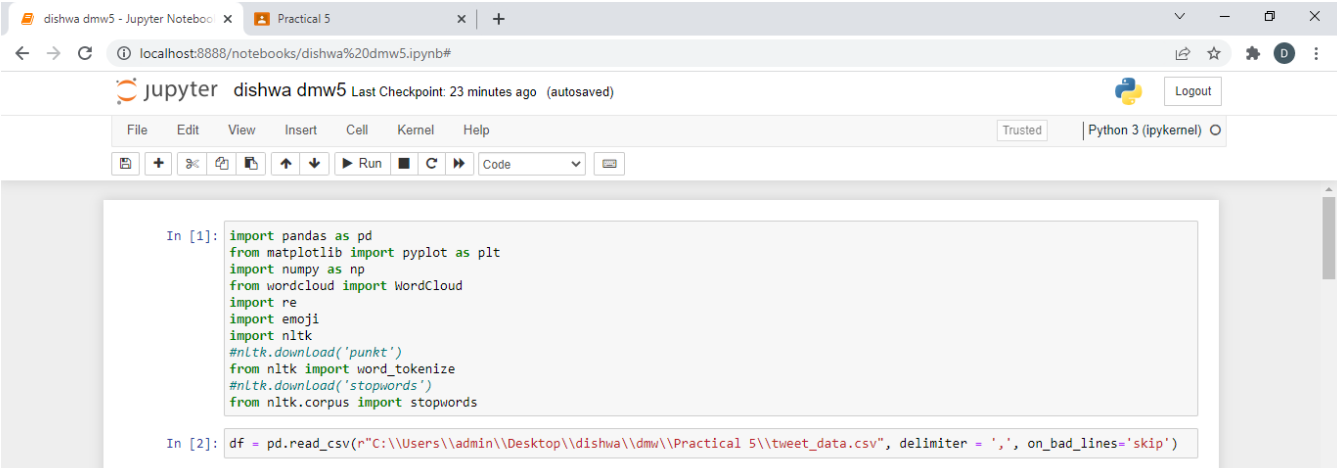
Institute of Computer Technology
B. Tech Computer Science and Engineering
Sub: Data Mining and Warehousing (2CSE60E27)

PRACTICAL 5: TEXT MINING, DATA VISUALIZATION, DATA CLEANING, LANGUAGE PROCESSING

Using the "tweet_data.csv" dataset, answer the following questions.

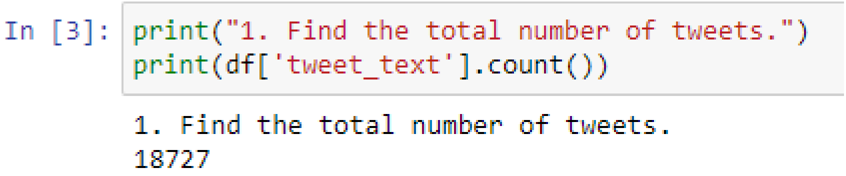

dishwa dmw5 - Jupyter Noteboo
Practical 5
localhost:8888/notebooks/dishwa%20dmw5.ipynb#
jupyter dishwa dmw5 Last Checkpoint: 23 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [1]: import pandas as pd
        from matplotlib import pyplot as plt
        import numpy as np
        from wordcloud import WordCloud
        import re
        import emoji
        import nltk
        #nltk.download('punkt')
        from nltk import word_tokenize
        #nltk.download('stopwords')
        from nltk.corpus import stopwords
```

```
In [2]: df = pd.read_csv(r"C:\\Users\\admin\\Desktop\\dishwa\\dmw\\Practical 5\\tweet_data.csv", delimiter = ',', on_bad_lines='skip')
```

Text Mining

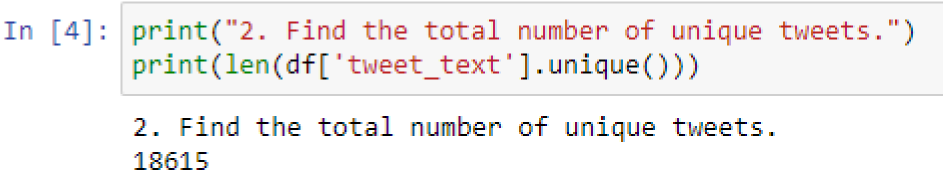
1) Find the total number of tweets.



