기초 이론부터 실무 실습까지

머신 러닝 익히기

programmers

Part 08. 이상 탐지

정 정 민

programmers

Chapter 18. Isolation Forest 실습

1. Credit Card Fraud Detection
2. EDA 및 전처리
3. 이상치 분석
4. 평가

Credit Card Fraud Detection

programmers

Credit Card Fraud Detection 데이터

- 이번 실습에서 사용할 데이터로 Kaggle의 공개 데이터 (링크)
- 신용카드 거래에서의 사기 탐지를 위해 설계된 데이터셋
  - 2013년 유럽 카드 소지자들의 거래 데이터를 포함
  - 총 284,807건의 거래 데이터가 있고
  - 그 중 492건(0.172%)의 사기 거래 데이터가 있음

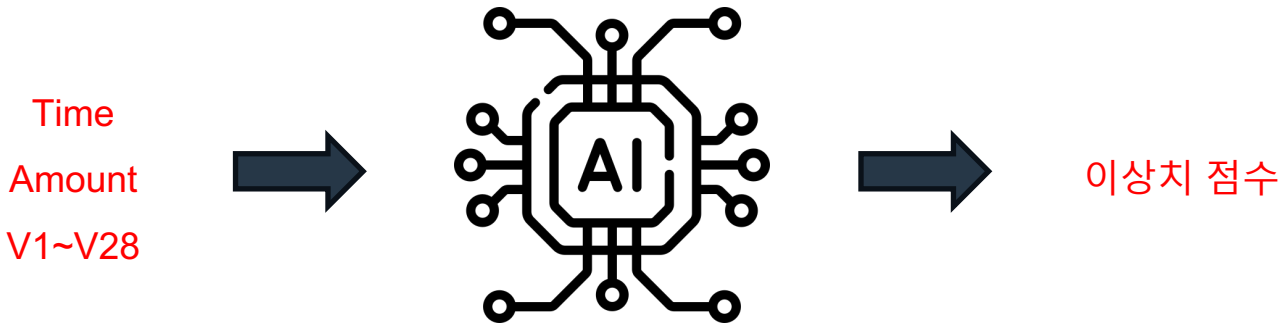
- 데이터의 크기가 너무 커서 10MB 정도로 랜덤 샘플링한 데이터 (링크)
  - 다운로드 받아주세요!
  - 전체 19,936건, 사기 34건 (0.171%) 데이터를 포함

- 변수로는 Time, Amount, V1~V28, Class
  - Time : 첫 거래 후 각 거래 사이 경과 시간 (초)
  - Amount : 거래 금액
  - V1~V28 : PCA로 얻은 수치형 입력 변수
  - Class : 정상 거래 (0)와 비정상 거래 (1)

문제 정의

- 풀어야 하는 문제
  - 주어진 거래 관련 데이터를 바탕으로 이상 거래 데이터를 탐지
    독립 변수

- 머신 러닝 모델의 입, 출력 정의
  - 입력 : 거래 관련 데이터
    - Time
    - Amount
    - V1 ~ V28
  - 출력 : 각 데이터 포인트 마다 할당된 이상치 점수 (종속 변수)
    - 후처리 필요



