```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import nltk
import re #regular expression  (remove special characters)
```

```
#id,location,target,text
#we use encoding here to encode the text,data like emoji

df=pd.read_csv('/content/drive/MyDrive/NLP/twitter_validation.csv',encoding='ISO-8859-1',header=N
df
```

	0	1	2	3
0	3364	Facebook	Irrelevant	I mentioned on Facebook that I was struggling ...
1	352	Amazon	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...
2	8312	Microsoft	Negative	@Microsoft Why do I pay for WORD when it funct...
3	4371	CS-GO	Negative	CSGO matchmaking is so full of closet hacking,...
4	4433	Google	Neutral	Now the President is slapping Americans in the...
...	...	...	...	...

```
df.columns=['ID','location','target','text']
df
```

	ID	location	target	text
0	3364	Facebook	Irrelevant	I mentioned on Facebook that I was struggling ...
1	352	Amazon	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...

```
df.head()
```

	ID	location	target	text
0	3364	Facebook	Irrelevant	I mentioned on Facebook that I was struggling ...
1	352	Amazon	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...
2	8312	Microsoft	Negative	@Microsoft Why do I pay for WORD when it funct...
3	4371	CS-GO	Negative	CSGO matchmaking is so full of closet hacking,...
4	4433	Google	Neutral	Now the President is slapping Americans in the...
998	8069	Microsoft	Positive	Bought a fraction of Microsoft today. Small wins.

```
df.tail()
```

	ID	location	target	text
995	4891	GrandTheftAuto(GTA)	Irrelevant	â□ï¸□ Toronto is the arts and culture capital...
996	4359	CS-GO	Irrelevant	tHIS IS ACTUALLY A GOOD MOVE TOT BRING MORE VI...
997	2652	Borderlands	Positive	Today sucked so itâ□□s time to drink wine n pl...
998	8069	Microsoft	Positive	Bought a fraction of Microsoft today. Small wins.
999	6960	johnson&johnson	Neutral	Johnson & Johnson to stop selling talc baby po...

```
df.shape
```

```
(1000, 4)
```

```
df.isna().sum()
```

```
ID          0
location    0
target      0
text        0
dtype: int64
```

```
df.dtypes
```

```
ID           int64
location    object
target      object
text        object
dtype: object
```

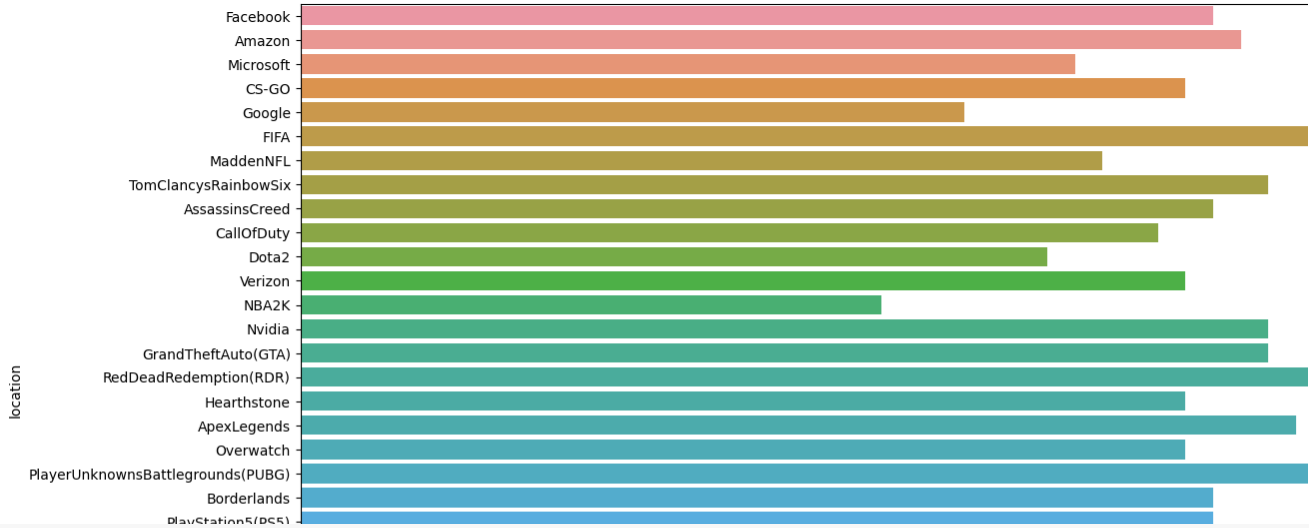
```
df['location'].value_counts()
```

```
RedDeadRedemption(RDR)               40
johnson&johnson                      39
FIFA                                 38
PlayerUnknownsBattlegrounds(PUBG)    38
```

```
LeagueOfLegends                 37
ApexLegends                     36
TomClancysRainbowSix            35
Nvidia                          35
GrandTheftAuto(GTA)             35
Amazon                          34
Fortnite                        34
Facebook                        33
PlayStation5(PS5)               33
AssassinsCreed                  33
Borderlands                     33
Overwatch                       32
Hearthstone                     32
Verizon                         32
CS-GO                           32
CallOfDuty                      31
Cyberpunk2077                   30
WorldOfCraft                    30
MaddenNFL                       29
Microsoft                       28
Dota2                           27
CallOfDutyBlackopsColdWar       27
Xbox(Xseries)                   26
Battlefield                     26
Google                          24
TomClancysGhostRecon            22
NBA2K                           21
HomeDepot                       18
Name: location, dtype: int64
```

```
plt.figure(figsize=(15,10))
sns.countplot(y=df['location'])
```

```
<Axes: xlabel='count', ylabel='location'>
```

