기계학습(8585) 기말고사 대체 과제 보고서

제품 review 데이터의 감성 분석

202120990 김도환

1. 감성 분석의 기본 이해

1) 감성 분석이란?

감성 분석은 텍스트 데이터를 기반으로 감정 또는 의견을 긍정, 부정으로 분류하는 작업입니다. 주로 소셜 미디어, 리뷰 사이트, 설문 조사 등에서 수집된 텍스트 데이터를 분석하여 사용자 의견을 파악하고, 이를 통해 제품 개선, 고객 만족도 향상 등 다양한 응용 분야에서 활용됩니다.

2) 감성 분석의 응용 분야

감성 분석은 다양한 분야에서 활용됩니다. 몇 가지 예시를 들자면, 제품 리뷰 분석을 통해 제품의 강점과 약점을 파악하고, 이를 개선하는 데 활용될 수 있고, 고객 서비스 채널에서 수집된 데이터를 분석하여 고객의 불만 사항을 빠르게 해결하고, 서비스 품질을 향상시키는 등 다양한 분야에서 활용될 수 있습니다.

2. 데이터 전처리

1) 데이터 로드하기

```
import pandas as pd

df = pd.read_csv("./wine_review.csv")
print(f"전체 리뷰 개수: {len(df)}\n")
print("===============   DataFrame info   ===============")
print(df.info())
```

```
전체 리뷰 개수: 2890
```

총 2,890개의 샘플이 존재합니다. 이제 감성 분석을 위해 사용할 칼럼을 선정합니다. `reviews.rating`, `reviews.text`, `reviews.title` 이 세 개의 칼럼을 주요 특성으로 사용하였습니다.

```
df = df[["reviews.rating", "reviews.text", "reviews.title"]]
df.head(10)
```

	reviews.rating	reviews.text	reviews.title
0	5.0	This a fantastic white wine for any occasion!	My Favorite White Wine
1	5.0	Tart, not sweet...very refreshing and delicious!	Yum!!
2	5.0	I was given this wine so it was a delightful s...	A New Favorite!
3	5.0	This is a phenomenal wine and my new favorite ...	Bold, Flavorful, Aromatic, Delicious
4	5.0	4 750ml bottles for the price of two With way ...	Yum! Plus, Environmentally Friendly!
5	5.0	I LOVE Becks NA. It tastes just like a regular...	Great Taste
6	3.0	This wine has a wonderful but strong aroma its...	Simply Wonderful
7	2.0	I would give one more star if it came clean on...	A Sweet Red.
8	5.0	Delicious and very affordable	NaN
9	5.0	This is a very smooth red with Aromas of cocoa...	Charles & Charles Red Blend

2) 데이터 중복값 및 결측치 제거

각 열에 대해서 중복을 제외한 샘플의 수를 카운트합니다.

```
df["reviews.rating"].nunique(), df["reviews.text"].nunique(), df["reviews.title"].nunique()
```

```
(5, 2550, 2158)
```

`reviews.text`를 확인해보면 중복을 제외한 경우 2,550개의 데이터가 존재한 것을 확인할 수 있습니다. 현재 2,890개의 데이터가 존재하므로 이는 현재 갖고 있는 데이터에 중복된 샘플이 있다는 의미입니다. 중복된 샘플을 제거해줍니다.

```
df.drop_duplicates(subset=["reviews.text"], inplace=True)
print(f"전체 리뷰 개수: {len(df)}\n")
```

