누수 예측 모델 구축

: SVC와 XGBoost를 활용하여

우영석  
*인공지능융합학과   
성균관대학교 소프트웨어융합대학*2024711029

*Abstract*— 이 프로젝트는 상하수도의 진동 데이터를 분석하고 이에 기반하여 누수를 예측하기 위한 머신러닝을 진행합니다. 핵심 작업은 진동 신호를 다섯 가지 구별 가능한 범주로 분류하는 다중 클래스 분류입니다. EDA, 전처리와 하이퍼파라미터 튜닝을 수행하여 Support Vector Classification와 XGBoost의 성능을 최적화하였습니다. SVC의 경우 선행연구보다 최적의 하이퍼파라미터 조합(gamma='scale', kernel='rbf'과 C=‘100’)을 탐색해냈습니다. 하지만 하이퍼파라미터 튜닝을 거친 XGBoost의 예측성능(94.30% > 90.37%)에는 못미쳐 예측 모델로 최종선정되지 못했습니다. 본 프로젝트의 결과는 실시간으로 다양한 누수 상태를 자동으로 식별하는 업무에 활용될 수 있을 것입니다..

Keywords—누수 감지, XGBoost, Support Vector Classification, EDA, Data Preprocessing

1. Introduction

본 프로젝트는 상하수도 데이터를 분석하고 누수를 예측하는 모델을 학습하는 것입니다. 누수 상태의 수도관에서 감지되는 진동과 일반적인 상태의 수도관에서 감지되는 진동에 대한 데이터가 있다면 인공지능을 통해 누수 상태 여부를 실시간으로 탐지해낼 수 있습니다. 따라서 누수 상태가 5개의 클래스로 라벨링된 상하수도 데이터를 분석하고 머신러닝을 진행하여 누수 상태를 탐지할 수 있는 모델을 훈련하였습니다.

본 프로젝트의 진행과정은 다음과 같습니다. 먼저 Data로드하고 EDA를 통해 데이터의 내용과 구조를 이해합니다. 둘째, 데이터에 대해 이해한 바를 가지고 데이터를 전처리합니다. 셋째, SVC와 XGBoost의 다양한 하이퍼파라미터 조합에 대해 교차검증을 실시하여 최적의 조합을 찾아냅니다. 넷째, 최적의 조합을 기반으로 훈련된 두 모델의 성능을 비교하여 보고합니다.

1. Body

1. 모델 설명

본 프로젝트에서는 두 가지 머신러닝 모델(SVC, XGBoost)을 기반으로 다중 클래스 분류 작업을 수행하였습니다.

* SVM (Support Vector Machine) 및 SVC (Support Vector Classification)

SVM은 선형 또는 비선형 분류 문제와 회귀 문제에 모두 활용될 수 있습니다. SVC는 SVM을 구현한 클래스 중 하나로, 특히 분류를 위해 설계되었습니다. 선형 SVM은 두 클래스 간의 가장 큰 마진을 찾는 방식으로 작동합니다. 학습 데이터 중 경계에 위치한 데이터 포인트들만이 모델의 결정 경계를 결정하는데 영향을 미치며, 이러한 샘플들을 '서포트 벡터'라고 부릅니다.

하드 마진 분류는 모든 데이터가 마진 바깥에 올바르게 분류되어야 하며, 이는 데이터가 선형적으로 구분될 수 있어야 하고 이상치에 민감하지 않아야 한다는 문제가 있습니다. 이를 해결하기 위해 소프트 마진 분류를 사용하여 일부 마진 오류를 허용하면서도 경계의 폭을 가능한 넓게 유지합니다. 소프트 마진 정도는 하이퍼파라미터 ‘C’를 통해 조절할 수 있습니다.

