반도체 제조 과정에서의 수율 향상을 위한 이진 분류 모델 비교 분석

CUAI 4기 스마트팩토리 B팀
임도연(소프트웨어학부), 유찬재(기계공학부), 최연찬(기계공학부)

[요약]
반도체 공정에서 효율성을 높이고자 하는 시도는 필수적이다. 따라서 본 연구에서는 UCI-SECOM 데이터에 전처리 및 스케일링을 적용하고, GridSearchCV를 통해 최적의 하이퍼 파라미터를 찾은 뒤, 이진 분류의 대표적인 5가지 모델로 학습시켰다. 이후 오차 행렬 기반의 평가 지표들을 통해 결과를 분석한 결과, GridSearchCV를 적용한 것이 더 높은 결과를 보이고 그 중 SVM 모델로 학습시킨 결과가 가장 뛰어남을 확인했다.

1. 서 론

복잡한 현대식 반도체 제조 공정은 일반적으로 센서 또는 공정 측정 지점에서 수집된 신호/변수를 모니터링하여 지속적으로 감시한다. 그러나 이러한 모든 신호가 특정 모니터링 시스템에서 동등하게 중요한 것은 아니다. 측정된 신호에는 유용한 정보, 관련 없는 정보 및 노이즈가 조합되어 있다. 따라서 엔지니어는 일반적으로 실제로 필요한 것보다 훨씬 많은 수의 신호를 얻는다. 각각의 신호들을 Feature로 고려할 경우, 수율과 가장 관련성이 높은 신호와 그렇지 않은 신호(노이즈)를 식별하기 위해 Feature 가공이 선행돼야 한다. 이후 가공된 신호들을 수율에 기여하는 주요 요인으로 활용한다. 이와 같은 과정을 통해 공정 처리량을 높이고 학습 시간을 단축하며 단위당 생산 비용을 절감할 수 있다.

본 논문은 반도체 제조과정에서의 특정 프로세스의 Pass/Fail을 예측하는 이진 분류 모델을 제안한다. 약 600개의 센서로 측정된 1600개 가량의 데이터를 가공한 후 머신러닝 알고리즘 중 분류에 특화된 XGBosst, SVM(Support Vector Machine), RandomForest, DecisionTree, Logistic Regression 모델로 학습시킨다. 모델들의 최적의 파라미터들을 적용하고 예측성능을 높이기 위해 GridSearchCV를 수행하여 모델들을 설계한다. 모델의 평가는 오차 행렬 기반의 정밀도(Precision score), 재현율(Recall score), F1_score를 이용하여 진행한다.

2. 본 론

1) 데이터 전처리

본 연구에서 활용한 데이터 셋은 반도체 제조(SECOM:Semiconducor Manufacturing) 특정 프로세스에서 센서를 통해 측정한 것으로, UCI Machine Learning Repository에서 제공한 것이다. 데이터는 각각 591개의 Feature를 가진 1567개의 example들로 구성돼있다. 591개의 Feature는 Time Stamp, 589개의 센서, 사내 라인 테스트에 대한 판정결과(Pass/Fail)로 이뤄져있다. 판정결과 Column의 '-1'은 Pass, '1'은 Fail을 나타낸다. 현실 데이터와 마찬가지로 이 데이터에는 개별 Feature에 따라 다양한 정도의 Null value들이 포함돼있다. 이는 데이터를 조사하고 전처리 혹은 기술들을 적용하는 과정에서 반드시 고려돼야 한다.
전처리에 앞서 데이터를 살펴본 결과 3가지 주요한 특징을 찾을 수 있었다.

1. 고유치가 1인 열들이 존재한다.
2. Pass/Fail 데이터 비율이 고르지 않다.
3. 데이터 값의 편차가 크다.

데이터에 Null값이 분포하고 있다는 점, 고유치가 1인 열들이 존재한다는 점을 고려하여 2가지 전처리 방법을 고안하였다.

1. 결측값이 900개가 넘는 열은 삭제하고 남은 결측값은 0으로 대체
2. 결측값이 50%가 넘는 열과 고유치가 1인 열은 삭제하고 남은 결측값은 앞, 뒤 행의 값으로 대체

이후, 두 가지 전처리 방법에 공통으로 Data Scaling을 진행하였다. 먼저, Pass/Fail 데이터 비율이 고르지 않은 점을 해결하기 위해 Oversampling을 적용해 데이터 불균형을 해소하였다. 실제로 코드를 구현해봄으로써 Oversampling을 적용한 것과 그렇지 않은 것과의 차이가 아주 크다는 것을 확인할 수 있었다. 다음으로, 데이터 간 편차를 조정하기 위해 StandardScaler를 적용했고, 각 Feature들의 평균을 0, 분산을 1로 만들어 모든 Feature들이 같은 스케일을 갖도록 표준화하였다. 마지막으로 상대적으로 많은 Feature들에 의해 과적합 이 발생하는 것을 방지하기 위해 PCA(주성분 분석)를 통해 차원을 축소하였다.

