```
------------------------------------------------------------

import nlp_utils
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer,
TfidfVectorizer


------------------------------------------------------------

df=pd.read_csv('train.csv')


------------------------------------------------------------
df.shape



------------------------------------------------------------------------
pd.set_option('display.max_colwidth', -1)

df['title']

df['text']



------------------------------------------------------------------

df['label'].value_counts()


------------------------------------------------------------------

df.isnull().sum()


------------------------------------------------------------------
df=df.dropna()





------------------------------------------------
df.reset_index(inplace=True)



_____________________________________________________________________________
_______________________________

import re
import string
```

```
_____________________________________________________________________________
_________________________

alphanumeric = lambda x: re.sub('\w*\d\w*', ' ', x)



_____________________________________________________________________________
_____________________________________________________


punc_lower = lambda x: re.sub('[%s]' % re.escape(string.punctuation), '
', x.lower())




_____________________________________________________________________________
_____________________________________________


remove_n = lambda x: re.sub("\n", " ", x)


_____________________________________________________________________________
_____________________________________


remove_non_ascii = lambda x: re.sub(r'[^\x00-\x7f]',r' ', x)


_____________________________________________________________________________
_____________________________________


df['text'] =
df['text'].map(alphanumeric).map(punc_lower).map(remove_n).map(remove_non
_ascii)



df['text']




_____________________________________________________________________________
_________________________


import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop_words=stopwords.words('english')
#DataFrame.apply(Function_to_apply_to_each_row)
def rem_stopword(data):
  li=[]
  for w in data.split():
    if w not in stop_words:
      li.append(w)
  return " ".join(li)
```

```
_____
data="All the students of Third Year CSM are studying NLP "
print(rem_stopword(data))
```

```
_____
```

```
from nltk.stem.porter import PorterStemmer
import re
ps = PorterStemmer()
corpus = []
for i in range(0, len(df)):
    review = re.sub('[^a-zA-Z]', ' ', df['text'][i])
    review = review.lower()
    review = review.split()

    review = [ps.stem(word) for word in review if not word in
stopwords.words('english')]
    review = ' '.join(review)
    corpus.append(review)
```

```
_____
```

```
Y=df['label']
```

```
_____
```

```
X_train, X_test, Y_train, Y_test = train_test_split(df['text'], Y,
test_size=0.30, random_state=40)
```

```
_____
_____


tfidf_vect = TfidfVectorizer(stop_words = 'english',max_df=0.7)
tfidf_train = tfidf_vect.fit_transform(X_train)
tfidf_test = tfidf_vect.transform(X_test)



_____
_____


count_vect = CountVectorizer(stop_words = 'english')
count_train = count_vect.fit_transform(X_train.values)
count_test = count_vect.transform(X_test.values)




_____
_____

from sklearn.naive_bayes import MultinomialNB

from sklearn import metrics
from sklearn.metrics import accuracy_score


clf = MultinomialNB()
clf.fit(tfidf_train, Y_train)
pred = clf.predict(tfidf_test)
score = metrics.accuracy_score(Y_test, pred)
print("accuracy:   %0.3f" % score)
cm = metrics.confusion_matrix(Y_test, pred)
print(cm)
```

```
clf = MultinomialNB()
clf.fit(count_train, Y_train)
pred1 = clf.predict(count_test)
score = metrics.accuracy_score(Y_test, pred1)
print("accuracy:   %0.3f" % score)
cm2 = metrics.confusion_matrix(Y_test, pred1)
print(cm2)
```