creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공학전문석사 학위 연구보고서

행동 시퀀스 트랜스포머를 이용한 구독 비즈니스의 고객 이탈 예측

Churn Prediction with Behavior Sequence Transformer for Subscription Business

2023년 2월

서울대학교 공학전문대학원
응용공학과 응용공학전공
김민창

행동 시퀀스 트랜스포머를 이용한 구독 비즈니스의 고객 이탈 예측

Churn Prediction with Behavior Sequence Transformer for Subscription Business

지도교수 구 윤 모

이 프로젝트 연구보고서를 공학전문석사 학위
연구보고서로 제출함
2023년 1월

서울대학교 공학전문대학원
응용공학과 응용공학전공
김민창

김민창의 공학전문석사 학위 연구보고서를 인준함
2023년 2월

위원장 한 훈 (인)
위 원 곽 우 영 (인)
위 원 구 윤 모 (인)

초 록

정기적으로 일정액을 납입하고 서비스를 이용하는 디지털 구독 비즈니스 시장은 최근 몇 년 동안 지속적으로 성장해 왔다. 통계 플랫폼인 Statista에 따르면 2020년 6,500억 달러로 집계된 전 세계 디지털 구독 시장의 규모는 2025년 1조 5,000억 달러로 연평균 18.2% 증가할 것으로 예상된다 [1]. 이제는 전통적인 구독 비즈니스인 정수기, 학습지, 이동통신부터 콘텐츠, 음원, 소프트웨어와 같은 디지털 구독 비즈니스까지 어디서나 쉽게 구독 서비스를 만날 수 있다.

서비스의 이용과 이탈이 자유로운 디지털 구독 비즈니스가 지속적으로 성장하기 위해서는 무엇보다 서비스를 이용 중인 구독자의 만족도를 유지하는 일이 중요하다. 신규 고객을 지속적으로 유치한다고 하더라도 평균적인 이용 기간이 짧다면 서비스 제공을 통해 얻는 이익 보다 획득 비용이 더 높아지는 문제가 발생한다. 따라서 구독자의 만족도를 예측하고 이탈을 방지하는 일은 지속 가능한 구독 비즈니스 모델을 만들기 위한 핵심적인 활동으로 여겨진다.

본 연구에서는 트랜스포머 아키텍처를 이용해 구독자의 미납 여부를 예측하는 Churn Behavior Sequence Transformer (Churn BST) 모델을 제안하였다. Churn BST 모델은 트랜스포머를 이용한 Sequence 모델링을 통해 시간의 흐름에 따른 구독자의 행동 변화를 동적으로 파악한다. 또한, Sequence 데이터와 Non-sequence 데이터의 결합을 통해 미납 고객의 일반적인 Latent Representations을 학습함으로써 과적합 문제를 해결하고 효과적인 예측을 수행한다.

Churn BST 모델은 비교 실험에서 서로 다른 이탈률을 가진 복수의

구독 비즈니스를 대상으로 기존의 머신러닝 및 딥러닝 방법론 보다 평균적으로 더 높은 예측 정확도를 보여주었다. 특히, 학습에 이용되지 않은 Unseen 데이터셋에서 성능이 저하되는 기존의 방법론에 비해 Precision-Recall AUC (PR AUC)를 25.97% 개선하는 성과를 거두었다. 다양한 데이터셋에서 강건한 (Robust) 성능을 보여준 Churn BST 모델은 지속적으로 구독자의 데이터 분포가 변화하는 실제 산업현장에서 보다 효과적인 대안을 제시한다.

주요어 : 딥러닝, 트랜스포머, 이탈 예측, 불균형 분류, 구독 비즈니스
학번 : 2021-23150

목 차

I. 서론 **1**

1.1 연구 목적 1

1.2 연구 내용 1

1.3 연구보고서의 구성 2

II. 배경 지식과 관련 연구 **4**

2.1 머신러닝 방법론 4

2.1.1 Logistic Regression 4

2.1.2 Random Forest 5

2.1.3 Gradient Boosting 6

2.2 딥러닝 방법론 7

2.2.1 다층 퍼셉트론 (MLP) 8

2.2.2 순환신경망 (RNN) 9

2.2.3 트랜스포머 (Transformer) 10

2.3 이탈 예측 분야 12

2.3.1 전자상거래와 구독 비즈니스 12

2.3.2 이탈 예측 모델 14

III. 문제 정의 및 방법론 **16**

3.1 문제 정의 16

3.1.1 데이터셋 분석 16

3.1.2 이탈 예측 문제의 정의 24

3.2 연구 방법론 24
3.2.1 Churn BST 모델의 구조 24
3.2.2 Churn BST 모델의 구현 상세 26
3.2.3 Churn BST와 BST 모델의 비교 28

