```
import pandas as pd
import numpy as np
```

```
train=pd.read_csv("/content/train[1].txt",encoding='ISO-8859-1', sep=';', names=["Description","Emotion"])
train
```

	Description	Emotion
0	i didnt feel humiliated	sadness
1	i can go from feeling so hopeless to so damned...	sadness
2	im grabbing a minute to post i feel greedy wrong	anger
3	i am ever feeling nostalgic about the fireplac...	love
4	i am feeling grouchy	anger
...	...	...
15995	i just had a very brief time in the beanbag an...	sadness
15996	i am now turning and i feel pathetic that i am...	sadness
15997	i feel strong and good overall	joy
15998	i feel like this was such a rude comment and i...	anger
15999	i know a lot but i feel so stupid because i ca...	sadness

16000 rows × 2 columns

```
test=pd.read_csv("/content/test[1].txt",encoding='ISO-8859-1', sep=';', names=["Description","Emotion"])
test
```

	Description	Emotion
0	im feeling rather rotten so im not very ambiti...	sadness
1	im updating my blog because i feel shitty	sadness
2	i never make her separate from me because i do...	sadness
3	i left with my bouquet of red and yellow tulip...	joy
4	i was feeling a little vain when i did this one	sadness
...	...	...
1995	i just keep feeling like someone is being unki...	anger
1996	im feeling a little cranky negative after this...	anger
1997	i feel that i am useful to my people and that ...	joy
1998	im feeling more comfortable with derby i feel ...	joy
1999	i feel all weird when i have to meet w people ...	fear

2000 rows × 2 columns

```
#combine both train and test
df=pd.concat([train,test])
df
```

	Description	Emotion

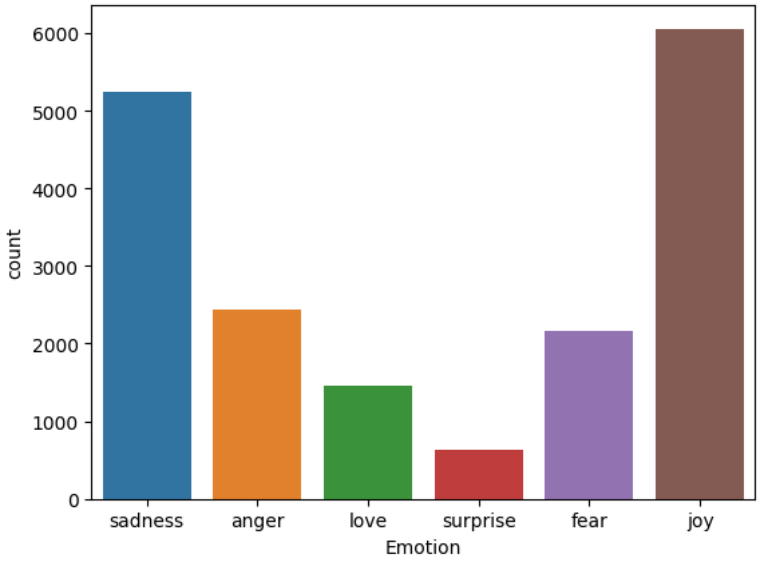
```
df.reset_index(drop=True, inplace=True)
df
```

	Description	Emotion
0	i didnt feel humiliated	sadness
1	i can go from feeling so hopeless to so damned...	sadness
2	im grabbing a minute to post i feel greedy wrong	anger
3	i am ever feeling nostalgic about the fireplac...	love
4	i am feeling grouchy	anger
...	...	...
17995	i just keep feeling like someone is being unki...	anger
17996	im feeling a little cranky negative after this...	anger
17997	i feel that i am useful to my people and that ...	joy
17998	im feeling more comfortable with derby i feel ...	joy
17999	i feel all weird when i have to meet w people ...	fear

18000 rows × 2 columns

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.countplot(x='Emotion', data=df)
```

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




```
df['Emotion']=df['Emotion'].map({'sadness':1, 'anger':2, 'love':3, 'surprise':4, 'fear':5, 'joy':6})
df
```

