기계학습(8585)

기말고사 대체과제

제품 review 데이터의 감성 분석

202020835 김민서

1. 감성 분석의 기본 이해

감성 분석은 텍스트 데이터를 분석하여 감정이나 의견의 긍정적, 부정적, 중립적 측면을 식별하는 자연어 처리의 한 분야입니다. 감성분석의 기본 원리는 텍스트 전처리, 토큰화, 단어 임베딩, 감성 사전, 모델 학습 및 예측입니다. 감성분석의 응용 분야로는 마케팅 및 브랜딩, 고객 서비스, 영화 및 콘텐츠 추천 등이 있습니다.

1. 데이터 전처리

2-1. 데이터 정리

* 분석에 필요한 값만 사용
* 알파벳과 공백을 제외한 모든 문자 제거 및 소문자 변환

**결과**

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

텍스트, 영수증, 폰트, 화이트이(가) 표시된 사진

자동 생성된 설명

특수기호가 사라진 것을 확인할 수 있습니다.

2-2. 텍스트 토큰화 및 불용어 제거

* 텍스트 데이터의 의미 있는 부분만을 남기고 분석의 정확도를 높이기 위해 NLTK 라이브러리를 사용하여 텍스트를 토큰화하고 불용어를 제거하는 작업을 수행
* stopwords.words("english")로 불용어 가져오기. "the", "is", "in"과 같은 분석에서 큰 의미를 가지지 않는 단어
* word\_tokenize(text)로 입력된 텍스트를 단어 단위로 토큰화

**결과**

텍스트, 영수증, 폰트, 화이트이(가) 표시된 사진

자동 생성된 설명

의미 없는 was, the, very와 같은 단어가 제거되었고 토큰화 된 것을 알 수 있습니다.

2-3. 텍스트 정규화 – 스테밍

* 단어의 어근을 추출하여 단어를 표준화하는 작업
* [stemmer.stem(word) for word in tokens]로 각 단어에 대해 스테머를 적용하여 어근을 추출, 추출된 어근들은 새로운 리스트(stemmed\_tokens)에 저장

**결과**

텍스트, 영수증, 폰트, 화이트이(가) 표시된 사진

자동 생성된 설명

Looking -> look, started -> start 등으로 보아 어근을 추출하여 새로운 리스트에 적은 것 확인 가능합니다.

1. 모델 선택 및 학습

감성 분석 시스템은 긍정과 부정의 두 가지 클래스로 예측을 합니다. logistic 회귀는 이진 분류를 할 수 있기에 logistic 회귀를 모델로 선택했습니다.

* vectorizer = TfidfVectorizer(max\_features=1000)로 텍스트 데이터를 TF-IDF 값으로 변환
* Review\_rating열의 값에 따라 감성 레이블을 설정합니다. 여기서는 4점 이상인 리뷰를 긍정(1), 3점 이하는 부정(0)으로 분류
* train\_test\_split(X, y, test\_size=0.2, random\_state=42)

전체 데이터의 20%를 테스트 세트로 사용하고 나머지 80%를 훈련 세트로 사용

**X\_test에 대한 예측 값을 계산한 결과**



0과 1로 잘 분류되었습니다.

Hyperparameters 그리드를 사용하여 로지스틱 회귀 모델을 훈련하고 교차 검증을 통해 최적의 Hyperparameters를 찾고 그 성능을 평가

**결과**

텍스트, 폰트, 화이트, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

C가 10으로 선택된 것은 모델이 훈련 데이터에서 규제를 약하게 하여, 복잡성을 증가시키면서 최적의 성능을 달성했다는 것을 나타냅니다. 이는 언더피팅을 줄이고 모델의 복잡성을 적절히 조정했다고 볼 수 있습니다.

정확도가 약 83.49% 정도로 일반화 성능이 좋을 것이라는 기대를 나타냅니다.

* cross\_validate() 교차 검증 함수를 사용

**결과**

{'fit\_time': array([0.54816961, 0.58683228, 0.62246585, 0.58998036, 0.54085636]), 'score\_time': array([0.0008893 , 0.00581002, 0.00090361, 0.00145364, 0.00089145]), 'test\_score': array([0.82417582, 0.83241758, 0.82051282, 0.85334555, 0.84417965])}

결과

'fit\_time': 모델을 훈련하는 시간

'score\_time': 검증하는 시간

'test\_score': 교차 검증의 최종 점수는 'test\_score'의 5개 점수를 평균하여 얻음



약 83.49%의 교차검증 점수가 나왔습니다.