EDA 및 전처리

강의 실습 코드 : 링크

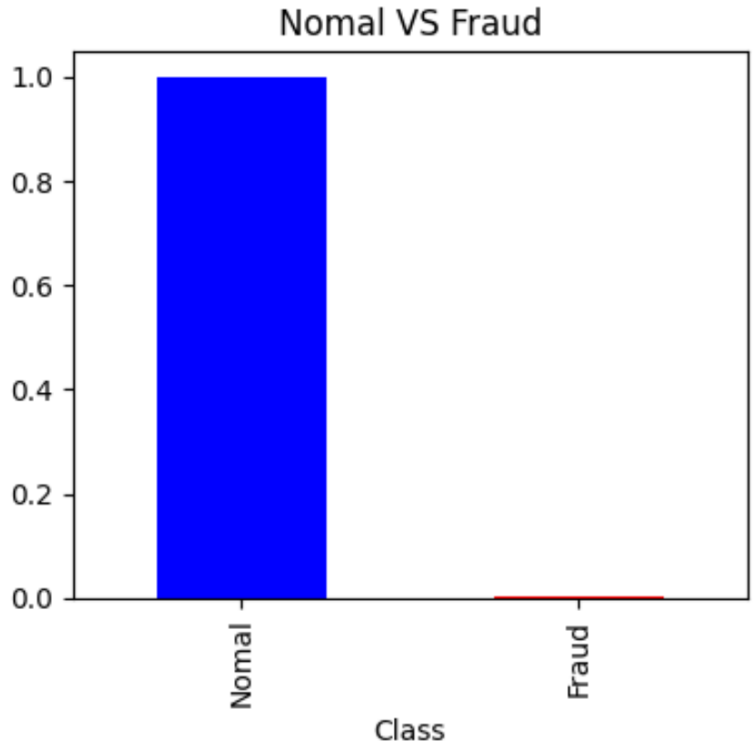
데이터 타입 및 고려사항

- **Class**
  - 범주형 데이터
  - 매우 심한 치우침이 있고 이상치를 탐지해야 함
- **Time, Amount**
  - 수치형 데이터
  - 값의 범위가 클 수 있음
  - 스케일링이 필요할지 확인해야 함
- **V1 ~ V28**
  - 수치형 데이터
  - 차원 축소의 결과물 데이터로 의미를 알기 쉽지 않음
  - 수치적으로 어떠한 특성과 분포를 갖고 있는지 확인해야 함

Class 시각화

- 매우 극단적인 치우침이 보임

- 즉, Fraud를 이상치로 판단할 수 있고
- 이상치 탐지 접근 방식으로 이상치를 탐지해야 함



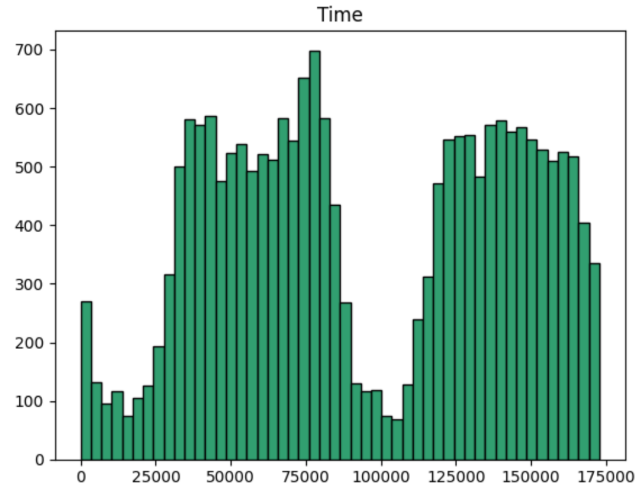
Time과 Amount 시각화

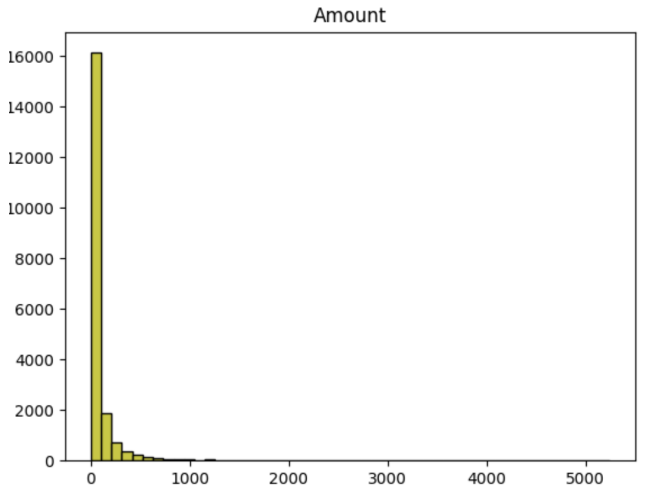
- **Time의 경우** 거래 시작 이후의 경과 시간을 나타내므로
- 기준 시간 정보가 있다면 주기성을 갖고 있을 수 있음
  - 낮에 한 거래와 밤에 한 거래와 같이
  - 그렇다면 주기 함수를 적용하는 것도 좋은 방법

- 하지만 여기에서는 추가적인 정보가 부족
- 큰 outlier가 없어 보이므로 Min-Max Scaling 을 사용

- **Amount**는 치우침이 있지만
- 신용 카드 거래 금액으로 의미가 없는 데이터는 아님

- 이를 처리하기 위해 Log 스케일링을 진행!





