33기 방학세미나

3팀
이정민
황호성
김형석
송다은
이현진
윤여원

INDEX

1

데이터 확인 및 EDA

1 데이터 확인 및 EDA

Train data 구조

```
train.info
✓ 0.0s                                                                    Python
<bound method DataFrame.info of          Id          0          1          2
0         0 -27.419000 -25.272000 -25.474000 -22.805000 -24.078000
1         1  -8.826692   8.233792   0.380527 -15.821410 -31.288668
2         2  -0.073413  -0.063125  -0.081681  -0.090575  -0.078336
3         3 -13.489000 -16.472000 -21.402000 -23.426000 -26.639000
4         4  -0.055060 -26.949334 -16.005953  20.003442 -21.654284
...     ...        ...        ...        ...        ...        ...
12995 12995 -25.626926 -29.257572 -19.743892 -35.944453 -32.077512
12996 12996  15.818102   0.637107   6.439438   2.253320 -20.231920
12997 12997  -0.311896   0.116477  -0.223599  -0.882955  -1.623315
12998 12998  -0.470953  -0.483925  -0.415385  -0.582755  -0.661472
12999 12999 -61.260582 -71.329726 -62.508748 -66.039632 -67.934239

              5          6          7          8  ...        494        495  \
0     -22.308000 -19.020000 -15.117000 -20.164000  ... -37.308000 -35.133000
1     -49.456777 -33.808110 -31.886476 -49.071073  ... -63.940770 -77.129207
2      -0.062810  -0.078310  -0.093963  -0.084305  ...  -0.247620  -0.235650
3     -24.816000 -26.520000 -24.136000 -27.843000  ... -24.373000 -21.811000
4     -33.971452 -22.177112  -4.009821 -22.374219  ...  -7.711350 -17.006321
...          ...        ...        ...        ...  ...        ...        ...
12995 -22.057527 -18.376308 -28.376480  -4.777856  ... -32.245758 -30.926059
12996 -21.554735  -6.585616   0.292068 -10.118205  ... -20.490845 -28.036757
12997  -2.999989  -3.613498  -3.532261  -3.129042  ...  -3.291602  -2.971378
12998  -0.459698  -0.566389  -0.583768  -0.457157  ...  -0.656539  -0.596851
12999 -56.107527 -65.957808 -56.101276 -66.634889  ... -48.330941 -46.280700
...
12997      1
12998      0
12999      0

[13000 rows x 505 columns]>
```

1. ‘0’~’499’의 **시계열** 데이터

2. **범주형** 자료 ‘S/N’, ‘Country’, ‘Label’ 등

3. **Masked data**

1 데이터 확인 및 EDA

표기법 통일

Country

'중국' '美国' 'china' '中国' 'Korea' 'america' 'USA' '미국' 'U.S.' '대한민국' '韩国' 'South Korea' '한국'

S/N

'PSCG-68053' 'PSCG-79993' 'PSFT-11445' ... 'PSFT-34971' 'PSFT-89252' 'PSCG-74202'

‘Country’, ‘S/N’ 의 표기가 다수 존재

표기명 통일 후 분석할 필요가 있음

1 데이터 확인 및 EDA

표기법 통일

Country

'중국' '美国' 'china' '中国' 'Korea' 'america' 'USA' '미국' 'U.S.' '대한민국'
'韩国' 'South Korea' '한국'

'china', 'korea', 'america' 로 표기법 통일

1 데이터 확인 및 EDA

표기법 통일

S/N

'PSCG-68053' 'PSCG-79993' 'PSFT-11445' ... 'PSFT-34971' 'PSFT-89252' 'PSCG-74202'

-이후 숫자를 제거하여 'PSCG', 'PSFT' 로 표기법 통일

PSFT보다 PSCG의 절댓값이 크다는 것 또한 확인!

1 데이터 확인 및 EDA

표기법 통일

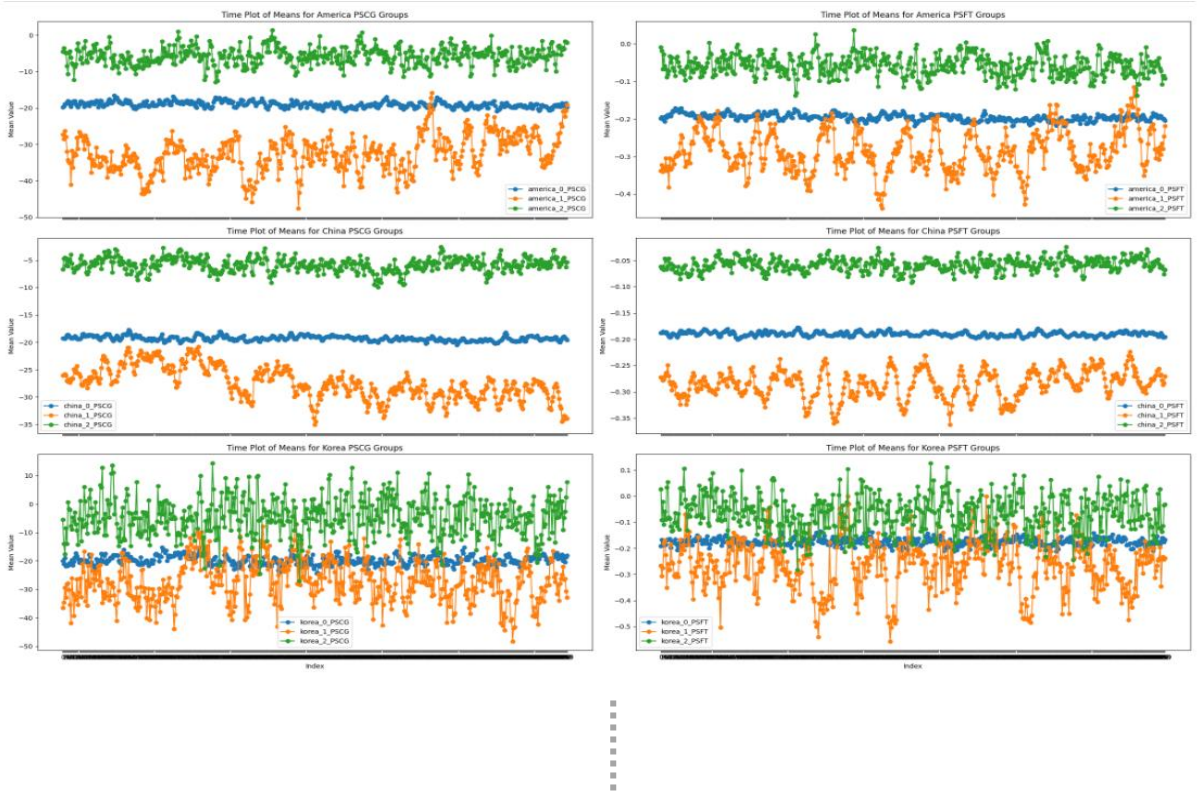
Year

'1990', '1991', '1992', '1993', '1994', ···, '2017', '2018', '2019'

