Importing all necessary liabraries

In [4]:

```
pip install autoscraper
```

```
Collecting autoscraper
  Downloading https://files.pythonhosted.org/packages/50/44/553afbb7624aaa16
e71546196c1f3beb170dd555a2822785889a9da5c2e7/autoscraper-1.1.12-py3-none-an
y.whl (https://files.pythonhosted.org/packages/50/44/553afbb7624aaa16e715461
96c1f3beb170dd555a2822785889a9da5c2e7/autoscraper-1.1.12-py3-none-any.whl)
Requirement already satisfied: requests in c:\users\tejas\anaconda3\lib\site
-packages (from autoscraper) (2.22.0)
Collecting bs4 (from autoscraper)
  Downloading https://files.pythonhosted.org/packages/10/ed/7e8b97591f6f4561
74139ec089c769f89a94a1a4025fe967691de971f314/bs4-0.0.1.tar.gz (https://file
s.pythonhosted.org/packages/10/ed/7e8b97591f6f456174139ec089c769f89a94a1a402
5fe967691de971f314/bs4-0.0.1.tar.gz)
Requirement already satisfied: lxml in c:\users\tejas\anaconda3\lib\site-pac
kages (from autoscraper) (4.4.1)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\tejas\anaconda3\li
b\site-packages (from requests->autoscraper) (2.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
 c:\users\tejas\anaconda3\lib\site-packages (from requests->autoscraper) (1.
24.2)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\tejas\anaco
nda3\lib\site-packages (from requests->autoscraper) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\tejas\anaconda
3\lib\site-packages (from requests->autoscraper) (2019.9.11)
Requirement already satisfied: beautifulsoup4 in c:\users\tejas\anaconda3\li
b\site-packages (from bs4->autoscraper) (4.8.0)
Requirement already satisfied: soupsieve>=1.2 in c:\users\tejas\anaconda3\li
b\site-packages (from beautifulsoup4->bs4->autoscraper) (1.9.3)
Building wheels for collected packages: bs4
  Building wheel for bs4 (setup.py): started
  Building wheel for bs4 (setup.py): finished with status 'done'
  Created wheel for bs4: filename=bs4-0.0.1-cp37-none-any.whl size=1278 sha2
56=59a365ab892efe3641549c73c1aafe6c22b531ed5b0184b311c9296cb3c46f10
  Stored in directory: C:\Users\Tejas\AppData\Local\pip\Cache\wheels\a0\b0\b
2\4f80b9456b87abedbc0bf2d52235414c3467d8889be38dd472
Successfully built bs4
Installing collected packages: bs4, autoscraper
Successfully installed autoscraper-1.1.12 bs4-0.0.1
Note: you may need to restart the kernel to use updated packages.
```

In [19]:

```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import tweepy
import re
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
wordnet = WordNetLemmatizer()
import re
from nltk.tokenize import sent_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer
import requests
from bs4 import BeautifulSoup as bs
from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException
from selenium.common.exceptions import ElementNotVisibleException
import time
from urllib.request import urlopen, urlretrieve
```

In [20]:

```
from autoscraper import AutoScraper
```

Business Problem

Extract reviews of any product from ecommerce website like amazon.

Emotion mining

Data collection from website imdb of moviews reviews

In [49]:

```
url="https://www.imdb.com/title/tt0108778/reviews?ref_=tt_ql_3"
```

In [50]:

```
html=urlopen(url)
```

In [51]:

```
content_bs=bs(html)
```

In [52]:

```
review=[]
```

In [61]:

```
reviews = content_bs.findAll("div",attrs={"class","text"})
for i in range(len(reviews)):
    reviews[i] = reviews[i].text
```

In [62]:

```
customer_reviews = pd.DataFrame(columns = ["reviews"],dtype=int)
```

In [63]:

```
customer_reviews['reviews']=reviews
```

In [64]:

```
customer_reviews.head()
```

Out[64]:

	reviews
0	There never has been a sitcom that truly pictu...
1	'Friends' is simply the best series ever aired...
2	Everyone says that Seinfield is the greatest s...
3	People are saying that friends is running out ...
4	Are you happy? watch Friends!\nare you sad? wa...

