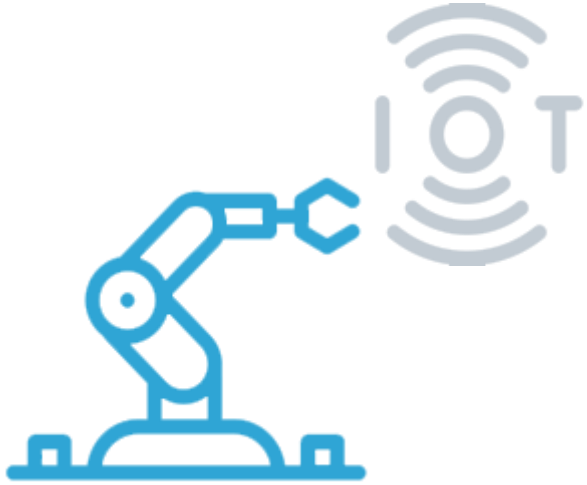
변종 악성코드에 대한 대비를 위한 최적의 악성코드 분류 모델 생성

IoT Malware Prediction

TABA 2조
곽성신, 곽보선, 박민영, 박서영, 이지영

프로젝트 배경

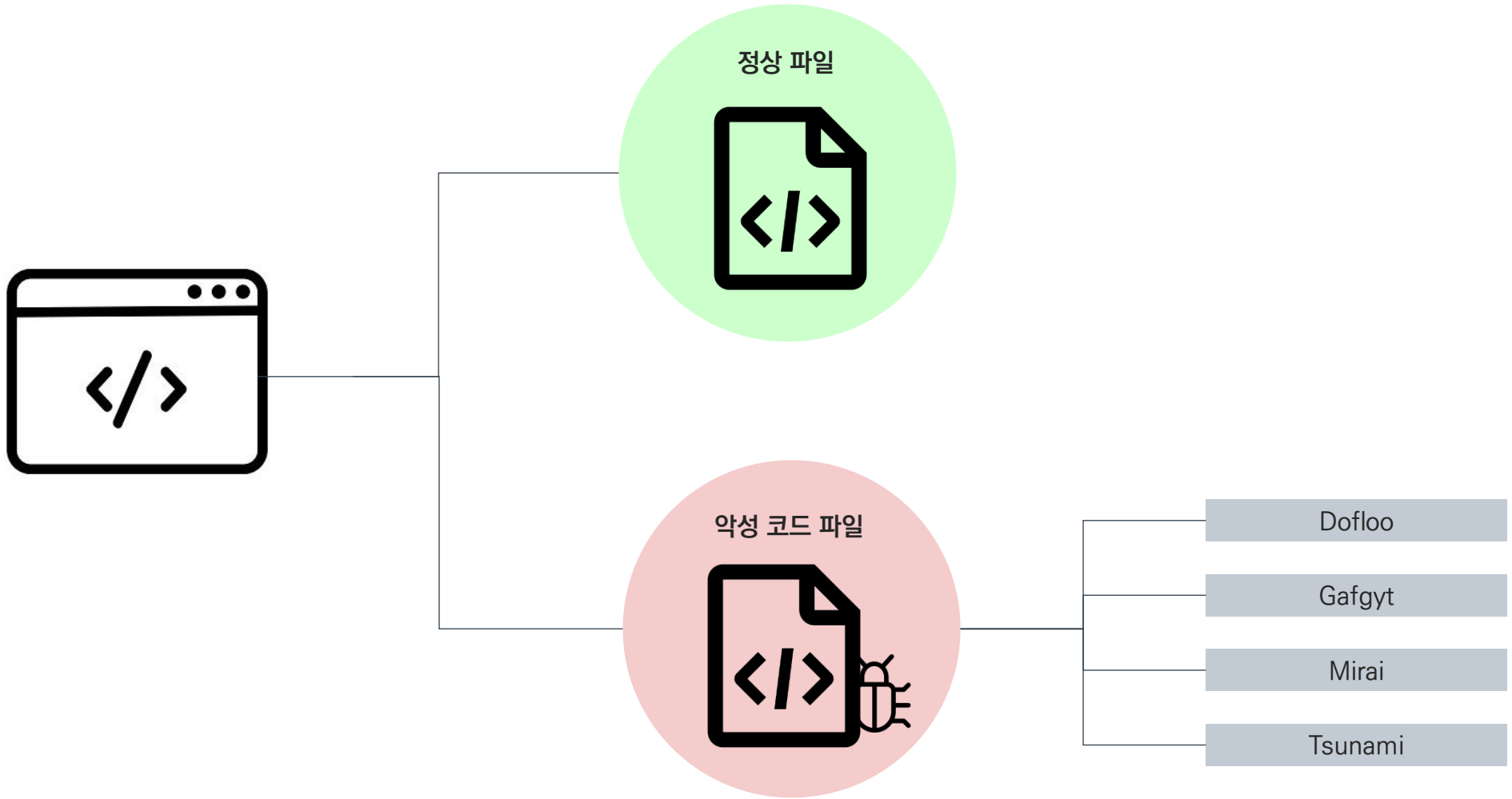
사물인터넷 보안 취약



약점을 노린 악성 코드

IoT 보안 위협 경계 강화 필요

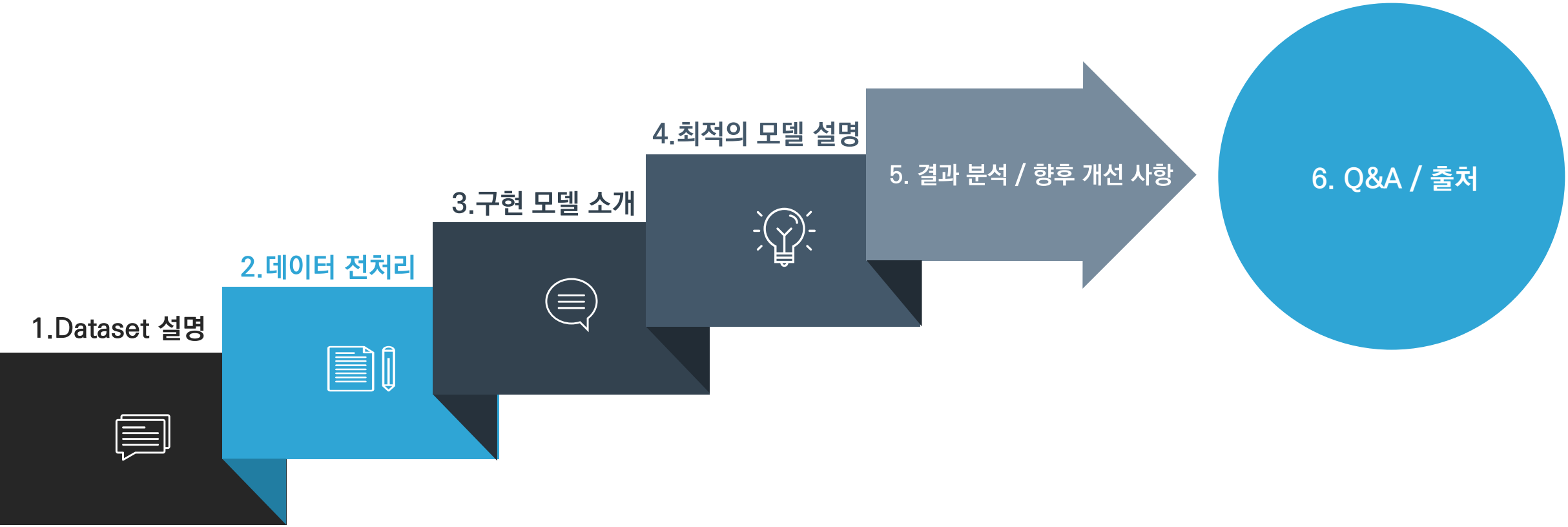
프로젝트 목표



역할 분담 및 프로젝트 진행 과정

- 데이터 시각화 : 이지영
- 데이터 전처리 : 박민영, 박서영
- 모델 학습 및 평가 : 곽보선, 곽성신, 박민영, 박서영

Contents



기본 데이터셋

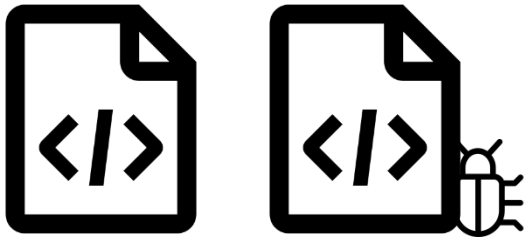
Dataset 설명

기본 원리

1

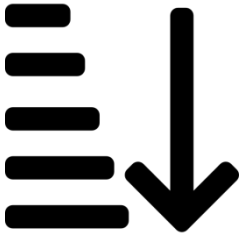
정상 파일

악성 코드 파일



Objdump

디스어셈블리 코드 생성



명령 코드 시퀀스 추출

ARM 기반
Instruction Set
Architecture

- Dofloo
