**기계학습 기말고사 대체 과제 보고서 - 제품 review 데이터의 감성 분석**

202121027 심유정

1. **감성 분석의 기본 이해**

감성 분석은 텍스트 데이터를 분석하여 그 안에 담긴 감정이나 의견을 파악하는 기술입니다. 주로 자연어 처리 기법을 활용하여 텍스트가 긍정적인지, 부정적인지 또는 중립적인지 판단합니다.

**2. 데이터 탐색**

사용한 데이터는 amazon\_uk\_shoes\_products\_dataset\_2021\_12.csv입니다**.**

* + 1. **데이터 프레임의 기본 정보 확인**

**텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명**

**텍스트, 폰트, 스크린샷, 문서이(가) 표시된 사진

자동 생성된 설명**

전체 행 수는 6823개이고, 열 수는 11개 입니다.

**결측값 개수는**

 review\_title 열에 1개의 결측값이 있습니다.

 review\_text 열에 9개의 결측값이 있습니다.

 helpful\_count 열에 4870개의 결측값이 있습니다.

**텍스트, 폰트, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

**텍스트, 폰트, 영수증, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

대부분의 리뷰 평점이 높은 편이며, 중앙값과 75% 백분위수가 모두 5점으로 나타났습니다. 평균과 중앙값이 비슷한 것으로 보아, 이상치의 영향이 크지 않은 것으로 보입니다.

**3. 데이터 전처리**

**1) 결측값 처리**

****

**2) 노이즈 제거**

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

Tukey의 방법에 따라 이상치를 제거하는 코드입니다. 'review\_rating' 열에서 이상치를 제거했습니다.

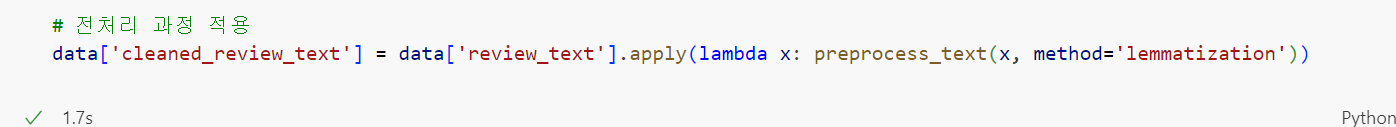
**2) 불용어 제거, 스테밍 및 표제어 추출**

**텍스트, 폰트, 라인, 화이트이(가) 표시된 사진

자동 생성된 설명**

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

****

stopwords.words('english')를 사용하여 영어 불용어 목록을 가져왔습니다.

PorterStemmer : 스테밍 수행

WordNetLemmatizer : 표제어 추출

**preprocess\_text 함수**: 이 함수는 주어진 텍스트를 전처리하는 역할을 합니다. 주요 단계는 다음과 같습니다:

 소문자 변환: text.lower()를 사용하여 텍스트를 소문자로 변환합니다.

 특수 문자 제거: re.sub() 함수를 사용하여 특수 문자를 제거합니다.

 단어 토큰화: word\_tokenize() 함수를 사용하여 문장을 단어로 분할합니다.

 불용어 제거: stop\_words 목록에 포함된 단어를 제거합니다.

 텍스트 정규화: 스테밍 또는 표제어 추출을 선택하여 단어를 정규화 합니다. 여기서는 표제어 추출을 선택합니다.

**전처리 과정 적용**: data['cleaned\_review\_text'] 열에 preprocess\_text 함수를 적용하여 텍스트 데이터를 전처리합니다. 새로운 열 'cleaned\_review\_text'에는 전처리된 텍스트가 저장됩니다.

**3) 감성 레이블 생성**

**텍스트, 폰트, 라인, 번호이(가) 표시된 사진

자동 생성된 설명**

리뷰 평점(review\_rating)을 기반으로 감성 레이블을 생성하는 함수와 이를 데이터프레임에 적용하는 과정입니다. 이 과정을 통해 각 리뷰에 대해 긍정적, 중립적, 부정적 감성을 나타내는 레이블을 추가했습니다.

 평점이 2 이하인 경우: -1 (부정적) //negative

 평점이 3인 경우: 0 (중립적) //neutral

 평점이 4 또는 5인 경우: 1 (긍정적) //positive

apply 메서드를 사용하여 review\_rating 열의 각 값을 sentiment\_label 함수에 전달하고, 그 결과를 새로운 sentiment 열에 저장했습니다.

