Group 15 TM names :

- Ahmed : 23PGAI0120
- Akash Deshwani: 23PGAI0035
- Harshada Suresh Jadhav: 23PGAI0101
- Rohan Mehta: 23PGAI0001

Installing the required packages

In [1]:
```
!pip install bs4
!pip install numpy
!pip install pandas
!pip install requests
```

```
Requirement already satisfied: bs4 in c:\users\coola\anaconda3\lib\site-packages (0.
0.1)
Requirement already satisfied: beautifulsoup4 in c:\users\coola\anaconda3\lib\site-pa
ckages (from bs4) (4.11.1)
Requirement already satisfied: soupsieve>1.2 in c:\users\coola\anaconda3\lib\site-pac
kages (from beautifulsoup4->bs4) (2.3.1)
Requirement already satisfied: numpy in c:\users\coola\anaconda3\lib\site-packages
(1.21.5)
Requirement already satisfied: pandas in c:\users\coola\anaconda3\lib\site-packages
(1.4.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\coola\anaconda3\lib\site-pack
ages (from pandas) (2021.3)
Requirement already satisfied: numpy>=1.18.5 in c:\users\coola\anaconda3\lib\site-pac
kages (from pandas) (1.21.5)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\coola\anaconda3\lib
\site-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\coola\anaconda3\lib\site-packages
(from python-dateutil>=2.8.1->pandas) (1.16.0)
Requirement already satisfied: requests in c:\users\coola\anaconda3\lib\site-packages
(2.27.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\coola\anaconda3\lib
\site-packages (from requests) (1.26.9)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\users\coola\anaconda3
\lib\site-packages (from requests) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\coola\anaconda3\lib\sit
e-packages (from requests) (2021.10.8)
Requirement already satisfied: idna<4,>=2.5 in c:\users\coola\anaconda3\lib\site-pack
ages (from requests) (3.3)
```

Importing the required packages

In [2]:
```
import json
import numpy as np
import pandas as pd
import re
import requests
from bs4 import BeautifulSoup
import csv
```

Importing the dataset from Json file

In [3]:
```
def read_file():
    a = "gg2015.json"
    df_original = pd.read_json(a)
    print(len(df_original))

    # Creating a copy of the dataframe for manipulating
    df = df_original.copy()

    # Creating a list of text for finding hosts
    text_host = df['text'].tolist()
    return(text_host)
```

In [4]:
```
# df_test = read_file()
# print(type(df_test))
# print(df_test[:5])
```

Preprocessing the data and converting it into a required list

In [5]:
```
# Preprocessing data
def preprocessing(data):
    processed_data = []
    remove_list = ['think', 'thinking', 'should', 'would', 'maybe', 'could']
    for i in data:
        for j in remove_list:
            if j not in i:
                processed_data.append(i)
    remove_words_list = ['golden', 'globe', 'globes']
    for i in range(len(processed_data)):
        for j in remove_words_list:
            processed_data[i] = re.sub(j, '', processed_data[i], flags = re.IGNORECASE
    return processed_data
```

In [6]:
```
# df_1 = preprocessing(df_test)
# print(df_1[:5])
# print(len(df_1))
```

Scraping the Top 1000 Actors and Actresses from the IMDB website

In [7]:
```
def scrape_actors():
    actorNames = []
    for single_page in range(1, 11):
        URL = f"https://www.imdb.com/list/ls058011111/?sort=list_order,asc&mode=detail
        r = requests.get(URL)
        soup = BeautifulSoup(r.text, 'html.parser')
        actorName = soup.find_all('h3',{'class':'lister-item-header'})
        actorName = [i.find('a').text.strip() for i in actorName]
        actorNames.extend(actorName)
    return actorNames
```

In [8]:
```
# scrape_list = scrape_actors()
# print(scrape_list[:5])
# print(len(scrape_list))
```

Extracting Hosts name from the Tweet text

In [9]:
```
# Finding Hosts
def get_hosts(data, actorNames):
    match = ''
    result={}
    for i in range(len(data)):
        found = re.search('(H|h)ost', data[i])
        if found:
            patt = re.compile(r'[A-Z][a-z]+ [A-Z][a-z]+')
            matches = patt.findall(data[i])
            for match in matches:
                if match in actorNames:
                    if match in result:
                        result[match] += 1
                    else:
                        result[match] = 1
    sorted_hosts = sorted(result.items(), key=lambda x:x[1], reverse=True)
    return(sorted_hosts[0:2])
```