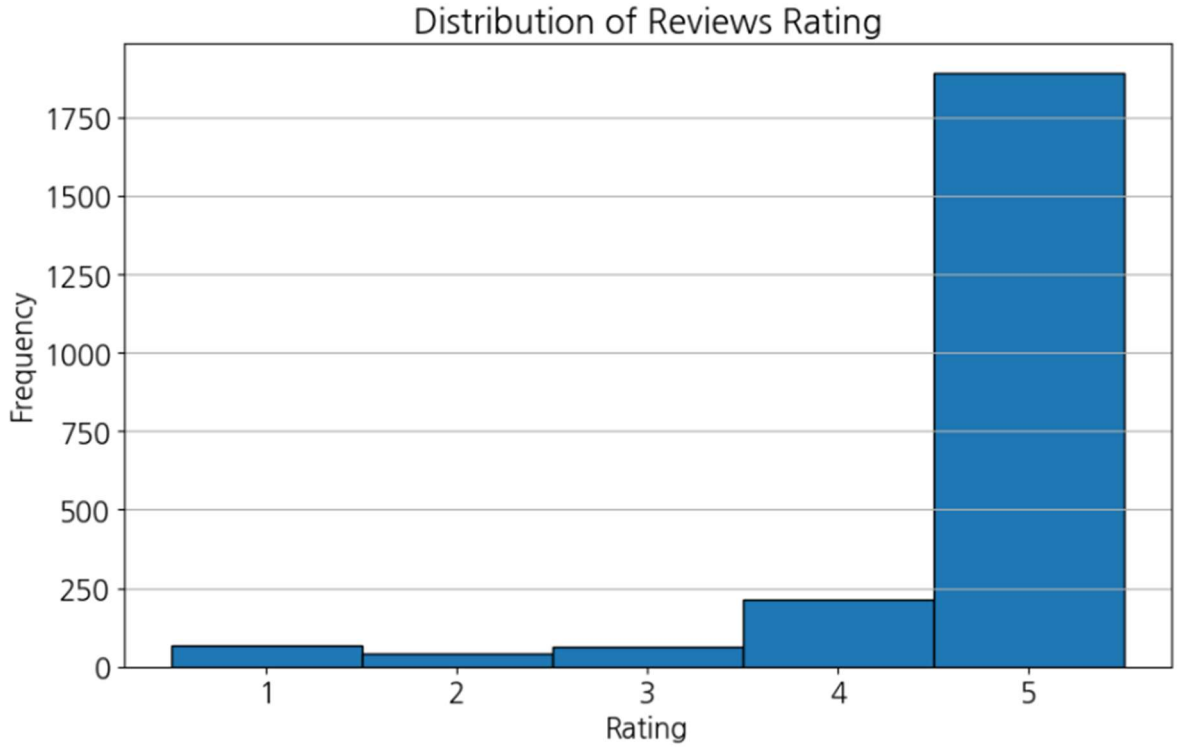
```
전체 리뷰 개수: 2551
```

`reviews.text`에 결측치가 있다면 해당 행은 제거해주고, `reviews.title`에 결측치가 있다면 해당 데이터를 공백으로 채워줍니다.

```
df = df.dropna(subset=["reviews.text"])
df["reviews.title"] = df["reviews.title"].fillna("")
df.head(10)
```

	reviews.rating	reviews.text	reviews.title
0	5.0	This a fantastic white wine for any occasion!	My Favorite White Wine
1	5.0	Tart, not sweet...very refreshing and delicious!	Yum!!
2	5.0	I was given this wine so it was a delightful s...	A New Favorite!
3	5.0	This is a phenomenal wine and my new favorite ...	Bold, Flavorful, Aromatic, Delicious
4	5.0	4 750ml bottles for the price of two With way ...	Yum! Plus, Environmentally Friendly!
5	5.0	I LOVE Becks NA. It tastes just like a regular...	Great Taste
6	3.0	This wine has a wonderful but strong aroma its...	Simply Wonderful
7	2.0	I would give one more star if it came clean on...	A Sweet Red.
8	5.0	Delicious and very affordable	
9	5.0	This is a very smooth red with Aromas of cocoa...	Charles & Charles Red Blend

전처리 후의 데이터셋 별점 분포도를 시각화하면 아래 그래프와 같습니다.



3) 데이터 전처리 과정

데이터 전처리 과정은 아래와 같습니다.

