In [1]:

```
import pandas as pd
import re
```

In [2]:

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

In [3]:

```
train.head()
```

Out[3]:

	id	label	tweet
0	1	0	@user when a father is dysfunctional and is s...
1	2	0	@user @user thanks for #lyft credit i can't us...
2	3	0	bihday your majesty
3	4	0	#model i love u take with u all the time in ...
4	5	0	factsguide: society now #motivation

In [4]:

```
train.drop("id",inplace=True,axis=1)
```

In [5]:

```
import nltk
nltk.download()
```

```
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/inde
x.xml (https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml)
```

Out[5]:

```
True
```

In [6]:

```


from nltk.stem import PorterStemmer
stemmer = PorterStemmer()

def clean_sentences(text):
    text = text.lower()
    text = re.sub(r"[^a-z0-9^,!.\/']", " ", text)
    text = " ".join(text.split())
    text = " ".join(stemmer.stem(word) for word in text.split())
    return text
```

In [7]:

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

In [8]:

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

In [9]:

```
x.head()
```

Out[9]:

```
0    user when a father is dysfunct and is so selfi...
1    user user thank for lyft credit i can't use ca...
2                                bihday your majesti
3    model i love u take with u all the time in ur !!!
4                            factsguid societi now motiv
Name: tweet, dtype: object
```

In [10]:

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

In [11]:

```
x_train.head()
```

Out[11]:

```
1036     user like the spread of peanut butter on white...
2380     watch made in america o.j. simpson..... 30for3...
31605             franci underwood seen leav marseil nojok
23437    get up get get enjoy music today free app free...
2669     my 1st juic experience! notsobad healthyliv ea...
Name: tweet, dtype: object
```

In [12]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [13]:

```
vectorizer = TfidfVectorizer(stop_words='english')
```

In [14]:

```
x_train = vectorizer.fit_transform(x_train)
```

In [15]:

```
x_test = vectorizer.transform(x_test)
```

In [16]:

```
from sklearn.svm import LinearSVC
```

In [17]:

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

In [18]:

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

Out[18]:

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

In [19]:

```

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

Out[19]:

```
array([[7369,   61],
       [ 227,  334]], dtype=int64)
```

In [20]:

```
accuracy_score(y_test,model.predict(x_test))
```

Out[20]:

```
0.9639594543861845
```

In [21]:

```
recall_score(y_test,model.predict(x_test))
```

Out[21]:

```
0.5953654188948306
```

In [22]:

```
precision_score(y_test,model.predict(x_test))
```

Out[22]:

```
0.8455696202531645
```

In [23]:

```
f1_score(y_test,model.predict(x_test))
```

Out[23]:

```
0.698744769874477
```

In [ ]:

```
```