90년대, 00년대, 10년대로 범주를 나누어 인코딩

1 데이터 확인 및 EDA

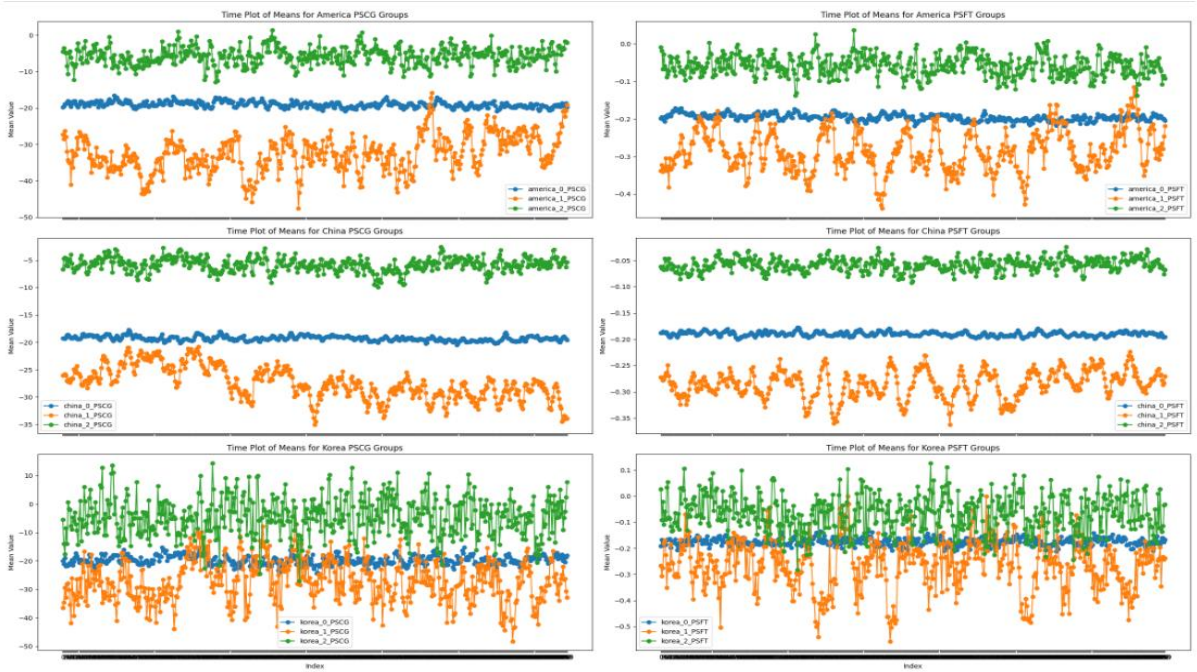
변수별 차이



'Label' '0', '1', '2'에 따라서 **평균값이 명확히 나뉨**

1 데이터 확인 및 EDA

변수별 차이

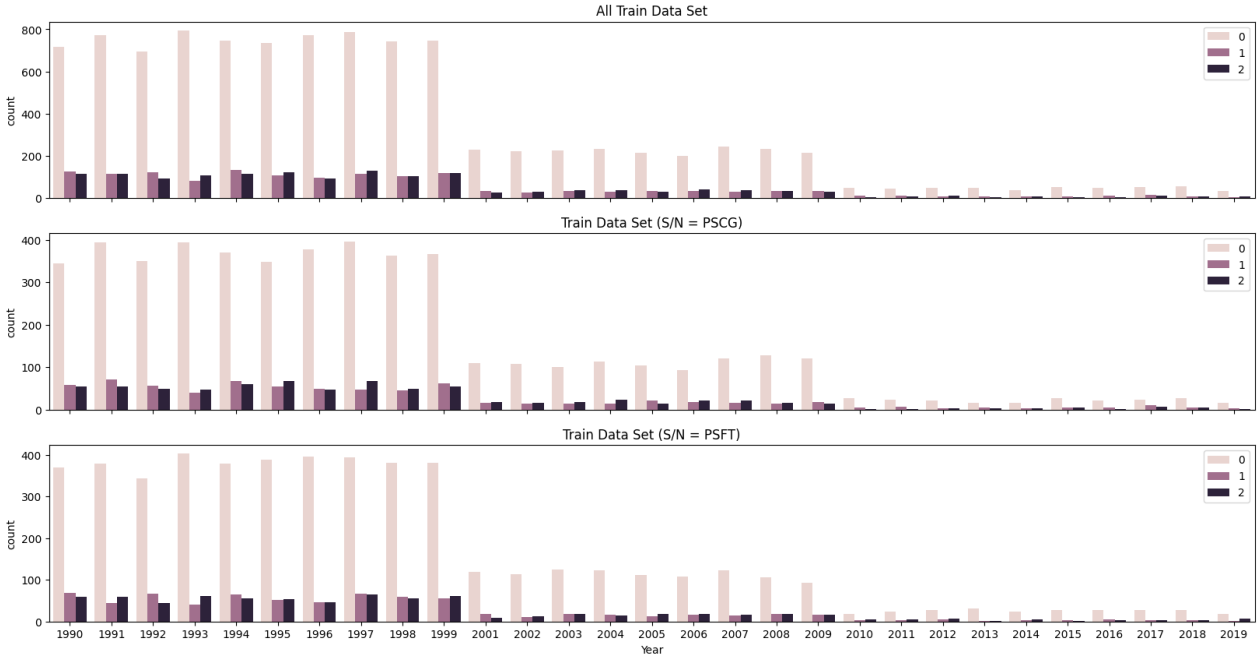


이후 모델링 시 scaling을 하면 이 특성이 무시되어 성능 하락 위험 有

1 데이터 확인 및 EDA

변수별 차이

Label ‘0’이 압도적으로 많으며,
‘Year’의 경우 **2000년 이전**이 이후보다 많음

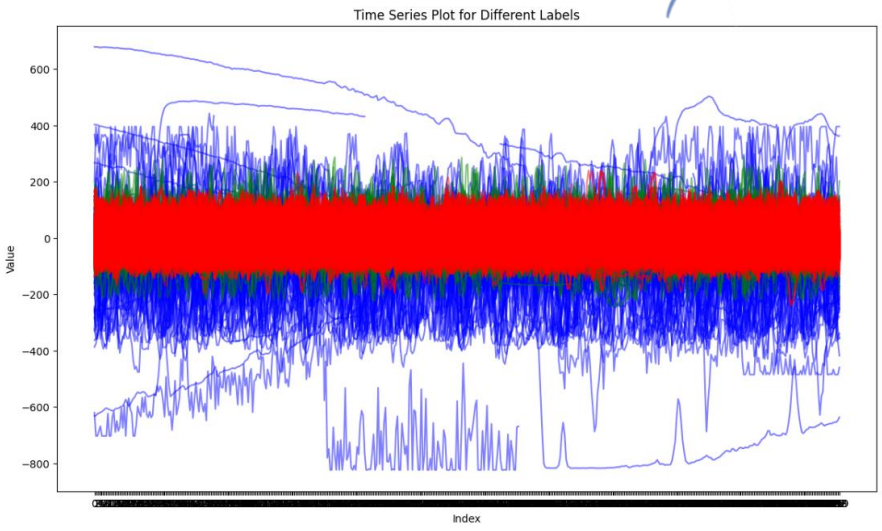


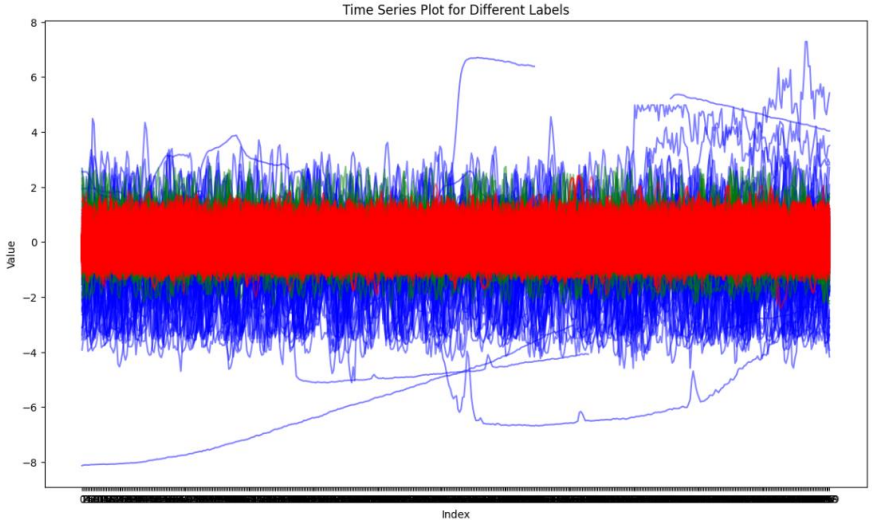
‘Label’ 과 ‘Year’ 별 불균형 존재

1 데이터 확인 및 EDA

변수별 차이

파란색 Label '1'의 변동성이 두드러짐

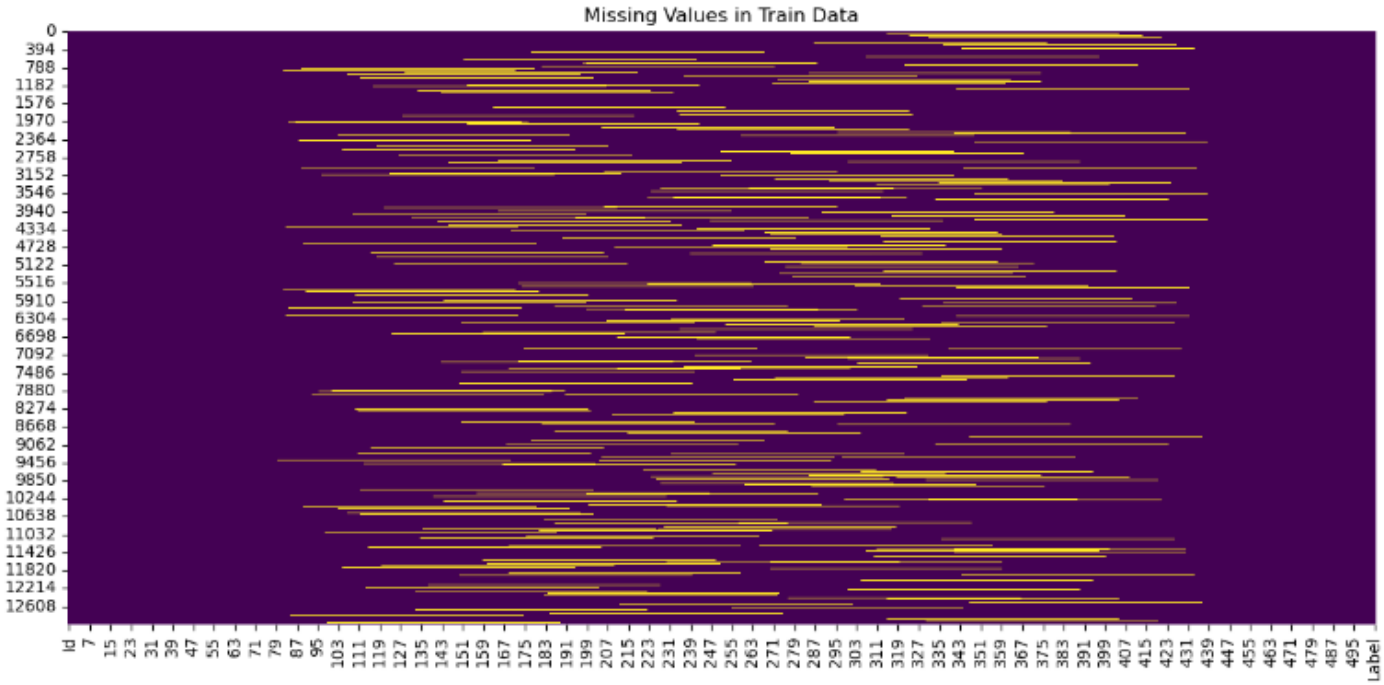




'Label' 별 변동성이 존재함을 확인함 '1'의 변동성이 큼

1 데이터 확인 및 EDA

결측치 확인



✓ trian 데이터에 587880개의 결측치 존재 확인

1 데이터 확인 및 EDA

결측치 확인


