Dataset Extraction

Importing the required libraries

In [1]:

```
import re
import pandas as pd
```

Checking each line if it starts with date and time to identify each unique message in the text file

In [2]:

```
def rawToDf(file, key):
#     Converts raw .txt file into a Data Frame

    split_formats = {
        '12hr' : '\d{1,2}/\d{1,2}/\d{2,4},\s\d{1,2}:\d{2}\s[APap][mM]\s-\s',
        '24hr' : '\d{1,2}/\d{1,2}/\d{2,4},\s\d{1,2}:\d{2}\s-\s',
        'custom' : ''
    }
    datetime_formats = {
        '12hr' : '%d/%m/%Y, %I:%M %p - ',
        '24hr' : '%d/%m/%Y, %H:%M - ',
        'custom': ''
    }

    with open(file, 'r', encoding='utf-8') as raw_data:
        # print(raw_data.read())
        raw_string = ' '.join(raw_data.read().split('\n')) # converting the list sp
        user_msg = re.split(split_formats[key], raw_string) [1:] # splits at all th
        date_time = re.findall(split_formats[key], raw_string) # finds all the date

        df = pd.DataFrame({'date_time': date_time, 'user_msg': user_msg}) # exporti

    # converting date-time pattern which is of type String to type datetime,
    # format is to be specified for the whole string where the placeholders are ext
    df['date_time'] = pd.to_datetime(df['date_time'], format=datetime_formats[key])

    # split user and msg
    usernames = []
    msgs = []
    for i in df['user_msg']:
        a = re.split('([\w\W]+?):\s', i) # lazy pattern match to first {user_name}:
        if(a[1:]): # user typed messages
            usernames.append(a[1])
            msgs.append(a[2])
        else: # other notifications in the group(eg: someone was added, some left .
            usernames.append("group_notification")
            msgs.append(a[0])


    # creating new columns
    df['user'] = usernames
    df['message'] = msgs

    # dropping the old user_msg col.
    df.drop('user_msg', axis=1, inplace=True)

    return df
```

In [3]:

```
df = rawToDf('chat.txt', '12hr')
```

In [4]:

```
df.head()
```

Out[4]:

	date_time	user	message
0	2020-01-26 16:19:00	group_notification	Messages and calls are end-to-end encrypted. N...
1	2020-01-24 20:25:00	group_notification	Tanay Kamath (TSEC, CS) created group "CODERS👦...
2	2020-01-26 16:19:00	group_notification	You joined using this group's invite link
3	2020-01-26 16:20:00	group_notification	+91 99871 38558 joined using this group's invi...
4	2020-01-26 16:20:00	group_notification	+91 91680 38866 joined using this group's invi...

In [5]:

```
# checking out number of unique authors of the messages
df['user'].unique()
```