* 로지스틱 회귀 모델의 강점
* 이진 데이터를 분석하고 변수 간의 관계에 통찰력을 제공하는 데 사용할 수 있습니다.
* 학습 속도가 매우 빨라 구현하기 쉽고, 결과 해석이 수월합니다.
* 약점
* 신뢰할 수 있는 결과를 얻기 위해 큰 표본 크기가 필요합니다. 표본 크기가 너무 작으면 신뢰할 수 없고 추정치가 불안정합니다.
* 로지스틱 회귀는 선형 결정 경계만을 학습할 수 있습니다. 따라서 입력 데이터가 복잡하게 구성되어 있거나 클래스가 비선형적으로 분포되어 있는 경우에는 성능이 제한될 수 있습니다.

1. 평가

텍스트, 영수증, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

정확도: 85.42%로 모델을 정확하게 예측

정밀도: 87.14%의 긍정 예측이 실제로 긍정적

재현율: 94.67% 실제로 긍정적인 모델 중 모델이 정확하게 긍정으로 예측

F1 점수: 90.75% 정밀도와 재현율의 조화 평균

ROC-AUC 점수: 값이 1에 가까울수록 모델이 양성 클래스를 잘 분류하고 있다고 나타냄 89.6%로 좋은 성능

전체적으로 85% 이상의 값으로 모델이 잘 훈련된 것을 알 수 있습니다.

언더피팅과 오버피팅

* 오버피팅은 Train 데이터에 모델이 너무 적합하게 학습되어 있기 때문에 Train 데이터가 아닌 다른 데이터가 들어오게 되면 정확도가 떨어지는 현상입니다.

오버피팅이 발생하면 모델은 훈련 데이터에서는 높은 성능을 보이지만, 새로운 데이터나 검증 데이터에서는 성능이 저하될 수 있습니다.

* 언더피팅은 모델이 너무 단순하거나, 학습 데이터의 다양성을 충분히 반영하지 못하는 경우 발생할 수 있습니다. 또한, 모델이 훈련 데이터의 패턴을 충분히 학습하지 못해, 훈련 데이터와 테스트 데이터 모두에서 성능이 낮은 상태를 말합니다

- 오버피팅 해결 방법

* 데이터 양 늘리기

모델은 데이터의 양이 적을수록 해당 데이터의 특징 패턴이나 노이즈까지 암기해버려서 오버피팅 될 확률이 높습니다. 그래서 데이터의 양을 늘릴수록 모델은 일반적인 패턴을 학습하여 방지할 수 있습니다.

* 모델의 복잡도 줄이기

모델의 능력을 줄인다고 표현할 수 있으며 이것은 데이터 양을 늘려주는 것과 반대로 학습하는 데이터의 수준에 맞게 모델의 능력을 조정해주는 방법입니다.

* 언더피팅 해결 방법
* 모델 복잡성 증가

모델의 복잡성을 높이거나, 다양한 특성을 추가하여 학습 데이터의 다양성을 반영하도록 합니다.

* 더 복잡한 모델 사용

모델의 복잡성을 늘려서 학습 데이터에 더 잘 맞출 수 있도록 합니다.

위 모델에서 훈련데이터의 성능은 0.8349, 검증 데이터의 성능은 0.8349262866로 성능이 낮지 않으며 밸런스가 맞음으로 오버피팅과 언더피팅에 해당하지 않습니다.

1. 시각화 및 해석: 결과를 시각화하고 해석

**혼동행렬**

텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

자동 생성된 설명

실제 긍정(1)인 리뷰일 때 예측도 긍정인 경우가 1006건으로 다른 라벨에 비해 긍정을 분류하는 데 좋은 성능을 보입니다. 또한, 부정(0)인 리뷰에 부정으로 예측하는 경우가 133건으로 총 1365건 중 1139건이 옳게 답했음을 알 수 있습니다. 하지만 부정인 리뷰를 긍정으로 예측하는 경우가 201건으로 두번째로 높은 것을 보아 부정인 리뷰를 긍정으로 인식하는 경우가 종종 있음을 나타냅니다.

**ROC 곡선**

텍스트, 라인, 그래프, 도표이(가) 표시된 사진

자동 생성된 설명

곡선이 대각선 위에 있고, 왼쪽 상단에 가까운 위치를 차지하고 있는 것을 보아 모델의 성능이 좋다고 볼 수 있습니다.