

## Modeling for Flood Prediction: Diverse Applications and Evaluation of Machine Learning and Deep Learning Approaches

김민석<sup>1)</sup>, 박수영<sup>1)</sup>, 이윤서<sup>1)</sup>, 한리<sup>2)</sup>

### Abstract

The application and evaluation of various machine learning and deep learning approaches in modeling for flood prediction are addressed in this study. Flood prediction has emerged as a critical issue due to the increase in climate change and extreme weather events, making the development of accurate prediction models essential. To this end, several machine learning and deep learning techniques were used to construct flood prediction models, and the performance of each approach was compared.

In basic regression analysis, there was little difference in R<sup>2</sup> scores between cases where normalization and dimensionality reduction techniques were applied and those where they were not. For machine learning models, cases without normalization and dimensionality reduction techniques achieved higher R<sup>2</sup> scores. Similarly, in deep learning models, the performance difference based on the application of these techniques was minimal. Finally, among various ensemble techniques applied, the Voting ensemble demonstrated the best performance.

The results indicate that while appropriate use of normalization and dimensionality reduction techniques can enhance the prediction accuracy of models, this is not always the case. By using and comparing the performance of various machine learning and deep learning models, meaningful analysis is provided in this study.

- 
- 1) All authors have equal contribution, Department of Statistics and Data Science, Chung-Ang University, Seoul 06974, Korea
  - 2) All authors have equal contribution, Department of Global Innovative Drug, Chung-Ang University, Seoul 06974, Korea

## 1. 서론

전 세계적으로 홍수, 이상기온, 사막화 같은 기후변화의 결과로 자연재해의 빈도가 증가하고 있다. 예를 들어, 2021년 독일과 벨기에를 강타한 홍수는 수백 명의 목숨을 앗아갔고, 수십억 달러의 경제적 피해를 초래했다. 홍수는 일반적으로 많은 비가 내려 강물이 넘치는 현상으로 발생한다. 이러한 현상은 많은 강수뿐만 아니라 강의 자연적 요인과 하천 관리 능력 등의 인위적인 요인들에 의해 발생한다. 이러한 요인들은 기후변화에 따라 더욱 악화될 수 있다.

일반적으로 홍수라고 하면 하천 홍수(River Flood)만을 고려하지만, 2022년 8월 강남역 일대 침수사고와 같은 도시 홍수(Urban Flood)도 전 세계적으로 증가하고 있는 추세다. 도시 홍수는 비가 지면으로 흡수되지 않아 발생한다. 대부분의 홍수는 예상치 못한 많은 강수량이 주요 요인이지만, 피해 확률이 높은 지역을 미리 예측할 수 있다면 경제적 피해는 물론 인명 피해도 줄일 수 있다. 홍수는 자연재해 중 가장 많은 사상자를 내는 재해일 뿐만 아니라, 홍수 이후 수인성 전염병이라는 추가적인 피해도 발생한다. 따라서 홍수 확률이 높은 지역을 예측하고 미리 대비책을 세우는 것이 매우 중요하다.

이러한 이유로 우리는 Kaggle 데이터 경진대회 “Regression with a Flood Prediction Dataset”에 참여하게 되었다. 대회의 최종 목적은 사회적 요인, 자연환경 등 다양한 요인을 분석하여 홍수 발생 확률을 예측하는 것이다. 이를 통해 우리는 회귀분석을 비롯한 머신러닝과 딥러닝 기법을 이용하여 예측 모델을 최적화하는 알고리즘을 개발하고자 한다. 사용되는 데이터는 장마 강도, 지형 배수 상태, 강 관리 상태 등의 다양한 변수를 포함하고 있다.

이 대회를 통해 우리는 다양한 모델을 비교 평가하고, 예측 정확도를 높이기 위한 최적의 방법을 찾는 것을 목표로 하며, 최종 결과는  $R^2$  score를 통해 평가된다.

## 2. 데이터 설명

### 2.1 데이터 구성

Kaggle의 "Regression with a Flood Prediction Dataset" 대회에서 제공하는 데이터는 Train, Test, Submission 데이터로 구성된다. Train 데이터는 [표 1]과 같으며, Test 데이터는 'Flood Probability' 칼럼을 제외하고 Train 데이터와 동일한 칼럼 이름을 가지고 있다. Submission 데이터는 최종 모델의 학습 결과로 얻어진 홍수 발생 확률 값을 추가하여 대회에 제출하는 데이터다.

[표 1] Train 데이터

ID	Monsoon Intensity	Topography Drainage	River Management	...	Political Factors	Flood Probability
0	5	8	5	...	3	0.445
1	6	7	4	...	3	0.450
2	6	5	6	...	3	0.530
3	3	4	6	...	5	0.535

[표 1]은 Train 데이터를 요약한 표다. 자연적 요인과 사회적 요인을 범주형 변수로 나타냈으며, 총 22개의 칼럼으로 구성되어 있다. Train 데이터셋에는 1,117,957개의 ID가 있으며, 각 칼럼의 의미는 [표 2]에 제시되어 있다.

[표 2] 데이터 셋 칼럼의 의미

Column	Mean
ID	임의 ID
Monsoon Intensity	장마 강도
Topography Drainage	지형 배수
River Management	강 관리
Deforestation	산림 파괴
Urbanization	도시화
Climate Change	기후 변화
Dams Quality	댐 품질
Siltation	퇴적물 쌓임
Agricultural Practices	농업 관행

Encroachments	침범
Ineffective Disaster Preparedness	비효과적인 재난 대비
Drainage Systems	배수 시스템
Coastal Vulnerability	연안 취약성
Landslides	산사태
Watersheds	유역
Deteriorating Infrastructure	악화된 인프라
Population Score	인구 점수
Wetland Loss	습지 상실
Inadequate Planning	불충분한 계획
Political Factors	정치적 요인
Flood Probability	홍수 가능성

각 칼럼은 기후, 지형, 인구 밀도, 강 관리 상태 등 다양한 변수들을 포함하고 있으며, 이를 통해 홍수 발생 확률을 예측할 수 있다. 이러한 변수들은 모델의 성능을 높이기 위해 중요한 역할을 한다.

## 2.2 데이터 전처리

### 2.2.1 StandardScaler (표준화)

StandardScaler는 데이터셋의 각 특징들을 표준화하는 방법이다. 표준화는 각 특징의 평균을 0, 표준편차를 1로 조정하여 이상치가 있는 데이터셋의 경우에도 차이를 크지 않게 조정해준다. 이를 통해 모델의 성능을 높이는 데 도움을 준다.

[그림 1] StandardScaler 적용식

$$z = \frac{x - \mu}{\sigma}$$

The original value .....  $x$  ..... The mean  
The standard deviation .....  $\sigma$

### 2.2.2 PCA (주성분 분석)

PCA는 고차원의 데이터를 저차원으로 줄이는 방법이다. 상관된 원래 변수를 주성분이라는 새로운 비상관 축으로 변환하면서 분산을 최대한 유지하여 노이즈와 불필요한 정보를 제거한다. 이로 인해 차원의 저주를 해결하고 모델의 과적합을 줄여 성능을 높일 수 있다.