```
In [3]: print("1. Find the total number of tweets.")
        print(df['tweet_text'].count())

        1. Find the total number of tweets.
        18727
```

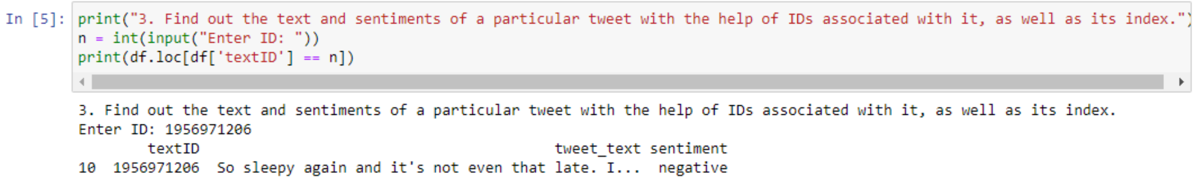
2) Find the total number of unique tweets.



```
In [4]: print("2. Find the total number of unique tweets.")
        print(len(df['tweet_text'].unique()))

        2. Find the total number of unique tweets.
        18615
```

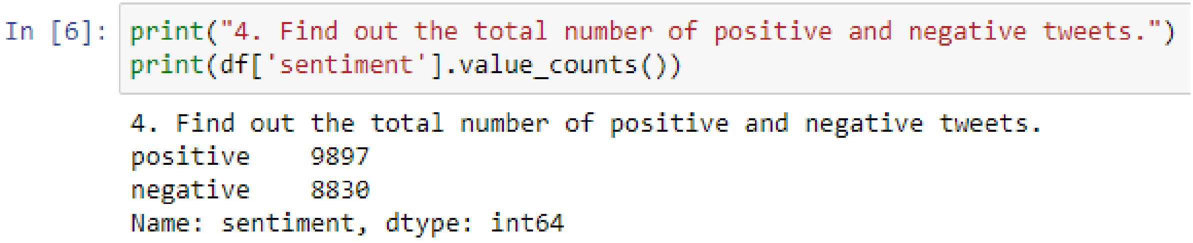
3) Find out the text and sentiments of a particular tweet with the help of IDs associated with it, as well as its index.



```
In [5]: print("3. Find out the text and sentiments of a particular tweet with the help of IDs associated with it, as well as its index.")
        n = int(input("Enter ID: "))
        print(df.loc[df['textID'] == n])
```

```
3. Find out the text and sentiments of a particular tweet with the help of IDs associated with it, as well as its index.
Enter ID: 1956971206
        textID                                          tweet_text sentiment
10  1956971206  So sleepy again and it's not even that late. I...   negative
```

4) Find out the total number of positive and negative tweets

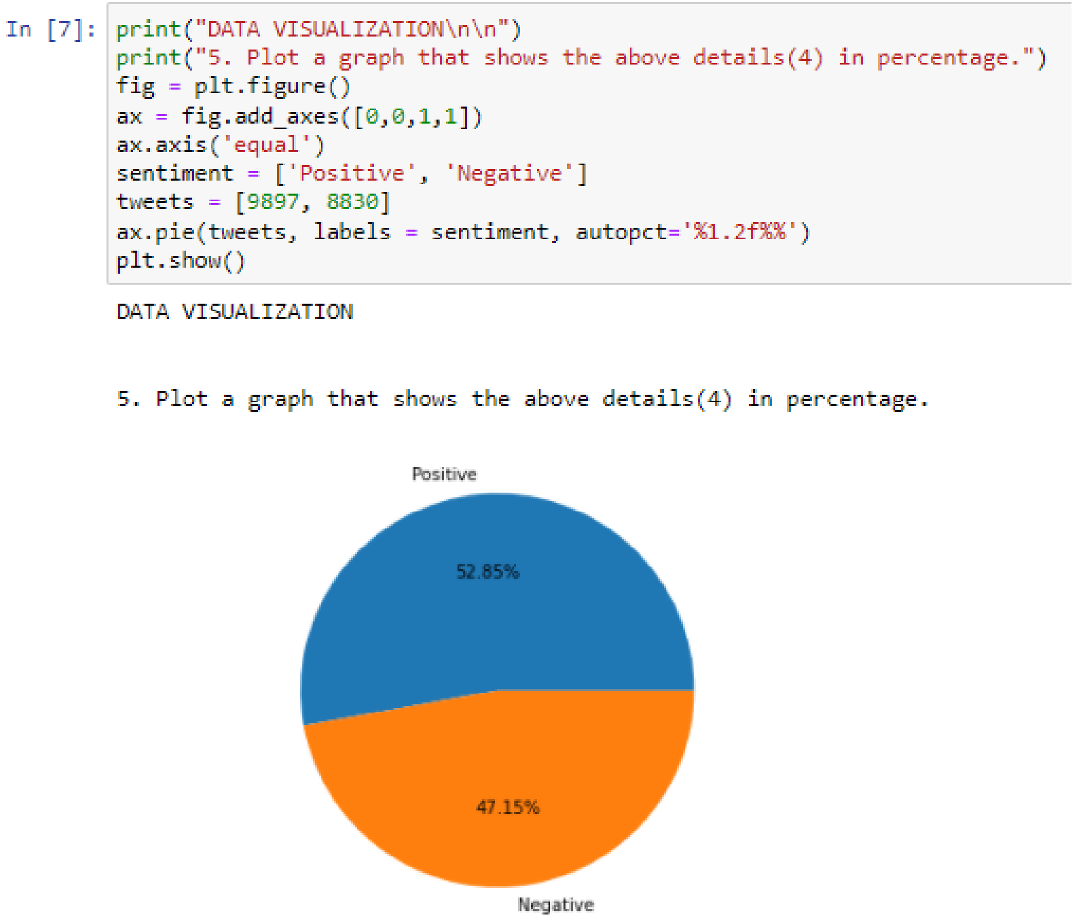