```
#download nlp
import nltk
nltk.download('wordnet')
nltk.download('stopwords')
nltk.download('punkt')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

```
cap=df.Description
cap
```

```
0                                i didnt feel humiliated
1        i can go from feeling so hopeless to so damned...
2         im grabbing a minute to post i feel greedy wrong
3        i am ever feeling nostalgic about the fireplac...
4                                  i am feeling grouchy
                               ...
17995    i just keep feeling like someone is being unki...
17996    im feeling a little cranky negative after this...
17997    i feel that i am useful to my people and that ...
17998    im feeling more comfortable with derby i feel ...
17999    i feel all weird when i have to meet w people ...
Name: Description, Length: 18000, dtype: object
```

```
#tokenization
from nltk import TweetTokenizer
tk=TweetTokenizer()
cap=cap.apply(lambda x: tk.tokenize(x)).apply(lambda x:" ".join(x))
cap
```

```
0                                i didnt feel humiliated
1        i can go from feeling so hopeless to so damned...
2         im grabbing a minute to post i feel greedy wrong
3        i am ever feeling nostalgic about the fireplac...
4                                  i am feeling grouchy
                               ...
17995    i just keep feeling like someone is being unki...
17996    im feeling a little cranky negative after this...
17997    i feel that i am useful to my people and that ...
17998    im feeling more comfortable with derby i feel ...
17999    i feel all weird when i have to meet w people ...
Name: Description, Length: 18000, dtype: object
```

```
#remove special charecters
nw_cap=cap.str.replace('[^a-zA-Z0-9]+',' ')
nw_cap
```

```
<ipython-input-33-863deb530781>:2: FutureWarning: The default value of regex will change from True to False in a future version.
  nw_cap=cap.str.replace('[^a-zA-Z0-9]+',' ')
0                                i didnt feel humiliated
1        i can go from feeling so hopeless to so damned...
2         im grabbing a minute to post i feel greedy wrong
3        i am ever feeling nostalgic about the fireplac...
4                                  i am feeling grouchy
                               ...
17995    i just keep feeling like someone is being unki...
17996    im feeling a little cranky negative after this...
17997    i feel that i am useful to my people and that ...
17998    im feeling more comfortable with derby i feel ...
17999    i feel all weird when i have to meet w people ...
Name: Description, Length: 18000, dtype: object
```

```
#tokenize each word in the text whose len is greater than 3
from nltk.tokenize import word_tokenize
nw_cap=nw_cap.apply(lambda x:' '.join(w for w in word_tokenize(x) if len(w)>3))
nw_cap
```

```
0                                  didnt feel humiliated
1        from feeling hopeless damned hopeful just from...
2                    grabbing minute post feel greedy wrong
3        ever feeling nostalgic about fireplace will kn...
4                                        feeling grouchy
                               ...
17995    just keep feeling like someone being unkind do...
17996    feeling little cranky negative after this doct...
17997    feel that useful people that gives great feeli...
17998    feeling more comfortable with derby feel thoug...
```

```
17999    feel weird when have meet people text like don...
Name: Description, Length: 18000, dtype: object
```

```
from nltk.stem import SnowballStemmer
stemmer=SnowballStemmer('english')
nw_cap=nw_cap.apply(lambda x:[stemmer.stem(i.lower()) for i in word_tokenize(x)]).apply(lambda x:' '.join(x))
nw_cap
```

```
0                                    didnt feel humili
1       from feel hopeless damn hope just from be arou...
2                         grab minut post feel greedi wrong
3       ever feel nostalg about fireplac will know tha...
4                                        feel grouchi
                              ...
17995    just keep feel like someon be unkind do wrong ...
17996    feel littl cranki negat after this doctor appoint
17997      feel that use peopl that give great feel achiev
17998    feel more comfort with derbi feel though start...
17999    feel weird when have meet peopl text like dont...
Name: Description, Length: 18000, dtype: object
```

```
#Remove stopwords

from nltk.corpus import stopwords
stop=stopwords.words('english')
nw_cap=nw_cap.apply(lambda x:[i for i in word_tokenize(x) if i not in stop]).apply(lambda x:' '.join(x))
nw_cap
```

```
0                                    didnt feel humili
1          feel hopeless damn hope around someon care awak
2                         grab minut post feel greedi wrong
3           ever feel nostalg fireplac know still properti
4                                        feel grouchi
                              ...
17995    keep feel like someon unkind wrong think back ...
17996               feel littl cranki negat doctor appoint
17997                feel use peopl give great feel achiev
17998      feel comfort derbi feel though start step shell
17999    feel weird meet peopl text like dont talk face...
Name: Description, Length: 18000, dtype: object
```

```
#vectorization
#to convert tokens to numeric data
from sklearn.feature_extraction.text import TfidfVectorizer
vec=TfidfVectorizer()
train_data=vec.fit_transform(nw_cap)
train_data
```

```
<18000x10134 sparse matrix of type '<class 'numpy.float64'>'
        with 148453 stored elements in Compressed Sparse Row format>
```

```
print(train_data)
```

```
  (0, 4253)     0.7885907572270544
  (0, 3201)     0.11919175887394268
  (0, 2377)     0.6032561166143257
  (1, 592)      0.484625418586205
  (1, 1297)     0.33654027188559155
  (1, 8237)     0.319609061463425
  (1, 443)      0.30810336480937944
  (1, 4196)     0.32026633692885687
  (1, 2113)     0.4315560148459396
  (1, 4197)     0.4025285834680238
  (1, 3201)     0.06293514482873948
  (2, 10036)    0.4028457592358073
  (2, 3801)     0.4465406948795003
  (2, 6804)     0.37704223770899853
  (2, 5657)     0.4524257602555041
  (2, 3756)     0.5353810397733767
  (2, 3201)     0.06952644658011282
  (3, 6971)     0.547193184372412
  (3, 8486)     0.25705547067358475
  (3, 4921)     0.23155742056059223
  (3, 3284)     0.560481466361797
  (3, 6091)     0.4033246147497133
  (3, 2999)     0.31691036035539205
  (3, 3201)     0.06033961764733965
  (4, 3831)     0.9908310492360172
  :     :
  (17996, 1967) 0.4247195564798564
  (17996, 5189) 0.2647755838897742
  (17996, 3201) 0.06512018011458856
  (17997, 70)   0.5600766336749307
```

```
  (17997, 3674) 0.4090150099845445
  (17997, 9549) 0.41543439418240746
  (17997, 6528) 0.3278411200160033
  (17997, 3796) 0.46282424603926753
  (17997, 3201) 0.15731836337132574
  (17998, 2290) 0.5355540949086142
  (17998, 7897) 0.49951109456455767
  (17998, 1703) 0.349974503819646
  (17998, 8467) 0.4016842416615563
  (17998, 8967) 0.29984151931550074
  (17998, 8434) 0.2746126453582357
  (17998, 3201) 0.12050333113416244
  (17999, 8900) 0.4091623443344274
  (17999, 9826) 0.3090013480242709
  (17999, 3111) 0.6312593525194524
  (17999, 5528) 0.34587453692075265
  (17999, 8789) 0.2868846177643161
  (17999, 6528) 0.2267594567785977
  (17999, 2557) 0.24568689960861495
  (17999, 5143) 0.14812837000289106
  (17999, 3201) 0.05440657751785149
```

```
x=train_data
x
```

```
<18000x10134 sparse matrix of type '<class 'numpy.float64'>'
        with 148453 stored elements in Compressed Sparse Row format>
```

```
y=df['Emotion']
y
```

```
0        1
1        1
2        2
3        3
4        2
        ..
17995    2
17996    2
17997    6
17998    6
17999    5
Name: Emotion, Length: 18000, dtype: int64
```