PCA를 진행하는 방식은 다음과 같다. 먼저, 평균을 계산한 후 데이터 중심화를

진행하여 데이터의 분포를 조정한다. 그 다음 공분산(Covariance) 행렬을 계산하고, 고유값과 고유벡터를 도출한다. 마지막으로 가장 큰 고유값을 갖는 고유벡터로부터 원하는 수의 주성분을 선택하여 최종적으로 PCA를 수행한다.

각 변수를 분석한 결과, 흥수 예측에 필요한 변수들 중 유의미하지 않은 변수는 없는 것으로 나타났다. 따라서 모든 변수를 종합적으로 고려해야 한다. 또한, 파생 변수를 만들어 분석할 경우 예측력이 향상된다는 점도 중요하다. 변수들의 값이 커질수록 흥수 확률이 높아진다는 점을 고려하여, 모든 값을 합산한 'Sum'이라는 새로운 파생변수를 만들어 분석에 활용했다.

이를 바탕으로 두 가지 방식으로 모델을 구축했다. 첫 번째 방식은 PCA와 StandardScaler를 적용하여 데이터의 표준화와 차원 축소를 통해 모델의 성능을 최적화하는 방법이다. 두 번째 방식은 PCA와 StandardScaler를 적용하지 않고 원본 데이터 그대로를 활용하여 모델을 구축하는 방법이다.

이런 두 가지 방식을 비교 평가하여 최적의 예측 성능을 도출하고자 한다.

### 3. 모델 설명

#### 3.1 회귀분석

회귀분석은 연속형 변수들에 대해 하나 또는 여러 개의 독립변수와 종속변수 사이의 상관관계를 분석하여 적합도를 측정하는 방법이다. 단순 회귀분석은 두 변수 사이의 선형 관계를 설명하는 분석이며, 회귀분석의 일반적인 모형은 [그림2]와 같다.

[그림 2] 회귀분석의 일반항

$$Y = a + bX + \epsilon$$

[그림 2]는 회귀분석의 일반적인 모형을 나타낸 것이다. 이 모형은 잔차와 각 변수들의 회귀계수 및 오차항을 기본적인 형태로 포함하고 있으며, 오차항은 정규분포를 가정한다. 회귀분석을 통해 변수들 간의 관계를 이해하고, 예측 모델의 성능을 평가할 수 있다.

회귀 분석은 다음과 같은 과정을 거친다. 먼저, 데이터 셋에서 독립변수와 종속변수를 정의한다. 그런 다음, 독립변수들이 종속변수에 미치는 영향을 분석하여 회귀계수를 계산한다. 마지막으로, 잔차를 분석하여 모델의 적합도를 평가한다.

회귀분석은 단순 회귀분석 외에도 다중 회귀분석, 로지스틱 회귀분석 등 다양한 형태가 있다. 다중 회귀분석은 여러 독립변수가 종속변수에 미치는 영향을 동시에 분석하며, 로지스틱 회귀분석은 이진 종속변수에 대한 예측에 사용된다.

#### 3.2 머신러닝 기법

##### 3.2.1 Gradient Boosting

Gradient Boosting은 앙상블(Ensemble) 기법 중 하나로서, 잔차에 대한 기울기를 이용하여 잔차가 줄어드는 방향으로 모델을 학습하는 기법이다. 경사하강법을 기반으로 하여 과적합을 줄이면서 분류, 예측 등의 기계 학습 성능을 높일 수 있다 (Jerome H. Friedman, 2001).

이를 기반으로 기계학습에서 사용하는 모델들을 Gradient Boosting Machine (GBM)이라고 한다.

### 3.2.2 LightGBM (Light Gradient Boosting Algorithm)

LightGBM은 데이터의 샘플 수를 줄이는 Gradient-based One-Side Sampling(GOSS)와 Feature 수를 줄이는 Exclusive Feature Bundling(EFB) 및 Gradient Boosting Decision Tree(GBDT)를 사용하는 알고리즘이다. 최적화 기법을 이용하여 대용량 데이터셋에서도 효율적인 학습이 가능하며, 다른 GBM 모델보다 빠른 수행 능력을 보여주고 기능을 추가할 수 있는 유연한 구조를 갖고 있다. 단점으로는 작은 데이터셋을 분석할 때 과적합에 민감하다는 점이 있다.

### 3.2.3 CatBoost (Category Boosting)

CatBoost는 기존 부스팅 알고리즘의 느린 학습 속도와 오버피팅 문제를 해소하는 알고리즘이다. 순서에 따라 모델을 만들고 예측하는 방식인 Ordered Boosting을 사용하며, 트리 기반 구조를 통해 예측 성능을 향상시킨다. 잔차 예측 방식에서는 Random Permutation을 통해 오버피팅을 방지한다. 또한 Feature Combination을 통해 이미 사용된 Categorical Feature를 결합하여 알고리즘이 자체적으로 변수 선택을 하게 한다(Prokhorenkova 등, 2018). 단점으로는 결측 데이터를 처리하지 않으며, 데이터가 수치형 변수로 이루어진 경우 Light Gradient Boosting Algorithm (LGBM)보다 학습 속도가 느리다는 특징이 있다.

### 3.2.4 XGBoost (Extreme Gradient Boosting)

XGBoost는 대용량 데이터 구조에서 병렬 수행을 통해 GBM보다 빠른 속도를 제공하며, 예측, 회귀, 분류 등과 같은 일반적인 기계 학습 모델에서 높은 성능을 보여준다. 많은 데이터 경진대회(Kaggle, Dacon)에서 높은 순위를 차지할 만큼 사용빈도가 매우 높다. 알고리즘 자체적으로 교차 검증, 성능 평가와 같은 과적합 방지 기능을 제공하며, Tree Pruning을 통해 분할 수를 줄여 일반화 성능을 향상한다 (Chen et 등, 2016). 단점으로는 많은 파라미터를 조정하는 데 시간이 오래 걸리고, 작은 데이터셋에서는 과적합 가능성성이 존재하며, 모델의 복잡성이 높아 해석이 용이하지 않다는 점이 있다.

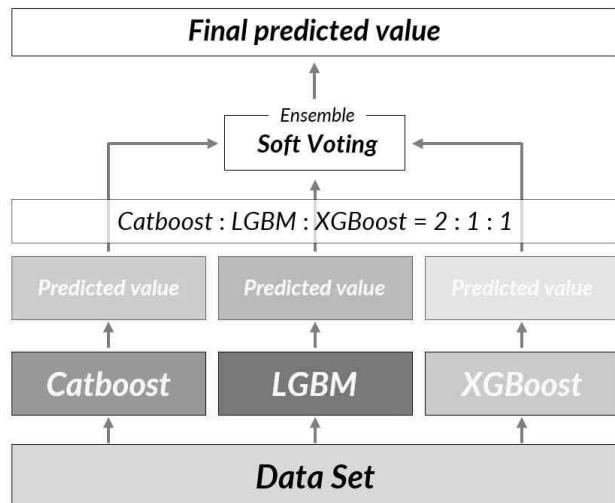
### 3.2.5 Voting

Voting은 여러 개의 모델(분류기)을 사용하여 동일 데이터에서 여러 모델로 학습을 진행하는 양상을 기법 중 하나다. Voting에는 Hard Voting과 Soft Voting이 있

다. Hard Voting은 각 모델이 만든 예측 값을 다수결로 결정하여 가장 많은 표를 얻은 예측 값을 최종 결과로 도출한다. Soft Voting은 각 모델이 예측한 클래스별 확률을 평균 내어 최종 예측 값을 도출한다 (Bauer, Eric 와 Ron Kohavi, 1999).