4) **데이터셋을 학습 및 테스트 세트로 분할**

**텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명**

**4. 모델 선택 및 학습**

**모델**: Logistic Regression

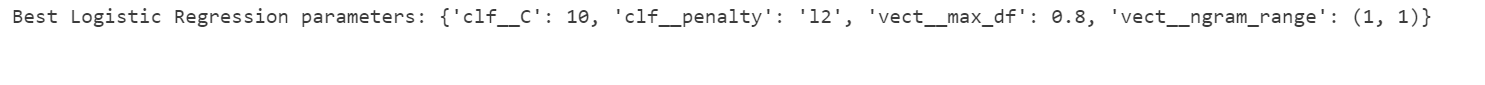
**전처리 파이프라인**: TfidfVectorizer를 사용하여 텍스트를 수치 벡터로 변환하고, Logistic Regression 모델을 학습시켰습니다.

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**텍스트, 폰트, 라인, 화이트이(가) 표시된 사진

자동 생성된 설명**

****

- **Pipeline을 사용하여 두 단계를 설정합니다:**

vect: TfidfVectorizer를 사용하여 텍스트 데이터를 TF-IDF 벡터로 변환합니다.

clf: LogisticRegression 모델을 사용하여 분류 작업을 수행합니다. 여기서는 최대 반복 횟수(max\_iter)를 200으로 설정했습니다.

**-** param\_grid\_lr 딕셔너리는 그리드 서치에서 탐색할 하이퍼파라미터를 정의합니다:

clf\_\_C: 정규화 강도를 조절하는 파라미터로, 값이 작을수록 강한 정규화를 적용합니다.

clf\_\_penalty: 패널티 유형입니다. 여기서는 L2 정규화('l2')만 사용합니다.

vect\_\_max\_df: 단어가 문서에서 나타나는 최대 비율로, 값이 클수록 자주 등장하는 단어를 포함합니다.

vect\_\_ngram\_range: n-그램의 범위로, (1,1)은 단일 단어, (1,2)는 단어 쌍을 포함합니다.

- **그리드 서치 설정**

GridSearchCV를 사용하여 그리드 서치를 설정합니다.

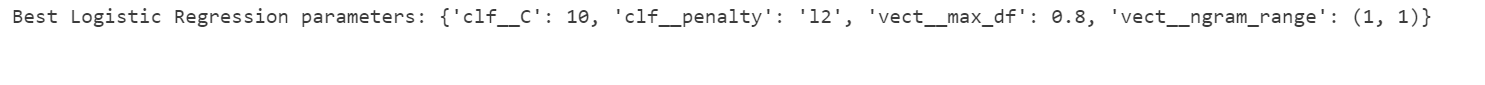
cv=5: 5-폴드 교차 검증을 사용합니다.

scoring='accuracy': 정확도를 기준으로 최적의 하이퍼파라미터를 찾습니다.

**- 최적의 파라미터와 모델**

best\_estimator\_: 그리드 서치에서 최적의 하이퍼파라미터로 학습된 모델을 반환합니다.

best\_params\_: 최적의 하이퍼파라미터 조합을 출력합니다.

****

**정규화 강도(C=10) :** 비교적 약한 정규화가 적용되었습니다. 모델이 데이터에 더 잘 맞추도록 하여 훈련 데이터에 대한 성능을 극대화합니다. 그러나 이는 과도한 학습을 초래할 수 있으며, 이는 오버피팅의 가능성을 높일 수 있습니다.

L2 **패널티 :** L2 정규화는 가중치의 제곱합을 패널티로 사용하여, 모델이 과도한 가중치를 갖지 않도록 조절합니다. 이는 모델의 일반화 성능을 유지하는데 도움이 됩니다.

**최대 문서 빈도**(0.8) : 너무 자주 등장하여 유용한 정보를 제공하지 않는 단어를 제거함으로써, 모델이 더 유의미한 특징에 집중할 수 있도록 합니다.

유니그램(n-gram) : 단어 단위의 특징을 사용하여 텍스트를 벡터화함으로써, 단어 수준에서 텍스트의 감성을 잘 파악할 수 있도록 합니다.

**교차 검증**: 교차 검증을 통해 모델의 견고성을 보장했습니다

**텍스트, 폰트, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

**cross\_val\_score 함수**:

cross\_val\_score 함수는 주어진 모델을 사용하여 교차 검증을 수행합니다.

X\_train과 y\_train 데이터를 사용하여 교차 검증을 수행하며, cv=5는 5-폴드 교차 검증을 의미합니다.

