In [140…

```
import numpy as np
import pandas as pd
```

In [141…

```
df_train = pd.read_csv("/Users/dev/Personal/DS & AI Class Notes/Data Sets/N
```

In [142…

```
df_test = pd.read_csv("/Users/dev/Personal/DS & AI Class Notes/Data Sets/Na
```

In [143…

```
# df_train.drop("label",axis=1,inplace=True)

# df_test.drop("label",axis=1,inplace=True)
```

In [144…

```
df_train
```

Out[144…

	text	label
0	I grew up (b. 1965) watching and loving the Th...	0
1	When I put this movie in my DVD player, and sa...	0
2	Why do people who do not know what a particula...	0
3	Even though I have great interest in Biblical ...	0
4	Im a die hard Dads Army fan and nothing will e...	1
...	...	...
39995	"Western Union" is something of a forgotten cl...	1
39996	This movie is an incredible piece of work. It ...	1
39997	My wife and I watched this movie because we pl...	0
39998	When I first watched Flatliners, I was amazed....	1
39999	Why would this film be so good, but only gross...	1

40000 rows × 2 columns

In [145…

```
df_test
```

Out[145…

	text	label
0	I always wrote this series off as being a comp...	0
1	1st watched 12/7/2002 - 3 out of 10(Dir-Steve ...	0
2	This movie was so poorly written and directed ...	0
3	The most interesting thing about Miryang (Secr...	1
4	when i first read about "berlin am meer" i did...	0
...	...	...
4995	This is the kind of picture John Lassiter woul...	1
4996	A MUST SEE! I saw WHIPPED at a press screening...	1
4997	NBC should be ashamed. I wouldn't allow my chi...	0
4998	This movie is a clumsy mishmash of various gho...	0
4999	Formula movie about the illegitimate son of a ...	0

5000 rows × 2 columns

In [146…

```
review = pd.merge(left=df_train,right=df_test,how="outer")
```

In [147…

```
review
```

Out[147…

	text	label
0	I grew up (b. 1965) watching and loving the Th...	0
1	When I put this movie in my DVD player, and sa...	0
2	Why do people who do not know what a particula...	0
3	Even though I have great interest in Biblical ...	0
4	Im a die hard Dads Army fan and nothing will e...	1
...	...	...
44937	This is the kind of picture John Lassiter woul...	1
44938	A MUST SEE! I saw WHIPPED at a press screening...	1
44939	NBC should be ashamed. I wouldn't allow my chi...	0
44940	This movie is a clumsy mishmash of various gho...	0
44941	Formula movie about the illegitimate son of a ...	0

44942 rows × 2 columns

In [148…

```
X = review["text"]
```

In [149…

```
y = review["label"]
```

In [150…

```
from sklearn.model_selection import train_test_split
```

In [151…

```
Xtrain,Xtest,ytrain,ytest  = train_test_split(X,y,test_size=.20)
```

In [152…

```
X.shape , Xtrain.shape , Xtest.shape
```

Out[152…

```
((44942,), (35953,), (8989,))
```

In [153…

```
y.shape , ytrain.shape , ytest.shape
```

Out[153…

```
((44942,), (35953,), (8989,))
```

In [154…

```
from sklearn.feature_extraction.text import CountVectorizer , TfidfTransfor
```

In [155…

```
cv = CountVectorizer()
```

In [156…

```
cv_Xtrain = cv.fit_transform(Xtrain)
```

In [157…

```
cv_Xtest = cv.transform(Xtest)
```

In [158…

```
cv_Xtrain
```

Out[158…

```
<35953x89233 sparse matrix of type '<class 'numpy.int64'>'
        with 4926279 stored elements in Compressed Sparse Row format>
```

In [159…

```
cv_Xtest
```

Out[159…

```
<8989x89233 sparse matrix of type '<class 'numpy.int64'>'
        with 1206722 stored elements in Compressed Sparse Row format>
```

In [160…

```
tfid = TfidfTransformer()
```

In [161…

```
tfid_Xtrain = tfid.fit_transform(cv_Xtrain)
```

In [162…

```
tfid_Xtest = tfid.transform(cv_Xtest)
```

In [163…

```
tfid_Xtrain
```

Out[163…

```
<35953x89233 sparse matrix of type '<class 'numpy.float64'>'
        with 4926279 stored elements in Compressed Sparse Row format>
```

In [164…

```
tfid_Xtest
```

Out[164…

```
<8989x89233 sparse matrix of type '<class 'numpy.float64'>'
        with 1206722 stored elements in Compressed Sparse Row format>
```

In [165…

```
from sklearn.naive_bayes import MultinomialNB
```

In [166…

```
mnb = MultinomialNB()
```

In [167…

```
mnb.fit(tfid_Xtrain,ytrain)
```

Out[167…

```
MultinomialNB()
```

In [168…

```
mnb.score(tfid_Xtest,ytest)
```

Out[168…

```
0.861052397374569
```

Testing With Func

In [116…

```
def sen_mulnb(X,y):
    from sklearn.model_selection import train_test_split
    Xtrain,Xtest,ytrain,ytest = train_test_split(X,y,test_size=.20)
    from sklearn.feature_extraction.text import CountVectorizer , TfidfTran
    cv = CountVectorizer()
    cv_Xtrain = cv.fit_transform(Xtrain)
    cv_Xtest = cv.transform(Xtest)
    tfid = TfidfTransformer()
    tfid_Xtrain = tfid.fit_transform(cv_Xtrain)
    tfid_Xtest = tfid.transform(cv_Xtest)
    from sklearn.naive_bayes import MultinomialNB
    mnb = MultinomialNB()
    mnb.fit(tfid_Xtrain,ytrain)
    print(f'Score is {mnb.score(tfid_Xtest,ytest)}')
```

In [117…

```
sen_mulnb(X,y)
```

```
Score is 0.8685059517187674
```

Making Pipeline

In [103…
```
from sklearn.feature_extraction.text import CountVectorizer , TfidfTransfor
from sklearn.naive_bayes import MultinomialNB
```

In [104…
```
from sklearn.pipeline import make_pipeline
```

In [105…
```
pipe = make_pipeline(CountVectorizer(),TfidfTransformer(),MultinomialNB())
```

In [106…
```
pipe
```

Out[106…
```
Pipeline(steps=[('countvectorizer', CountVectorizer()),
                ('tfidftransformer', TfidfTransformer()),
                ('multinomialnb', MultinomialNB())])
```

In [107…
```
pipe.fit(X,y)
```

Out[107…
```
Pipeline(steps=[('countvectorizer', CountVectorizer()),
                ('tfidftransformer', TfidfTransformer()),
                ('multinomialnb', MultinomialNB())])
```

In [108…
```
pipe.score(X,y)
```

Out[108…
```
0.9010279916336612
```

With Train_test

In [109…
```
pipe1 = make_pipeline(CountVectorizer(),TfidfTransformer(),MultinomialNB())
```

In [110…
```
pipe1.fit(Xtrain,ytrain)
```

Out[110…
```
Pipeline(steps=[('countvectorizer', CountVectorizer()),
                ('tfidftransformer', TfidfTransformer()),
                ('multinomialnb', MultinomialNB())])
```

In [111…
```
pipe1.score(Xtest,ytest)
```

Out[111…
```
0.8637223272889086
```

With TfidfVectorizer

In [112…
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [113…

```
pipe3 = make_pipeline(TfidfVectorizer(),MultinomialNB())
```

In [114…

```
pipe3.fit(Xtrain,ytrain)
```

Out[114…

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

In [115…

```
pipe3.score(Xtest,ytest)
```

Out[115…

```
0.8637223272889086
```