기계학습 기말고사 대체과제 보고서

소프트웨어전공 201920995 이상혁

목차

1. 감성 분석의 기본 이해

2. 데이터 전처리

2-1. 데이터 소개

2-2. 데이터 전처리 과정

2-3. 데이터의 텍스트 전처리 과정

3. 데이터셋 분할

4. 모델 선택 및 학습

4-1. 모델 선택

4-2. 학습

5. 평가

6. 시각화 및 해석

6-1. 혼동 행렬 시각화 및 해석

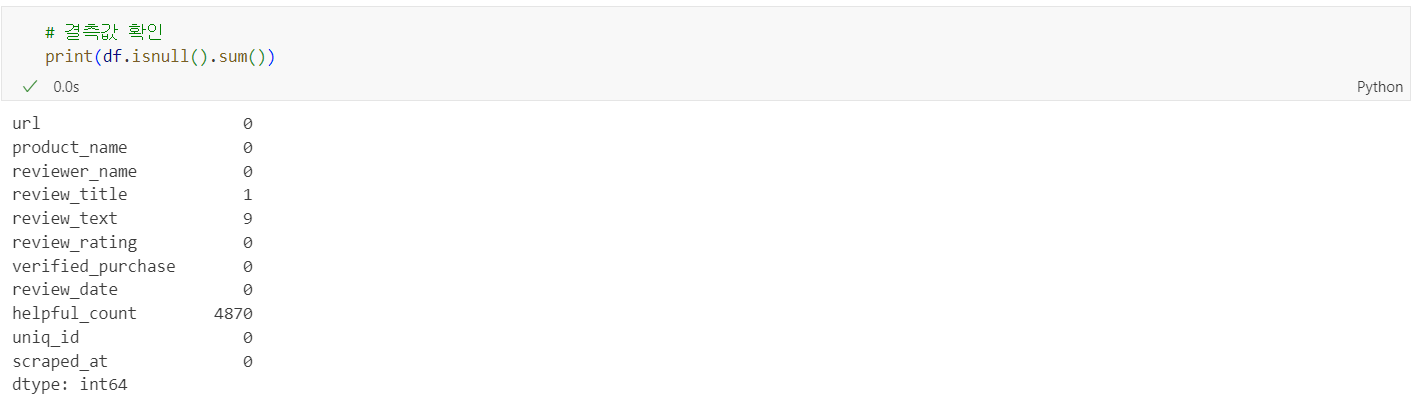
6-2. ROC 곡선 시각화 및 해석

1. 감성분석의 기본 이해

감성 분석은 텍스트 데이터를 긍정, 부정 또는 중립 등의 감성 범주로 분류하는 기법입니다. 이는 주로 고객 리뷰, 소셜 미디어 포스트, 설문 조사 등의 데이터를 분석하는 데 사용됩니다. 감성 분석은 고객 만족도 분석, 마케팅 전략 수립, 제품 개선 등 다양한 응용 분야에서 활용됩니다.

2. 데이터 전처리

2-1. 데이터 소개



데이터는 홀수학번이므로 Amazon UK Shoes Products Dataset 2021/12를 사용했습니다. 이 데이터에는 총 11개의 컬럼으로 구성되어 있는데 url, product\_name, reviewer\_name, review\_title, review\_text, review\_rating, verified\_purchase, review\_date, helpful\_count, uniq\_id, scraped\_at이 있습니다.

2-2. 데이터 전처리 과정



결측값 처리:

- 분석: 'review\_text' 컬럼에서 결측값 9개, 'helpful\_count' 컬럼에서 결측값 4870개를 확인했습니다.

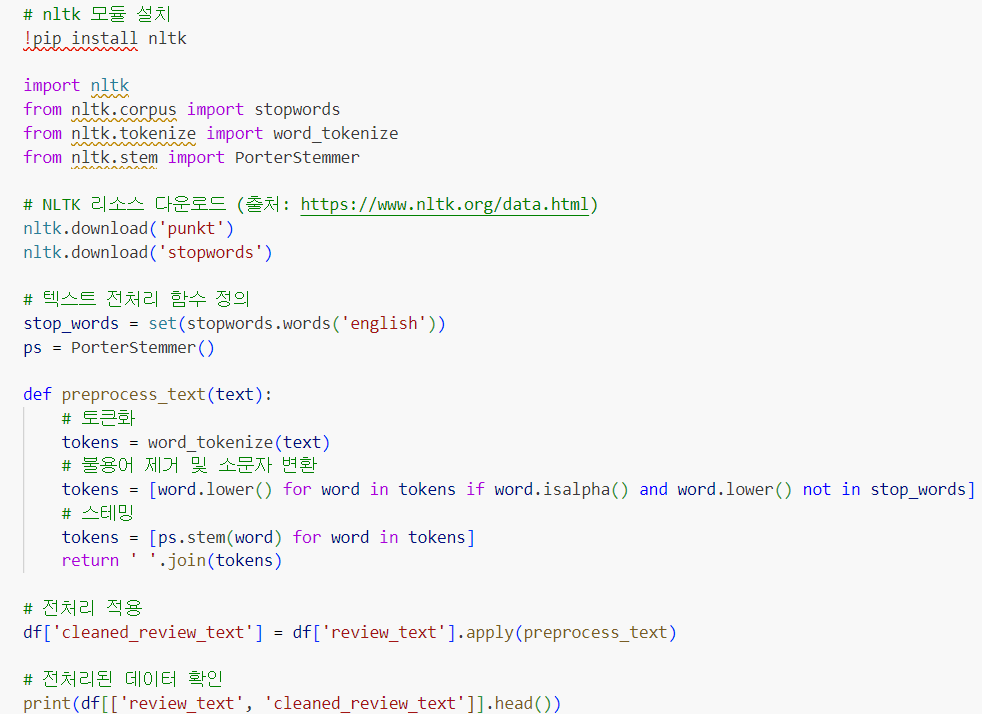
- 처리 방법: 'review\_text'의 결측값 9개를 포함한 행 제거, 중복된 'review\_text'를 가진 행을 제거하여 데이터 중복성을 줄였습니다.