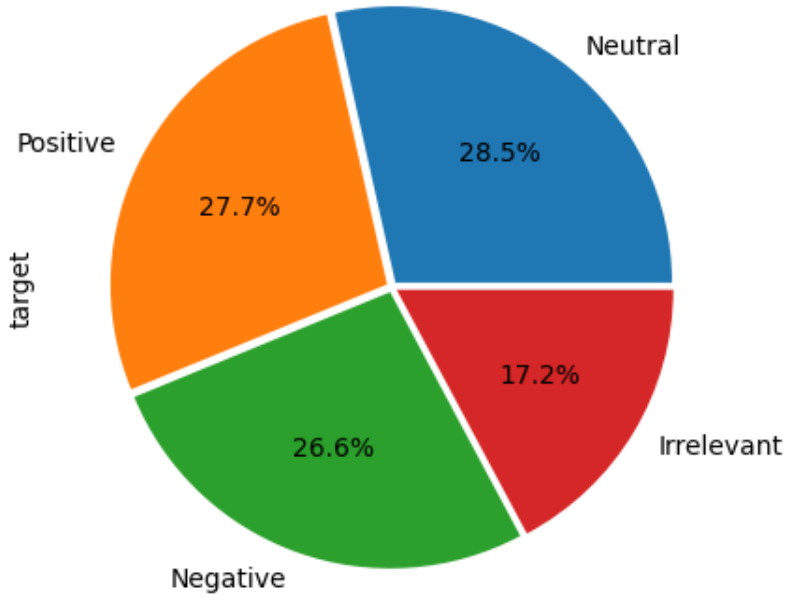



```
tar=df['target'].value_counts()
tar
```

```
Neutral       285
Positive      277
Negative      266
Irrelevant    172
Name: target, dtype: int64
```

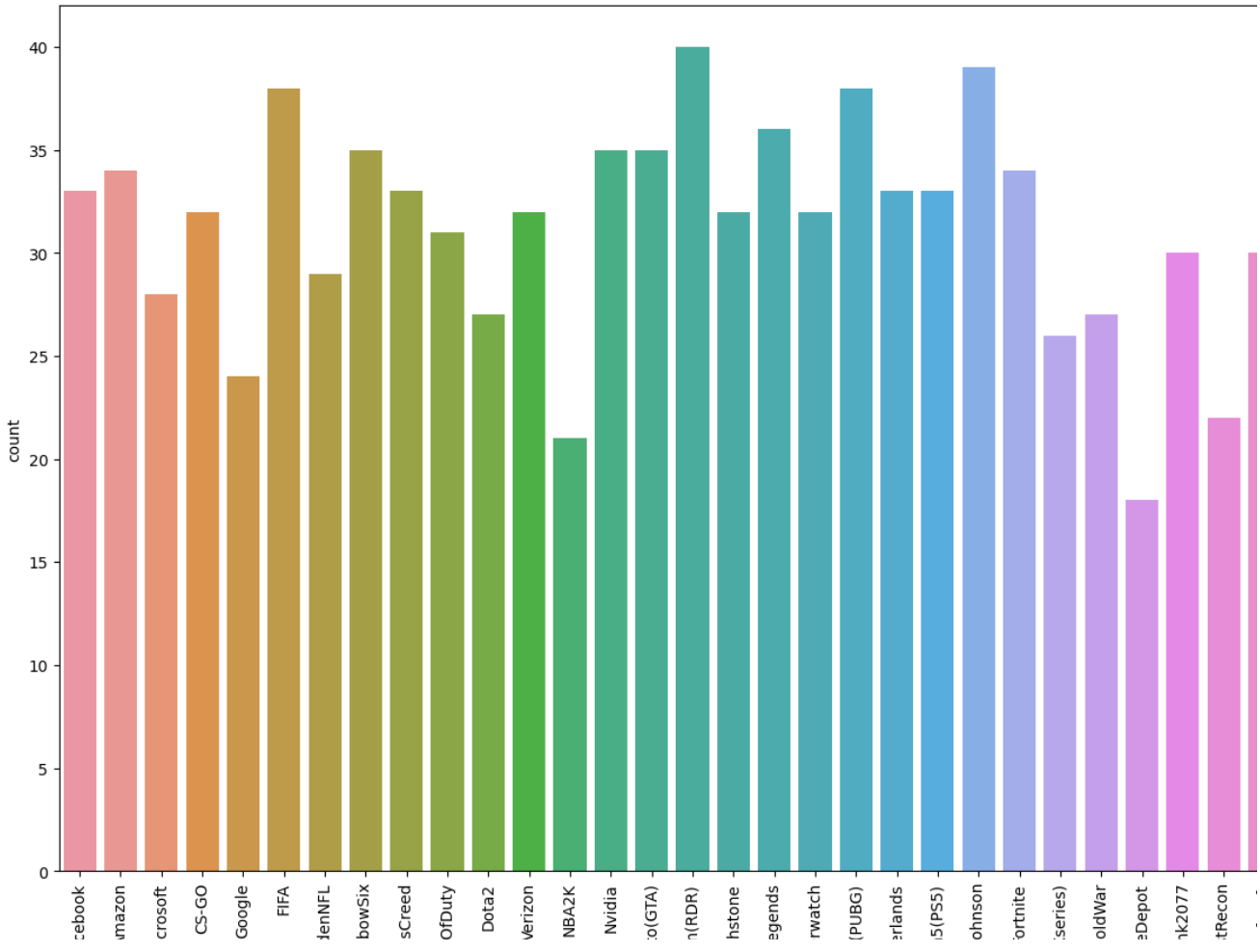
```
tar.plot(kind='pie',explode=(0.02,0.02,0.02,0.02),autopct='%1.1f%%')
```

