건설기계 오일 상태 AI 분류 경진대회

팀명 : 런닝머신

팀원 : 강보원

김동률

박혜빈

송승우

**목차**

1. **프로젝트 개요**
2. 프로젝트 기대효과
3. 개발 환경
4. 개발 일정
5. 데이터 수집
6. 프로젝트 추진 체계
7. **데이터 분석**
   1. 데이터 설명
   2. 데이터 탐색
   3. 데이터 전처리
8. Train 데이터

1) 결측치 확인 후 삭제 / 대체

2) Label Encoding

3) Scaling 사용

4) Over Sampling 사용

5) K-Fold와 stratified K-Fold

1. Test 데이터
2. **훈련 및 모델 생성**
   1. 세트 분할
   2. 테스트 모델 생성
3. XGBoost
4. lightGBM
5. NGBoost
6. **프로젝트 결과 및 진행상황** 
   1. 프로젝트 결과
   2. DACON 진행상황
   3. 프로젝트 소감
7. **참고 자료 및 문헌**
8. **프로젝트 개요**
   1. **프로젝트 기대효과**

Machine Learning을 학습하게 되면서 알고리즘에 대한 관심도가 높아졌다. 특히나 새로운 도전보다 기존 수업에 대한 이해도를 높일 수 있는 학습을 중심으로 Machine Learning의 학습 향상을 기대하게 되었고, 현재 진행 중인 공모전에도 참여를 해보고자 한다. 이를 통해 DACON에서 실시하는 건설기계 오일 상태 AI 분류 경진대회를 통해 데이터 분석 능력 향상을 도모하게 되었다.

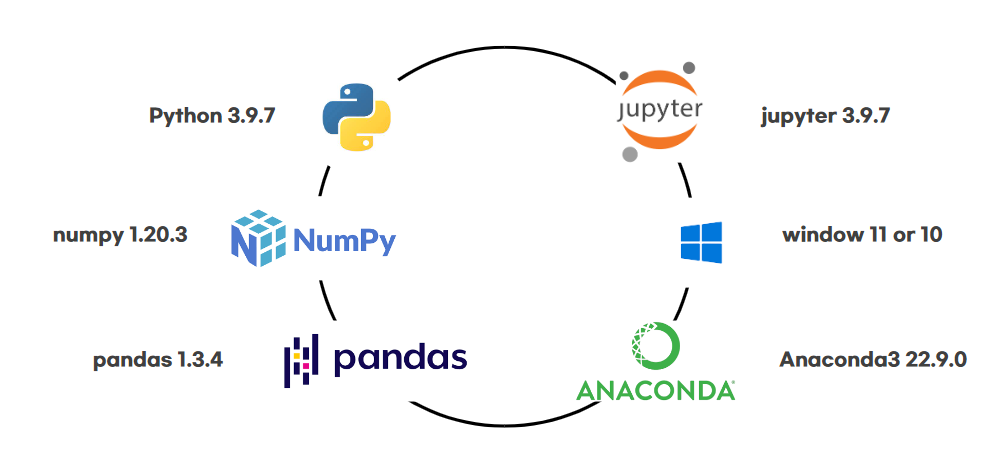
제공 받은 건설기계 오일 데이터 분석을 통해 내연 기관 내에서 엔진오일의 상태를 측정하고, 모델 개발을 통해 적절한 교체 주기를 유추 해 볼 수 있다. 또한 엔진 마모 감소에 따른 엔진 수명이 증가되어 엔진 기능의 최적화를 확인할 수 있고, 엔진오일에 사용되는 요소들을 분리해 산화에 따른 환경오염의 방지를 기대해 볼 수 있도록 한다.

이에 원하는 값을 도출하기 위해 전처리를 실시하고, 특정 알고리즘을 통해 모델을 학습한 후 원하는 목표치(높은 macro f1 score)에 도달할 수 있도록 프로젝트를 진행할 예정이다.

이러한 건설기계 오일 데이터 내에는 엔진오일의 양면성을 띄고 있는 요소들이 존재해 학습 증진에 좋을 것이라 예상하면서 Machine Learning의 학습을 지나 Deep Learning의 학습에까지 이어지게 된다면 더 나은 결과값을 도출해낼 수 있을 것이라 예상한다.

1. **프로젝트 개요**

**b. 개발 환경**



⚬ 운영 체제 -> Windows 11 / Windows 10

⚬ 언어 -> Python : 3.9.7

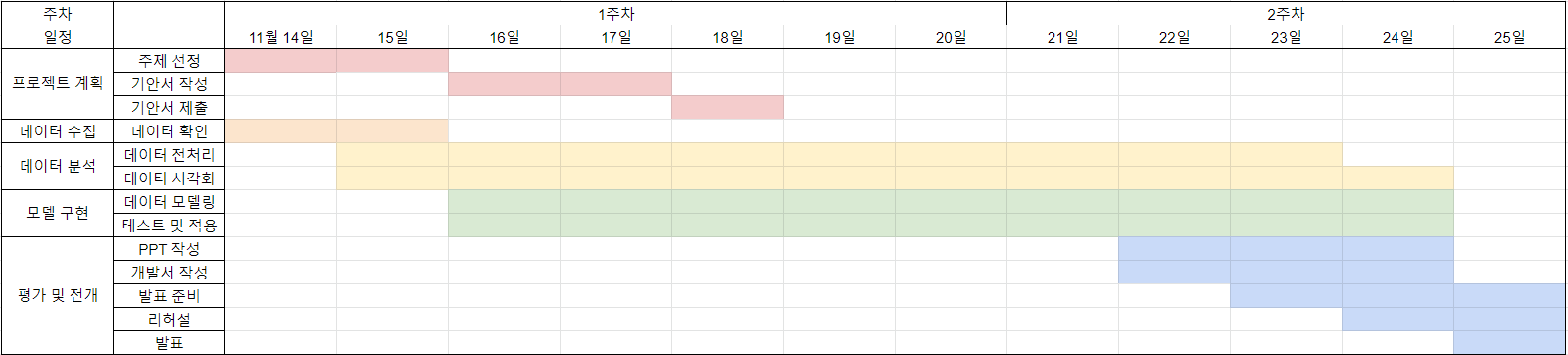
⚬ 데이터 분석, 전처리 -> Numpy : 1.20.3 / Pandas : 1.3.4

⚬ IDE -> Anaconda3 : 22.9.0, Jupyter lab : 3.9.7

⚬ System -> Memory 20G, Hard 256GB

1. **프로젝트 개요**

**c. 개발 일정**



프로젝트 계획 1) 주제 선정 : 11월 14일 ~ 15일

2) 기안서 작성 : 11월 16일 ~ 17일

3) 기안서 제출 : 11월 18일

데이터 수집 1) 데이터 확인 : 11월 14일 ~ 15일

데이터 분석 1) 데이터 전처리 : 11월 15일 ~ 23일

2) 데이터 시각화 : 11월 15일 ~ 24일

모델 구현 1) 데이터 모델링 : 11월 16일 ~ 24일

2) 테스트 및 적용 : 11월 16일 ~ 24일

평가 및 전개 1) PPT 작성 : 11월 22일 ~ 24일

2) 개발서 작성 : 11월 22일 ~ 24일

3) 발표 준비 : 11월 23일 ~ 25일

4) 리허설 : 11월 24일 ~ 25일

5) 발표 : 11월 25일

1. **프로젝트 개요**

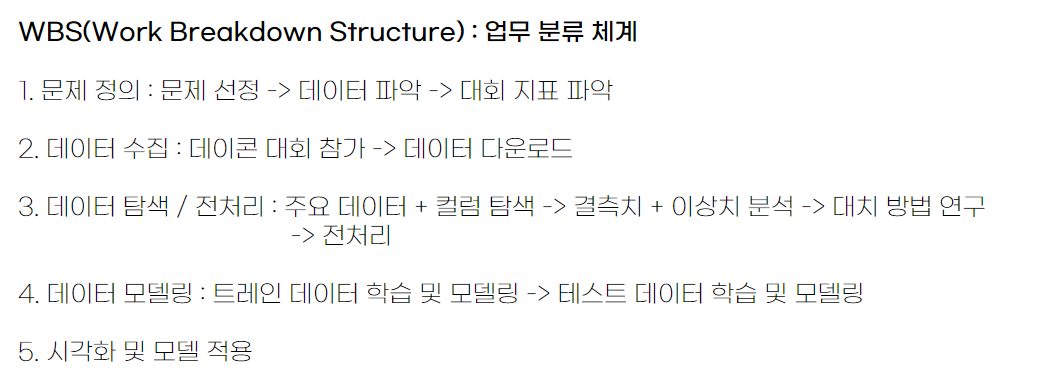
**d. 데이터 수집**



: DACON의 ‘건설기계 오일 상태 분류 AI 경진대회’를 통해 제공 받은 세 개의 데이터를 사용한다.

1. 오일 Train 데이터
2. 오일 Test 데이터
3. oil\_info(오일정보) 데이터
4. **프로젝트 개요**

**e. 프로젝트 추진체계**



: 빅데이터 분석의 5단계인 WBS(Work Breakdown Structure)을 통해 문제 정의, 데이터 수집,

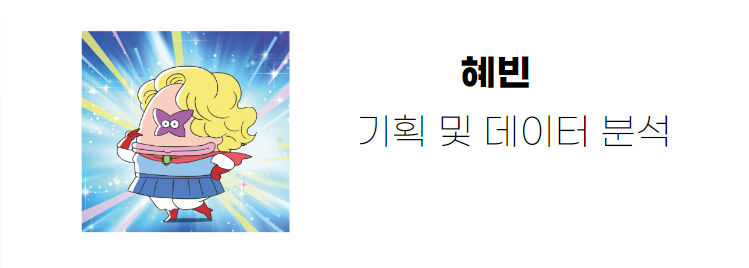
데이터 탐색 및 전처리, 데이터 모델링, 시각화 및 모델 적용을 활용한다.

1. **프로젝트 개요**

**e. 프로젝트 역할 및 임무**



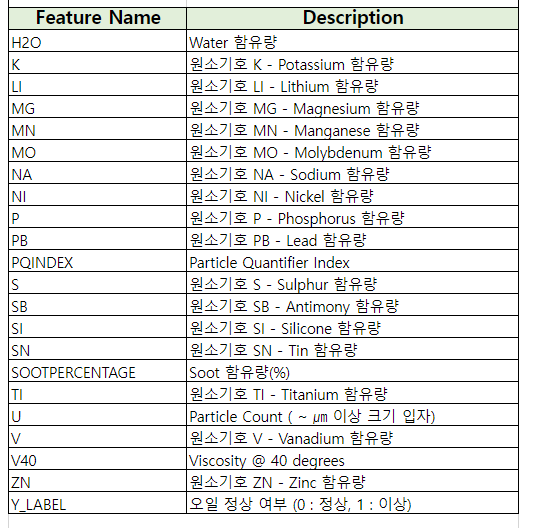




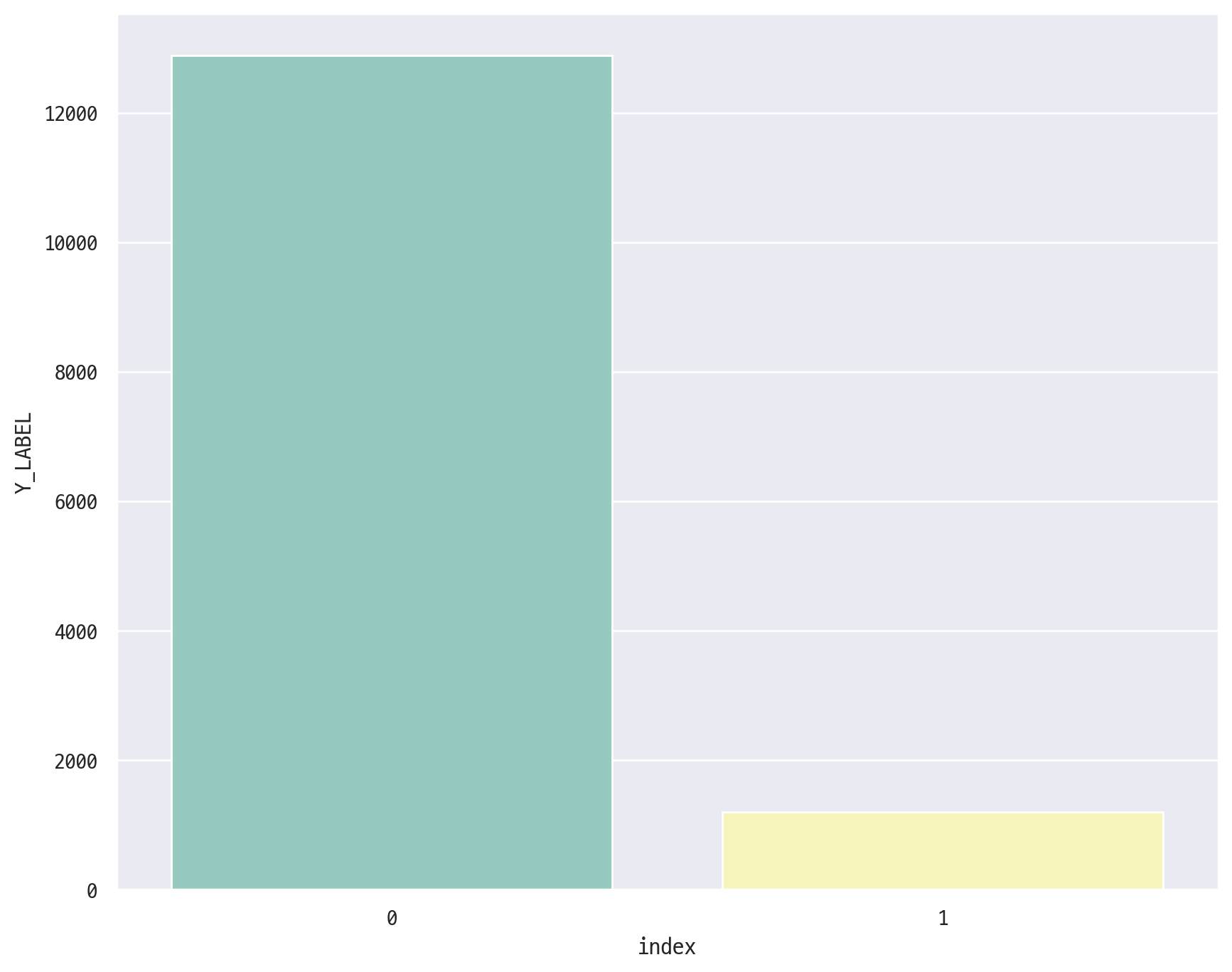
1. **데이터 분석**
   1. **데이터 설명 - data\_info 확인**

: DACON에서 제공받은 각 Column별 데이터 세부 설명





* 1. **데이터 탐색**



**12892**

**1203**

: 데이터 전처리를 시작하기 전, 데이터 편향을 확인한다. Y\_LABEL 값을 통해 0의 값은 12892, 1의 값은 1203로 극심한 편향을 가지고 있음을 확인할 수 있다. 이를 통해 편향을 줄이는 작업이 필수적이라 판단했다.

