In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
data = pd.read_csv('Tweets.csv')
```

In [3]:
```
data.head()
```

Out[3]:

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	airline	airline_sentiment_gold
0	570306133677760513	neutral	1.0000	NaN	NaN	Virgin America	NaN
1	570301130888122368	positive	0.3486	NaN	0.0000	Virgin America	NaN
2	570301083672813571	neutral	0.6837	NaN	NaN	Virgin America	NaN
3	570301031407624196	negative	1.0000	Bad Flight	0.7033	Virgin America	NaN
4	570300817074462722	negative	1.0000	Can't Tell	1.0000	Virgin America	NaN

In [4]:
```
data = data[['airline_sentiment','text']]
```

In [5]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

In [39]:
```
cv = CountVectorizer(max_df = 0.95,min_df = 10)
```

In [40]:
```
from nltk.stem import SnowballStemmer
from nltk.tokenize import word_tokenize



def remove_punc(string):
    punc = '''!()-[]{};:'"\,<>./?@#$%^&*_~'''
    for char in string:
        if char in punc:
            string = string.replace(char,"")
    return string

def stem_text(string):
    ps = SnowballStemmer(language = 'english')
    words = word_tokenize(string)
    sentence = []
    for word in words:
        sentence.append(ps.stem(word))
    return " ".join(sentence)

def lower(string):
    return string.lower()



def clean_text(string):
    string = remove_punc(string)
    string = stem_text(string)
    return string.lower()
```

In [41]:
```
clean_text(data['text'][1])
```

Out[41]:
```
'virginamerica plus youv ad commerci to the experi tacki'
```

In [42]:
```
data['text'] = data['text'].apply(clean_text)
```

In [43]:
```
data.head()
```

Out[43]:

	airline_sentiment	text
0	neutral	virginamerica what dhepburn said
1	positive	virginamerica plus youv ad commerci to the exp...
2	neutral	virginamerica i didnt today must mean i need t...
3	negative	virginamerica it realli aggress to blast obnox...
4	negative	virginamerica and it a realli big bad thing ab...

In [44]:
```
X_matrix = cv.fit_transform(data['text'])
```

In [46]:
```
X_matrix
```

Out[46]:
```
<14640x1645 sparse matrix of type '<class 'numpy.int64'>'
        with 208656 stored elements in Compressed Sparse Row format>
```

In [47]:
```
count_vect_df = pd.DataFrame(X_matrix.todense(), columns=cv.get_feature_names())
```

In [48]:
```
count_vect_df
```

Out[48]:

	10	100	1000	11	12	13	130	14	140	15	...	york	you	youd	youll	your	youv	yr	yyz	zero	zone
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	1	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
14635	0	0	0	0	0	0	0	0	0	0	...	0	1	0	0	0	0	0	0	0	0
14636	0	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	0
14637	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
14638	0	0	0	0	0	0	0	0	0	0	...	0	2	0	0	1	0	0	0	0	0
14639	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

14640 rows × 1645 columns

In [49]:
```
df = pd.concat([data, count_vect_df], axis=1)
```

In [50]:
```
df.head()
```

Out[50]:

	airline_sentiment	text	10	100	1000	11	12	13	130	14	...	york	you	youd	youll	your	youv	yr	yyz	zero	zone
0	neutral	virginamerica what dhepburn said	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	positive	virginamerica plus youv ad commerci to the exp...	0	0	0	0	0	0	0	0	...	0	0	0	0	0	1	0	0	0	0
2	neutral	virginamerica i didnt today must mean i need t...	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	negative	virginamerica it realli aggress to blast obnox...	0	0	0	0	0	0	0	0	...	0	0	0	0	1	0	0	0	0	0
4	negative	virginamerica and it a realli big bad thing ab...	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 1647 columns

In [20]:
```
df.shape
```

Out[20]:
```
(14640, 13925)
```

In [51]:
```
df.drop('text',1,inplace  =True)
```

In [52]:
```
df['airline_sentiment'].value_counts()
```

Out[52]:
```
negative    9178
neutral     3099
positive    2363
Name: airline_sentiment, dtype: int64
```

In [53]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
```

In [54]:
```
df.head()
```

Out[54]:

	airline_sentiment	10	100	1000	11	12	13	130	14	140	...	york	you	youd	youll	your	youv	yr	yyz	zero	zone
0	neutral	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	positive	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	1	0	0	0	0
2	neutral	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	negative	0	0	0	0	0	0	0	0	0	...	0	0	0	0	1	0	0	0	0	0
4	negative	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 1645 columns

In [56]:
```
X_train,X_test,y_train,y_test = train_test_split(df.drop('airline_sentiment',1),df['airline_sentiment'],stratify = d
```

In [57]:
```
lm = LogisticRegression()
```

In [58]:
```
lm.fit(X_train,y_train)
```

```
C:\Users\yashm\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:765: ConvergenceWarning: lbfgs failed to c
onverge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-learn.org/stable/modules/preprocessing.h
tml)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (https://scikit-learn.org/stable/modu
les/linear_model.html#logistic-regression)
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

Out[58]:
```
LogisticRegression()
```

In [64]:
```
print("The testing Classification report:\n\n " ,classification_report(lm.predict(X_test),y_test))
print("The training Classification report:\n\n " ,classification_report(lm.predict(X_train),y_train))

```

```
The testing Classification report:

                precision    recall  f1-score   support

    negative       0.89      0.86      0.87      2380
     neutral       0.61      0.63      0.62       745
    positive       0.66      0.73      0.69       535

    accuracy                           0.79      3660
   macro avg       0.72      0.74      0.73      3660
weighted avg       0.80      0.79      0.80      3660

The training Classification report:

                precision    recall  f1-score   support

    negative       0.95      0.91      0.93      7126
     neutral       0.75      0.80      0.78      2184
    positive       0.82      0.87      0.85      1670

    accuracy                           0.89     10980
   macro avg       0.84      0.86      0.85     10980
weighted avg       0.89      0.89      0.89     10980
```

In [ ]:
```

```