- 텍스트 정규화: 텍스트 데이터를 소문자로 변환하고, 구두점 및 특수 문자를 제거하여 일관된 형태로 만듭니다.
- 토큰화: 텍스트 데이터를 단어 단위로 분리합니다.
- 불용어 제거: 의미 없는 단어(불용어)를 제거하여 중요한 정보만 남깁니다.
- 표제어 추출: 단어의 원형을 추출하여 동일한 의미의 단어를 일관되게 처리합니다.

```python
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer, PorterStemmer
import nltk

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

# 전처리 함수 정의
def preprocess_text(text):
    # 노이즈 제거
    text = text.lower()                 # 소문자 변환
    text = re.sub(r'\d+', '', text)     # 숫자 제거
    text = re.sub(r'\s+', ' ', text)    # 추가 공백 제거
    text = re.sub(r'[^\w\s]', '', text) # 특수 문자 제거
    # 토큰화
    tokens = word_tokenize(text)
    # 불용어 제거
    tokens = [word for word in tokens if word not in stopwords.words("english")]
    # 표제어 추출
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(word) for word in tokens]
    return ' '.join(tokens)

# 텍스트 전처리
df["reviews.text"] = df["reviews.text"].apply(preprocess_text)
df["reviews.title"] = df["reviews.title"].apply(preprocess_text)

df.head(10)
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     /home/students/cs/202120990/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     /home/students/cs/202120990/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     /home/students/cs/202120990/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

	reviews.rating	reviews.text	reviews.title
0	5.0	fantastic white wine occasion	favorite white wine
1	5.0	tart sweetvery refreshing delicious	yum
2	5.0	given wine delightful surprise find flavorful ...	new favorite
3	5.0	phenomenal wine new favorite red	bold flavorful aromatic delicious
4	5.0	ml bottle price two way le packaging yes pleas...	yum plus environmentally friendly
5	5.0	love beck na taste like regular ale smell like...	great taste
6	3.0	wine wonderful strong aroma bit bitter bite st...	simply wonderful
7	2.0	would give one star came clean bottle called s...	sweet red
8	5.0	delicious affordable	
9	5.0	smooth red aroma cocoa coffee tobacco sweet bl...	charles charles red blend

3. 모델 선택 및 학습

1) 모델 선택

수업에서 사용한 VADER 감정 추론 모델을 사용하여 텍스트의 감정이 긍정적인지 부정적인지 판

단합니다. NLTK의 VADER를 통해 리뷰 제목과 리뷰 내용을 감성 분석하여 compound 값을 각각 저장한 후, 두 값을 평균 내서 리뷰의 최종 VADER compound를 계산합니다.

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer

# NLTK 패키지 다운로드
nltk.download("vader_lexicon")

# VADER 초기화
vader_sentiment = SentimentIntensityAnalyzer()

# VADER를 사용하여 감성 점수 계산
df["text_vader_scores"] = df["reviews.text"].apply(lambda review: vader_sentiment.polarity_scores(review))
df["title_vader_scores"] = df["reviews.title"].apply(lambda review: vader_sentiment.polarity_scores(review))

# compound 점수만 추출하여 별도의 컬럼에 저장
df["text_vader_compound"] = df["text_vader_scores"].apply(lambda score_dict: score_dict["compound"])
df["title_vader_compound"] = df["title_vader_scores"].apply(lambda score_dict: score_dict["compound"])
df["vader_compound"] = (df["text_vader_compound"] + df["title_vader_compound"])/2

df[["reviews.rating", "reviews.text", "reviews.title", "text_vader_compound", "title_vader_compound", "vader_compou
```

