

과제 설명 정리

제품 개발 프로세스

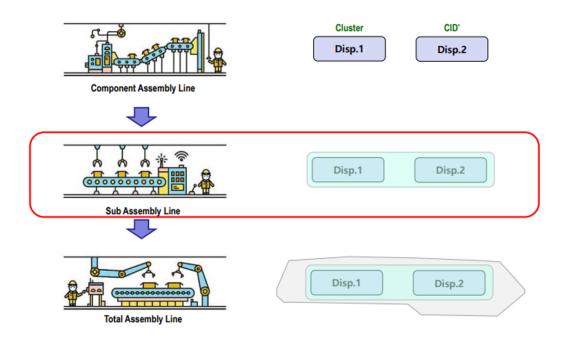
수주 → 개발 → 검증 → 생산

- 수주
- : 개발 차종 및 콘센트 결정되면 차량 주요 부품 공급업체 선정, 제품 공급사와 성능 및 가격 조율 등 협상 완료되면 수주가 확정됨
 - 개발
- : 수주 확정되면 본격적으로 제품 개발에 돌입
- 검증
- : 개발 업무 완료되면 제품에 대한 검증을 수행
- : 검증 영역은 크게 2가지로 구분. 가상검증과 실물검증
- 생산
- : 실물 검증을 무사히 통과하면 해당 제품을 정식으로 생산함
- : 부품을 조합하는 sub assembly line 그리고 최종제품 조립하는 total assembly line을 순차적으로 거치며 제품 만들어짐



제품 개발 프로세스 과정에서 발생하는 불량을 확인하고 주요 불량을 예측하는 AI 모델을 개발하는 것이 이번 해커톤 문제

일체형 디스플레이 제품 제작 프로세스



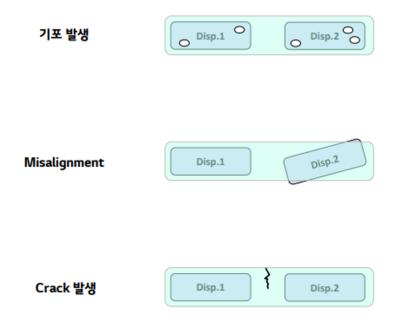
- 1. Componet Assembly Line에서 클러스터와 CID 디스플레이 각가 조립
- 2. Sub Assembly Line에서 두 디스플레이를 일체형으로 만들기 위해 대형 글라스에 두 개의 디스플레이를 합착하여 제품 형태를 하나의 세트로 만듬
- 3. Total Assembly Line은 완제품을 만드는 공정.



해커톤 문제는 바로 두번째 공정인 Sub Assembly Line에서 출력되는 데이터를 기반으로 출제

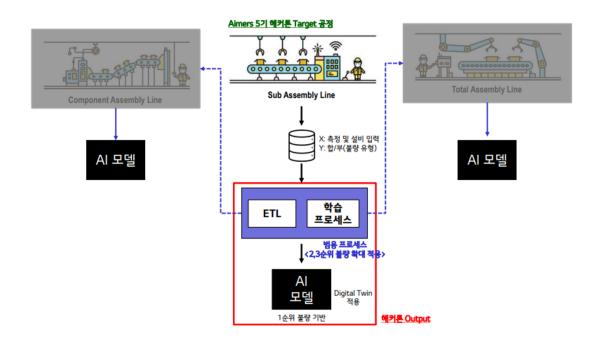
두번째 공정 역할을 기억하자

Sub Assembly Line에서 주로 발생하는 불량 유형



- 1. 기포가 발생하는 유형
- 2. Misalignment
- : 합착하는 과정에서 디스플레이가 조금만 비틀어져도 외관상 품질 불량 문제로 판정
- 3. 글라스에 크랙이 발생
- : 합착이나 경화과정에서 레진 수축으로 인해 글라스 영향을 줌

해커톤과 실무 적용에 생각하는 것



해당 데이터를 기반으로 **ETL**과 **학습 프로세스를 개발**하고 최종 AI 모델을 개발하게 될 것이다.

AI모델은 <u>불량 유형 중에 가장 빈도가 높은 것을 차지하는 것을 기반으로 ML 모델을 개발하</u>고 평가할 예정이다.

현업에서는 AI 모델의 유지보수 그리고 해당 프로세스를 확대 적용까지 고려해야 하기 때문에 ETL과 학습 프로세스가 범용 프로세스를 만들 수 있게 범용성을 갖추기를 기대하고 있다.

