1. 데이터 탐색

Creditcard.csv는 거래시간(time), 거래 특성(v1~v28), 거래 금액(amount), 사기 여부(class)로 구성 되어있다. 총 데이터 수는 284807개이며 이 중 99.82%가 정상 거래, 나머지 약 0.18%가 사기거래이다.

2. 샘플링

사기 거래는 유지하며 정상거래만 10,000건으로 무작위 추출하였다. 샘플링 후에는 정상거래가 95.31%, 사기 거래가 약 4.68%로 사기거래의 비율이 증가하였다.

3. 전처리

거래금액은 큰 스케일을 고려하여 StandardScaler를 사용하여 정규화를 진행했다.

4. 학습 / 테스트 데이터 분할

학습, 테스트 데이터를 8:2 비율로 나누었다. 각 데이터셋에서 정상, 사기 거래 비율은 다음과 같다:

Y_train:

Class

0 0.953056

1 0.046944

Y_test:

Class

0 0.953311

1 0.046689

5. SMOTE

데이터 탐색 / 샘플링 단계에서 보이듯, 사기거래의 건수는 일반 거래에 비해 굉장히 낮음을 확인할 수 있다. 이런 데이터를 그래도 사용하여 모델을 학습시킨다면, 모델은 소수 케이스에 대한 학습을 제대로 할 수 없고, 대부분의 데이터를 정상거래로 예측하게 될 것이다.

따라서 SMOTE를 사용해 주어진 데이터를 기반으로 사기거래에 대한 새로운 data point들을 만들어 학습을 도울 필요가 있다. SMOTE 적용 전,후 train data에서 사기거래 정상거래의 수는 다음과 같다:

SMOTE 이전:

Class

0 7999

1 394

SMOTE 이후:

Class

0 7999

1 7999

6. 모델 및 성능 평가

분류 모델은 RandomForestClasifier를 사용하였고 class_weight='balanced' 옵션을 통해 사기 거래에 더 큰 중요도를 부여하고자 했다.

모델 예측의 성능은 다음과 같이 나왔다:

Class	*Precision*	*Recall*	*F1-score*	*Support*
0	0.99	0.99	0.99	2001
1	0.88	0.89	0.88	98

PR-AUC : 0.9505

두 클래스에 모두 Recall은 0.8, F1 score는 0.88을 넘기는 수치를 달성하며 대부분의 사기거래를 탐지하였다. PR-AUC 역시 0.95로 높은 값을 보여 불균형 데이터 상황에서도 우수한 분류 성능을 보였다.

사기 거래에 대한 Recall을 높이기 위해 threshold를 0.35로 설정한 결과, recall 점수는 0.95로 크게 높아졌지만, f1 점수가 0.86으로 떨어지며 기준치에 도달하지 못했다.