```
# 테스트 셋 내 결측치 위치 확인

na_positions = np.where(df_test.isna())
na_indices = list(zip(na_positions[0], na_positions[1]))
```

결측치 위치 확인 결과,

'80'~'438' 사이 90개씩 연속적으로 분포함을 확인!

✓ trian 데이터에 587880개의 결측치 존재 확인

2

데이터 전처리

2 데이터 전처리

결측치 보간

선형 보간

두 인접한 데이터 점 사이를 직선으로 연결하여 중간의 결측치를 예측하는 방법

스플라인 보간

데이터 점들을 다항식으로 연결하여 결측치를 예측하는 방법

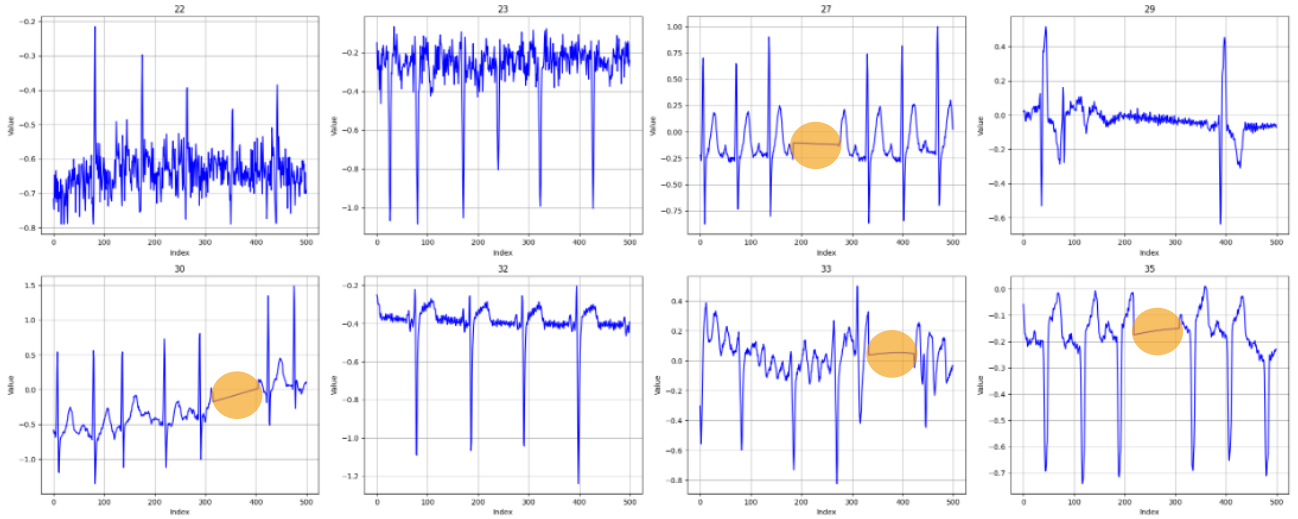
FFT 보간

신호를 주파수 도메인으로 변환하여 결측치를 채우는 방법

결측치 보간

선형 보간

두 인접한 데이터 점 사이를 직선으로 연결하여 중간의 결측치를 예측하는 방법



변동성을 전혀 반영하지 못함

2 데이터 전처리

결측치 보간

선형 보간

두 인접한 데이터 점 사이를 직선으로 연결하여 중간의 결측치를 예측하는 방법

다중 로지스틱 회귀

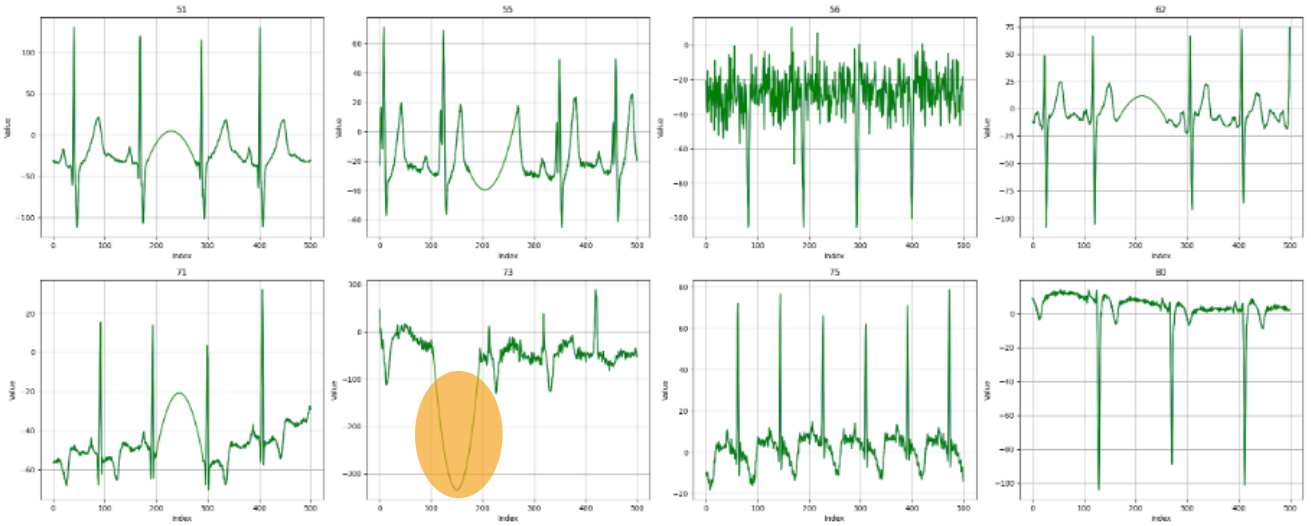
여러 독립 변수를 사용해 범주형 종속 변수의 발생 확률을 예측하는 통계 기법