```
In [6]: print("4. Find out the total number of positive and negative tweets.")
        print(df['sentiment'].value_counts())
```

```
4. Find out the total number of positive and negative tweets.
positive    9897
negative    8830
Name: sentiment, dtype: int64
```

Data Visualization

5) Plot a graph that shows the above details(4) in percentage.

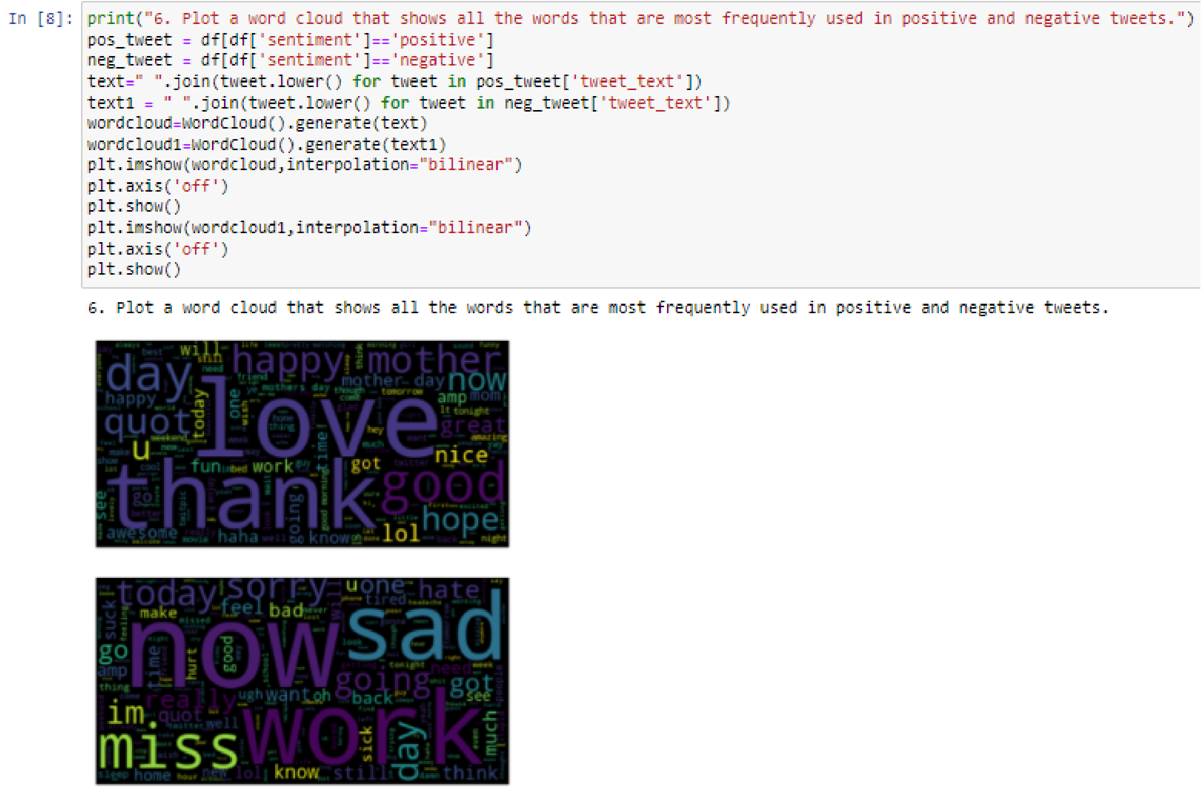