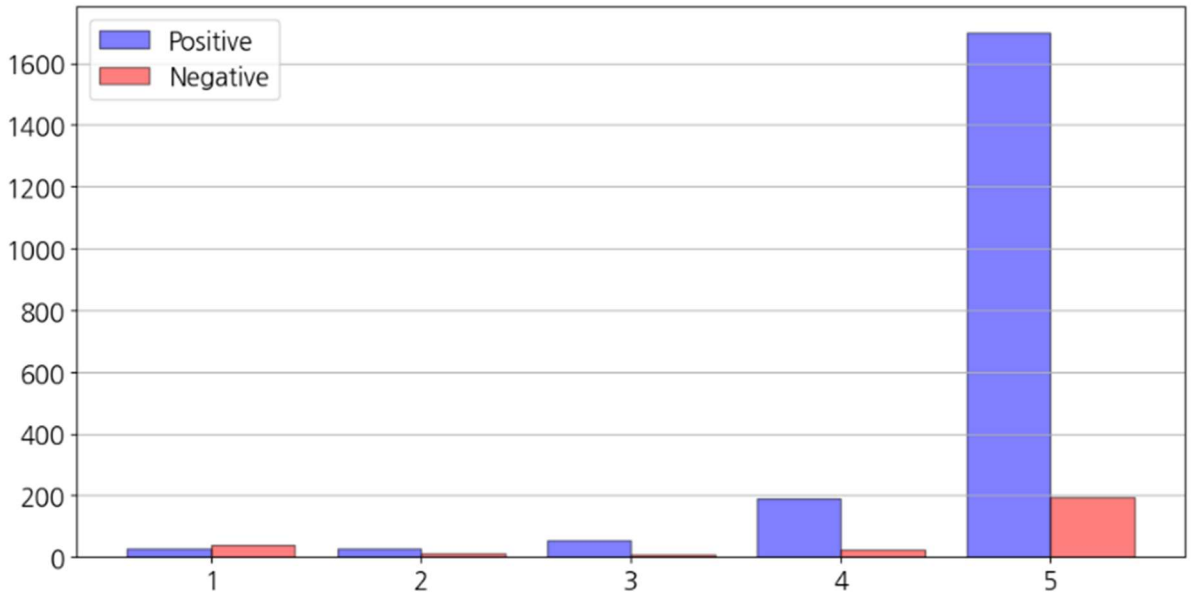
```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     /home/students/cs/202120990/nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

	reviews.rating	reviews.text	reviews.title	text_vader_compound	title_vader_compound	vader_compound
0	5.0	fantastic white wine occasion	favorite white wine	0.5574	0.4588	0.50810
1	5.0	tart sweetvery refreshing delicious	yum	0.5719	0.0000	0.28595
2	5.0	given wine delightful surprise find flavorful ...	new favorite	0.9118	0.4588	0.68530
3	5.0	phenomenal wine new favorite red	bold flavorful aromatic delicious	0.4588	0.7430	0.60090
4	5.0	ml bottle price two way le packaging yes pleas...	yum plus environmentally friendly	0.6669	0.4939	0.58040
5	5.0	love beck na taste like regular ale smell like...	great taste	0.9771	0.6249	0.80100
6	3.0	wine wonderful strong aroma bit bitter bite st...	simply wonderful	0.7269	0.5719	0.64940
7	2.0	would give one star came clean bottle called s...	sweet red	0.6908	0.4588	0.57480
8	5.0	delicious affordable		0.5719	0.0000	0.28595
9	5.0	smooth red aroma cocoa coffee tobacco sweet bl...	charles charles red blend	0.8122	0.0000	0.40610

VADER compound 값이 0.05 이상인지 아닌지를 기준으로, 이상이면 1(긍정), 아니라면 0(부정)으로 분류해 라벨을 붙여줍니다.

	reviews.rating	reviews.text	reviews.title	vader_compound	label
0	5.0	fantastic white wine occasion	favorite white wine	0.50810	1
1	5.0	tart sweetvery refreshing delicious	yum	0.28595	1
2	5.0	given wine delightful surprise find flavorful ...	new favorite	0.68530	1
3	5.0	phenomenal wine new favorite red	bold flavorful aromatic delicious	0.60090	1
4	5.0	ml bottle price two way le packaging yes pleas...	yum plus environmentally friendly	0.58040	1
5	5.0	love beck na taste like regular ale smell like...	great taste	0.80100	1
6	3.0	wine wonderful strong aroma bit bitter bite st...	simply wonderful	0.64940	1
7	2.0	would give one star came clean bottle called s...	sweet red	0.57480	1
8	5.0	delicious affordable		0.28595	1
9	5.0	smooth red aroma cocoa coffee tobacco sweet bl...	charles charles red blend	0.40610	1

7번 행을 보면 rating이 2.0인데 감성 분석 결과는 긍정으로 나왔습니다. 위와 같이 별점과 VADER 감성 추론 모델의 결과가 다르게 나올 수 있습니다.



위 그래프는 긍정, 부정 리뷰 별 별점 분포도를 시각화 한 것입니다. 3~5점대 리뷰는 긍정으로 분류한 리뷰가 많지만, 1~2점대 리뷰에서는 제대로 분류하지 못한 모습입니다.

2) 데이터셋 분할

이제 데이터셋을 학습 데이터셋과 테스트 데이터셋으로 분할합니다. 데이터 분할은 8:2 비율로 진행했습니다.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=123)
```

분할 전에 텍스트 데이터는 모두 TF-IDF 벡터화를 해주었습니다.

3) 모델 학습 및 튜닝

모델은 로지스틱 회귀 모델을 사용하였습니다. 하이퍼파라미터 튜닝 기법으로는 그리드 서치를 사용하였습니다.