첫 점수로 Public 893점이라는 낮은 점수…

결측치 보간

스플라인 보간

데이터 점들을 다항식으로 연결하여 결측치를 예측하는 방법

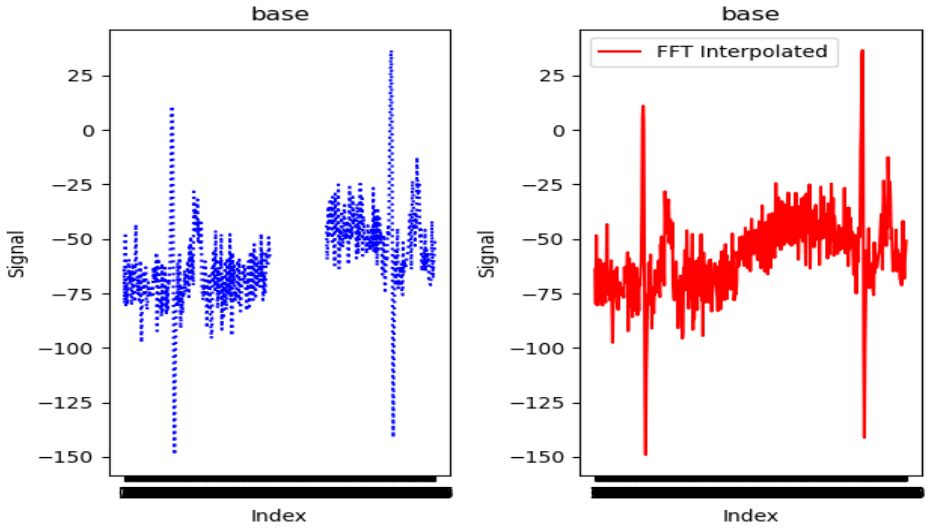


변동성 일부 반영하지만 가끔 과도하게 추정

결측치 보간

FFT 보간

신호를 주파수 도메인으로 변환하여 결측치를 채우는 방법



과도하지 않게 노이즈를 추정하고 일부 추세/계절성을 반영함

2 데이터 전처리

결측치 보간

FFT 보간

Seasonal smoothing을 이용하고자 했으나 일부 그룹에서 주기 확인 어려움

신호를 주파수 도메인으로 변환하여 결측치를 채우는 방법

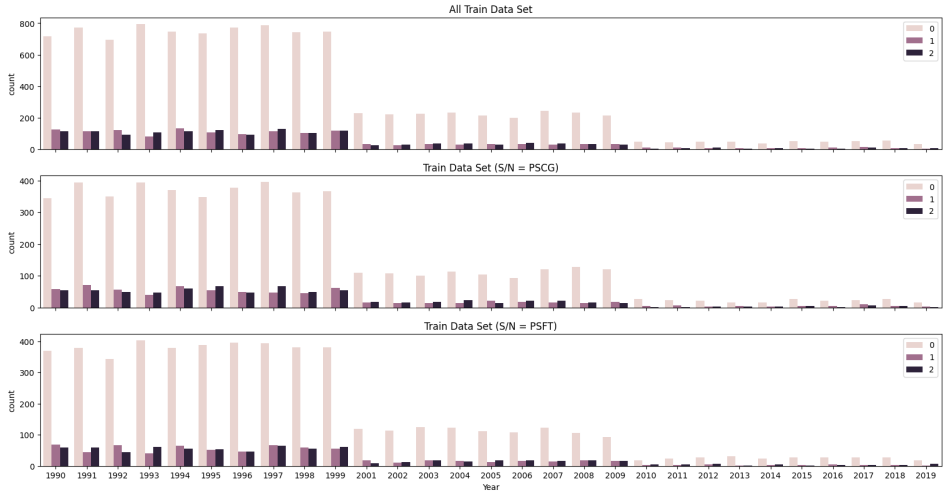
결국 고속 푸리에 변환 사용🥳!

과도하지 않게 노이즈를 추정해내서 사용하기로 결정!

클래스 불균형 해소

클래스 불균형

각 수준(클래스)에 따른 관측치 개수의 차이가 큰 경우



'Label' 과 'Year' 변수에 불균형이 존재함

클래스 불균형 해소 | Oversampling

오버 샘플링(Over sampling)

소수의 클래스를 다수의 클래스에 맞추어 관측치를 증가시키는 방법

SMOTE(Synthetic Minority Over-sampling Technique)

소수 범주의 데이터를 가상으로 만들어내는 방법

클래스 불균형 해소 | Oversampling

오버 샘플링(Over sampling)

소수의 클래스를 다수의 클래스에 맞추어 관측치를 증가시키는 방법

Random Over Sampling

랜덤으로 소수 클래스의 데이터를 복제하는 방법

클래스 불균형 해소 | Undersampling

언더 샘플링(Under sampling)

소수의 클래스는 변형하지 않고,

다수의 클래스를 소수의 클래스에 맞추어 관측치를 감소시키는 방법

Random Under Sampling

랜덤으로 다수의 클래스에 해당하는 데이터를 제거하는 방법

2 데이터 전처리

scaling

스케일링(scaling)

데이터의 각 특징을 일정한 범위로 변환해 모델의 학습 성능을 향상시키는 기법

```
Weighted score: 6433
Weighted Acc: 0.7516943211030614

Weighted score: 5298
Weighted Acc: 0.6426491994177583
```

EDA에서 예상했던 것처럼 스케일링 시 모델 성능이 하락함

3

모델링

3 모델링

모델 목록

모델링의 세계에 빠져드는 3팀.. 수많은 시행착오를 겪게되는데..

CNN	KNN	XGBoost	LightGBM

3 모델링

CNN

CNN

KNN

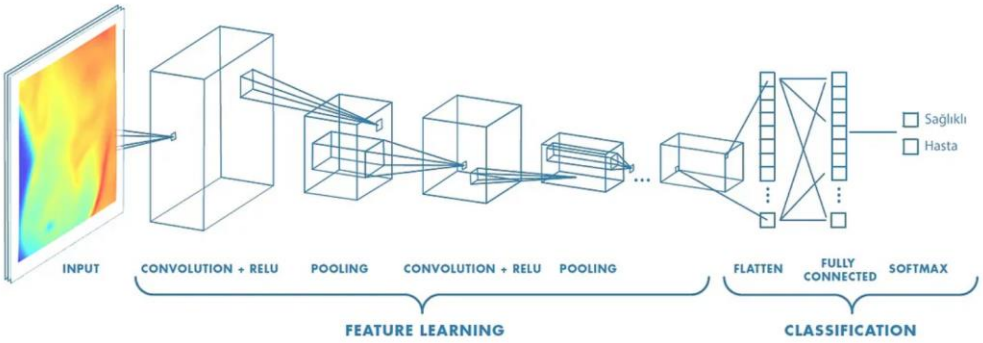
XGBoo

LightGBI

이튿날 케찹이가 꿈에 부풀어 찾아온 논문에서 말하길..

