24/03/2023

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 episo
de you'll be hooked. They are right, as this is exactly what happened with m
e.<br /><br />The first thing that struck me about Oz was its brutality and u
nflinching scenes of violence, which set in right from the word GO. 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 o
f the word.<br /><br />It is called OZ as that is the nickname given to the O
swald Maximum Security State Penitentary. It focuses mainly on Emerald City,
an experimental section of the prison where all the cells have glass fronts a
nd face inwards, so privacy is not high on the agenda. Em City is home to man
y..Aryans, Muslims, gangstas, Latinos, Christians, Italians, Irish and mor
e....so scuffles, death stares, dodgy dealings and shady agreements are never
far away.<br /><br />I would say the main appeal of the show is due to the fa
ct that it goes where other shows wouldn't dare. Forget pretty pictures paint
ed for mainstream audiences, forget charm, forget romance...OZ doesn't mess a
round. The first episode I ever saw struck me as so nasty it was surreal, I c
ouldn't say I was ready for it, but as I watched more, I developed a taste fo
r Oz, and got accustomed to the high levels of graphic violence. Not just vio
lence, but injustice (crooked guards who'll be sold out for a nickel, inmates
who'll kill on order and get away with it, well mannered, middle class inmate
s being turned into prison bitches due to their lack of street skills or pris
on experience) Watching Oz, you may become comfortable with what is uncomfort
able viewing....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 [3]:

```
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 = 0.5
```

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 worke
rs.
[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
```