Web Mining (CSE3024)

Lab Assignment 7

Name: **Kritika Mishra**

Registration Number: **16BCI0041**

Slot: L15+L16

Date: 3rd October 2018

Question:
Building a Text Classifier Using Naive Bayes to classify the Movie data into Positive and Negative Sentiment.

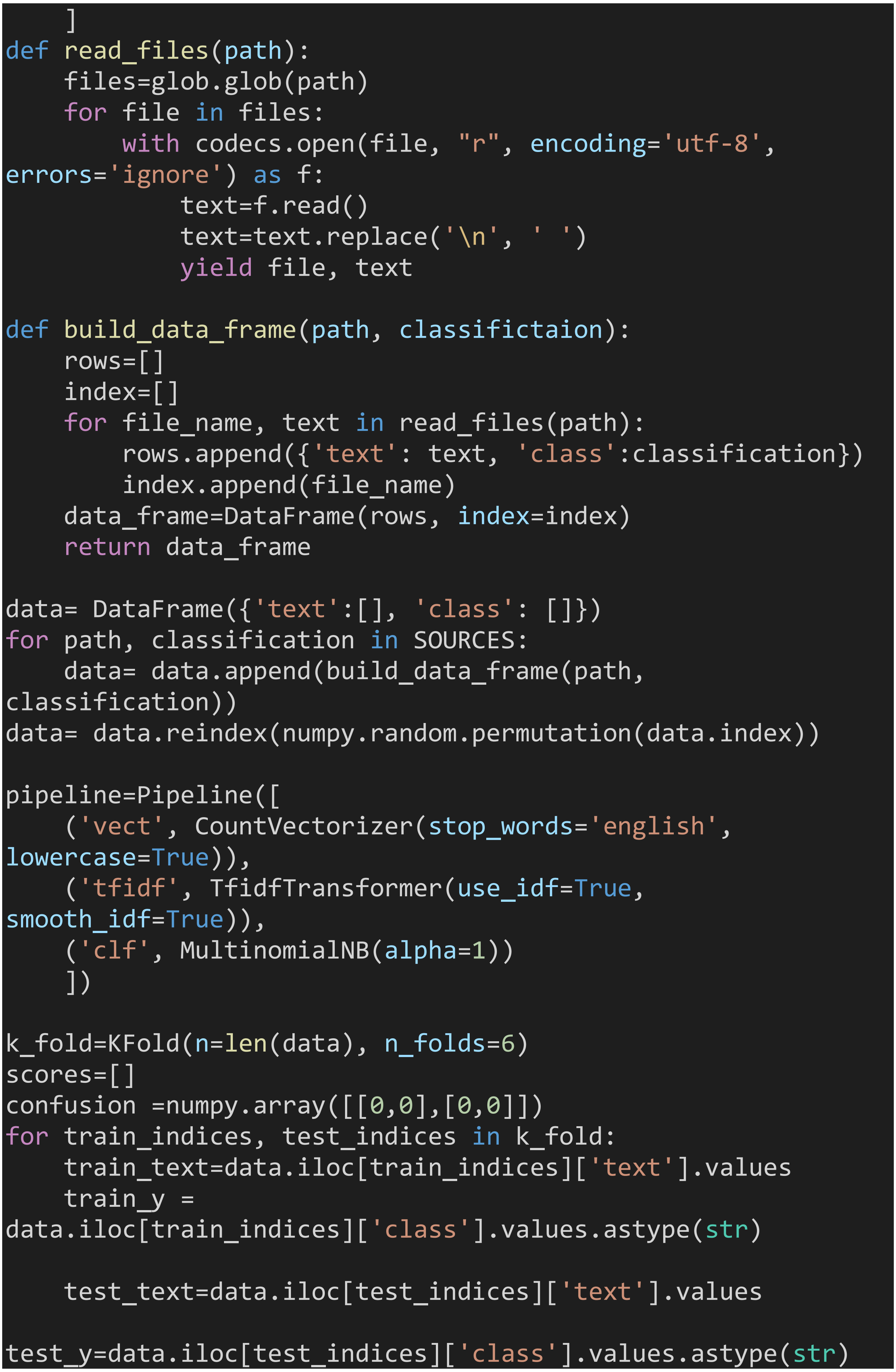
- **Use any of the Toolkit / Package to perform the process**
- **Print out the Accuracy and Confusion Matrix of Classification**
- **Document the step by step process and upload with output and Code**

Note: Dataset can be generated or downloaded from the internet. Please specify the source of the dataset in the documentation steps of this program.

