In [1]:
```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer,ENGLISH_STOP_WORDS
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB,BernoulliNB
from sklearn.metrics import accuracy_score, classification_report
from nltk.corpus import stopwords
```

In [2]:
```
data = pd.read_csv('Sentiment140.csv', encoding='latin-1', header=None)
data
```

Out[2]:

	0	1	2	3	4	5
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....
...	...	...	...	...	...	...
1599995	4	2193601966	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	AmandaMarie1028	Just woke up. Having no school is the best fee...
1599996	4	2193601969	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	TheWDBoards	TheWDB.com - Very cool to hear old Walt interv...
1599997	4	2193601991	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	bpbabe	Are you ready for your MoJo Makeover? Ask me f...
1599998	4	2193602064	Tue Jun 16 08:40:49 PDT 2009	NO_QUERY	tinydiamondz	Happy 38th Birthday to my boo of alll time!!! ...
1599999	4	2193602129	Tue Jun 16 08:40:50 PDT 2009	NO_QUERY	RyanTrevMorris	happy #charitytuesday @theNSPCC @SparksCharity...

1600000 rows × 6 columns

In [3]:
```
data.columns = ['target', 'id', 'date', 'flag', 'user', 'text']
data.columns
```

Out[3]:
```
Index(['target', 'id', 'date', 'flag', 'user', 'text'], dtype='object')
```

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

Out[4]:

	target	text
0	0	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	is upset that he can't update his Facebook by ...
2	0	@Kenichan I dived many times for the ball. Man...
3	0	my whole body feels itchy and like its on fire
4	0	@nationwideclass no, it's not behaving at all....
...	...	...
1599995	4	Just woke up. Having no school is the best fee...
1599996	4	TheWDB.com - Very cool to hear old Walt interv...
1599997	4	Are you ready for your MoJo Makeover? Ask me f...
1599998	4	Happy 38th Birthday to my boo of alll time!!! ...
1599999	4	happy #charitytuesday @theNSPCC @SparksCharity...

1600000 rows × 2 columns

In [5]:
```
data['target'] = data['target'].apply(lambda x: 0 if x == 0 else 1)
```

```
<ipython-input-5-46c780893f34>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  data['target'] = data['target'].apply(lambda x: 0 if x == 0 else 1)
```

In [6]:
```
X_train, X_test, y_train, y_test = train_test_split(data['text'], data['target'], test_size=0.
X_train, X_test, y_train, y_test
```

Out[6]:
```
(1374558    @jbtaylor WIth ya. &quot;I'd like a Palm Pre, ...
 1389115    felt the earthquake this afternoon, it seems t...
 1137831          Ruffles on shirts are like so in, me Likey
 790714     Pretty bad night into a crappy morning....FML!...
 1117911               @dcbriccetti yeah, what a clear view!
                                 ...
 259178     this song's middle change just doesn't want to...
 1414414                 @officialnjonas Good luck with that
 131932               @ProudGamerTweet I rather average 32370
 671155     Pickin up @misstinayao waitin on @sadittysash ...
 121958     @ home studying for maths wooot ! im so going ...
 Name: text, Length: 1280000, dtype: object,
 541200                  @chrishasboobs AHHH I HOPE YOUR OK!!!
 750        @misstoriblack cool , i have no tweet apps  fo...
 766711     @TiannaChaos i know  just family drama. its la...
 285055     School email won't open  and I have geography ...
 705995                                 upper airways problem
                                 ...
 839535     @crowsond You will have to very careful what y...
 1023175                     Busy weekend of photo shoots!!!!
 1349377    @InterweaveNews Thanks for the link, that -is-...
 1086942    Grounded for the weekend. But really... I dont...
 158976           @BlipUp It uploads but gives a broken link
 Name: text, Length: 320000, dtype: object,
 1374558    1
 1389115    1
 1137831    1
 790714     0
 1117911    1
           ..
 259178     0
 1414414    1
 131932     0
 671155     0
 121958     0
 Name: target, Length: 1280000, dtype: int64,
 541200     0
 750        0
 766711     0
 285055     0
 705995     0
           ..
 839535     1
 1023175    1
 1349377    1
 1086942    1
 158976     0
 Name: target, Length: 320000, dtype: int64)
```

In [7]:
```
vectorizer = CountVectorizer(max_features=5000,stop_words='english')
vectorizer
```

Out[7]:
```
CountVectorizer(max_features=5000, stop_words='english')
```

In [8]:
```
X_train_vec = vectorizer.fit_transform(X_train)
X_train_vec
```

Out[8]:
```
<1280000x5000 sparse matrix of type '<class 'numpy.int64'>'
        with 6922955 stored elements in Compressed Sparse Row format>
```

In [9]:
```
X_test_vec = vectorizer.transform(X_test)
X_test_vec
```

Out[9]:
```
<320000x5000 sparse matrix of type '<class 'numpy.int64'>'
        with 1730354 stored elements in Compressed Sparse Row format>
```