**2. 데이터 분석**

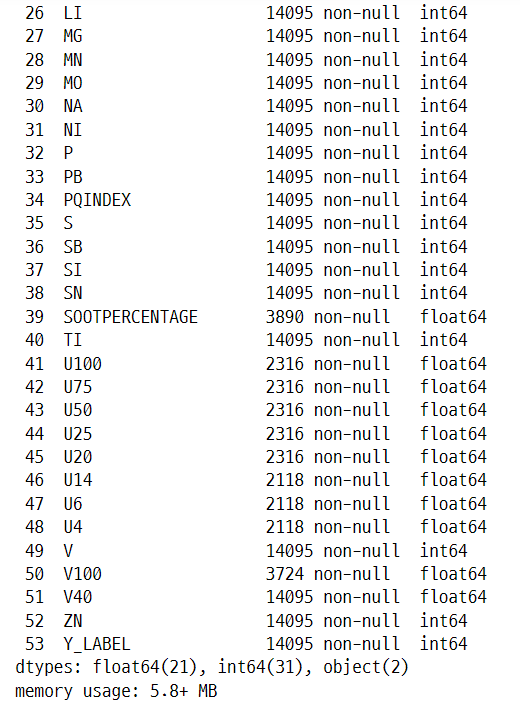
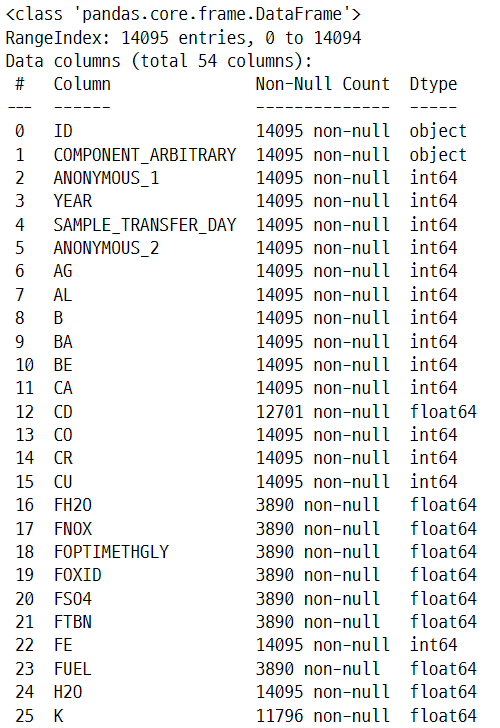
* 1. **데이터 전처리**

1. **오일 Train 데이터**
2. **결측치 확인 후 삭제 / 대체**
3. **삭제**
4. **0으로 대체**
5. **Iterative Imputer**
6. **Label Encoding : ‘COMPONENT\_ARBITRARY’, ‘YEAR’ 사용**
7. **StandardScaler 사용**
8. **Over Sampling 사용 이유**

**2. 데이터 분석**

**b. 데이터 전처리**

1. **오일 Train 데이터**



: 오일 Train 데이터를 통해 14095개의 데이터값과 54개의 columns를 확인할 수 있다. 이를 통해 결측치를 확인하고 난 뒤 다음과 같은 세 가지 방법을 실시한다. 가장 먼저 삭제를 해보고, 다음으로 0으로 대체, Iterative Imputer를 실시한다. 이후, Label Encoding을 통해 범주형 데이터를 수치형 데이터로 변환하고, StandardScaler를 사용해보고자 한다.

**1. 결측치 확인 후 결측치 제거**

1) 삭제

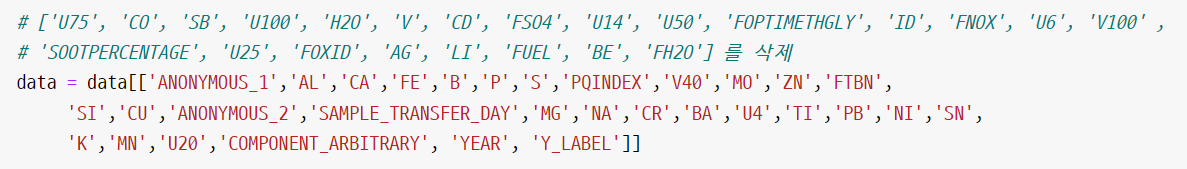
: feature\_importance가 10 이하인 columns들을 삭제한다.

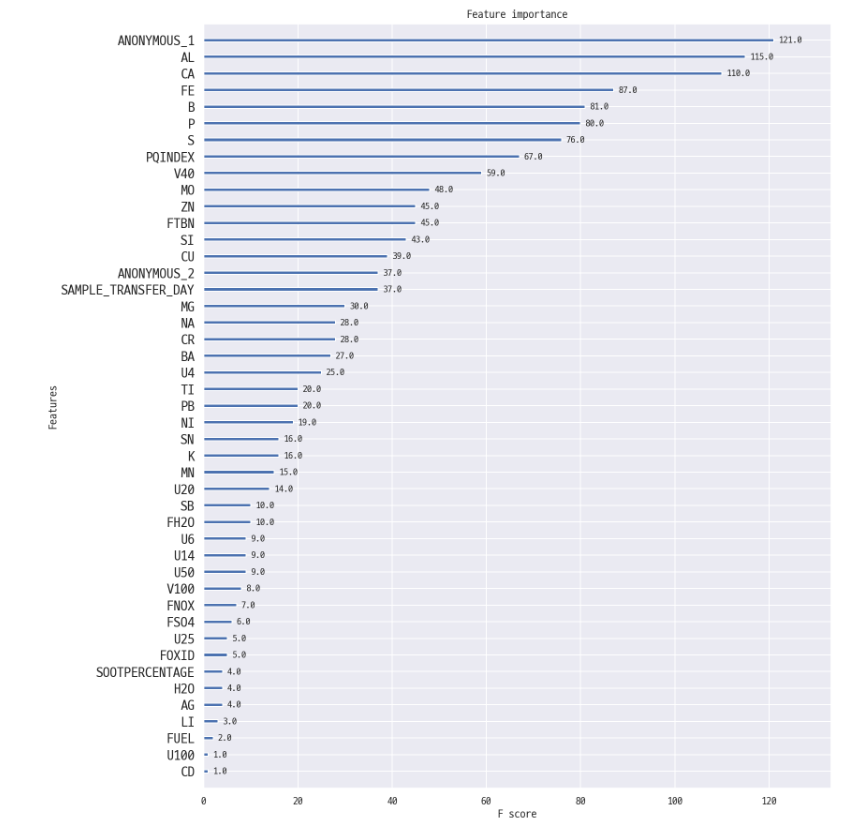
-> 'U75', 'CO', 'SB', 'U100', 'H2O', 'V', 'CD', 'FSO4', 'U14', 'U50', 'FOPTIMETHGLY', 'ID',

'FNOX', 'U6', 'V100', 'SOOTPERCENTAGE', 'U25', 'FOXID', 'AG', 'LI', 'FUEL', 'BE',

'FH2O'

=> **총 23개 삭제**







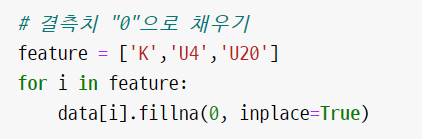
**-> feature\_importance가 10 이하인 columns들을 삭제**

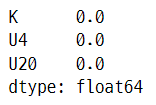
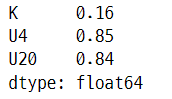
2) 0으로 대체 : ‘K, U4, U20’

- ‘K’는 함유되어 있지 않다고 판단

- ‘U4’, ‘U20’는 해당하는 크기의 입자가 발견되지 않았다고 판단

-> 검출되지 않았다고 판단하여 0으로 대체한다.

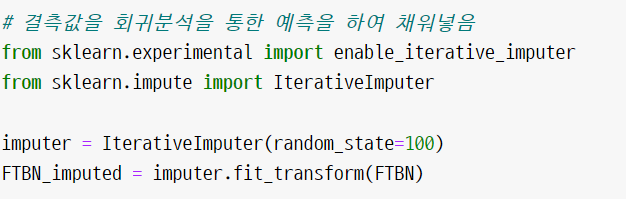




**대체 전 대체 후**

3) Iterative Imputer

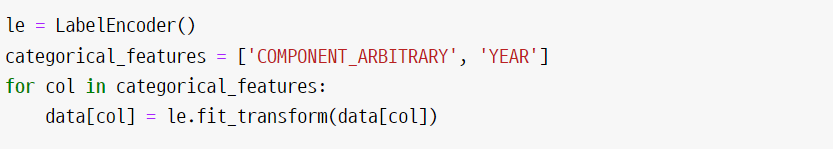
-> FTBN을 이용해 회귀분석을 실행한 후 결측된 전알칼리값을 예측한다.

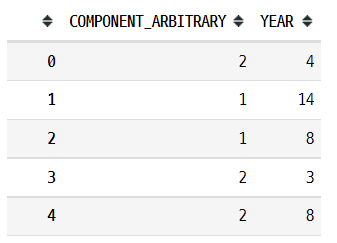
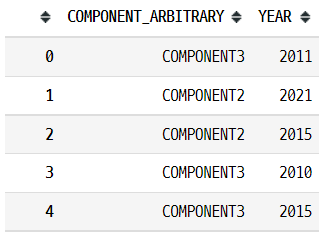


**2. Label Encoding : ‘COMPONENT\_ARBITRARY’, ‘YEAR’ 사용**

-> n개의 범주형 데이터를 0부터 n-1까지의 수치 데이터로 표현한다.

이러한 Train 데이터 내에서 ‘COMPONENT\_ARBITRARY’와 ‘YEAR’ 컬럼만 사용한다.





**3. Scaling 사용**

1. **Scaling을 사용하는 이유**

* 다차원의 값들을 쉽게 비교 분석하기 쉽게 만들어 준다.
* 자료의 오버플로우(overflow)나 언더플로우(underflow)를 방지한다.
* 독립 변수의 공분산 행렬의 조건수(condition number)를 감소시켜 최적화 과정에서의 안정성 및 수렴 속도를 향상 시킨다.

**(2) 사용한 Scaling**

* log : 정규성을 높이고 분석에서 정확한 값
* MinMaxScaler : 최대/최소값이 각각 1과 0이 되도록 조정
* Normalizer : 유클리드 거리가 1이 되도록 데이터를 조정
* StrandardScaler : 기본 스케일, 평균과 표준편차 사용
* Yeo-Johnson : 정규성, 등분산성 가정 만족하지 않을 경우 처리

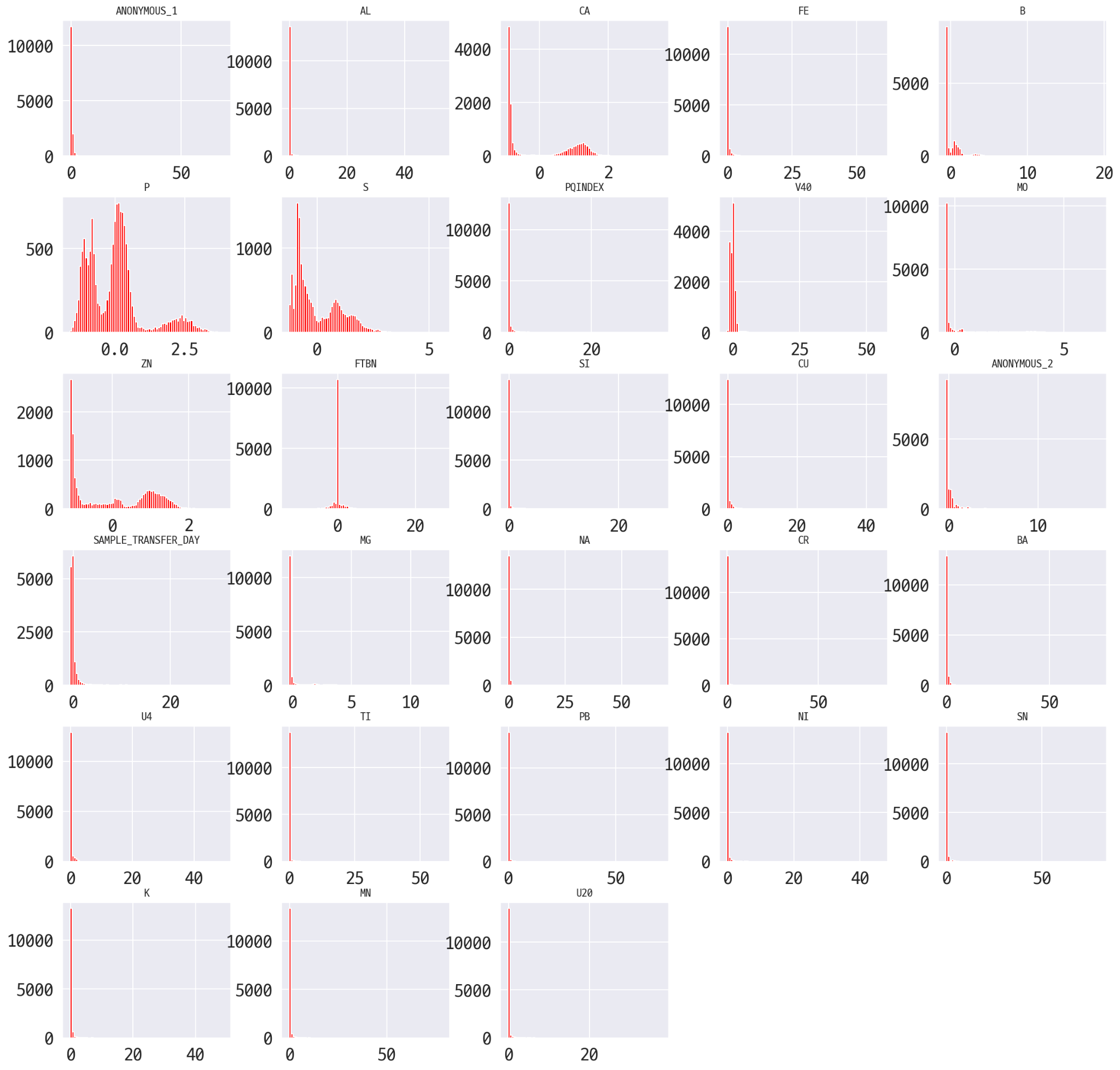
(연속형 변수 중 치우침이 심한 변수가 많다. 이를 변수변환하여 정규분포에 가까운 형태로 맞추게 되면 예측모형의 성능 향상과 안정성을 기대할 수 있다.)

