```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
```

```
nltk.download('stopwords')
nltk.download('punkt')
```

```
    [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
```

```
df=pd.read_csv('/content/Emotion_classify_Data.csv',encoding='ISO-8859-1')
df
```

	Comment	Emotion
0	i seriously hate one subject to death but now ...	fear
1	im so full of life i feel appalled	anger
2	i sit here to write i start to dig out my feel...	fear
3	ive been really angry with r and i feel like a...	joy
4	i feel suspicious if there is no one outside l...	fear
...	...	...
5932	i begun to feel distressed for you	fear
5933	i left feeling annoyed and angry thinking that...	anger
5934	i were to ever get married i d have everything...	joy
5935	i feel reluctant in applying there because i w...	fear
5936	i just wanted to apologize to you because i fe...	anger

5937 rows × 2 columns

Next steps: Generate code with df View recommended plots

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

```
    Comment    0
    Emotion    0
    dtype: int64
```

```
df.dtypes
```

```
    Comment    object
    Emotion    object
    dtype: object
```

```
df['Emotion'].unique()
```

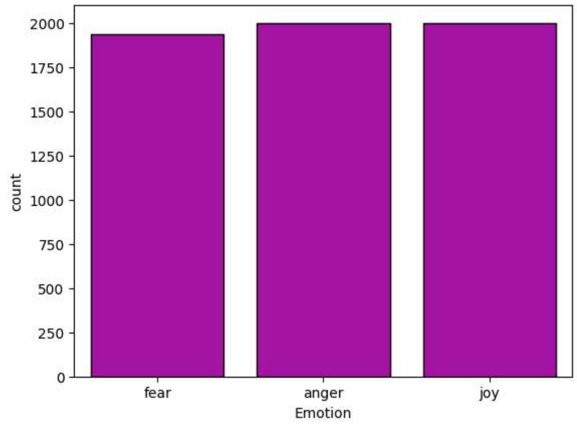
```
    array(['fear', 'anger', 'joy'], dtype=object)
```

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

```
    anger    2000
    joy      2000
    fear     1937
    Name: Emotion, dtype: int64
```

```
sns.countplot(x=df['Emotion'],data=df,color='m',edgecolor='k')
```

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



```
#Maping the samples under Emotion to numerical values
df['Emotion']=df['Emotion'].map({'anger':-1,'joy':1,'fear':0})
df['Emotion'].unique()
```