- Gafgyt
- Mirai
- Tsunami
- Benign

기능 별로 카테고리 분류

Sliding Window 기법

엔트로피 히스토그램 계산

&

특징 추출

기본 Dataset

생성된 Malware 데이터셋은 121개의 특징을 가지고 있으며 총 5종

전체 데이터셋 구조

	0 ~ 120 칼럼	label
0	(시퀀스 값)	Mirai
1	…	Gafgyt
…	…	…

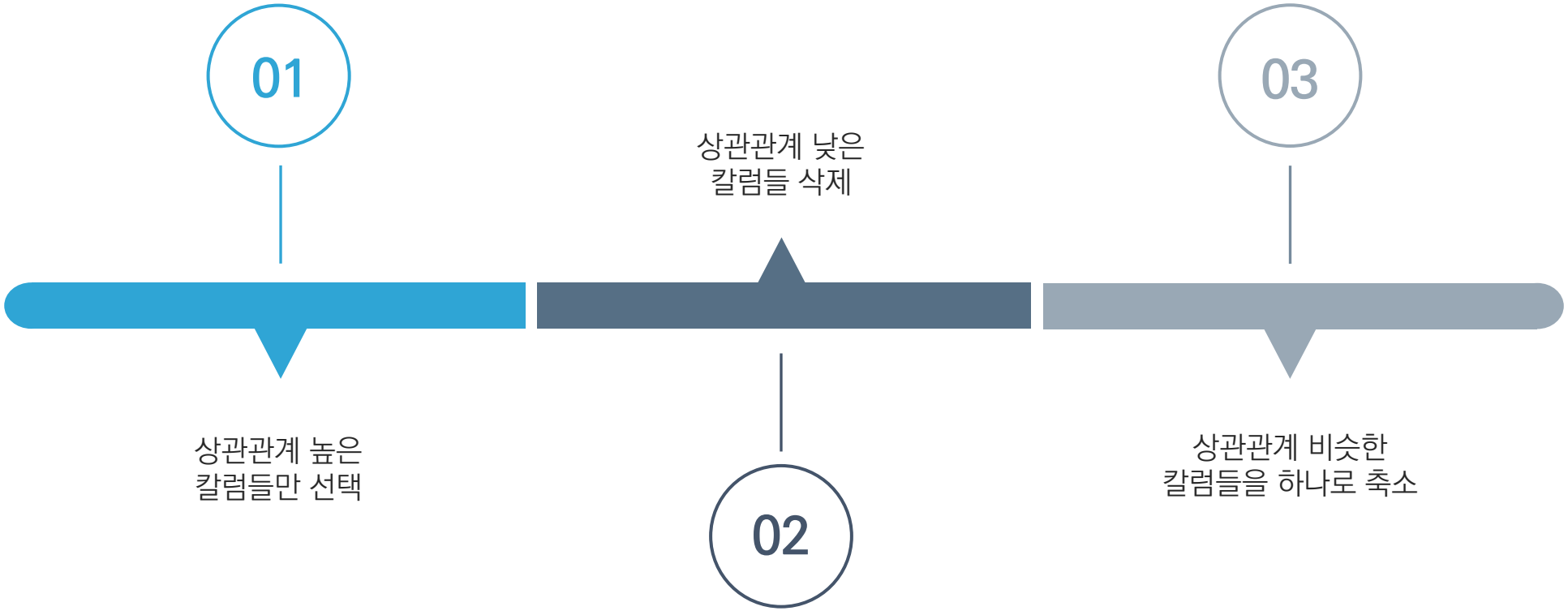
코드 분류 비율

코드 분류	개수	비율	
Dofloo	844	4.2%	악성 코드
Gafgyt	8,582	43.2%	
Mirai	7,404	37.2%	
Tsunami	467	2.3%	
Benign	2,591	13.0%	정상 코드
전체	19,888	100.0%	

데이터 전처리를 한 데이터셋

데이터 전처리

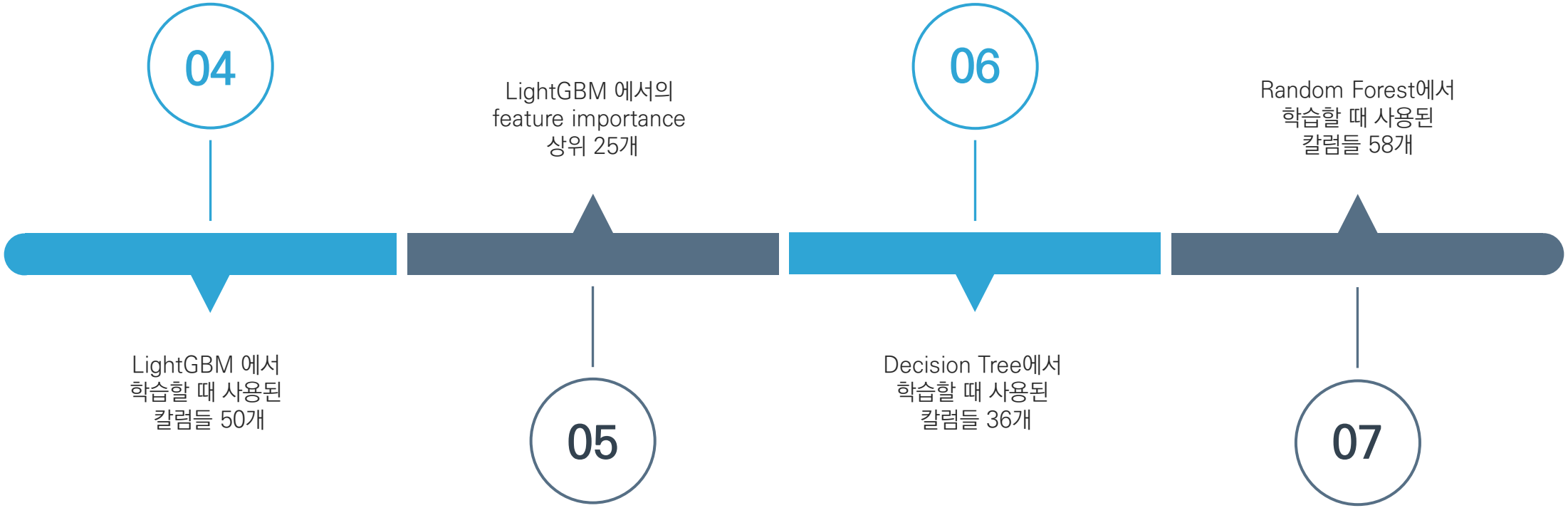
상관관계



데이터 전처리를 한 데이터셋

데이터 전처리

feature importance



데이터 전처리를 한 데이터셋

데이터 전처리

기타

08

간단 전처리
(0만 있는 칼럼,
uuid 칼럼 삭제 등
기타 불필요한 것 삭제)

