
# LG Aimers 5기

## Phase2 온라인 해커톤 발표자료

팀 : IBA_6.2

# 데이터 전처리

# 의미가 없다고 판단한 컬럼 제거

LG Aimers

- 아래의 경우와 같은 컬럼은 의미가 없다고 판단하여 순차적으로 제거 진행
  - 모든 행이 결측치로 구성된 컬럼 → train data의 278개 컬럼 제거
  - 모두 동일한 값으로 구성된 칼럼 → train data의 35개 컬럼 제거
  - 중복 컬럼 → 중복된 컬럼들 중 한 컬럼만 남기고 제거

    → train data의 26개 컬럼 제거
  - 'OK'와 결측치만 가진 컬럼 → 'HEAD NORMAL COORDINATE X AXIS(Stage1) Judge Value_Dam' 변수 제거

- 기존 464개였던 train data의 컬럼이 124개로 감소
- train data에서 제거된 컬럼을 동일하게 test data에서도 제거 진행

# OK 값으로 오염된 컬럼 처리

LG Aimers

- 아래의 컬럼은 OK 값으로 오염됐을뿐더러 결측치도 존재
  - HEAD NORMAL COORDINATE X AXIS(Stage1) Collect Result_Dam
  - HEAD NORMAL COORDINATE X AXIS(Stage1) Collect Result_Fill1
  - HEAD NORMAL COORDINATE X AXIS(Stage1) Collect Result_Fill2

- 해당 컬럼들이 토출 좌표의 X값이라는 점에 착안해서 Y, Z값을 이용해서 결측치 대체 시도
  - OK값으로 오염된 경우도 결측치로 취급하고 진행
  - 대체되는 값은 새로운 값이 아닌, 해당 열에 존재하는 값들로 대체 진행
    1. 결측치가 없는 행들의 X, Y, Z 총 합의 평균을 구함
    2. 1번에서의 평균에서 결측치가 있는 행들의 Y, Z 값을 뺌
    3. 2번에서 구한 값과 가장 가까운 고유값으로 대체

# 상관관계가 매우 높은 컬럼들 처리

LG Aimers

- 상관관계가 1 or -1인 컬럼들 존재
  - 해당 컬럼들의 관계를 보니, 중복도가 높다는 것을 발견(카디널리티가 낮다)

| HEAD NORMAL COORDINATE Y AXIS(Stage2) Collect Result_Fill2 | HEAD NORMAL COORDINATE Y AXIS(Stage3) Collect Result_Fill2 | HEAD NORMAL COORDINATE Z AXIS(Stage2) Collect Result_Fill2 |
|---|---|---|
| 428.0 | 427.9 | 243.7 |
| 427.9 | 428.0 | 243.5 |
| 1324.2 | 1324.2 | 243.5 |

| CURE END POSITION X Collect Result_Fill2 | CURE START POSITION X Collect Result_Fill2 |
|---|---|
| 240 | 1020 |
| 1020 | 240 |

[표 1, 2 해당 컬럼들의 고유한 행들]

# 상관관계가 매우 높은 컬럼들 처리

LG Aimers

- 해당 컬럼들을 하나의 컬럼으로 합치는 시도 진행
  - 기본적으로 사칙연산을 통해 여러 컬럼들을 하나의 컬럼으로 통합
  - 합치기 전의 값들의 차이는 유지하기 위해, 해당 컬럼들의 고유값 개수를 유지하는 것을 원칙으로 컬럼들을 통합
  - 124개 였던 train data의 컬럼이 98개로 감소
  - 해당 시도를 test data에도 똑같이 적용

- 위의 시도를 상관관계가 0.99 이상 or -0.99 미만인 컬럼들에게도 적용
  - 고유값 개수를 유지하는 원칙을 지킬 수 있는 경우에만 컬럼 통합 진행
  - 98개 였던 train data의 컬럼이 77개로 감소
  - 해당 시도를 test data에도 똑같이 적용

# 범주형 변수 인코딩

LG Aimers

- 범주가 2개인 범주형 변수는 0, 1로 구성된 이진변수로 인코딩
  - 대상 컬럼 :  Equipment_Dam, Equipment_Fill1, Equipment_Fill2, Chamber Temp. Judge Value_AutoClave

- 범주가 3개 이상인 범주형 변수는 라벨인코딩과 원-핫 인코딩 중 성능이 좋았던 원-핫 인코딩 진행
  - 대상 컬럼 : Model.Suffix_Dam, Workorder_Dam
  - 각 범주형 변수의 범주 개수만큼 컬럼이 늘어남 → train data의 컬럼 개수 : 745개

# 불균형 데이터 문제

LG Aimers

- 제품 이상여부 판별 프로젝트는 이진분류 문제이며, 데이터 불균형 문제 존재
  - train data의 target 변수 : (Normal : 38158개, AbNormal : 2350개)