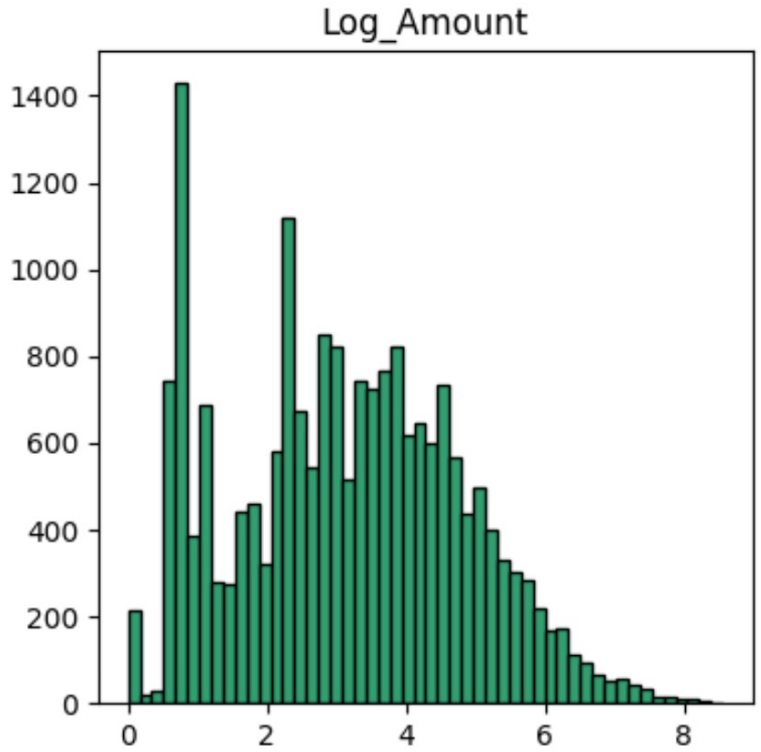
Log Amount

- Log 스케일링을 진행
- 데이터 분포가 정규 모형으로 변형
  - 꼬리 부분이 줄어들고 고르게 분포

- Log(0) 의 값을 피하기 위해 +1 값을 추가
  - Amount의 경우 1 정도의 크기가 큰 문제가 되지 않음
  - 이 값이 영향을 미치는 데이터인지 확인 필요!

- 단순 Amount 보다는 Log_Amount 값 활용

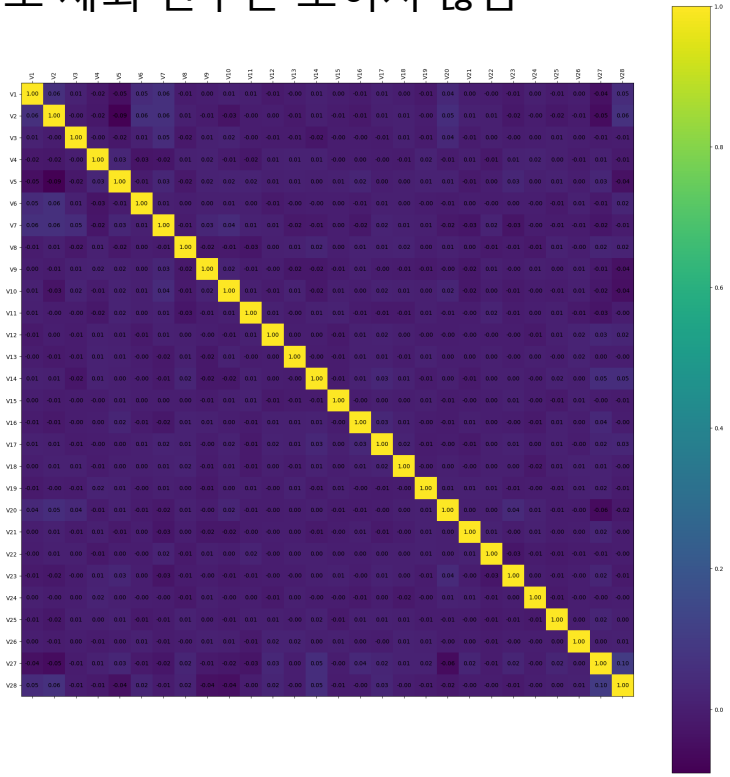
```
credit['Log_Amount'] = np.log(credit['Amount'] + 1)
```