2) 데이터 모델링

2.1 XGBoost

XGBoost(eXtra Gradient Boost)는 트리 기반의 앙상블 학습에서 가장 각광받고 있는 알고리즘 중 하

나로 분류(Classification)와 회귀(Regression) 분석에 뛰어난 예측성능을 발휘한다. XGboost는 앙상블 모델 중 Boosting의 한 종류이다. 이때 Boosting이란 약한 분류기를 세트로 묶어 정확도를 예측하는 기법이다. XGBoost는 GBM(Gradient Boosting Model) 알고리즘의 단점을 보완해주고자 나온 만큼, GBM 대비 빠른 수행시간, Tree pruning(나무 가지치기)를 통한 과적합 규제(Regularization)의 장점을 가진다.
먼저 본 연구는 이진 분류를 목적으로 하고 있기에 eval_metric을 'error'로 설정하여 모델을 설계하였다. 또한 트리의 깊이를 제한하는 파라미터인 max_depth를 조절하여 과대 적합을 방지하고 모델의 성능을 개선하고자 하였다. 이에 grid search cross validation을 이용하여 최적의 max_depth 값에 대한 탐색을 수행하였다. max_depth 값의 범위는 2부터 12까지 설정하였으며, 교차 검증의 cv 파라미터를 10로 설정하여 탐색을 진행하였다. 교차 검증 결과 전처리 방법 1은 max_depth가 12, 전처리 방법 2는 max_depth가 8일 때 최적의 평균 정확도 0.973, 0.978을 보임에 따라 XGBoost에 적용할 max_depth의 값은 각각 12, 8로 선정하여 설계하였다.

2.2 SVM

SVM은 데이터 분석, 패턴 인식 등 다양한 목적을 위해 사용하는 지도 학습 알고리즘 중 하나로 분류(Classification)나 회귀(Regression) 분석에 사용되며 특히 분류 쪽에 성능 뛰어나다. SVM은 서포트 벡터(Support Vectors)를 사용해서 결정 경계(Decision Boundary)를 정의하고, 분류되지 않은 점을 해당 결정 경계와 비교해서 분류하게 된다. SVM은 커널 서포트 벡터 머신(Kernalized Support Vector Machine)이라고도 불리는데, 이때의 커널이란 새로운 특성을 만들지 않고 고차원 분류기를 학습시킬 수 있도록 한 것으로, 주어진 데이터를 고차원의 특징 공간으로 사상해 원래의 차원에선 보이지 않던 선형(초평면:Hyperplane)이 데이터를 분류할 수 있도록 한 것이다. 본 연구에서는 RBF (Radial Basis Function) 커널을 사용하여 모델을 설계하였다.
SVM은 결정 경계가 어떻게 정의되는지에 따라 성능의 차이가 결정되며, 최적의 결정 경계를 찾는 것이 중요하다. 최적의 결정 경계를 찾기 위해 적용되는 파라미터는 C와 gamma가 있다. C는 결정 경계의 마진을 조절하여 이상치의 허용 범위를 조절하는 파라미터이며, gamma는 결정 경계의 유연성을 조절하여 모델의 과대적합을 방지하는 파라미터이다. Grid search cross validation을 이용하여 최적의 결정 경계를 구하기 위해 c와 gamma의 최적값에 대한 탐색을 수행하였다. 파라미터 C와 gamma의 범위는 0.001, 0.01, 0.1, 1, 10, 100, 1000으로 설정하였으며, 교차검증의 cv 파라미터를 10로 설정하여 탐색을 진행하였다. 교차 검증 결과 C가 10, gamma가 0.01일 때 각각의 전처리 방법에 따른 최적의 평균 정확도가 0.995, 0.997로 가장 높은 값을 보임에 따라 SVM에 적용할 C와 gamma의 값을 10과 0.01로 선정하여 설계하였다.