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

```
comment=df.Comment
comment
```

```
    0       i seriously hate one subject to death but now ...
    1                      im so full of life i feel appalled
    2       i sit here to write i start to dig out my feel...
```

```
3       ive been really angry with r and i feel like a...
4       i feel suspicious if there is no one outside l...
                              ...
5932                   i begun to feel distressed for you
5933    i left feeling annoyed and angry thinking that...
5934    i were to ever get married i d have everything...
5935    i feel reluctant in applying there because i w...
5936    i just wanted to apologize to you because i fe...
Name: Comment, Length: 5937, dtype: object
```

```
#Tokenize the comments(text)
from nltk.tokenize import TweetTokenizer
tk=TweetTokenizer()
comment=comment.apply(lambda x:tk.tokenize(x)).apply(lambda x:' '.join(x))
comment
```

```
0       i seriously hate one subject to death but now ...
1                      im so full of life i feel appalled
2       i sit here to write i start to dig out my feel...
3       ive been really angry with r and i feel like a...
4       i feel suspicious if there is no one outside l...
                              ...
5932                   i begun to feel distressed for you
5933    i left feeling annoyed and angry thinking that...
5934    i were to ever get married i d have everything...
5935    i feel reluctant in applying there because i w...
5936    i just wanted to apologize to you because i fe...
Name: Comment, Length: 5937, dtype: object
```

```
#Replace special characters
comment=comment.str.replace('[^a-zA-Z0-9]+',' ')
comment
```

```
<ipython-input-12-e171e0f89fd0>:2: FutureWarning: The default value of regex will change from True to False in a future version.
  comment=comment.str.replace('[^a-zA-Z0-9]+',' ')
0       i seriously hate one subject to death but now ...
1                      im so full of life i feel appalled
2       i sit here to write i start to dig out my feel...
3       ive been really angry with r and i feel like a...
4       i feel suspicious if there is no one outside l...
                              ...
5932                   i begun to feel distressed for you
5933    i left feeling annoyed and angry thinking that...
5934    i were to ever get married i d have everything...
5935    i feel reluctant in applying there because i w...
5936    i just wanted to apologize to you because i fe...
Name: Comment, Length: 5937, dtype: object
```

```
#Collect Meaningful words
comment=comment.apply(lambda x:[i for i in tk.tokenize(x) if len(i)>=3]).apply(lambda x:' '.join(x))
comment
```

```
0       seriously hate one subject death but now feel ...
1                                 full life feel appalled
2       sit here write start dig out feelings and thin...
3       ive been really angry with and feel like idiot...
4       feel suspicious there one outside like the rap...
                              ...
5932                        begun feel distressed for you
5933    left feeling annoyed and angry thinking that w...
5934    were ever get married have everything ready of...
5935    feel reluctant applying there because want abl...
5936    just wanted apologize you because feel like he...
Name: Comment, Length: 5937, dtype: object
```

```
#Stemming
from nltk.stem import SnowballStemmer
sm=SnowballStemmer('english')
comment=comment.apply(lambda x:[sm.stem(i.lower()) for i in tk.tokenize(x)]).apply(lambda x:' '.join(x))
comment
```

```
0       serious hate one subject death but now feel re...
1                                    full life feel appal
2       sit here write start dig out feel and think th...
3       ive been realli angri with and feel like idiot...
4       feel suspici there one outsid like the raptur ...
                              ...
5932                          begun feel distress for you
5933    left feel annoy and angri think that was the c...
5934    were ever get marri have everyth readi offer h...
5935    feel reluct appli there becaus want abl find c...
5936    just want apolog you becaus feel like heartles...
Name: Comment, Length: 5937, dtype: object
```

```
#Remove stop words
from nltk.corpus import stopwords
stop=stopwords.words('english')
comment=comment.apply(lambda x:[i for i in tk.tokenize(x) if i not in stop]).apply(lambda x:' '.join(x))
comment
```

```
0         serious hate one subject death feel reluct drop
1                                    full life feel appal
2       sit write start dig feel think afraid accept p...
3       ive realli angri feel like idiot trust first p...
4       feel suspici one outsid like raptur happen someth
                              ...
5932                                  begun feel distress
5933       left feel annoy angri think center stupid joke
5934    ever get marri everyth readi offer becaus got ...
5935    feel reluct appli becaus want abl find compani...
5936         want apolog becaus feel like heartless bitch
Name: Comment, Length: 5937, dtype: object
```

```
#Vectorization
from sklearn.feature_extraction.text import TfidfVectorizer
vec=TfidfVectorizer()
data=vec.fit_transform(comment)
print(data)
```

```
  (0, 1591)     0.43420650932123844
  (0, 4373)     0.33620763506591916
  (0, 1941)	0.06171955131947581
  (0, 1305)	0.43420650932123844
  (0, 5168)	0.4451952347765912
  (0, 3723)	0.26141211626069766
  (0, 2436)	0.29248647578357173
  (0, 4684)	0.3924297485187133
  (1, 257)	0.6781252855536237
  (1, 3107)	0.42132200002671333
  (1, 2154)	0.5946117082381293
  (1, 1941)	0.09524067309947233
  (2, 3236)	0.22339759099383713
  (2, 3375)	0.3209635533048003
  (2, 4046)	0.3481591157665799
  (2, 28)	0.3250383459859768
  (2, 98)	0.3055508325481411
  (2, 5396)	0.23454510010769763
  (2, 1435)	0.4607268115268175
  (2, 5072)	0.25799078720365554
  (2, 6069)	0.28604357777911876
  (2, 4809)	0.3282976807622391
  (2, 1941)	0.05839319785621638
  (3, 3979)	0.3689044968268168
  (3, 1998)	0.3583244748964783
  :	:
  (5934, 1804)	0.1655354126877792
  (5934, 461)	0.14080641044157638
  (5934, 2373)	0.23652890533730583
  (5934, 5289)	0.20234549192969004
  (5934, 2286)	0.18769111194631122
  (5934, 1941)	0.03967059468675665
  (5935, 262)	0.4350663044128318
  (5935, 1027)	0.41272921666459295
  (5935, 3916)	0.27383212289416026
  (5935, 3058)	0.32656974661947213
  (5935, 10)	0.3038072510673401
  (5935, 461)	0.20023511289456317
  (5935, 1986)	0.28232428952674155
  (5935, 2978)	0.21363842814370324
  (5935, 5878)	0.21514325715891092
  (5935, 4373)	0.30730620692923544
  (5935, 1941)	0.05641395147270715
  (5935, 3723)	0.23894033779949272
  (5936, 536)	0.5117767204404504
  (5936, 254)	0.5492249029254934
  (5936, 2463)	0.4836998913121322
  (5936, 461)	0.26645584078231815
  (5936, 5878)	0.28629432993158427
  (5936, 3116)	0.20945694915610483
  (5936, 1941)	0.07507088369375213
```

```
#separate x any y then proceed with supervised ML algorithm
x=data
```