우리는 [그림 3]과 같이 각기 다른 모델들을 양상을 도킹하여 예측 성능을 최적화하기 위해 Voting 기법을 사용했다. 먼저, 각 모델을 데이터셋에 맞게 Fitting 시킨 후 예측을 진행하였고, 실제 테스트 데이터의 예측 결과를 기준으로 CatBoost가 가장 높은 성능을 보였다. 이를 바탕으로 CatBoost에 더 높은 가중치를 부여하여 최종 예측을 수행하였다. 구체적으로, XGBoost와 LGBM에 각각 1의 가중치를, CatBoost에는 2의 가중치를 부여하여 최종 예측 모델을 구성하였다.

[그림 3] 사용한 Ensemble (Soft Volting) 구조

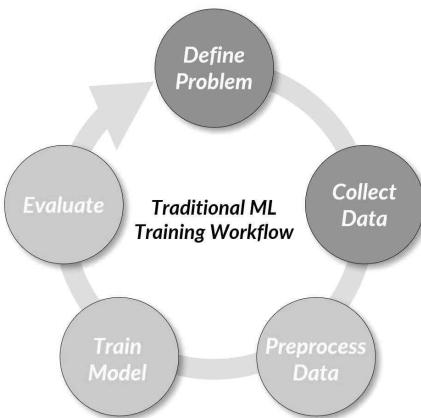


### 3.2.6 AutoML(Automated machine learning)

Automated Machine Learning(AutoML)은 실제 현실 데이터 문제에 적용할 수 있는 머신러닝 모델들을 자동화하여 최적의 알고리즘을 찾아주고, 하이퍼파라미터 적용 방법도 제시한다. AutoML의 가장 큰 장점은 기존의 머신러닝 방법에서는 각각의 모델을 불러와 하나씩 코드를 수행해야 했던 방식에서 벗어나, 다양한 모델을 간단한 코드로 구현하고 최적의 모델에 대한 파라미터를 얻을 수 있다는 점이다. AutoML 도구 중 하나인 AutoGluon은 Amazon Web Services(AWS)에서 개발되

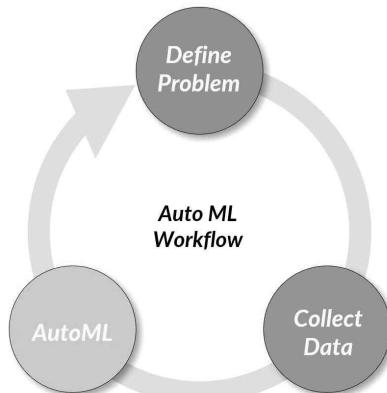
었으며, 텍스트, 이미지, 테이블(Tabular) 형식의 데이터에서 강력한 예측 성능을 제공한다(Erickson, N 등, 2020).

[그림 4] 고전적인 방식의 머신러닝 수행 과정



[그림 4]는 고전적인 방식의 머신러닝 수행 과정을 도식화한 것이다. 먼저 문제를 정의한 후, 데이터를 수집하여 전처리 과정을 거친다. 그런 다음, 예측하고자 하는 방법에 맞는 머신러닝 모델을 선정하고 훈련을 시킨 후, 평가를 진행한다. 이 방식은 많은 시간과 노력이 필요하며, 각 단계에서 사용자의 개입이 필수적이다.

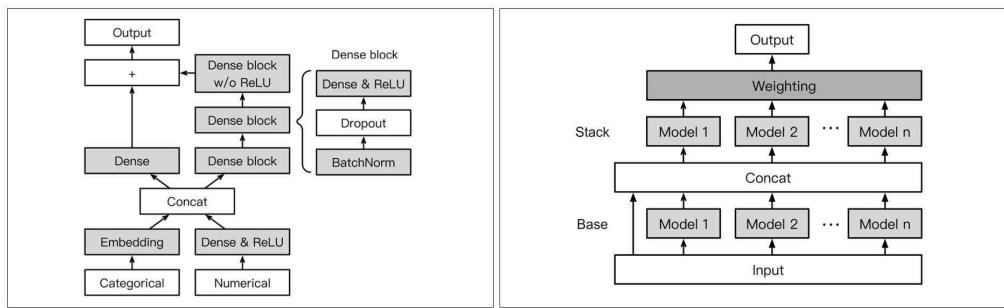
[그림 5] AutoML의 수행 과정



이에 반해 [그림 5]은 AutoML 방식의 수행 과정을 도식화한 것이다. 문제를 정의하고 데이터를 수집하는 방식은 동일하지만, 전처리 과정부터 평가 과정까지 모

는 단계를 자동으로 분석하고 결과를 도출한다. 이를 통해 하이퍼파라미터 튜닝과 같은 사용자의 고도화된 지식이 필요한 부분을 알고리즘이 대신 수행할 뿐만 아니라, 모델 평가까지 맡아 최적화된 모델을 쉽게 만들어 낼 수 있다. 본 대회에 참가한 대부분의 팀이 AutoML을 사용하여 최종 결과를 제출한 만큼, AutoML의 사용빈도는 매우 높다고 볼 수 있다.

[그림 6] AutoGluon의 아키텍처



[그림 6]은 AutoGluon의 구조와 다중층을 나타낸 것이다. AutoGluon은 범주형 변수와 수치형 변수를 각각 임베딩과 Dense 층 및 ReLU 활성화 함수를 통해 결합하여 신경망이 두 변수를 독립적으로 학습할 수 있게 돋는다. 또한, 다중 스태킹 구조를 통해 각 모델의 출력 값을 결합하고, 하이퍼파라미터를 찾는 과정을 거쳐 최종 모델을 선정한다. AutoGluon의 경우, 데이터 학습에 실패하는 경우가 다른 프레임워크보다 적고 처리 시간도 가장 적게 걸리는 장점이 있다.

[그림 7] AutoGluon의 성능 비교

Framework	Wins	Losses	Failures	Champion	Avg. Rank	Avg. Rescaled Loss	Avg. Time (min)
AutoGluon	-	-	<b>1</b>	<b>23</b>	<b>1.8438</b>	<b>0.1385</b>	201
H2O AutoML	4	26	8	2	3.1250	0.2447	220
TPOT	6	27	5	5	3.3750	0.2034	235
GCP-Tables	5	20	14	4	3.7500	0.3336	<b>195</b>
auto-sklearn	6	27	6	3	3.8125	0.3197	240
Auto-WEKA	4	28	6	1	5.0938	0.8001	244

Framework	Wins	Losses	Failures	Champion	Avg. Rank	Avg. Percentile	Avg. Time (min)
AutoGluon	-	-	<b>0</b>	<b>7</b>	<b>1.7143</b>	<b>0.7041</b>	<b>202</b>
GCP-Tables	3	7	1	3	2.2857	0.6281	222
H2O AutoML	1	7	3	0	3.4286	0.5129	227
TPOT	1	9	1	0	3.7143	0.4711	380
auto-sklearn	3	8	<b>0</b>	1	3.8571	0.4819	240
Auto-WEKA	0	10	1	0	6.0000	0.2056	221

[그림 7]는 AutoGluon과 다른 프레임워크를 비교한 것이다. 여러 AutoML 도구 중, H2O AutoML은 알고리즘 선정, 유의미한 특성 선택, 하이퍼파라미터 튜닝 및 반복적인 모델링을 자동화하는 데 강점을 가진 프레임워크이다(Erin LeDell 과 Sebastien Poirier, 2020). TPOT은 특히 트리 기반 모델이나 회귀 모델에서 우수한 성능을 보여주며, 다수의 파이프라인을 통해 최적의 성능을 도출하는 모델을 제공한다(Olson, R.S 등, 2016).

