In [1]:
```
import numpy as np
import re
import pickle
import nltk
from nltk.corpus import stopwords
from sklearn.datasets import load_files
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Kaira\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

Out[1]:
```
True
```

In [2]:
```
#import dataset
reviews = load_files('txt_sentoken/')
```

In [3]:
```
#membagi data/teks dan target/kelasnya
X,y = reviews.data, reviews.target
```

In [4]:
```
#create/preprocessing corpus
corpus = []
for i in range(0, len(X)):
    #ganti karakter non word (tanda baca dll) dg spasi
    review = re.sub(r'\W',' ', str(X[i]))
    #ganti karakter tunggal dg spasi
    review = re.sub(r'\s+[a-z]\s+',' ',review)
    #ganti huruf tunggal di awal kalimat dg spasi
    review = re.sub(r'^[a-z]\s+',' ',review)
     #ganti spasi extra dg spasi tunggal
    review = re.sub(r'\s+',' ',review)
     #masukkan di list corpus
    corpus.append(review)
```

In [5]:
```
len (corpus)
```

Out[5]:
```
2000
```

In [6]:
```
# contoh isi dokumen
corpus [0]
```

Out[6]:
```
' arnold schwarzenegger has been an icon for action enthusiasts since the late 80 but lately his films have been very sloppy an
d the one liners are getting worse nit hard seeing arnold as mr freeze in batman and robin especially when he says tons of ice
jokes but hey he got 15 million what it matter to him nonce again arnold has signed to do another expensive blockbuster that ca
n compare with the likes of the terminator series true lies and even eraser nin this so called dark thriller the devil gabriel
byrne has come upon earth to impregnate woman robin tunney which happens every 1000 years and basically destroy the world but a
pparently god has chosen one man and that one man is jericho cane arnold himself nwith the help of trusty sidekick kevin pollac
k they will stop at nothing to let the devil take over the world nparts of this are actually so absurd that they would fit righ
t in with dogma nyes the film is that weak but it better than the other blockbuster right now sleepy hollow but it makes the wo
rld is not enough look like 4 star film nanyway this definitely doesn seem like an arnold movie nit just wasn the type of film
you can see him doing nsure he gave us few chuckles with his well known one liners but he seemed confused as to where his chara
cter and the film was going nit understandable especially when the ending had to be changed according to some sources naside fo
rm that he still walked through it much like he has in the past few films ni sorry to say this arnold but maybe these are the e
nd of your action days nspeaking of action where was it in this film nthere was hardly any explosions or fights nthe devil made
few places explode but arnold wasn kicking some devil butt nthe ending was changed to make it more spiritual which undoubtedly
ruined the film ni was at least hoping for cool ending if nothing else occurred but once again was let down ni also don know wh
y the film took so long and cost so much nthere was really no super affects at all unless you consider an invisible devil who w
as in it for 5 minutes tops worth the overpriced budget nthe budget should have gone into better script where at least audience
s could be somewhat entertained instead of facing boredom nit pitiful to see how scripts like these get bought and made into mo
vie ndo they even read these things anymore nit sure doesn seem like it nthankfully gabriel performance gave some light to this
poor film nwhen he walks down the street searching for robin tunney you can help but feel that he looked like devil nthe guy is
creepy looking anyway nwhen it all over you re just glad it the end of the movie ndon bother to see this if you re expecting so
lid action flick because it neither solid nor does it have action nit just another movie that we are suckered in to seeing due
to strategic marketing campaign nsave your money and see the world is not enough for an entertaining experience '
```

In [7]:
```
#Membangun model BOW (bow hanya melihat frekuensi, tanpa melihat posisi)
from sklearn.feature_extraction.text import CountVectorizer
#membuat vektor BOW, max_features=jumlah n kata terpenting
#min_df = jumlah kata yg kurang dr ini diabaikan
#max_df = 0.6 -> jika kata muncul di lebih dari 60% dok,
#maka kata diabaikan
# stopwords membuang kata yang tidak penting
vectorizer = CountVectorizer(max_features=2000, min_df=3, max_df=0.6, stop_words=stopwords.words('english'))
```

In [8]:
```
#membuat matrix BOW (baris=dokumen, col=kata terpenting)
# setiap dokumen akan memiliki vektor
X = vectorizer.fit_transform(corpus).toarray()
```

In [9]:
```
len (X)
```

Out[9]:
```
2000
```

In [10]:
```
#mentransfer dari model BOW menjadi model TfIdf
# model tf-idf juga termasuk model bahasa, yg lebih baik dari bow
Xt = X
from sklearn.feature_extraction.text import TfidfTransformer
transformer = TfidfTransformer()
Xt = transformer.fit_transform(X).toarray()
```

In [11]:
```
#membuat test set dan training set
from sklearn.model_selection import train_test_split
text_train,text_test, sent_train, sent_test = train_test_split(Xt,y,test_size=0.2,random_state=0)
#ket:
#text_train = text untuk Learning
#text_test = text untuk testing
#sent_train = kelas dokumen untuk training
```

```
#sent_test = kelas dokumen untuk testing
```

In [12]:
```
#membangun classifier dg algoritma logistic regression
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()
classifier.fit(text_train,sent_train)
```

Out[12]:
```
LogisticRegression()
```

In [13]:
```
# menguji akurasi classifier
# hasil prediksi kelas disimpan di sent_pred
sent_pred = classifier.predict(text_test)
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
print(confusion_matrix(sent_test,sent_pred))
print(classification_report(sent_test,sent_pred))
print(accuracy_score(sent_test,sent_pred))
```

```
[[168  40]
 [ 21 171]]
              precision    recall  f1-score   support

           0       0.89      0.81      0.85       208
           1       0.81      0.89      0.85       192

    accuracy                           0.85       400
   macro avg       0.85      0.85      0.85       400
weighted avg       0.85      0.85      0.85       400

0.8475
```

In [14]:
```
cm = confusion_matrix(sent_test, sent_pred)
print(cm)
```

```
[[168  40]
 [ 21 171]]
```

In [ ]:
```

```