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

```
array([ 0, -1,  0, ...,  1,  0, -1])
```

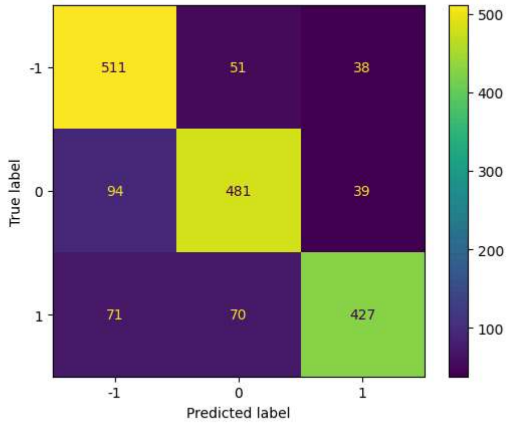
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
y_train
```

```
array([-1,  0,  0, ..., -1,  0,  0])
```

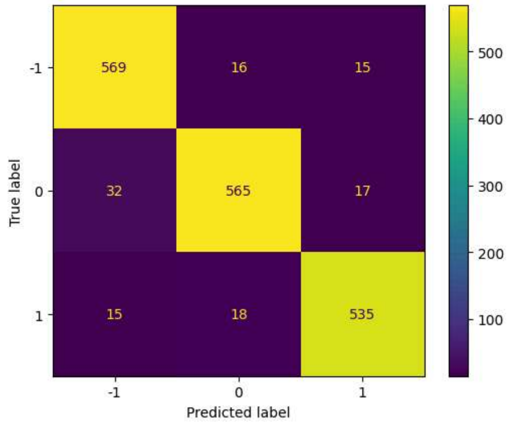
```
#Model creation
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import BernoulliNB
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report,ConfusionMatrixDisplay
model1=KNeighborsClassifier(n_neighbors=7)
model2=DecisionTreeClassifier(criterion='entropy')
model3=RandomForestClassifier(n_estimators=10,criterion='entropy',random_state=42)
model4=SVC()
model5=BernoulliNB()
lst=[model1,model2,model3,model4,model5]
```

```
for i in lst:
  print("model is",i)
  i.fit(x_train,y_train)
  y_pred=i.predict(x_test)
  cm=confusion_matrix(y_test,y_pred)
  print("Accuracy score is",accuracy_score(y_test,y_pred))
  print(cm)
  labels=[-1,0,1]
  print(classification_report(y_test,y_pred))
  cmd=ConfusionMatrixDisplay(cm,display_labels=labels)
  cmd.plot()
  plt.show()
```

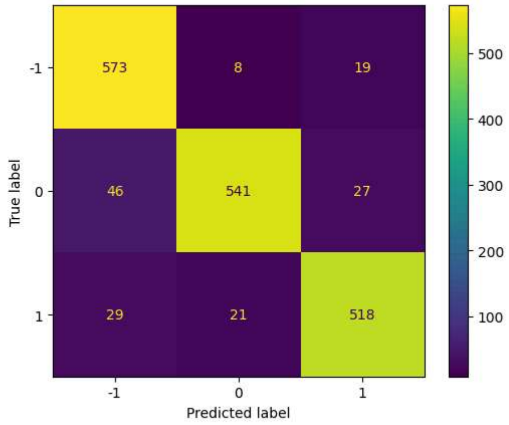
```
model is KNeighborsClassifier(n_neighbors=7)
Accuracy score is 0.7962962962962963
[[511  51  38]
 [ 94 481  39]
 [ 71  70 427]]
              precision    recall  f1-score   support