처리 결과:

- 결측값 처리 후: 'review\_text'의 결측값은 모두 제거되어 0개로 확인되었고 'helpful\_count'의 결측값은 4870개에서 4653개로 감소, 처리되지 않은 결측값이 남아 있습니다.

- 중복값 처리 후: 'review\_text' 컬럼을 기준으로 중복된 데이터를 제거하여 중복값이 0개 확인됐습니다.

2-3. 데이터의 텍스트 전처리 과정



NLTK 모듈 설치 및 리소스 다운로드: nltk모듈을 설치하고, 필요한 리소스인 'punkt'와 'stopwords'를 다운로드합니다. 'punkt'는 토큰화(tokenization)를 위한 데이터, 'stopwords'는 불용어(stopwords)를 제거하기 위한 데이터입니다.

텍스트 전처리 함수 정의:

- 토큰화(Tokenization): word\_tokenize()함수를 사용하여 문장을 단어 단위로 쪼갭니다.

- 불용어 제거(Stopword Removal): stopwords.words('english')를 사용하여 영어 불용어를 세트로 정의하고, 이를 기반으로 tokens에서 불용어를 제거합니다.

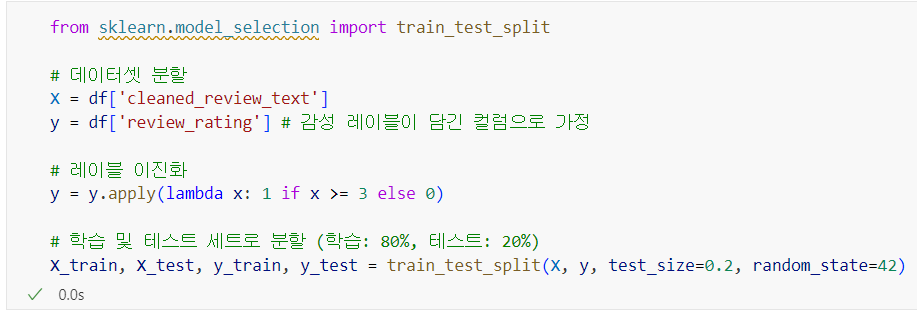
- 소문자 변환: 각 단어를 소문자로 변환하여 일관된 처리를 합니다.

- 스테밍(Stemming): PorterStemmer()를 사용하여 각 단어를 어간 추출하여 표준화합니다.

전처리 적용: df['cleaned\_review\_text']열에 preprocess\_text함수를 적용하여 'review\_text' 열의 각 텍스트 데이터를 전처리합니다. 전처리 과정은 텍스트의 품질을 향상시키고, 머신 러닝 모델에 적합한 형태로 변환하는 역할을 합니다.

전처리된 데이터 확인: print(df[['review\_text', 'cleaned\_review\_text']].head())를 통해 원본 텍스트와 전처리된 텍스트를 비교하여 확인합니다.

3. 데이터셋 분할



- 데이터 준비:

1) cleaned\_review\_text: 텍스트 전처리 과정을 거친 후의 리뷰 텍스트 데이터입니다.

2) review\_rating: 각 리뷰에 대한 별점을 나타내는 레이블입니다.

- 레이블 이진화: review\_rating컬럼의 값을 기준에 따라 이진 분류 형태로 변환합니다. 여기서는 별점이 3 이상인 리뷰를 긍정적(1), 3 미만인 리뷰를 부정적(0)으로 설정하였습니다.

- 학습 및 테스트 세트 분할:

1) train\_test\_split함수를 사용하여 데이터셋을 학습 세트(X\_train, y\_train)와 테스트 세트(X\_test, y\_test)로 분할하였습니다.

2) 데이터 분할 비율은 학습 세트 80%, 테스트 세트 20%로 설정하였고, random\_state를 42로 고정하여 재현성을 확보하였습니다.

