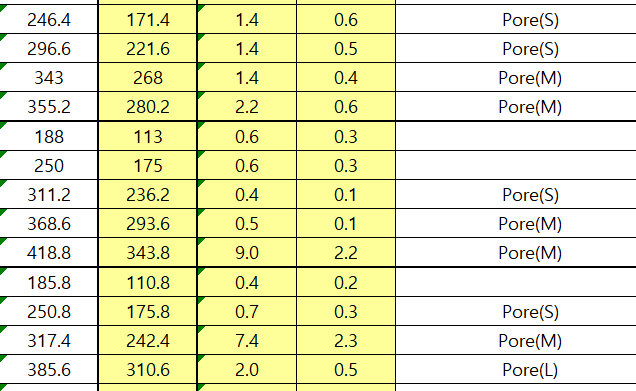
**현재 상황 문제점**

-엑셀에 시뮬레이션 데이터를 보면 표준편차가 낮은 데이터도 이상을 가지고 있는 경우가 대다수입니다.



-이 말은 단순히 이상을 가지고 있다고 전부 표준편차가 큰 것은 아님을 의미합니다.

(단, 표준편차가 클 경우엔 거의 모든 데이터에서 이상 결과가 나옵니다. 이러한 경우에는 Boxplot방식으로 표준편차가 큰 것을 이상치로 판별 가능합니다.)

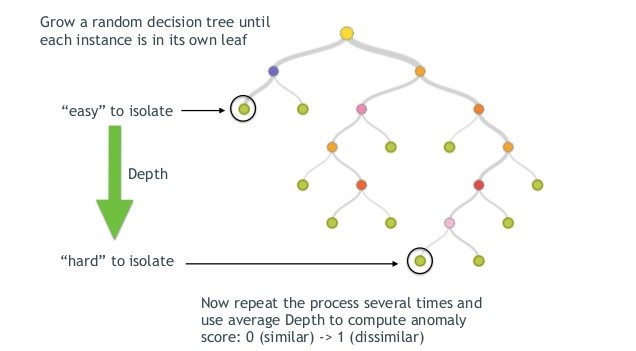
-따라서 다변량 이상치 분석을 하기 위해 **Isolation Forest, LOF, 의사결정 트리**를 사용할 수 있을 것 같은데, **Isolation Forest**와 **의사결정 트리**는 학습 데이터로 모델을 생성하는 방식이고 LOF는 직접 해보진 않았지만 거리 계산 기반인 것으로 알고 있습니다.

-다변량 분석을 하는 방식은 거의 나온 게 없어 **Isolation Forest**, **의사결정 트리**의 학습 모델을 만들거나 **LOF**를 활용하는 방법을 사용해야 할 것 같은데, 현실적으로 지금 주어진 시뮬레이션 데이터가 152개(측정불가 제외)로 매우 양이 적고 그 데이터 중 약 89개의 데이터가 비고란에 Pore혹은 번짐(이상 데이터)을 가지고 있습니다. 전체적인 데이터 양이 늘어나야 할 것 같으며, 정상인 데이터의 비율 또한 많아져야 할 것 같습니다.

-하단 부는 **Isolation Forest**와 **의사결정 트리**를 학습할 때, 어떤 데이터 집합(속성)들을 학습 데이터로 넣는지에 따라 나오는 정확도 결과를 테스트한 모습입니다.

-제가 생각 할 때, 하단부에 이상인 것을 정상이라고 판단하는 것의 대부분은 표준편차가 낮은데 이상을 가진 것을 표준편차가 낮으니 이상을 안가진 것으로 보아 틀린 답을 내놓는 것 같습니다.

**Isolation Forest**

* 다차원 데이터셋에서 효율적으로 작동하는 아웃라이어 제거 방법이다. Isolation Forest는 랜덤하게 선택된 Feature의 MinMax값을 랜덤하게 분리한 관측치들로 구성된다.
* 재귀 분할은 트리 구조로 나타낼 수 있으므로 샘플을 분리하는데 필요한 분할 수는 루트 노드에서 종결 노드까지의 경로 길이와 동일하다.
* 이상치에 대한 무작위 분할을 그 경로가 현저하게 짧아진다. 따라서 특정 샘프에 대해 더 짧은 경로길이를 생성할 때 아웃라이어일 가능성이 높다.

