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 [7]:
```
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
```

In [10]:
```
cv = TfidfVectorizer()
```

In [11]:

```
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 [13]:

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

In [14]:

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

In [15]:

```
count_vect_df = pd.DataFrame(X_matrix.todense(), columns=cv.get_feature_names())
```

In [16]:

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

In [17]:

```
df.head()
```

Out[17]:

	airline_sentiment	text	00	0011	0016	006	0162389030167	0162424965446	0162431184663	0167560070877	...	zj76	zkatcher	zombi	z
0	neutral	virginamerica what dhepburn said	0.0	0.0	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	0.0	0.0	...	0.0	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	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	0.0	0.0	...	0.0	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	0.0	0.0	

5 rows × 13925 columns

In [18]:

```
df.drop('text',1,inplace  =True)
```

In [20]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
```

In [21]:

```
test,y_train,y_test = train_test_split(df.drop('airline_sentiment',1),df['airline_sentiment'],stratify = df['airline_sent
```

In [22]:
```
rf = RandomForestClassifier()
```

In [23]:
```
rf.fit(X_train,y_train)
```

Out[23]:
```
RandomForestClassifier()
```

In [25]:
```
print("The testing Classification report:\n\n " ,classification_report(rf.predict(X_test),y_test))
print("The training Classification report:\n\n " ,classification_report(rf.predict(X_train),y_train))

```

```
The testing Classification report:

                precision    recall  f1-score   support

    negative       0.97      0.75      0.84      2970
     neutral       0.35      0.66      0.45       405
    positive       0.41      0.84      0.55       285

    accuracy                           0.74      3660
   macro avg       0.57      0.75      0.61      3660
weighted avg       0.85      0.74      0.78      3660

The training Classification report:

                precision    recall  f1-score   support

    negative       1.00      1.00      1.00      6885
     neutral       0.99      1.00      1.00      2312
    positive       1.00      0.99      0.99      1783

    accuracy                           1.00     10980
   macro avg       1.00      1.00      1.00     10980
weighted avg       1.00      1.00      1.00     10980
```

In [ ]:
```

```