```
In [7]: print("DATA VISUALIZATION\n\n")
        print("5. Plot a graph that shows the above details(4) in percentage.")
        fig = plt.figure()
        ax = fig.add_axes([0,0,1,1])
        ax.axis('equal')
        sentiment = ['Positive', 'Negative']
        tweets = [9897, 8830]
        ax.pie(tweets, labels = sentiment, autopct='%1.2f%%')
        plt.show()
```

```
DATA VISUALIZATION


5. Plot a graph that shows the above details(4) in percentage.
```

6) Plot a word cloud that shows all the words that are most frequently used in positive and negative tweets.

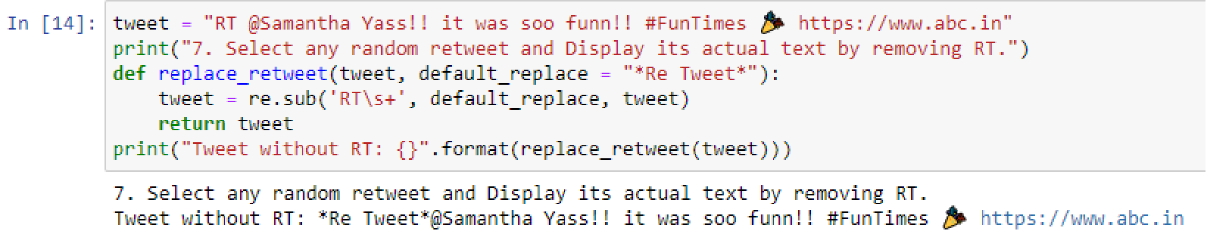


```
In [8]: print("6. Plot a word cloud that shows all the words that are most frequently used in positive and negative tweets.")
        pos_tweet = df[df['sentiment']=='positive']
        neg_tweet = df[df['sentiment']=='negative']
        text=" ".join(tweet.lower() for tweet in pos_tweet['tweet_text'])
        text1 = " ".join(tweet.lower() for tweet in neg_tweet['tweet_text'])
        wordcloud=WordCloud().generate(text)
        wordcloud1=WordCloud().generate(text1)
        plt.imshow(wordcloud,interpolation="bilinear")
        plt.axis('off')
        plt.show()
        plt.imshow(wordcloud1,interpolation="bilinear")
        plt.axis('off')
        plt.show()
```

