데이터 전처리 방향

1. 결측치 처리

📍 **결측치 존재 변수**

- 'ip_src'
- 'port_src'
- 'ip_dst'
- 'port_dst'
- 'duration'
- 'rate_fwd_pkts'
- 'rate_bwd_pkts'
- 'payload_fwd_mean'
- 'payload_bwd_mean'
- 'iat_avg_packets'

📍 **결측치 비율**

port_src	0.281190
iat_avg_packets	0.230019
ip_src	0.209351
port_dst	0.190599
payload_fwd_mean	0.148762
payload_bwd_mean	0.148762
rate_bwd_pkts	0.140595
ip_dst	0.108676
rate_fwd_pkts	0.097591
duration	0.089507

- **IP 주소 관련 변수 (ip_src, ip_dst)**

  → 결측치 'unknown'으로 대체

- **포트 번호 관련 변수 (port_src, port_dst)**

  → 결측치 **-1** 으로 대체

  📌
  - 포트 번호의 결측치를 0으로 대체하면 안 됨

    (포트 번호 내에서 0도 의미가 있는 수이기 때문)

- **duration(통신 시간)**
- **패킷 전송 속도 관련 변수 (rate_fwd_pkts, rate_bwd_pkts)**

  → 결측치 중앙값으로 모두 대체

  📌 이상치를 제외하고 대부분의 값이 몰려있음

- **페이로드 평균 바이트 관련 변수 (payload_fwd_mean, payload_bwd_mean)**

  → 결측치 중앙값으로 모두 대체

  📌 이상치를 제외하고 대부분의 값이 몰려있음

  - payload_fwd_mean의 값이 Null일 때, payload_bwd_mean의 값도 Null임을 확인

    → 페이로드 결측치 여부를 설명하는 **payload_missing 파생변수 생성**

- **iat_avg_packets(패킷 간 평균 시간 간격(초)**

  → 결측치 중앙값으로 모두 대체

  - 결측 자체가 공격의 힌트일 가능성 존재

    → 패킷 간 평균 시간 간격의 결측치 여부를 설명하는 **iat_avg_packets_missing 파생변수 생성**

2. 파생 변수 생성

결측치 관련 변수 생성

- **페이로드(Payload)의 결측치 여부**를 설명하는 파생변수 생성
- **패킷 간 평균 시간 간격(iat_avg_packets)의 결측치 여부**를 설명하는 파생 변수 생성

송수신 방향의 패킷 개수 비율 파생변수

- 관련 변수: pkt_count_fwd, pkt_count_bwd
- 패킷의 수가 한 쪽 방향으로만 집중될 시에 비정상일 가능성이 존재 → **송신 및 수신 방향의 패킷 개수 비율 비교** 필요

  → `pkt_count_fwd / pkt_count_bwd` 값으로 파생 변수 생성

  - **기존 변수 삭제**

패킷 간 평균 시간 간격 파생변수

- DoS, DDoS 공격에서 짧은 시간 내에 수 많은 패킷이 전송되므로 정상 트래픽보다 패킷 간 평균 시간 간격이 매우 작은 값을 가짐 → 즉, **매우 작은 값일 시에 공격 타입일 가능성이 높음**

  → **iat_avg_packets의 1.0m/s 이하의 값**을 파생변수로 생성

초당 패킷 전송 수 관련 파생변수

- 관련 변수: rate_fwd_pkts, rate_bwd_pkts

- **정상:** `rate_fwd_pkts` = `rate_bwd_pkts`
- **비정상:** `rate_fwd_pkts` > `rate_bwd_pkts`

  → 요청은 많고 응답이 없음 (서버 다운 및 포트 닫힘 등)
- **정보 유출/악성 트래픽:**

  `rate_fwd_pkts` < `rate_bwd_pkts`

  → 서버가 오히려 많은 데이터를 돌려줌

→ `rate_fwd_pkts / rate_bwd_pkts == 1` 또는 `rate_fwd_pkt / rate_bwd_pkts ≠ 1` 의 여부로 정상 및 비정상을 확인할 수 있는 파생변수 생성

