* **분석 개요 ==========================================================================**

1. **사용 언어 :** Python
2. **분석 환경 :** [CPU] Apple M2, [RAM] 8GB, [GPU] Colab T4
3. **분석 데이터 정보**
   1. 수집 기관 :
   2. 데이터 내용 :
   3. 데이터 파일 :
4. **분석 데이터 구조**
   1. Features : 500 columns (Sensor values)
   2. Samples : 4921 raw (Train 3601, Test 1320)
   3. Labels : 1 or 0 (Normal or Abnormal)
5. **학습 및 예측 목표**
   1. 500개 각 센서 값들을 학습하여 그 상태가 Normal(1) 인지 Abnormal(0) 인지 추측
   2. Train 3601개의 샘플을 학습 후 Test 1320개 샘플의 상태를 추측
6. **데이터 학습에 사용된 모델**
   1. LogisticRegression
   2. TabularPredictor (Auto ML)
   3. XGBClassifier
   4. LGBMClassifier
   5. CatBoostClassifier
   6. RandomForestClassifier
   7. **CNN (Convolutional Neural Network)**
7. **데이터 학습 및 예측 결론**
   1. Best 모델 : CNN (Convolutional Neural Network)
   2. CNN 모델은 주로 이미지나 지역적 패턴이 있는 데이터를 학습 및 예측할 때 사용되는 모델로, 분석 대상인 데이터셋도 인접 하는 센서 값과 함께 패턴을 이루는 특성이 있기 때문에 CNN 모델에서 효과적으로 학습 되는 것으로 보임

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

* **데이터 분석 (1) ======================================================================**

1. 설비에서 자동 저장 된 25개의 csv 파일 요약본 확인
   * 25개의 각 csv 파일마다 공정별 데이터가 저장 되어 있음

**텍스트, 스크린샷, 폰트, 흑백이(가) 표시된 사진

자동 생성된 설명텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명**

* + 아래는 25개 파일에 대한 결과 요약 데이터프레임 (1행 = 1개 csv 파일 정보)
  + 결과 라벨(yes or no), 공정 완료여부(yes or no) 등 6개의 정보를 담고 있음

**텍스트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명**

* + - Machining\_finalized (공정 완료여부) 항목이 ‘no’ 일 경우, pssed\_visual\_inspecion(외관검사) 항목은 ‘NAN’ 으로 되어있음 (외관검사를 진행하지 않았기 때문)

1. 25개의 csv 파일 크기 확인 및 라벨 클래스의 빈도 균형 확인
   * 먼저, **csv** 파일명 25개를 리스트로 저장 (각 파일명을 for 문으로 불러오기 위해)

텍스트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

* + - cnc\_data 폴더에는 25개의 원본 csv 파일과 1개의 결과 요약 파일(train.csv)이 있음
    - 결과 요약 파일을 제외하고 25개의 csv 파일명을 리스트로 저장
  + 25개 원본 csv 파일을 데이터프레임 형태로 변환 후 리스트에 저장

텍스트, 스크린샷, 디자인이(가) 표시된 사진

자동 생성된 설명

* + - summary\_df : 결과 요약 데이터프레임
    - 각 csv파일(각 행)의 크기 정보를 추가

-> 라벨 열(paased\_visual\_inspection)의 빈도 분포를 확인 하기 위함

* + 라벨 클래스별 빈도 확인

텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

* + - 공정 완료 여부, 외관검사 합격 여부 feature를 기준으로 groupby() 하여 클래스 빈도 확인
      * 공정 미완료 fail : 3228 (10.07%)
      * 공정 완료 fail : 6175(19.27%)
      * 공정 완료 pass : 22645(70.66%)
    - pass, fail 의 비중이 약 7:3 정도로 나타남 -> 학습 하는데는 문제 없을 것
* **데이터 전처리 (1) ====================================================================**

1. 25개의 csv파일에 결과 요약 데이터를 합친 뒤 25개 파일을 1개의 파일로 만들기
   * 결과 요약 feature를 각 데이터프레임에 합치기

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* + - concat() 처리 후 25개 데이터프레임의 열은 56개
  + 25개 데이터 프레임을 하나로 합치기

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

* + - concat\_df : 총 32048개의 행(데이터)를 가진 데이터 프레임

1. 불필요한 feature 제거, 이진 데이터 1과0으로 변환, 원핫인코딩

텍스트, 스크린샷, 폰트, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

* + - 모든 값이 같거나 의미 없는 feature 제거 (7개 열)
    - 이진 데이터 값은 1과 0으로 변환 (3개 열 : 라벨, 공정완료여부, tool\_condition)
    - 원핫인코딩 처리 (2개 열 : 프로그램 번호, 현재 공정명)
    - 전처리 완료 후 데이터 프레임 : df
  + 데이터 전처리 완료 후 데이터프레임 정보
* **데이터 분석 (2) ======================================================================**