- 샘플링을 통한 데이터 불균형 문제를 해결 시도
  - 여러 샘플링 방법 중 Random OverSampling, Random UnderSampling을 최종 후보군으로 선정
    - 이 외 시도한 샘플링 기법들 : EditedNearestNeighbours, CondensedNearestNeighbour, NeighbourhoodCLeaningRule ADASYN, OneSidedSelection TomekLinks, SMOTE, SMOTETom다, SMOTEENN 등
  - Random UnderSampling의 경우는 소수 클래스의 개수 대비 몇 배수까지 샘플링할지를 실험을 통해 선정
    - 최종 후보군 : 2배, 3배

# 모델 구축

# 모델 구축

LG Aimers

- 모델 검증 방법 : Stratified-K Fold 방법
  - 분류 문제의 경우, 각 폴드 y값의 클래스 비율을 동일하게 하는 검증 방법이기 때문

- 실험할 모델 선정
  - RandomForest, XGBoost, LightGBM, CatBoost를 후보군으로 선정

- 모델 훈련 과정
  1. Stratified- K Fold로 데이터를 train set, validation set으로 분할
  2. train set에 대해 샘플링 기법 적용
  3. 샘플링 기법이 적용된 train set을 이용하여 모델 학습 진행
  4. 학습된 모델을 validation set으로 성능 확인
  5. 이를 K번(k = 5로 진행) 반복하여 얻은 성능의 평균값을 확인

# 앙상블 모델

LG Aimers

- 앙상블 모델 : 여러 개의 기본 모델을 활용하여 만든 하나의 새로운 모델
  - Stacking, bagging 등 여러 앙상블 기법을 적용해보았으며, 가장 성능이 좋았던 weighted blending 기법을 선택

- weighted blending : 각 모델의 예측값에 대하여 weight를 곱하여 최종 output을 계산
  - 총 weight는 1이 되어야함
  - 분류 모델의 경우, 클래스를 예측할 확률을 예측값으로 사용

- weighted blending에 사용된 모델들

  앞선 실험들 중 가장 성능이 좋았던 모델 3개를 선정
  - RandomOverSampling + CatBoost → weight : 0.5
  - RandomUnderSampling 2배수 + CatBoost → weight : 0.3
  - RandomUnderSampling 3배수 + RandomForest → weight : 0.2

# 임계값 최적화 진행

LG Aimers

- 이진 분류 모델은 특정 클래스에 속할 확률을 예측하며, 그 확률을 기준으로 특정 임계값을 넘으면 해당 클래스로 예측 진행
- 보통 임계값은 0.5로 지정 → 최적의 성능을 보여주는 임계값을 탐색
  - 도출한 최적 임계값 : 0.5109999999999878
  - 해당 임계값을 최종 예측에 적용

# 후처리 진행

LG Aimers

- 데이터를 살펴보는 중 일정한 규칙 발견
  - Equipment 관련 변수들(Equipment_Dam, Equipment_Fill1, Equipment_Fill2)은 dispenser #1 or dispenser #2 값으로 구성
  - 각 변수들이 동일한 번호를 가지지 않으면 불량'이라는 규칙을 발견
  - 모델을 이용한 예측이 완료된 후, 해당 규칙에 맞지 않은 예측값에 대해서 수정 진행

| Equipment_Dam | Equipment_Fill1 | Equipment_Fill2 | target | train data의 개수 |
|---|---|---|---|---|
| dispenser #1 | dispenser #1 | dispenser #1 | 정상 or 불량 | 25011 |
| dispenser #1 | dispenser #1 | dispenser #2 | 불량 | 6 |
| dispenser #1 | dispenser #2 | dispenser #1 | 불량 | 0 |
| dispenser #1 | dispenser #2 | dispenser #2 | 불량 | 13 |
| dispenser #2 | dispenser #1 | dispenser #1 | 불량 | 10 |
| dispenser #2 | dispenser #1 | dispenser #2 | 불량 | 0 |
| dispenser #2 | dispenser #2 | dispenser #1 | 불량 | 5 |
| dispenser #2 | dispenser #2 | dispenser #2 | 정상 or 불량 | 15461 |

[표 3 발견한 규칙]

# 최종 성능

LG Aimers

- 성능은 Stratified-K Fold 평균 f1 score를 의미
  - RandomOverSampling + CatBoost 성능 : 0.196268
  - RandomUnderSampling 2배수 + CatBoost 성능 : 0.192071
  - RandomUnderSampling 3배수 + RandomForest 성능 : 0.196817

  - 최종 weighted blending 모델 성능: 0.217305
  - 임계값 최적화 적용 후 성능 : 0.219159

  - Public Score : 0.219157
  - Private Score : 0.229658