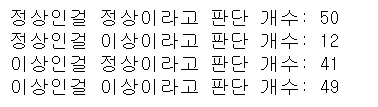
**Isolation Forest Test 결과**

Case1)

clf = IsolationForest(n\_estimators=100, contamination=0.60, random\_state=42)

learning\_data = credit\_data[['두께1','두께2','두께3','두께4','두께5','Gap','평균 두께','표준 편차','비고']]

-결과-

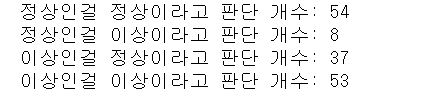


Case2)

clf = IsolationForest(n\_estimators=300, contamination=0.60, random\_state=42)

learning\_data = credit\_data[['두께1','두께2','두께3','두께4','두께5','Gap','평균 두께','표준 편차','시트 두께','비고']]

-결과-

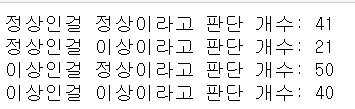


Case3)

clf = IsolationForest(n\_estimators=300, contamination=0.60, random\_state=1)

learning\_data = credit\_data[['P/B Ratio','바인더 함량','소포제 함량','점도','코팅 속도','Gap','표준 편차','평균 두께','시트 두께','비고']]# 비고 열을 없앤다.

-결과-

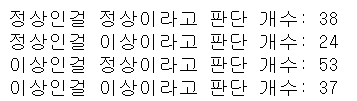


Case4)

clf = IsolationForest(n\_estimators=300, contamination=0.60, random\_state=1)

learning\_data = credit\_data[['P/B Ratio','바인더 함량','소포제 함량','점도','코팅 속도','실내 온도','실내 습도','Gap','비고']]

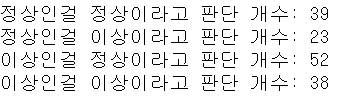
-결과-



Case5)

clf = IsolationForest(n\_estimators=300, contamination=0.60, random\_state=1)

learning\_data = credit\_data[['P/B Ratio','바인더 함량','소포제 함량','점도','코팅 속도','실내 온도','실내 습도','Gap','두께1','두께2','두께3','두께4','두께5','평균 두께','표준 편차','시트 두께','표준 편차.1','비고']]



**의사결정 트리 Test 결과**

Case1)

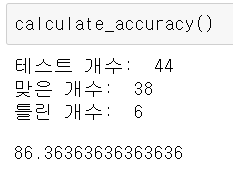
feature\_names = ["Gap", "average\_thickness", "standard\_deviation",

"lab\_temperature", "lab\_humidity"]

seed = 37

model = DecisionTreeClassifier(max\_depth=5,

random\_state=seed)



Case2)

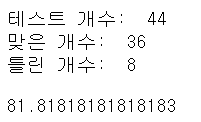
feature\_names = ["pbratio", "binder\_con", "defoamer","final\_temp",

"viscosity","coating\_speed","Gap","lab\_temperature","lab\_humidity"]

seed = 37

model = DecisionTreeClassifier(max\_depth=5,

random\_state=seed)



Case3)

feature\_names = ["pbratio", "binder\_con", "defoamer","final\_temp",

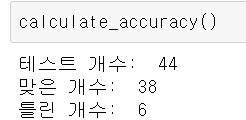
"viscosity","coating\_speed","Gap","lab\_temperature","lab\_humidity",

"average\_thickness", "standard\_deviation"]

seed = 37

model = DecisionTreeClassifier(max\_depth=5,

random\_state=seed)



Case4)

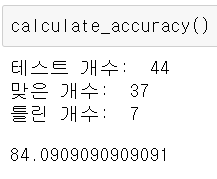
feature\_names =

['thickness1','thickness2','thickness3','thickness4','thickness5','sheet\_thickness','Gap','lab\_humidity','standard\_deviation']

seed = 37

model = DecisionTreeClassifier(max\_depth=5,

random\_state=seed)



**LOF (구현은 아직 하지 않았고 어떤 결과물로 나오는지만 보았습니다.)**