한편, GCP-Tables는 구글에서 개발한 AutoML 기법으로, 사용자가 원시 데이터를 입력하면 자동으로 기계 학습을 수행한다. 이와 비슷하게 auto-sklearn은 sklearn 라이브러리와 유사한 사용 방식을 제공하며, 알고리즘 선택과 하이퍼파라미터 튜닝을 자동으로 지원한다. (Efficient와 Robust, 2015) 또한, Auto-WEKA는 분류 알고리즘에서 높은 성능을 발휘하며, 하이퍼파라미터 최적화, 모델 선택 및 변수 선택을 자동으로 수행한다(Chris Thornton 등, 2013).

이 프레임워크들 중 AutoGluon은 데이터 학습 실패율이 다른 프레임워크에 비해 현저히 낮으며, 처리 시간이 가장 짧다는 장점을 가지고 있다. 이러한 특징 덕분에 AutoGluon은 텍스트, 이미지, 테이블 형식의 데이터에서 강력한 예측 성능을 제공할 수 있다. AutoGluon의 구조는 범주형 변수와 수치형 변수를 각각 임베딩과 Dense 층 및 ReLU 활성화 함수를 통해 결합하여 신경망이 두 변수를 독립적으로 학습할 수 있게 됩니다. 또한, 다중 스테킹 구조를 통해 각 모델의 출력 값을 결합하고, 하이퍼파라미터 최적화를 통해 최종 모델을 선정한다.

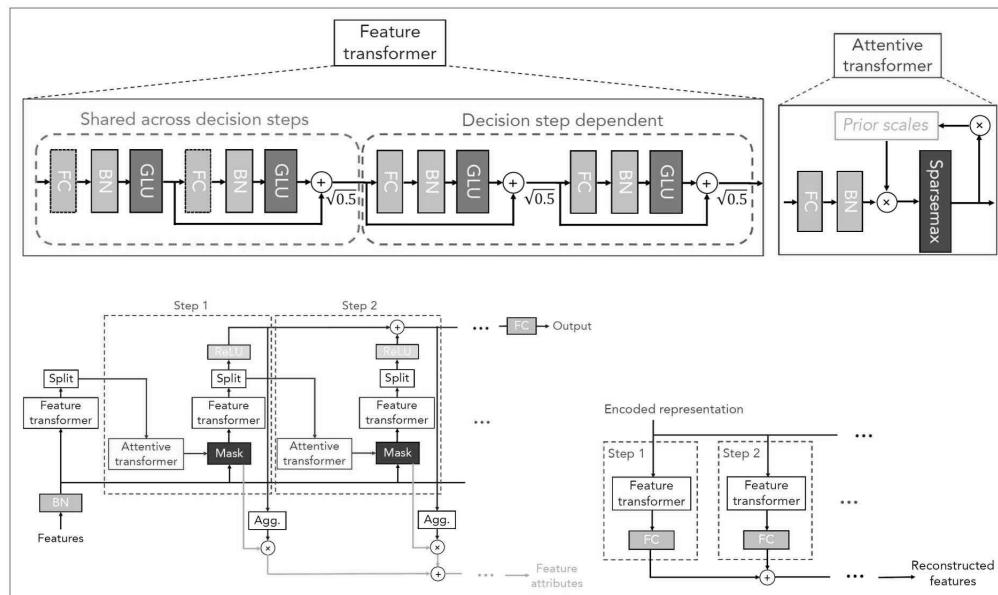
이처럼 AutoGluon은 머신러닝 모델의 자동화와 최적의 알고리즘을 찾는 데 있어 매우 유용하며, 특히 데이터 학습 과정에서의 안정성과 효율성 면에서 다른 프레임워크보다 뛰어나다.

### 3.3 딥러닝 기법

#### 3.3.1 TabNet

TabNet은 트리 기반 모델과 딥러닝의 장점을 결합한 모델로, 정형 데이터에 딥러닝을 적용하고자 하는 시도로 개발되었다. TabNet은 원시 데이터를 입력받아 중요한 특징을 자동으로 학습하며, 다양한 데이터 유형과 복잡한 패턴을 효율적으로 처리할 수 있다(Arik, Sercan O.와 Tomas Pfister, 1908).

[그림 8] TabNet의 주요 구성 요소



[그림 8]은 TabNet의 주요 구성 요소를 나타낸다. 상단 두 그림의 경우에는 Feature Transformer와 Attentive Transformer의 구성 요소이며, 하단 두 그림의 경우에는 각각 Encoder와 Decoder 구조를 나타낸 그림이다.

TabNet의 주요 구성 요소의 경우에는 Feature Transformer와 Attentive Transformer로 나뉜다. Feature Transformer는 데이터를 처리하고 학습하는 데 중요한 역할을 하며, Attentive Transformer는 각 의사결정 단계에서 중요한 변수를 선택하고 그 중요도를 평가한다. 이러한 구성 요소들을 통해 TabNet은 Encoder와 Decoder 구조로 작동한다.

Feature Transformer는 데이터를 처리하고 학습하는 데 중요한 역할을 한다. 이 구조는 크게 두 부분으로 나뉘는데, 하나는 모든 의사결정 단계에서 공유되는 부분이고, 다른 하나는 각 의사결정 단계마다 독립적으로 작동하는 부분이다. 이로 인해 TabNet은 글로벌 패턴을 학습하면서도 각 단계에서 세부적인 패턴을 학습할 수 있다.

Attentive Transformer는 FC (Fully Connected Layer), BN (Batch Normalization), Prior scales, Sparsemax로 구성한다. 이 구성 요소들은 입력 데이터를 변환하고 정규화하며, 이전 단계에서 중요한 변수들에 가중치를 부여하여 각 변수의 중요도를 평가하고, 중요한 변수들만 선택하게 한다.