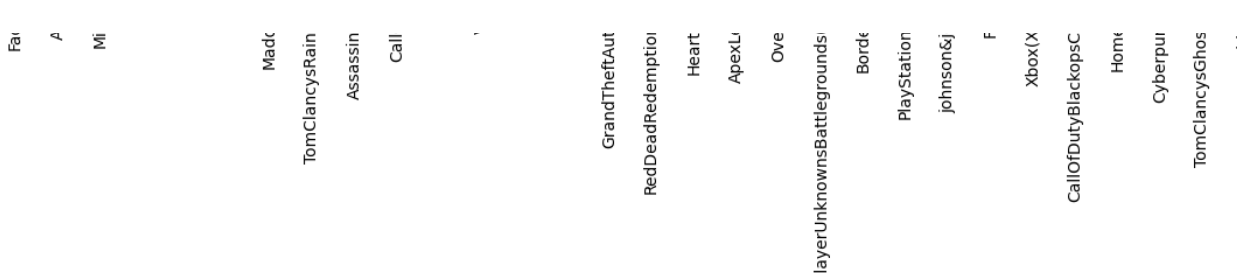
```
<Axes: ylabel='target'>
```




```
plt.figure(figsize=(15,10))
sns.countplot(x='location',data=df)
plt.xticks(rotation=90)
```

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]),
 [Text(0, 0, 'Facebook'),
  Text(1, 0, 'Amazon'),
  Text(2, 0, 'Microsoft'),
  Text(3, 0, 'CS-GO'),
  Text(4, 0, 'Google'),
  Text(5, 0, 'FIFA'),
  Text(6, 0, 'MaddenNFL'),
  Text(7, 0, 'TomClancysRainbowSix'),
  Text(8, 0, 'AssassinsCreed'),
  Text(9, 0, 'CallOfDuty'),
  Text(10, 0, 'Dota2'),
  Text(11, 0, 'Verizon'),
  Text(12, 0, 'NBA2K'),
  Text(13, 0, 'Nvidia'),
  Text(14, 0, 'GrandTheftAuto(GTA)'),
  Text(15, 0, 'RedDeadRedemption(RDR)'),
  Text(16, 0, 'Hearthstone'),
  Text(17, 0, 'ApexLegends'),
  Text(18, 0, 'Overwatch'),
  Text(19, 0, 'PlayerUnknownsBattlegrounds(PUBG)'),
  Text(20, 0, 'Borderlands'),
  Text(21, 0, 'PlayStation5(PS5)'),
  Text(22, 0, 'johnson&johnson'),
  Text(23, 0, 'Fortnite'),
  Text(24, 0, 'Xbox(Xseries)'),
  Text(25, 0, 'CallOfDutyBlackopsColdWar'),
  Text(26, 0, 'HomeDepot'),
  Text(27, 0, 'Cyberpunk2077'),
  Text(28, 0, 'TomClancysGhostRecon'),
  Text(29, 0, 'WorldOfCraft'),
  Text(30, 0, 'LeagueOfLegends'),
  Text(31, 0, 'Battlefield')])
```



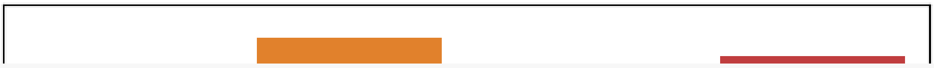



```
df['target'].value_counts()
```

```
Neutral       285
Positive      277
Negative      266
Irrelevant    172
Name: target, dtype: int64
```

