EEG 기반의 알코올 중독 분류

: FFT 변환과 XGBoost를 중심으로

서울과학기술대학교 산업정보시스템전공

22101911 왕현식, 데이터 전처리 및 분석, 발표자료 제작 및 발표

22101942 한다은, 데이터 전처리 및 분석, 보고서 작성

요약

본 프로젝트에서는 Kaggle의 EEG 데이터를 사용하여 FFT(고속 푸리에 변환)와 XGBoost 알고리즘을 사용하여 알코올 중독 여부를 분류하는 머신 러닝 모델을 개발했다. 데이터 전처리 과정에서는 데이터 분할, 리쉐이프, 노치 필터, FFT 및 밴드 파워 계산을 진행했다.

모델링 과정에서는 XGBoost, CatBoost, Random Forest, Voting Classifier를 사용한 앙상블 모델을 비교하여, 그 중 가장 높은 검증 정확도를 보인 XGBoost 모델을 선정했다. 정확도 뿐만 아니라 혼동 행렬과 ROC 곡선을 통해 모델의 성능을 평가했고, 그 결과 두 클래스에서 균형 잡힌 성능을 가지고 있음을 확인했다.

최종 제출한 모델은 Kaggle에서 0.88333의 정확도를 달성하였으며, 매우 높은 정확도를 보이지는 못하는 한계가 있다. 이 결과는 EEG 신호 처리에서 도메인 지식과 적절한 데이터 전처리의 중요성을 시사한다.

1. 서론

본 프로젝트의 목적은 kaggle에서 제공된 EEG 데이터를 활용하여 알코올 중독 여부를 분류하는 머신 러닝 모델을 개발하고 성능을 평가하는 것이다.

2. 사용 데이터

kaggle에서 제공된 알코올 중독과 정상 상태의 EEG 측정 데이터가 train 데이터와 test 데이터로 나누어 제공되었다.

2.1 데이터의 형태

데이터는 'input'과 'label' 두 개의 키를 가진 딕셔너리 형태로 제공되었다.

2.1.1. input

Input 데이터의 형태는 (1080, 256, 64, 1)이다. 이는 256hz동안 64개의 전극에서 측정된 1080개의 샘플 데이터를 의미한다.

2.1.2. label

Label 데이터의 형태는 (1080,)이다. 1080개의 샘플 중 540개는 정상 데이터, 540개는 알코올 중독 데이터로 균형 데이터셋임이 확인되었다.

3. 분석 방법

3.1. 데이터 전처리

3.1.1. 데이터 분할

1080개의 샘플 데이터를 test_size = 0.2를 적용하여 분할했다. Label의 형태가 1과 0이 차례로 정렬된 형태이기 때문에 shuffle=True로 설정했다. 그 결과 X_train.shape: (864, 256, 64, 1), X_valid.shape: (216, 256, 64, 1)의 형태가 되었다.

3.1.2 데이터 reshape

(864, 256, 64, 1) 형태의 train 데이터를 squeeze하여 (864, 256, 64)의 형태로 만든 다음, 시간축의 정보를 보존하기 위해 전치시킨 후 평탄화했다. 그 결과 (864*64, 256)의 형태로 변환했다. Valid 데이터에도 같은 과정을 적용하여 (216*64, 256)의 형태로 변환했다.

변화한 input데이터의 형태에 맞추어 label 데이터도 확장했다. 각각의 데이터를 64번씩 반복하게 하여 train label: (55296,), valid label: (13824,)의 형태로 변환했다.

3.1.3. notch 필터 적용

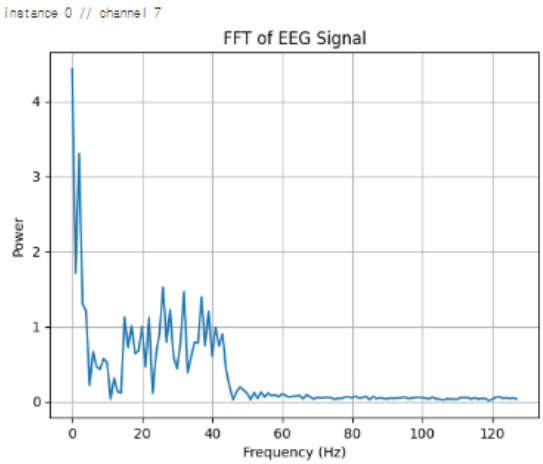
노치 필터(notch filter)는 특정 주파수를 제거하거나 억제하는 데 사용된다. EEG 데이터와 같은 신호 데이터를 분석할 때, 노이즈 제거를 위해 특정 주파수의 잡음을 제거하기 위해 사용했다. 노치 필터는 시간 도메인에서 특정 주파수 성분을 제거하는 필터이기 때문에 푸리에 변환 단계 전에 사용했다.