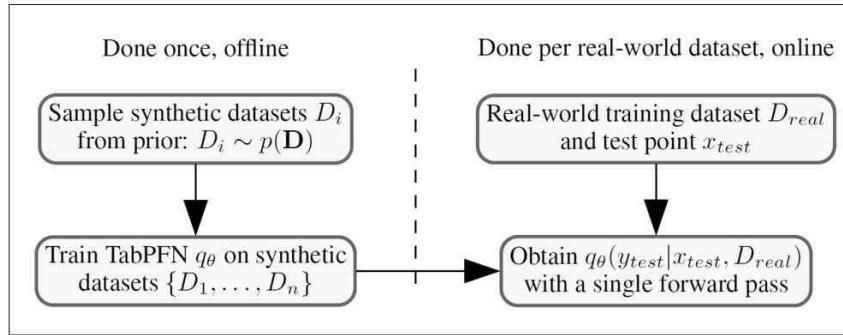
TabNet의 Encoder는 Batch Normalization, Feature Transformer, Attentive Transformer로 구성되며, 각 단계에서 중요한 특징을 선택하고 이를 통해 모델이 중요한 특징에 집중할 수 있게 한다. Decoder는 주로 Self-supervised 학습에서 사용되며, 인코더에서 얻은 정보를 바탕으로 데이터를 재구성하는 역할을 한다.

TabNet은 이러한 구조를 통해 별도의 전처리 없이 원시 데이터를 입력받아 경사하강법 최적화 방법을 통해 중단간 학습이 가능하다. 의사 결정 단계마다 중요한 변수를 선택하여 학습 및 해석이 용이하며, 사전 비지도 학습으로 정형 데이터에서도 뛰어난 성능을 발휘한다. 따라서 TabNet은 트리 기반 모델의 강점과 딥러닝의 장점을 결합하여 정형 데이터에서도 높은 성능과 해석력을 제공하는 모델이다.

### 3.3.2 TabPFN

TabPFN은 정형 데이터(Tabular data)에 대해 초고속 분류를 수행하는 사전 훈련된 Transformer 모델이다. TabPFN은 Prior-Data Fitted Network(PFN) 개념을 기반으로 한다. PFN은 사전 적합 단계에서 주어진 사전 분포에 대해 베이지안 추론을 근사화하며, 추론 단계에서는 데이터셋을 샘플링하여 사전 훈련하고 실제 데이터셋에 대한 예측을 제공한다(Noah Hollmann 등, 2023).

[그림 9] PFN의 사전적합(Prior-fitting)과 추론(Inference)



[그림 9]의 좌측에서 PFN은 오프라인 단계에서 주어진 사전 분포의 확률 분포를 근사하는 방법을 학습한다. 오프라인 학습을 완료한 후, 우측의 온라인 단계에서는 새로운 데이터셋에 대해 Forward Pass로 예측을 수행한다.

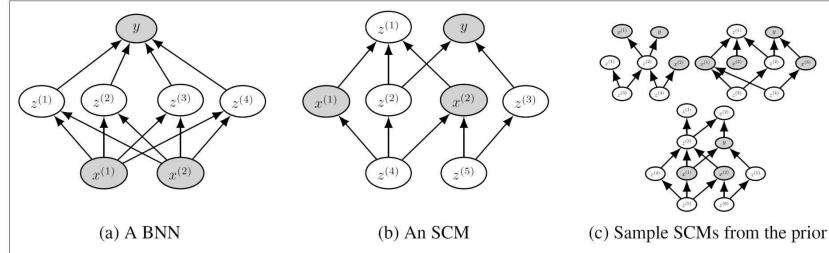
TabPFN은 기존의 PFN 아키텍처를 두 가지 방식으로 수정했다. 첫째, Attention 마스크를 수정하여 추론 시간을 단축했다. 둘째, 제로 패딩을 통해 다양한 Feature 수를 가진 데이터셋에서도 모델이 작동할 수 있도록 했다. 이를 통해 하이퍼 파라미터 튜닝 없이도 강력한 성능과 빠른 실행 속도를 제공하며, Gradient Boosted Decision Trees(XGBoost, CatBoost, LightGBM 등)보다 수치형 데이터셋에서 더 빠르고 통계적으로 유의미한 성능을 보여준다.

TabPFN은 크게 사전 적합 단계와 추론 단계로 구성된다. 사전 적합 단계에서 Structural Causal Models(SCM)과 Bayesian Neural Networks(BNN) 사전으로부터 샘플링된 데이터에 대해 TabPFN을 한 번 훈련한다. 이 과정에서 배치당 512개의 합성 생성 데이터셋으로 12개 층의 Transformer를 쌓아 18,000개의 배치로 훈련한다. 훈련 과정은 8개의 Nvidia RTX 2080 Ti GPU가 장착된 기계에서 총 20시간이 소요되며, 모든 평가에 사용되는 단일 네트워크가 생성된다. 이 훈련 단계는 비용이 많이 들지만, 오프라인에서 한 번만 수행되면 TabPFN의 알고리즘 개발의 일환이다.

추론 단계에서 TabPFN은 데이터셋 사전에 대한 Posterior Predictive Distribution(PDD)를 근사화한다. 인과관계를 모델링하는 SCM과 BNN 사전을 혼합하여 marginal 예측을 근사하고, 이 두 사전을 통해 데이터에 대한 인과적이고 단순한 설명에 주력한다.

이러한 특성 덕분에 TabPFN은 정형 데이터에 대해 매우 효율적이고 효과적인 분류 성능을 제공하며, 기존의 트리 기반 모델들보다 빠르고 높은 성능을 자랑한다. TabPFN은 특히 다양한 Feature 수를 가진 데이터셋에서도 강력한 성능을 유지하며, 데이터 분석과 머신러닝 모델 개발에 있어 중요한 도구로 자리 잡고 있다.

[그림 10] BNN과 SCM 사전(prior)의 Architecture



[그림 10] 은 BNN과 SCM 사전에서 데이터를 생성하는 그래프의 개요이다. 입력값  $x$ 는 관측되지 않은 노드 $z$ 를 통해 출력값  $y$ 로 매핑된다.

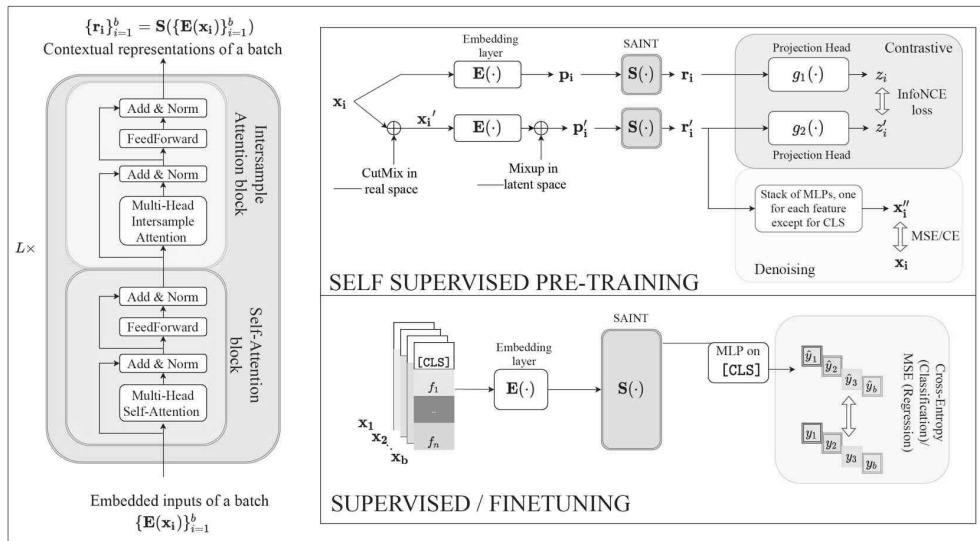
### 3.3.3 SAINT

Self-Attention and Intersample Attention Transformations(SAINT)는 정형 데이터를 처리하기 위한 새로운 하이브리드 딥러닝 접근법이다. 정형 데이터가 딥러닝에서 효과적이지 못했던 첫 번째 이유는 정형 데이터는 이미지와 달리 행과 열의 순서가 중요하지 않아 전통적인 포지셔널 임베딩 기법을 적용하기 어렵다는 점이다. 또한, 정형 데이터는 연속형, 범주형, 순서형 값들이 혼합되어 있어 데이터 간의 상관관계를 파악하기 어렵다는 문제가 있다. 마지막으로, 정형 데이터는 모델이 예측해야 할 타겟 값(label)을 가지고 있지 않아 지도 학습이 불가능하다는 이유로 딥러닝에서 활발히 다뤄지지 않는다는 점이다(Somepalli 등, 2021).