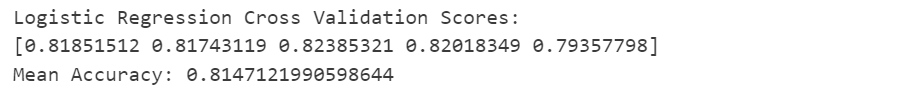
scoring='accuracy'는 모델의 성능 평가 지표로 정확도를 사용합니다.

**교차 검증 점수 출력**:

lr\_cv\_scores는 각 폴드에서의 정확도 점수를 저장합니다.

print(lr\_cv\_scores)는 각 폴드에서의 정확도 점수를 출력합니다.

print("Mean Accuracy:", lr\_cv\_scores.mean())는 모든 폴드에서의 평균 정확도를 출력합니다.

****

**교차 검증 정확도 점수**: [0.81851512, 0.81743119, 0.82385321, 0.82018349, 0.79357798]

**평균 정확도**: 0.8147121990598644

* + - * 1. **교차 검증 점수 분포**

각 폴드의 정확도 점수가 약 0.794에서 0.824 사이에 분포하고 있습니다.

가장 높은 정확도는 0.82385321이고, 가장 낮은 정확도는 0.79357798입니다.

정확도 점수의 변동이 큰 편은 아니므로 모델의 성능이 안정적인 것으로 보입니다.

* + - * 1. **평균 정확도**

평균 정확도는 0.8147로, 모델이 대체로 약 81.47%의 정확도로 예측을 하고 있음을 나타냅니다. 이는 모델이 꽤 좋은 성능을 보이고 있음을 의미합니다.

**4. 평가**

정확도, 정밀도, 재현율, F1 점수 및 ROC-AUC와 같은 지표를 사용하여 모델 평가.

오버피팅과 언더피팅을 논의하고 이를 해결하는 방법 학습.

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**-** best\_lr\_model.predict(X\_test):

최적화된 Logistic Regression 모델 (best\_lr\_model)을 사용하여 테스트 데이터 (X\_test)에 대한 예측을 수행합니다. 예측 결과는 lr\_predictions에 저장됩니다.

- classification\_report(y\_test, lr\_predictions):

y\_test와 lr\_predictions를 비교하여 정밀도(Precision), 재현율(Recall), F1 점수(F1 Score), 지원(Support) 등의 성능 지표를 계산하고, 이를 포괄적으로 보여줍니다.

**-** print("Accuracy:", accuracy\_score(y\_test, lr\_predictions)):

accuracy\_score(y\_test, lr\_predictions)를 사용하여 정확도(Accuracy)를 계산하고 출력합니다. 정확도는 전체 샘플 중에서 올바르게 예측된 샘플의 비율입니다.

**-** confusion\_matrix(y\_test, lr\_predictions) :

y\_test와 lr\_predictions를 비교하여 혼동 행렬(Confusion Matrix)을 생성합니다.

혼동 행렬은 예측 결과와 실제 결과를 비교하여 각 클래스별로 TP, TN, FP, FN의 개수를 나타냅니다.

**모델 평가 결과 해석**

**텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명**

**1) Classification Report**

**Precision, Recall, F1-Score, and Support**:

 **Precision (정밀도)**: 예측된 긍정 클래스 중 실제로 긍정인 비율.

 **Recall (재현율)**: 실제 긍정 클래스 중 올바르게 예측된 비율.

 **F1-Score**: 정밀도와 재현율의 조화 평균.

 **Support**: 각 클래스의 실제 샘플 수.

**Class -1 (Negative Sentiment)**:

* 정밀도: 0.69 (예측된 음성 중 실제로 음성인 비율)
* 재현율: 0.57 (실제 음성 중 올바르게 예측된 비율)
* F1-Score: 0.62
* 지원: 88

**Class 0 (Neutral Sentiment)**:

* 정밀도: 0.25 (예측된 중립 중 실제로 중립인 비율)
* 재현율: 0.07 (실제 중립 중 올바르게 예측된 비율)
* F1-Score: 0.12
* 지원: 40

**Class 1 (Positive Sentiment)**:

* 정밀도: 0.80 (예측된 긍정 중 실제로 긍정인 비율)
* 재현율: 0.94 (실제 긍정 중 올바르게 예측된 비율)
* F1-Score: 0.86
* 지원: 262

**Accuracy**:

* 모델의 전체 정확도: 0.7641 (76.41%)