09

0이 대부분인 칼럼 중,
큰 분포가 있지 않은
칼럼 제거

10

0이 아닌 값들이
모두 benign인 경우의
칼럼 제거

11

데이터셋에서
상위 1% 값을 0으로 만들고
의미 없어진 칼럼 제거

12

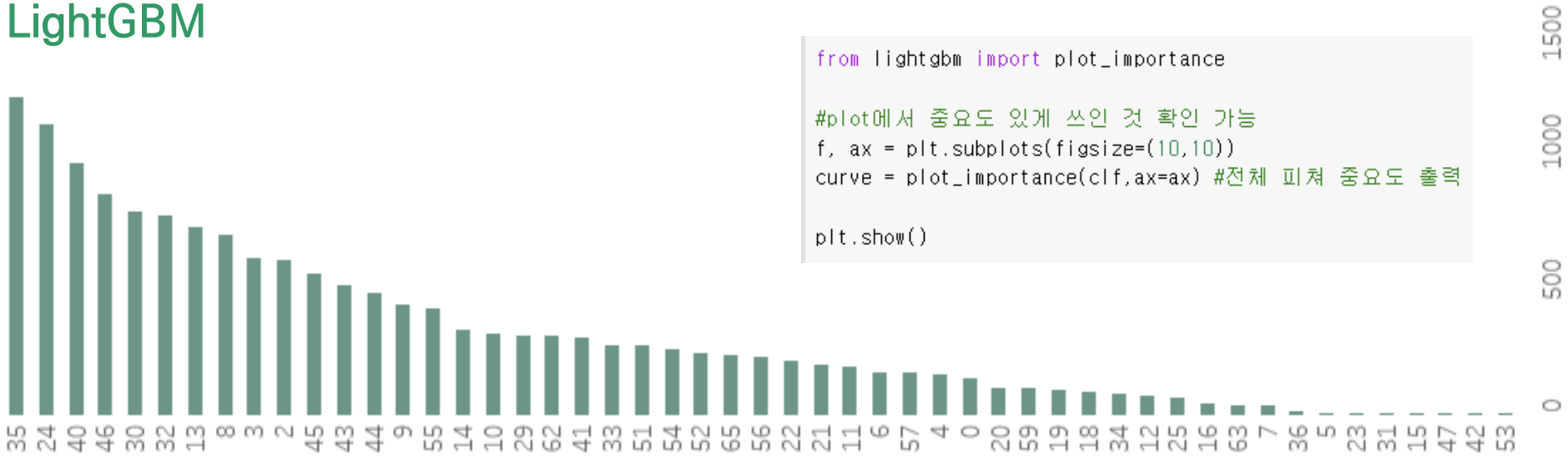
데이터셋에서
상위 25% 값을
적당히 큰 수로 설정

데이터 전처리

최적의 모델에서 사용한 데이터셋 – LightGBM

LightGBM

```
from lightgbm import plot_importance

#plot에서 중요도 있게 쓰인 것 확인 가능
f, ax = plt.subplots(figsize=(10,10))
curve = plot_importance(clf,ax=ax) #전체 피쳐 중요도 출력

plt.show()
```



데이터 전처리를 한 데이터셋

데이터 전처리

최적의 모델에서 사용한 데이터셋 – LightGBM

그래프에서
칼럼명 추출

```
# feature importance에서 나온 중요 columns만 뽑기
# 1. 전체 column 뽑기 / 2. 절반만 가져오기

#중요도가 0이 아닌 feature(column)의 개수
len_yticks = len(curve.get_yticklabels())

#각 중요 column을 담을 리스트 생성
importanceList50 = list()
importanceList25 = list()

#1. 전체 column 뽑기
  #사용한 모든 column을 담은 리스트
for i in range(0, len_yticks):
    #column명 리스트에 추가
    importanceList50.append(curve.get_yticklabels()[i].get_text())

#2. 절반만 가져오기
  #사용한 column의 상위 25개만 담은 리스트
    #순서대로 정렬이 되어있어 절반부터 시작해도 됨
    #중요 column을 절반으로 줄여도 문제가 있을지 궁금해서 추가함
for i in range(int(len_yticks/2), len_yticks):
    #column명 리스트에 추가
    importanceList25.append(curve.get_yticklabels()[i].get_text())

#잘 생성됐는지 확인
print(len(importanceList50))
print(len(importanceList25))
```

값 정렬
label 칼럼 추가
데이터프레임화

```
#정렬 후, 데이터프레임화
importanceList50.sort()
importanceList25.sort()
print(importanceList50)
print(importanceList25)

importanceList50.append('label')
importanceList25.append('label')

importanceList50_df = data[list(map(str,importanceList50))]
importanceList25_df = data[list(map(str,importanceList25))]
```

```
['0', '10', '11', '12', '13', '14', '15', '16', '18', '19', '2', '2
['10', '13', '14', '2', '22', '24', '29', '3', '30', '32', '33', '3
```

	0	10	11	12	13	14	15
0	0.000000	0.509727	0.000000	0.000000	2.208603	0.187500	0.0
1	0.000000	0.171875	0.000000	0.000000	3.160784	0.694216	0.0
2	0.000000	0.000000	0.141986	0.000000	1.409876	0.268901	0.0
3	0.000000	0.530216	0.128705	0.000000	2.507974	0.187500	0.0
4	0.000000	0.470665	0.093750	0.000000	3.611514	0.694216	0.0
...	...	...	...	...	...	...	...

