텍스트 데이터 분류 모델 만들기

네이버 기사 데이터를 활용하여 TfidfVectorizer와 MultinomialNB을 이용하여 분류 모델 만드는 방법을 학습합니다.

- Index
  1. 패키지 추가
  2. 데이터 로드
  3. 학습 데이터와 테스트 데이터 나누기
  4. 모델 만들기 (모델 학습)
  5. 모델 성능 확인
  6. 모델을 사용하여 문장의 카테고리 예측하기

1. 패키지 추가 ¶

In [14]:

```
import pandas as pd
import numpy as np
import pickle
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report
```

2. 데이터 로드

- Category
  - 100 - 정치
  - 101 - 경제
  - 102 - 사회
  - 103 - 생활/문화
  - 104 - 세계
  - 105 - IT/과학

In [16]:

```
article_df = pd.read_csv("articles_1200.csv")
print(len(article_df))
article_df.tail(2)
```

1200

Out[16]:

	title	link	category	content
1198	S10 5G 100만·V50 30만대 팔렸다…이통사 보조금 축소 '속도조절'	https://news.naver.com/main/read.nhn?mode=LSD&...	105	5G 세계 최초 상용화 후 80여일만 130만 돌파서울 광화문 KT매장에서 시민들이...
1199	태양광 시설로 사라진 농지면적 3년간 5618.8㏊… 여의도 20배	https://news.naver.com/main/read.nhn?mode=LSD&...	105	3년 동안 태양광 사업으로 사라진 농지 면적이 서울 여의도 면적의 20배에 육박한 것...

3. 학습 데이터와 테스트 데이터 나누기

In [17]:

```
X_train, X_test, y_train, y_test = train_test_split(article_df.content, article_df.category, tes
len(X_train), len(X_test), len(y_train), len(y_test)
```

Out[17]:

(1080, 120, 1080, 120)

4. 모델 만들기 (모델 학습)

In [18]:

```
clf = Pipeline([
    ('vect', TfidfVectorizer()),
    ('clf', MultinomialNB(alpha=0.01)),
])
```

In [19]:

```
%%time
model = clf.fit(X_train.values.astype("str"), y_train)
```

CPU times: user 717 ms, sys: 54.9 ms, total: 772 ms
Wall time: 818 ms

5. 모델 성능 확인

In [20]:

```
y_pred = model.predict(X_test)
```

In [21]:

```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

         100       0.86      1.00      0.93        19
         101       0.88      0.67      0.76        21
         102       0.83      0.80      0.82        25
         103       0.69      0.64      0.67        14
         104       1.00      0.89      0.94        19
         105       0.75      0.95      0.84        22

    accuracy                           0.83       120
   macro avg       0.84      0.83      0.83       120
weighted avg       0.84      0.83      0.83       120
```

6. 모델을 사용하여 문장의 카테고리 예측하기

In [22]:

```
classification_dict = {
    100:"정치",
    101:"경제",
    102:"사회",
    103:"생활/문화",
    104:"세계",
    105:"IT/과학",
}
```

In [23]:

```
with open("model.pkl", "wb") as f:
    pickle.dump(model, f)
```

In [24]:

```
with open("model.pkl", "rb") as f:
    load_model = pickle.load(f)
```

In [25]:

```
proba = load_model.predict_proba(["트럼프 미국 대통령 중국과 무역협상 타결"])[0]
list(zip(classification_dict.values(), np.round(proba, 3)))
```

Out[25]:

```
[('정치', 0.062),
 ('경제', 0.079),
 ('사회', 0.018),
 ('생활/문화', 0.009),
 ('세계', 0.828),
 ('IT/과학', 0.005)]
```

In [26]:

```
classification_dict.values()
```

Out[26]:

```
dict_values(['정치', '경제', '사회', '생활/문화', '세계', 'IT/과학'])
```

In [27]:

```
category = model.predict(["트럼프 미국 대통령 중국과 무역협상 타결"])[0]
classification_dict[category]
```

Out[27]:

```
'세계'
```