          -1       0.76      0.85      0.80       600
           0       0.80      0.78      0.79       614
           1       0.85      0.75      0.80       568

    accuracy                           0.80      1782
   macro avg       0.80      0.80      0.80      1782
weighted avg       0.80      0.80      0.80      1782
```



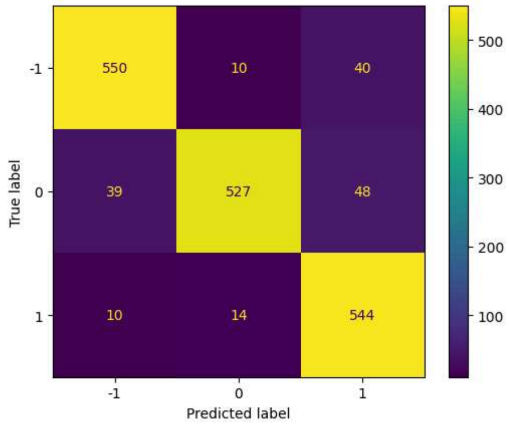
```
model is DecisionTreeClassifier(criterion='entropy')
Accuracy score is 0.9365881032547699
[[569  16  15]
 [ 32 565  17]
 [ 15  18 535]]
              precision    recall  f1-score   support

          -1       0.92      0.95      0.94       600
           0       0.94      0.92      0.93       614
           1       0.94      0.94      0.94       568

    accuracy                           0.94      1782
   macro avg       0.94      0.94      0.94      1782
weighted avg       0.94      0.94      0.94      1782
```



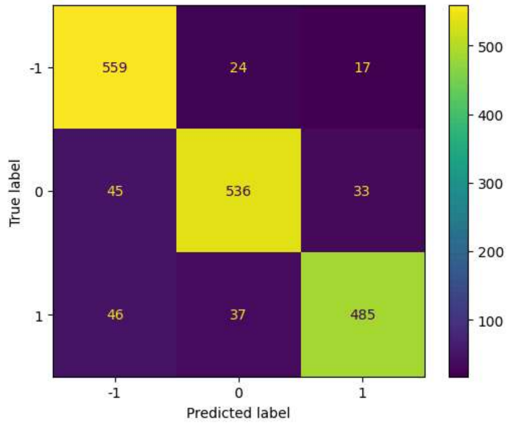
```
model is RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=42)
Accuracy score is 0.9158249158249159
[[573   8  19]
 [ 46 541  27]
 [ 29  21 518]]
              precision    recall  f1-score   support

          -1       0.88      0.95      0.92       600
           0       0.95      0.88      0.91       614
           1       0.92      0.91      0.92       568

    accuracy                           0.92      1782
   macro avg       0.92      0.92      0.92      1782
weighted avg       0.92      0.92      0.92      1782
```



```
model is SVC()
model is SVC()
Accuracy score is 0.909652076318743
[[550  10  40]
 [ 39 527  48]
 [ 10  14 544]]
              precision    recall  f1-score   support

          -1       0.92      0.92      0.92       600
           0       0.96      0.86      0.90       614
           1       0.86      0.96      0.91       568

    accuracy                           0.91      1782
   macro avg       0.91      0.91      0.91      1782
weighted avg       0.91      0.91      0.91      1782
```



```
model is BernoulliNB()
Accuracy score is 0.8866442199775533
[[559  24  17]
 [ 45 536  33]
 [ 46  37 485]]
              precision    recall  f1-score   support

          -1       0.86      0.93      0.89       600
           0       0.90      0.87      0.89       614
           1       0.91      0.85      0.88       568

    accuracy                           0.89      1782
   macro avg       0.89      0.89      0.89      1782
weighted avg       0.89      0.89      0.89      1782
```


True label
Predicted label