```
sns.countplot(x='target',data=df)
```

```
<Axes: xlabel='target', ylabel='count'>
```



```
df['text'].value_counts()
```

```
Wow
2
I mentioned on Facebook that I was struggling for motivation to go for a run the other
day, which has been translated by Tomâs great auntie as âHayley canât get out of
bedâ and told to his grandma, who now thinks Iâm a lazy, terrible person ð¤£    1
Update: I actually finished Odyssey. I wish the three-ponged questline structure was
communicated a bit more clearly but that game is a lot of fun once it hits its stride.
twitter.com/iv_javy/statusâ¦                                                     1
Know Your History...unless it involves China....because like....we have investors and
shit...
1
Congratulations to @ninamanning78 on publication day. Hope you have a fabulous day and
that your book flies ððð
1

..
eSports: Curiosities you probably didnât know about Red Dead RedemptionÂ 2
goalsn.com/2020/02/esportâ¦ https://t.co/JMLBSxAUou
1
Nvidia GeForce Now: Google Stadia rival suffers anotherÂ blow news89.net/nvidia-geforceâ¦
https://t.co/9ijNjV7L4C
1
I SAW A TWEET ABOUT HOW IT'S A STORY TOLD OVER AND OVER CUZ THEY'RE MAD MEN DIED IN THE
WAR AND IT DIDN'T HAVE BATTLEFIELD 1 TREATMENT, I CAN'T EVEN THINK OF ANOTHER WW1 MOVIE
PPL WOULD EVEN KNOW                                                          1
Bro I had an abusive internet boyfriend that used to call me a stupid bitch whenever
niggas used to like my pics on facebook and I used to beg him to not be mad at me. I was
such a stupid bitch FOR REAL                                                   1
Johnson & Johnson to stop selling talc baby powder in U.S. and Canada j.mp/3e1YtDV
(Reuters) https://t.co/dsaUTgb5p9
1
Name: text, Length: 999, dtype: int64
```

```
df.drop(df.index[(df['target']=='Irrelevant')],axis=0,inplace=True)
df
```

	ID	location	target	text
1	352	Amazon	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...

```
#reset index values
df.reset_index(drop=True,inplace=True)
df
```

	ID	location	target	text
0	352	Amazon	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...
1	8312	Microsoft	Negative	@Microsoft Why do I pay for WORD when it funct...
2	4371	CS-GO	Negative	CSGO matchmaking is so full of closet hacking,...
3	4433	Google	Neutral	Now the President is slapping Americans in the...
4	6273	FIFA	Negative	Hi @EAHelp Iâ□□ve had Madeleine McCann in my c...
...	...	...	...	...
823	314	Amazon	Negative	Please explain how this is possible! How can t...
824	9701	PlayStation5(PS5)	Positive	Good on Sony. As much as I want to see the new...
825	2652	Borderlands	Positive	Today sucked so itâ□□s time to drink wine n pl...
826	8069	Microsoft	Positive	Bought a fraction of Microsoft today. Small wins.
827	6960	johnson&johnson	Neutral	Johnson & Johnson to stop selling talc baby po...

828 rows × 4 columns

```
df.drop(['ID','location'],axis=1,inplace=True)
df
```

	target	text
0	Neutral	BBC News - Amazon boss Jeff Bezos rejects clai...
1	Negative	@Microsoft Why do I pay for WORD when it funct...
2	Negative	CSGO matchmaking is so full of closet hacking,...
3	Neutral	Now the President is slapping Americans in the...
4	Negative	Hi @EAHelp Iâ□□ve had Madeleine McCann in my c...
...	...	...
823	Negative	Please explain how this is possible! How can t...
824	Positive	Good on Sony. As much as I want to see the new...
825	Positive	Today sucked so itâ□□s time to drink wine n pl...
826	Positive	Bought a fraction of Microsoft today. Small wins.
827	Neutral	Johnson & Johnson to stop selling talc baby po...

828 rows × 2 columns

```
# df['target']=df['target'].str.replace('Neutral','0')
# df['target']=df['target'].str.replace('Negative','-1')
```

```
# df['target']=df['target'].str.replace('Positive','1')
# df['target']=df['target'].astype(float)
# df
#         OR
#using map function
df['target']=df['target'].map({'Positive':1,'Negative':-1,'Neutral':0})
df
```

	target	text
0	0	BBC News - Amazon boss Jeff Bezos rejects clai...
1	-1	@Microsoft Why do I pay for WORD when it funct...
2	-1	CSGO matchmaking is so full of closet hacking,...
3	0	Now the President is slapping Americans in the...
4	-1	Hi @EAHelp Iâ□□ve had Madeleine McCann in my c...
...	...	...
823	-1	Please explain how this is possible! How can t...
824	1	Good on Sony. As much as I want to see the new...
825	1	Today sucked so itâ□□s time to drink wine n pl...
826	1	Bought a fraction of Microsoft today. Small wins.
827	0	Johnson & Johnson to stop selling talc baby po...

828 rows × 2 columns

```
df.dtypes
```

```
target     int64
text      object
dtype: object
```

```
nltk.download('wordnet')
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
True
```

```
tweets=df.text
tweets
```

```
0      BBC News - Amazon boss Jeff Bezos rejects clai...
1      @Microsoft Why do I pay for WORD when it funct...
2      CSGO matchmaking is so full of closet hacking,...
3      Now the President is slapping Americans in the...
```

