# Homework Assignment Hw 7

보고서 및 논문 윤리 서약

1. 나는 보고서 및 논문의 내용을 조작하지 않겠습니다.
2. 나는 다른 사람의 보고서 및 논문의 내용을 내 것처럼 무단으로 복사하지 않겠습니다.
3. 나는 다른 사람의 보고서 및 논문의 내용을 참고하거나 인용할 시 참고 및 인용 형식을 갖추고 출처를 반드시 밝히겠습니다.
4. 나는 보고서 및 논문을 대신하여 작성하도록 청탁하지도 청탁받지도 않겠습니다.

나는 보고서 및 논문 작성 시 위법 행위를 하지 않고, 명지인으로서 또한 공학인으로서 나의 양심과 명예를 지킬 것을 약속합니다.

학 과 : 융합소프트웨어학부 데이터테크놀로지전공

과 목 : 인공지능

담당교수 : 전종훈

강좌 번호: 6019

학 번 : 60182196

이 름 : 이동혁 (서명)

1.

(a).

```
from sklearn.datasets import load_files
import nltk
from nltk import word_tokenize
from nltk.stem.porter import PorterStemmer
import numpy as np

reviews_train = load_files("/Users/leedonghyeok/Downloads/aclImdb/train")
reviews_test = load_files("/Users/leedonghyeok/Downloads/aclImdb/test")

text_train, y_train = reviews_train.data, reviews_train.target
text_test, y_test = reviews_test.data, reviews_test.target

print("done")
```

```
done
```

(b).

```
text_train = [doc.replace(b"<br />", b" ") for doc in text_train]
text_test = [doc.replace(b"<br />", b" ") for doc in text_test]

print("done")

# Having an intelligent interesting script doesn't hurt either.<br /><br />Stargate SG1 is currently one of my favorite programs.
# Having an intelligent interesting script doesn't hurt either.  Stargate SG1 is currently one of my favorite programs.
```

```
done
```

(c).

```
from sklearn.feature_extraction.text import CountVectorizer

stemmer = PorterStemmer()

vect = CountVectorizer(stop_words = 'english').fit(text_train)
X_train = vect.transform(text_train)
X_test = vect.transform(text_test)

print("done")
```

```
done
```

(d).

```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

model = GaussianNB()
model.fit(X_train.toarray(),y_train)
pre = model.predict(X_test.toarray())

ac_score = accuracy_score(y_test,pre)
print("정답률 = {:0,.1f}".format(ac_score))
```

```
정답률 = 0.6
```

2.

(a).

```
import pandas as pd
import numpy as np

df_train = pd.read_csv("/Users/leedonghyeok/Downloads/ratings_train.txt", delimiter = '₩t', keep_default_na = False)
df_test = pd.read_csv("/Users/leedonghyeok/Downloads/ratings_test.txt", delimiter = '₩t', keep_default_na = False)

text_train = df_train['document']
y_train = df_train['label']

text_test = df_test['document']
y_test = df_test['label']

print("done")
```

```
done
```

# (B).

```
from sklearn.feature_extraction.text import CountVectorizer

sample = text_train[1]

print(sample)

import konlpy
from konlpy.tag import Okt

twitter_tag = Okt()

def twitter_tokenizer(text):
    malist = twitter_tag.pos(text, norm = True, stem = True)
    r = []
    for word in malist:
        if not word[1] in ['Josa','Eomi','Punctuation','KoreanParticle']:
            r.append(word[0])
    return r

vect = CountVectorizer(tokenizer = twitter_tokenizer).fit(text_train)

X_train = vect.transform(text_train)

print(twitter_tokenizer(sample))
print("done")

# 흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나
# '흠', '포스터', '보고', '초딩', '영화', '줄', '오버', '연기', '가볍다', '않다'
```

흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나

```
/Users/leedonghyeok/opt/anaconda3/lib/python3.8/site-packages/sklearn/featu
be used since 'tokenizer' is not None'
  warnings.warn("The parameter 'token_pattern' will not be used"
```

['흠', '포스터', '보고', '초딩', '영화', '줄', '오버', '연기', '가볍다', '않다']
done

(c).

```
from sklearn.feature_extraction.text import TfidfVectorizer

vect2 = TfidfVectorizer().fit(text_train)
X_train = vect2.transform(text_train)
X_test = vect2.transform(text_test)

print(X_train[:3],"₩n")
print(X_train[:3].toarray())
```

```
  (0, 248358)   0.6303222409610997
  (0, 246232)   0.26458844458802766
  (0, 99567)    0.5193460454404283
  (0, 71119)    0.5128026059076282
  (1, 273335)   0.39339783492704455
  (1, 255126)   0.48708202211988
  (1, 190112)   0.48708202211988
  (1, 167602)   0.4594654475140474
  (1, 16352)    0.39953955182265427
  (2, 57394)    1.0

[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

(d).

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

nb1 = MultinomialNB().fit(X_train, y_train)

pre = nb1.predict(X_test)

ac_score1 = metrics.accuracy_score(y_test,pre)
print("정답률 = {:0,.1f}".format(ac_score1))
print(ac_score1)
```

```
정답률 = 0.8
0.8276
```

(e).

```
nb2 = MultinomialNB(alpha = 0.6).fit(X_train, y_train)

pre2 = nb2.predict(X_test)

ac_score2 = metrics.accuracy_score(y_test,pre2)
print("정답률 = {}".format(ac_score2), ", alpha = {}".format(0.6))
```

```
정답률 = 0.8276 , alpha = 0.6
```