첫 번째 문제를 해결하기 위해 SAINT는 모든 특징을 결합된 밀집 벡터 공간(dense vector space)으로 투영하고, 셀프 어텐션(self-attention)과 행 간의 상관관계를 고려하는 인터샘플 어텐션(intersample attention) 기법을 제시한다. 이는 KNN 알고리즘과 유사한 방식으로, 유사한 데이터들의 특성을 활용하여 최적화된 예측 값을 도출한다. 두 번째 문제는 모든 특성을 하나의 벡터 공간에 투영함으로써 보완한다. 마지막 문제는 Contrastive pre-training을 활용하여 레이블이 부족한 경우에도 우수한 성능을 달성할 수 있도록 하였다.

SAINT는 이러한 기법들을 통해 정형 데이터에서도 딥러닝의 강점을 활용할 수 있게 하였으며, 특히 데이터 간의 복잡한 상관관계를 효과적으로 모델링할 수 있게 한다. 이를 통해 SAINT는 정형 데이터에서 높은 예측 성능을 제공하며, 전통적인 머신러닝 기법들을 대체하거나 보완할 수 있는 강력한 도구로 자리매김하고 있다.

[그림 11] SAINT 모형



Intersample attention은 단일 데이터 행의 특성뿐만 아니라, 같은 배치 내 다른 모든 데이터 샘플과의 어텐션 점수(attention score)도 계산하는 방식이다. 이는 하나의 샘플을 예측할 때 같은 배치 내 다른 행의 데이터 특성을 참고하는 방식으로, 모델이 학습되며 어떤 데이터의 특성을 고려할 것인지에 대해 점차 최적화되는 과정을 거친다. 이와 더불어, 특정 데이터 내에서 누락되거나 이상치가 있는 특성을 배치 내 다른 유사한 데이터로 대체하여 보완하는 역할을 수행한다. SAINT의 주된 기법이라고 할 수 있는 Intersample attention은 모델의 성능을 크게 향상시킨다.

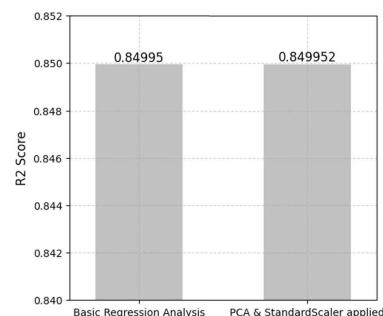
SAINT는 또한 Contrastive loss와 Denoising loss를 사용한다. Contrastive

learning은 같은 데이터 행의 두 가지 버전(원본과 mixup, cutmix 기법으로 증강된 버전)을 가까이 위치시키고, 다른 데이터 행들은 멀리 떨어지게 학습한다. Denoising loss는 노이즈가 있는 입력에서 원래 데이터를 예측하여 손실을 최소화하는 방법이다. 이러한 두 가지 손실을 줄여 최적화하는 방식으로 학습이 이루어진다.

이러한 기법들을 통해 SAINT는 정형 데이터에서 높은 예측 성능을 달성하며, 특히 데이터 간의 복잡한 상관관계를 효과적으로 모델링할 수 있다. SAINT는 정형 데이터에서의 딥러닝 활용을 극대화하여, 전통적인 머신러닝 기법을 대체하거나 보완할 수 있는 강력한 도구로 자리 잡고 있다.

#### 4. 실험 및 결과

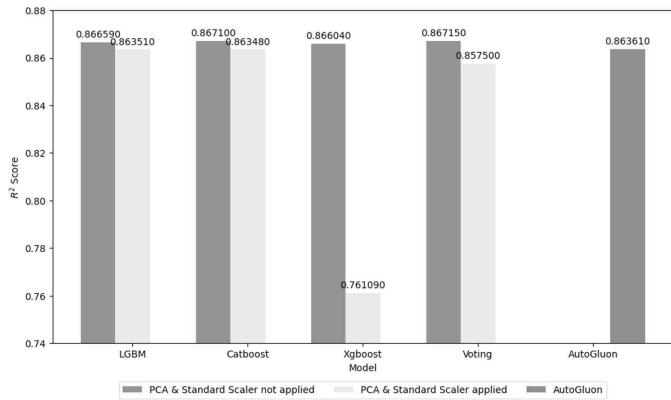
[그림 12] 상황별 회귀분석의  $R^2$  Score



[그림 12]는 회귀분석을 이용한 분석 결과이다. 회귀분석을 이용하여 최종적으로 얻은  $R^2$  Score 값은 각각 0.84995와 0.849952이다. PCA와 StandardScaler를 적용한 경우와 적용하지 않은 경우의  $R^2$  Score 값이 동일하다. 이는 데이터 전처리 과정에서 PCA와 StandardScaler를 적용하는 것은 유의미하지 않다고 판단을 하였다.

다음 [그림 13]의 경우에는 머신러닝 모델을 이용한 분석 결과이다.

[그림 13] 상황별 머신러닝 모델의  $R^2$  Score

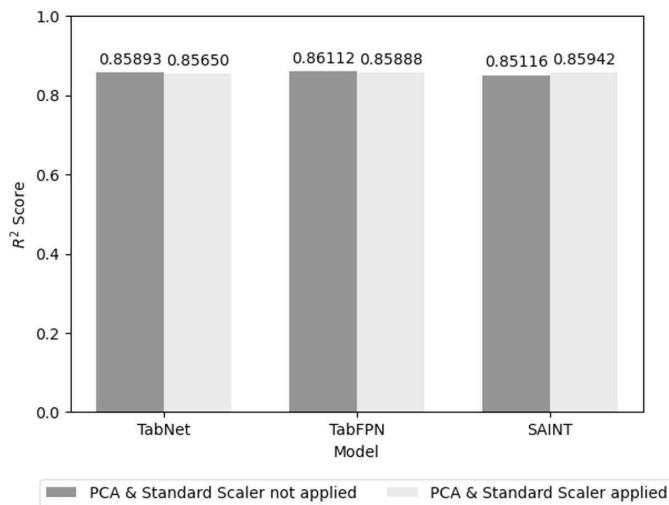


[그림 13]의 경우에는 머신러닝 모델에서 PCA와 StandardScaler 적용의 유무에 따른  $R^2$  Score 값에 대한 그래프이다. 파란색이 PCA와 StandardScaler 적용하지 않은 상황이고, 회색이 PCA와 StandardScaler 적용을 하지 않은 상황에서의  $R^2$  Score 값을 나타낸다. LGBM과 CatBoost,XGBoost를 기본 모델로 사용을 하였고, 양상블은 LGBM, CatBoost,XGBoost에 대하여 Soft Voting을 활용해 모형을 제작했다. 가중치는 모델 순서대로 1:1:2 라는 가중치를 활용하였다. 머신러닝 모델에서 PCA와 StandardScaler 적용 하지 않은 경우 양상블을 활용한 경우의  $R^2$  Score 값이 0.86715로 가장 높았다. PCA와 StandardScaler 적용한 경우에는 CatBoost 모델을 사용한 경우의  $R^2$  Score 값이 0.86348로 가장 높았다.

