데이터 전처리 최종

클래스 불균형 문제

- 언더샘플링 + SMOTE 혼합 적용 -> 모델 학습에서 진행

결측치

- 포트 번호 관련: -1로 변경
- payload 평균 바이트: -1로 변경
- iat: 중앙값으로 변경
- duration: 중앙값으로 변경
- rate_fwd_pkts, rate_bwd_pkts: 중앙값으로 변경

로그 변환+스케일링

- duration
- pkt_count_fwd, pkt_count_bwd
- rate_fwd_pkts, rate_bwd_pkts
- rate_fwd_bytes, rate_bwd_bytes
- payload_fwd_init, payload_bwd_init
- tcp_win_fwd_init, tcp_win_bwd_init
- iat

결측치 존재 변수일 시에 중앙값으로 결측치 처리 후 로그 변환 적용

📍 train 데이터셋 내 변수의 결측치에 전체 중앙값 대신 'attack_type 별 중앙값'을 넣으려고 했지만, test 데이터셋에는 attack_type 변수가 존재하지 않고 train 데이터셋에 과적합 확률이 높아 모두 **전체 중앙값**으로 대체

불필요한 변수 삭제

- ID 변수 제거 / IP 관련 삭제
- port_src (송신 방향 포트번호는 예측에 필요없다고 판단)
- payload 변수 관련해서 fwd 삭제 (bwd와 동일하다고 판단)

범주화

- TCP flag 중 SYN 범주화 처리 > 클래스 3개로 범주화

- 0: SYN 패킷 없음 (정상)
- 1,2: 약간의 SYN 요청
- 3 이상: SYN Flooding 가능성 존재

- 포트 번호 범주화 처리

- 0~1024: 그대로 놔두기
- 1024 이후: 특징적인 포트번호 상위 5가지 확인하여 특정 + 나머지는 같이 범주화

파생변수 생성

- **tcp_psh_count** 파생변수

- PSH 의 유무를 설명하는 이진 변수 생성 / 기존 변수 삭제
  - 0: PUSH 없음
  - 1: PUSH 존재
- 공격 플래그 정보로 중요할 것이라 판단

- **초당 패킷 전송 수 관련 (rate_fwd_pkts, rate_bwd_pkts)**

📍
- ‘rate_fwd_pkts/rate_bwd_pkts’ 생성 / 기존 변수 삭제
- 송수신 간 트래픽 비율 의미

- **송수신 패킷 비율 (pkt_count_fwd, pkt_count_bwd) 파생변수**

📍
- ‘pkt_count_fwd/pkt_count_bwd’생성 / 기존 변수 삭제
- 송수신 패킷의 비율이 극단적일 수록 공격을 받았을 가능성이 높다고 판단

→ 비율 관련 변수는 추후 Feature importance를 확인한 후 제거 판단 수행

- **패킷 간 평균 시간 간격** 파생변수

📍
- **iat_avg_packets <=1.0m/s** 인 값의 이진 파생변수 생성
- 시간 패턴을 반영(공격 유형 별 특징 존재)
- 패킷 간 평균 시간 간격이 매우 짧을 수록 공격을 받았을 가능성이 높다고 판단

- payload의 결측치 관련 파생변수 생성

📍
- **payload_missing** 파생변수 생성 (결측치 아니면 0, 결측치면 1)
- 결측치 자체가 공격 신호일 확률 존재

LabelEncoding 진행

- protocol
- post_dst
- attack_type

추후 결과가 안 나오면 추가할 변수

- iat ~ : NA임을 나타내는 파생변수 추가

  → (결측치 자체가 유의미하다고 생각)

<F1 score을 높이기 위한 전략 생각>

1. 클래스 불균형 문제 해결 -> 언더샘플링 + SMOTE 혼합

- 공격 유형 중 Web_XSS, SSH_Brute_Force 같은 경우, 데이터가 굉장히 적음
- 이런 적은 공격 유형을 무시하게 되면 모델이 못 잡을 위험성 ↑
- SMOTE를 통해 소수 클래스 데이터를 늘리고 다수 클래스는 언더 샘플링하여 균형을 맞춤

2. 결측치 처리 + 결측치 정보 활용

- 포트 번호, payload 등의 결측은 단순 누락이 아닌 공격 신호일 가능성이 높음

  → 결측치 대체를 통해 이를 하나의 플래그로 반영

3. 로그 변환으로 이상치 완화 및 학습 안정화 수행

- pkt_count_fwd, duration 등의 변수 -> 분포가 매우 극단적인 것을 확인 가능
- 과도한 이상치를 완화하기 위해 로그 변환을 수행하여 모델이 균형 있게 학습하도록 함

4. TCP 플래그 및 패킷 비율 등의 파생 변수 활용

- TCP 플래그 (SYN, PSH 등)은 공격 유형 별 특징을 반영하고 있음

  → 송신/수신 패킷 비율은 공격자의 트래픽 패턴 차이를 보여주기도 함

5. 패킷 간 평균 시간 간격에 대한 이진 변수 생성(iat_avg_packets)

- 공격자의 경우 요청을 빠르거나 느리게 보내는 특징이 존재

  → iat_avg_packets <= 1.0 m/s의 경우 빠른 공격 시도 가능성이 높음 (특히 DoS, DDoS)

- 시간 패턴을 학습하여 공격과 정상 흐름 구분력을 향상 시킴

6. 지속시간 대비 패킷 수(트래픽 밀도) 파생변수 생성

- 단순 패킷 수, 지속 시간에 비해 초당 얼마나 한 번에 보내는 지에 대한 밀도 변수 필요성 반영

→ 공격 탐지에 중요한 신호가 될 것이라 판단

- 짧은 시간 내 수천 패킷을 보내는 경우 비정상일 경우가 높음 (DDos, Prot Scanning 등의 가능성 존재)

7. 불필요한 변수 제거 및 범주화

- ID, IP 등의 경우 예측이 의미가 없음 + 노이즈 데이터 생성 가능성↑ → 제거
- 포트 번호, TCP 플래그 등은 범주화를 통해 모델이 더욱 쉽게 패턴 인식을 수행하도록 함

  → 범주를 나누면 비정상 포트를 인식하는데 더욱 효과적일 수 있음