Threshold자동계산 코드추가

# 목표

* 현재 모든 모델이 학습 후 deafault값(1)으로 설정됨
* 학습 완료 후 자동으로 Threshold를 구하는 기능을 추가하고자 함

# 수정사항 및 차이점 확인

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model Name** | **수정 파일경로** | **데이터 경로** | **데이터 분리** | **test데이터 적용** |
| **ANNOY** | src\api\_server\apis \multiproc\proc\_annoy.py | InfluxDB | 앞의 5일 데이터를 Test 데이터로 사용 | Test\_data에window\_size를 적용시켜서 사용(vectorize) |
| **VAE** | src\api\_server\apis \multiproc\proc\_vae.py | InfluxDB | 앞의 5일 데이터를 Test 데이터로 사용 | Test\_data그대로 사용 |
| **ICA** | src\api\_server\apis \multiproc\proc\_ica.py | InfluxDB | 앞의 5일 데이터를 Test 데이터로 사용 | Test\_data그대로 사용 |
| **AAKR** | src\api\_server\apis \routes\aakr.py | request.data | 전체데이터를 8:2비율로 나눠서 사용 | Test\_data그대로 사용 |

# 기존 방식 확인, 변경 및 추가사항

* ANNOY기존 방식, 변경사항

Model\_training (api\_server/apis/multiproc/proc/annoy.py파일)  
influx에서 데이터 조회 -> 전체 데이터 concat후 scaler fit -> 각 pkl파일 scaler적용  
-> 조정된 데이터로 모델 학습 -> 모델 및 스케일러 저장

influx에서 데이터 조회 -> train\_test데이터 split -> train데이터로만 scaler fit ->   
-> train, test 데이터 scaler적용 -> train데이터로 모델 학습 -> 모델 및 스케일러 저장  
-> 학습된 모델로 test진행 -> test\_data 크기 조정 -> threshold계산 -> threshold csv파일로 저장

* VAE기존 방식, 변경사항

Model\_training (api\_server/apis/multiproc/proc/vae.py파일)  
influx에서 데이터 조회 -> 전체 데이터 concat후 scaler fit -> 각 pkl파일 scaler적용  
-> 조정된 데이터로 모델 학습 -> 모델 및 스케일러 저장

influx에서 데이터 조회 -> train\_test데이터 split -> train데이터로만 scaler fit ->   
-> train, test 데이터 scaler적용 -> train데이터로 모델 학습 -> 모델 및 스케일러 저장  
-> 학습된 모델로 test진행 -> threshold계산 -> threshold csv파일로 저장

* ICA기존 방식, 변경사항

Model\_training (api\_server/apis/multiproc/proc/ica.py파일)  
influx에서 데이터 조회 -> 전체 데이터 concat후 scaler fit -> 각 pkl파일 scaler적용  
-> 조정된 데이터로 모델 학습 -> 모델 및 스케일러 저장

influx에서 데이터 조회 -> train\_test데이터 split -> train데이터로만 scaler fit ->   
-> train, test 데이터 scaler적용 -> train데이터로 모델 학습 -> 모델 및 스케일러 저장  
-> 학습된 모델로 test진행 -> threshold계산 -> threshold csv파일로 저장

* AAKR기존 방식, 변경사항

Model\_training (api\_server/apis/multiproc/proc/aakr.py파일)  
Request에서 데이터 확보 -> 모델저장

Request에서 데이터 확보 -> train\_test데이터 split -> train데이터로 모델저장  
-> 학습된 모델로 test진행 -> threshold계산 -> threshold csv파일로 저장

* 추가사항

threshold계산함수가 들어있는 .py생성 (src\tag\_setting\threshold\_calculator.py)

기존방식 (잔차 평균 + 6 x 잔차 표준편차), 잔차의 min값, 잔차의 max값,   
raw\_data와의 이동평균 잔차값, 잔차 상위 95%값의 5가지 방식을 사용함

# 추후 계획

* threshold계산 방법 수정
* 현재는 csv파일로 결과가 저장되지만 추후에는 model객체에 threshold를 추가하여  
  화면에서 선택할 수 있는 기능 추가 예정