```
4      Hi @EAHelp Iâ□□ve had Madeleine McCann in my c...
                              ...
823    Please explain how this is possible! How can t...
824    Good on Sony. As much as I want to see the new...
825    Today sucked so itâ□□s time to drink wine n pl...
826    Bought a fraction of Microsoft today. Small wins.
827    Johnson & Johnson to stop selling talc baby po...
Name: text, Length: 828, dtype: object
```

```
f=lambda x,y:x+y
```

```
#preprocessing text data
from nltk.tokenize import TweetTokenizer
tk=TweetTokenizer()
tweets=tweets.apply(lambda x:tk.tokenize(x)).apply(lambda x:" ".join(x))
tweets

#please explain how this is possible

#please explain how this is possible
#explain
#how
#this
#is
#possible
```

```
0      BBC News - Amazon boss Jeff Bezos rejects clai...
1      @Microsoft Why do I pay for WORD when it funct...
2      CSGO matchmaking is so full of closet hacking ...
3      Now the President is slapping Americans in the...
4      Hi @EAHelp Iâ □ □ ve had Madeleine McCann in m...
                              ...
823    Please explain how this is possible ! How can ...
824    Good on Sony . As much as I want to see the ne...
825    Today sucked so itâ □ □ s time to drink wine n...
826    Bought a fraction of Microsoft today . Small w...
827    Johnson & Johnson to stop selling talc baby po...
Name: text, Length: 828, dtype: object
```

```
#Removing Special character
#re  :regular expression
tweets=tweets.str.replace('[^a-zA-Z0-9]+',' ')
print(tweets)
```

```
<ipython-input-107-d93fe5b901b7>:3: FutureWarning: The default value of regex will change f
  tweets=tweets.str.replace('[^a-zA-Z0-9]+',' ')
0      BBC News Amazon boss Jeff Bezos rejects claims...
1       Microsoft Why do I pay for WORD when it funct...
2      CSGO matchmaking is so full of closet hacking ...
3      Now the President is slapping Americans in the...
4      Hi EAHelp I ve had Madeleine McCann in my cell...
                              ...
823    Please explain how this is possible How can th...
824    Good on Sony As much as I want to see the new ...
825    Today sucked so it s time to drink wine n play...
826     Bought a fraction of Microsoft today Small wins
827    Johnson Johnson to stop selling talc baby powd...
Name: text, Length: 828, dtype: object
```

```
str1='Wonderful @peaccock!12345#@!'
str2=re.sub('[^a-zA-Z0-9]+'," ",str1)       #sub : used for replacing  ('^' means it allows to pr
print(str2)                                                             # ('+' means it re move com
```

```
Wonderful peaccock 12345
```

```
#Snowballstemmer
from nltk.stem import SnowballStemmer
stemmer=SnowballStemmer('english')
tweets=tweets.apply(lambda x:[stemmer.stem(i.lower()) for i in tk.tokenize(x)]).apply(lambda x:'
tweets
```

```
0      bbc news amazon boss jeff bezo reject claim co...
1      microsoft whi do i pay for word when it functi...
2      csgo matchmak is so full of closet hack it s a...
3      now the presid is slap american in the face th...
4      hi eahelp i ve had madelein mccann in my cella...
                            ...
823    pleas explain how this is possibl how can they...
824    good on soni as much as i want to see the new ...
825    today suck so it s time to drink wine n play b...
826       bought a fraction of microsoft today small win
827    johnson johnson to stop sell talc babi powder ...
Name: text, Length: 828, dtype: object
```

```
from nltk.corpus import stopwords
sw=stopwords.words('english')
tweets=tweets.apply(lambda x:[i for i in tk.tokenize(x) if i not in sw]).apply(lambda x:" ".join(
tweets
```

```
0      bbc news amazon boss jeff bezo reject claim co...
1      microsoft whi pay word function poor samsungus...
2           csgo matchmak full closet hack truli aw game
3      presid slap american face realli commit unlaw ...
4      hi eahelp madelein mccann cellar past 13 year ...
                            ...
823    pleas explain possibl let compani overcharg sc...
824    good soni much want see new ps5 go right much ...
825    today suck time drink wine n play borderland s...
826             bought fraction microsoft today small win
827    johnson johnson stop sell talc babi powder u c...
Name: text, Length: 828, dtype: object
```

```
#Vectorization
from sklearn.feature_extraction.text import TfidfVectorizer
vec=TfidfVectorizer()
train_data=vec.fit_transform(tweets)
train_data             #x-variable
print(train_data)
```

```
  (0, 734)      0.2456282916977699
  (0, 521)      0.2456282916977699
  (0, 3599)     0.19717097284763976
  (0, 878)      0.11802111433875656
  (0, 1055)     0.2456282916977699
  (0, 1184)     0.22139963227270484
  (0, 2088)     0.12503547156807654
  (0, 342)      0.2135997461023189
  (0, 907)      0.16343698819954885
```