해커톤 문제 소개

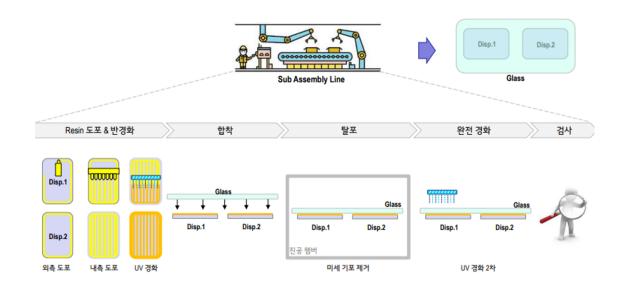
일부 공정에서 나오는 데이터를 활용해서 해당 데이터들을 기반으로 어떤 불량이 나오는지 그런 예측 모델을 개발하고 프로세싱하는 과제이다.

• 과제 배경

디스플레이 제품군이 이번 과제에 다룰 데이터

이번 과제에 해당되는 '원인 분석' 같은 경우에는 개별 공정이나 특정분야에서 나오는 데이터를 이용해 영향 인자를 파악하고 그 영향 인자를 이용해 불량률을 개선하고 새로운 방안을 제시할 수 있다.

• 과제 소개 및 목표



이번 과제에서 쓰일 데이터는 sub assembly line에서 나오는 데이터이다.

쉽게 말하면 글라스와 영상이 나오는 각각의 디스플레이들이 합착되어 sub assembly로 만들어지는 과정이라고 생각하면 된다.

Sub Assembly Line?

• 두 개의 디스플레이를 하나의 대형 글라스로 붙이는 공정

대형 글라스를 붙이는 과정 5단계

- 1. 글라스와 디스플레이 합착하기 위한 레진도포 & 반경화
- 2. 합착 공정
- 3. 미세 기포 제거하는 탈포 공정
- 4. UV로 완전 경화하는 공정
- 5. 검사 공정



이러한 공정들에서 나오는 다양한 데이터들을 활용하여 어떻게 하면 공정 중에서 발생하는 불량들을 잘 예측하고 판별할 수 있는 모델을 개발할까라는 문제 인식에서 출발했다.

주어지는 데이터에는 불량 발생의 원인이 될 수 있는 여러 인자들 즉 x값과 불량으로 판별되는 y값들이 있다. 제품 생산 중 불량 빈도가 가장 높은 불량을 y값으로 놓고 불량 예측 ai모델을 개발하고 프로세스화하는 것이다. 우리는 x값도 알고 y값도 알고 있다. 일반적으로 '지도학습'이라고 배웠을거다.

불량 예측 분류 모델을 개발하여 모델 성능을 높이고 불량 발생 여부를 예측해 보는 것이다. 성능 검증으로는 분류 모델에서 많이 쓰이는 **F1-score**를 쓰면 된다.

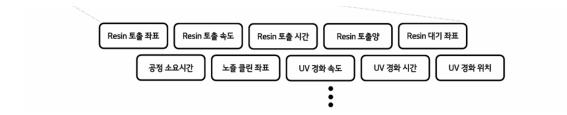
데이터 분석과 학습을 통해 불량 원인이 되는 주요 feature값들을 추출할 수 있을 것이고 이는 향후에 해당되는 불량 요인을 공정에서 조정하여 조치를 취할 수 있고 불량을 개선할 수 있을 것으로 보고 있다.

최종적으로는 불량률을 개선하고 이에 따라 제품 폐기 비용 감소로 인한 원가절감에 기여할수 있다. 정리해서 말씀드리면 해당 과제는 주어진 데이터로 좋은 성능을 내는 **불량 예측 ai** 모델을 개발 및 프로세스화 해야하고 주요 feature들이 무엇인지 확인할 수 있어야 한다.



지도학습 모델 사용 F1-score 사용

- 데이터 소개
- 1. 레진도포와 반경화

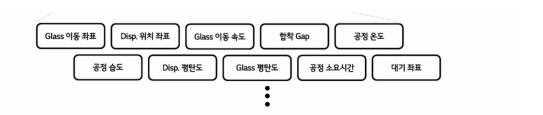


레진 도포에는 Dam이라고 불리는 **외측 도포**와 Fill이라고 불리는 **내측 도포**로 나뉘어 있으며 여기서 나오는 데이터로는 레진 토출 좌표 그리고 속도, 시간, 토출양 등이 나오고 대기 좌

표와 해당 노즐이 클린되는 좌표 등도 나오고 있다.

그리고 각 공정마다 기본적으로 공정 경화 시간과 위치도 데이터로 나온다.