In [10]:
```
# host = get_hosts(df_1, scrape_list)
# print(host)
```

Hurrah! We have successfully extracted the hosts name from the tweet text

Extracting the awards from the tweet text and storing it in a list

In [11]:
```
def get_awards(data):
    match = ''
    result = {}
    for i in range(len(data)):
        found = re.search('(B|b)est', data[i])
        if found:
            found1 = re.search(r'\bWin\b | \bWins\b | \bwin\b | \bwins\b | \bgoes to\b
            if found1:
                patt = re.compile(r'Best [A-Z][a-z]+ [A-Z*a-z ]+ - [A-Z*a-z ]+')
                matches = patt.findall(data[i])
                for match in matches:
                    if match.title() in result:
                        result[match.title()] += 1
                    else:
                        result[match.title()] = 1
    sorted_awards = sorted(result.items(), key=lambda x:x[1], reverse=True)
    return(sorted_awards[0:50])
```

In [12]:
```
# awards = get_awards(df_1)
# print(awards[:5])
# print(len(awards))
```

we have found the awards but it is not in the required format and it has some extra characters (noise) in it

Searching for the Presenters name in the tweet text

In [13]:
```
# Finding Presenters
def get_presenters(data, actorNames):
    match = ''
    result={}
    for i in range(len(data)):
        found = re.search('(P|p)resent(ing|ed|er|ers)', data[i])
        if found:
            patt = re.compile(r'[A-Z][a-z]+ [A-Z][a-z]+')
            matches = patt.findall(data[i])
            for match in matches:
                if match in actorNames:
                    if match in result:
                        result[match] += 1
                    else:
                        result[match] = 1
    sorted_presenters = sorted(result.items(), key=lambda x:x[1], reverse=True)
    return(sorted_presenters[0:15])
```

In [14]:
```
# presenters = get_presenters(df_1, scrape_list)
# print(presenters[:5])
# print(len(presenters))
```

Nominees for all the awards

In [15]:
```
# Finding Nominees
def get_nominees(data, actorNames):
    match = ''
    result={}
    for i in range(len(data)):
        found = re.search('(N|n)omin(ee|ees|ated)', data[i])
        if found:
            patt = re.compile(r'[A-Z][a-z]+ [A-Z][a-z]+')
            matches = patt.findall(data[i])
            for match in matches:
                if match in actorNames:
                    if match in result:
                        result[match] += 1
                    else:
                        result[match] = 1
    sorted_presenters = sorted(result.items(), key=lambda x:x[1], reverse=True)
    return(sorted_presenters[0:100])
```

In [16]:
```
# nom = get_nominees(df_1, scrape_list)
# print(nom[:5])
# print(len(nom))
```

Finally Winners of all the awards

In [17]:
```
def get_winners(data, awards, actorNames):
    match = ''
    result = {}
    for i in range(len(data)):
        for j in range(len(awards)):
            found = re.search(awards[j][0].lower(), data[i].lower())
            if found:
                found1 = re.search(r'\bWin\b | \bWins\b | \bwin\b | \bwins\b | \bgoes
                if found1:
```

```
                    patt = re.compile(r'[A-Z][a-z]+ [A-Z][a-z]+')
                    matches = patt.findall(data[i])
                    for match in matches:
                        if match in actorNames:
                            if match.title() in result:
                                result[match.title()] += 1
                            else:
                                result[match.title()] = 1
    sorted_awards = sorted(result.items(), key=lambda x:x[1], reverse=True)
    return(sorted_awards[0:50])
```

In [18]:
```
# winnerslist = get_winners(df_1, awards, scrape_list)
# print(winnerslist[:5])
# print(len(winnerslist))
```

In [19]:
```
def report(hosts, awards, presenters, nominees, winners):

    print('Hosts: ')
    for i in hosts:
        print(i[0])
    print()
    print('Presenters: ')
    for i in presenters:
        print(i[0])
    print()
    print('Award Names: ')
    for i in awards:
        print(i[0])
    print()
    print('Nominees: ')
    for i in nominees:
        print(i[0])
    print()
    print('Winners: ')
    for i in winners:
        print(i[0])
    print()
```