19888 rows × 51 columns

데이터프레임 확인

사용한 모델

	Decision Tree	Random Forest	KNN
특징	**데이터의 패턴**을 찾아내고 트리 모양으로 **조건에 따라 데이터를 분할**하며 예측	**여러 데이터 샘플을 추출**하고 **각 샘플마다 Decision Tree**를 만들어 결과를 종합	데이터들 사이의 **거리를 측정**하여 **어떤 데이터가 가까운지 파악**하고 새로운 데이터를 분류
장점	단순한 규칙, 특성 별 중요도 파악 용이	Decision Tree 한계점 보완, 결측치에도 높은 정확도를 보임	간단한 알고리즘, 수치형 데이터 분류 작업에 용이
단점	과적합 발생하기 쉬움, 모델이 복잡할 경우 성능 저하	많은 메모리 사용, 속도 느림	데이터 양이 많은 경우 계산 속도 저하
기본 데이터 셋	0.951100	0.963545	0.974041
효과적인 데이터셋	상관관계 낮은 칼럼 삭제한 데이터셋	DT에서 중요도 높게 나온 칼럼 36개를 추출한 데이터셋	LightGBM 돌렸을 때 중요도 높게 나온 칼럼 25개를 추출한 데이터셋
최고 정확도	0.955123	0.964865	0.976241

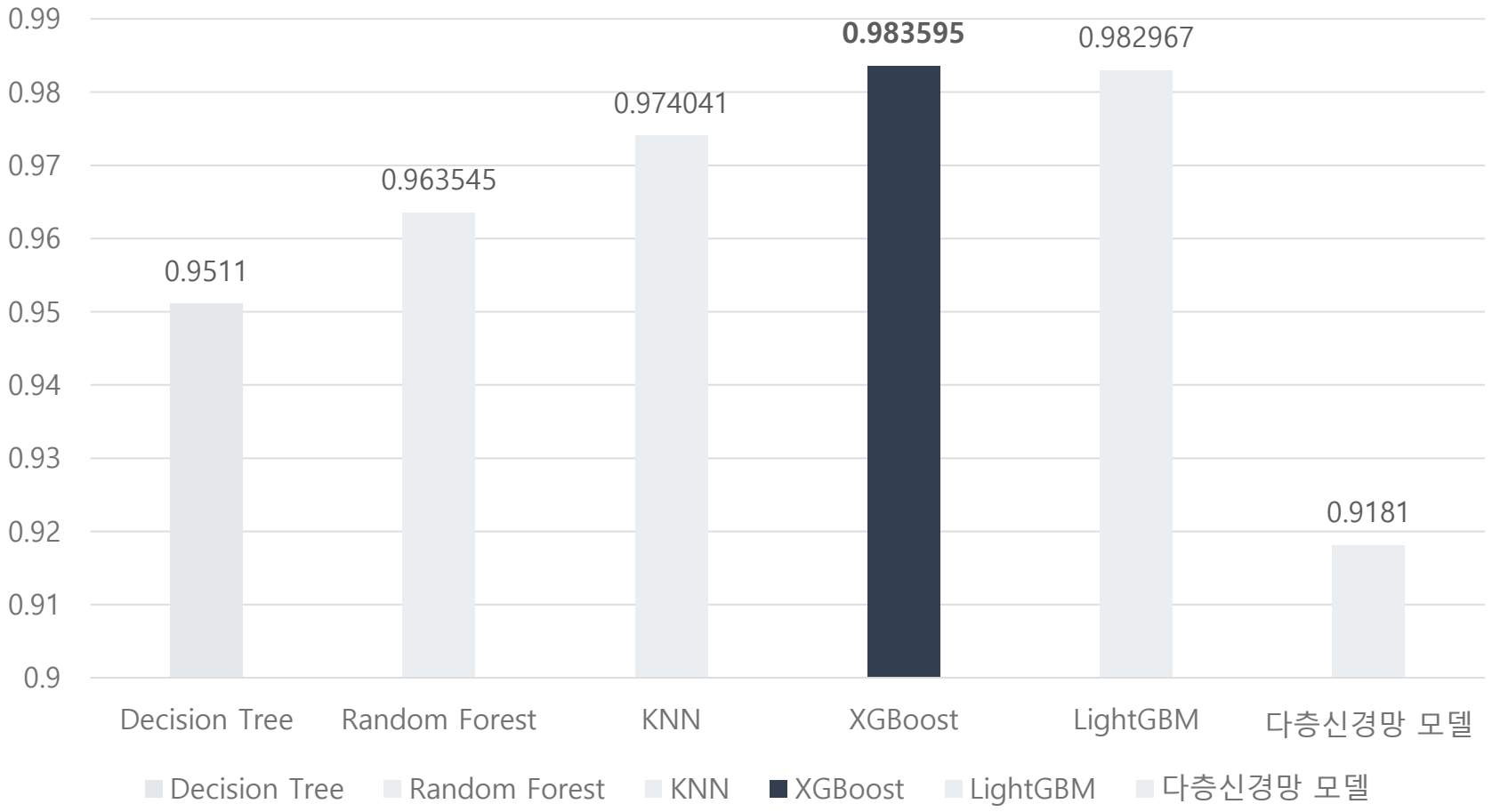
사용한 모델

	XGBoost	LightGBM
특징	이전 모형에서의 실제값과 예측값의 오차를 훈련데이터로 투입하고 **gradient를 이용하여 오류를 보완하는 방식을 병렬적으로 처리**	최대 손실 값을 가지는 **리프 노드를 지속적으로 분할하여 비대칭적인 트리를 생성**하고 예측 오류 손실을 최소화
장점	병렬 처리로 학습/분류 속도 빠름, 과적합 방지가 잘 되어 있음	XGBoost보다 학습 시간/메모리 사용량 적음
단점	데이터 개수가 적을 경우 과적합 가능성이 있음	데이터 개수가 적을 경우 과적합 발생하기 쉬움
기본 데이터 셋	0.983595	0.982967
효과적인 데이터셋	LightGBM에서 중요도 높게 나온 칼럼 50개를 추출한 데이터셋	RF에서 중요도 높게 나온 칼럼 58개를 추출한 데이터셋
최고 정확도	0.983721	0.983030

결과치가 낮게 나온 모델 및 구현에 실패한 모델

	다층신경망 모델	K-Mean	SVM
특징	**여러 개의 퍼셉트론 뉴런을 여러 층으로 쌓아** 입력층과 출력층 사이에 하나 이상의 은닉층을 가지고 있는 신경망	반복적인 연산을 거듭하여 **점점 비슷한 데이터들끼리 모여서 군집을 만드는 방법**	주어진 데이터가 **어느 카테고리에 속할지 분류를 위한 기준을 정의**하는 선형 분류 모델
장점	대량의 데이터의 정보를 잡아내고 매우 복잡한 모델 생성 가능	많은 양의 데이터 빠르게 분류 가능, 데이터 사전 정보 없이 분석 가능	분류문제나 예측문제 동시에 사용 가능
단점	같은 의미를 가진 동질의 데이터에서 잘 작동함	이상치의 영향을 많이 받음	여러 테스트를 통해 최적화 모형 구축 가능, 모형 구축 시간이 오래 걸림
학습 환경	기본 데이터 셋	기본 데이터셋 - 군집 2개일 때	-
최고 정확도	0.9181	0.40391	-

기본 데이터셋에 대한 모델 별 정확도 비교



프로젝트 목표와 부합하는 알고리즘

- 해당 코드가 정상 코드인지 악성 코드인지 구별
- 최고의 정확도

XGBoost

파라미터 설정

과적합 방지를 위한 파라미터

- max_depth
- eta
- subsample
- colsample_bytree
- n_estimators

```
params = {
    # Parameters that we are going to tune.
    'max_depth':[4,6],
    'eta':[0.01,0.1,0.2],
    'subsample': [0.5,1],
    'colsample_bytree': [0.5,1],
}
```

```
xgb = XGBClassifier(n_estimators=500)
```

GridSearchCV를 사용하여 최적의 모델을 위한 파라미터 추출

```
from sklearn.model_selection import GridSearchCV
gs = GridSearchCV(xgb, params, cv=5)
gs.fit(x_train, y_train)
print("Best Parameters : ", gs.best_params_)
```

