



+ 코드 + 텍스트

✓ RAM  
디스크

## 2-1) 네이버 영화리뷰 감정분석

```
[20] 1 import warnings
      2 # 경고메세지 끄기
      3 warnings.filterwarnings(action='ignore')
      4
      5 # 참고) 다시 출력하게 하기
      6 # warnings.filterwarnings(action='default')
```

```
[21] 1 from google.colab import drive
      2 drive.mount('/content/drive')
```

Mounted at /content/drive

```
[22] 1 import pandas as pd
      2 df = pd.read_csv('/content/drive/MyDrive/강의자료/핀인사이트_강의자료 /2023 강의자료 업데이트/자연어처리/실습/data/review.csv',
      3                sep = '\t')
      4 # df = df[:100]
      5 df.head()
```

	id	document	label
0	9976970	아 더빙.. 진짜 짜증나네요 목소리	0
1	3819312	흠...포스터보고 초딩영화줄....오버연기조차 가볍지 않구나	1
2	10265843	너무재밌었다그래서보는것을추천한다	0
3	9045019	교도소 이야기구먼 ..솔직히 재미는 없다..평점 조정	0
4	6483659	사이몬페그의 익살스런 연기가 돋보였던 영화!스파이더맨에서 늙어보이기만 했던 커스틴 ...	1

1초 [23]

```
1 from sklearn.model_selection import train_test_split
2 from sklearn.feature_extraction.text import CountVectorizer
3 from sklearn.feature_extraction.text import TfidfTransformer
4 from sklearn.naive_bayes import MultinomialNB
5
6 X_train, X_test, y_train, y_test = train_test_split(df['document'], df['label'], random_state = 0)
7 count_vect = CountVectorizer()
8 X_train_counts = count_vect.fit_transform(X_train)
9
10 tfidf_transformer = TfidfTransformer()
11 X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
12
13 model = MultinomialNB()
14 model.fit(X_train_tfidf, y_train)
```

▼ MultinomialNB  
MultinomialNB()

0초 [24]

```
1 len(X_test)
```

2500

0초 [25]

```
1 # 예측
2 y_predict = model.predict(count_vect.transform(X_test))
3 y_predict
```

array([1, 1, 0, ..., 0, 0, 1])

0초

```
1 from sklearn.metrics import classification_report
2 print(classification_report(y_test, y_predict, target_names=['부정( 0 )', '긍정( 1 )']))
3
4 # macro: 단순평균
5 # weighted: 각 클래스에 속하는 표본의 갯수로 가중평균
6 # accuracy: 정확도. 전체 학습데이터의 개수에서 각 클래스에서 자신의 클래스를 정확하게 맞춘 개수의 비율.
```

0초 [26] 5 # weighted: 각 클래스에 속하는 표본의 갯수로 가중평균  
6 # accuracy: 정확도. 전체 학습데이터의 개수에서 각 클래스에서 자신의 클래스를 정확하게 맞춘 개수의 비율.

	precision	recall	f1-score	support
부정( 0 )	0.76	0.76	0.76	1248
긍정( 1 )	0.76	0.76	0.76	1252
accuracy			0.76	2500
macro avg	0.76	0.76	0.76	2500
weighted avg	0.76	0.76	0.76	2500

✓ 0초 [27] 1 test = '강력한 몰입감을 선사하는 두 배우의 연기력!! 오랜만에 정말 색다른 영화였다'  
2  
3 print(model.predict((count\_vect.transform([test]))))  
4  
5 print(model.predict\_proba((count\_vect.transform([test]))))

[1]  
[[0.01840793 0.98159207]]

✓ 0초



1