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

```
<12600x10134 sparse matrix of type '<class 'numpy.float64'>'
        with 103918 stored elements in Compressed Sparse Row format>
```

```
x_train.shape
```

```
(12600, 10134)
```

```
x_test.shape
```

```
(5400, 10134)
```

```
x.shape
```

```
(18000, 10134)
```

```
#show classsification report using all supervised models
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
svm_model=SVC()
nb_model=MultinomialNB()
rf_model=RandomForestClassifier()
de_model=DecisionTreeClassifier()
lstmodel=[svm_model,nb_model,rf_model,de_model]
```

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

```
print("*"*100)
print(classification_report(y_test,y_pred))
print("*"*100)
```

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

           1       0.86      0.91      0.88      1607
           2       0.86      0.72      0.79       685
           3       0.81      0.40      0.54       455
           4       0.78      0.45      0.57       198
           5       0.84      0.67      0.75       641
           6       0.74      0.93      0.83      1814

    accuracy                           0.81      5400
   macro avg       0.82      0.68      0.73      5400
weighted avg       0.81      0.81      0.80      5400

****************************************************************************************************
MultinomialNB()
****************************************************************************************************
              precision    recall  f1-score   support

           1       0.71      0.89      0.79      1607
           2       0.93      0.33      0.49       685
           3       0.94      0.06      0.12       455
           4       1.00      0.01      0.01       198
           5       0.90      0.25      0.39       641
           6       0.60      0.97      0.74      1814

    accuracy                           0.67      5400
   macro avg       0.84      0.42      0.42      5400
weighted avg       0.75      0.67      0.60      5400

****************************************************************************************************
RandomForestClassifier()
****************************************************************************************************
              precision    recall  f1-score   support

           1       0.89      0.88      0.89      1607
           2       0.83      0.81      0.82       685
           3       0.77      0.59      0.67       455
           4       0.67      0.74      0.71       198
           5       0.81      0.78      0.80       641
           6       0.83      0.89      0.86      1814

    accuracy                           0.83      5400
   macro avg       0.80      0.78      0.79      5400
weighted avg       0.83      0.83      0.83      5400

****************************************************************************************************
DecisionTreeClassifier()
****************************************************************************************************
              precision    recall  f1-score   support

           1       0.83      0.85      0.84      1607
           2       0.77      0.80      0.79       685
           3       0.61      0.58      0.60       455
           4       0.67      0.67      0.67       198
           5       0.78      0.78      0.78       641
           6       0.82      0.80      0.81      1814
```

```
dfe=pd.DataFrame()
dfe['Actual']=y_test
dfe['Predicted']=y_pred
dfe['Difference']=y_test-y_pred
dfe.sort_index()
```

1 to 25 of 5400 entries Filter

index	Actual	Predicted	Difference
0	1	1	0
3	3	3	0
5	1	1	0
8	6	6	0
14	6	6	0
15	6	6	0
19	5	5	0
20	2	2	0
27	1	4	-3
29	6	6	0
30	6	1	5
31	5	5	0
33	2	2	0
34	6	6	0

Double-click (or enter) to edit

42	6	6	0
46	6	1	5
47	3	3	0
50	6	6	0
54	6	6	0
57	4	3	1
59	6	3	3
61	3	3	0
63	5	2	3

Show 25 per page 1 2 10 100 200 210 216

Like what you see? Visit the data table notebook to learn more about interactive tables.

✓ 0s completed at 3:40 PM