Classification of Time-Series Images Using Deep Convolutional Neural Networks

Nima Hatami, Yann Gavet, Johan Debayle



이미지나 시계열 데이터와 같은 격자 구조 데이터를 처리하고 특징을 추출하는 데 탁월한 심층 신경망

케찹이의 꿈… 딥러닝 마스터

3 모델링

CNN

CNN

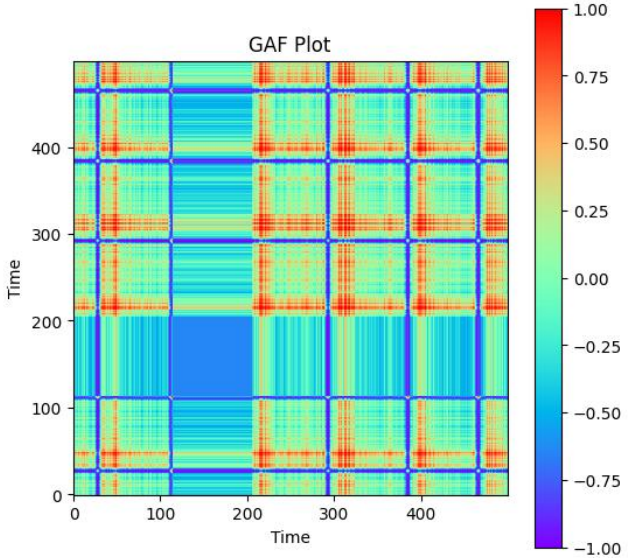
KNN

XGBoo

LightGBI

각고의 노력 끝에 이미지 변환에 성공했으나

성능이 그리 높지 않았음…



케찹이의 꿈… 딥러닝 마스터

3 모델링

KNN

CNN

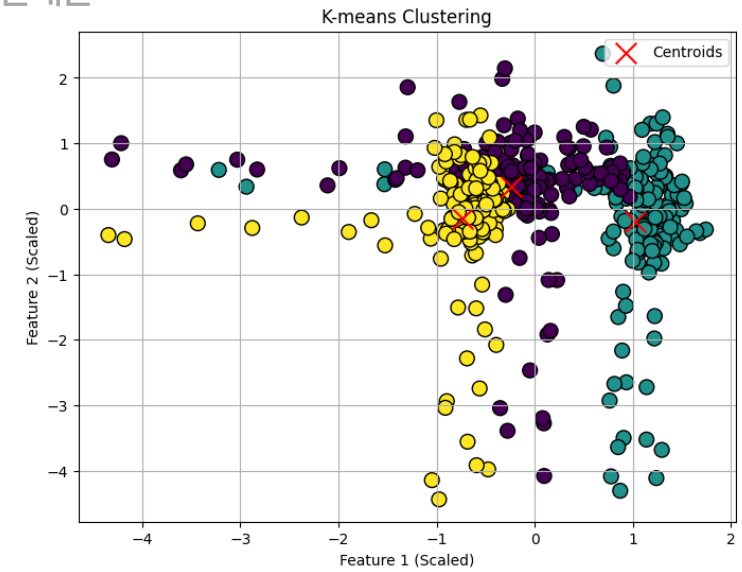
KNN

XGBoost

솔미도 많은 고민을 하였다..

LightGBM

원래는…



비지도학습인 K-means Clustering 이용 후

주어진 라벨과 얼마나 일치하는지 확인하고자 함

3 모델링

KNN

CNN

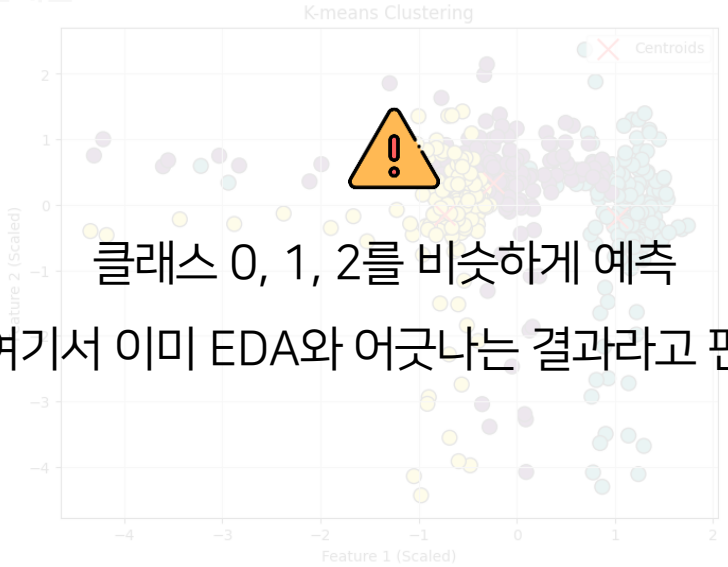
KNN

XGBoost

솔미도 많은 고민을 하였다..

LightGBM

원래는…



클래스 0, 1, 2를 비슷하게 예측

여기서 이미 EDA와 어긋나는 결과라고 판단

비지도학습인 K-means Clustering 이용 후

주어진 라벨과 얼마나 일치하는지 확인하고자 함

3 모델링

KNN

CNN

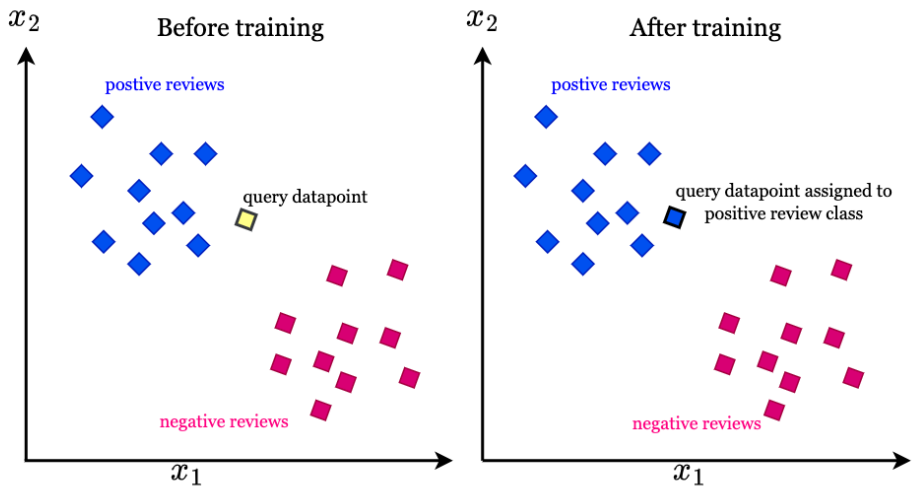
KNN

XGBoost

솔미도 많은 고민을 하였다..

LightGBM

새로운 데이터 포인트의 클래스를 결정하기 위해,
주어진 데이터 포인트와 가장 가까운 K개의 이웃 데이터
포인트를 참조하여 가장 많은 클래스를 선택하는 방식