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV

model = LogisticRegression()

param_grid = {
    "C": [0.001, 0.01, 0.1, 1, 10, 100],
    "max_iter": [100, 1000]
}

grid_search = GridSearchCV(model, param_grid, cv=5, scoring="accuracy")
grid_search.fit(X_train, y_train)

print("Best parameters:", grid_search.best_params_)
print("Test accuracy:", grid_search.score(X_test, y_test))
```

```
Best parameters: {'C': 10, 'max_iter': 100}
Test accuracy: 0.8862745098039215
```

가장 모델 성능이 높았던 파라미터 값은 C는 10, max_iter는 100이었습니다.

최적의 모델로 모델 학습을 진행합니다.

```
# 최적의 모델을 사용 데이터 학습 진행
model = grid_search.best_estimator_
model.fit(X_train, y_train)
yt_pred = model.predict(X_train)
y_pred = model.predict(X_test)
```

4. 평가

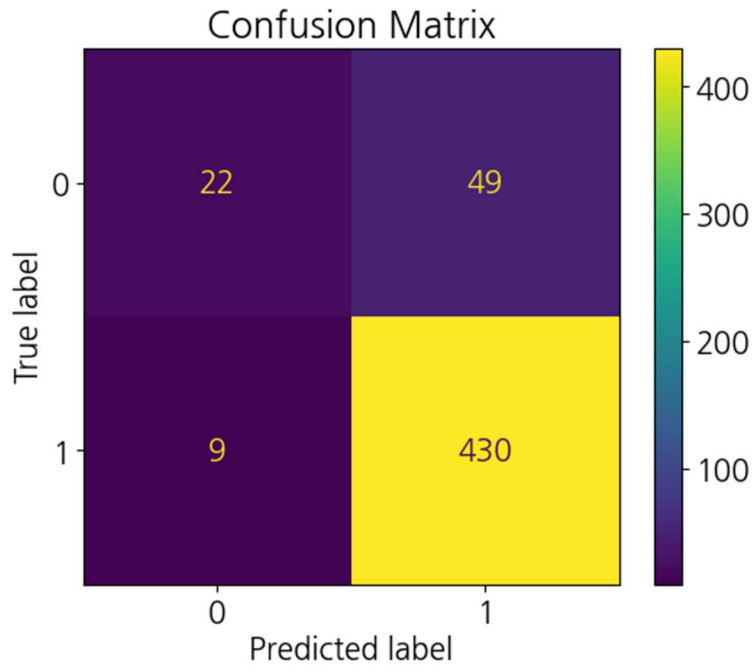
학습된 모델을 평가합니다. 모델 평가 지표로는 정확도, 정밀도, 재현율, F1-score, ROC-AUC를 사용했습니다.

```
정확도   | Train: 0.977941  Test: 0.886275
정밀도   | Train: 0.978218  Test: 0.871527
재현율   | Train: 0.977941  Test: 0.886275
F1-score| Train: 0.977072  Test: 0.866453
ROC-AUC | Train: 0.912748  Test: 0.644679

================== Classification Report (Train) ==================
              precision    recall  f1-score   support

           0       0.99      0.83      0.90       248
           1       0.98      1.00      0.99      1792

    accuracy                           0.98      2040
   macro avg       0.98      0.91      0.94      2040
weighted avg       0.98      0.98      0.98      2040


================== Classification Report ( Test) ==================
              precision    recall  f1-score   support

           0       0.71      0.31      0.43        71
           1       0.90      0.98      0.94       439