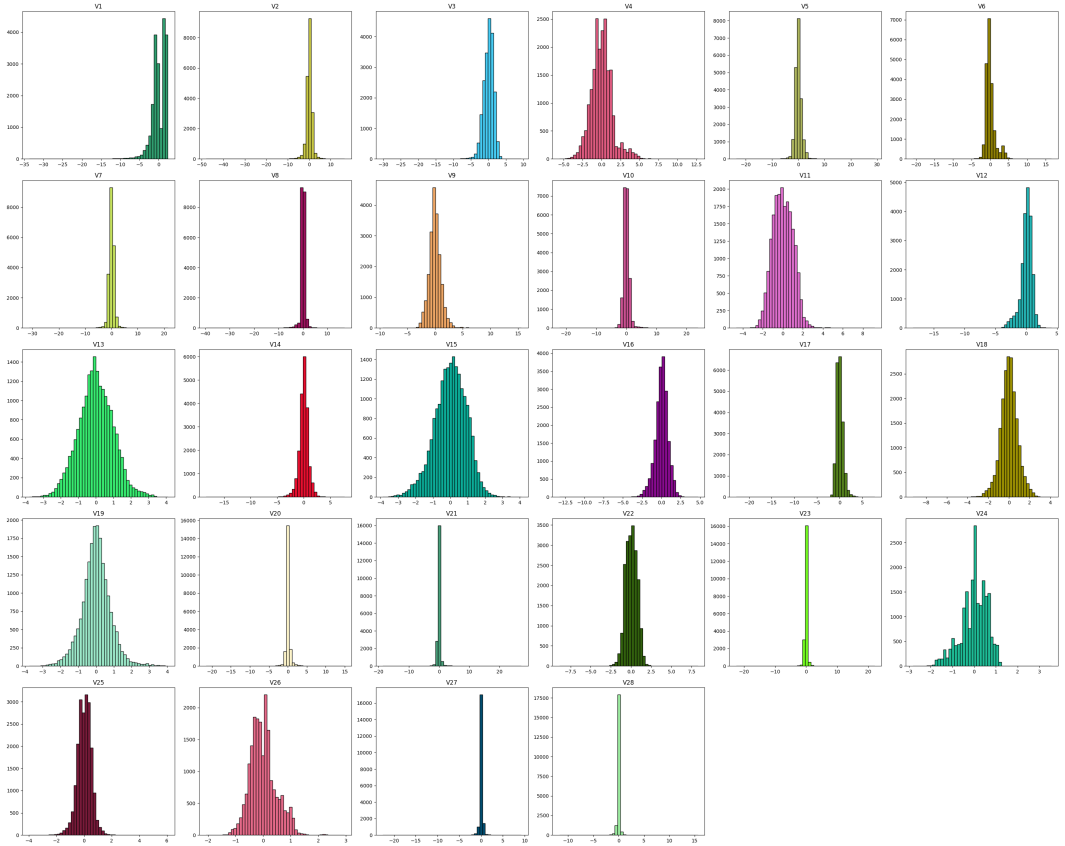


V1 ~ V28 데이터 시각화

- 전반적으로 정규 분포를 보임
  - 평균이 0에 근접함
  - 추가적인 스케일링도 필요하지 않음
- 또한, 큰 이상치도 보이지 않음

- 상관관계를 보았을 때에도 제외 변수는 보이지 않음





이상치 분석

강의 실습 코드 : 링크

Isolate Forest 모델 학습

- 입력 매개 변수인 contamination의 값은
- 원본 데이터에서의 이상치 비율을 알고 있으므로 해당 값을 사용

```
n_estimators = 100
max_samples = 'auto'
contamination = num_Fraud/(num_Normal+num_Fraud)

# Isolation Forest 생성 및 학습
from sklearn.ensemble import IsolationForest
IForest = IsolationForest(n_estimators=n_estimators,
                          max_samples=max_samples,
                          contamination=contamination,
                          random_state=seed)
IForest.fit(credit_combined)
```