Cleaning the text

In [66]:

```
txt_upd = ' '.join(reviews)
```

In [67]:

```
txt_upd = re.sub("[^A-Za-z" "]+"," ",txt_upd).lower() #remove special character
txt_upd = re.sub("[0-9" "]+"," ",txt_upd).lower() #remove numbers
txt_upd = re.sub(r'^https?:\/\/.*[\r\n]*', '', txt_upd).lower()
```

In [68]:

```
text_tokens = word_tokenize(txt_upd)
```

In [72]:

```
tokens_without_sw = [word for word in text_tokens if not word in stopwords.words()]
```

Creating a DataFrame

In [73]:

```
tf = TfidfVectorizer()
```

In [74]:

```
text_tf = tf.fit_transform(tokens_without_sw)
```

In [75]:

```
feature_names = tf.get_feature_names()
dense = text_tf.todense()
denselist = dense.tolist()
df =pd.DataFrame(denselist, columns=feature_names)
```

In [76]:

```
df
```

Out[76]:

	ability	able	absolute	absolutely	acted	acting	actor	actors	actresses	actually	...	w
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
1372	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
1373	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
1374	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
1375	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
1376	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	

1377 rows × 616 columns

In [77]:

```
word_list = ' '.join(df)
```

In [78]:

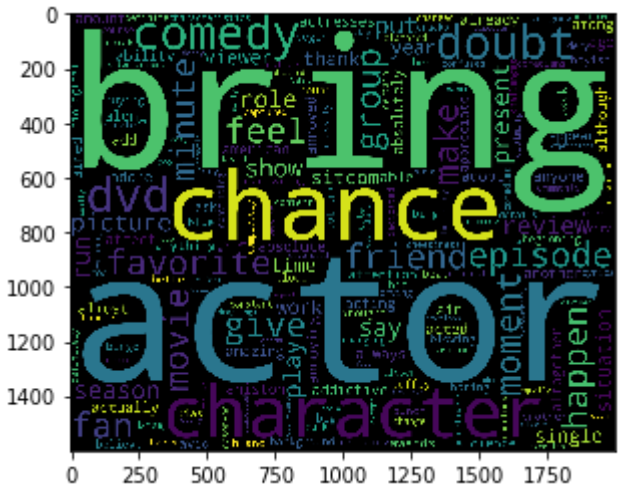
```
wordcloud = WordCloud(background_color='black',
                      width=2000,
                      height=1600).generate(word_list)
```

In [80]:

```
plt.imshow(wordcloud)
```

Out[80]:

```
<matplotlib.image.AxesImage at 0x1d03f4faf08>
```



Perform Emotion Mining

In [81]:

```
with open("C:/Users/Tejas/Downloads/ExcelR DS assignments/Deep_Learning_ExcelR Assignment/p
    positive_words = pw.read().split("\n")
```

In [82]:

```
positive_words = positive_words[35:]
positive_words
```

```
 'adroitly',
 'adulate',
 'adulation',
 'adulatory',
 'advanced',
 'advantage',
 'advantageous',
 'advantageously',
 'advantages',
 'adventuresome',
 'adventurous',
 'advocate',
 'advocated',
 'advocates',
 'affability',
 'affable',
 'affably',
 'affectation',
 'affection',
```

In [95]:

```
with open("C:/Users/Tejas/Downloads/ExcelR DS assignments/Deep_Learning_ExcelR Assignment/n
    negative_words = nw.read().split("\n")
```