최적의 모델

XGBoost

최적의 파라미터 조합으로 적용

```
from sklearn.model_selection import GridSearchCV
gs = GridSearchCV(xgb, params, cv=5)
gs.fit(x_train, y_train)
print("Best Parameters : ", gs.best_params_)
```

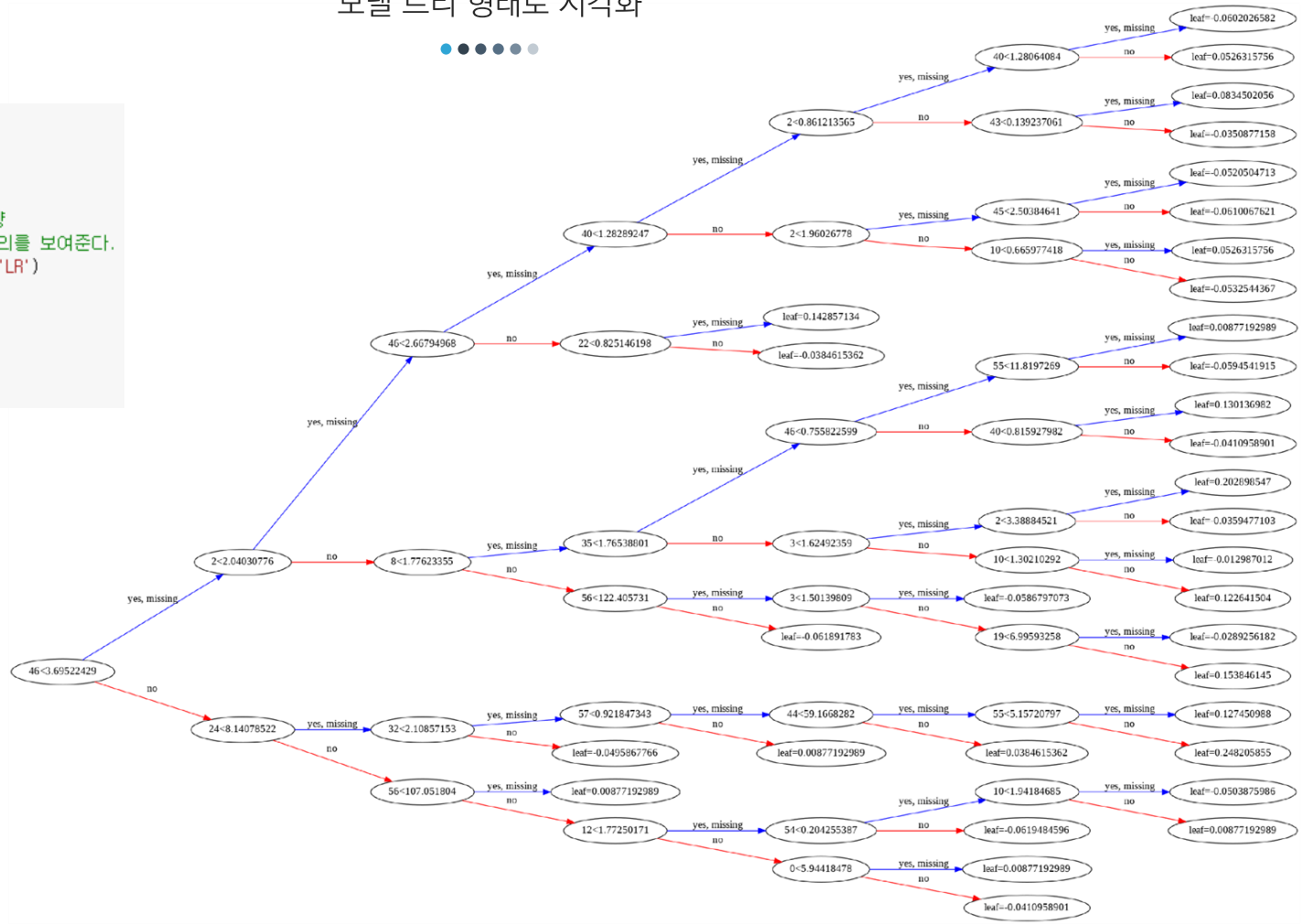
```
Best Parameters :  {'colsample_bytree': 1, 'eta': 0.1, 'max_depth': 6, 'subsample': 0.5}
```

```
xgb_clf = XGBClassifier(n_estimators=500,
                        max_depth= 6,
                        colsample_bytree=1.0,
                        eta=0.1,
                        subsample=0.5
                       )

xgb_clf.fit(x_train,y_train)
```

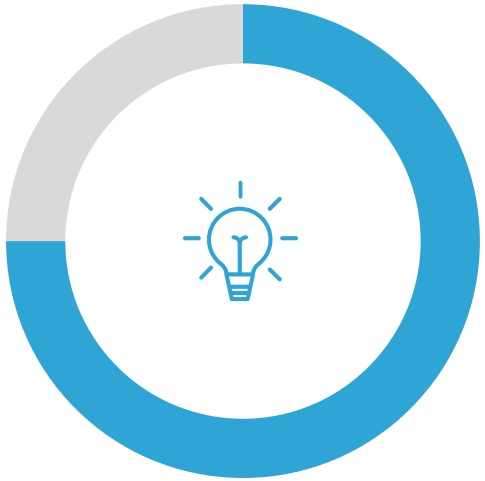
XGBoost

모델 트리 형태로 시각화


```python
import xgboost as xgb
import matplotlib.pyplot as plt

# num_trees : 그림을 여러개 그릴시 그림 번호
# rankdir : 트리의 방향, 디폴트는 위아래 방향
# rankdir="LR" : 왼쪽에서 오른쪽 방향으로 트리를 보여준다.
xgb.plot_tree(xgb_clf, num_trees=0, rankdir='LR')

fig = plt.gcf()
fig.set_size_inches(150, 100)

plt.show()
```



결과 분석



결과1

모든 모델이 정확도 95% 이상

결과2

정확도 비교

Decision Tree 〈 Random Forest 〈 KNN 〈 LightGBM 〈 XGBoost

향후 방향성

학습 소요 시간 등 다양한 방면을 종합해서 최적의 모델 개선 예정

향후 개선 과제

이번에 시도하지 못했지만 진행하면 좋은 결과를 가져올 것 같은 체크리스트 생성

✓ 데이터 전처리를 하는 부분을 컬럼 별로 큰 범위의 값 모두를 조정하는 방향으로 진행한다.

데이터셋 전처리에서 각 컬럼 별로 상위 25% 데이터들을 적당히 큰 수로 조정하는 과정을 진행할 때,

나머지 75%의 데이터들 중에도 값이 너무 큰 값들이 있었다.

✓ 인공지능 모델이 제대로 학습된 것인지 확인하는 더 많은 방법을 찾고 다양한 측면에서 인공지능을 평가한다.

✓ K_Means을 사용해볼 때 단순히 기본 데이터 셋을 적용만 시켜볼 것이 아니라 이상치를 제거하는 등의 시간을 조금 더 들여 높은 정확도를 만들어 본다.


Q&A

출처

논문

BAEK, Byunghyun, et al. **Histogram Entropy Representation and Prototype Based Machine Learning Approach for Malware Family Classification**. IEEE Access, 2021, 9: 152098-152114.

MUN, Sunghyun, et al. **Opcode category sequence feature and machine learning for analyzing IoT malware**. In: Proceedings of the Korea Information Processing Society Conference. Korea Information Processing Society, 2021. p. 914-917.

