Input

```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import re
```

```
df=pd.read_csv('Text_Emotion.csv')
df
```

	text	emotion
0	carefully word blog posts amount criticism hea...	😟
1	cannot remember little mermaid feeling carefre...	😊
2	not feeling super well turns cold knocked next...	😊
3	feel honored part group amazing talents	😊
4	think helping also began feel pretty lonely lo...	😟
...	...	...
60518	mobileech looking new sms sending solution get...	😟
60519	kind feeling little bit lost world moment some...	😟
60520	knew	😊
60521	well feel unloved unappreciated	😟
60522	like	NaN

60523 rows × 2 columns

Input

```
df.text=df.text.str.lower()
```

```
def fix_punctuation(text):
  return re.sub("`","'",text)
df['text']=df['text'].astype(str).apply(lambda x: fix_punctuation(x))
```

```
!pip install contractions --quiet
import contractions
def fix_contraction(text):
  return contractions.fix(text)
df['text']=df['text'].astype(str).apply(lambda x: fix_contraction(x))
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 289.9/289.9 kB 4.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 110.8/110.8 kB 8.6 MB/s eta 0:00:00
```

```
def cleaning(text):
  text=re.sub('[^a-zA-Z0-9]|https?://\S+|www.\S+|<.*?>',"",text)
  text=re.sub('\s+','',text)
  return text
df['text']=df['text'].astype(str).apply(lambda x: cleaning(x))
```

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

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

```
from nltk.corpus import stopwords
s=list(stopwords.words('english'))
s.remove('no')
s.remove('nor')
s.remove('not')
s=set(s)
def rem_s(text):
  return " ".join([word for word in text.split()if word not in s])
df['text']=df['text'].astype(str).apply(lambda x: rem_s(x))


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

```
text       0
emotion    1
dtype: int64
```

```
df.dropna(inplace=True)


df.duplicated().sum()
```

```
2
```

```
df.drop_duplicates(inplace=True)


x=df['text']
y=df['emotion']


from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split


cv=CountVectorizer(lowercase=True,ngram_range=(1,1))


X_ml=cv.fit_transform(x)
X_ml.shape
```

```
(60520, 60517)
```

```
X_train, X_test, y_train, y_test = train_test_split(X_ml,y,test_size=0.1,
                                                    random_state=0)
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((54468, 60517), (6052, 60517), (54468,), (6052,))
```

```
from sklearn.neural_network import MLPClassifier
model=MLPClassifier(activation='logistic', alpha=0.01, batch_size=64,
                    early_stopping=True, hidden_layer_sizes=(128),
                    learning_rate='adaptive', max_iter=1000, shuffle=True)
```

```
model.fit(X_train,y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:693: UserWarning: Training interrupted by user
  warnings.warn("Training interrupted by user.")
```

```
MLPClassifier
MLPClassifier(activation='logistic', alpha=0.01, batch_size=64,
              early_stopping=True, hidden_layer_sizes=128,
              learning_rate='adaptive', max_iter=1000)
```

```
y_pred=model.predict(X_test)
```

```
from sklearn.metrics import (classification_report, confusion_matrix,
                             accuracy_score)
```

```
confusion_matrix(y_test,y_pred)
```

```
array([[   0, 2887],
       [   0, 3165]])
```

```
print(classification_report(y_test,y_pred))
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           ☹       0.00      0.00      0.00      2887
           🙂       0.52      1.00      0.69      3165

    accuracy                           0.52      6052
   macro avg       0.26      0.50      0.34      6052
weighted avg       0.27      0.52      0.36      6052

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-
  _warn_prf(average, modifier, msg_start, len(result))
```

```
accuracy_score(y_test,y_pred)
```

```
0.5229676140118968
```