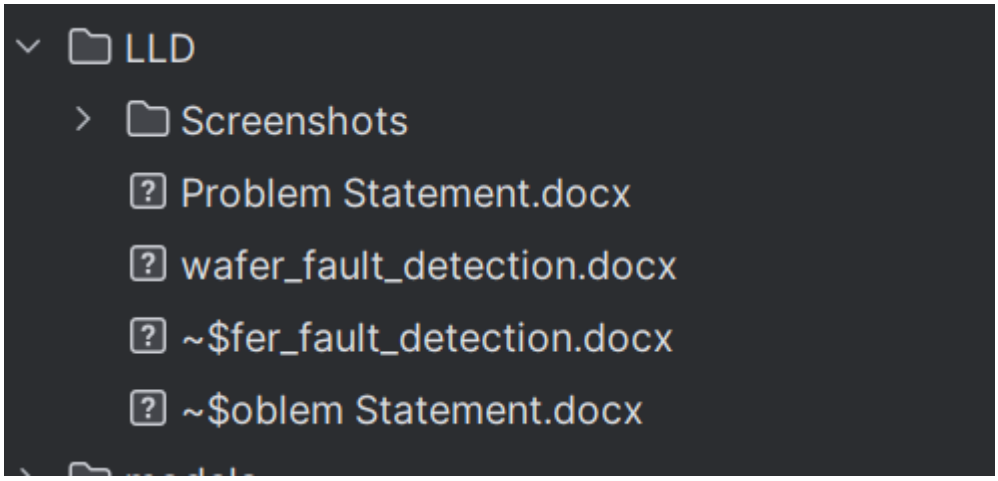


LLD라는 폴더에 워드 문서 있는데, 여기에 어떤 방식으로 진행되는지랑 코드가 어떤 역할 하는지 적혀있습니다!

Knn imputer

:KNN알고리즘 이용 가까운 이웃의 수를 정하고 결측치를 채우는 방식.

입력된 데이터에 decision boundary를 잡고 그 안에 데이터 값들을 바탕으로 데이터를 처리.

CNN

합성곱 신경망을 이용한 기계 학습은 데이터를 비선형하게 학습하여 새로운 데이 터를 정확히 처리할 수 있도록 하는 분야로 다양한 영역에서 활용되고 있다

또한 특징값을 일일이 입력하지 않아도 영상과 정답(라벨)을 입력하면, 매 epoch마다 특징을 추출하여, 자동으로 분류하기 위한 가중치를 업데이트 해나간다.

XGBoost

https://zephyrus1111.tistory.com/232

Gradient Boosting 방법 중 하나 (extreme gradient boosting)

트리 기반 병렬 처리를 효율적으로 빠르게 학습하고 예측하는 기법

Random forest: 의사결정나무 모델 기반,

Boosting: 중복된 데이터 사용해서 데이터의 편향이 올라감

decision tree: 데이터 전부를 사용 (오버피팅 되기 쉬움)

Random forest : 특정 데이터셋을 가져온다 (랜덤하게)

간단하게 렌덤한 특징들을 뽑아서

GridSearchCV : 성능향상 위해 쓰이는거.

사용자가 직접 모델의 하이퍼 파라미터의 값을 리스트로 입력하면 값에 대한 경우의 수마다 예측 성능을 측정 평가하여 비교하며 최적의 하이퍼 파라미터를 찾는 과정

k-means clustering : k개의 클러스터 (비슷한 집단)을 만들어 유사한 데이터 포인트끼리 grouping 패턴을 찾아내는 것.

