Report

28기 이근하

1. **데이터 로드 및 기본 탐색**

   creditcard.csv의 데이터는 Time, V1~V28, Amount, Class의 31개의 열과 0,1,2,3,4의 총 5개의 행으로 이루어져 있다.

   이때 정상 거래 수는 284,315개, 사기 거래 수는 492개이다.

2. **샘플링**

   샘플링 전의 Class의 비율은 정상 거래가 약 99.8273, 사기 거래가 약 0.1727이다.

   정상 거래의 10,000건을 무작위로 샘플링 한 이후의 Class의 비율은 정상 거래가 약 95.3107, 사기 거래는 약 4.6893이다. 즉, 샘플링을 진행하였을 때, 정상 거래의 비율이 약간 증가하였음을 알 수 있었다.

3. **데이터 전처리**

   X는 Class 열을 제외한 데이터프레임이며, y는 Class 열만을 포함한 데이터프레임으로 분할하였다.

4. **학습 데이터와 테스트 데이터 분할**

   데이터프레임 y의 학습 셋 Class 비율은 정상 거래가 약 95.3056, 사기 거래가 약 4.6944이다. 데이터프레임 y의 테스트 셋 Class 비율은 정상 거래가 약 95.3311, 사기 거래가 약 4.6689으로 학습 셋과 테스트 셋의 Class 비율이 굉장히 유사하다는 점을 확인할 수 있었다.

5. **SMOTE 적용**

   현재 정상 거래와 사기 거래의 데이터가 굉장히 많은 차이를 보인다. 즉, 이러한 모델을 학습시킬 경우에 전부 정상이라고 예측을 해도 정확도가 높은 문제가 발생할 수 있다. 즉, 적은 클래스 (사기 거래)를 단순히 복사하는 것이 아닌, 생성하여 증가시키는 방식인 SMOTE를 사용하여서 현실을 반영하면서 새로운 사기 거래 데이터를 생성할 수 있다.

   SMOTE 적용 전의 사기 거래 건수는 394건, SMOTE 적용 후의 사기 거래 건수는 7999건으로 SMOTE 적용을 통해서 사기 거래 건수가 정상 거래 건수와 동일하게 조정된 것을 확인할 수 있었다.

6. **모델 학습**

   모델 학습을 위해서 Logistic Regression(로지스틱 회귀) ML 모델을 선정하였다.

   이때, predict은 대부분의 결과가 0으로 출력되고, predict_proba의 경우 1.23347333e-01, 7.26815827e-02 등 사기 거래가 발생될 확률로 나타나게 된다.

정상 거래에서의 Precisoin은 0.9970, Recall은 0.9880, F1-score는 0.9925로 측정되었다.

사기 거래에서의 Precisoin은 0.7931, Recall은 0.9388, F1-score는 0.8598로 측정되었다.

PR-AUC의 값은 약 0.9539로 측정되었다.

이때, 다른 조건들은 만족되었지만, 사기 거래의 F1-score이 목표치를 달성하지 못하였다. 이에 현재 0.5로 설정된 Threshold를 증가시켜보고자 하였다.

7. 최종 성능 평가

Threshold를 증가시킬 경우 Recall은 감소하고 Precision은 증가하여 F1-score이 증가하는 결과를 일으킬 수 있다. 이에 F1-score의 값을 증가하여 목표치를 달성하기 위해서 Threshold 값을 0.6으로 조정해보았다.

정상 거래에서의 Precisoin은 0.9965, Recall은 0.9930, F1-score는 0.9947로 측정되었다.

정상 거래에서의 Precisoin은 0.8667, Recall은 0.9286, F1-score는 0.8966로 측정되었다.

PR-AUC의 값은 약 0.9539로 측정되었다.

이는 최종적으로 목표치인 Recall >= 0.80, F1-score>= 0.88, PR-AUC>=0.90을 충족하여서 적합한 모델이라고 평가할 수 있다.