제거한 주파수 대역은 50hz, 60hz이다. 이 두 대역을 필터링한 이유는 EEG 데이터를 측정할 당시 전력선 잡음을 제거하기 위해서이다. 제공받은 데이터가 어느 국가의 데이터인지 알 수 없으므로, 국제 표준에 해당하는 50Hz, 60Hz를 선택했다. 60Hz는 주로 미국, 캐나다, 멕시코, 한국, 필리핀 등의 전력선 잡음 주파수이고, 50Hz는 주로 유럽과 아시아 일부 국가, 아프리카, 오세아니아 등의 전력선 잡음 주파수이다.

3.1.4. FFT(고속 푸리에 변환) 적용

푸리에 변환은 시간이나 공간 영역의 함수를 주파수 영역으로 변환하는 것이다.

FFT 변환 후 결과를 시각화하여 전체적으로 살펴본 결과 5Hz~50Hz에서 대부분 신호가 튀는 모습을 보였다. 아래는 첫번째 인스턴스의 6번째 채널을 FFT 변환한 것을 시각화 한 결과이다.



3.1.5. band power 함수 적용

초기에는 FFT 변환 후 신호가 튀는 부분을 필터링하여 역 푸리에 변환을 통해 다시 시간 영역으로 변환하려고 했으나, 분류 문제이기 때문에 꼭 시간 영역으로 돌릴 필요가 없다고 생각했다.

그래서 FFT 변환의 결과에 band power함수를 적용했다. Band power는 EEG 데이터에서 특정 주파수 대역의 전력(power)를 측정하는 것이다.

처음에는 대표적인 뇌파 대역인 델타파(0.5~4Hz), 세타파(4~8Hz), 알파파(8~13Hz), 베타파(13~30Hz), 감마파(30Hz 이상) 대역으로 세분화하여 각 대역의 평균 파워를 계산했다. 그러나 이후 모델 적용과 평가 단계에서 성능이 잘 나오지 않아 각 대역들을 더 세분화했다.

점점 세분화하다가 아예 모든 대역을 일정한 간격(1Hz)으로 나눠서 모델을 적용하니 성능이 약간 올라가는 것을 확인할 수 있었다. 그래서 1Hz 간격으로 모든 주파수 대역을 세분화하고, 각 대역의 평균 파워를 계산하는 함수를 정의한 후 푸리에 변환을 적용한 train데이터와 valid데이터에 각각 적용했다.

3.1.6. Scaling

데이터 스케일링을 위해 StandardScaler를 사용하여 band power함수가 적용된 train 데이터와 test 데이터를 스케일링했다.

3.2. Modeling

모델링 과정에서는 여러 종류의 알고리즘을 사용해 보고 validation accuracy가 가장 높은 알고리즘을 선택했다.

3.2.1. XGBoost

먼저 XGBoost 모델을 사용한 이유는 부스팅 모델 중 높은 예측 성능을 가지고 있고, 하이퍼파라미터로 과적합을 방지하는 규제도 추가할 수 있기 때문에 효과적일 것이라고 생각했다.

XGBoost 모델은 속도가 빠르기 때문에 training accuracy와 validation accuracy를 확인하면서 직접 하이퍼파라미터를 튜닝했다. 그 결과 가장 높은 성능을 보인 하이퍼파라미터 조합은 다음과 같다.

n_estimators	300	gamma	1.5
max_depth	9	reg_lambda	5
Learning_rate	0.1	reg_alpha	5
subsample	0.5	min_child_weight	8
Colsample _bytree	1		

위의 하이퍼파라미터 조합으로 모델링했을 때, train accuracy: 0.89, valid accuracy: 0.71의 성능을 보였다.