LEE, Hyunjong; EUH, Seongyul; HWANG, Doosung. **API Feature Based Ensemble Model for Malware Family Classification**. Journal of the Korea Institute of Information Security & Cryptology, 2019, 29.3: 531-539.

LEE, Hyunjong; EUH, Seongyul; HWANG, Doosung**. Distributed Processing System Design and Implementation for Feature Extraction from Large-Scale Malicious Code.** KIPS Transactions on Computer and Communication Systems, 2019, 8.2: 35-40.

공식 문서

xgboost parameter, xgboost, https://xgboost.readthedocs.io/en/stable/parameter.html

axes, bbox, text, https://matplotlib.org/stable/api

LGBM parameter, https://lightgbm.readthedocs.io/en/latest/Parameters.html

출처

https://school.coding-x.com/course (전체적인 인공지능 모델)

https://goldenrabbit.co.kr/2022/07/14/%ED%99%95%EC%8B%A4%ED%9E%88-%EC%95%8C%EC%95%84%EB%91%90%EB%A9%B4-%EB%A7%8C%EC%82%AC%EA%B0%80-%ED%8E%B8%ED%95%B4%EC%A7%80%EB%8A%94-%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-10%EA%B0%80%EC%A7%80-%EC%95%8C/ (전체적인 인공지능 모델)

https://injo.tistory.com/48 (LGBM feature importance 시각화)

https://runingcoding.tistory.com/33 (LGBM)

https://romg2.github.io/mlguide/04_%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-%EC%99%84%EB%B2%BD%EA%B0%80%EC%9D%B4%EB%93%9C-04.-%EB%B6%84%EB%A5%98-LightGBM/ (LGBM 파라미터와 디폴트 파라미터)

https://nurilee.com/2020/04/03/lightgbm-definition-parameter-tuning/ (LGBM 파라미터)

https://wikidocs.net/14605 (그래프의 객체 정보 추출)

https://inuplace.tistory.com/570 (RF 파라미터)

https://todayisbetterthanyesterday.tistory.com/51 (RF)

https://www.kaggle.com/code/prashant111/random-forest-classifier-feature-importance (RF feature importance)

https://m.blog.naver.com/dotorimj2/222178563106 (PCA)

https://seong6496.tistory.com/136 (파이썬 특정확장자 파일 찾기)

https://velog.io/@ohxhxs/%ED%8C%8C%EC%9D%B4%EC%8D%AC-%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-%EA%B5%90%EC%B0%A8%EA%B2%80%EC%A6%9D-KFold-StratifiedKFold-crossvalscoreGridSearchCV (교차검증 - KFold, GridSearchCV)

https://karupro.tistory.com/79?category=955612 (교차검증)

https://hleecaster.com/ml-svm-concept/ (svm 모델)

velog.io/@rosesua318/12장-다층-인공-신경망을-밑바닥부터-구현(딥러닝 모델)

velog.io/@raffier/머신러닝-데이터모델 (데이터 모델 분류)

https://seominseok4834.github.io/machine%20learning/4.classification/#xgboostextra-gradient-boost (classification)

출처

https://seominseok4834.github.io/machine%20learning/4.classification/#xgboostextra-gradient-boost (classification)

https://www.kaggle.com/code/lifesailor/xgboost (xgboost)

https://ko.wikipedia.org/wiki/서포트_벡터_머신(서포트 벡터 머신)

https://velog.io/@hhhs101/머신러닝-KNN (KNN)

https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=winddori2002&logNo=221659080425 (DT, 교차검증)

https://injo.tistory.com/15 (DT 시각화)

http://www.gisdeveloper.co.kr/?p=9932 (모델평가기준 – Confusion Matrix)

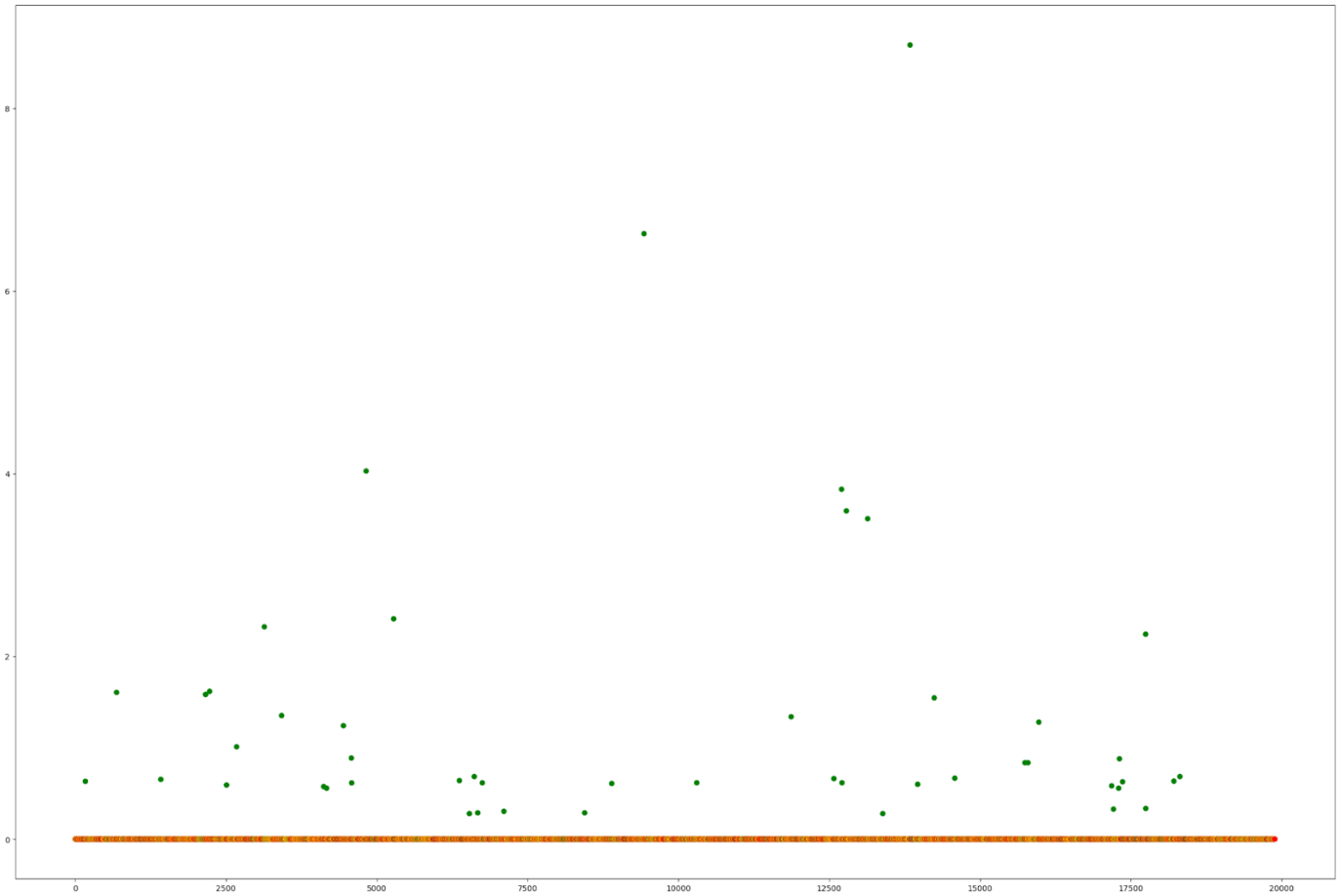
https://free-eunb.tistory.com/14 (DT 시각화 – Graphviz 모듈)

https://jimmy-ai.tistory.com/260 (IQR 기반 이상치 탐지 및 제거)

https://chancoding.tistory.com/191 (파이썬 Pandas apply 함수)

https://wooono.tistory.com/293 (데이터프레임 칼럼 값 조건 변경)

