1. **Dirty data** : Missing, +, %, 실수 값 (정수가 아닌 맞지 않는 값) 넣기

넣은 columns:

https\_token: 55개의 더러운 데이터

length\_url: 34개의 더러운 데이터

phish\_hints: 46개의 더러운 데이터

nb\_at: 20개의 더러운 데이터

page\_rank: 43개의 더러운 데이터

이미 존재하는 dirty column:

domain\_age -> 음수값 1837개 존재

domain\_registration\_length -> 음수값 46개 존재

1. **Data Cleaning**

* 더러운 데이터가 삽입된 5개 컬럼(https\_token, length\_url, phish\_hints, nb\_at, page\_rank)에 대해 유효하지 않은 행 제거
* domain\_registration\_length가 음수인 행 제거
* status 컬럼을 숫자형으로 변환 (legitimate → 0, phishing → 1)
* url 컬럼 제거 (사용 불가능한 데이터)

1. **Feature Creation**

* domain\_age\_missing: domain\_age < 0 일 때 1, 아니면 0
* domain\_age\_created: domain\_age > 0 일 때 원본 값, 아니면 0
* 기존 domain\_age 컬럼 삭제

1. **Feature Selection**

* RandomForest Importance와 Mutual Information gain 두 가지 방법 사용
* 각 방법에서 상위 20개 특성 선택
* 두 방법에서 공통으로 선택된 특성들만 최종 선택 (16개)
* 선택된 특성들을 시각화하여 비교

1. **Scaling**

* Log scaling: nb\_hyperlinks, web\_traffic, longest\_word\_path, length\_url, ratio\_extRedirection, longest\_words\_raw, avg\_word\_path
* Min-Max Scaler: safe\_anchor
* RobustScaler: nb\_www, domain\_age\_created
* Z-score scaling: page\_rank
* QuantileTransformer: links\_in\_tags, ratio\_digits\_url, ratio\_extHyperlinks, ratio\_intHyperlinks
* Secondary scaling: 일부 컬럼(Log scaler 적용 후 평균이나 표준편차가 큰 컬럼) 에 대해 추가적인 Z-score 표준화 적용

1. **PCA (Feature Reduction)**

* 누적 설명 분산이 95% 이상이 되는 최소 컴포넌트 수 결정
* 누적 설명 분산 그래프 시각화
* 첫 번째와 두 번째 주성분에 대한 산점도(scree plot) 시각화
* 최종적으로 10개의 주성분 선택하여 결과 저장

----------------------- 코드 설명 ---------------------

1. **데이터 준비 및 더러운 데이터 생성 (create\_dirtydata.py)**

• 원본 데이터: dataset\_phishing.csv

• 결과: dataset\_phishing\_dirty.csv

1. **데이터 정제** (data\_cleaning.py)

* 입력: dataset\_phishing\_dirty.csv
* 처리:
* 더러운 데이터가 있는 5개 컬럼의 유효하지 않은 행 제거
* domain\_registration\_length 음수값 제거
* status 컬럼 변환 (legitimate → 0, phishing → 1)
* url 컬럼 제거
* 결과: dataset\_phishing\_clean.csv

1. **특성 생성** (feature\_creation.py)

* 입력: dataset\_phishing\_clean.csv
* 생성된 특성:
* domain\_age\_missing: domain\_age < 0 일 때 1, 아니면 0
* domain\_age\_created: domain\_age > 0 일 때 원본 값, 아니면 0
* 결과: dataset\_phishing\_feature\_creation.csv

1. **특성 선택** (feature\_selection.py)

* 입력: dataset\_phishing\_feature\_creation.csv
* 방법:
* RandomForest Importance
* Mutual Information gain
* 두 방법에서 공통으로 선택된 16개 특성 선택
* 시각화: 상위 20개 특성 비교 그래프
* 결과: selected\_features\_intersection.csv

1. **스케일링** (scaling.py)

* 입력: selected\_features\_intersection.csv
* 적용된 스케일링:
* Log scaling: 7개 컬럼
* Min-Max Scaler: safe\_anchor
* RobustScaler: nb\_www, domain\_age\_created
* Z-score scaling: page\_rank
* QuantileTransformer: 4개 컬럼
* Secondary scaling: 5개 컬럼에 추가 Z-score 표준화
* 결과: final\_matrix.csv

1. **차원 축소** (pca.py)

* 입력: final\_matrix.csv
* 처리:
* 누적 설명 분산 95% 기준으로 컴포넌트 수 결정
* 시각화: 누적 설명 분산 그래프, 2D 산점도
* 최종 10개 주성분 선택
* 결과: pca\_result.csv

1. **EDA (탐색적 데이터 분석)**

eda\_for\_scaling.py: 스케일링을 위한 EDA

다양한 eda\_\*.png 파일들: 각 특성별 시각화

**[참고]**

* 컬럼 의미 및 순위.txt: 데이터셋의 컬럼 설명과 중요도 순위