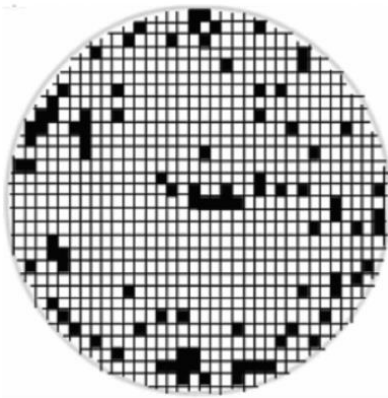


웨이퍼 불량 여부 확인 방법 – Wafer Bin Map



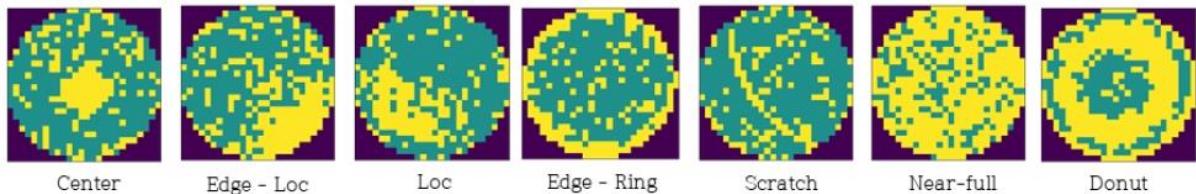
(WBM)

가장 많은 불량 – 웨이퍼 패턴 불량

Ex) 가장자리 고리 모양 패턴 – 에칭 공정 문제

원형 및 스크래치 패턴 – CMP 공정 문제

웨이퍼 패턴 불량 종류 (WBM 기준 불량 패턴)



- center : 불량 패턴이 웨이퍼 정 가운데에 위치한 경우
- edge : 불량 패턴이 웨이퍼 가장자리에 위치한 경우
- scratch : 불량 패턴이 얇은 선의 형태로 위치한 경우
- edge-ring : 불량 패턴이 안이 고리 형태 혹은 고리의 일부의 형태를 가질 경우
- local : 불량 패턴이 불록하며(Convex) 두께 있는 덩어리 형태를 가질 경우
- edge-local : 불량 패턴이 웨이퍼 가장자리에 위치하며, 불록하며(Convex) 두께 있는 덩어리 형태를 가질 경우
- none : 웨이퍼 맵 상에 군집을 이룬 불량 칩이 없을 경우
- near-full : 웨이퍼 맵 상에 군집을 이룬 불량 칩이 대부분 일 경우
- random : 웨이퍼 맵 상에 군집을 이룬 불량 칩이 near-full보다 적고 none보다 많을 경우

패턴 원인

- center type : due to abnormality of RF(Radio Frequency) power, abnormality in liquid pressure, due to velocity on the outside of the wafer is fast.

- local type : due to slit valve leak, abnormality during robot handsoff or abnormality in the pump.
- random type : due to contaminated pipes, abnormality in showerhead, or abnormality in control wafers.
- scratch type : mainly due to abnormality during robot handsoff or wafer impacts.
- donut type : Plasma-induced oxide damage can be modelled as damage produced by electrical current or voltage stress. Plasma processing causes MOSFET parameter degradation, from which one can deduce the plasma charging current. The effect of plasma etching on silicon-oxide interface reliability is also presented. The interface traps generated by plasma processing can be passivated with a forming gas anneal.
- edge-ring type : due to polishing pressure, the velocity of the polishing head and the table, the processing time for the wafer, the chemical reactions occurring on the wafer surface, the hydrodynamic condition of the slurry, and the properties of the pad.

머신러닝 알고리즘

1. K-means : 머신러닝 비지도 학습에 속하는 알고리즘, 데이터를 K개의 군집으로 묶는 역할
 - 각 군집의 평균을 활용하여 K개의 군집으로 묶으며, 각 클러스터의 중심과 데이터들의 평균 거리를 활용함
 - > 클러스터링에 사용 : 상대적으로 유사한 항목들을 찾아 클러스터를 생성하고 각 클러스터에 대해 별도의 모델을 훈련시킴
 - => 데이터 클러스터링에 사용
2. Random forest : decision tree 모델 여러개를 훈련시켜 그 결과를 종합해 예측하는 앙상블 알고리즘
 - 각 tree 모델을 훈련시킬 때 배깅(bagging) 방식을 사용
3. XGBoost : Gradient boosting 프레임워크의 최적화된 구현으로 설계된 전체 오픈소스 라이브러리
 - 과적합 방지를 위한 기법이 추가된 알고리즘 / 회귀와 분류 문제를 모두 지원 / 성능과 자원 효율이 좋음

2,3 -> 각 클러스터에 대해 학습

=> 예측 모델 구축

4. knn imputer : 가까운 이웃의 수를 정하고 그 이웃들을 이용하여 결측치(null)를 채우는 방식

-> 데이터 전처리에 사용 : null값이 있는지 확인하고 해당값을 주변 k개 이웃의 평균으로 채우고, 표준편차가 0인 열은 제거 (열의 모든 값이 동일하면 모델 훈련에 의미가 없기 때문)

=> 누락된 값(null) 대체

5. grid search cv : 관심있는 매개변수들을 대상으로 가능한 모든 조합을 시도해 보며 최적 하이퍼 파라미터 튜닝을 하는 것

-> 학습한 데이터의 정확도를 비교하여 최적의 모델을 찾음

최적의 하이퍼파라미터 탐색