**(3) 선택한 Scaling : log, StandardScaler, Yeo-Johnson 를 선택**

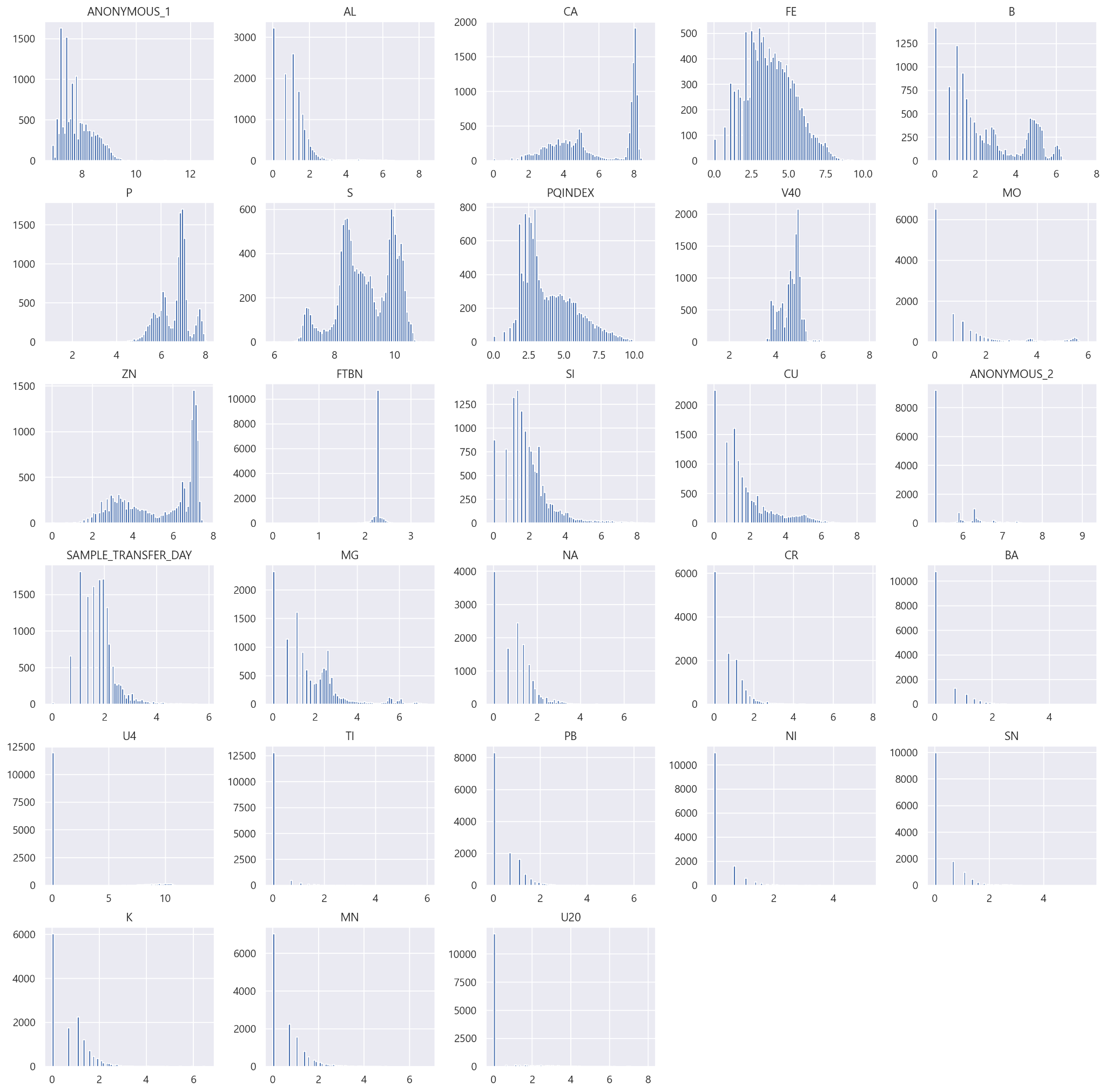
-> 'ID', 'COMPONENT\_ARBITRARY', 'YEAR', 'Y\_LABEL'을 이용해 drop한 후 스케일링을

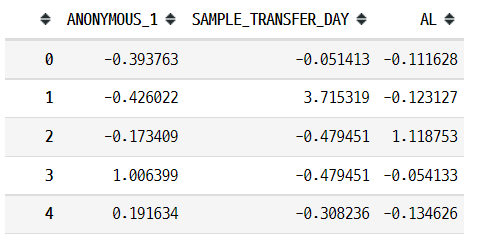
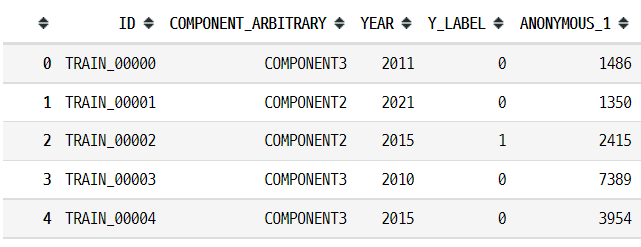
처리한다.