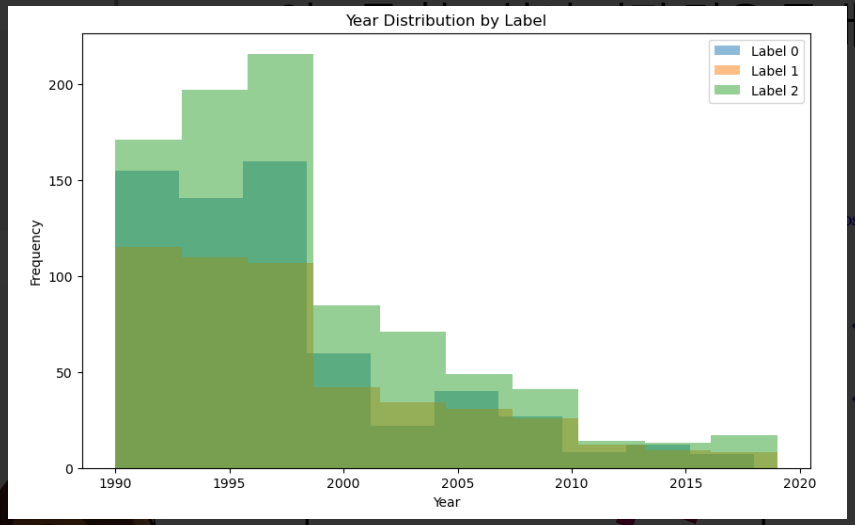


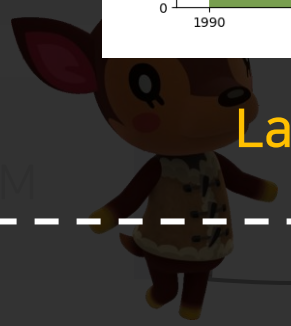
3 모델링

KNN

거리기반 알고리즘이라 scaling에 민감하게 반응함

데이터 특성 상 scaling 시 **특성이 제거**될 수 있음





Label 2가 과도하게 추정됨

XGBoost

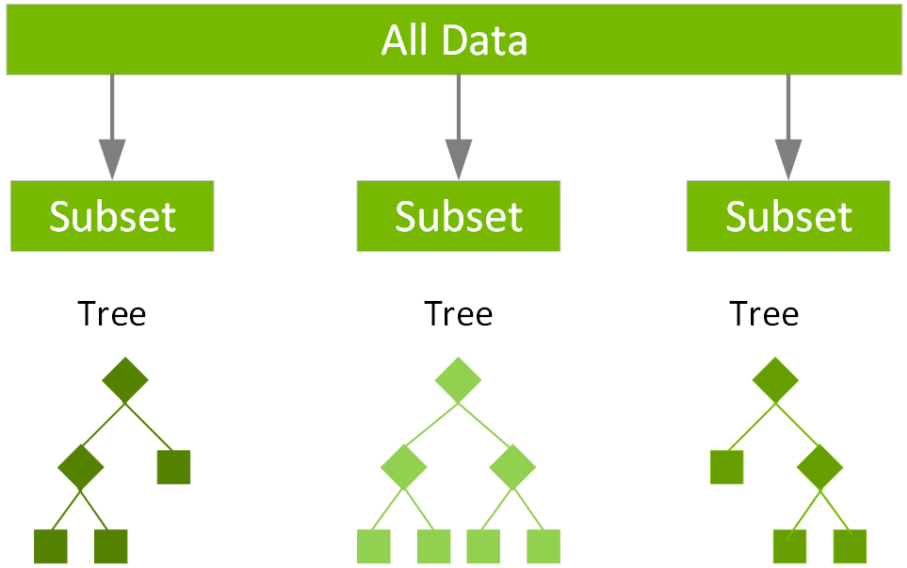
CNN

KNN

XGBoost

LightGBM

병렬 연산을 활용한 의사결정 나무 기반의 부스팅 알고리즘으로, 빠르고 효율적인 다중분류 모델 학습이 가능



LightGBM

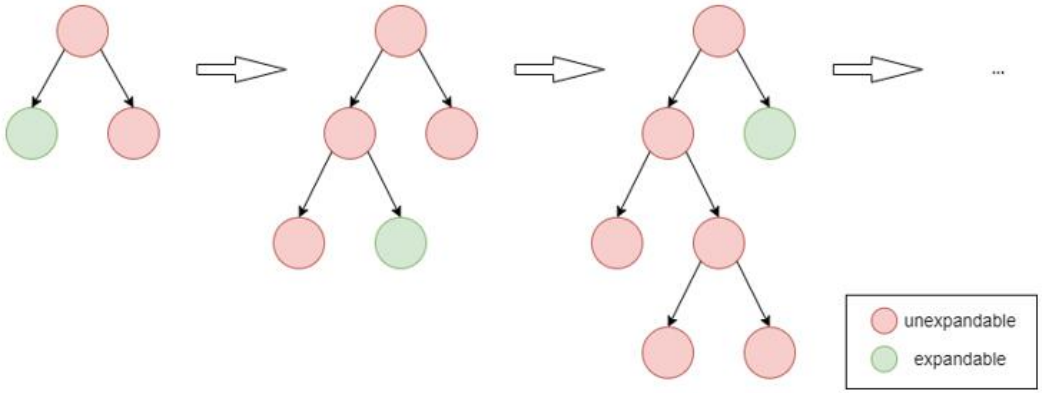
CNN

KNN

XGBoost

LightGBM

LightGBM은 대용량 데이터에 적합한 경량화된 그레디언트 부스팅 프레임워크로, 높은 성능과 빠른 학습 속도를 제공하여 다중분류 문제에 효과적



3 모델링

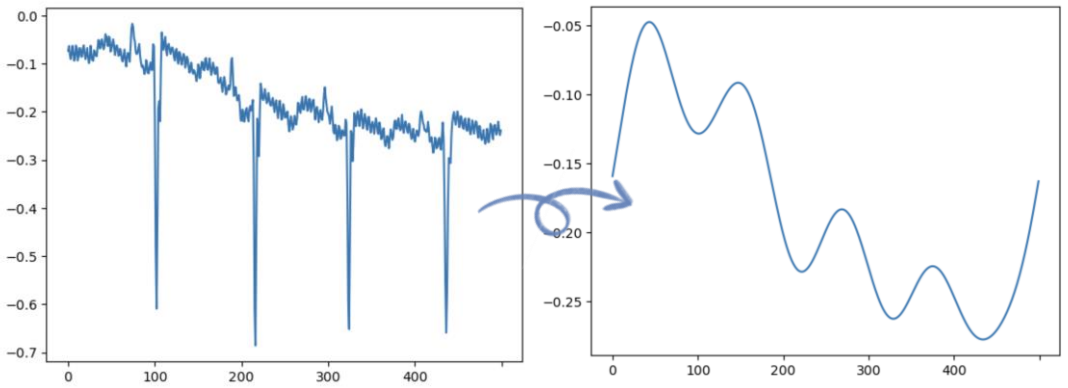
XGB/LGBM – 노이즈 제거

CNN

호성&여원&현진 앙상블

XGBoost

LightGBM



고속 푸리에 변환(FFT)를 이용한 노이즈 제거

개형상 데이터를 정확하게 추정하지 못함

하지만 모델의 accuracy와 weighted score는 올라감

과적합이 의심됨

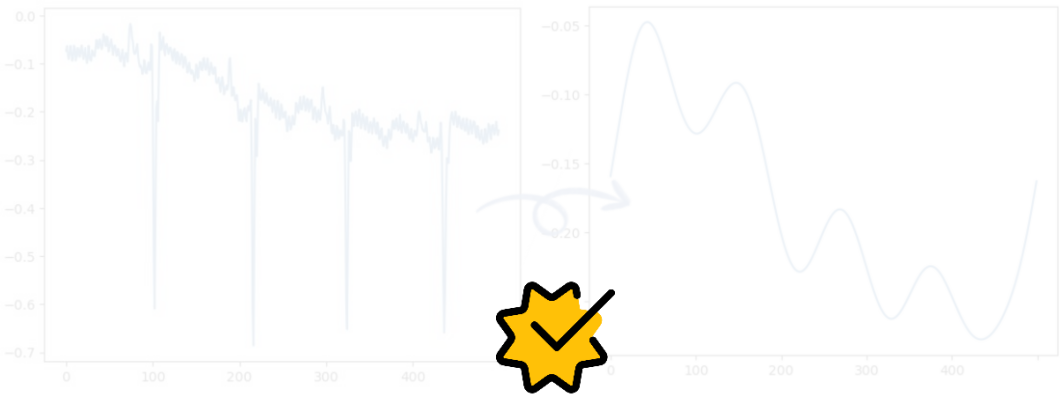
3 모델링

XGB/LGBM – 노이즈 제거

CNN

XGBoost

LightGBM



노이즈 제거 대신
파생변수를 추가적으로 고려하기로 함

3 모델링

XGB/LGBM – 샘플링

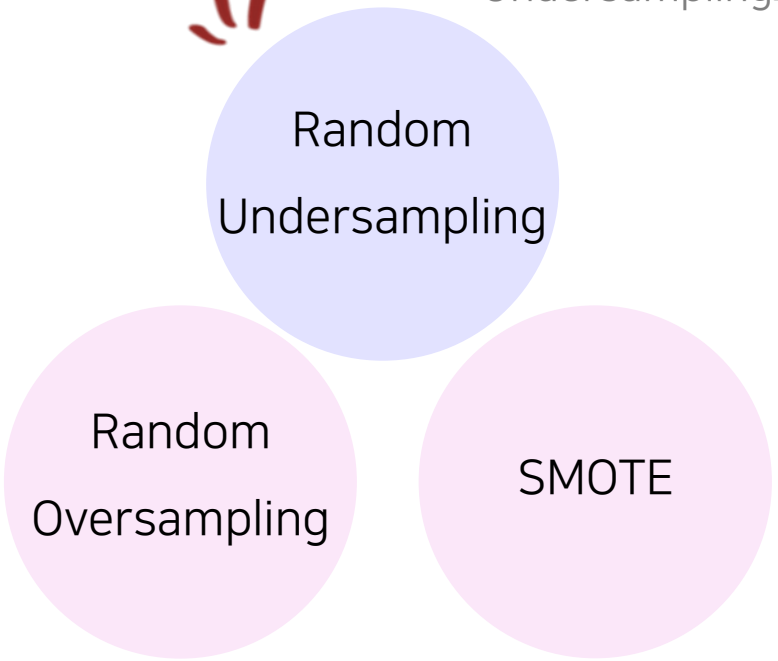
CNN

호성&여원&현진 앙상블

XGBoost

LightGBM

Undersampling으로 결정!