비선형 데이터셋을 다룰 때는 커널 SVM을 사용할 수 있습니다. 커널 트릭을 통해 실제로는 특성을 추가하지 않으면서 다항식 특성을 추가한 것과 같은 효과를 내어, 복잡한 데이터셋을 더 잘 분류할 수 있게 합니다. 대표적인 커널로는 다항식 커널과 가우시안 RBF(Radial Basis Function) 커널이 있습니다. 가우시안 RBF 커널은 특히 각 샘플의 영향 범위를 조절하는 하이퍼파라미터 ‘gamma’를 가지고 있습니다. 후에 모델링 과정에서 앞서 언급한 하이퍼파라미터인 ‘C’, ‘kernel’, ‘gamma’을 튜닝하였습니다.

* XGBoost

XGBoost (eXtreme Gradient Boosting)는 그레디언트 부스팅을 기반으로 한 고성능, 고속의 실행과 병렬 처리를 지원하는 라이브러리입니다. XGBoost는 과적합 방지를 위한 정규화 기능을 포함하여 다양한 시스템에서 빠르게 실행할 수 있도록 설계되었습니다. 또한, 결측치 처리, 트리 가지치기, 레벨별 트리 분할 방식 등을 지원하여 효율적인 학습이 가능합니다.

XGBoost는 분산 컴퓨팅을 지원하여 큰 데이터셋에서도 효율적으로 학습할 수 있으며, 스케일링이 용이합니다. 이 알고리즘은 정확도 향상뿐만 아니라 학습 과정의 속도를 개선하기 위해 많은 최적화가 이루어졌습니다. 데이터를 여러 구간으로 나누어 각 구간에서 가장 좋은 분할을 찾는 근사 알고리즘을 사용하며, 이는 대규모 데이터셋을 빠르게 처리할 수 있도록 돕습니다.

2. 코드 분석 및 결과 분석

- EDA & Data preprocessing

스크린샷, 텍스트, 도표, 그래프이(가) 표시된 사진

자동 생성된 설명

위 그래프는 훈련 데이터셋의 각 라벨의 빈도를 막대 그래프로 나타낸 것입니다. 각 라벨을 설명하면 ‘normal’은 정상음, ‘out’은 외부누수, ‘in’은 내부누수, ‘other’은 환경음, ‘noise’는 기계음입니다. 라벨 별로 샘플 수가 큰 차이를 보이는 것을 알 수 있습니다. 이러한 불균형은 모델의 학습 과정에 영향을 줄 수 있습니다. 따라서 모델의 하이퍼파라미터를 검증하기 위한 데이터셋을 구축할 때 라벨 별 샘플 비율을 지켜야 학습이 잘 이뤄졌는지 평가가 제대로 이뤄질 수 있습니다.

그래프, 라인, 텍스트, 도표이(가) 표시된 사진

자동 생성된 설명

위 그래프는 상하수도 데이터에서 각 라벨별 주파수 대응 진동수(스펙트럼 밀도)를 0Hz부터 5120Hz까지 시각화한 것입니다.[1] 0Hz에서는 모든 데이터가 0이므로 학습 시에는 배제해야 합니다.

train\_df.drop('0HZ', axis=1, inplace=True)

test\_df.drop('0HZ', axis=1, inplace=True)

또한 대다수의 주파수 범위에서는 라벨 간에 큰 차이를 보이지 않으며, 비슷한 경향을 나타냅니다. 따라서 주파수 데이터에서 나타나는 평균값을 추가적인 특성으로 활용하여 모델의 성능을 향상시키고자 했습니다.[2]

frequency\_columns = [f"{i}HZ" for i in range(10, 5130, 10)]

train\_df['mean\_frequency'] = train\_df[frequency\_columns].mean(axis=1)

텍스트, 그래프, 스크린샷, 도표이(가) 표시된 사진

자동 생성된 설명

위 그래프는 각 누수 검사 시기별로 감지된 최대 Hz 값을 나타낸 것입니다. ‘normal’과 ‘noise’, ‘in’ 라벨은 다른 라벨들과 확연히 다른 값을 가집니다. 하지만 ‘other’과 ‘out’ 라벨은 일부 시점에서만 차이를 보입니다. 따라서, 각 시점에서 감지된 최대 Hz 값의 총합을 데이터에 추가하여 모델의 성능을 향상시킬 필요가 있습니다.[2]

max\_hz\_cols = [f"MAX{i}" for i in range(0, 19, 2)]

train\_df['sum\_max\_hz'] = train\_df[max\_hz\_cols].sum(axis=1)