3.2.2. CatBoost

CatBoost도 예측 성능이 높고 과적합을 방지할 수 있으며, 대칭 트리 구조를 사용하기 때문에 비교적 빠른 예측 속도를 제공한다고 알게 되어서 사용해 보게 되었다.

Iterations	1000	loss_function	Log loss
depth	6	verbose	0
Learning_rate	0.03	border_count	32
l2_leaf_reg	3		

위의 하이퍼파라미터 조합으로 모델링했을 때, train accuracy: 0.79, valid accuracy: 0.71의 성능을 보였다.

3.2.3. Random Forest

앞의 두 모델은 부스팅 모델을 사용했으니 배깅 모델도 사용해 봐야겠다고 생각하여 random foreset를 사용하게 되었다.

Random Forest의 경우 GridsearchCV를 이용하여 최적의 하이퍼파라미터 조합을 찾았다.

n_estimators	300	min_samples_split	5
max_depth	12	min_samples_leaf	6

위의 하이퍼파라미터 조합으로 모델링했을 때,

train accuracy: 0.83, valid accuracy: 0.67의 성능을 보였다.

3.2.4. Ensemble

VotingClassifier를 이용하여 앞에서 사용했던 개별 모델들(XGBoost, CatBoost, Random Forest)을 결합하여 하나의 앙상블 모델을 만들었다. 옵션으로는 각 개별 모델의 예측값을 투표로 합산하여 최종 예측값을 결정하는 hard voting을 사용했다. 이때 train accuracy: 0.83, valid accuracy: 0.70의 성능을 보였다.

3.2.5 최적 모델 선택

여러 알고리즘을 적용하고 validation accuracy가 가장 높은 모델을 선정한 결과 최종적으로 XGBoost가 선택되었다.

3.3. Model evaluation

3.3.1. 혼동 행렬

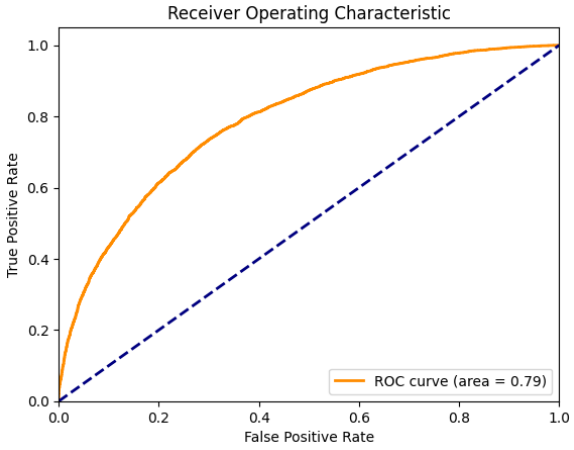
더 다양한 방법으로 모델을 평가하기 위해 모델로 예측된 valid 라벨과 진짜 valid 라벨을 이용하여 혼동 행렬을 확인했다. 확인 결과는 다음과 같다.

	False	True
False	TN = 4896	FP = 2016
True	FN = 1900	TP = 5012

전체적인 정확도가 매우 좋지는 않지만 모델이 양성과 음성 클래스 모두에서 상당히 균형잡인 성능을 보임을 알 수 있다.

3.3.2. ROC curve

모델의 성능을 한 눈에 알아볼 수 있도록 roc 커브를 시각화한 결과는 아래와 같다.

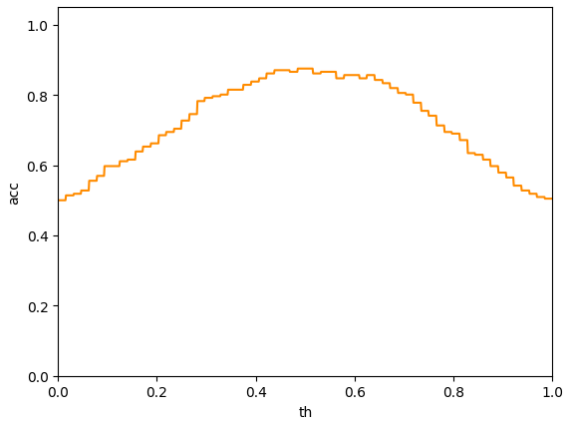


3.4. Voting

3.4.1. 임계값 결정

앞 단계에서 나온 정확도와 혼동 행렬, roc커브는 데이터 전처리 단계에서 64개로 확장한 라벨에 대한 결과이다. 따라서 다시 원래의 데이터 형태로 변환해주어야 한다. 64개씩 1그룹으로 묶어 해당 그룹의 평균이 임계값 이상이면 라벨을 1로 판단하게 한다.