2.3 RandomForest

RandomForest는 동일한 알고리즘으로 여러 분류기를 만든 후 보팅을 통해 최종 결정을 한다. 이는 다른 앙상블 알고리즘들 보다 빠른 수행 속도를 가지고 또한 다방면에 걸쳐 높은 예측 성능을 보이고 있다. 또한 RandomForest는 DecisionTree를 기반으로 한 부트스트래핑 방식으로 데이터 세트를 중첩되게 분리한다.
먼저 RandomForest 학습모델의 n_estimators를 100으로 통일시켜 변수를 줄였다. 또한 트리의 성장을 제한해 과대 적합을 방지하고 모델의 성능을 개선하고자 max_depth를 조절하였다. 이에 grid search cross validation을 이용하여 최적의 max_depth 값에 대한 탐색을 수행하였다. max_depth 값의 범위는 20부터 60까지 설정하였으며, 교차 검증의 cv 파라미터를 10로 설정하여 탐색을 진행하였다. 교차 검증 결과 전처리 방법 1은 max_depth가 60, 전처리 방법 2는 max_depth가 52일 때 최적의 평균 정확도 0.983, 0.987을 보임에 따라 RandomForest에 적용할 max_depth의 값은 각각 60, 52로 선정하여 설계하였다.

2.4 DecisionTree

DecisionTree는 ML 알고리즘 중 직관적으로 이해하기 쉬운 알고리즘으로, 학습을 통해 데이터에 있는 규칙을 찾아내 트리(Tree) 기반의 분류 규칙을 만드는 것이다. 결정 트리는 매우 쉽고 직관적으로 파악이 가능하다는 장점이 있다. 또한 정보의 균일도만 신경쓰면 되므로 특별한 경우를 제외하고는 각 피처의 스케일링과 정규화 같은 전처리 작업이 필요 없다. 하지만 많은 규칙으로 트리의 깊이가 깊어진다면 과적합으로 결정 트리의 예측 성능이 저하될 수 있음을 유의해야 한다.
트리의 성장을 제한하여 과적합을 방지하기 위해 트리의 최대 깊이를 규정하는 max_depth를 사용하였다. 이에 grid search cross validation을 이용하여 최적의 max_depth 값에 대한 탐색을 수행하였다. max_depth 값의 범위는 20부터 60까지 설정했으며, 교차 검증의 cv 파라미터를 10로 설정하여 탐색을 진행하였다. 교차 검증 결과 전처리 방법 1은 max_depth가 24, 전처리 방법 2는 max_depth가 32일 때 최적의 평균 정확도 0.847, 0.848을 보임에 따라 RandomForest에 적용할 max_depth의 값은 각각 24, 32로 선정하여 설계하였다.

2.5 LogisticRegression

로지스틱 회귀는 선형 회귀 방식을 분류에 적용한 알고리즘이다. 그렇기 때문에 로지스틱 회귀는 분류에 사용되며 선형 회귀 계열이다. 여기서 가중치(weight) 변수가 선형인지 아닌지에 따라 회귀가 선형 또는 비선형으로 결정된다. 로지스틱 회귀는 학습을 통해 선형 함수의 회귀 최적선을 찾는 선형 회귀와는 달리 시그모이드(Sigmoid) 함수 최적선을 찾고 이 시그모이드 함수의 반환 값을 확률로 간주해 확률에 따라 분류를 결정한다. 로지스틱 회귀는 가볍고 빠르다는 점과 이진 분류 예측 성능이 뛰어나 이진 분류의 기본 모델로 자주 사용된다.

본 연구에서 사용된 데이터 또한 이진 분류를 기반으로 하고 있기 때문에 이진 분류에 뛰어난 성능을 보이는 로지스틱 회귀를 사용해 결과를 도출했다. 여기서 로지스틱 회귀의 성능을 높이기 위해 LogisticRegression클래스의 주요 하이퍼 파리미터인 penalty와 C를 사용했다. penalty를 통해 규제의 유형을 설정하고 C를 이용하여 규제 강도를 조절하고자 했다. grid search cross validation을 이용하여 하이퍼 파라미터 peanlty와 C를 최적화하는 값을 탐색했다. penalty는 l2, l1으로 설정하고 C는 0.01, 0.1, 1, 5, 10으로 설정하고 cv는 10으로 하여 진행했다. 그 결과 전처리 방법 1에 대해서는 C는 1, penalty는 l2가 나왔고 전처리 방법 2에 대해서는 C는 10, peanlty는 l2가 도출되었다. 평균 정확도 또한 0.790, 0.807이 나옴에 따라 위와 같이 하이퍼 파라미터를 설계했다.