텍스트, 스크린샷, 그래프, 라인이(가) 표시된 사진

자동 생성된 설명

위 그래프는 각 누수 검사 시기별로 감지된 최대 진동수를 나타낸 것입니다. 누수 검사 시기별 최대 Hz와 유사하게 ‘other’과 ‘out’ 라벨의 값이 육안으로 어떤 차이를 가지는지 구분하기 힘듭니다. 그 차이를 명확히 하는 데이터도 추가하기 위해 모든 시점의 값을 총합한 값도 열로 추가했습니다.

max\_amp\_cols = [f"MAX{i}" for i in range(1, 20, 2)]

train\_df['sum\_max\_freq'] = train\_df[max\_amp\_cols].sum(axis=1)

지도학습을 진행하기에 앞서 타겟값인 ‘label’ 열을 숫자로 인코딩해줘야 합니다. 따라서 라벨 인코딩을 통해 훈련 데이터셋과 테스트 데이터셋을 라벨 인코딩하였습니다.

label\_encoder = LabelEncoder()

y\_train\_encoded = label\_encoder.fit\_transform(y\_train)

y\_test\_encoded = label\_encoder.transform(y\_test)

또한 앞서 살펴본 대로 많은 데이터가 굉장히 다른 범주의 값들을 가지므로 모델이 일부 데이터에 편향되어 훈련되지 않도록 하기 위해서는 데이터 표준화를 거쳐야 합니다. 따라서 StandardScaler를 통해 표준화를 거쳤습니다.[2]

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

* Modeling & Results

먼저 SVC 모델을 GridSearchCV를 통해 교차검증과 하이퍼파라미터 튜닝을 실시했습니다. 첫 번째 GridSearchCV 시도에서는 gamma, kernel, C의 다양한 조합을 테스트하였습니다. 이 과정에서 gamma='scale'과 kernel='rbf' 조합이 가장 우수한 성능을 나타내었습니다.

두 번째 시도에서는 최적의 gamma와 kernel 설정을 유지한 상태에서 ‘C’ 값을 바꿔서 수행했습니다. ‘C’ 값에 대한 탐색 범위를 10부터 900까지 확장하여 다양한 정규화 강도에서 모델의 성능을 평가하였습니다. 아래와 같이 C=‘100’에서 최고의 성능을 보였습니다. 따라서 본 프로젝트에서 SVC의 최적의 하이퍼파라미터 조합은 gamma='scale', kernel='rbf'과 C=‘100’입니다.

Best cross-validation accuracy: 0.8974245552669287

0.861 for {'C': 10}

0.897 for {'C': 100}

0.895 for {'C': 500}

0.891 for {'C': 900}

본 프로젝트에서 실시한 그리드서치를 통해 도출된 SVC의 하이퍼파라미터 조합은 선행 연구[2]에서 사용된 조합과 비교했을 때, 미세하게 더 높은 예측 성능을 보였습니다. 교차 검증 시 평균 정확도(89.79% > 89.56%)와 테스트 데이터에 대한 정확도(90.37% > 90.17%) 모두 보다 나은 결과를 보였습니다. 하지만 XGBoost의 예측 성능(데스트 데이터에 대한 정확도: 94.30%)은 뛰어넘지 못해 최종 선정되진 못했습니다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

