=1000, stop_word ='english*)
bow = bow_vectorizer.fit_transform(df[ `clean_tweet'J)
```

Sentiment Prediction Model:

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import f1_score, accuracy_score
# training
model I = LogisticRegression()
model.fit(x_train, y_train)
LogisticRegression()
# testing
pred = model.predict(x_test)
f1_score(y_test, predl)
0.49763033175355453
accuracy_score(y_test, ,pred)
0.9469403078463271
```