```
6. Plot a word cloud that shows all the words that are most frequently used in positive and negative tweets.
```

Data Cleaning

7) Select any random retweet and Display its actual text by removing RT



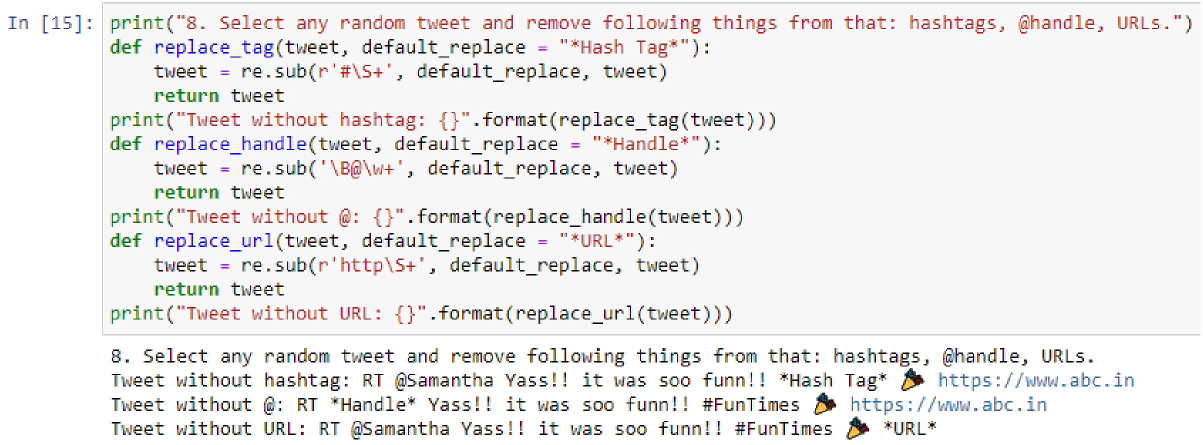
```
In [14]: tweet = "RT @Samantha Yass!! it was soo funn!! #FunTimes 🎉 https://www.abc.in"
         print("7. Select any random retweet and Display its actual text by removing RT.")
         def replace_retweet(tweet, default_replace = "*Re Tweet*"):
             tweet = re.sub('RT\s+', default_replace, tweet)
             return tweet
         print("Tweet without RT: {}".format(replace_retweet(tweet)))
```

```
7. Select any random retweet and Display its actual text by removing RT.
Tweet without RT: *Re Tweet*@Samantha Yass!! it was soo funn!! #FunTimes 🎉 https://www.abc.in
```

8) Select any random tweet and remove following things from that:

- hashtags
- @handle
- URLs

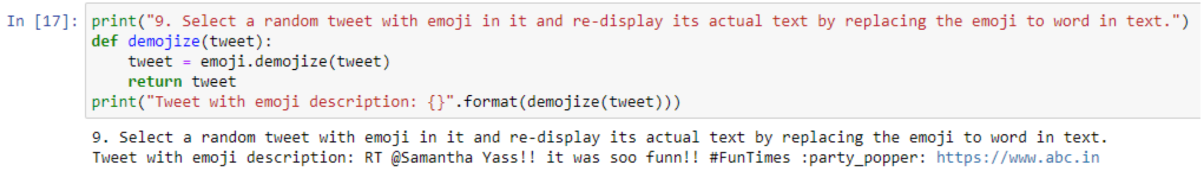
After removal, display actual tweet text.



```
In [15]: print("8. Select any random tweet and remove following things from that: hashtags, @handle, URLs.")
         def replace_tag(tweet, default_replace = "*Hash Tag*"):
             tweet = re.sub(r'#\S+', default_replace, tweet)
             return tweet
         print("Tweet without hashtag: {}".format(replace_tag(tweet)))
         def replace_handle(tweet, default_replace = "*Handle*"):
             tweet = re.sub('\B@\w+', default_replace, tweet)
             return tweet
         print("Tweet without @: {}".format(replace_handle(tweet)))
         def replace_url(tweet, default_replace = "*URL*"):
             tweet = re.sub(r'http\S+', default_replace, tweet)
             return tweet
         print("Tweet without URL: {}".format(replace_url(tweet)))

         8. Select any random tweet and remove following things from that: hashtags, @handle, URLs.
         Tweet without hashtag: RT @Samantha Yass!! it was soo funn!! *Hash Tag* 🎉 https://www.abc.in
         Tweet without @: RT *Handle* Yass!! it was soo funn!! #FunTimes 🎉 https://www.abc.in
         Tweet without URL: RT @Samantha Yass!! it was soo funn!! #FunTimes 🎉 *URL*
```

9) Select a random tweet with emoji in it and re-display its actual text by replacing the emoji to word in text

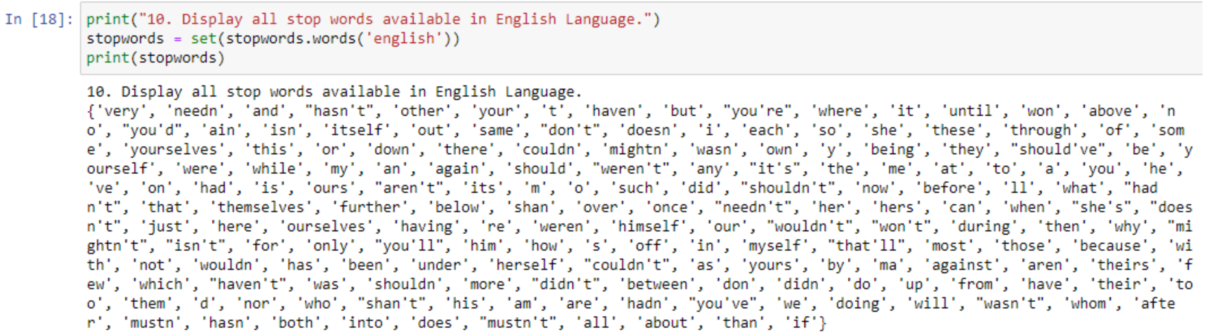