```
  (0, 852)      0.2018385048310212
  (0, 2878)     0.2456282916977699
  (0, 607)      0.2456282916977699
  (0, 1915)     0.23145543449136263
  (0, 674)      0.2135997461023189
  (0, 412)      0.14270670271218175
  (0, 2389)     0.365996231282465
  (0, 576)      0.46291086898272527
  (1, 842)      0.4055823664694651
  (1, 3014)     0.4055823664694651
  (1, 2670)     0.3821800909185634
  (1, 1469)     0.4055823664694651
  (1, 3819)     0.36557591217188057
  (1, 2572)     0.3126902562590763
  (1, 3778)     0.26216072802580975
  (1, 2251)     0.24555654927912696
  :     :
  (825, 672)    0.22981061112100945
  (825, 1048)   0.2315686698425631
  (825, 3502)   0.21750175079084832
  (825, 3352)   0.2904718522758868
  (825, 2638)   0.17148706662740873
  (826, 1444)   0.5079831062080814
  (826, 3195)   0.47867226429410115
  (826, 677)    0.4174215841659411
  (826, 3789)   0.353278941165688
  (826, 3512)   0.34523850330234374
  (826, 2251)   0.3075542453642147
  (827, 222)    0.3085327836746446
  (827, 1186)   0.3085327836746446
  (827, 2925)   0.3085327836746446
  (827, 166)    0.3085327836746446
  (827, 2326)   0.3085327836746446
  (827, 755)    0.2602967814546125
  (827, 3070)   0.24766572568673034
  (827, 3400)   0.24766572568673034
  (827, 3319)   0.21723219669277322
  (827, 1763)   0.14999073362910845
  (827, 2688)   0.23368360020019152
  (827, 544)    0.22309506625787626
  (827, 1937)   0.37089519596031884
  (827, 878)    0.14824588278342318
```

```
train_data.shape     #x-variable
```

```
(828, 3929)
```

```
y=df['target'].values
y
```

```
array([ 0, -1, -1,  0, -1,  1,  1,  1, -1,  1,  1, -1,  0, -1,  1,  1, -1,
        1, -1, -1,  0, -1,  0,  0, -1, -1,  1,  1, -1,  1, -1,  0,  0,  1,
        0,  1,  0,  0,  0,  1,  0, -1, -1, -1,  0,  1, -1, -1,  1,  1,  1,
        1,  1, -1, -1,  1,  1, -1,  0, -1,  0, -1,  1, -1, -1,  1,  1,  1,
        0,  0,  0,  1,  1,  0,  1,  0, -1, -1,  0,  0, -1,  1, -1, -1, -1,
        0,  1,  0, -1,  1,  1,  0,  1,  0,  1, -1,  0,  0,  0, -1,  0, -1,
        0,  0,  1,  1,  0, -1, -1,  1, -1,  0, -1,  1,  0, -1,  0,  1,  0,
        1,  1,  0,  0,  0,  0,  1,  0,  1,  1, -1,  0,  0,  0,  0, -1,  0,
        1, -1,  0, -1,  0, -1, -1, -1,  1,  1,  1,  0,  0,  1,  0,  0,  0,
        1,  0, -1, -1,  0,  1,  1,  0,  1,  1,  0,  0, -1, -1, -1, -1,  1,
        0,  0,  1,  1,  1,  1, -1,  1,  1,  0, -1, -1, -1,  1,  1, -1, -1,
        1,  1, -1,  1,  1, -1,  1,  0, -1,  0,  0,  1, -1,  1,  1,  0,  1,
```