얼마나 많은 클러스터 필요한지 결정

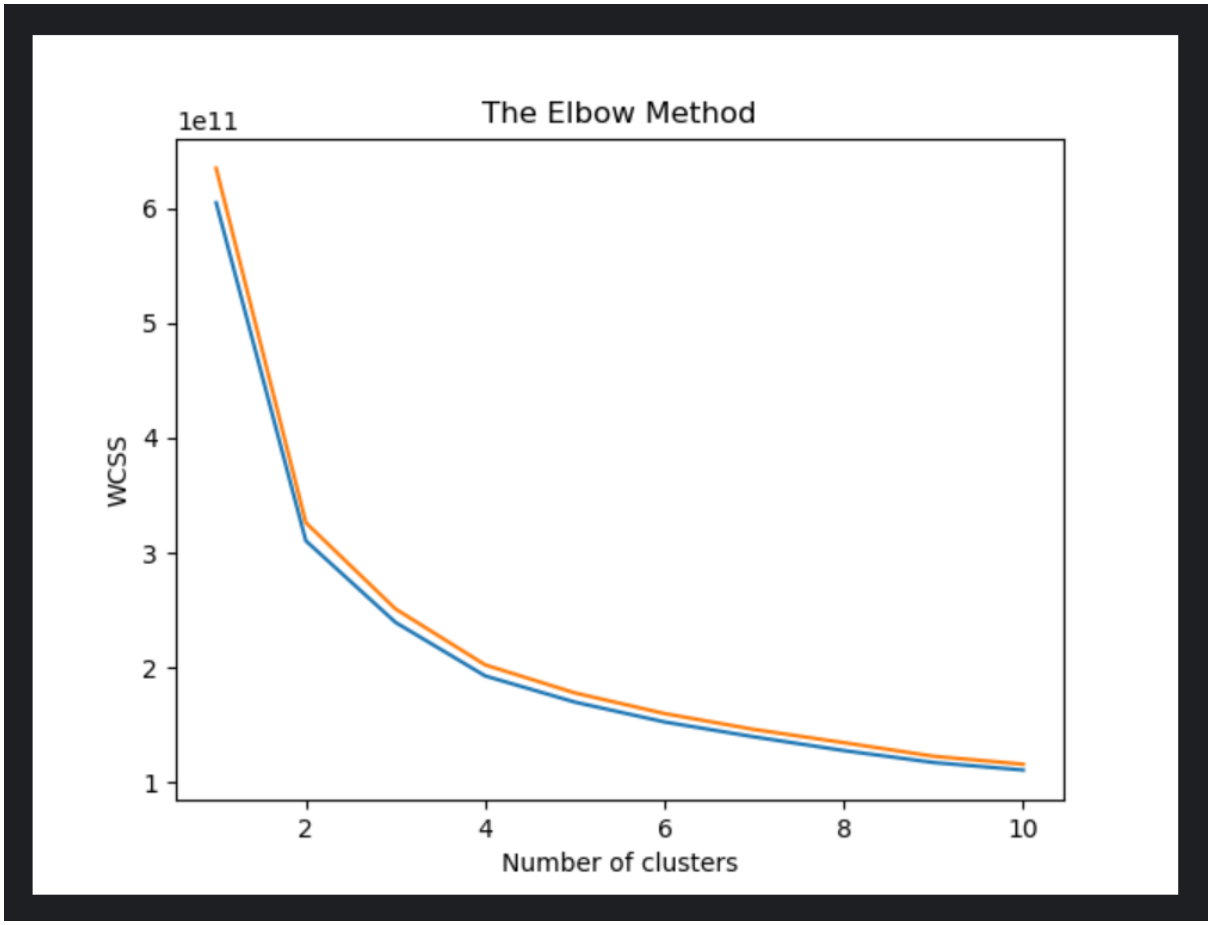
Preprocessing.py : KNN imputer사용.

1. Remove_columns : 전처리를 하기 전에 데이터 파일 내에 필요없는 열 정보를 제거하는 역할. (Ex: wafer 데이터가 없는 열을 삭제)
2. Separate_label_feature: 입역된 데이터에서 label열을 분리하여 feature와 label을 각 데이터프레임으로 변환함. Ex: wafer 1 (label) 데이터들 (feature)
3. Is_null_present : 데이터에 값이 없는 경우 Knn imputer(k=3, NaN 결측치를 채우는 과정임) 원하는 인접 이웃 수(k=3)에서 가중 또는 가중평균을 사용한다. (가중평균 사용함)
4. Get_coloumn_with_zero_std_deviation : 열의 표준편차(standard deviation)가 0인지 찾는것. 표준편차가 0이란 뜻은 데이더 전체가 일정한 값을 가짐을 의미. 다 good 아니면 다 bad. Drop.col_to_drop을 사용해 데이터프레임에서 불필요한 열을 삭제한다.

Clustering,py: Kmeans 사용

Kmeans에 값을 입력해줘야 하는데 (크러스터링 알고리즘 군집 개수) 이걸 정하려고 elbow plot씀

1. Elbow_plot: 군집을 추가로 늘려가며 (우린 42까지 함)군집내 변동성이 급감하는 군집 개수를 찾는 것. 이 뜻은 유사한 데이터끼리 잘 묶였다고 파악하는 것.



(실제 저장된 elbow method png) preprocessing_data 안에 있다. (여기에 null data도 있음)

WCSS(within cluster sum of squares)는 모든 클러스터에 있는 각 데이터가 중심까지의 거리를 제곱하여 합을 계산하는 것.