**Macro Average**:

* 각 클래스의 성능 지표의 단순 평균.
* 정밀도: 0.58, 재현율: 0.53, F1-Score: 0.53

**Weighted Average**:

* 각 클래스의 성능 지표를 지원 (Support) 비율로 가중 평균.
* 정밀도: 0.72, 재현율: 0.76, F1-Score: 0.73
  + 1. **Confusion Matrix**

|  |  |  |  |
| --- | --- | --- | --- |
| Actual \ Predicted | **-1** | **0** | **1** |
| **-1** | **50** | **5** | **33** |
| **0** | **9** | **3** | **28** |
| **1** | **13** | **4** | **245** |

**Class -1 (Negative Sentiment)**:

* 50개의 샘플이 올바르게 음성으로 예측됨 (True Negative).
* 5개의 샘플이 잘못 중립으로 예측됨 (False Neutral).
* 33개의 샘플이 잘못 긍정으로 예측됨 (False Positive).

**Class 0 (Neutral Sentiment)**:

* 9개의 샘플이 잘못 음성으로 예측됨 (False Negative).
* 3개의 샘플이 올바르게 중립으로 예측됨 (True Neutral).
* 28개의 샘플이 잘못 긍정으로 예측됨 (False Positive).

**Class 1 (Positive Sentiment)**:

* 13개의 샘플이 잘못 음성으로 예측됨 (False Negative).
* 4개의 샘플이 잘못 중립으로 예측됨 (False Neutral).
* 245개의 샘플이 올바르게 긍정으로 예측됨 (True Positive).
  + 1. **결론**

**- 결론**

모델이 긍정적 감성(1)을 잘 예측하는 반면, 중립적 감성(0)에 대해서는 성능이 낮습니다.

특히 중립 감성(0)의 재현율이 매우 낮고, 대부분의 중립 리뷰가 긍정으로 잘못 분류됩니다.

**5. 시각화 및 해석**

**텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명**

plot\_confusion\_matrix 함수는 주어진 혼동 행렬(cm)을 시각화합니다.

* cm: 혼동 행렬.
* title: 플롯의 제목.

sns.heatmap: seaborn의 히트맵 함수로, 2D 행렬 데이터를 색상으로 표현합니다.

* cm: 히트맵으로 표현할 데이터.
* annot=True: 셀에 값(annotation)을 표시.
* fmt='d': 정수 형식으로 값을 표시.
* cmap='Blues': 블루 색상 맵을 사용.
* xticklabels: x축의 라벨을 'negative', 'neutral', 'positive'로 설정.
* yticklabels: y축의 라벨을 'negative', 'neutral', 'positive'로 설정.

 plt.ylabel('Actual'): y축 라벨을 'Actual'로 설정.

 plt.xlabel('Predicted'): x축 라벨을 'Predicted'로 설정.

 plt.title(title): 플롯의 제목을 설정.

 plt.show(): 플롯을 화면에 출력.

텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

자동 생성된 설명

**Negative**:

* 실제로 Negative인 리뷰 중에 모델이 Negative로 예측한 경우는 50건입니다.
* 실제로 Negative인 리뷰 중에 모델이 Neutral로 예측한 경우는 5건입니다.
* 실제로 Negative인 리뷰 중에 모델이 Positive로 예측한 경우는 33건입니다.

**Neutral**:

* 실제로 Neutral인 리뷰 중에 모델이 Negative로 예측한 경우는 9건입니다.
* 실제로 Neutral인 리뷰 중에 모델이 Neutral로 예측한 경우는 3건입니다.
* 실제로 Neutral인 리뷰 중에 모델이 Positive로 예측한 경우는 28건입니다.

**Positive**:

* 실제로 Positive인 리뷰 중에 모델이 Negative로 예측한 경우는 13건입니다.
* 실제로 Positive인 리뷰 중에 모델이 Neutral로 예측한 경우는 4건입니다.
* 실제로 Positive인 리뷰 중에 모델이 Positive로 예측한 경우는 245건입니다.

=> 대각선에 위치한 값들은 모델이 정확하게 예측한 경우입니다. 이 값이 높을수록 모델의 성능이 좋다고 볼 수 있습니다. Negative와 Positive 클래스에 대한 정확도는 상당히 높은 편입니다. 그러나 Neutral 클래스에 대한 정확도는 낮은 편입니다. 모델이 Neutral 리뷰를 Negative나 Positive로 잘못 분류하는 경향이 있습니다. Negative나 Positive 리뷰는 비교적 명확한 특징이 있어서 모델이 잘 분류할 수 있지만, Neutral 리뷰는 중립적인 특징이 많아서 분류가 어려운 것으로 보입니다.