```
       -1, -1,  1,  1,  1,  1,  0,  0,  1, -1,  0,  1,  0, -1,  0,  0, -1,
        1,  1, -1,  0,  1,  0, -1,  0, -1,  1,  1, -1, -1, -1,  1, -1,  0,
        1,  0,  0, -1,  1, -1,  1, -1,  0,  0,  1, -1,  0, -1,  1, -1,  1,
        1,  1,  1,  1,  1, -1, -1,  1, -1,  0,  0,  0,  1,  0,  1, -1,  0,
        0,  0,  0, -1,  1, -1, -1,  1,  1,  0,  0, -1, -1, -1,  0,  1,  0,
       -1,  1,  0, -1, -1, -1,  1,  0,  0, -1,  1,  1,  0,  1,  0,  0,  1,
        1, -1,  0,  1, -1,  0, -1, -1,  1,  1,  1,  1,  0, -1,  0,  1,  0,
        1, -1, -1, -1,  1,  0,  1, -1,  0, -1,  1,  1,  1,  1,  0,  0,  0,
       -1,  1,  1,  0, -1,  1,  0, -1, -1, -1, -1, -1,  0,  0,  0,  1,  1,
       -1, -1,  0, -1,  0,  0, -1,  1, -1,  1,  1,  1,  0,  1,  0,  0, -1,
        1,  0,  0,  0,  0,  0,  0,  0,  0, -1, -1,  1,  1,  0, -1, -1,  1,
        1, -1,  1,  1,  1,  1,  1,  0, -1,  1,  0,  0,  1,  1,  1,  1,  0,
       -1, -1, -1, -1,  0,  1, -1, -1,  1,  1,  0,  0, -1, -1,  1,  0, -1,
       -1, -1,  0,  0,  1, -1, -1, -1,  0,  0,  0, -1, -1,  1, -1,  0, -1,
        0,  1, -1,  0,  1,  1, -1,  0,  0,  1, -1, -1,  0,  0, -1,  1, -1,
        0, -1, -1, -1,  1, -1,  1, -1,  1, -1, -1,  0, -1,  0, -1,  1, -1,
        0, -1, -1,  0,  0,  1, -1,  1,  0,  0,  0,  0, -1,  0,  0,  0, -1,
       -1,  0,  1,  0,  0, -1,  0,  1,  0,  0,  0,  0,  0,  1,  0,  1,  1,
        1,  0, -1,  1,  0,  0, -1,  1,  0,  0, -1,  0, -1,  0,  1, -1,  1,
       -1, -1,  0,  0,  0,  0,  1,  1,  1, -1, -1,  0,  1,  0,  0, -1,  1,
        1,  0,  1, -1, -1,  0,  1, -1,  1, -1,  0,  1,  1,  0,  0,  0,  1,
        0, -1,  0,  0, -1,  1, -1,  0,  1,  1,  1,  1,  0, -1,  0,  1,  1,
        1,  1,  1, -1,  0,  1,  0,  0, -1, -1, -1,  0,  1,  0, -1,  1,  1,
        1,  0,  1, -1,  0, -1,  0, -1,  0,  0,  1, -1,  1,  1,  0, -1,  0,
       -1, -1, -1, -1,  1,  1,  1,  1,  0, -1, -1,  1, -1, -1,  0,  0,  1,
        0, -1,  0,  1, -1,  0,  1, -1,  0,  0,  1, -1,  0, -1,  1,  1,  0,
        1,  0,  1, -1,  0,  0,  0,  1,  0,  0, -1,  1,  0, -1, -1,  0,  0,
        1, -1, -1, -1, -1,  1,  0,  0,  1,  0, -1,  1,  1, -1,  1,  1,  0,
       -1,  0,  1,  1, -1, -1, -1,  1, -1,  0, -1,  0,  0,  1,  1, -1,  0,
        1, -1, -1, -1, -1, -1, -1, -1, -1,  0, -1,  0,  0,  0,  1,  0,  0,
        0, -1,  0,  1,  0, -1, -1,  1,  0,  1,  0,  1,  0, -1,  1,  1,  1,
        1, -1, -1,  1,  0,  0,  0,  0,  0,  0, -1, -1, -1, -1,  1, -1,  0,
        1,  0, -1,  1,  1, -1,  1,  0,  0,  1, -1,  0, -1,  0,  1,  1,  0,
       -1,  1, -1, -1,  0, -1,  0, -1,  1,  0, -1, -1,  1,  1, -1,  0, -1,
        0,  0,  0,  0,  0,  0,  1,  0,  1,  1,  1, -1,  0,  1,  0,  1,  0,
        1,  0,  1,  0, -1, -1,  1,  1,  1,  1,  0, -1,  1,  1, -1, -1, -1,
        0,  1,  0,  1,  1,  0,  1, -1,  1,  1,  1,  0])
```

```
#train_test_split
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(train_data,y,test_size=0.30,random_state=42)
y_train
```

```
array([ 1,  1, -1, -1,  0, -1,  0,  1,  1,  0, -1,  0, -1, -1,  1,  0, -1,
        1, -1, -1,  1,  0,  1, -1, -1,  0,  0,  1, -1,  1, -1,  0,  0, -1,
       -1, -1, -1,  0,  0,  1, -1,  0,  0, -1,  1,  1,  1, -1,  0,  1, -1,
       -1,  1,  0,  1, -1, -1,  1,  1, -1,  1,  0,  1,  1,  0,  1,  0,  0,
       -1,  1,  0,  1, -1, -1, -1, -1, -1, -1, -1,  0, -1,  1, -1,  0,  1,
        0,  1,  1,  0,  1, -1,  1,  0, -1,  1, -1, -1,  0,  0, -1,  0,  1,
       -1, -1,  1, -1,  0,  1,  1,  0,  1,  0, -1,  1,  1,  0,  0,  0,  0,
        1, -1,  1,  1,  1,  1,  0,  1,  0, -1,  0,  0,  1,  0, -1, -1, -1,
       -1,  1,  1,  1, -1,  1,  0,  1,  1,  1,  1,  0,  0, -1, -1,  0,  0,
        0, -1,  0,  0,  0,  1,  1,  0, -1, -1,  0,  0,  0, -1, -1, -1, -1,
       -1, -1,  0,  0, -1, -1,  0,  1, -1, -1,  1, -1,  0,  0, -1, -1, -1,
        0,  0, -1,  0,  0,  1,  0, -1, -1, -1,  0,  1,  1,  1,  1,  1,  1,
        0,  1, -1,  1, -1, -1, -1,  0, -1,  1,  1, -1,  1, -1,  0,  0, -1,
        1,  0, -1,  1,  1,  0,  1, -1, -1, -1,  1,  0,  0, -1,  0,  0,  0,
        0,  1,  1, -1,  1,  1,  0,  1,  0, -1, -1,  1,  1,  1,  1,  1,  1,
        0,  1,  0,  0,  1, -1,  0,  1, -1,  1, -1,  0,  0,  1,  0,  1,  0,
        1, -1,  1,  1,  0,  1,  0, -1,  0,  1,  0,  0,  1,  0, -1,  0,  1,
        1,  0, -1,  1, -1,  0,  1,  1, -1,  1, -1,  0,  0, -1,  0,  0,  1,
        0,  0,  1,  1,  0,  0,  0, -1,  0,  0, -1,  0, -1,  0,  1, -1,  0,
        1,  0,  1,  1,  0, -1, -1,  0, -1, -1,  0, -1,  1, -1, -1,  1,  0,
```