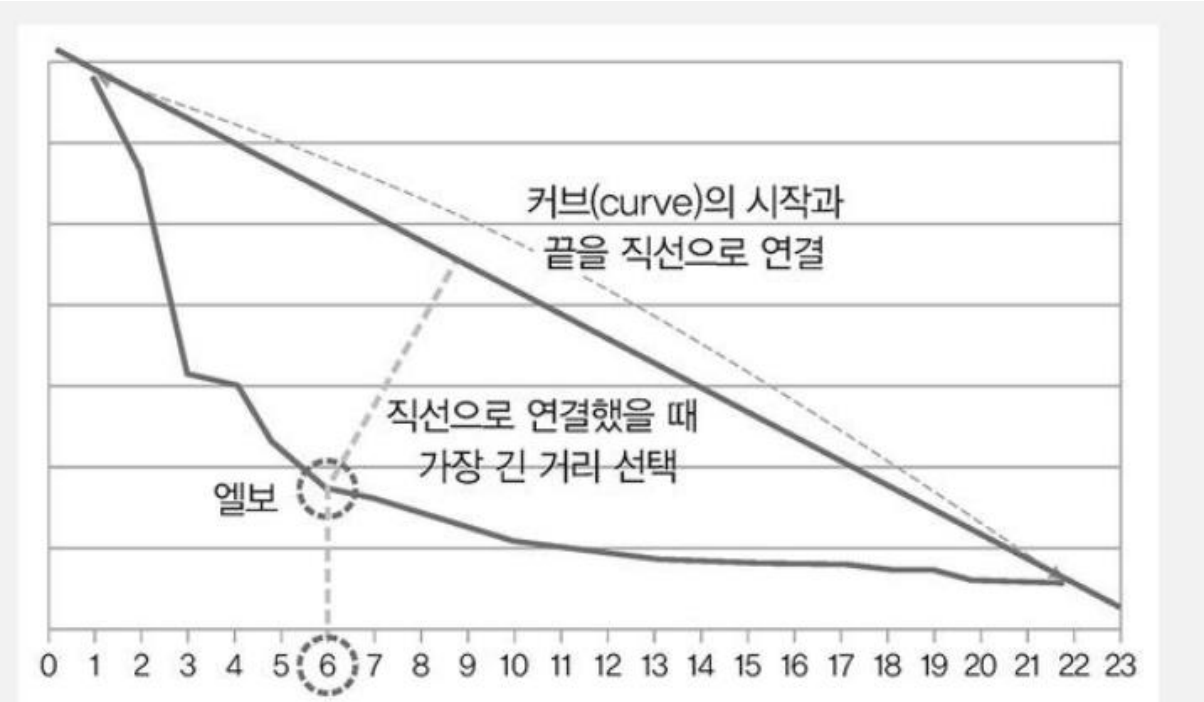
$$WCSS = \sum_{C_k}^{C_n} \left(\sum_{d_i in C_i}^{d_m} \text{distance}(d_i, C_k)^2 \right)$$

C : 클러스터의 중심 값

d : 클러스터 내에 있는 데이터

2. Create_clusters: 위에서 정한 Kmeans값을 토대로 데이터 처리를 하는 단계.

클러스터링 결과와 모델을 저장하고, 클러스터 정보를 포함하는 새로운 열을 데이터프레임에 추가. 이 파일을 KMeans라는 파일로 저장함,



이제 학습할 모델 찾아야함.

Tuner.py : Random forest , xgboost 각각 사용.

GridSearch라는걸 쓰는데, 우리가 지정해준 parameter들의 후보군 조합 중에서 가장 best 를 찾아줌. (gridsearchCV가 이거) 다르게 말하면 각 하이퍼파라미터에 대해 가능한 값 범위를 정의하고 모든 가능한 조합의 그리드를 생성. Validation 세트에다가 하이퍼 파라미터 조합으로 모델을 훈련하고 평가. (하이퍼파라미터를 튜닝(optimization) 하는 이유는, 하이퍼 파라미터 설정에 따라 모델 성능이 상이하기 때문이다)

파라미터 vs 하이퍼 파라미터?

	Hyperparameter	Parameter
설명	초매개변수 모델 학습 과정에 반영되는 값 학습 시작 전에 미리 조정	매개변수 모델 내부에서 결정되는 변수 데이터로부터 학습 또는 예측되는 값
예시	학습률 손실 함수 배치 사이즈	정규분포의 평균, 표준편차 선형 회귀 계수 가중치, 편향
직접 조정 가능	O	X

https://velog.io/@emseoyk/%ED%95%98%EC%9D%B4%ED%8D%BC%ED%8C%8C%EB%9D%BC%EB%AF%B8%ED%84%B0-%ED%8A%9C%EB%8B%9D 참고