3) 데이터 평가 및 분석

앞 장에서 전처리 및 학습을 진행한 모델들을 이용하여 분류 성능 평가를 진행하였다. 성능 평가는 오차 행렬을 기반으로 Precision score, Recall score, F1_score를 사용하였다. 표 1은 앞서 말한 두 가지의 전처리 방법, GridSearchCV 적용 유무를 바탕으로 각각의 오차 행렬을 참고하여 계산한 분류 모델들의 성능 지표이다.

표 1: 분류모델 성능 지표

Class		전처리 방법 1		전처리 방법 2	
		Default	GridSearchCV	Default	GridSearchCV
XGBoost	F1_score	0.9746	0.9838	0.9785	0.9798
	Recall	0.9529	0.9681	0.9604	0.9654
	Precision	0.9529	0.9681	0.9604	0.9654
SVM	F1_score	0.9797	0.9890	0.9615	0.9917
	Recall	0.9679	0.9945	0.9330	0.9972
	Precision	0.9679	0.9945	0.9330	0.9972
Random Forest	F1_score	0.9850	0.9891	0.9890	0.9903
	Recall	0.9810	0.9811	0.9864	0.9944
	Precision	0.9810	0.9811	0.9864	0.9944
DecisionTree	F1_score	0.8678	0.8682	0.8534	0.8514
	Recall	0.8164	0.8215	0.8103	0.8251
	Precision	0.8164	0.8215	0.8103	0.8251
Logistic Regression	F1_score	0.8022	0.8023	0.8211	0.8251
	Recall	0.7871	0.7871	0.8021	0.8034
	Precision	0.7871	0.7871	0.8021	0.8034

표 1의 예측 성능 결과를 바탕으로 크게 3가지 관점으로 나눠 분석을 진행했다.

3.1 전처리 방법에 따른 예측성능

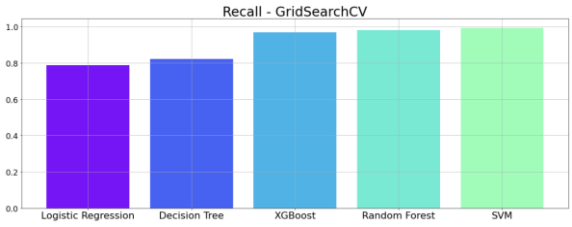
본 연구는 크게 2가지 전처리 방법으로 모델을 설계하였다. 본래 단순히 결측값만을 제거한 전 처리 방법 1보다 고유치도 고려한 전처리를 방법 2의 예측성능이 더 높은 경향을 보일 것을 기대했지만, 실제로는 전처리 방법에 따른 경향을 보이지 않았다. 이는 전처리 과정에서 결측값 및 고유치가 1인 열을 가공하는 방법만 달리했을 뿐 이후의 스케일링 과정은 동일하게 진행했기 때문에 각각의 방법 간의 큰 차이가 없었던 것으로 보인다. 비록 경향성은 뚜렷하게 보이진 않았지만 두 전처리 방법 모두 최대 0.98-0.99의 정밀도, 재현율, F1_score를 보임에 따라 두 전처리 방법 모두 모델 설계에 있어 효과적임을 확인했다.

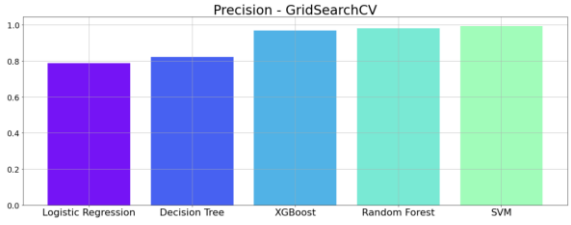
3.2 GridSearchCV 적용 유무에 따른 예측 성능

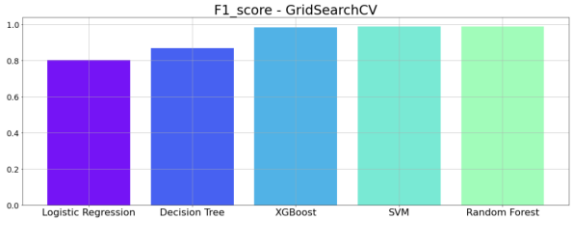
본 연구에서는 예측 성능을 높이고자 GridSearchCV를 통해 최적의 파라미터를 찾고 이를 바탕으로 학습을 진행하였다. 위의 표를 바탕으로 분석한 결과, 모든 학습 알고리즘에서 GridSearchCV를 적용한 것이 그렇지 않은 것보다 더 높은 예측 성능을 보임을 확인 할 수 있었다. 같은 전처리 방법 및 알고리즘이라 할 지라도, 하이퍼 파라미터 조정만으로도 충분히 예측성능을 향상시킬 수 있음을 직접 확인했다.