```
       -1,  0,  0,  0,  1, -1,  0,  1,  0,  1,  0, -1,  1, -1, -1,  0,  0,
       -1,  1,  0,  1, -1,  1,  0,  1,  0,  1,  0, -1,  1, -1,  0,  1,  1,
        1,  0,  0, -1,  0, -1,  1,  0,  1, -1,  1,  1,  1, -1,  0, -1, -1,
        1,  1, -1, -1,  0,  1, -1, -1, -1,  1,  0,  1,  0,  0, -1,  0,  0,
        0,  0, -1,  0,  1,  0, -1,  1,  1,  0,  0,  0,  1,  1,  0,  1,  1,
        1,  1,  0,  0,  0,  0,  1, -1, -1,  0,  0, -1, -1, -1, -1,  1, -1,
        1, -1, -1, -1, -1, -1, -1,  1,  1,  0,  0,  1,  0, -1,  0,  1, -1,
        1,  0,  1,  0,  0,  0,  0,  1,  1, -1, -1,  0,  1, -1,  0,  1,  1,
       -1, -1,  1, -1,  0, -1,  0,  1,  0,  1,  0, -1,  0,  1, -1, -1,  0,
       -1,  0,  1,  1,  1,  1,  1,  1,  0, -1,  0,  1, -1, -1, -1,  0,  0,
       -1, -1,  0,  1,  0,  0,  1,  0,  0,  0,  1, -1, -1,  1, -1,  0,  0,
       -1,  0,  1,  0, -1,  0, -1,  1, -1, -1, -1,  0, -1, -1,  0,  1,  0,
        1,  1,  0,  1, -1, -1,  0,  0,  1,  1,  1,  1,  0,  0,  0,  0,  0,
        0,  1, -1,  0,  0, -1, -1,  0, -1,  0, -1,  0, -1,  1,  0, -1,  0,
        0])
```

```
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
svm_model=SVC()
nb_model=MultinomialNB()
knn_model=KNeighborsClassifier()
rf_model=RandomForestClassifier()
df_model=DecisionTreeClassifier()
lst_model=[svm_model,nb_model,knn_model,rf_model,df_model]
```

```
from sklearn.metrics import confusion_matrix,classification_report
for i in lst_model:
  print(i)
  i.fit(x_train,y_train)
  y_pred=i.predict(x_test)
  print("*********************************************")
  print(classification_report(y_test,y_pred))
  print("*********************************************")
  print("*********************************************")
  print(confusion_matrix(y_test,y_pred))
```

```
SVC()
*********************************************
              precision    recall  f1-score   support

          -1       0.71      0.56      0.62        79
           0       0.47      0.76      0.58        79
           1       0.72      0.46      0.56        91

    accuracy                           0.59       249
   macro avg       0.63      0.59      0.59       249
weighted avg       0.64      0.59      0.59       249

*********************************************
*********************************************
[[44 31  4]
 [ 7 60 12]
 [11 38 42]]
MultinomialNB()
*********************************************
              precision    recall  f1-score   support
```

```
          -1       0.62      0.59      0.61        79
           0       0.48      0.57      0.52        79
           1       0.66      0.58      0.62        91

    accuracy                           0.58       249
   macro avg       0.59      0.58      0.58       249
weighted avg       0.59      0.58      0.58       249

**********************************************
**********************************************
[[47 25  7]
 [14 45 20]
 [15 23 53]]
KNeighborsClassifier()
**********************************************
              precision    recall  f1-score   support

          -1       0.42      0.82      0.56        79
           0       0.49      0.32      0.38        79
           1       0.67      0.32      0.43        91

    accuracy                           0.48       249
   macro avg       0.53      0.49      0.46       249
weighted avg       0.54      0.48      0.46       249

**********************************************
**********************************************
[[65  9  5]
 [45 25  9]
 [45 17 29]]
RandomForestClassifier()
**********************************************
              precision    recall  f1-score   support

          -1       0.51      0.72      0.60        79
           0       0.51      0.56      0.53        79
```