In [96]:

```
negative_words = negative_words[35:]
negative_words
```

```
 'aching',
 'acrid',
 'acridly',
 'acridness',
 'acrimonious',
 'acrimoniously',
 'acrimony',
 'adamant',
 'adamantly',
 'addict',
 'addicted',
 'addicting',
 'addicts',
 'admonish',
 'admonisher',
 'admonishingly',
 'admonishment',
 'admonition',
 'adulterate',
 'adulterated',
```

In [97]:

```
txt_neg_in_nw = ' '.join([word for word in df if word in negative_words])
```

In [98]:

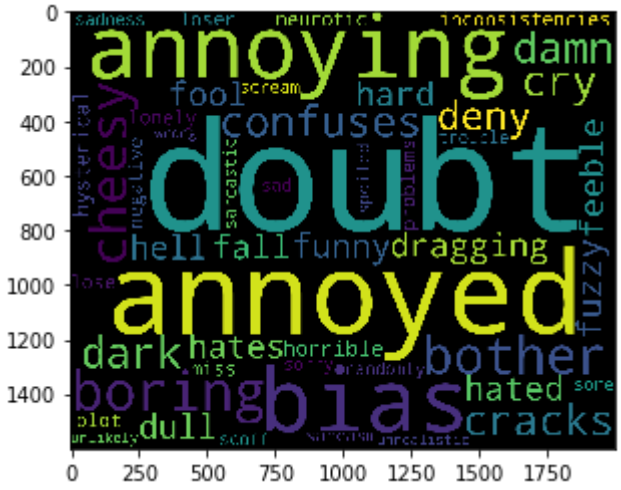
```
wordcloud_neg1 = WordCloud(
                        background_color='black',
                        width=2000,
                        height=1600
                       ).generate(txt_neg_in_nw)
```

In [99]:

```
plt.imshow(wordcloud_neg1)
```

Out[99]:

```
<matplotlib.image.AxesImage at 0x1d03f9f3808>
```



In [89]:

```
txt_neg_in_pw = ' '.join([word for word in df if word in positive_words])
```

In [93]:

```
wordcloud_pos = WordCloud(
                        background_color='black',
                        width=2000,
                        height=1600
                       ).generate(txt_neg_in_pw)
```

In [94]:

```
plt.imshow(wordcloud_pos)
```

Out[94]:

```
<matplotlib.image.AxesImage at 0x1d03f826788>
```

In [102]:

```
with open("C:/Users/Tejas/Downloads/ExcelR DS assignments/Deep_Learning_ExcelR Assignment/s
    stop_words = sw.read().split("\n")
```

In [103]:

```
txt_neg_in_sw = ' '.join([word for word in df if word in stop_words])
```

In [105]:

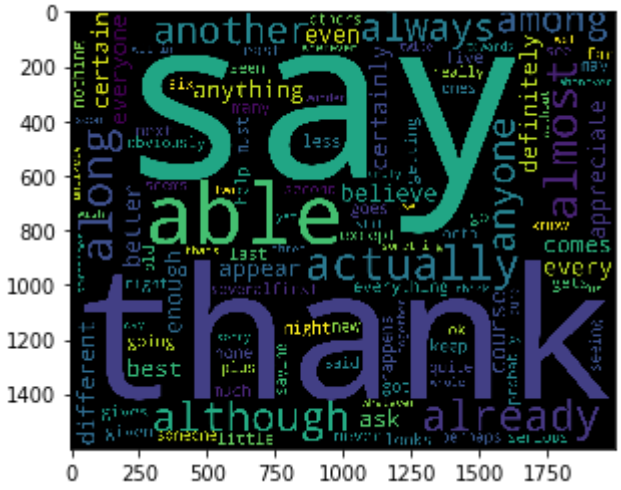
```
wordcloud_stop = WordCloud(
                      background_color='black',
                      width=2000,
                      height=1600
                     ).generate(txt_neg_in_sw)
```

In [106]:

```
plt.imshow(wordcloud_stop)
```

Out[106]:

```
<matplotlib.image.AxesImage at 0x1d03bb32c48>
```



>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>The End!!
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

In [ ]: