N Bharatvaz Reddy - 20BCE2844

In [ ]:

```
import spacy
m = spacy.load('en_core_web_sm')
```

In [ ]:

```

```

In [1]:

```
import pandas as pd
df = pd.read_csv('IMDB Dataset.csv')
```

In [2]:

```
df.head(10)
```

Out[2]:

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive
5	Probably my all-time favorite movie, a story o...	positive
6	I sure would like to see a resurrection of a u...	positive
7	This show was an amazing, fresh & innovative i...	negative
8	Encouraged by the positive comments about this...	negative
9	If you like original gut wrenching laughter yo...	positive

In [3]:

```
df['review'][0]
```

Out[3]:

```
"One of the other reviewers has mentioned that after watching just 1 Oz ep
isode you'll be hooked. They are right, as this is exactly what happened w
ith me.<br /><br />The first thing that struck me about Oz was its brutali
ty and unflinching scenes of violence, which set in right from the word G
O. Trust me, this is not a show for the faint hearted or timid. This show
pulls no punches with regards to drugs, sex or violence. Its is hardcore,
in the classic use of the word.<br /><br />It is called OZ as that is the
nickname given to the Oswald Maximum Security State Penitentary. It focuse
s mainly on Emerald City, an experimental section of the prison where all
the cells have glass fronts and face inwards, so privacy is not high on th
e agenda. Em City is home to many..Aryans, Muslims, gangstas, Latinos, Chr
istians, Italians, Irish and more....so scuffles, death stares, dodgy deal
ings and shady agreements are never far away.<br /><br />I would say the m
ain appeal of the show is due to the fact that it goes where other shows w
ouldn't dare. Forget pretty pictures painted for mainstream audiences, for
get charm, forget romance...OZ doesn't mess around. The first episode I ev
er saw struck me as so nasty it was surreal, I couldn't say I was ready fo
r it, but as I watched more, I developed a taste for Oz, and got accustome
d to the high levels of graphic violence. Not just violence, but injustice
(crooked guards who'll be sold out for a nickel, inmates who'll kill on or
der and get away with it, well mannered, middle class inmates being turned
into prison bitches due to their lack of street skills or prison experienc
e) Watching Oz, you may become comfortable with what is uncomfortable view
ing....thats if you can get in touch with your darker side."
```

Stemming

In [4]:

```
from nltk.stem.porter import PorterStemmer
porter = PorterStemmer()
```

In [5]:

```
def stemmer_tokenize (text):
    return [porter.stem(word) for word in text.split()]
```

In [6]:

```
stemmer_tokenize('coders like coding and thus they code')
```

Out[6]:

```
['coder', 'like', 'code', 'and', 'thu', 'they', 'code']
```

In [7]:

```
import nltk
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\dell\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[7]:

```
True
```

TF - IDF Vectorizer

In [13]:

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer(strip_accents = None,
                        lowercase = False,
                        tokenizer = stemmer_tokenize,
                        use_idf = True,
                        norm = 'l2',
                        smooth_idf = True)
```

In [14]:

```
Y = df.sentiment.values
X = tfidf.fit_transform(df.review)
```

Document classification

In [15]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, random_state = 1, test_size =
```

In [16]:

```
import pickle
from sklearn.linear_model import LogisticRegressionCV
```

In [17]:

```
clf = LogisticRegressionCV(cv = 5,
                     scoring = 'accuracy',
                     random_state = 0,
                     n_jobs = 1,
                     verbose = 2,
                     max_iter = 300).fit(X_train, Y_train)
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent wo
rkers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:  2.0min remaining:
0.0s
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed: 11.0min finished
```

In [18]:

```
saved_model = open('saved_model.sav','wb')
```

In [19]:

```
pickle.dump(clf, saved_model)
```

In [20]:

```
saved_model.close()
```

Model evaluation

In [21]:

```
filename = 'saved_model.sav'
saved_clf = pickle.load(open(filename, 'rb'))
```

In [22]:

```
saved_clf.score(X_test, Y_test)
```

Out[22]:

```
0.89004
```