```
In [17]: print("9. Select a random tweet with emoji in it and re-display its actual text by replacing the emoji to word in text.")
         def demojize(tweet):
             tweet = emoji.demojize(tweet)
             return tweet
         print("Tweet with emoji description: {}".format(demojize(tweet)))

         9. Select a random tweet with emoji in it and re-display its actual text by replacing the emoji to word in text.
         Tweet with emoji description: RT @Samantha Yass!! it was soo funn!! #FunTimes :party_popper: https://www.abc.in
```

Language Processing

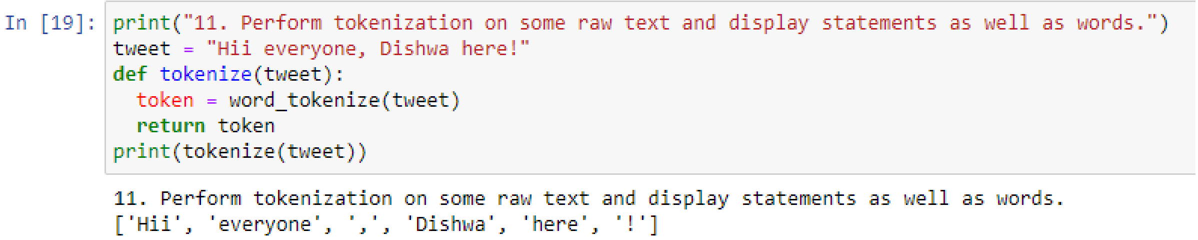
10) Display all stop words available in English Language



```
In [18]: print("10. Display all stop words available in English Language.")
         stopwords = set(stopwords.words('english'))
         print(stopwords)

         10. Display all stop words available in English Language.
         {'very', 'needn', 'and', "hasn't", 'other', 'your', 't', 'haven', 'but', "you're", 'where', 'it', 'until', 'won', 'above', 'n
         o', "you'd", 'ain', 'isn', 'itself', 'out', 'same', "don't", 'doesn', 'i', 'each', 'so', 'she', 'these', 'through', 'of', 'som
         e', 'yourselves', 'this', 'or', 'down', 'there', 'couldn', 'mightn', 'wasn', 'own', 'y', 'being', 'they', "should've", 'be', 'y
         ourself', 'were', 'while', 'my', 'an', 'again', 'should', "weren't", 'any', "it's", 'the', 'me', 'at', 'to', 'a', 'you', 'he',
         've', 'on', 'had', 'is', 'ours', "aren't", 'its', 'm', 'o', 'such', 'did', "shouldn't", 'now', 'before', 'll', 'what', "had
         n't", 'that', 'themselves', 'further', 'below', 'shan', 'over', 'once', "needn't", 'her', 'hers', 'can', 'when', "she's", "does
         n't", 'just', 'here', 'ourselves', 'having', 're', 'weren', 'himself', 'our', "wouldn't", "won't", 'during', 'then', 'why', "mi
         ghtn't", "isn't", 'for', 'only', "you'll", 'him', 'how', 's', 'off', 'in', 'myself', "that'll", 'most', 'those', 'because', 'wi
         th', 'not', 'wouldn', 'has', 'been', 'under', 'herself', "couldn't", 'as', 'yours', 'by', 'ma', 'against', 'aren', 'theirs', 'f
         ew', 'which', "haven't", 'was', 'shouldn', 'more', "didn't", 'between', 'don', 'didn', 'do', 'up', 'from', 'have', 'their', 'to
         o', 'them', 'd', 'nor', 'who', "shan't", 'his', 'am', 'are', 'hadn', "you've", 'we', 'doing', 'will', "wasn't", 'whom', 'afte
         r', 'mustn', 'hasn', 'both', 'into', 'does', "mustn't", 'all', 'about', 'than', 'if'}
```

11) Perform tokenization on some raw text and display statements as well as words



