

# ML 과제 결과 보고서

YBIGTA 28기 첨단컴퓨팅학부 임수빈

## 1. 개요

본 과제는 신용카드 사기 탐지(Credit Card Fraud Detection) 데이터셋을 활용하여, 클래스 불균형이 심한 데이터를 처리하고 분류 모델을 만들어보는 것을 목표로 했습니다. 머신러닝을 처음 접해보는 입장에서 단순히 정확도(Accuracy)만 보는 것이 아니라, Recall과 F1-score 등 다양한 지표의 중요성을 이해하는 데 중점을 두었습니다.

## 2. SMOTE 적용 이유 및 결과

처음 데이터를 까봤을 때 정상 거래(Class 0)에 비해 사기 거래(Class 1) 비율이 말도 안 되게 적다는 걸 확인함. 이런 '불균형 데이터' 상태에서 모델을 그냥 학습시키면, 모델이 대다수인 정상 거래만 맞추려 하고 소수인 사기 거래는 무시하는 편향이 생길 게 뻔했음.

사기 탐지 시스템에서는 실제 사기(1)를 정상(0)이라고 놓치는 것(False Negative)이 제일 치명적임. 그래서 모델이 소수 클래스인 사기 패턴을 제대로 학습할 수 있도록 데이터를 늘려줄 필요가 있었음.

이를 위해 데이터를 단순히 복붙하는 게 아니라, 데이터 간의 특성을 반영해 새로운 데이터를 합성해 주는 SMOTE(Synthetic Minority Over-sampling Technique) 기법을 써서 Class 1 데이터를 증강시킴.

## 3. 모델 선정 및 학습

분류 문제에서 성능이 우수하고 과적합 방지에 유리하다고 알려진 Random Forest 모델을 선택했습니다.

- 사용한 모델: RandomForestClassifier
- 사용한 파라미터: random\_state=42, n\_jobs=-1(나머지는 기본값 사용)

## 4. 최종 성능 평가

### 4-1. 평가 개요

학습이 완료된 Random Forest 모델을 검증하기 위해, 학습 과정에서 전혀 사용되지 않은 테스트셋(Test Set) 20%를 활용하여 최종 성능을 측정하였음. 불균형 데이터 특성상 Accuracy(정확도)보다는 사기 거래(Class 1)를 얼마나 잘 찾아내는지가 중요하므로 Recall, F1-score, PR-AUC를 중점적으로 확인함.

### 4-2. 상세 지표 달성을 현황

테스트 결과, 과제에서 제시한 모든 정량적 목표치를 여유 있게 초과 달성하는 쾌거를 거둠.

- Recall (재현율): 0.89(목표 0.80 대비 +0.09달성)
- F1-score: 0.92(목표 0.88 대비 +0.04달성)
- PR-AUC: 0.9538(목표 0.90 대비 +0.0538달성)

### 4-3. 지표별 상세 해석

#### - Recall (0.89)의 의미:

신용카드 사기 탐지 시스템에서 가장 핵심적인 지표임. 실제 사기 거래 100건 중 89건을 모델이 정확히 '사기'라고 잡아냈다는 뜻임.

SMOTE를 통해 사기 데이터의 패턴을 충분히 학습시킨 덕분에, 놓치는 사기 거래(False Negative)를 최소화할 수 있었음.

#### - F1-score (0.92)의 의미:

Recall을 높이려다 보면 정상 거래를 사기로 잘못 분류(Precision 하락)하기 쉬운데, 0.92라는 높은 점수는 두 지표 간의 균형이 매우 훌륭함을 보여줌.

즉, 사기도 잘 잡으면서 억울한 정상 사용자(False Positive)도 거의 만들지 않는 이상적인 모델임.

#### - PR-AUC (0.9538)의 의미:

0.95가 넘는 수치는 모델의 예측 확률(Probability) 자체가 매우 신뢰할 만하다는 것을 증명 함. 특정 Threshold에 의존하지 않고 전반적으로 안정적인 분류 성능을 보여줌.

#### 4-4. 종합 결론

별도의 복잡한 임계값(Threshold) 튜닝 없이도 기본 Random Forest 모델과 SMOTE 전처리만으로 목표 성능을 초과 달성함. 이는 데이터 불균형 해소가 모델 학습에 얼마나 지대한 영향을 미치는지 보여주는 결과이며, 실제 금융 시스템에 적용해도 손색없을 정도의 유의미한 성능을 확보했다고 판단됨.

### 5. 고찰 및 결론

처음엔 "사기 거래 다 잡으려면 무조건 Recall 1.0 찍어야 되는 거 아님?"이라고 단순하게 생각했는데, Precision이랑 Trade-off 관계라 마냥 그럴 수는 없다는 걸 배움.

원래 계획은 성능이 목표치에 미달되면 predict\_proba 임계값(Threshold)을 조정해서 Recall을 억지로라도 높여볼 생각이었음. 하지만 막상 돌려보니 별다른 튜닝 없이 기본 모델(Random Forest)에 SMOTE만 적용했는데도 목표치를 가뿐히 초과 달성함.

결과적으로 불균형 데이터 문제 해결에는 복잡한 모델 튜닝보다 확실한 데이터 전처리(SMOTE)가 훨씬 결정적인 역할을 한다는 걸 체감함.