AutoML 방법 중 AutoGluon을 적용한 경우, PCA와 StandardScaler를 적용하지 않은 경우가 더 효율적이라고 판단해 전처리 과정을 적용하지 않은 방법만 분석을 진행하였다. 그 결과, LightGBMXT\_BAG\_L2라는 모델이 채택 되었는데 이는 LightGBM을 기반으로 하여 배깅 양상블의 일부이고 양상블 구조의 두 번째 레벨에서 작동을 하는 AutoML에서 사용되는 모델이다.  $R^2$  Score 값은 0.86361로 PCA와 StandardScaler 적용 하지 않은 경우의 모든 모델의  $R^2$  Score 값보다 낮았다.

결과적으로 보았을 때 양상블을 활용한 경우의  $R^2$  Score 값이 0.86715로 가장 높았다.

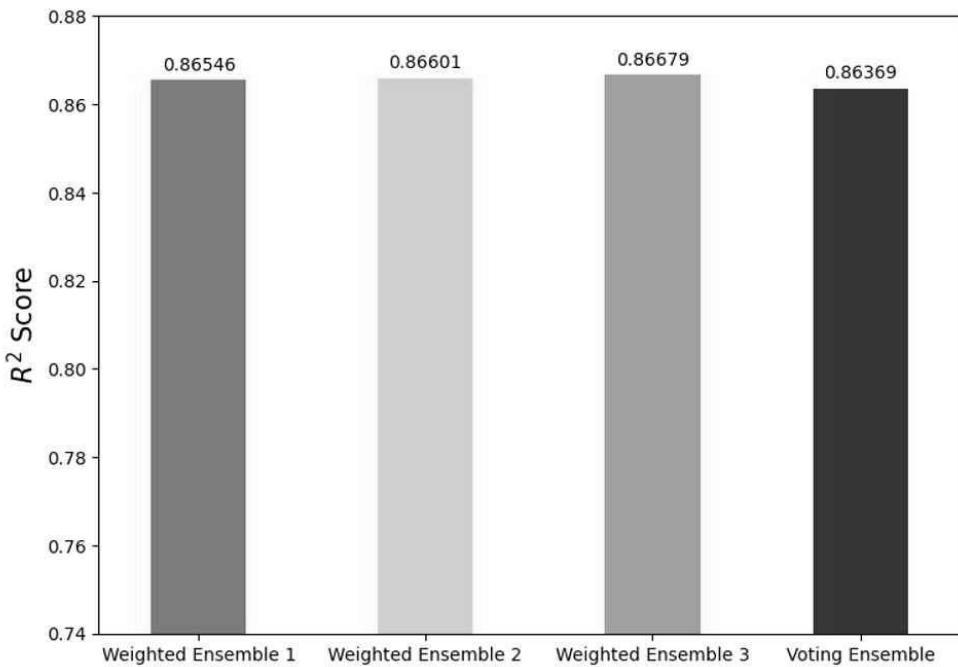
[그림 14] 상황별 딥러닝 모델의  $R^2$  Score



[그림 14]의 경우에는 딥러닝 모델에서 PCA와 StandardScaler 적용의 유무에 따른  $R^2$  Score 값에 대한 그래프이다. 파란색이 PCA와 StandardScaler 적용하지 않은 상황이고, 회색이 PCA와 StandardScaler 적용을 하지 않은 상황에서의  $R^2$  Score 값을 나타낸다.

정형 데이터에서 사용할 수 있는 딥러닝 모델 중 성능이 좋다고 알려진 TabNet, TabPFN, SAINT 모델을 사용했다. PCA와 StandardScaler 적용하지 않은 상황에서 TabPFN의  $R^2$  Score 값이 0.86112 으로 가장 높았다. PCA와 StandardScaler 적용한 경우에는 SAINT 모델의  $R^2$  Score 값이 0.85942 으로 가장 높았다.

[그림 15] 머신러닝과 딥러닝의 앙상블 모델의  $R^2$  Score



[그림 15]는 PCA와 StandardScaler를 적용하지 않은 경우의 머신러닝 모델과 딥러닝 모델을 양상을 기법으로 활용한 결과를 그래프로 나타낸 것이다. PCA와 StandardScaler를 적용하지 않은 경우에는에서의 R<sup>2</sup> Score 값이 전반적으로 더 높게 나타났기 때문에, 이 전처리 과정을 수행하지 않은 상태에서의 모델 양상을만을 고려했다.

머신러닝 모델로는 LGBM, CatBoost, XGBoost 세 모델을 전부 사용했으며, 딥러닝 모델로는 가장 성능이 좋았던 TabPFN 모델을 사용했다. 평균값을 활용한 양상을의 경우, 각 모델에 동일한 가중치를 부여하여 계산했다.

Weighted Ensemble 1은 머신러닝 모델 양상들의 최종 결과와 TabPFN의 최종 예측 결과의 평균값을 이용한 것이고, Weighted Ensemble 2는 머신러닝 양상들 최종 결과와 TabPFN의 최종 결과를 6:4 비율로 양상한 것이다. Weighted Ensemble 3은 머신러닝 양상들 최종 결과와 TabPFN의 최종 예측 결과를 8:2 비율로 양상한 결과이다. 마지막으로, Voting Ensemble은 LGBM, CatBoost, XGBoost와 TabPFN 모델을 동일한 가중치로 양상한 것이다.

가장 높은 R<sup>2</sup> Score를 보인 경우는 머신러닝 모델들과 TabPFN을 8:2 비율로 양상한 방법으로, 이때의 값은 0.86679였다. 최종적으로 R<sup>2</sup> Score가 가장 높게 나온 경우는 PCA와 StandardScaler를 적용하지 않은 상태에서의 양상으로, R<sup>2</sup>

Score 값은 0.86715로 나타났다. 이는 2400개 팀 중 상위 30%에 해당하는 성과이다.

## 5. 결론 및 개선방안

### 5.1 분석 결과

분석 결과, 전반적으로 머신러닝 모델(CatBoost, LGBM, XGBoost)이 딥러닝 모델(TabNet, TabPFN, SAINT)보다 대회에서 순위에 반영되는  $R^2$  Score 지표에서 더 높은 값을 보였다. 특히, 범주형 변수로 구성된 데이터셋에서는 정형 데이터를 다루는 데 특화된 CatBoost 모델이 가장 우수한 성능을 나타냈다. CatBoost 모델에서 PCA 및 StandardScaler를 적용하지 않은 경우의  $R^2$  Score 값은 0.86720으로 가장 높았으며, 이는 Kaggle 대회에서 전체 2300팀 중 상위 30%에 해당하는 점수이다.