1. **StandardScaler Scaling**

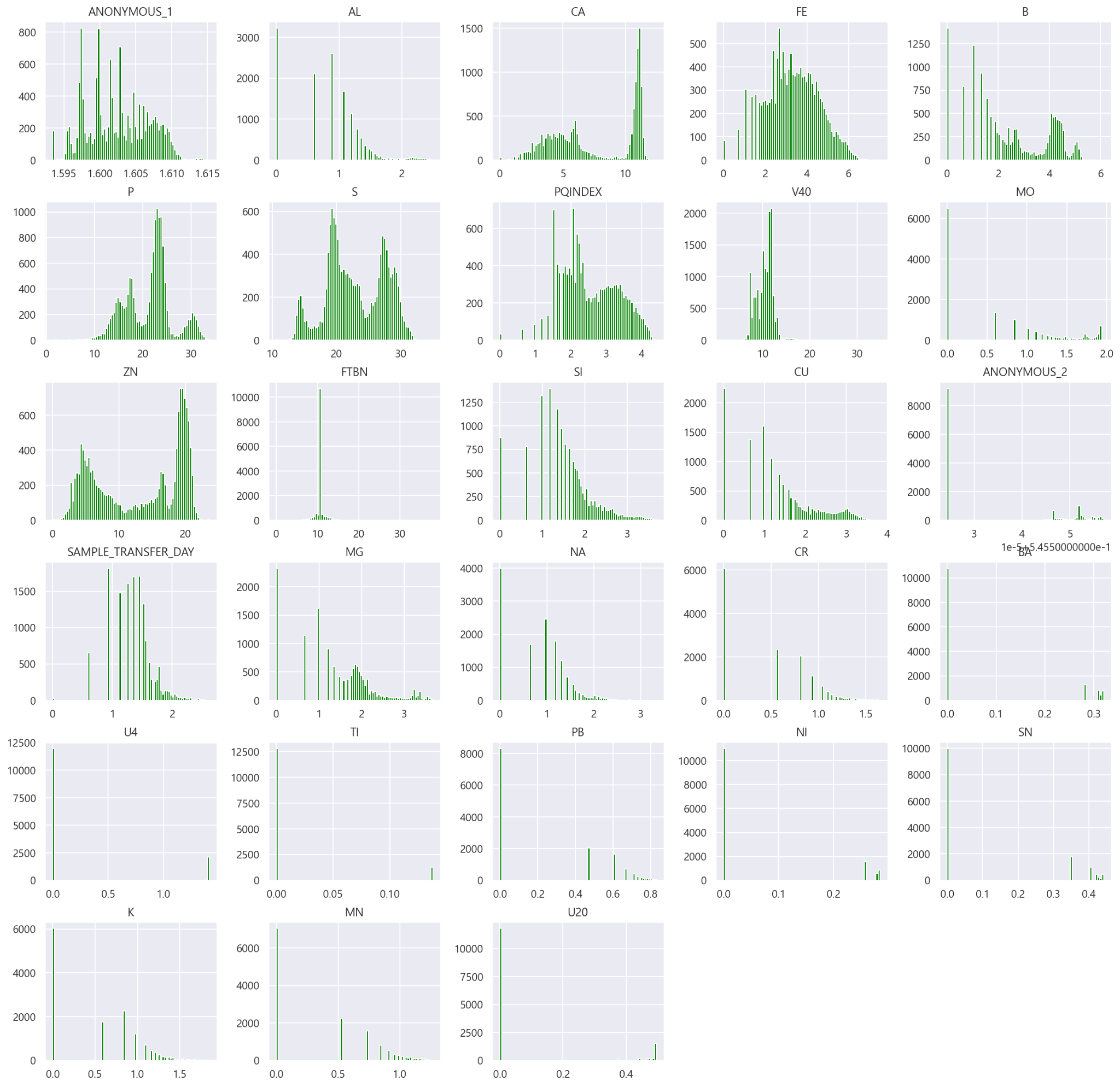


1. **log Scaling**





1. **Yeo\_Johnson Scaling**



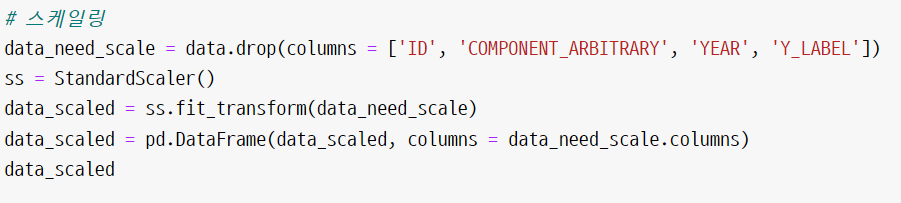
**4. Over Sampling 사용 이유**

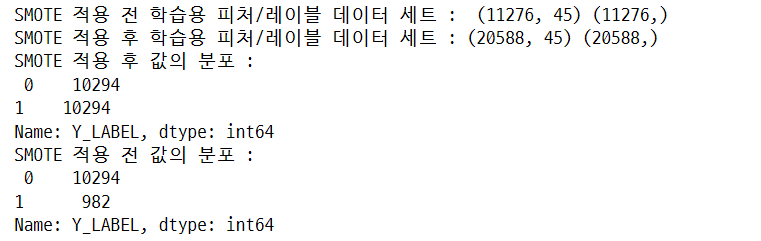
1. Over Sampling 선택

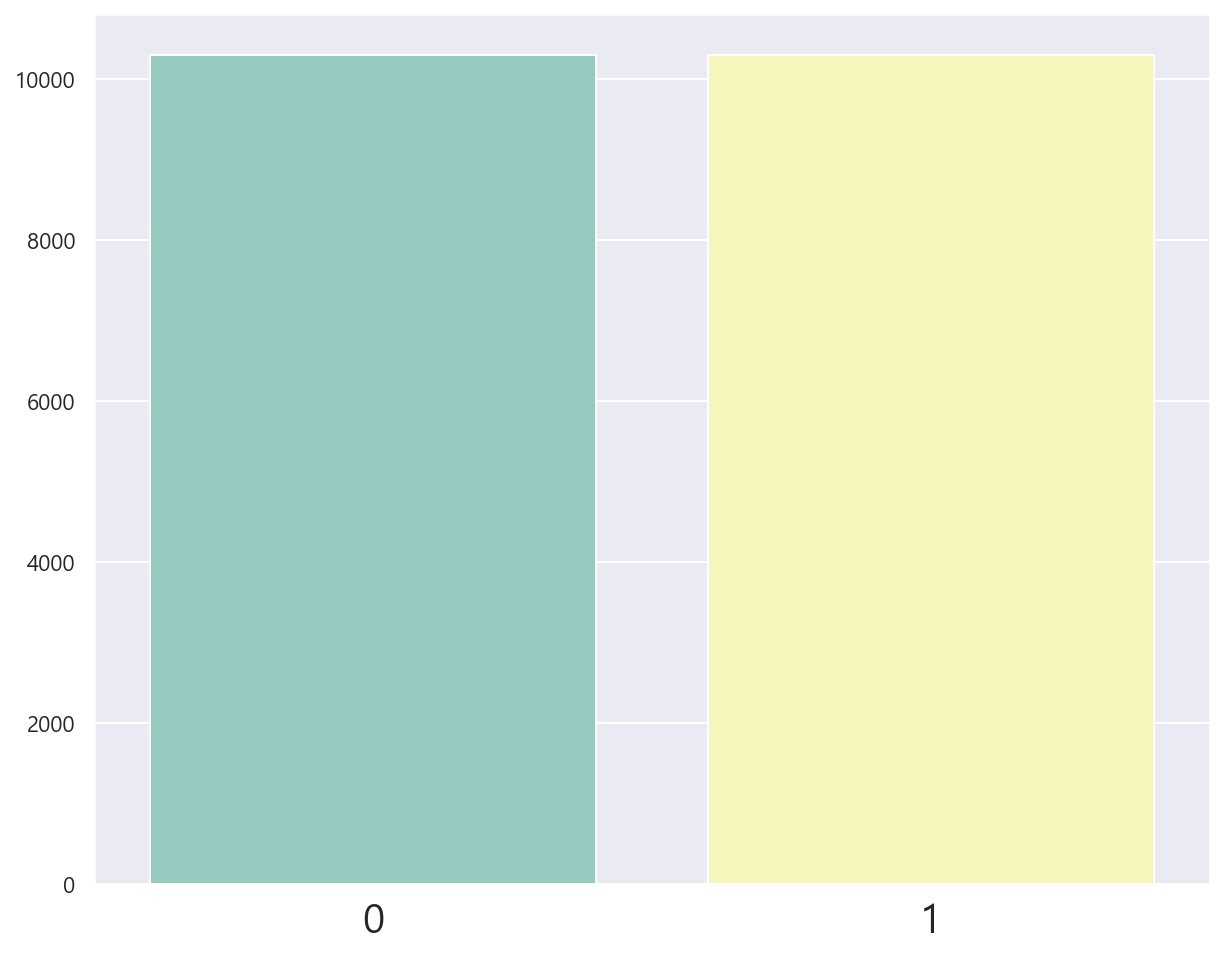
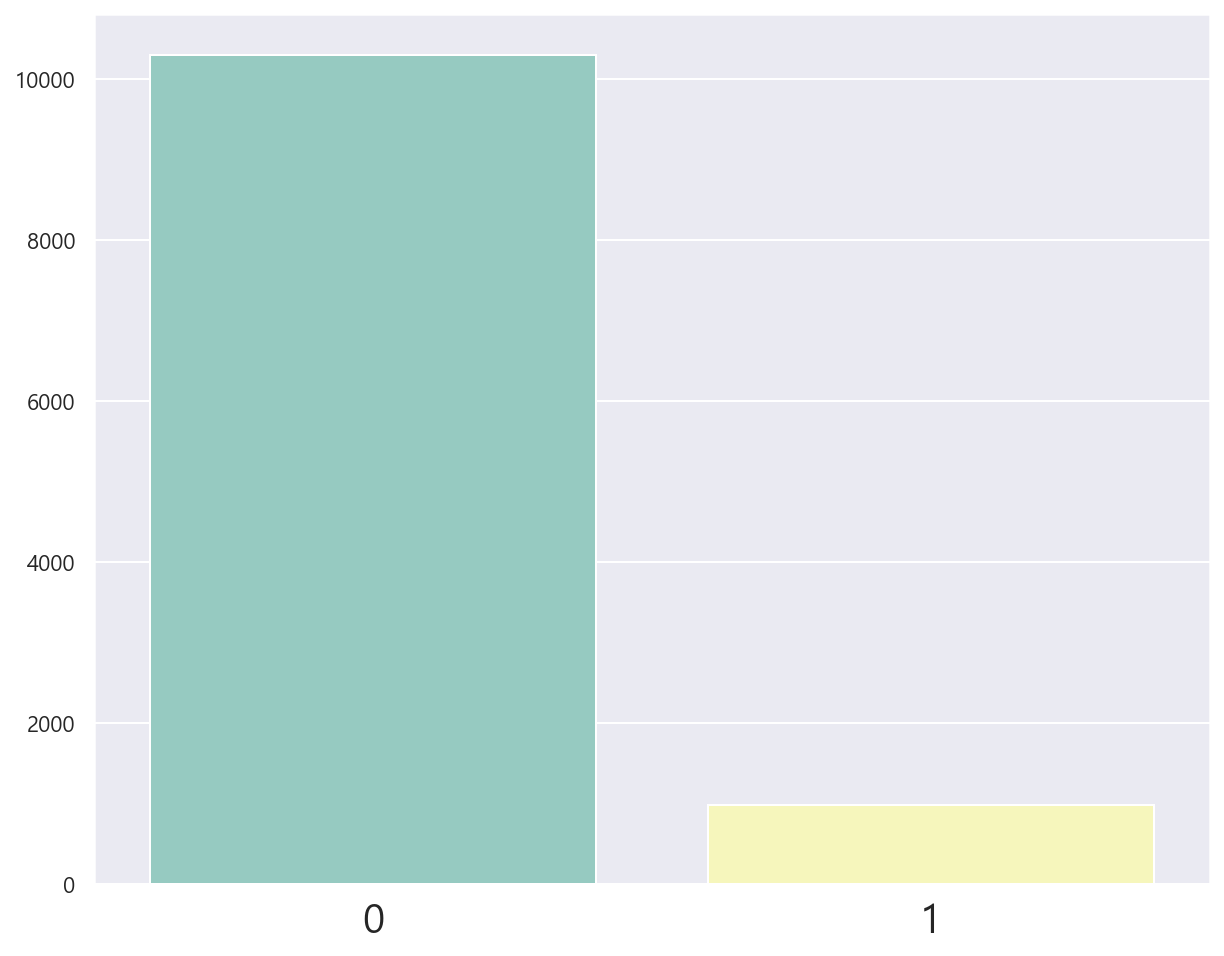
\* Under Sampling : 잠재적 정보의 가치가 높은 데이터도 버릴 가능성이 존재한다. 이로 인해 데이터가 사라지는 위험이 존재한다.

1. SMOTE(Synthetic Minority Over-sampling Technique)

: 낮은 비율로 존재하는 클래스 데이터를 최근접 이웃 (k-NN) 알고리즘을 활용해 새롭게 생성한다. 오버 샘플링 기법 중 단순 무작위 추출을 통해 데이터의 수를 늘리는 방법도 존재하는데, 데이터를 단순하게 복사하기 때문에 과적합 문제가 발생하기도 한다. 이에 반해 SMOTE는 알고리즘을 기반으로 데이터를 생성하므로, 과적합 발생 가능성이 단순 무작위 방법보다 적다.







**적용 전 적용 후**

-> 기존에 편향이 극심했던 Y\_LABEL을 SMOTE를 통해 0과 1의 값 분포가 같아진다.

**5. K-Fold와 stratified K-Fold**

-> sklearn에서는 K-Fold 교차검증을 구현하기 위해 K-Fold와 stratified K-Fold 클래스를 제공한다.

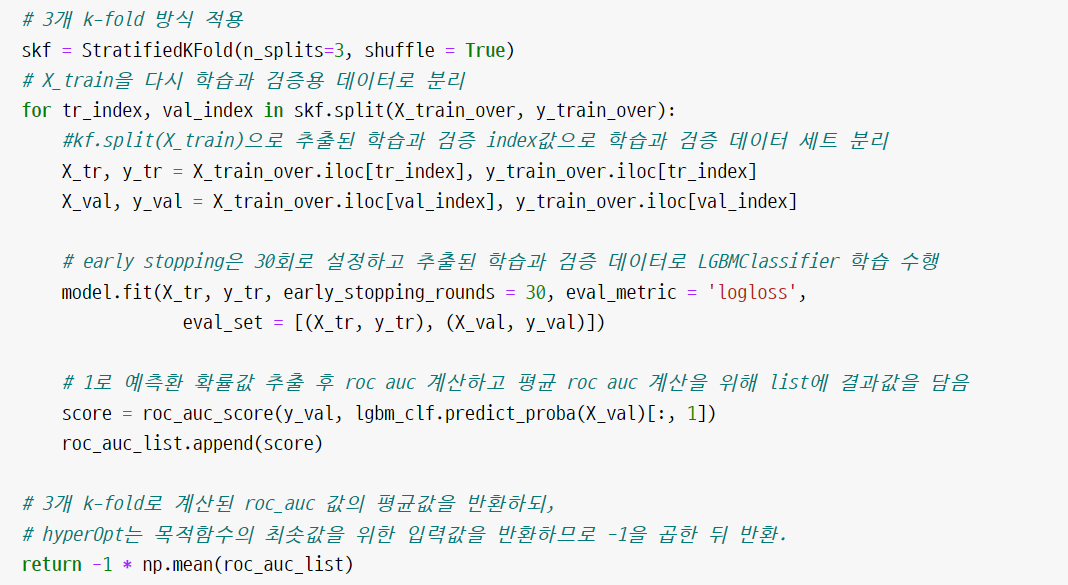
(\* K-Fold 교차 검증 : K개의 홀드세트에 K번의 학습과 검증 평가를 반복 수행하는 것)

1. K-Fold : 가장 보편적인 K-Fold 방법.

학습데이터 셋과 검증데이터 셋으로 나누어 진행한다.

1. stratified K-Fold : 불균형한 DataSet을 위한 KFold 방법.

K-Fold의 문제점인 검증데이터의 비율을 일정하게 유지하지 못하는 것을

일정하게 유지하고, 교차 검증을 진행한다.



-> 기존 K-Fold의 편향이 너무 심하게 불균형을 해소하기 위해 stratified K-Fold를 사용한다.

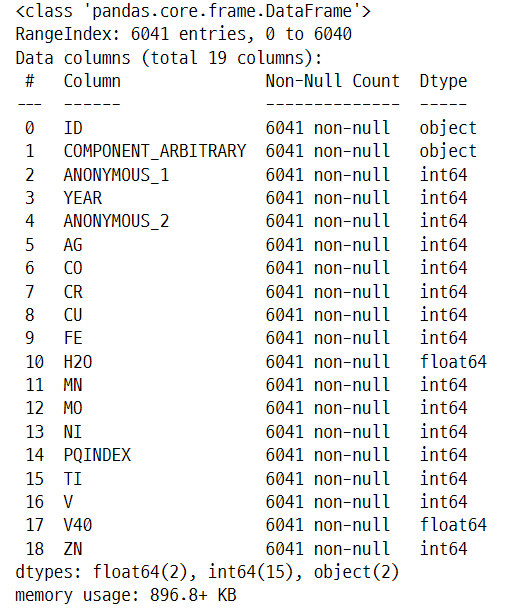
**2. 데이터 분석 - 1) 데이터 전처리**

**(2) 오일 Test 데이터**

1. **결측치 확인 후 결측치 없음 확인**
2. **Label Encoding : ‘COMPONENT\_ARBITRARY’, ‘YEAR’ 사용**

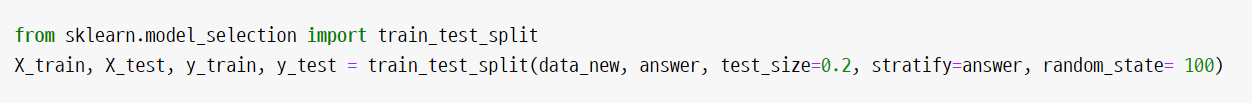
**-> Train 데이터와 동일하게 적용한다.**

1. **StandardScaler 사용**

**(2) 오일 Test 데이터**

: 오일 Test 데이터를 통해 6041개의 데이터값과 19개의 columns를 확인할 수 있다. 이를 통해 결측치를 확인하고 난 뒤 결측치 없음을 확인하고, 다음과 같은 두 가지 방법을 실시한다. Label Encoding을 통해 Train 데이터와 동일한 방법으로 범주형 데이터를 수치형 데이터로 변환한다. 이후 StandardScaler를 사용해보고자 한다.

**3. 훈련 및 모델 생성 - 1) 세트 분할**



1) test size : 테스트 데이터셋의 비율(float)이나 갯수(int)

2) stratify : 지정한 Data의 비율을 유지

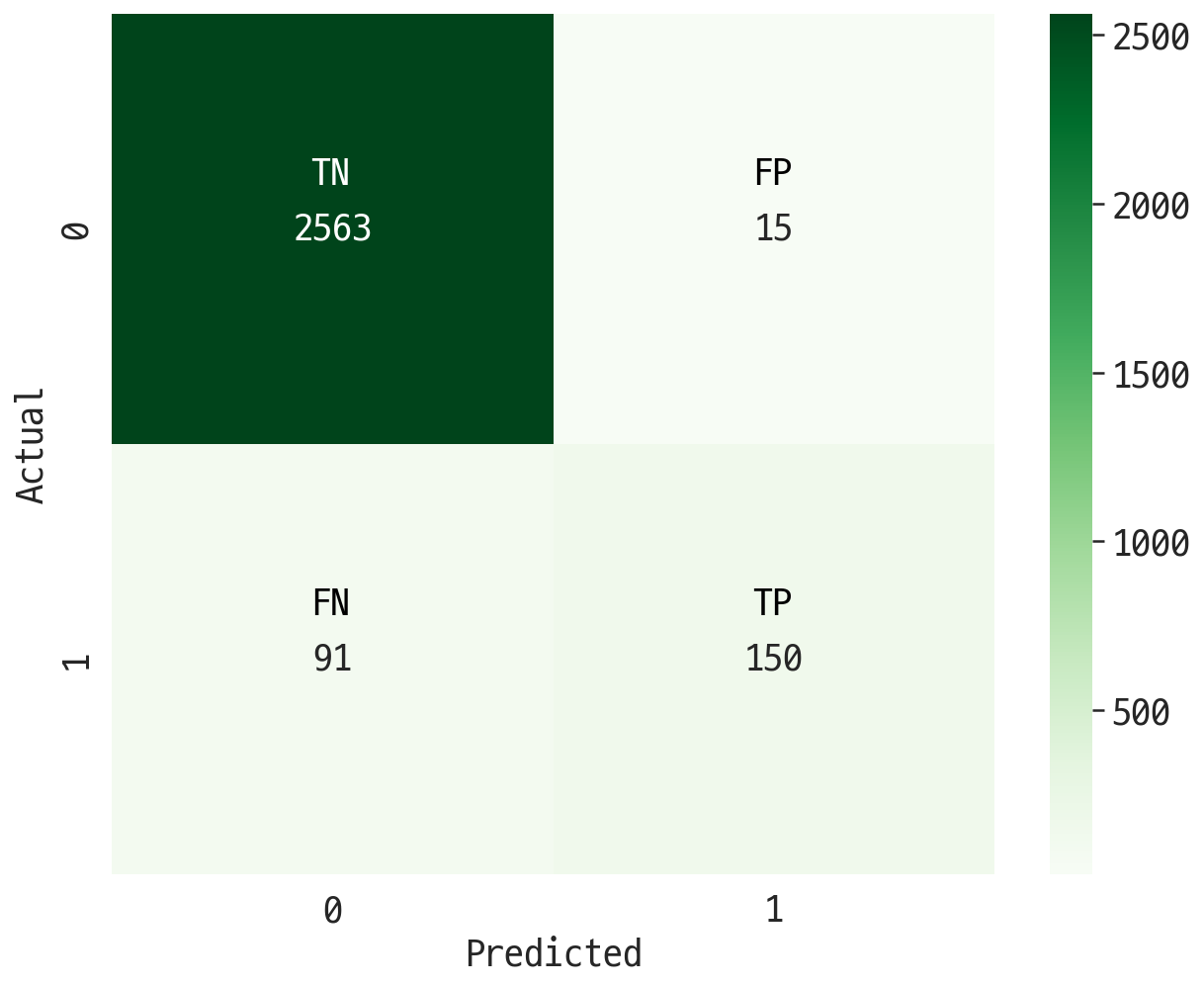
3) random\_state : 데이터 분할시 이루어지는 shuffle을 위한 시드값