2. 합착 공정

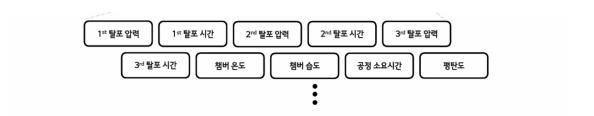


앞서서 디스플레이에 레진을 도포하고 경화시켰고 이번 공정은 글라스와 합착시키는 것이다. 여기서 나오는 feature로는 글라스의 이동 좌표, 속도, 디스플레이의 위치 좌표들이 있고 글라스와 디스플레이 합착 간극, 갭들이 있을 것이다.

제품의 평탄도도 볼 수 있으며 기본적으로 온도, 습도 등의 데이터들도 수집되고 있다.

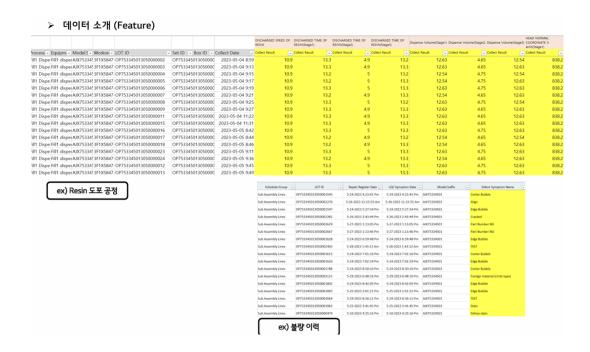
<u>해커톤 과제에서는 일부 공정 및 feature만 제공할</u> 수 있음을 인지해주기 바란다.

3. 탈포 공정



합착 후 미세 기포가 남아있을 수 있으므로 해당 공정에서 남아있는 기포를 제거하는 것이다. 적정 온도와 습도에 맞춰서 고압을 가하여 탈포하는 것이고 여기서 나오는 데이터들로는 첫번째, 두번째 등의 탈포 압력, 시간 등이 나온다. 그리고 진공 챔버에서의 온도, 습도, 공정소요시간, 평탄도 등도 수집되게 된다.

데이터 일부



시간별로 각 공정의 feature값들이 실시간으로 수집되고 있다.

<u>레진 도포 공정의 데이터는 화면에 보이는 것과 같으며 경화나 합착, 탈포 공정에서의 데이터도 이와 유사할 것이다.</u> 그리고 <u>불량으로 검출되는 이력들이 정리된 데이터도 아래와 같</u>이 데이터가 수집되고 있다.

위 데이터의 각 컬럼들을 조금 더 설명드리면 해당 프로세스는 fill1이라고 적혀있고 이는 내측 레진을 뜻하며 LOT ID, **Set ID는 제품 하나의 고유 넘버링**으로 보면 된다.

과제 주요 단계 간략히 설명

- 데이터 수집에 대한 절차를 거치지 않아도 된다.
- 모델링을 위한 **데이터 전처리**를 진행해야 한다.

전처리에는 결측치, 이상치, 중복데이터, 불균형데이터 처리 등이 있고 정규화, 스케일링 등과 같은 방법들이 있다. 각 팀에게 맞게 수행하면 된다!

다만 권고 드리는 방법은 데이터 전처리에 앞서 어느 정도 데이터 이해가 먼저 되어야 한다. 그래야 쓸 수 있는 데이터와 쓸 수 없는 데이터를 구분할 수 있고 필요하다면 시각화를 통해 분석과 해석을 진행해야 한다. 그 과정에서 특이점을 발견하거나 인사이트를 얻을 수도 있을 것이다.

데이터 이해 및 전처리가 완료되었다면 모델링을 진행하게 된다.

앞서 설명해드린 대로 y값인 불량 유형으로는 화면에 보이는 것과 같이 여러 불량들이 있을 것이다. 이물, 기포,, 등등 그 중에서 이번 과제에서는 불량 빈도가 가장 높은 것을 대상으로 학습을 진행하시면 되고 해당 불량에 따라 이진분류, 다중분류 등을 선택해서 수행하면 된다. 그 과정에서 해당 불량에 대해서 주요한 인자들이 무엇인지 추출할 수 있을 것이다.

모델 성능 향상 방법

모델 성능을 높이기 위해서는

- 데이터 전처리를 다시 하는 방법
- 교차검증
- hyperparameter tuning
- → 이 중 정답은 없다.

다만 잘못된 방법으로는

모델이 학습 데이터의 과적합 즉, overfitting되면

테스트 데이터의 일반화가 어려워질 수 있으므로 이 점은 유의하시면 된다.