그 다음 모델로 XGBoost 를 Hyperopt를 통해 하이퍼파라미터를 베이지안 최적화 기법을 통해 튜닝했습니다. 본 프로젝트는 5개의 클래스에 대해 분류하는 작업을 머신러닝하는 것으로 ‘multi:softmax’ 목적함수로 설정하고 평가 지표로 ‘mlogloss’로 설정하였습니다. max\_depth, min\_child\_weight, gamma, reg\_alpha, reg\_lambda, subsample, colsample\_bytree, learning\_rate에 대해서는 XGBoost 공식문서를 참고해서 탐색범위를 설정하였습니다.[3]

from hyperopt import hp, fmin, tpe, STATUS\_OK, Trials

xgb\_search\_space = {

'max\_depth': hp.quniform("max\_depth", 3, 10, 1),

'min\_child\_weight': hp.quniform('min\_child\_weight', 1, 10, 1),

'gamma': hp.uniform('gamma', 0.0, 0.5),

'reg\_alpha': hp.quniform('reg\_alpha', 0, 1, 0.1),

'reg\_lambda': hp.uniform('reg\_lambda', 1, 100),

'subsample': hp.uniform('subsample', 0.5, 1),

'colsample\_bytree': hp.uniform('colsample\_bytree', 0.5, 1),

'learning\_rate': hp.uniform('learning\_rate', 0.01, 0.3),

}

총 100회의 시도 끝에, 교차 검증 정확도가 가장 높은 하이퍼파라미터 조합은 다음과 같이 결정되었습니다. 이때의 5-fold CV를 통한 교차검증 정확도는 약 0.947[[1]](#footnote-1)입니다.

max\_depth: 10

min\_child\_weight: 1

gamma: 0.156

reg\_alpha: 0.2

reg\_lambda: 4.633

subsample: 0.908

colsample\_bytree: 0.501

learning\_rate: 0.300

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

최종적으로 테스트 데이터셋에 대한 예측을 수행하고 정확도를 계산했습니다. XGBoost모델은 약 94%의 정확도를 달성했으며, 클래스 별 성능도 SVC보다 모두 높은 수준으로 나타났습니다. F1-score를 기준으로 정리하면 “클래스 0(‘normal’) – 0.92 > 0.86, 클래스 1(‘out’) 0.88 > 0.79, 클래스 2(‘in’) 1.00 = 1.00, 클래스 3(‘other’) 0.89 > 0.80, 클래스 4(‘noise’) 0.93 > 0.89”입니다.

1. Conclusion

이 프로젝트는 SVC와 XGBoost를 사용하여 상하수도의 진동 데이터를 통해 누수를 탐지하는 모델을 구현하고 비교하였습니다. EDA를 통해 Data Preprocessing을 철저히 진행하였습니다. SVC는 GridSearch를 총 2회 실시하여 선행연구보다 더 최적의 하이퍼파라미터 조합을 탐색해냈습니다. XGBoost는 Hyperopt를 통해 하이퍼파라미터 튜닝을 실시했습니다. 그 결과 최적의 SVC이 보여준 예측 성능보다 XGBoost가 우수한 성능을 보였습니다.

본 프로젝트는 데이터의 정보 누출에 대한 한계가 있습니다. AIhub의 상수관로 누수 감지 데이터는 라벨 별로 별도의 파일로 저장되어 있습니다. 그러나 본 프로젝트는 데이터 로드 후에 랜덤하게 섞어주는 과정을 거치지 않았습니다. 즉, 모델이 유사한 데이터를 순서대로 학습하게 한 한계가 있습니다. 두 모델 모두 테스트 데이터에 대해 충분한 성능을 보였지만 데이터 셔플을 실시한 후에 모델 훈련을 했다면 보다 나은 성능을 보일 수 있습니다.

1. References

[1] 김병학, 전제성, 이수안. (2022). 상수관로 진동 센서 데이터를 이용한 누수 감지 머신러닝 모델. *한국정보과학회 학술발표논문집, 2022(제주)*. 1361- 1363.

[2] 최준규, 임성빈. (2023). 누수 잡음 크기 스펙트럼을 이용한 SVM 기반의 상수관로 누수 감지 및 분류. *전자공학회논문지*, *60*(2), 6-14, 10.5573/ieie.2023.60.2.6

[3] XGboost Official Documentation. (2024, June, 15). <https://xgboost.readthedocs.io/en/stable/python/python_api.html>

1. 모든 부동소주점 수치는 소수점 넷째자리에서 반올림했습니다. [↑](#footnote-ref-1)