## 5.2 한계점 및 개선방안

이번 연구를 진행하며 느낀 첫 번째 한계점은 전처리 방식의 부적절함이다. PCA와 StandardScaler를 동시에 사용하여 전처리하는 방식과 전처리를 전혀 적용하지 않는 두 가지 방식으로 나누어 진행했다. 대회가 끝난 후 Kaggle에 공개된 우승팀의 코드를 확인한 결과, 차원 축소나 표준화를 수행한 팀은 많지 않았다. 이는 설명 변수들이 범주형 변수로 이루어져 있고 데이터셋의 범위가 거의 동일하여 전처리를 수행할 필요가 없다는 것을 의미한다.

두 번째 한계점은 파생 변수에 관한 것이다. 분석을 본격적으로 진행하기 전, 설명 변수의 값을 모두 더한 ‘Sum’이라는 새로운 설명 변수를 추가했다. 그러나 각 모델의 결과들이 ‘Sum’ 변수에만 집중되어 예측값이 도출되었다. 만약 설명 변수들의 중앙값(Median)을 사용하거나 값을 모두 곱한 열(Column)을 추가했다면, 다양한 파생 변수 설정을 통해 더 풍부한 결과를 얻을 수 있었을 것이다.

세 번째 한계점은 양상을 기법의 한계이다. 양상을 기법은 하나의 모델을 사용했을 때보다 더 나은 결과와 성능을 기대할 수 있어야 한다. 그러나 실제 분석 결과, 양상을 기법의  $R^2$  Score는 CatBoost를 단독으로 사용했을 때와 비슷한 값을 보였다. 따라서 세 개의 모델을 사용하는 것보다 RandomForest와 같은 예측 모델을 추가하고, 하이퍼파라미터 조정을 통해 각 모델의 성능을 최대한 높인 후 양상을 진행하는 것이 더 좋은 결과를 도출할 수 있을 것이다.

네 번째 한계점은 AutoGluon의 런타임이다. AutoGluon을 이용하여 주어진 데이터셋을 입력하면 자동으로 타겟 변수를 가장 잘 예측하는 방향으로 하이퍼파라미터를 조정하고 모델을 제시한다. AutoGluon을 런타임 2시간으로 사용했을 때의  $R^2$  Score 값은 0.86361이었다. 이 대회의 우승팀도 AutoGluon을 이용하여 최적의 모델과 파라미터를 찾아 분석했으며,  $R^2$  Score 값은 0.86905였다. 즉, AutoGluon을 이용해 최적의 모델과 파라미터를 찾는 시간이 길어질수록 더 좋은 결과를 얻을 수 있다는 결론에 도달했다.

다섯 번째 한계점은 정형 데이터 분석에 있어서 딥러닝 모델의 한계다. 딥러닝을 이용한 분석 방법에서도 PCA와 StandardScaler를 적용한 경우와 적용하지 않은 경우 두 가지 방식으로 진행했다. 그 결과, PCA 및 StandardScaler를 적용하지 않은 TabPFN 기법의  $R^2$  Score 값이 0.86112로 딥러닝 기법 중에서는 가장 높았다. 그러나 이는 머신러닝 기법인 AutoGluon의  $R^2$  Score 값보다 0.00249 낮았다. 이는 딥러닝 모델이 아직 정형 데이터 예측에 있어서 머신러닝 성능을 뛰어넘지 못한다는 한계점을 보여준다.

또한, 딥러닝 모델 TabNet, TabPFN, SAINT는 Pytorch를 기반으로 하는 반면, 수업에서는 TensorFlow를 기반으로 딥러닝 모델들을 학습하여 코드 진행과 모델

링 부분에서 추가적인 이해가 필요했다. 특히, 딥러닝 모델에서 코드를 구성할 때 새롭게 클래스를 만들어 직접 구현해야 하는 과정은 상당한 시간이 소요되었다.

딥러닝 모델 SAINT의 경우, Validation 데이터셋과 Test 데이터셋에 대한  $R^2$  Score 값은 0~1 사이의 확률 값으로 도출되었지만, Train 데이터셋에서는 음수 값이 나와 모델이 잘 적합되지 않았다고 판단했다. 이는 모델 구현 방식에서 코드를 변경하거나, Self-Attention 기법에 대한 깊은 이해를 통해 해결할 수 있을 것이다.

이번 연구를 진행하며 느낀 한계점을 보완하기 위한 첫 번째 방안은 SAINT 모델의 적극적인 활용이다. SAINT 모델은 Transformer의 Self-Attention 기반 알고리즘으로, 자연어 처리에서 사용되는 기법과 유사한 부분이 있다. 이를 참고하여 정형 데이터에서 분석하는 방법을 연구해 더 좋은 결과를 도출할 수 있다.

다음으로, 정형 데이터셋을 다루는 다양한 딥러닝 기법들(TabNet, NODE, GrowNet, AutoInt, SAINT)을 분석 단계에서 적극적으로 사용하는 것이다. NODE, GrowNet, AutoInt에 대해서도 TabNet, TabPFN, SAINT처럼 개념과 분석에 대한 종합적인 이해를 한다면, 여러 딥러닝 기법을 활용한 정형 데이터 분석에서 더 다양한 결과를 얻을 수 있을 것이다.

## 참고문헌

- [1] Arik, Sercan Ö., and Tomas Pfister. "Tabnet: Attentive interpretable tabular learning." Proceedings of the AAAI conference on artificial intelligence. Vol. 35. No. 8. 2021.
- [2] Bauer, Eric and Ron Kohavi, An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting and Variants, Machine Learning Vol 36, pp.105–142, 1999
- [3] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016.
- [4] Erickson, N.; Mueller, J.; Shirkov, A.; Zhang, H.; Larroy, P.; Li, M.; Smola, A. AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data. arXiv . March 13, 2020
- [5] Feurer, Matthias, et al. "Efficient and robust automated machine learning." Advances in neural information processing systems 28 (2015).
- [6] Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." Annals of statistics (2001): 1189–1232.
- [7] Hollmann, Noah, et al. "TabPFN: A transformer that solves small tabular classification problems in a second." arXiv preprint arXiv:2207.01848 (2022).
- [8] LeDell, Erin, and Sébastien Poirier. "H2o automl: Scalable automatic machine learning." Proceedings of the AutoML Workshop at ICML. Vol. 2020. San Diego, CA, USA: ICML, 2020.
- [9] Olson, Randal S., et al. "Automating biomedical data science through tree-based pipeline optimization." Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30--April 1, 2016, Proceedings, Part I 19. Springer International Publishing, 2016.
- [10] Prokhorenkova, Liudmila, et al. "CatBoost: unbiased boosting with categorical features." Advances in neural information processing systems 31 (2018).

- [11] Somepalli, Gowthami, et al. "Saint: Improved neural networks for tabular data via row attention and contrastive pre-training." arXiv preprint arXiv:2106.01342 (2021).
- [12] Thornton, Chris, et al. "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms." Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. 2013.