

```
In [15]: import numpy as np
import pandas as pd
df=pd.read_csv("emotion.csv")
data
```

```
Out[15]:
```

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

16000 rows × 2 columns

```
In [16]: df.shape
```

```
Out[16]: (16000, 2)
```

```
In [17]: df.info()
```

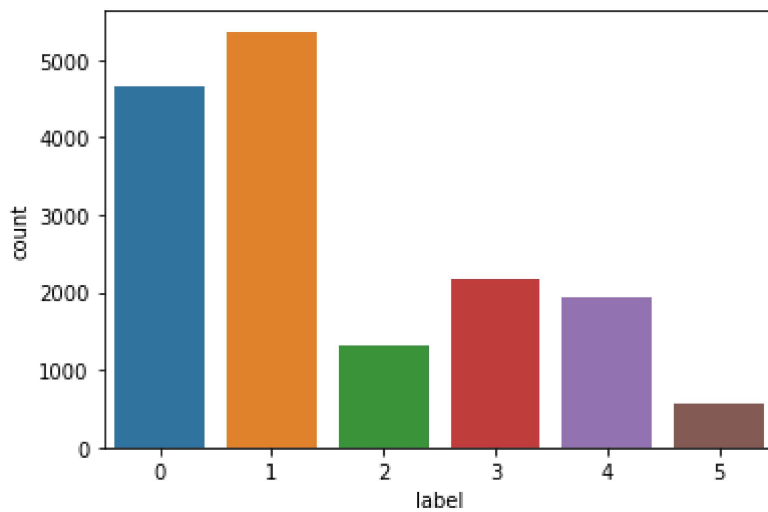
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16000 entries, 0 to 15999
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype
---  -
0    text    16000 non-null    object
1    label    16000 non-null    int64
dtypes: int64(1), object(1)
memory usage: 250.1+ KB
```

```
In [20]: df.label.value_counts()
```

```
Out[20]: 1    5362
0    4666
3    2159
4    1937
2    1304
5     572
Name: label, dtype: int64
```

```
In [21]: import seaborn as sns
sns.countplot(x=df.label)
```

```
Out[21]: <AxesSubplot:xlabel='label', ylabel='count'>
```



```
In [22]: df.isna().sum
```

```
Out[22]: <bound method DataFrame.sum of          text  label
0         False  False
1         False  False
2         False  False
3         False  False
4         False  False
...         ...    ...
15995      False  False
15996      False  False
15997      False  False
15998      False  False
15999      False  False

[16000 rows x 2 columns]>
```

```
In [25]: #converting text into lower case
df['text']=df['text'].apply(lambda x:"".join(x.lower() for x in x.split()))
```

```
In [26]: import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to C:\Users\MRUH
[nltk_data] 37\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!
```

```
Out[26]: True
```

```
In [32]: #Filter stop words
from nltk.corpus import stopwords
stop=stopwords.words('english')
df['text']=df['text'].apply(lambda x:"".join (x for x in x.split() if x not in stop))
```

```
In [35]: # Do Lemmatization

from nltk.stem import WordNetLemmatizer
from textblob import Word
df['text'] = df['text'].apply(lambda x: " ".join([Word(word).lemmatize() for word in x.
df['text'].head()
```

```
Out[35]: 0          ididntfeelhumiliated
1  icangofromfeelingsohopelesstosodamnedhopefulju...
2          imgrabbingaminutetopostifeelgreedywrong
```

```
3     iameverfeelingnostalgicaboutthefireplaceiwillk...
4                                     iamfeelinggrouchy
Name: text, dtype: object
```

```
In [37]: from sklearn.feature_extraction.text import TfidfVectorizer
         tfidf = TfidfVectorizer()
         X = tfidf.fit_transform(df['text'])
         X = X.toarray()
         y = df.label.values
```

```
In [ ]:
```