```
In [19]: print("11. Perform tokenization on some raw text and display statements as well as words.")
         tweet = "Hii everyone, Dishwa here!"
         def tokenize(tweet):
           token = word_tokenize(tweet)
           return token
         print(tokenize(tweet))

         11. Perform tokenization on some raw text and display statements as well as words.
         ['Hii', 'everyone', ',', 'Dishwa', 'here', '!']
```

Code:

```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
from wordcloud import WordCloud
import re
import emoji
import nltk
#nltk.download('punkt')
from nltk import word_tokenize
#nltk.download('stopwords')
from nltk.corpus import stopwords
df = pd.read_csv(r"C:\\Users\\admin\\Desktop\\dishwa\\dmw\\Practical 5\\tweet_data.csv", delimiter = ',', on_bad_lines='skip')

print("TEXT MINING\n")
print("\n1. Find the total number of tweets.")
print(df['tweet_text'].count())
print("2. Find the total number of unique tweets.")
print(len(df['tweet_text'].unique()))

print("3. Find out the text and sentiments of a particular tweet with the help of IDs associated with it, as well as its index.")
n = int(input("Enter ID: "))
print(df.loc[df['textID'] == n])

print("4. Find out the total number of positive and negative tweets.")
print(df['sentiment'].value_counts())

print("DATA VISUALIZATION")
print("\n5. Plot a graph that shows the above details(4) in percentage.")
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
ax.axis('equal')
sentiment = ['Positive', 'Negative']
tweets = [9897, 8830]
ax.pie(tweets, labels = sentiment, autopct='%1.2f%%')
plt.show()

print("6. Plot a word cloud that shows all the words that are most frequently used in positive and negative tweets.")
pos_tweet = df[df['sentiment']=='positive']
neg_tweet = df[df['sentiment']=='negative']
text=" ".join(tweet.lower() for tweet in pos_tweet['tweet_text'])
text1 = " ".join(tweet.lower() for tweet in neg_tweet['tweet_text'])
wordcloud=WordCloud().generate(text)
wordcloud1=WordCloud().generate(text1)
plt.imshow(wordcloud,interpolation="bilinear")
plt.axis('off')
plt.show()
plt.imshow(wordcloud1,interpolation="bilinear")
plt.axis('off')
plt.show()

print("DATA CLEANING")
tweet = "RT @Samantha Yass!! it was soo funn!! #FunTimes 🎉 https://www.abc.in"
print("7. Select any random retweet and Display its actual text by removing RT.")
def replace_retweet(tweet, default_replace = "*Re Tweet*"):
    tweet = re.sub('RT\s+', default_replace, tweet)
    return tweet
print("Tweet without RT: {}".format(replace_retweet(tweet)))

print("8. Select any random tweet and remove following things from that: hashtags,
@handle, URLs.")
def replace_tag(tweet, default_replace = "*Hash Tag*"):
    tweet = re.sub(r'#\S+', default_replace, tweet)
    return tweet
print("Tweet without hashtag: {}".format(replace_tag(tweet)))
def replace_handle(tweet, default_replace = "*Handle*"):
    tweet = re.sub('\B@\w+', default_replace, tweet)
    return tweet
print("Tweet without @: {}".format(replace_handle(tweet)))
def replace_url(tweet, default_replace = "*URL*"):
    tweet = re.sub(r'http\S+', default_replace, tweet)
    return tweet
print("Tweet without URL: {}".format(replace_url(tweet)))

print("9. Select a random tweet with emoji in it and re-display its actual text by replacing
the emoji to word in text.")
def demojize(tweet):
    tweet = emoji.demojize(tweet)
    return tweet
print("Tweet with emoji description: {}".format(demojize(tweet)))

print("LANGUAGE PROCESSING\n10. Display all stop words available in English
Language.")
stopwords = set(stopwords.words('english'))
print(stopwords)

print("11. Perform tokenization on some raw text and display statements as well as
words.")
tweet = "Hii everyone, Dishwa here!"
def tokenize(tweet):
  token = word_tokenize(tweet)
  return token
print(tokenize(tweet))
```