4. 모델 선택 및 학습



4-1. 모델 선택: TF-IDF 벡터화와 로지스틱 회귀 분류기를 결합하여 감성 분석 모델을 구성하는 과정을 담고 있습니다.

- TF-IDF 벡터화 (TfidfVectorizer):

1) TF-IDF(Term Frequency-Inverse Document Frequency)는 텍스트 데이터를 수치화하는 방법 중 하나입니다. 각 단어의 상대적인 중요도를 계산하여 벡터화합니다.

2)TfidfVectorizer클래스는 텍스트 데이터를 TF-IDF 값으로 변환하는 역할을 합니다.

- 로지스틱 회귀 분류기 (LogisticRegression):

1) 로지스틱 회귀는 이진 분류 모델로, 선형 방정식을 사용하여 입력 특성과 각 클래스 사이의 관계를 학습합니다.

2) LogisticRegression클래스는 데이터를 학습하여 각 클래스(긍정적인 리뷰와 부정적인 리뷰)에 속할 확률을 예측합니다.

- 파이프라인 구성: Pipeline클래스를 사용하여 TF-IDF 벡터화와 로지스틱 회귀 분류기를 연결하였습니다.

4-2. 학습

- 학습 (Model Training): model.fit(X\_train, y\_train)코드를 통해 학습 세트(X\_train, y\_train)를 이용하여 모델을 학습시킵니다.

- 예측 (Prediction): 학습된 모델을 사용하여 테스트 세트(X\_test)에 대해 예측을 수행하고, 이를 y\_pred변수에 저장합니다.

- 평가 지표 계산: accuracy\_score, precision\_score, recall\_score, f1\_score함수를 사용하여 모델의 성능을 평가합니다.

Accuracy (정확도): 전체 예측 중 올바르게 예측한 비율

Precision (정밀도): 긍정으로 예측한 것 중 실제로 긍정인 비율

Recall (재현율): 실제 긍정 중 긍정으로 예측한 비율

F1 Score (F1 점수): Precision과 Recall의 조화 평균으로 계산된 지표

-결과:



Accuracy: 0.848

Precision: 0.850

Recall: 0.992

F1 Score: 0.915

5. 평가

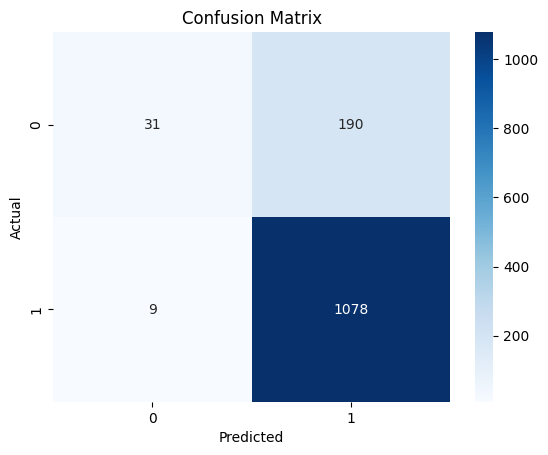
이 모델은 전반적으로 높은 정확도와 재현율을 보이며, 특히 Recall이 높아 실제 긍정 리뷰를 대부분 잘 예측하는 경향을 보입니다. Precision 역시 상당히 높아 잘못된 긍정 예측이 적습니다. F1 Score는 Precision과 Recall의 조화 평균으로, 모델의 전반적인 성능을 평가하는 중요한 지표입니다.

6. 시각화 및 해석

혼동 행렬과 ROC 곡선을 시각화하기 위한 코드

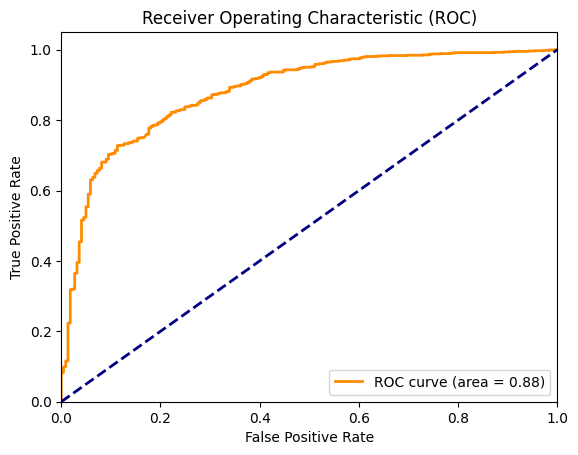


6-1. 혼동 행렬 시각화 및 해석



혼동 행렬을 시각화한 것을 보면 잘못 분류한 데이터는 총 199개가 나왔고 잘 분류한 데이터는 총 1109개가 나온 것을 볼 수 있습니다.

6-2. ROC 곡선 시각화 및 해석



ROC 곡선을 시각화한 그래프를 보면 x=y 그래프인 파란색 그래프는 기준선을 표시한 것이고 곡선을 그리는 노란색 그래프가 ROC 그래프입니다. ROC 그래프 아래면적이 0.88로 0.9에 가깝기 때문에 우수한 성능을 보여주는 모델인것 같습니다.