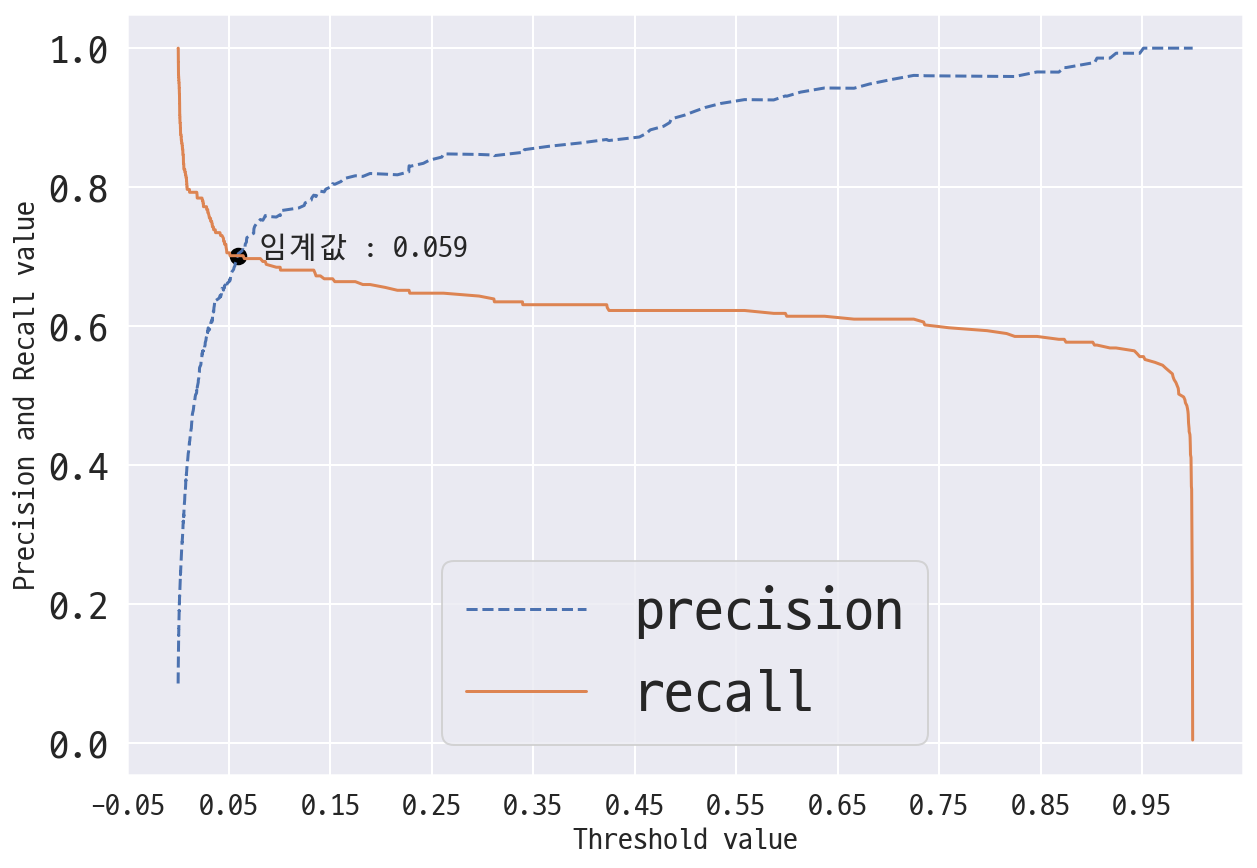
**3. 훈련 및 모델 생성 - 2) 테스트 모델 생성**

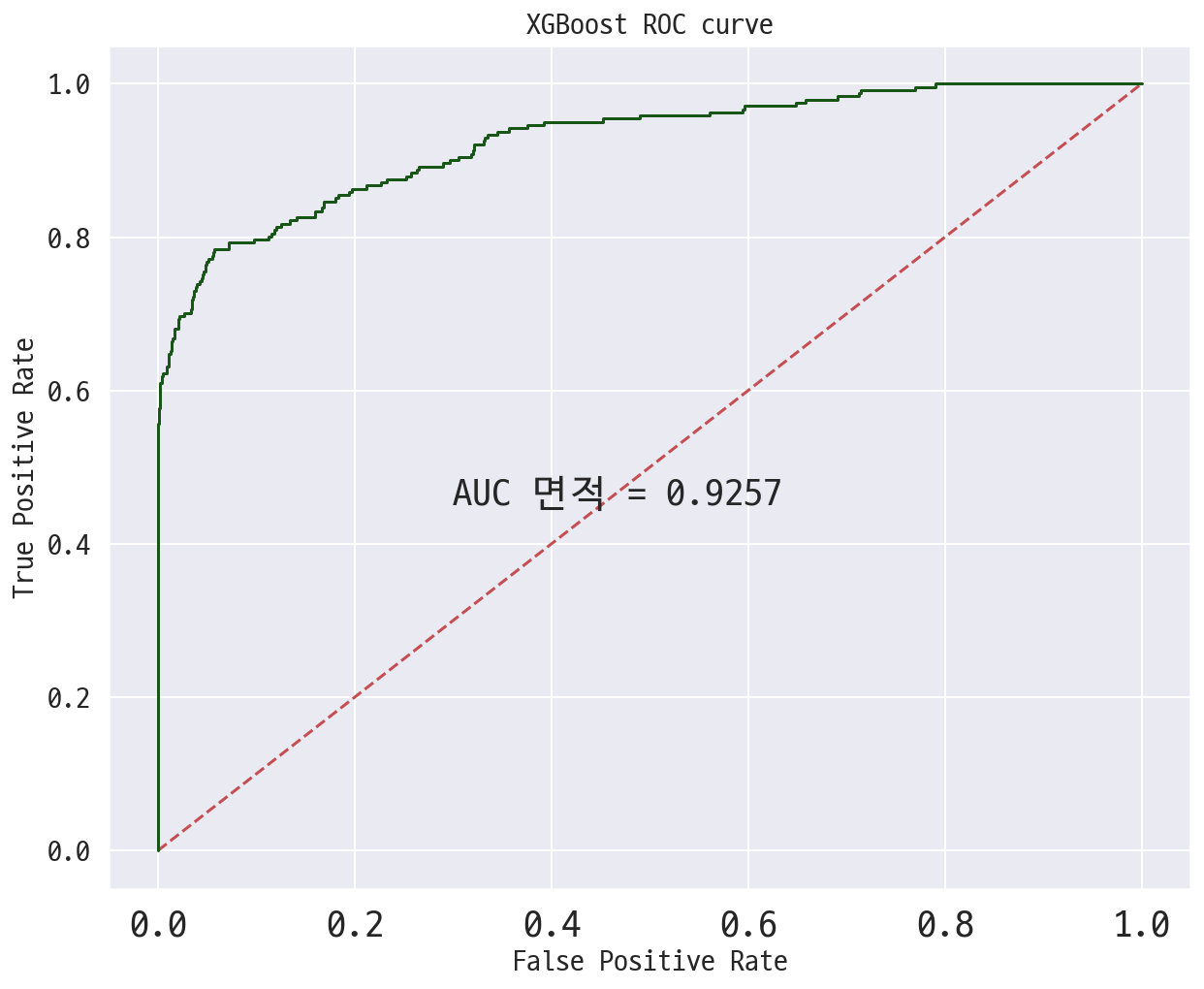
**-> 2종 오류에 해당하는 개수를 최소화하고 AUC 값을 최대화하는 작업을 수행한다.**

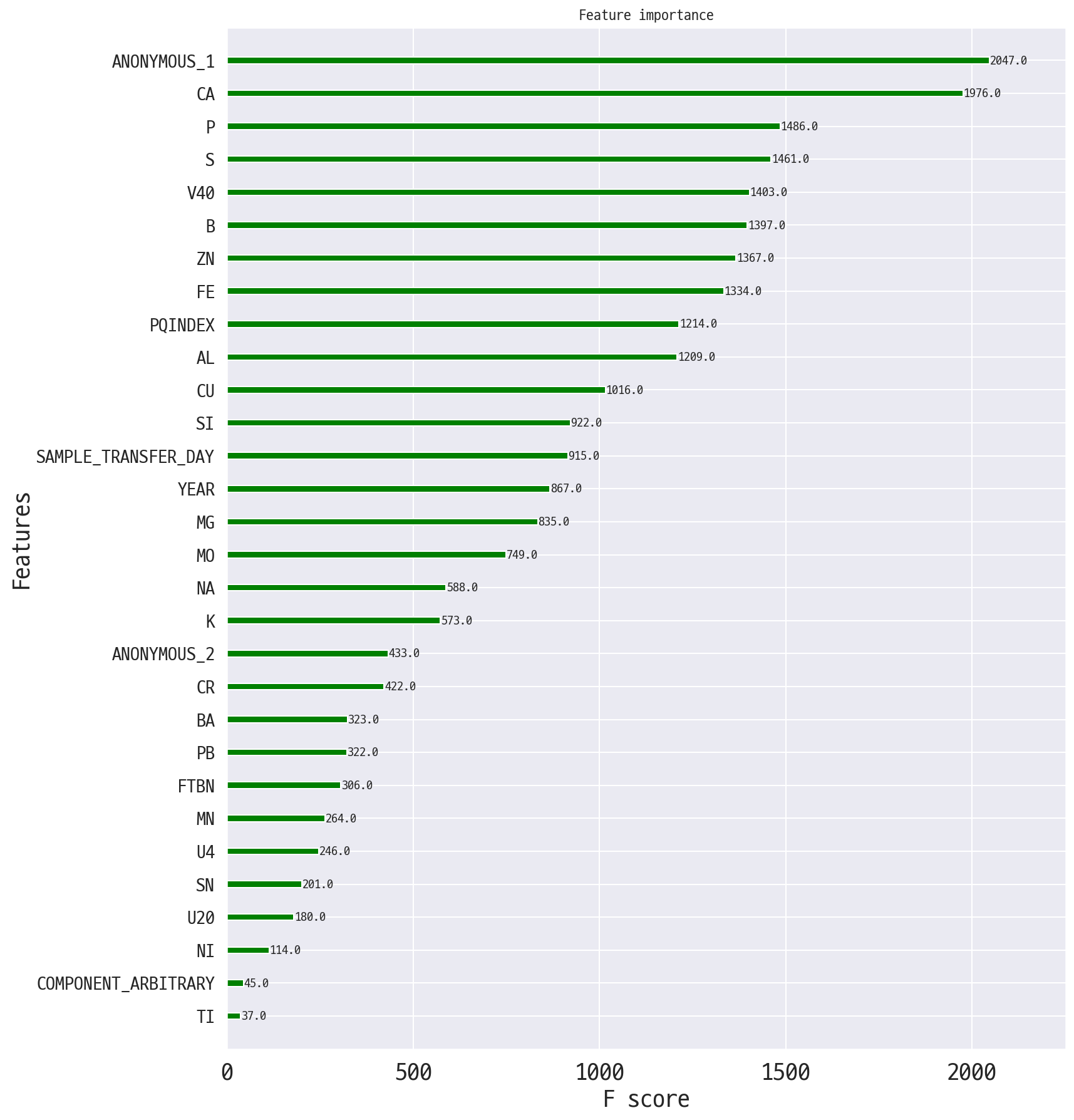
* **테스트 모델을 생성하기 위해 여러 전처리 과정과 알고리즘 선택, 하이퍼파라미터 튜닝, 교차 검증 등을 수행한다.**
* **혼동행렬, 재현율+정밀도, ROC-AUC, feature\_importance 순서로 진행**