노이즈 제거 없이 진행해 Oversampling시
노이즈의 영향이 더 커진 것으로 판단됨

3 모델링

XGB/LGBM - 평가지표

CNN

호성&여원&현진 앙상블

XGBoost

LightGBM

Accuracy

전체 데이터에서 모델이 올바르게 예측한 샘플의 비율

모든 클래스의 예측 성능을 동일하게 고려

Weighted score

클래스별 정확도를 가중 평균하여 계산한 점수

precision 1, precision2

Label1과 2에 대해 모델이 해당 클래스로 예측한 샘플 중

실제로 해당 클래스인 샘플의 비율 TP / (TP + FP)

3 모델링

XGB/LGBM - 평가지표

CNN

호성&여원&현진 앙상블

XGBoost

LightGBM

Accuracy

전체 데이터에서 모델이 올바르게 예측한 샘플의 비율

모든 클래스의 예측 결과를 동일하게 고려

Class Weight가 크게 반영되므로

Label 1, 2의 정밀도를 높이기로 결정

클래스별 정확도를 가중 평균하여 계산한 점수

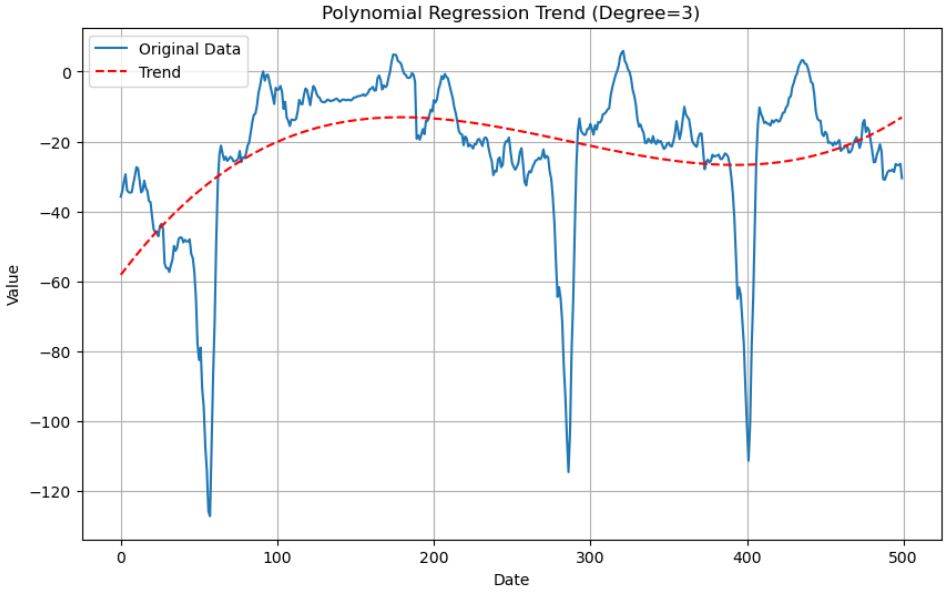
precision 1, precision2

Label1과 2에 대해 모델이 해당 클래스로 예측한 샘플 중

실제로 해당 클래스인 샘플의 비율 TP / (TP + FP)

3 모델링

파생변수 추가



plot_x1

1차항 계수

plot_x2

2차항 계수

plot_x3

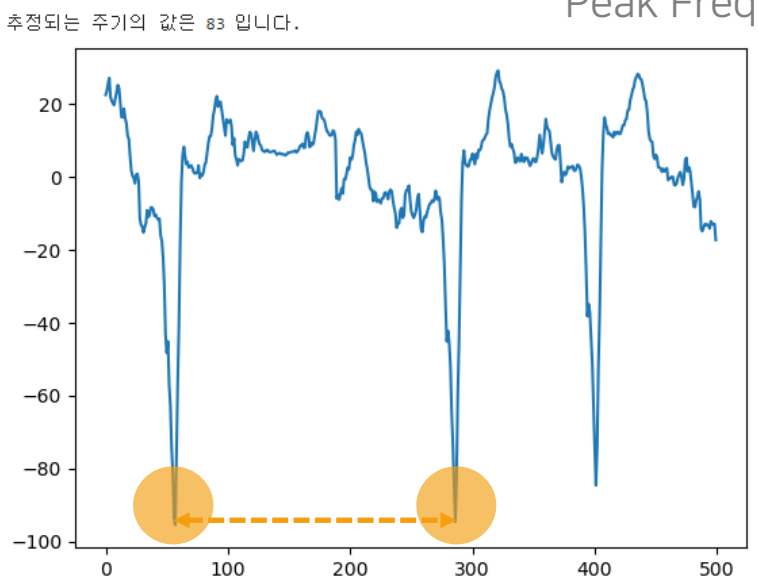
3차항 계수

다항회귀를 통해 추세를 추정하고 회귀식의 계수를 파생변수로 이용

3 모델링

파생변수 추가

Peak Frequency 이용!



periods

추정 주기 값

추세를 제거한 데이터에 대해 계절성을 FFT 로 추정해 주기 값을 파생변수로 이용

3 모델링

파생변수 추가

plot_x1_mean

범주별 평균 플롯 추세 다항회귀1차항 계수

plot_x2_mean

범주별 평균 플롯 추세 다항회귀 2차항 계수

plot_x3_mean

범주별 평균 추세 다항회귀 3차항 계수

Periods_mean

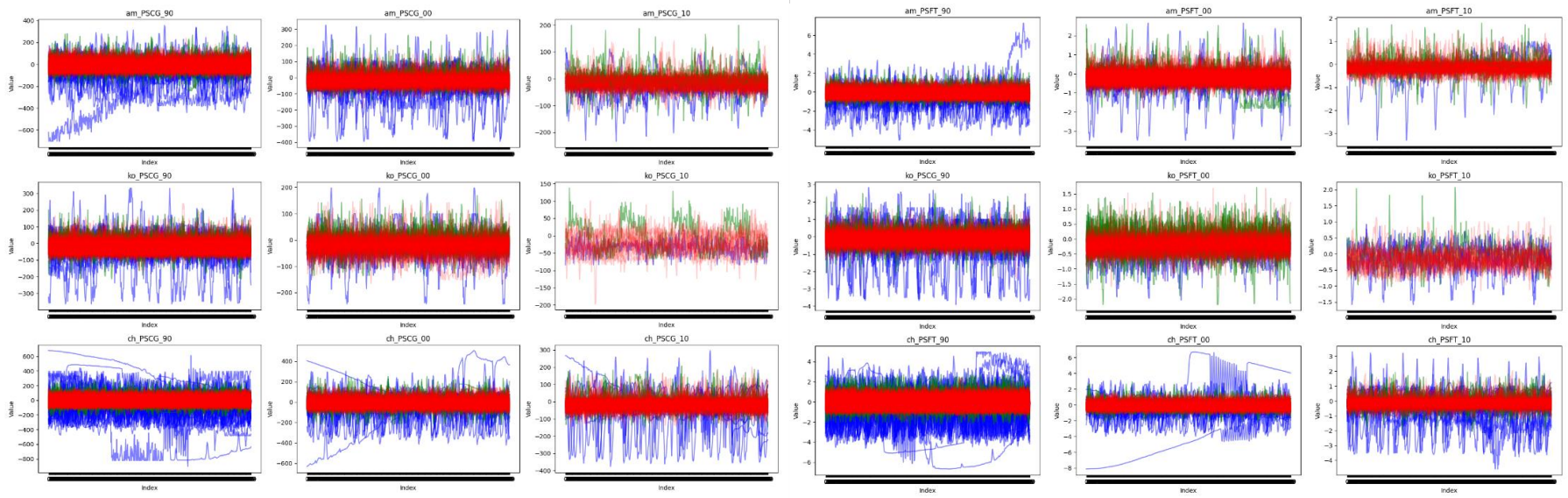
범주별 평균 플롯 추정 주기 값

4

최종 모델

4 최종모델

Categorical Variables



범주형 변수들에 따라 데이터를 18개 그룹으로 분류해서 time plot을 그려봄

그룹 안에서 라벨이 명확하게 나뉘거나 특징을 가지지 않는 것을 확인

4 최종모델

Categorical Variables

크래머의 V (Crammer's V)

2개 이상의 수준을 지닌 두 범주형 변수 간 연관성을 파악하는 데 사용

연속형 상관계수처럼 0~1 사이의 값을 지님

$$V = \sqrt{\frac{\chi^2}{n(\min(I,J)-1)}}$$

n : 전체 도수의 합 , χ^2 : $\sum \frac{(O-E)^2}{E}$, I, J : 각 변수들 수준의 개수

범주형 변수들을 선택하는 과정에서 **상관관계로 인한 문제**가 발생하지 않도록

Crammer's V를 먼저 계산!