Out[5]:

```
array(['group_notification', '+91 96536 93868',
       'Dheeraj Lalwani (TSEC, CS)', '+91 99201 75875', '+91 95949 0
8570',
       '+91 79778 76844', '+91 90499 38860', 'Tanay Kamath (TSEC, C
S)',
       'Saket (TSEC, CS)', '+91 77568 95072', 'Rohit Pathak (TSEC, C
S)',
       '+91 75078 05454', 'Darshan Rander (TSEC, IT)', '+91 79774 68
083',
       '+91 70394 60876', '+91 96191 55044', '+91 90678 93300',
       'Mohit Varma (TSEC, CS)', '+91 79770 56210',
       'Chirag Sharma (TSEC, CS)', 'Vivek Iyer (TSEC, Biomed)',
       'Tushar Nankani', '+91 81696 22410', '+91 89764 07509',
       '+91 78758 66747', 'Ankit (TSEC, CS)', '+91 86556 33169',
       '+91 76663 28147', '+91 88284 70904', '+91 97698 67348',
       'Vivek (TSEC, CS)', 'Hardik Raheja (TSEC, CS)', '+91 91680 38
866',
       'Pranay Thakur (TSEC, CS)', 'Mittul Dasani (TSEC, CS)',
       'Kartik Soneji (TSEC, CS)', '+91 77180 43697', '+91 99676 844
79',
       'Shreya (TSEC, IT)', '+91 96190 16721', '+91 89833 85127',
       '+91 82080 02653', '+91 99675 58551', '+91 90822 59476',
       'Prithvi Rohira (TSEC, CS)', '+91 90820 98830',
       'Mohammed (TSEC, EXTC)', '+91 96992 89993', '+91 83690 2169
3',
       '+91 75064 86714', 'Pratik K (TSEC CS, SE)',
       'Farhan Irani (TSEC IT, SE)', '+91 77000 27264',
       'Harsh Kapadia (TSEC IT, SE)', 'Saurav Upoor (TSEC CS, SE)',
       '+91 77180 82108', '+91 86559 19035', '+91 77150 51136',
       '+91 91671 28174', '+91 84335 18102', '+91 84529 62233',
       '+91 81080 96759', '+91 77384 72938', '+91 93243 92133',
       '+91 97681 67131', '+91 98206 01141', '+91 84540 03063',
       '+91 99693 94098', '+91 91363 39446', '+91 98192 22032',
       '+91 88305 26885', '+91 70208 31915', '+91 98702 02065',
       '+91 88282 22720', '+91 97027 35002', '+91 87796 52381',
       '+91 97739 65140', '+91 97571 15289', 'Rishab Saini (TSEC CS,
TE)',
       '+91 94208 78848', '+91 93598 18687', '+91 73043 57388',
       '+91 98331 51331', '+91 80979 84068', '+91 77158 99478',
       '+91 79776 23387', '+91 99697 55118', '+91 95119 48511',
       '+91 98337 61116', '+91 82916 21138', '+91 88889 97733',
       '+91 97697 60869', '+91 99672 39663', '+91 87796 70896',
       '+91 98191 73361', '+91 70219 80066', '+91 81696 11905',
       '+91 72762 35231', '+91 79775 35465', '+91 97027 04646',
       '+91 70450 40641', '+91 99204 26955', '+91 99696 99151',
       '+91 98333 66146', '+91 95940 62134', '+91 77189 86205',
       '+91 97694 89970', '+91 99302 21772', '+91 77109 79055',
       '+91 96648 44643', '+91 98337 47258', 'Keyul Jain (TSEC, C
S)',
```

```
       '+91 98198 16330', '+91 88798 05171', '+91 92842 87810',
       '+91 72495 29889', '+91 91677 97590',
       'Trushant Narwani (TSEC, CS)', '+91 86528 77025',
       '+91 77383 38799', 'Shubham Chettiar (TSEC CS, TE)',
       '+91 86059 72817', '+91 83292 66084', '+91 82080 03744',
       '+91 98670 44401', '+91 77098 73262', 'Sahil A (TSEC, CS-B)',
       '+91 96194 00980', '+91 99304 97064', '+91 77699 70908',
       '+91 98337 26449', '+91 97847 88658', '+91 82916 40581',
       '+91 91670 43943', '+91 94044 50783', '+91 90821 58843',
       '+91 97022 69539', '+91 73036 41107', '+91 88795 52797',
       'Akash Khatri (TSEC, CS)', '+91 91525 25452', '+91 79778 0398
5',
       '+91 91725 67828', '+91 98206 14506', '+91 70218 25025',
       '+91 94200 70678', '+91 99203 34360', '+91 96374 40537',
       '+91 98199 01072', '+91 91673 86883', '+91 73032 50500',
       '+91 91362 39673', '+91 98501 32687', 'Kritanjali',
       '+91 98709 38217'], dtype=object)
```

In [6]:

```
# checking out random 10 samples from the dataset
df.sample(10)
```

Out[6]:

	date_time	user	message
1585	2020-02-21 22:53:00	Tanay Kamath (TSEC, CS)	Sorry
8131	2020-06-27 21:06:00	Tanay Kamath (TSEC, CS)	If anyone here wants to team up with me and @9...
2532	2020-03-01 10:49:00	+91 79770 56210	jay_3124
6341	2020-05-16 21:40:00	Darshan Rander (TSEC, IT)	CSS ig🤔
4087	2020-03-27 13:45:00	+91 99693 94098	Ubuntu or windows? I need some help with windo...
4728	2020-04-12 18:50:00	+91 97681 67131	<Media omitted>
7857	2020-06-19 02:34:00	Kartik Soneji (TSEC, CS)	USE DOCKER.
10783	2020-08-26 17:05:00	Dheeraj Lalwani (TSEC, CS)	Sahi bola
2141	2020-02-24 23:09:00	Tushar Nankani	<Media omitted>
4390	2020-04-05 09:36:00	+91 78758 66747	

In [7]:

```
# checking for null data
df.isna().sum()
```

Out[7]:

```
date_time    0
user         0
message      0
dtype: int64
```

loading the cleaned dataset into the csv file

In [10]:

```
df.to_csv('Whatsapp_Chat_Table.csv')
```

In [11]:

```
df.head()
```

Out[11]:

	date_time	user	message
0	2020-01-26 16:19:00	group_notification	Messages and calls are end-to-end encrypted. N...
1	2020-01-24 20:25:00	group_notification	Tanay Kamath (TSEC, CS) created group "CODERS🧑...
2	2020-01-26 16:19:00	group_notification	You joined using this group's invite link
3	2020-01-26 16:20:00	group_notification	+91 99871 38558 joined using this group's invi...
4	2020-01-26 16:20:00	group_notification	+91 91680 38866 joined using this group's invi...

In [12]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13655 entries, 0 to 13654
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype         
---  ------     --------------  -----         
 0   date_time  13655 non-null  datetime64[ns]
 1   user       13655 non-null  object        
 2   message    13655 non-null  object        
dtypes: datetime64[ns](1), object(2)
memory usage: 320.2+ KB
```

In [13]:

```
row, col = df.shape
```

In [14]:

```
f"No of Rows :{row} and Columns = {col}"
```

Out[14]:

```
'No of Rows :13655 and Columns = 3'
```

In [17]:

```
df['date_time'].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 13655 entries, 0 to 13654
Series name: date_time
Non-Null Count  Dtype         
--------------  -----         
13655 non-null  datetime64[ns]
dtypes: datetime64[ns](1)
memory usage: 106.8 KB
```