1. **Yeo-Johnson + Over Sampling -> HyperOpt + XGBoost / AUC : 0.9257**

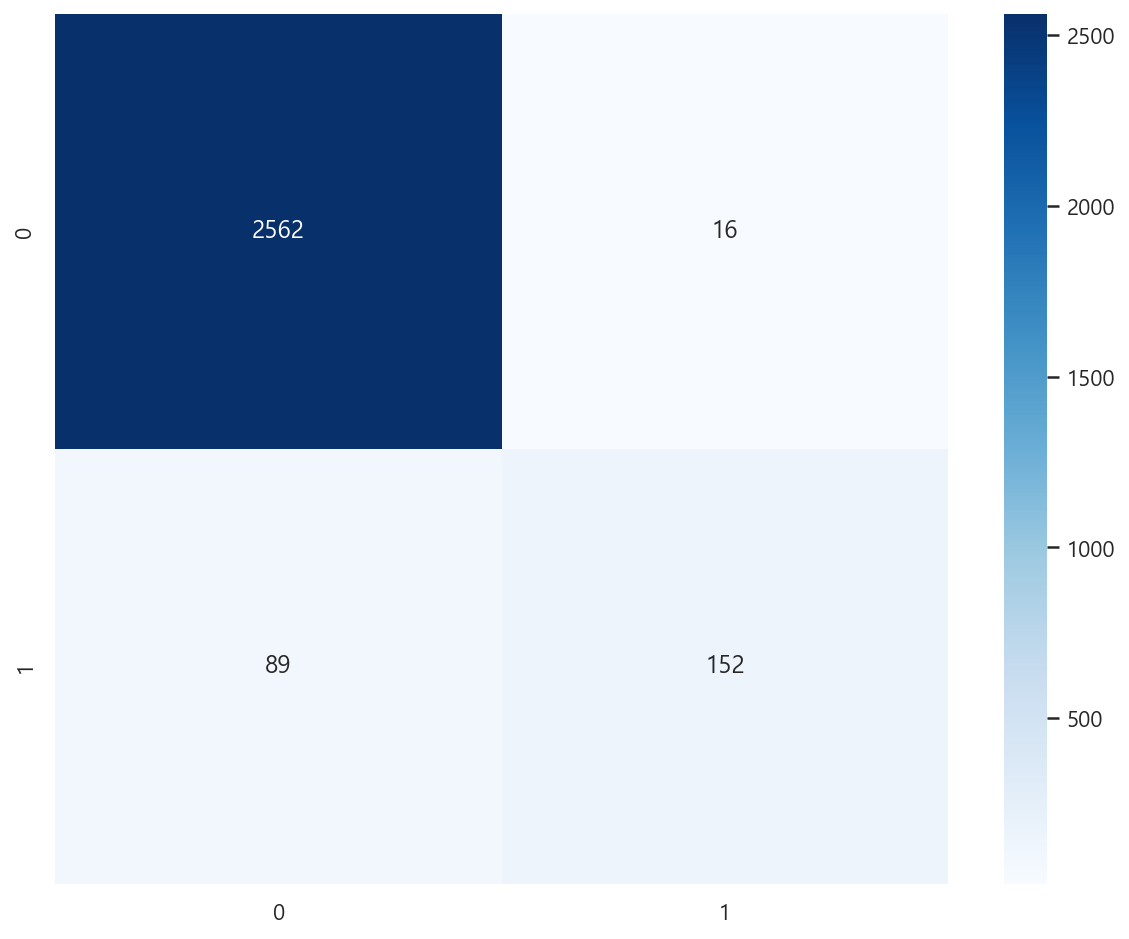


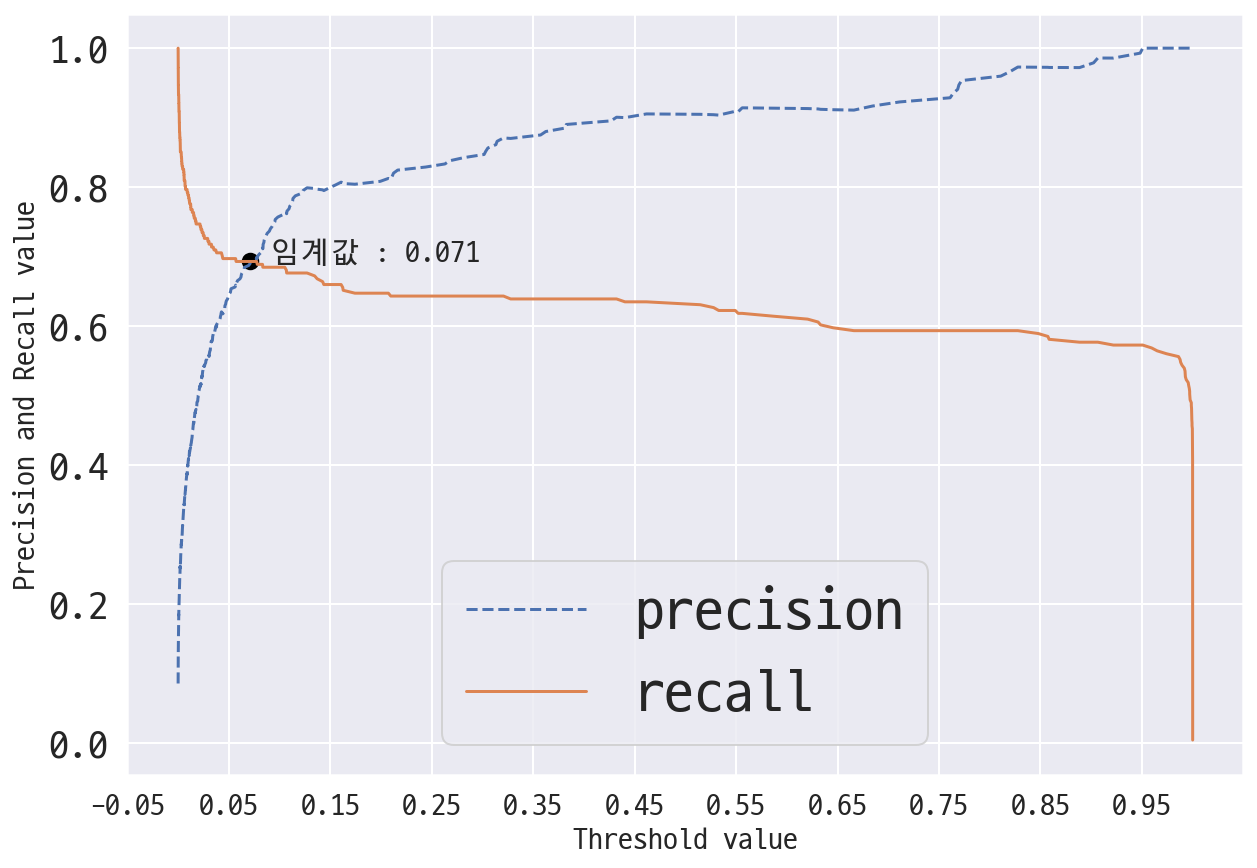


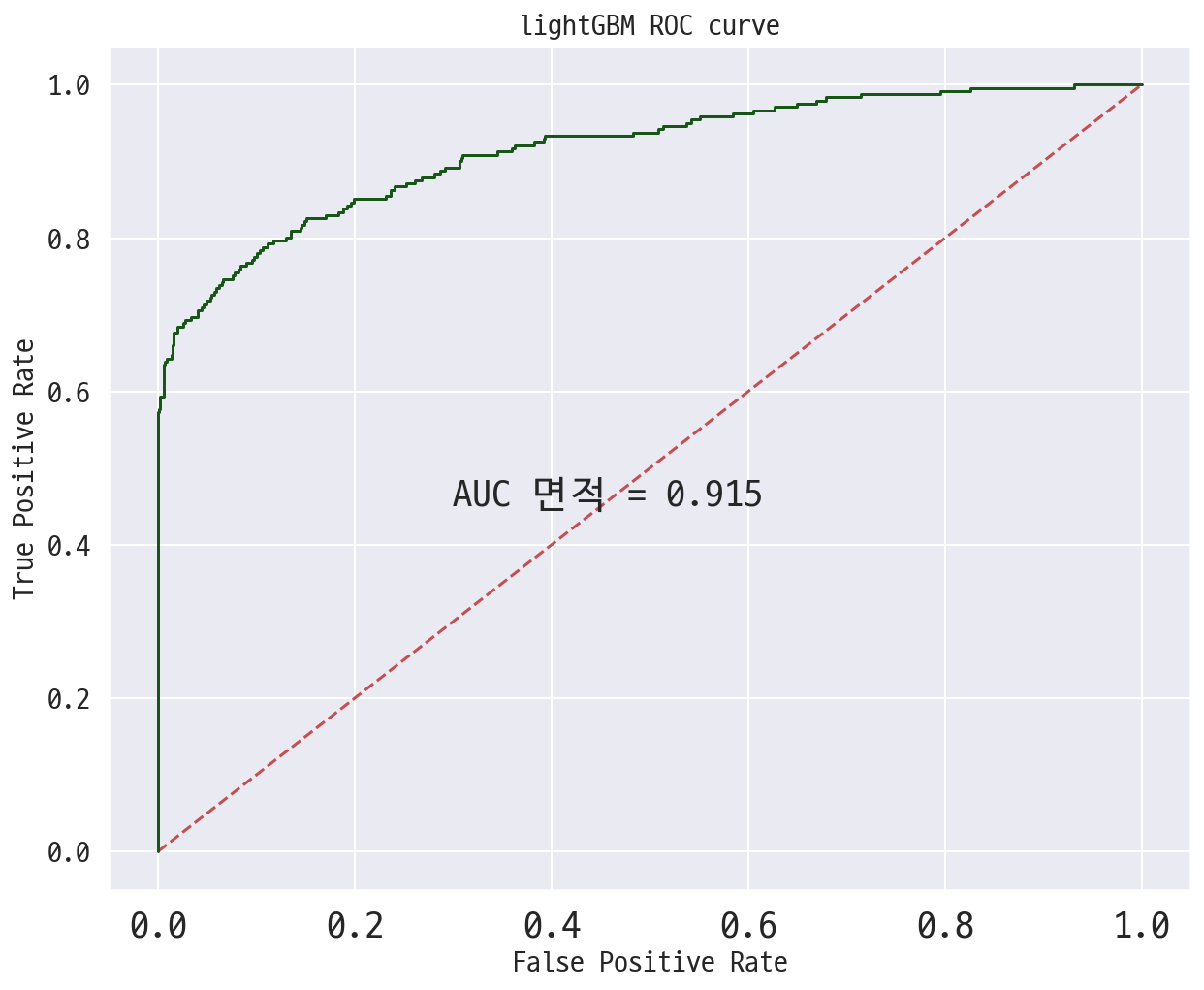


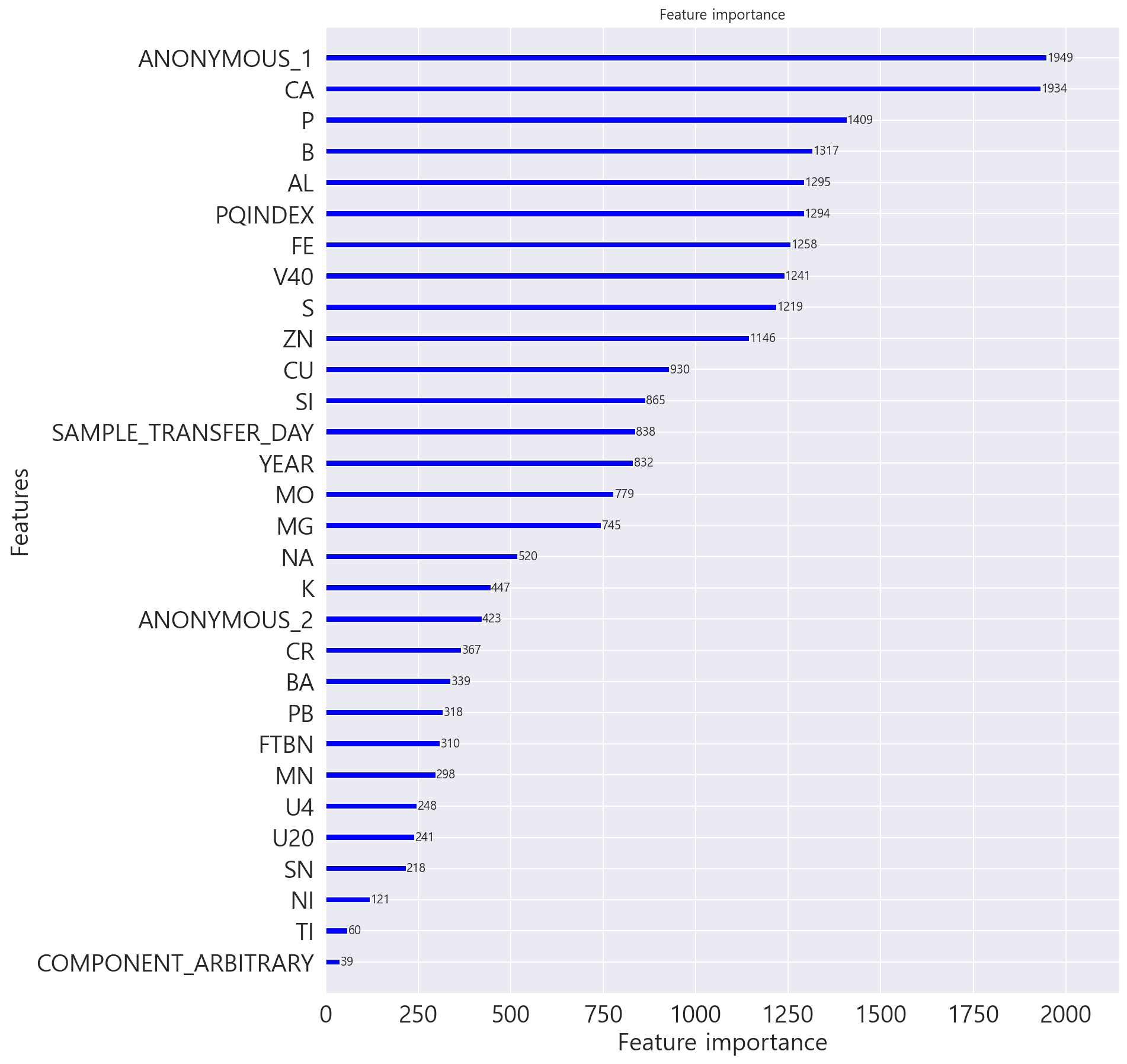


1. **log+ Over Sampling -> HyperOpt + LightGBM / AUC : 0.915**

