

▼ Importing the required libraries

```
import re
import pandas as pd
import numpy as np
#import emoji
from collections import Counter
import matplotlib.pyplot as plt
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

Double-click (or enter) to edit

```
!pip install emoji
import emoji
```

```
Collecting emoji
  Downloading emoji-1.6.1.tar.gz (170 kB)
    |████████████████████████████████████████| 170 kB 21.4 MB/s
Building wheels for collected packages: emoji
  Building wheel for emoji (setup.py) ... done
  Created wheel for emoji: filename=emoji-1.6.1-py3-none-any.whl size=169314 sha256=4
  Stored in directory: /root/.cache/pip/wheels/ea/5f/d3/03d313ddb3c2a1a427bb4690f1621
Successfully built emoji
Installing collected packages: emoji
Successfully installed emoji-1.6.1
```

```
!pip install nltk
import nltk.sentiment.vader
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.7/dist-packages (3.2.5)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from nl
/usr/local/lib/python3.7/dist-packages/nltk/twitter/__init__.py:20: UserWarning: The
  warnings.warn("The twython library has not been installed. ")
```

```
import nltk
nltk.download('all')
```

```
[nltk_data] Downloading collection 'all'
[nltk_data] |
[nltk_data] | Downloading package abc to /root/nltk_data...
[nltk_data] |   Unzipping corpora/abc.zip.
[nltk_data] | Downloading package alpino to /root/nltk_data...
[nltk_data] |   Unzipping corpora/alpino.zip.
[nltk_data] | Downloading package averaged_perceptron_tagger to
[nltk_data] |   /root/nltk_data...
[nltk_data] |   Unzipping taggers/averaged_perceptron_tagger.zip.
[nltk_data] | Downloading package averaged_perceptron_tagger_ru to
```

```
[nltk_data] | /root/nltk_data...
[nltk_data] | Unzipping
[nltk_data] | taggers/averaged_perceptron_tagger_ru.zip.
[nltk_data] | Downloading package basque_grammars to
[nltk_data] | /root/nltk_data...
[nltk_data] | Unzipping grammars/basque_grammars.zip.
[nltk_data] | Downloading package biocreative_ppi to
[nltk_data] | /root/nltk_data...
[nltk_data] | Unzipping corpora/biocreative_ppi.zip.
[nltk_data] | Downloading package bllip_wsj_no_aux to
[nltk_data] | /root/nltk_data...
[nltk_data] | Unzipping models/bllip_wsj_no_aux.zip.
[nltk_data] | Downloading package book_grammars to
[nltk_data] | /root/nltk_data...
[nltk_data] | Unzipping grammars/book_grammars.zip.
[nltk_data] | Downloading package brown to /root/nltk_data...
[nltk_data] | Unzipping corpora/brown.zip.
[nltk_data] | Downloading package brown_tei to /root/nltk_data...
[nltk_data] | Unzipping corpora/brown_tei.zip.
[nltk_data] | Downloading package cess_cat to /root/nltk_data...
[nltk_data] | Unzipping corpora/cess_cat.zip.
[nltk_data] | Downloading package cess_esp to /root/nltk_data...
[nltk_data] | Unzipping corpora/cess_esp.zip.
[nltk_data] | Downloading package chat80 to /root/nltk_data...
[nltk_data] | Unzipping corpora/chat80.zip.
[nltk_data] | Downloading package city_database to
[nltk_data] | /root/nltk_data...
[nltk_data] | Unzipping corpora/city_database.zip.
[nltk_data] | Downloading package cmudict to /root/nltk_data...
[nltk_data] | Unzipping corpora/cmudict.zip.
[nltk_data] | Downloading package comparative_sentences to
[nltk_data] | /root/nltk_data...
[nltk_data] | Unzipping corpora/comparative_sentences.zip.
[nltk_data] | Downloading package comtrans to /root/nltk_data...
[nltk_data] | Downloading package conll2000 to /root/nltk_data...
[nltk_data] | Unzipping corpora/conll2000.zip.
[nltk_data] | Downloading package conll2002 to /root/nltk_data...
[nltk_data] | Unzipping corpora/conll2002.zip.
[nltk_data] | Downloading package conll2007 to /root/nltk_data...
[nltk_data] | Downloading package crubadan to /root/nltk_data...
[nltk_data] | Unzipping corpora/crubadan.zip.
[nltk_data] | Downloading package dependency_treebank to
[nltk_data] | /root/nltk_data...
[nltk_data] | Unzipping corpora/dependency_treebank.zip.
[nltk_data] | Downloading package dolch to /root/nltk_data...
[nltk_data] | Unzipping corpora/dolch.zip.
[nltk_data] | Downloading package europarl_raw to
[nltk_data] | /root/nltk_data...
```