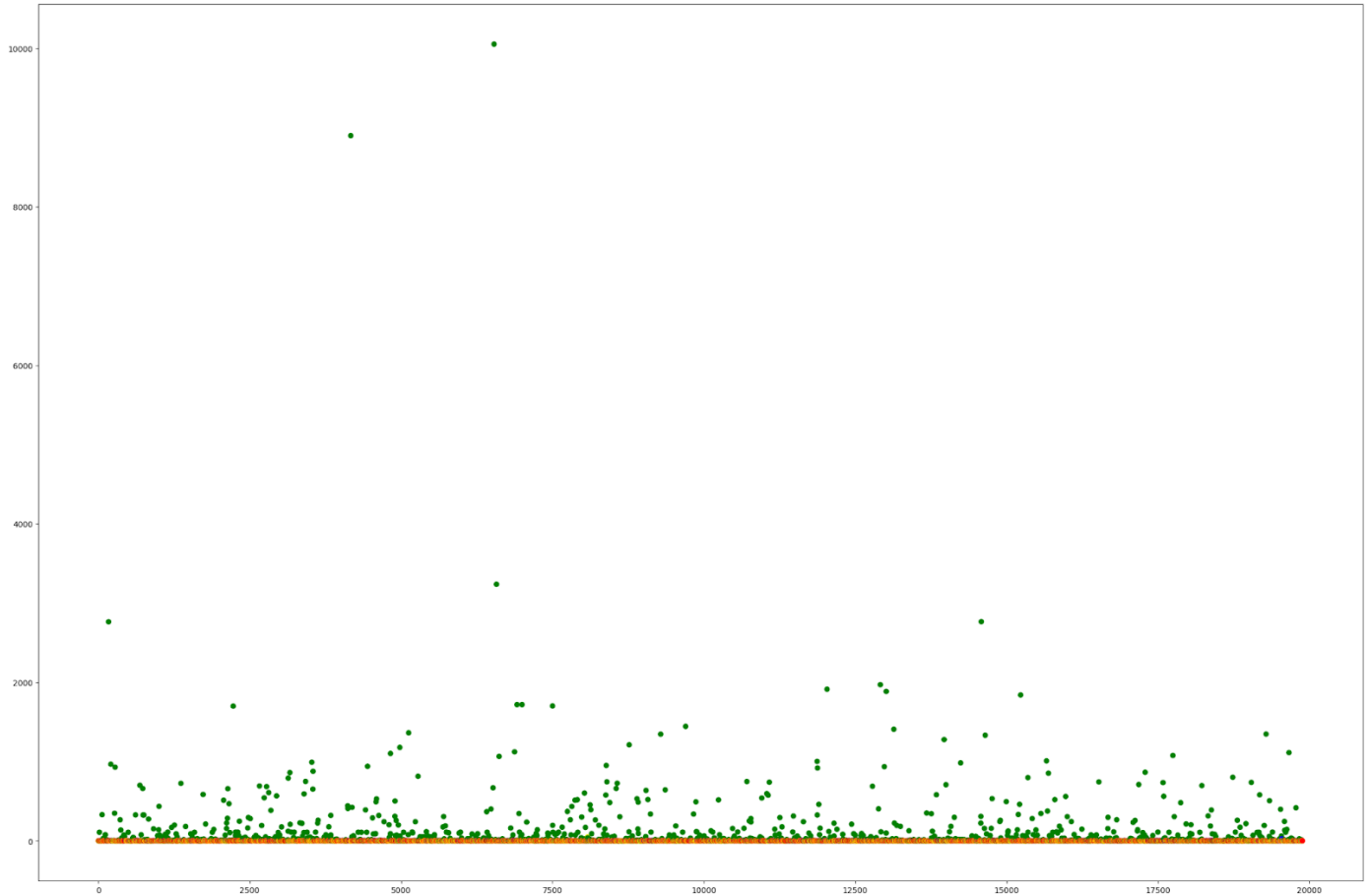
Q&A 보충자료



0 보다 큰 값들이 모두 benign 인 경우의 칼럼 제거 데이터셋 관련

ex) 39번 칼럼

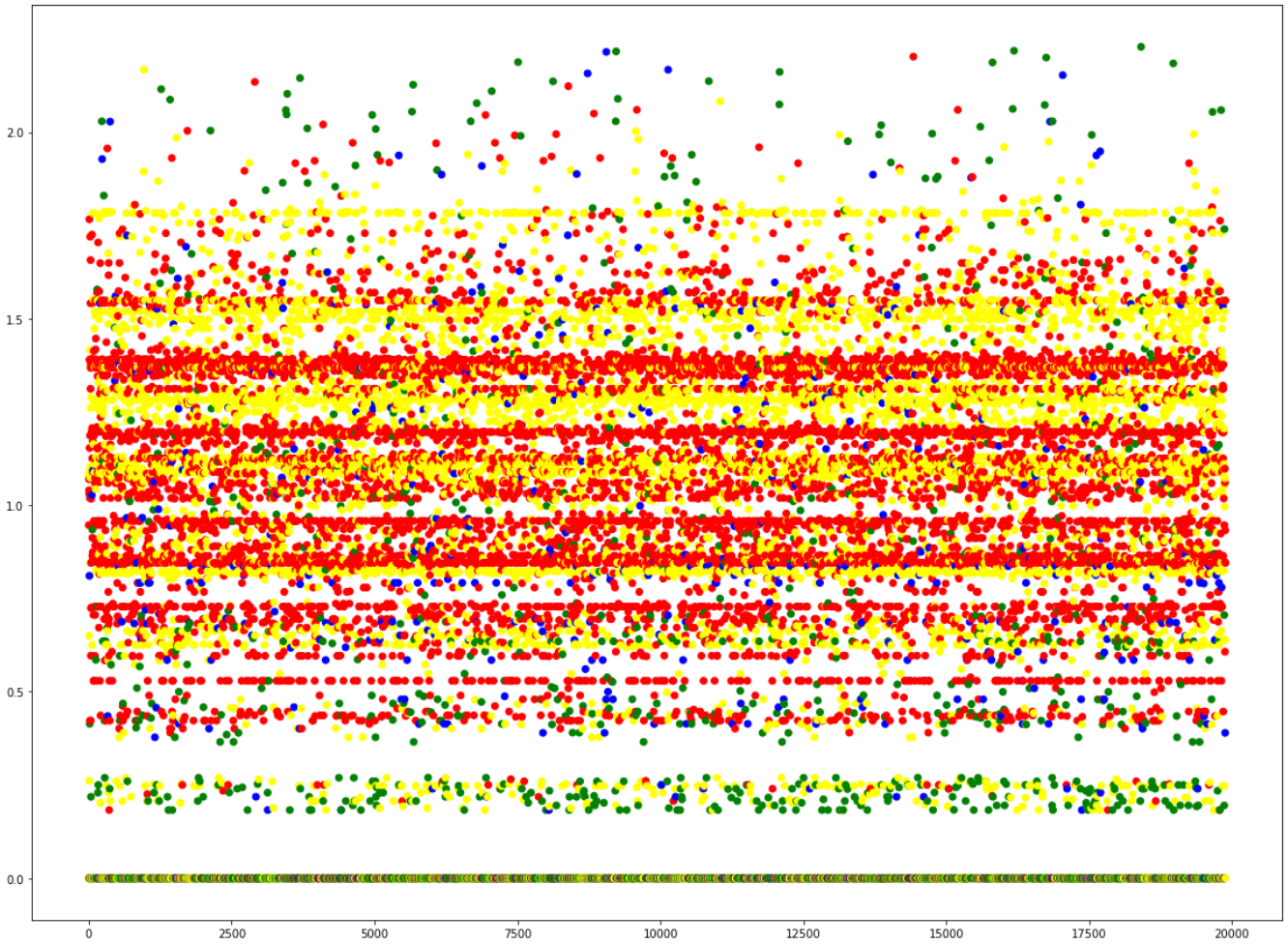
Q&A 보충자료



데이터셋에서 상위 25퍼 값을 적당히 큰 수로 설정하기(1)

ex) 기본데이터셋 32번 칼럼

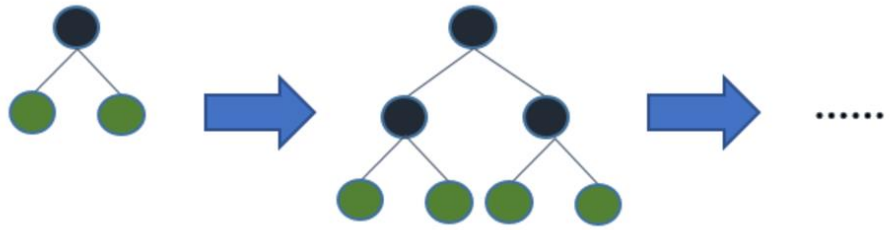
Q&A 보충자료