4 최종모델

Categorical Variables

크래머의 V (Crammer's V)

2개 이상의 수준을 지닌 두 범주형 변수 간 연관성을 파악하는 데 사용

연속형 상관계수처럼 0~1 사이의 값을 지님

```python
from scipy.stats import chi2_contingency
import numpy as np

cross_tab = pd.crosstab(df['S/N'],df['Country'])

# 카이제곱 통계량 계산
chi2, p, dof, ex = chi2_contingency(cross_tab)

# 총 샘플 수
n = cross_tab.sum().sum()

# Cramér's V 계산
cramer_v = np.sqrt(chi2 / (n * (min(cross_tab.shape) - 1)))

print(f"Cramér's V: {cramer_v}")
```

```
Cramér's V: 0.02815437865848964
```

```python
from scipy.stats import chi2_contingency
import numpy as np

cross_tab = pd.crosstab(df['Year'],df['Country'])

# 카이제곱 통계량 계산
chi2, p, dof, ex = chi2_contingency(cross_tab)

# 총 샘플 수
n = cross_tab.sum().sum()

# Cramér's V 계산
cramer_v = np.sqrt(chi2 / (n * (min(cross_tab.shape) - 1)))

print(f"Cramér's V: {cramer_v}")
```

```
Cramér's V: 0.010567890161578032
```

```python
from scipy.stats import chi2_contingency
import numpy as np

cross_tab = pd.crosstab(df['Year'],df['S/N'])

# 카이제곱 통계량 계산
chi2, p, dof, ex = chi2_contingency(cross_tab)

# 총 샘플 수
n = cross_tab.sum().sum()

# Cramér's V 계산
cramer_v = np.sqrt(chi2 / (n * (min(cross_tab.shape) - 1)))

print(f"Cramér's V: {cramer_v}")
```

```
Cramér's V: 0.008360258342189546
```

인코딩 된 Year, S/N, Country 사이의 Crammer's V가 **0.05이하**

매우 작은 연관성을 가지고 있다고 판단함

4 최종모델

Categorical Variables

크래머의 V (Crammer's V)

2개 이상의 수준을 지닌 두 범주형 변수 간 연관성을 파악하는 데 사용

연속형 상관계수처럼 0~1 사이의 값을 지님

범주형 변수들의 다양한 조합을 시도

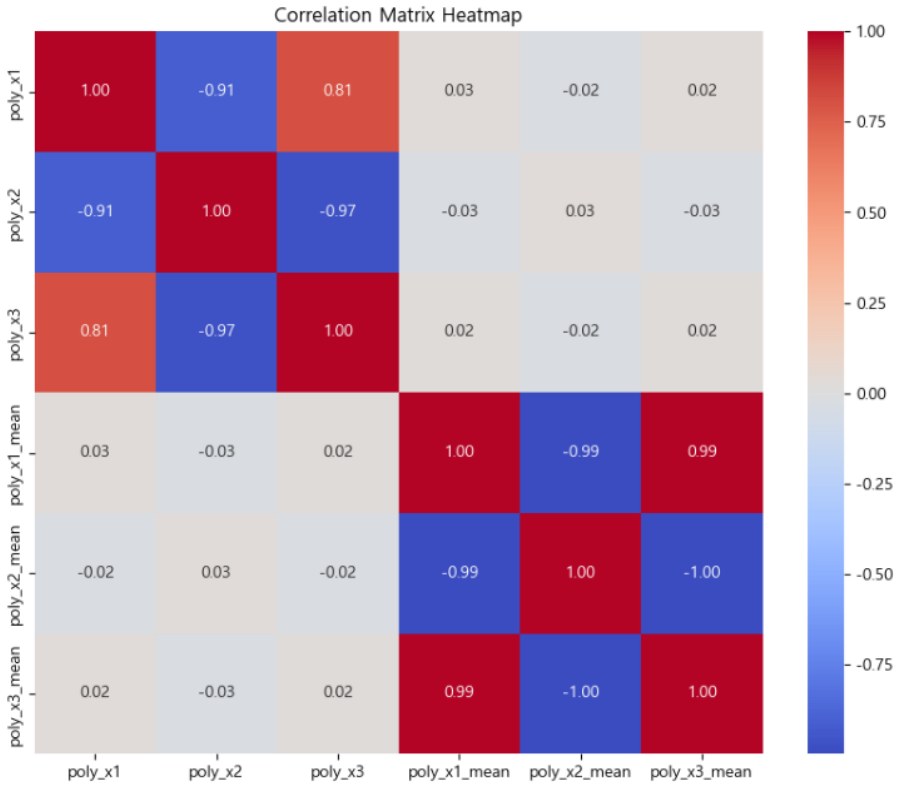
Feature selection결과 범주형 열들은 모델링에 이용하지 않기로 함

인코딩 된 Year, S/N, Country 사이의 Crammer's V가 0.05이하

매우 작은 연관성을 가지고 있다고 판단함

4 최종모델

파생변수 간의 상관관계



강한 양의 상관관계

poly_x1과 poly_x3

poly_x1_mean과 poly_x3_mean

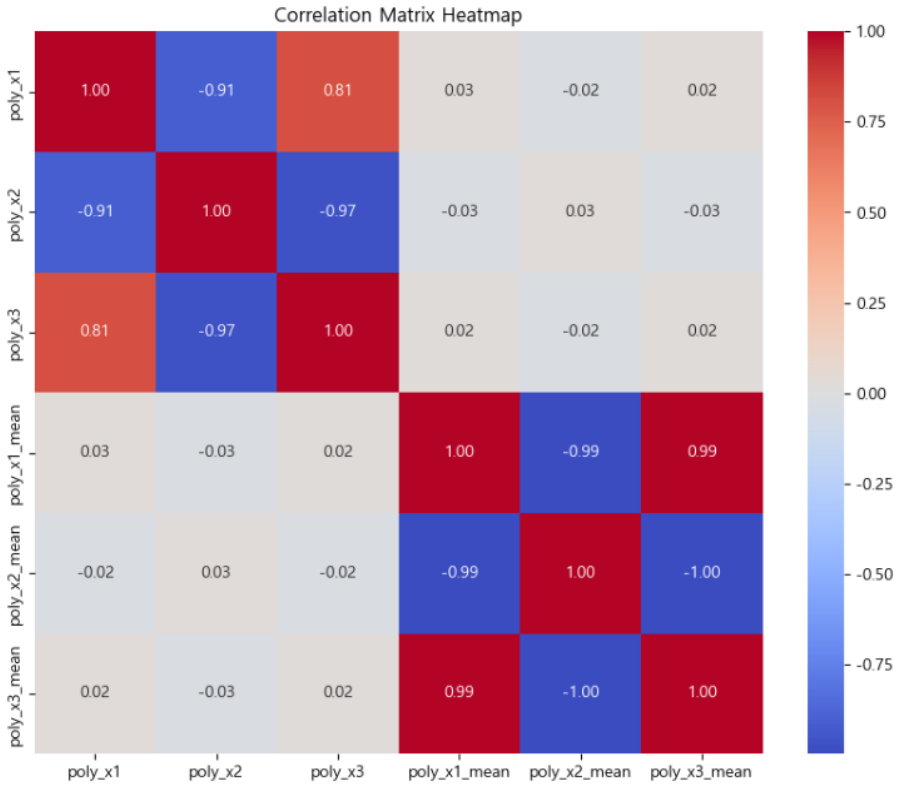
강한 음의 상관관계

poly_x2와 poly_x1, poly_x3

poly_x2_mean과
poly_x1_mean, poly_x3mean

4 최종모델

파생변수 간의 상관관계



양의 상관관계를 고려해

poly_x1_mean

음의 상관관계를 고려해

Poly_x2와 poly_x2_mean

변수들을 삭제

4 최종모델

파라미터 튜닝

XGBoost

LGBM에 비해 지속적으로 높은 성능을 보인 XGBoost에
해당 변수들을 넣고 파라미터를 튜닝하는데 집중해서 모델 완성

감사합니다




성능 올리기도 중요하지만 밥은 먹고 해야지…





비가 와도.. 화재 경보가 울려도.. 짱박혀서 간식 까먹으며 코딩하는 우리



솔미(정민)

처음 시작했을 때 진짜 막막해서 손도 못대고 있었는데 마지막 날이 되니까 해보고 싶은 시도가 너무 많았던 방학 세미나입니다… 그동안 여러가지 시도해보고 다같이 부대끼면서 즐거운 시간이었습니다 ^__^ 다들 파이팅

서머야~ 후기쓰세욤

서머(여원)

참돌이(호성)

참돌아~ 후기쓰세욤

케찹(형석)

케챱아~ 후기쓰세욤

학기가 끝났는데도 매일매일 학교에 나오느라 정말 수고 많으셨습니다!
늘 회귀팀에만 콕 박혀 있었는데 이렇게 다른 팀 오빠, 친구, 동생들과 함께할 수 있어서 뜻깊었던 것 같습니다. 다음 학기 피셋도 화이팅입니다!!

뽀야미(다은)

귀오미(현진)

귀오먀~ 후기쓰세욤