```
pip install vaderSentiment
```

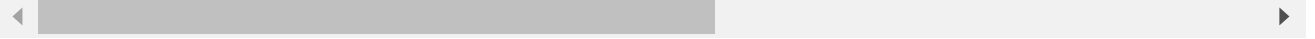
Collecting vaderSentiment

Downloading vaderSentiment-3.3.2-py2.py3-none-any.whl (125 kB)

125 kB 31.4 MB/s

```
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in /usr/local,
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages
```

Installing collected packages: vaderSentiment
 Successfully installed vaderSentiment-3.3.2



Definitions Of Functions like date,time,author

```
# Extract the Date time
def date_time(s):
    pattern='^([0-9]+)(\\/)([0-9]+)(\\/)([0-9]+), ([0-9]+):([0-9]+)[ ]?(AM|PM|am|pm)? - '
    result=re.match(pattern, s)
    if result:
        return True
    return False

# Extract contacts
def find_contact(s):
    s=s.split(":")
    if len(s)==2:
        return True
    else:
        return False

# Extract Message
def getMessage(line):
    splitline=line.split(' - ')
    datetime= splitline[0];
    date, time= datetime.split(', ')
    message=" ".join(splitline[1:])

    if find_contact(message):
        splitmessage=message.split(": ")
        author=splitmessage[0]
        message=splitmessage[1]
    else:
        author=None
    return date, time, author, message

data=[]
conversation='chat.txt'
with open(conversation, encoding="utf-8") as fp:
    fp.readline()
    messageBuffer=[]
    date, time, author= None, None, None
    while True:
        line=fp.readline()
        if not line:
            break
        line=line.strip()
        if date_time(line):
            if len(messageBuffer) >0:
                data.append([date, time, author, ''.join(messageBuffer)])
            messageBuffer.clear()
```

```

        date, time, author, message=getMessage(line)
        messageBuffer.append(message)
    else:
        messageBuffer.append(line)

```

```

df=pd.DataFrame(data, columns=["Date", "Time", "contact", "Message"])
df['Date']=pd.to_datetime(df['Date'])

```

```

data=df.dropna()
from nltk.sentiment.vader import SentimentIntensityAnalyzer
sentiments=SentimentIntensityAnalyzer()
data["positive"]=[sentiments.polarity_scores(i)["pos"] for i in data["Message"]]
data["negative"]=[sentiments.polarity_scores(i)["neg"] for i in data["Message"]]
data["neutral"]=[sentiments.polarity_scores(i)["neu"] for i in data["Message"]]

```

```
data.head()
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:7: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html>
import sys
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:8: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html>

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:9: SettingWithCopyWarning: A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: <https://pandas.pydata.org/pandas-docs/stable/10min.html>
if __name__ == '__main__':

	Date	Time	contact	Message	positive	negative	neutral
0	2019-08-13	6:04 PM	Nikhil Jadhav	Hii	0.000	0.0	1.000
1	2019-08-13	6:05 PM	👉👉Rutuja Sanjay Kadam👉👉	Hii	0.000	0.0	1.000
2	2019-08-13	6:05 PM	Nikhil Jadhav	College Suru zal ka	0.000	0.0	1.000
3	2019-08-13	6:05 PM	👉👉Rutuja Saniav	..	0.000	0.0	1.000

```

x=sum(data["positive"])
y=sum(data["negative"])
z=sum(data["neutral"])

```

```

def score(a,b,c):
    if (a>b) and (a>c):
        print("Positive ")
    if (b>a) and (b>c):
        print("Negative ")

```

```

print('negative',
      if (c>a) and (c>b):
        print("Neutal")

```

```
score(x,y,z)
```



Neutal

```
df.contact.unique()
```

```
array(['Nikhil Jadhav', '👁️👁️Rutuja Sanjay Kadam👉❤️', None], dtype=object)
```

```
media_messages = df[df['Message'] == 'Oke'].shape[0]
print(media_messages)
```

```
0
```

```
def split_count(text):
```

```

    emoji_list = []
    data = regex.findall(r'\X', text)
    for word in data:
        if any(char in emoji.UNICODE_EMOJI for char in word):
            emoji_list.append(word)

```

```
    return emoji_list
```

```

import regex
df["emoji"] = df["Message"].apply(split_count)
emojis = sum(df['emoji'].str.len())
df.head(50)

```

	Date	Time	contact	Message	emoji
0	2019-08-13	6:04 PM	Nikhil Jadhav	Hii	👋
1	2019-08-13	6:05 PM	👋 ✨ Rutuja Sanjay Kadam 🙌❤️	Hii	👋
2	2019-08-13	6:05 PM	Nikhil Jadhav	College Suru zal ka	👋
3	2019-08-13	6:05 PM	👋 ✨ Rutuja Sanjay Kadam 🙌❤️	Ha	👋
4	2019-08-13	6:05 PM	👋 ✨ Rutuja Sanjay Kadam 🙌❤️	Ha num konakde asto	👋
5	2019-08-13	6:05 PM	👋 ✨ Rutuja Sanjay Kadam 🙌❤️	Nikhil ki nikita	👋
6	2019-08-13	6:05 PM	Nikhil Jadhav	Maza ahe	👋
7	2019-08-13	6:05 PM	Nikhil Jadhav	Nikhil	👋
8	2019-08-13	6:05 PM	👋 ✨ Rutuja Sanjay Kadam 🙌❤️	Okk	👋
9	2019-08-13	6:05 PM	👋 ✨ Rutuja Sanjay Kadam 🙌❤️	Tuje jle ka start	👋
10	2019-08-13	6:05 PM	Nikhil Jadhav	Ho	👋
11	2019-08-13	6:06 PM	Nikhil Jadhav	College cha induction chalu ahe ki abhiyas	👋
12	2019-08-13	6:07 PM	👋 ✨ Rutuja Sanjay Kadam 🙌❤️	Ajun inductionch	👋
13	2019-08-13	6:07 PM	👋 ✨ Rutuja Sanjay Kadam 🙌❤️	Tumche	👋
14	2019-08-13	6:07 PM	Nikhil Jadhav	Kiti tarki paryant	👋
15	2019-08-13	6:08 PM	👋 ✨ Rutuja Sanjay Kadam 🙌❤️	3 4 days	👋
16	2019-08-13	6:08 PM	👋 ✨ Rutuja Sanjay Kadam 🙌❤️	Tumche kiti divs	👋
17	2019-08-13	6:08 PM	Nikhil Jadhav	17aug paryant	👋
18	2019-08-13	6:09 PM	👋 ✨ Rutuja Sanjay Kadam 🙌❤️	Okk	👋
19	2019-08-13	6:09 PM	👋 ✨ Rutuja Sanjay Kadam 🙌❤️	Timing ky ahe clq	👋

There are 5921 words in all the messages.