- **기존 변수는 삭제**

---

3. 불필요 변수 삭제

ID 변수 삭제

- 예측에 불필요한 정보

IP 주소 관련 변수 삭제

- 공격자에 대한 정보를 알아내는 것이 아니라 사이버 공격 유형을 예측하는 것이 목표이므로 불필요한 정보

  → **ip_scr, ip_dst 삭제**

---

4. 비대칭 분포 변수 → 로그 변환

왜도 확인 후 로그 변환

→ 크게 개선이 되지 않은 변수는 제외

- **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**

---

5. 범주화

포트 번호 변수 범주화

- 관련 변수: port_src, port_dst

📌

이름	포트 번호 범위	설명
0 ~ 1023	잘 알려진 포트(Well-known port)	시스템 사용 번호
1024 ~ 49151	등록된 포트(Registered port)	특정 프로토콜이나 어플리케이션에서 사용하는 번호
49152 ~ 65535	동적/사설 포트 (Dynamic/Private port)	어플리케이션에서 임시로 사용하는 포트 (랜덤 할당)

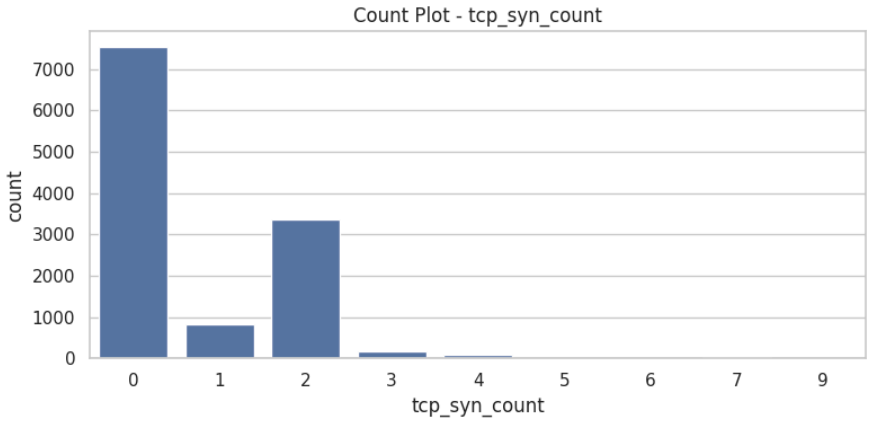
- 포트 번호의 범주가 너무 많음

  → **-1, 0~1023, 1024~49151, 49152~65535 4개로 범주화**

  → **기존 변수 삭제?**

TCP SYN 패킷 변수 범주화

- 관련 변수: tcp_syn_count



- **0, 1~2, 3 이상** → 3개의 범주로 범주화 진행

📌

- 0 → SYN 패킷 없음, 정상
- 1, 2 → 약간의 SYN 요청 (정상 또는 정찰 시도)
- 3 ~ → SYN Flooding 가능성 존재

→ **기존 변수 삭제?**

TCP PSH 패킷 변수 이진화

- 관련 변수: tcp_psh_count
- 클래스를 확인해봤을 때, 0인 값이 8505개로 전체 값에서 매우 높은 비율을 보임

  → `tcp_psh_count == 0` : PUSH가 없음

  → **0**과 **0이 아닌 값** 두 범주로 과적합 방지를 위해 이진화 진행

  → 기존 변수 삭제

  - **PUSH의 유무를 설명**하는 변수

---

6. Label Encoding 및 One-Hot Encoding

Protocol 변수 Label Encoding

- TCP, UDP 두 개의 범주만 존재

TCP RST 패킷 변수 Label Encoding

- 관련 변수: tcp_rst_count

  → RST 패킷 수는 세션 강제 종료를 시도한 횟수로, 현재 데이터의 범주는 0과 1로 이루어져있음

  - `tcp_rst_count == 1` : 강제 종료 시도가 존재

  → 따라서 강제 종료 시도 여부를 의미하는 범주형 변수이므로 Label Encoding

---

5)에서 범주화한 변수 Label Encoding 및 One-Hot Encoding

타겟 변수 One-Hot Encoding

1. Benign: 8791개로 정상이 전체 데이터의 대부분을 차지

   → Benign vs Attack으로 이진화

2. 샘플 수가 상대적으로 적은 범주들 통합