Dataset:
http://www.cs.cornell.edu/people/pabo/movie-review-data/

Code:

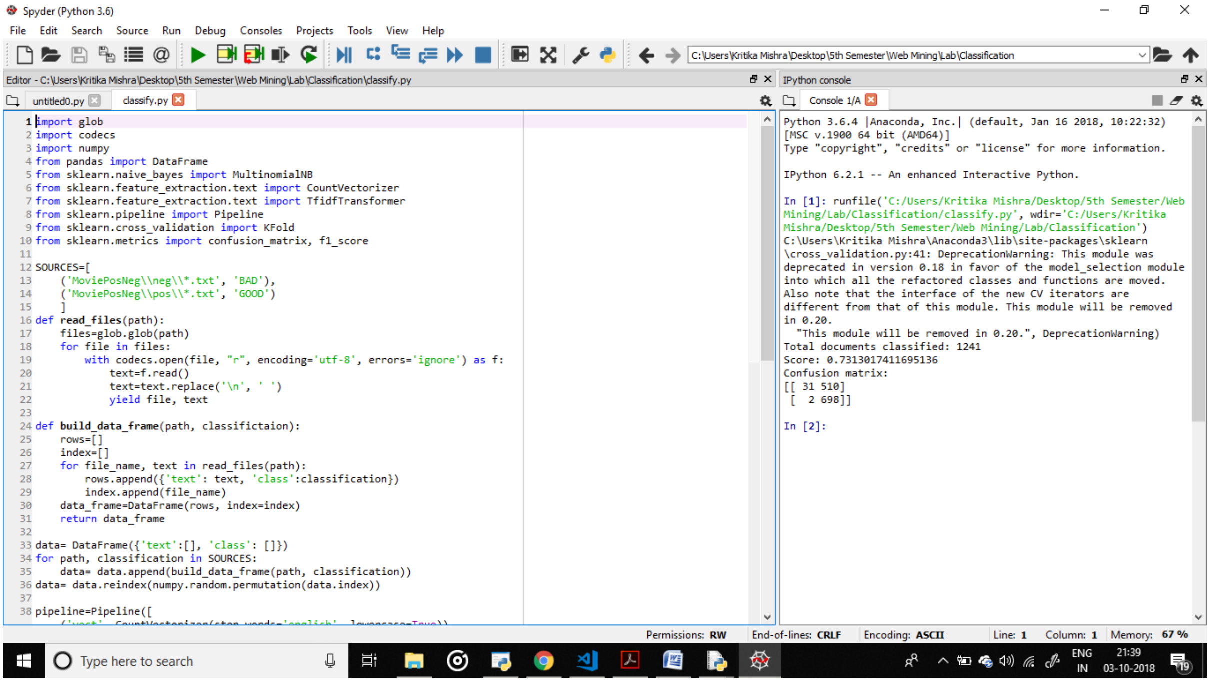
```
import glob
import codecs
import numpy
from pandas import DataFrame
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.pipeline import Pipeline
from sklearn.cross_validation import KFold
from sklearn.metrics import confusion_matrix, f1_score

SOURCES=[
    ('MoviePosNeg\\neg\\*.txt', 'BAD'),
    ('MoviePosNeg\\pos\\*.txt', 'GOOD')
```



```
    ]
def read_files(path):
    files=glob.glob(path)
    for file in files:
        with codecs.open(file, "r", encoding='utf-8',
errors='ignore') as f:
            text=f.read()
            text=text.replace('\n', ' ')
            yield file, text

def build_data_frame(path, classifictaion):
    rows=[]
    index=[]
    for file_name, text in read_files(path):
        rows.append({'text': text, 'class':classification})
        index.append(file_name)
    data_frame=DataFrame(rows, index=index)
    return data_frame

data= DataFrame({'text':[], 'class': []})
for path, classification in SOURCES:
    data= data.append(build_data_frame(path,
classification))
data= data.reindex(numpy.random.permutation(data.index))

pipeline=Pipeline([
    ('vect', CountVectorizer(stop_words='english',
lowercase=True)),
    ('tfidf', TfidfTransformer(use_idf=True,
smooth_idf=True)),
    ('clf', MultinomialNB(alpha=1))
    ])

k_fold=KFold(n=len(data), n_folds=6)
scores=[]
confusion =numpy.array([[0,0],[0,0]])
for train_indices, test_indices in k_fold:
    train_text=data.iloc[train_indices]['text'].values
    train_y =
data.iloc[train_indices]['class'].values.astype(str)

    test_text=data.iloc[test_indices]['text'].values

test_y=data.iloc[test_indices]['class'].values.astype(str)
```

```python
    pipeline.fit(train_text, train_y)
    predictions=pipeline.predict(test_text)

    confusion+= confusion_matrix(test_y, predictions)
    score=f1_score(test_y, predictions, pos_label='GOOD')
    scores.append(score)

print('Total documents classified:', len(data))
print('Score:' ,sum(scores)/len(scores))
print('Confusion matrix:')
print(confusion)
```