평가

강의 실습 코드 : 링크

분류 문제로서의 평가

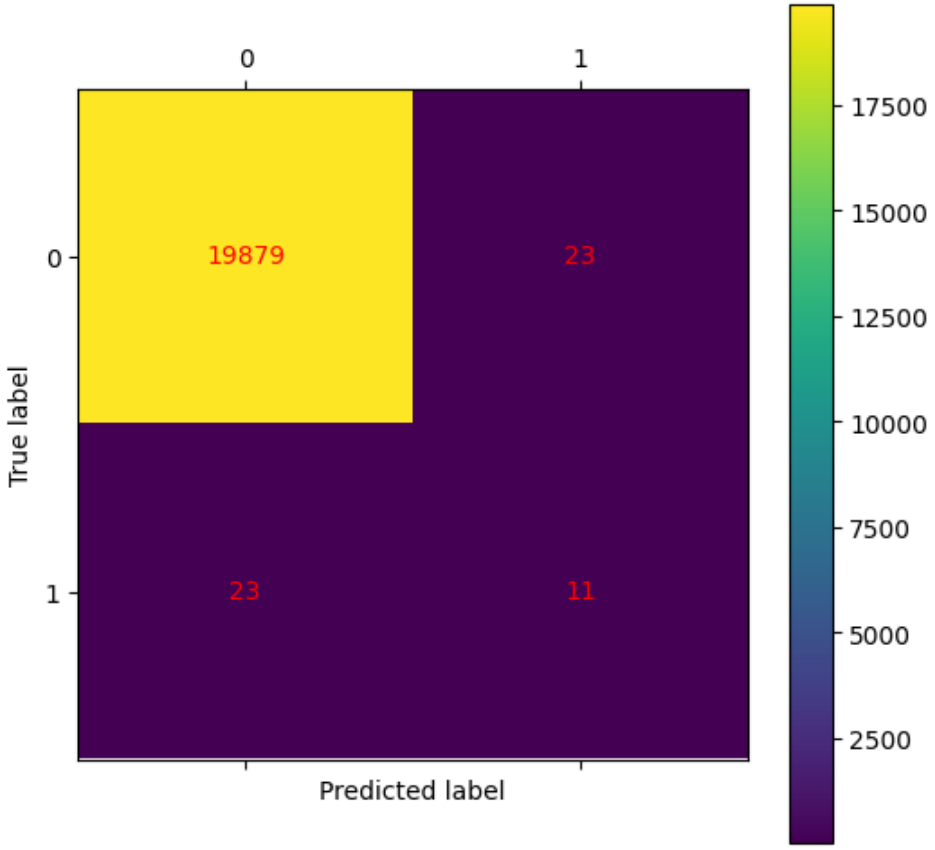
- 이상치 데이터의 레이블이 되어있으므로
- 양성 데이터를 예측하는 방식으로 평가를 진행

- 이상치 데이터는 전체 데이터에서 극 소수이므로
- 정밀도, 재현율, F1 값이 그렇게 좋지는 않음
- Confusion Matrix를 통해 결과 확인

```
y_true = credit['Class']

y_pred = IForest.predict(credit_combined)
y_pred = np.where(y_pred == 1, 0, 1)


accuracy = accuracy_score(y_true, y_pred) # 99.77 %
precision = precision_score(y_true, y_pred) # 32.35 %
recall = recall_score(y_true, y_pred) # 32.35 %
f1 = f1_score(y_true, y_pred) # 32.35 %
```



E.O.D