데이터셋에서 상위 25퍼 값을 적당히 큰 수로 설정하기(2)

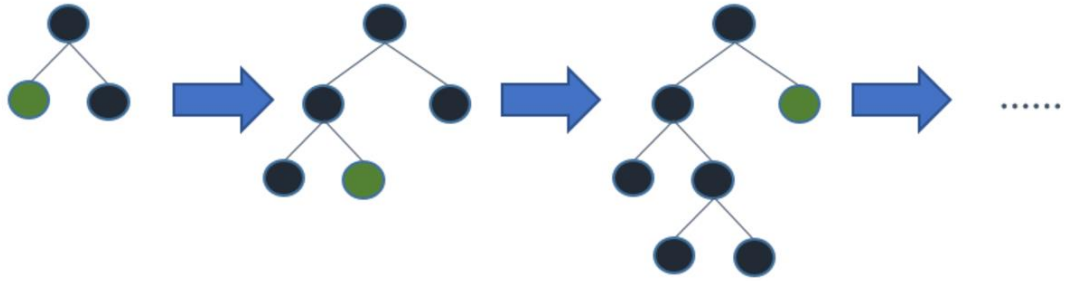
ex) 조작데이터셋 32번 칼럼

Q&A 보충자료



Level-wise tree growth

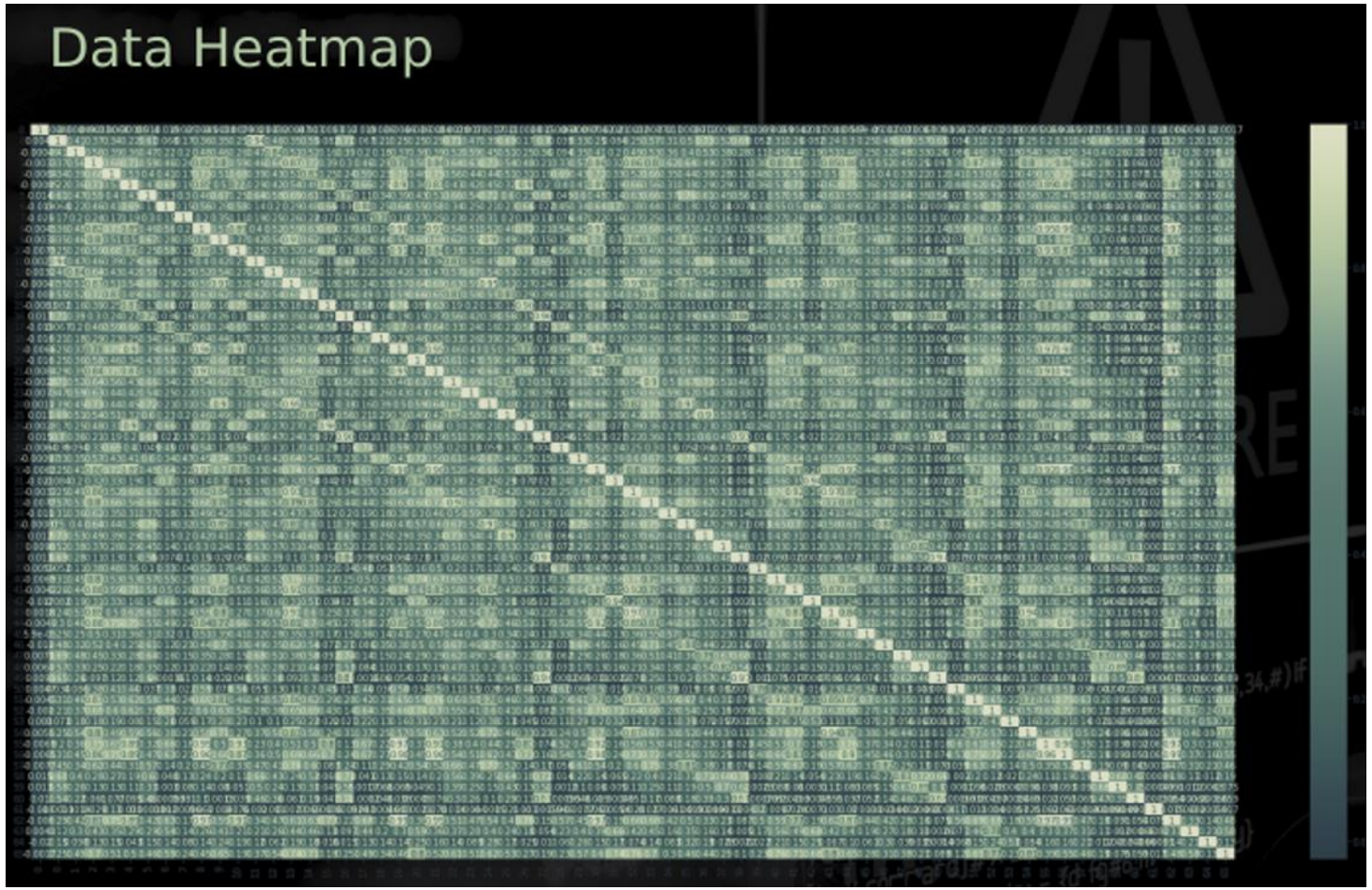
다른 Boosting 알고리즘 작동 방식



Leaf-wise tree growth

LGBM 작동 방식

Q&A 보충자료



각 칼럼별 상관관계를
heatmap으로 표현한 것