임계값을 결정하기 위해 임계값을 0부터 0.001씩 증가시키면서 결정된 valid 라벨과 실제 valid 라벨로 accuracy를 측정했다. 그 결과 임계값이 0.5일 때 valid accuracy 0.875로 가장 높았다. 이를 한눈에 알아보기 쉽게 시각화하면 다음과 같다.



3.4.2. 최종 예측 데이터의 혼동행렬

Voting까지 마친 최종적인 예측 validation label 216개와 진짜 validation label을 활용하여 혼동 행렬을 나타냈다. 다음은 차례로 임계값이 0.45, 0.5, 0.55일 때의 혼동행렬이다.

임계값: 0.45	False	True
False	TN = 104	FP = 4
True	FN = 24	TP = 84

임계값: 0.5	False	True
False	TN = 99	FP = 9
True	FN = 18	TP = 90

임계값: 0.55	False	True
False	TN = 88	FP = 20
True	FN = 9	TP = 99

임계값이 0.5보다 작을 때는 negative label을 잘 맞추지 못하고, 임계값이 0.5보다 클 때는 positive label을 잘 맞추지 못함을 알 수 있다.

따라서 voting 단계에서는 64개로 이루어진 1개의 그룹에 대해, 라벨의 평균이 0.5 이상이면 그 그룹의 라벨을 1로 결정한다.

4. 최종 결과

앞의 모든 단계를 거쳐서 최종적으로 120개의 예측 라벨을 만들고 이를 csv파일로 변환하여 kaggle에 제출했다. 그 결과 정확도 0.88333을 얻을 수 있었다.

높은 정확도는 아니지만 valid accuracy = 0.875, test accuracy = 0.883이므로 오버피팅 문제는 일어나지 않았을 것으로 추측한다.

5. 개선할 점

5.1. 도메인 지식

프로젝트 발표 때 다른 조들의 진행 방법을 들어보니 대다수가 PSD를 사용했다는 것을 알게 되었다. 분석 과정에서 푸리에 변환과 band power등을 사용하긴 했지만, 도메인 지식이 부족하여 제대로 활용하지 못한 것 같다. EEG나 신호 처리 분야에 대해 더 이해한 상태에서 체계적으로 접근해야 한다고 느꼈다.

5.2. 접근방법

이번 프로젝트의 경우 처음에 (1080, 256, 64,1) 데이터를 먼저 모델에 넣을 수 있는 형태로 만드는 것에 집중했던 것 같다. 그래서 바로 평탄화를 적용했는데, 이 단계가 조금 성급했던 것 같다는 생각이 들었다.

처음에 전처리할 때 64개의 채널을 다 쓰는 것이 아니라 bad channel을 확인하고 필터링했어야 한다고 생각한다. 전처리 단계에서 노이즈가 제거되지 않았기 때문에 그 뒤의 FFT나 band power를 적용하는 과정에서 대표적인 뇌파 대역으로 세분화해 봤을 때 성능이 증가하지 않은 것 같다.

5.3. 모델 선택

물론 모델을 잘 선택하고, 하이퍼파라미터를 잘 선택하는 것도 꼭 필요한 일이지만, 이번 프로젝트에서는 모델의 종류나 하이퍼파라미터를 고민하는 것보다는 전처리 결과에 따라 결과값이 바뀌는 것 같았다.

또한 이번 프로젝트에서는 배깅, 부스팅 등 앙상블 모델을 활용했는데, 전처리를 어떻게 했느냐에 따라 로지스틱 회귀 같은 비교적 간단한 모델로도 분류가 될 수 있음을 알았다.

6. 참고문헌

[1]https://youtu.be/Mc9PHZ3H36M?si=ktE-nDG1s1krq2x2

[2] David Lee, Hee-Jae Lee, Sang-Goog Lee. (2014). Motor Imagery EEG Classification Method using EMD and FFT

[3] https://www.mdpi.com/2076-3417/12/11/5413

[4] https://www.mdpi.com/2076-

[5] 대한뇌파연구회. (2017). 뇌파분석의 기법과 응용: 기초에서 임상연구까지.