In [20]:
```
text_host = read_file()
text_host1 = preprocessing(text_host)
actorNames = scrape_actors()
hosts = get_hosts(text_host1, actorNames)
awards = get_awards(text_host1)
presenters = get_presenters(text_host1,actorNames)
nominees = get_nominees(text_host1, actorNames)
winners = get_winners(text_host1, awards, actorNames)
report(hosts, awards, presenters, nominees, winners)
```

```
1754153
Hosts:
Amy Poehler
Tina Fey

Presenters:
Jennifer Aniston
Salma Hayek
Dakota Johnson
Naomi Watts
Benedict Cumberbatch
Meryl Streep
Tina Fey
Katie Holmes
Jennifer Lopez
Kate Hudson
Amy Poehler
Paul Rudd
Bill Hader
Kevin Hart
Jeremy Renner

Award Names:
Best Actor In A Motion Picture - Drama
Best Motion Picture - Comedy Or Musical
Best Supporting Actor In A Television Series - Matt Bomer
Best Television Series Actor - Drama
Best Actress In A Motion Picture - Comedy Or Musical
Best Motion Picture - Drama
Best Actress In A Motion Picture - Drama
Best Animated Feature Film - How To Train Your Dragon
Best Actor In A Motion Picture - Comedy Or Musical
Best Supporting Actor In A Motion Picture - J
Best Supporting Actress In A Motion Picture - Patricia Arquette
Best Miniseries Or Motion Picture Made For Television - Fargo
Best Actor In A Tv Movie Or Miniseries - Billy Bob Thornton
Best Actress In A Tv Movie Or Miniseries - Maggie Gyllenhaal
Best Foreign Film - Leviathan
Best Television Series - Drama
Best Actor In A Television Series - Comedy Or Musical
Best Television Series - Comedy Or Musical
Best Actress In A Television Series - Drama
Best Motion Picture - Drama
Best Motion Picture - Musical Or Comedy
Best Performance By An Actor In A Television Series - Drama
Best Actor In A Motion Picture - Drama
Best Actor In A Motion Picture - Musical Or Comedy
Best Original Song - Motion Picture
Best Original Song - Motion Picture Goes To
Best Performance By An Actress In A Motion Picture - Drama
Best Original Song - Motion Picture
Best Performance By An Actor In A Motion Picture - Drama
Best Motion Picture - Comedy Or Musical Http
Best Television Series - Musical Or Comedy
Best Television Series - Drama
Best Performance By An Actor In A Tv Series - Musical Or Comedy
Best Performance By An Actress In A Television Series - Drama
Best Motion Picture - Comedy
Best Original Score - Motion Picture Goes To J
Best Motion Picture - Drama Http
```

```
Best Original Song - Motion Picture For
Best Original Score - Motion Picture Win
Best Motion Picture - Comedy Or Musical
Best Original Song - Lana Del Rey
Best Supporting Actress In A Motion Picture - Patricia Arquette
Best Original Song - Motion Picture Http
Best Television Series - Drama At The
Best Motion Picture - Comedy
Best Miniseries At The  S - Variety Http
Best Original Score - Motion Picture Http
Best Original Score - Motion Picture For
Best Actress In A Motion Picture - Drama
Best Motion Picture - Drama Goes To

Nominees:
George Clooney
Kevin Spacey
Amy Adams
Bill Murray
Rosamund Pike
Meryl Streep
Julianne Moore
Christoph Waltz
Amy Poehler
Jennifer Aniston
Benedict Cumberbatch
Eddie Redmayne
Keira Knightley
Jake Gyllenhaal
Felicity Jones
James Spader
Channing Tatum
Don Cheadle
Jessica Lange
Jessica Chastain
Viola Davis
Naomi Watts
Maggie Gyllenhaal
Jeremy Renner
Emma Stone
Michael Keaton
Kate Hudson
Michael Fassbender
Patricia Arquette
David Oyelowo
Ethan Hawke
Helen Mirren
Emily Blunt
Allison Janney
Kathy Bates
Kate Mara
Harrison Ford
Bill Hader
Claire Danes
Reese Witherspoon
Kristen Wiig
Chris Pine
Clive Owen
Mark Ruffalo
Tina Fey
```

```
Jennifer Lopez
Jack Black
Scarlett Johansson
Kevin Hart
Jennifer Lawrence
Steve Carell
Peter Dinklage
Liev Schreiber
Edward Norton
Owen Wilson
Robert Duvall
Paul Rudd
Kerry Washington
Jane Fonda
Kate Beckinsale
Jon Voight
Katherine Heigl
Ralph Fiennes
Angelina Jolie
Lily Tomlin
Charlie Hunnam
Jared Leto
Jon Hamm
Katie Holmes
Anna Kendrick
Bradley Cooper
Salma Hayek
Gwyneth Paltrow
Robin Wright
Angela Bassett
Tom Cruise
Lisa Kudrow
Anna Faris
June Squibb
Katharine Hepburn
Michelle Monaghan
Oscar Isaac
Chris Rock
Colin Farrell
Chris Pratt
Ben Affleck
Bryan Cranston
Mindy Kaling
Hilary Swank
Jeff Goldblum
Will Smith
John Krasinski
Liam Neeson
Shailene Woodley
Rene Russo
Colin Firth
Vince Vaughn
Richard Jenkins
Chris Evans
Adrien Brody

Winners:
Julianne Moore
Patricia Arquette
Kevin Spacey
```

```
Amy Adams
Michael Keaton
Maggie Gyllenhaal
Eddie Redmayne
Henry Cavill
Jennifer Aniston
Benedict Cumberbatch
Rosamund Pike
Emily Blunt
Jamie Foxx
Felicity Jones
```

In [ ]: