```
import pandas as pd
import re

train=pd.read_csv("train.csv")

train.head()

# drop col 'id' (as it is of no use) and replace it in the same variable
train.drop("id",inplace=True,axis=1)

train.head()

temp = train.groupby("label").size()
temp

import nltk
#nltk.download()

from nltk.stem import PorterStemmer
stemmer = PorterStemmer()

def clean_sentences(text):
    text = text.lower()   # convert text to lower case
    text = re.sub(r"[^a-z0-9^,!.\/']", " ", text)    # remove special char's
    text = " ".join(text.split())
    text = " ".join(stemmer.stem(word) for word in text.split())   # do stemming
    return text

x = train['tweet']
y = train['label']

x = x.map(lambda a: clean_sentences(a))

x.head()

pip install sklearn

from sklearn.model_selection import train_test_split

 # split the dataset into training set & testing set
 # data is split in a stratified fashion
x_train, x_test, y_train, y_test = train_test_split(x,y,stratify=y,random_state=42)

x_train.head()

from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(stop_words='english')

x_train = vectorizer.fit_transform(x_train)

x_test = vectorizer.transform(x_test)

from sklearn.svm import LinearSVC

model = LinearSVC(C=1.05, tol=0.5)

model.fit(x_train,y_train)

from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, f1_score, recall_score
confusion_matrix(y_test,model.predict(x_test))

accuracy_score(y_test,model.predict(x_test))

recall_score(y_test,model.predict(x_test))

precision_score(y_test,model.predict(x_test))

f1_score(y_test,model.predict(x_test))

# sample_text = ['I hate you']
# sample_text = ['I dont hate you']
sample_text = ['you are a bad person.']
sample_text = list(map(lambda a: clean_sentences(a), sample_text))
sample_text

sample_text = vectorizer.transform(sample_text)

model.predict(sample_text)[0]
```

✓ 2s completed at 15:28 ● ×