3.3 알고리즘에 따른 예측성능 및 최적의 알고리즘

본 연구에서는 이진 분류의 대표적인 알고리즘 5가지로 모델을 학습시켰다. XGBoost, SVM, RandomForest로 학습시킨 모델들은 모두 0.950을 상회하며 높은 예측 성능을 보였지만 Decision Tree, Logistic Regression에서는 0.800 정도의 비교적 낮은 예측 성능을 보였다. 이를 참고하여 추후에 이진분류 모델을 설계할 시 다음의 3가지 알고리즘(XGBoost, SVM, RandomForest)을 우선적으로 고려한다면 좋을 결과가 기대된다.
다음으로 모델들간의 예측성능을 비교한 결과, GridSearchCV를 통해 최적의 하이퍼 파라미터를 찾고 SVM 알고리즘으로 학습시킨 모델이 F1_score 0.9917, Recall score 0.9972, Precision score 0.9972의 결과로 최상의 예측성능을 보였다.







3. 결 론

본 연구에서는 센서를 통해 수집된 신호/변수 데이터를 활용해 반도체 제조 특정 프로세스의 Pass/Fail을 예측하는 이진 분류 모델을 만들었다. 데이터 전처리 과정에서 결측값을 900개 이상인 Column을 제거하는 방법 1과 결측값의 비율이 50% 이상인 Column과 고유치가 1인 Column을 삭제하는 방법 2로 나누어 진행하였다. 이 후 데이터 불균형을 해결하고자 Oversampling, StandardScaler, PCA 차원축소를 적용해 데이터 스케일링을 하였다. 결괏값을 비교해봄으로써 데이터의 Target으로 하는 Feature가 한쪽으로 편향돼 있다면 Oversampling 및 Undersampling의 스케일링이 필수적임을 확인하였다. 이 후, 이진 분류의 대표적인 5가지 알고리즘 XGBoost, SVM, DecisionTree, RandomForest, LogisticRegression으로 모델을 학습시키고 Confusion Matrix 기반의 평가 지표를 바탕으로 결과를 확인하였다.
평가 및 결과를 분석한 결과 다음의 세 가지를 확인해 볼 수 있었다. 먼저 전처리 방법에 따른 결과 추이가 보이지 않았다는 것이다. 본래 예측성능을 높이기 위해 가장 먼저 고려돼야 하는 것이 데이터 전처리이다. 하지만 본 연구에서는 결측값 및 고유치가 1인 열을 가공하는 방법만 달리했을 뿐 이후의 스케일링 과정은 동일하게 진행했기 때문에 전처리 방법 간의 큰 차이 없이 두 방법 모두 대체로 좋은 예측성능을 보였다.
다음으로 GridSearchCV를 통해 각각 모델 알고리즘의 최적의 하이퍼 파라미터를 찾고 이를 적용하여 학습한 것이 그렇지 않은 것보다 예측성능이 더 높게 나오는 경향성을 보였다. 동일한 전처리 방법 및 알고리즘이라 할지라도, 하이퍼 파라미터 조정을 통해 충분히 예측성능이 향상할 수 있음을 직접 확인했다.
마지막으로 본 연구에서 가장 좋은 예측성능을 보인 것은 전처리 방법 2를 적용한 뒤 GridSearchCV로 교차검증을 하고 SVM 알고리즘으로 학습시킨 모델로 F1_score는 0.9917, Recall score는 0.9972, Precision score는 0.9972의 결과를 보였다.
위 연구를 바탕으로 이진 분류 모델을 설계하여 SECOM(Semiconductor Manufacturing) 데이터에 적용한다면 반도체 공정 처리량을 높이고 단위당 생산 비용을 절감해 생산량이 효율적으로 향상할 수 있음을 기대한다.

참고 문헌

1) 정용진.이종성.오창헌, “PM10 예측 성능 향상을 위한 이진 분류 모델 비교 분석”, 한국정보통신학회 논문지, Jan, 2021
2) 권철민, 파이썬 머신러닝 완벽 가이드
3) UCI SECOM Dataset.Available: UCI SECOM Dataset | Kaggle
4) XGBoost.Available: [ML/DL] XGboost의 정의와 구현 및 hyper parameter 설정 — 나무늘보의 개발 블로그 (tistory.com)
5) 커널 서포트 벡터 머신.Available: 2.3.7 커널 서포트 벡터 머신 | 텐서 플로우 블로그 (Tensor ≈ Blog) (tensorflow.blog)