1. 공정 미완료 데이터에 대한 분석 (학습에 포함 시켜야 할지 확인)
   * 공정 미완료 데이터는 외관 검사가 진행되지 않았기 때문에 ‘fail’로 일괄 처리 했음. 불량의 공정 조건을 찾는데 학습 노이즈가 될 수 있음
   * 각 Feature 별로 공정 완료/미완료 클래스를 구분하여 평균값 분포 확인

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

* + - 공정 완료/미완료 클래스의 feature별 평균값에 차이가 있음
    - 단순히 시간적으로 공정이 미완료 된 것이 아니라, 조건의 차이로 인해 문제가 발생되어 공정이 완료되지 않은 것으로 보임
    - 따라서, 미완료 항목을 포함하여 학습 시키는 것이 맞음. (fail 클래스의 feature 특징을 가지고 있으므로)

1. 공정 완료/미완료 클래스의 현재 진행 공정 분포 확인
   * 공정 미완료 클래스의 공정 분포가 일부 앞 공정에만 쏠려 있을 경우 학습에 영향을 줄 수도 있어 확인이 필요
     + 만약 미완료 클래스의 공정이 모두 1,2번 공정에서 끝난다면 학습 결과 1,2번 공정의 데이터는 모두 fail로 판정할 확률이 높음

텍스트, 스크린샷, 다채로움, 원이(가) 표시된 사진

자동 생성된 설명

* + - 공정 완료 클래스는 현재 공정의 빈도가 균일함
    - 공정 미완료 클래스는 현재 공정의 빈도가 균일 하지는 않으나, 학습에는 문제 되지 않을 것으로 보임
    - 전체 클래스는 현재 공정의 빈도가 균일함

1. 전체 데이터의 feature 별 평균 값 분포 확인

**텍스트, 소프트웨어, 라인, 웹 페이지이(가) 표시된 사진

자동 생성된 설명**

* + - 라벨 클래스별로 feature값에 차이가 나타남 --> 모델 학습이 가능함
  + 라벨 클래스별 차이가 큰 feature 확인

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

* + - 클래스별 값 차이가 큰 Top3 feature가 결과 요약 파일에 있는 feature
    - 결과 요약파일의 정보가 결과 예측에 직접적으로 영향을 줄 것으로 예상
    - 그러나 결과 요약파일은 설비에서 자동으로 얻어지는 데이터가 아니므로 포함/미포함 하여 학습 시켜볼 필요 있음. (먼저, 포함시켜서 학습)

1. 라벨(target) 과 상관계수 분포 확인

텍스트, 스크린샷, 라인, 디자인이(가) 표시된 사진

자동 생성된 설명

* + - target과 상관계수 높은 feature top10 항목과 target 클래스 별 평균값 차이가 큰 feature top10 은 일치함

-> 라벨 클래스별 평균 값 차이가 큰 feature가 클래스 결정에 영향을 준다.

1. 결과에 가장 영향을 많이 주는 feature 값이 클래스별로 어떻게 차이가 나는지 확인

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**텍스트, 도표, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**[clamp\_pressure]**

0 클래스 83%가 4.0

-> 클래스별 압력 값의 분포 차이가 크기 때문에 결과를 예측할 때 크게 영향을 줄 수 있음

**[feedrate]**

0 클래스만 12, 15 값을 가짐

-> 결과를 예측할 때 직접적인 영향을 줄 수 있음

**텍스트, 도표, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

* **모델 학습 및 결과 예측 (1) ==========================================================**

1. 4가지 머신러닝 모델 사용하여 학습 및 결과 예측
   * 학습 전 train, test 데이터셋 나누기, 라벨 열 나누기

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

* + - test\_size = 0.3 (전체 데이터 중 30%를 test 데이터 셋으로)
    - stratify = df[‘target’] : train, test 데이터셋을 분리한 이후에도 지정한 column의 클래스 비율을 일정하게 유지함
    - 학습 시킬 train 데이터의 feature는 59개 - x\_train shape : (22433, 59)
* **데이터 분석 및 학습 실험 결과 =====================================================**

1. 최적의 학습 모델
   * CNN (Convolutional Neural Network)
   * 정확도 : 97.2 ~ 97.5 %
2. 데이터 전처리 후 학습 결과 (3가지)
   * 데이터 전처리 효과 없음

(1) 평균 센서 값 높은 샘플 제거 (5개, 10개, 20개 행 제거)

(2) 403번 이후 센서만 학습 (98개 열로만 학습)

(3) target 상관관계 하위 센서 제거 (10개, 30개 열 제거)

1. 모델별 학습 결과 요약

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명