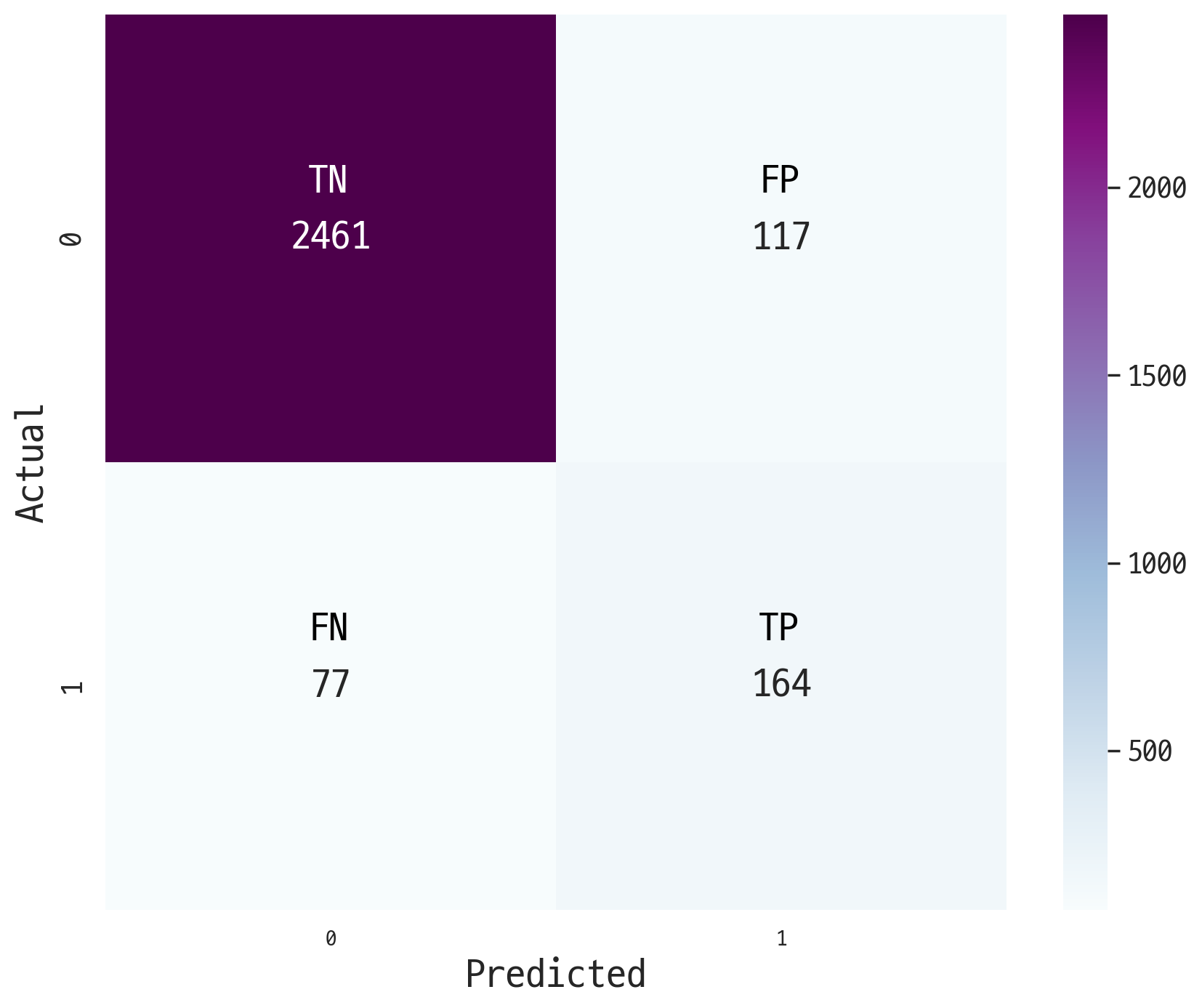


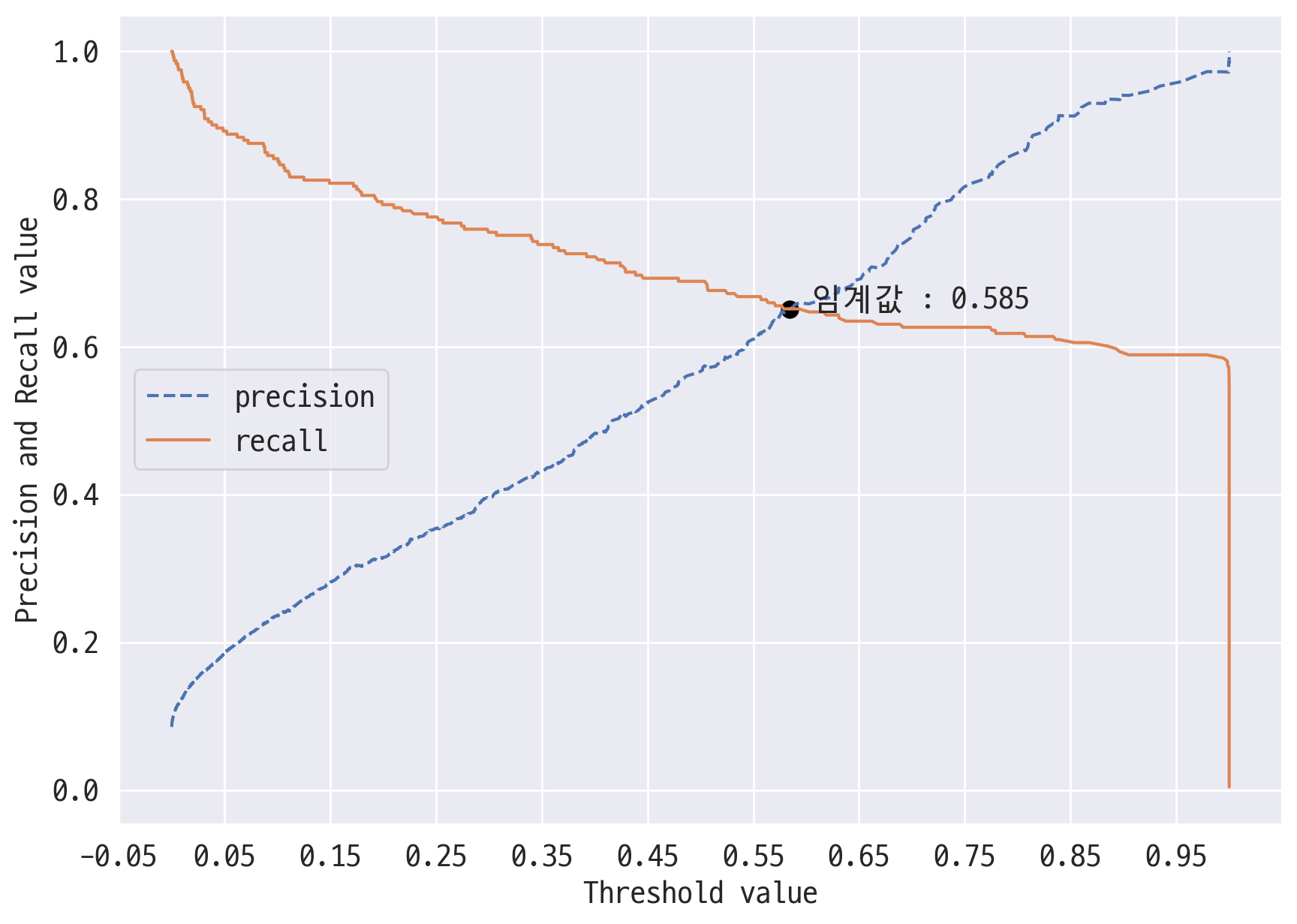


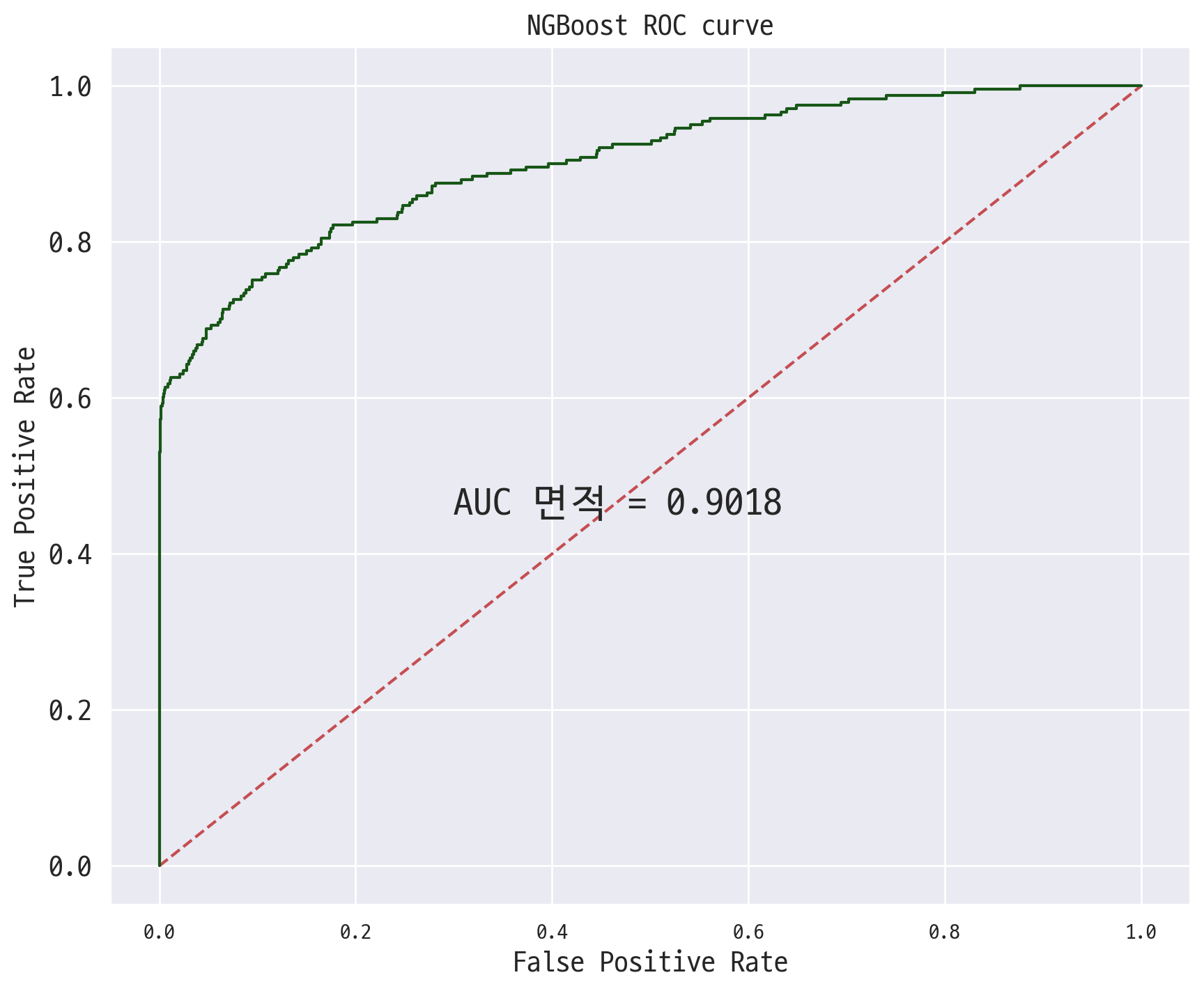




1. **log+ Over Sampling -> GridSearchCV + NGBoost / AUC : 0.9018**

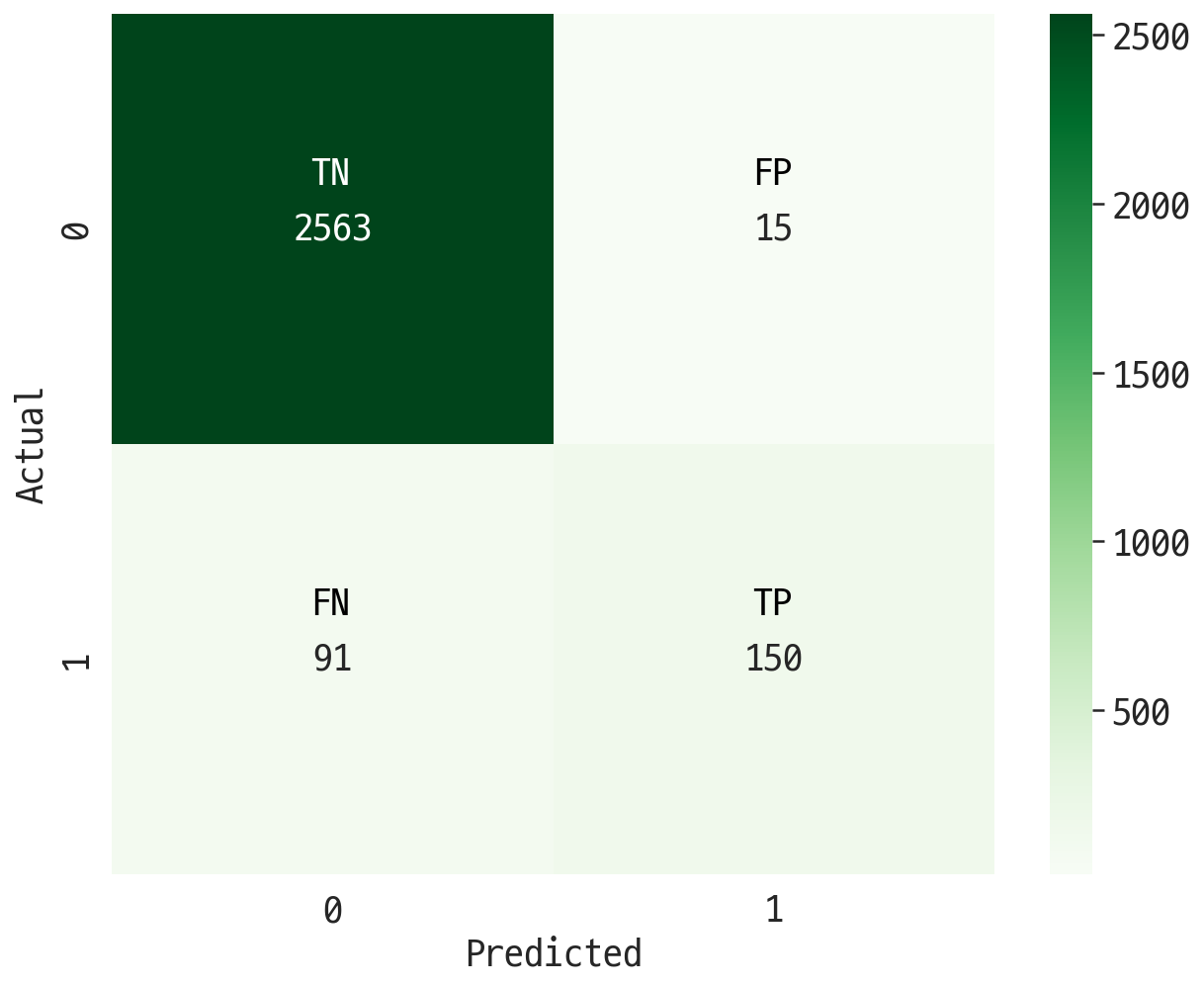
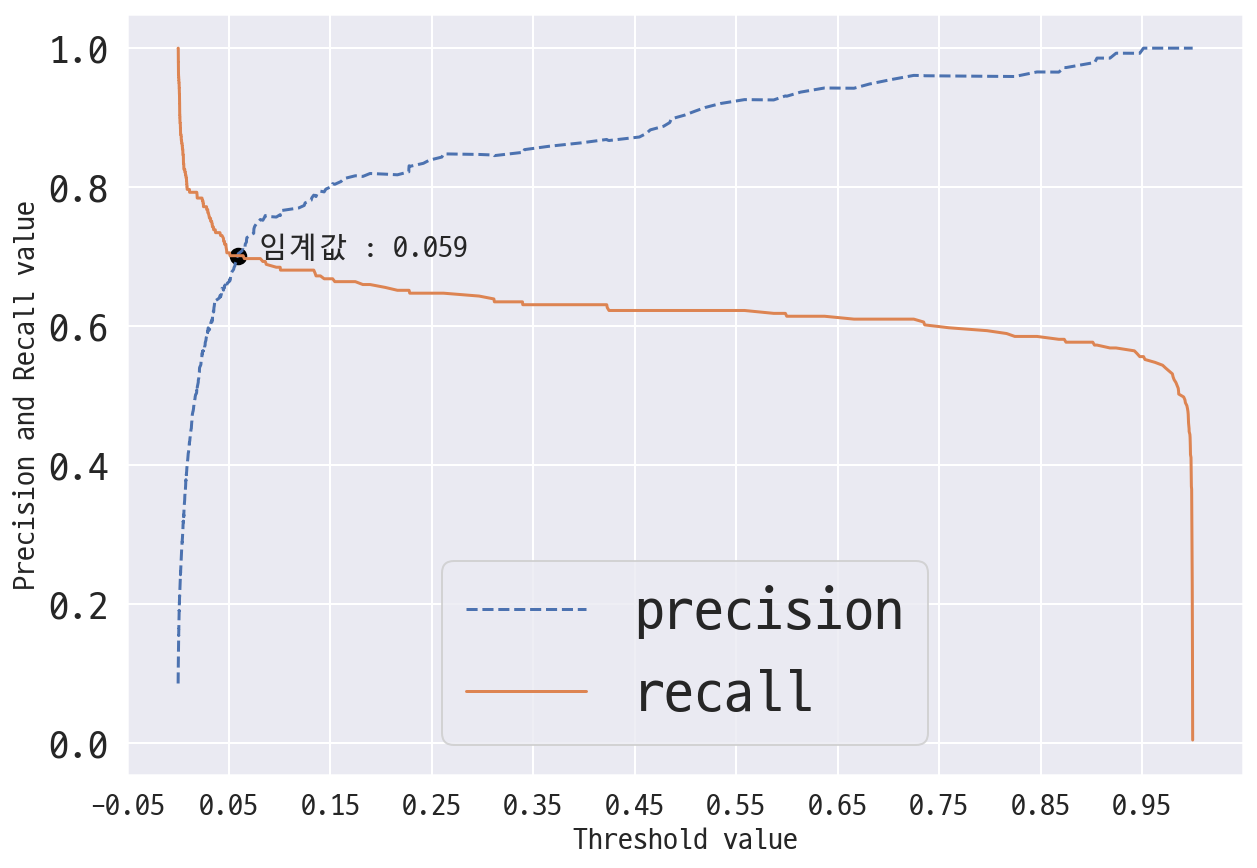