In [10]:
```
nb_classifier = MultinomialNB()
nb_classifier
```

Out[10]:
```
MultinomialNB()
```

In [11]:
```
nb_classifier.fit(X_train_vec, y_train)
```

Out[11]:
```
MultinomialNB()
```

In [12]:
```
y_pred = nb_classifier.predict(X_test_vec)
y_pred
```

Out[12]:
```
array([1, 1, 1, ..., 1, 0, 0], dtype=int64)
```

In [13]:
```
accuracy = accuracy_score(y_test, y_pred)
accuracy
```

Out[13]:
```
0.75525
```

In [14]:
```
report = classification_report(y_test, y_pred)
report
```

Out[14]:
```
'              precision    recall  f1-score   support\n\n           0       0.75      0.77
0.76    159494\n           1       0.76      0.74      0.75    160506\n\n    accuracy
0.76    320000\n   macro avg       0.76      0.76      0.76    320000\nweighted avg       0.7
6      0.76      0.76    320000\n'
```

In [15]:
```
nb_classifier = BernoulliNB()
nb_classifier
```

Out[15]:
```
BernoulliNB()
```

In [16]:
```
nb_classifier.fit(X_train_vec, y_train)
```

Out[16]:
```
BernoulliNB()
```

In [17]:
```
y_pred = nb_classifier.predict(X_test_vec)
y_pred
```

Out[17]:
```
array([1, 1, 1, ..., 1, 0, 0], dtype=int64)
```

In [18]:
```
accuracy = accuracy_score(y_test, y_pred)
accuracy
```

Out[18]:
```
0.758084375
```

In [19]:
```
report = classification_report(y_test, y_pred)
report
```

Out[19]:
```
'              precision    recall  f1-score   support\n\n           0       0.76      0.75
0.75    159494\n           1       0.75      0.77      0.76    160506\n\n    accuracy
0.76    320000\n   macro avg       0.76      0.76      0.76    320000\nweighted avg       0.7
6      0.76      0.76    320000\n'
```

In [20]:
```
#MultinomialNB has a marginally better accuracy (0.7708375 vs. 0.77039375) and slightly better
```

In [21]:
```
vectorizer = TfidfVectorizer(max_features=5000,stop_words='english')
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
```

In [22]:
```
from sklearn.pipeline import make_pipeline
```

In [24]:
```
pipeline_multinomial = make_pipeline(TfidfVectorizer(), MultinomialNB())
pipeline_multinomial
```

Out[24]:
```
Pipeline(steps=[('tfidfvectorizer', TfidfVectorizer()),
                ('multinomialnb', MultinomialNB())])
```

In [25]:
```
pipeline_multinomial.fit(X_train, y_train)
```

Out[25]:
```
Pipeline(steps=[('tfidfvectorizer', TfidfVectorizer()),
                ('multinomialnb', MultinomialNB())])
```

In [27]:
```
predictions_multinomial = pipeline_multinomial.predict(X_test)
predictions_multinomial
```

Out[27]:
```
array([1, 0, 1, ..., 1, 0, 0], dtype=int64)
```

In [29]:
```
accuracy_multinomial = accuracy_score(y_test, predictions_multinomial)
accuracy_multinomial
```

Out[29]:
```
0.773371875
```

In [31]:
```
vectorizer_binary = CountVectorizer(max_features=5000, stop_words='english', binary=True)
X_train_vec_binary = vectorizer_binary.fit_transform(X_train)
X_test_vec_binary = vectorizer_binary.transform(X_test)
```

In [32]:
```
nb_classifier_bernoulli = BernoulliNB()
nb_classifier_bernoulli.fit(X_train_vec_binary, y_train)
```

Out[32]:
```
BernoulliNB()
```

In [33]:
```
y_pred_bernoulli = nb_classifier_bernoulli.predict(X_test_vec_binary)
y_pred_bernoulli
```

Out[33]:
```
array([1, 1, 1, ..., 1, 0, 0], dtype=int64)
```

In [34]:
```
accuracy_bernoulli = accuracy_score(y_test, y_pred_bernoulli)
accuracy_bernoulli
```

Out[34]:
```
0.758225
```

In [35]:
```
report_bernoulli = classification_report(y_test, y_pred_bernoulli)
report_bernoulli
```

Out[35]:
```
'              precision    recall  f1-score   support\n\n           0       0.76      0.75
0.76    159494\n           1       0.75      0.77      0.76    160506\n\n    accuracy
0.76    320000\n   macro avg       0.76      0.76      0.76    320000\nweighted avg       0.7
6       0.76      0.76    320000\n'
```

In [36]:
```
nb_classifier_multinomial = MultinomialNB()
nb_classifier_multinomial.fit(X_train_vec_binary, y_train)
```

Out[36]:
```
MultinomialNB()
```

In [37]:
```
y_pred_multinomial = nb_classifier_multinomial.predict(X_test_vec_binary)
y_pred_multinomial
```

Out[37]:
```
array([1, 1, 1, ..., 1, 0, 0], dtype=int64)
```

In [38]:
```
accuracy_multinomial = accuracy_score(y_test, y_pred_multinomial)
accuracy_multinomial
```

Out[38]:
```
0.755528125
```

In [39]:
```
report_multinomial = classification_report(y_test, y_pred_multinomial)
report_multinomial
```

Out[39]:
```
'              precision    recall  f1-score   support\n\n           0       0.75      0.77
0.76    159494\n           1       0.76      0.74      0.75    160506\n\n    accuracy
0.76    320000\n   macro avg       0.76      0.76      0.76    320000\nweighted avg       0.7
6      0.76      0.76    320000\n'
```

In [ ]:
```
#BernoulliNB performs slightly better than MultinomialNB in terms of accuracy for Binary Count
```