Output:


Spyder (Python 3.6)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Kritika Mishra\Desktop\5th Semester\Web Mining\Lab\Classification
Editor - C:\Users\Kritika Mishra\Desktop\5th Semester\Web Mining\Lab\Classification\classify.py
untitled0.py
classify.py
IPython console
Console 1/A
Permissions: RW
End-of-lines: CRLF
Encoding: ASCII
Line: 1
Column: 1
Memory: 67 %
Type here to search
ENG
IN
21:39
03-10-2018

```python
import glob
import codecs
import numpy
from pandas import DataFrame
from sklearn.naive_bayes import MultinomialNB
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.pipeline import Pipeline
from sklearn.cross_validation import KFold
from sklearn.metrics import confusion_matrix, f1_score

SOURCES=[
    ('MoviePosNeg\\neg\\*.txt', 'BAD'),
    ('MoviePosNeg\\pos\\*.txt', 'GOOD')
    ]
def read_files(path):
    files=glob.glob(path)
    for file in files:
        with codecs.open(file, "r", encoding='utf-8', errors='ignore') as f:
            text=f.read()
            text=text.replace('\n', ' ')
            yield file, text

def build_data_frame(path, classifictaion):
    rows=[]
    index=[]
    for file_name, text in read_files(path):
        rows.append({'text': text, 'class':classification})
        index.append(file_name)
    data_frame=DataFrame(rows, index=index)
    return data_frame

data= DataFrame({'text':[], 'class': []})
for path, classification in SOURCES:
    data= data.append(build_data_frame(path, classification))
data= data.reindex(numpy.random.permutation(data.index))

pipeline=Pipeline([
```

```
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 10:22:32)
[MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.2.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Kritika Mishra/Desktop/5th Semester/Web
Mining/Lab/Classification/classify.py', wdir='C:/Users/Kritika
Mishra/Desktop/5th Semester/Web Mining/Lab/Classification')
C:\Users\Kritika Mishra\Anaconda3\lib\site-packages\sklearn
\cross_validation.py:41: DeprecationWarning: This module was
deprecated in version 0.18 in favor of the model_selection module
into which all the refactored classes and functions are moved.
Also note that the interface of the new CV iterators are
different from that of this module. This module will be removed
in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
Total documents classified: 1241
Score: 0.7313017411695136
Confusion matrix:
[[ 31 510]
 [  2 698]]

In [2]:
```

Output:

```
Total documents classified: 1241

Score: 0.7313017411695136

Confusion matrix:

[[ 31 510]

[  2 698]]
```