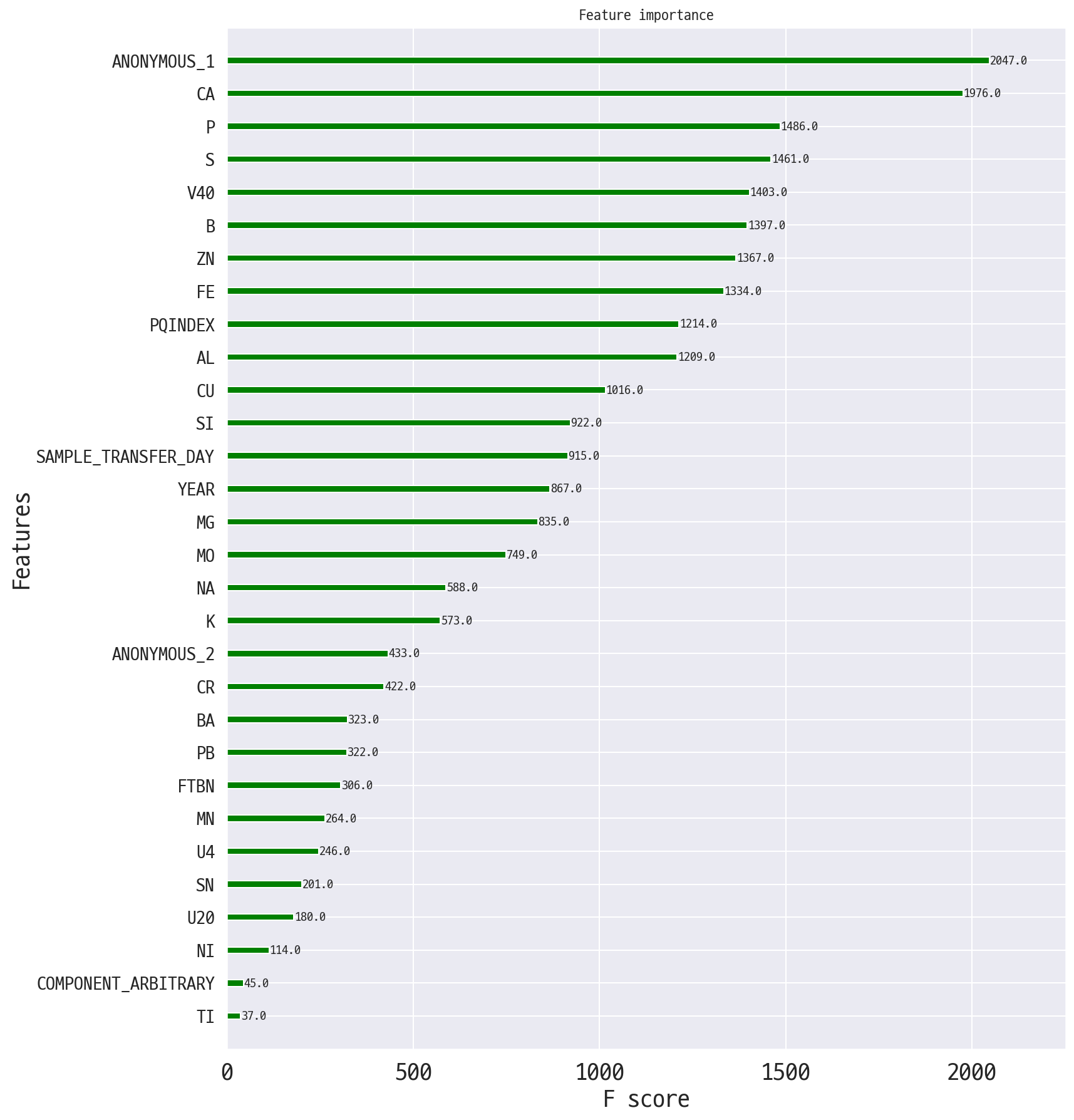
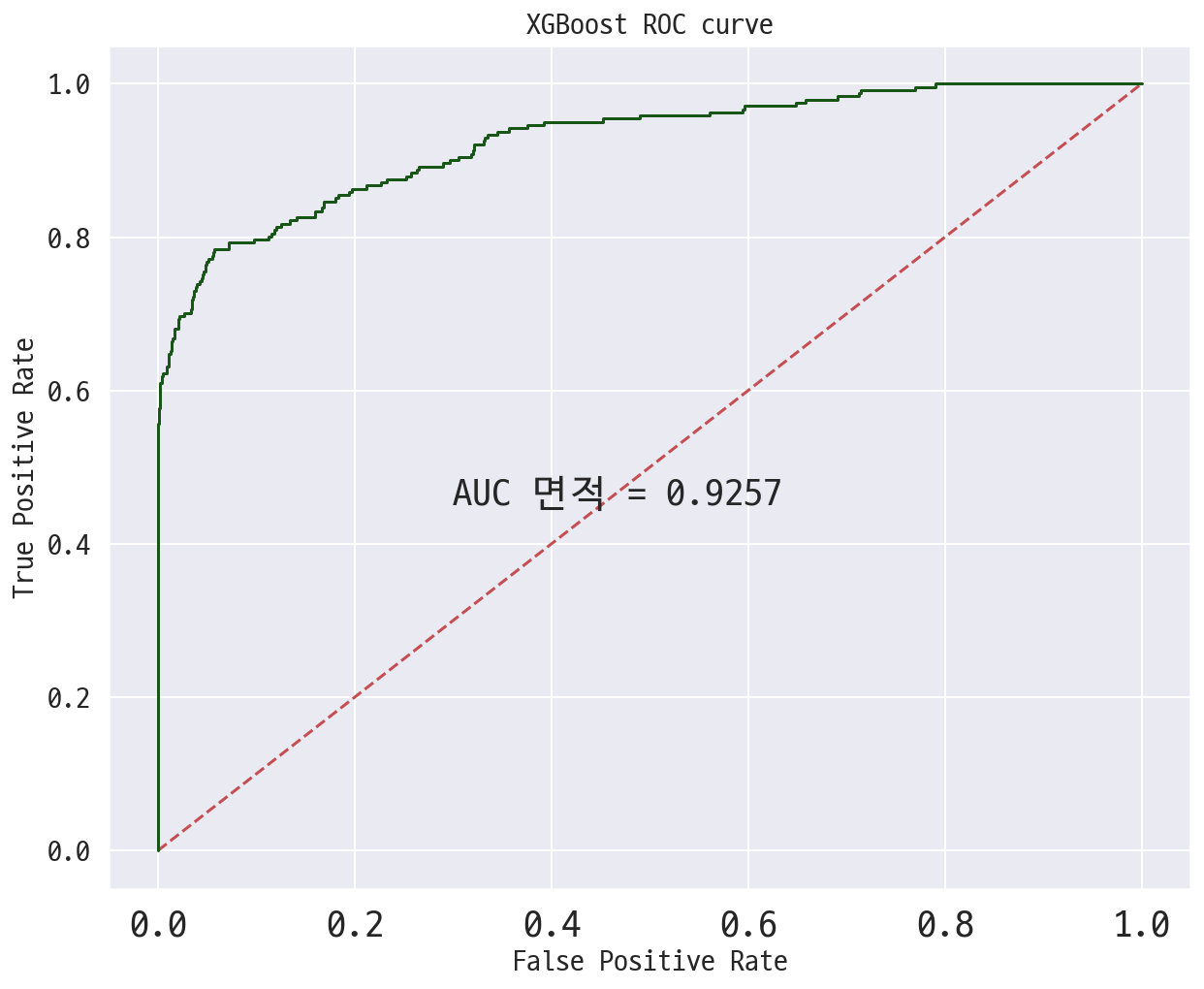




**-> NGBoost에는 feature\_importance의 결과값이 존재하지 않는다.**

**4. 프로젝트 결과 및 진행상황 -1) 프로젝트 결과**

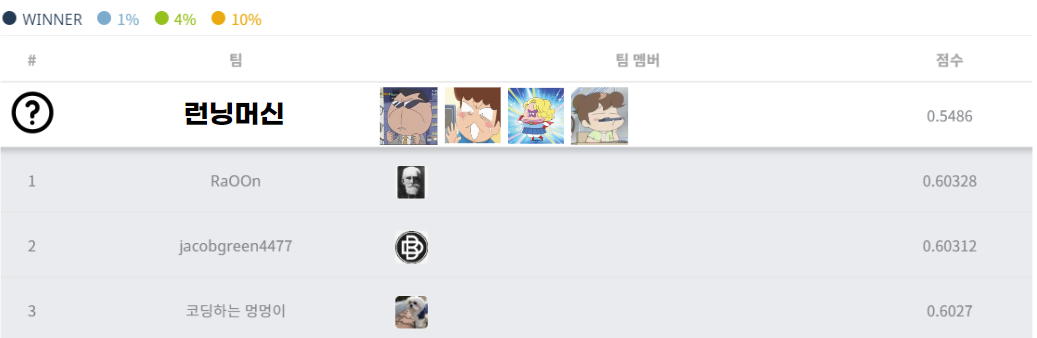




**-> 가장 높은 평가점수를 기록한 모델**

**: Yeo-Johnson + Over Sampling -> XGBoost / AUC : 0.9257**

**4. 프로젝트 결과 및 진행상황 -2) DACON 진행상황**



**-> 추후 전처리 및 알고리즘을 보완할 예정이다.**

**4. 프로젝트 결과 및 진행상황 -3) 프로젝트 소감**

1. **강보원**

**: 처음 머신러닝을 배울 때는 수업에 따라가기 바빴는데, 프로젝트를 진행하며, 분류에 필요한 알고리즘이 무엇인지 직접 찾고 공부해보는 과정을 거치면서 머신러닝이 친근해지기 시작했다. 프로젝트를 수행하면서 어떤 방법으로 전처리를 해야할지에 대해 고민하는 시간이 길었다. 도메인에 대한 분석을 통해 전처리에 대한 방향을 잡아갈 수 있었고, 전처리를 한 후, 결과치가 더 나아져서 전처리를 잘했구나라는 안도감이 들었다. 또한, 처음에는 GridSearch를 이용해서 알고리즘을 실행시켰는데 시간이 많이 오래 걸렸고, 이후 HyperOPT를 채택해서 실행했다. HyperOPT를 이용하면서 하이퍼 파라미터에 대해서 공부하는 시간을 갖게 되어서 좋았다. 프로젝트가 끝난 후에도 여러 모델에 대한 공부를 더 해나가고 싶다.**

1. **김동률**

**: 머신러닝을 직접 돌려볼 때에는 잘 이해가 되지 않는 부분이 많았다. 물론 쉽지는 않았다. 너무 오래걸려서 짜증이 날 때도 있었고, 수많은 오류에 빠져 헤어나오지 못할 때에는 조용히 노트북을 닫아버린적도 있었다. 하지만, 위에서 언급했던 것 처럼 많은 경우의 수를 두고 계속해서 알고리즘을 실행하다보니 조금씩 이해되고 깨우쳐가는 쾌감이 있었다. 그리고 팀원들과 같이 프로젝트를 진행하면서 즐겁게 그리고 서로를 이해하며 도우면서 보냈던 시간이 굉장히 보람되게 느껴지는 시간이었다.**

1. **박혜빈**

**: 지금까지 했던 수업들 중에 가장 어려운 시간이라 생각해서 그런지 프로젝트를 시작할 때에도 기대보다 걱정이 앞섰다. 수업도 따라가기 벅차게 느껴졌는데 실제로 해볼 수 있을까 하는 생각이 더 들었기 때문이다. 하지만 프로젝트는 내 예상과 달리 훨씬 즐거웠고, 이후의 과정들에 조금 더 기대감을 가지게 되었다. 새로운 목표보다 학습을 주된 목표로 프로젝트를 진행하기로 결심하고 난 뒤에는 모든 게 어려웠다. 도메인에 대해 이렇게까지 공부를 해본 적이 있었는지, 전처리를 이정도까지 처리해야 하는지, 모델을 선정하는 과정까지 정말 한순간도 쉽지 않았다. 하지만, 그 덕분에 잊혀지지 않을 순간을 만들었던 시간이여서 정말 뜻깊었다. 프로젝트는 이렇게 끝나더라도 이후에 공모전을 더 진행하게 됐는데 끝까지 좋은 결과를 만들어가고 싶다.**

1. **송승우**

**: 프로젝트 기간 동안 많이 성장되었다고 생각한다. 내가 알고 있는 모든 기술을 모르는 기술을 사용했고, 점수가 조금씩 올라갈 때마다 마음이 떨렸고 흥분됐다. 그만큼 재미있게 보내서 시간이 빨리 가는 것을 몰랐다. 마치 어제 팀을 정하고 오늘이 끝이다라는 느낌이 들었다. 그래서 더 많은 학습하고 적용하고 싶은 기술들이 많았을 만큼 프로젝트 기간이 더 길었으면 좋겠다는 아쉬움이 있었다. 처음에 공모전에 참가 했을 때 평가지표 점수가 올라가지 않았고, 이번 프로젝트 기회에 성장한 바탕으로 공모전에서도 잘할 수 있기를 기대한다. 그 덕에 우리 팀명 '러닝머신' 처럼 잘 뛰었다라고 생각한다.**

**5. 참고 자료 및 사이트**

1. **참고 자료**

**1. 기계 상태 진단을 위한 인-라인형 오일 모니터링 스마트 센서**

**(날짜 : 2022년 11월 24일)**

**https://koreascience.kr/article/JAKO200827651798049.pdf**

1. **참고 사이트**

**1. DACON 건설기계 오일 상태 분류 AI 경진대회 (날짜 : 2022년 11월 24일)**

**https://dacon.io/competitions/official/236013/overview/description**

**2. 오일 샘플 테스트 - 작동 방식 (날짜 : 2022년 11월 24일)**

**https://www.oillab.co.nz/how-it-works/your-results/**