    accuracy                           0.89       510
   macro avg       0.80      0.64      0.68       510
weighted avg       0.87      0.89      0.87       510
```

학습 데이터셋의 지표의 경우 높지만, 테스트 데이터셋의 지표는 상대적으로 낮은 것을 확인할 수 있습니다. 특히, ROC-AUC 지표가 매우 낮게 나왔습니다. 이는 모델이 오버피팅이 되었을 가능성을 나타냅니다.

5. 시각화 및 해석

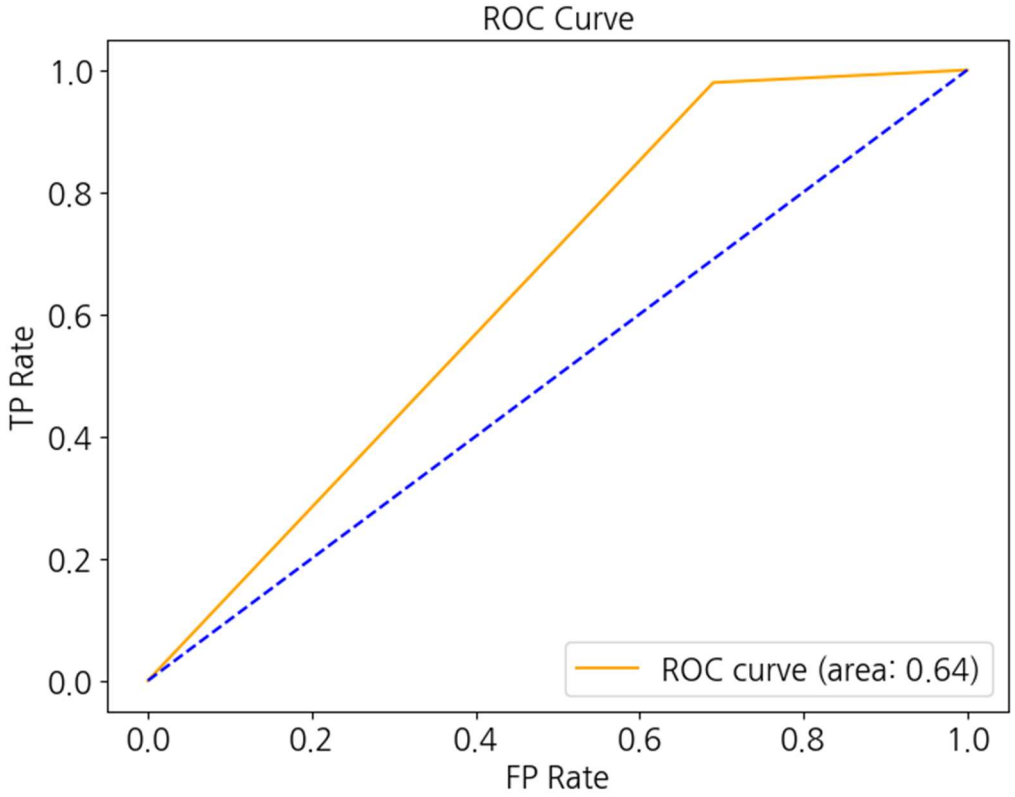
혼동행렬과 ROC 곡선 그래프를 시각화 합니다.



혼동 행렬을 보면 긍정 리뷰에 관해서는 분류를 잘 하였으나, 부정 리뷰에 대해서는 분류를 잘

하지 못한 것을 알 수 있습니다.

다음으로 ROC 곡선을 보겠습니다.



곡선 아래 면적(AUC) 값은 0.5에서 1 사이의 값을 가집니다. AUC 값이 0.5에 가까울수록 모델의 분류 성능이 낮고, 1에 가까울수록 분류 성능이 높습니다.

위 모델의 경우 AUC 값이 0.64로 모델의 성능이 낮음을 확인할 수 있습니다. 반면 학습 데이터의 AUC 값은 0,91로 높았던 것을 생각해보면 오버피팅 문제가 발생하였을 가능성이 높습니다.

6. 결론

학습된 모델을 평가해보았을 때, 학습 데이터의 평가 지표와 테스트 데이터의 평가 지표가 꽤 차이가 나는 것을 확인할 수 있었고, 특히 부정 리뷰에 대한 감성 분류가 테스트 데이터에서 잘 이루어지지 못한 것을 확인할 수 있었습니다.

이는 데이터셋의 불균형이 큰 원인이라고 예상됩니다. 이 문제는 더 많은 리뷰 데이터를 골고루 수집하여 모델의 성능을 강화해 볼 수 있고, 부정 리뷰의 데이터 비율을 늘리거나 긍정 리뷰의 비율을 줄이는 방법으로 해결해 볼 수 있다고 생각합니다.

또한 모델의 경우에도 로지스틱 회귀 모델이 아닌 소프트 벡터 머신(SVM)이나 랜덤 포레스트와 같은 다양한 분류 모델을 사용하여 감성 분석 모델의 성능을 높여볼 수 있을 것으로 보입니다.