IV. 실험 및 평가 30
4.1 실험 조건 30
4.1.1 데이터 전처리 30
4.1.2 성능 지표 32
4.1.3 비교 모델 38
4.2 평가 및 분석 39
4.2.1 기본 실험 40
4.2.2 확장 실험 49

V. 결론 53

참고 문헌 55

Abstract 59

그 림 목 차

그림 1. 트랜스포머 아키텍처 [2, p. 3] 10
그림 2. 행동 시퀀스 트랜스포머 아키텍처 [3, p. 2] 12
그림 3. 이탈 예측 프레임워크 15
그림 4. 조직별 납입 및 미납 분포 19
그림 5. 성별 납입 및 미납 분포 21
그림 6. 연령별 납입 및 미납 분포 21
그림 7. 지역별 납입 및 미납 분포 22
그림 8. 이산형 및 연속형 변수들의 피어슨 상관 계수 23
그림 9. Churn BST 아키텍처 25
그림 10. Sliding Windows 방법 31
그림 11. 연속형 변수 total_num_payments의 표준화 33
그림 12. 오차 행렬 (Confusion Matrix) 34
그림 13. Churn BST의 Train Loss 40
그림 14. Churn BST의 Validation PR AUC 41
그림 15. Churn BST의 Precision-Recall Curve 41
그림 16. Churn BST의 ROC Curve 42
그림 17. Churn BST의 Confusion Matrix 42
그림 18. 머신러닝 비교 모델의 Precision-Recall Curve 44
그림 19. 머신러닝 비교 모델의 ROC Curve 45
그림 20. 머신러닝 비교 모델의 Confusion Matrix 46
그림 21. 딥러닝 비교 모델의 Precision-Recall Curve 48
그림 22. 딥러닝 비교 모델의 ROC Curve 48

그림 23. 딥러닝 비교 모델의 Confusion Matrix 49
그림 24. Churn BST 확장 실험의 Precision-Recall Curve 52
그림 25. Churn BST 확장 실험의 ROC Curve 52
그림 26. Churn BST 확장 실험의 Confusion Matrix 52

표 목 차

표 1. 변수의 구성 (이산형 및 연속형) 17
표 2. 변수의 구성 (범주형) 17
표 3. 조직별 납입 및 미납 분포 18
표 4. 개별 변수의 기술통계량 (이산형 및 연속형) 19
표 5. 개별 변수의 기술통계량 (범주형) 20
표 6. Churn BST 임베딩 레이어의 구성 27
표 7. Churn BST 네트워크의 구성 27
표 8. Churn BST와 BST 모델의 비교 29
표 9. 연속형 변수의 표준화 (StandardScaler) 32
표 10. 연속형 변수의 표준화 (RobustScaler) 32
표 11. 머신러닝 모델과의 성능 비교 43
표 12. 딥러닝 모델과의 성능 비교 47
표 13. Unseen 데이터셋의 조직별 납입 및 미납 분포 49
표 14. Unseen 데이터셋 조직 F에 대한 성능 비교 50
표 15. Unseen 데이터셋 조직 G에 대한 성능 비교 51

제 1 장

서론

1.1 연구 목적

본 연구의 주요 목적은 과적합 문제를 피해 복수의 구독 비즈니스를 대상으로 보다 일반화된 이탈 예측 성능을 달성하는 모델을 개발하는 것이다. 일반화된 모델을 이용하면 모델이 학습하지 못한 새로운 환경에서의 예측 정확도를 높일 수 있을 뿐만아니라 서로 다른 조직의 데이터를 개별적으로 학습하고 하이퍼파라미터를 조율하는데 투입되는 비용을 감소시켜 유지보수성을 높일 수 있다는 장점이 있다.

1.2 연구 내용

본 연구에서는 정기 납입자의 미납 가능성을 예측하는 Churn BST 모델을 제안하고, 기존 머신러닝 및 딥러닝 방법론과의 대규모 비교 평가를 진행하였다.

첫째, 모델의 개발과 학습에는 국내 구독 플랫폼을 통해 수집한 납입 데이터 84만 개를 이용했다. 각각의 데이터는 서로 다른 5개 구독 비즈니스의 구독자 정보이며 평균 이탈률은 4∼10%로 서로 다른 분포와 특성을 보여준다. 추가적으로, 학습에 이용되지 않은 Unseen 데이터에 대한 실험을 진행하기 위해 2개의 구독 비즈니스로부터 납입 데이터 5만 개를 수집하여 별도의 테스트용 데이터셋을 구성하였다.

둘째, 트랜스포머를 이용해 선행 연구에서 다루어지지 못했던 Sequence 형식의 월별 납입 패턴을 효과적으로 모델링하는 방법을 제안했다. 또한, 트랜스포머의 출력값과 별도로 인코딩한 Non-sequence 형식의 연령, 납입수단, 로그인 횟수 등의 보조 정보를 결합하여 Feedforward Network (FFN) 레이어를 통해 1 (미납)과 0 (납입) 여부를 예측하도록 모델을 구현했다.

셋째, 학습 데이터셋과 Unseen 데이터셋에 대한 대규모의 비교 실험을 진행하였다. 총 8개의 머신러닝 방법론 (K-Nearest Neighbor, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, AdaBoost, XGBoost, LightGBM)과 3개의 딥러닝 방법론 (MLP, RNN, LSTM)으로 비교 모델을 구현하였고, Precision-Recall AUC (PR AUC)를 기준으로 성능을 평가하였다. 실험 결과 본 연구에서 제안한 Churn BST 모델이 Unseen 데이터셋에서 차상위 비교 모델 보다 PR AUC를 25.97% 개선하는 성과를 거두었다.

1.3 연구보고서의 구성

본 연구보고서는 총 5개의 장으로 구성되어 있으며 각 장의 내용은 다음과 같다. 1장에서는 본 연구를 수행하게 된 목적과 의의, 그리고 핵심적인 연구 내용에 대해 설명한다. 2장에서는 본 연구의 기반이 되는 배경 지식을 소개하고, 본 연구와 선행 연구들 간의 차별성을 확인한다. 3장에서는 본 연구에서 풀고자 하는 미납자 예측 문제의 특징을 정의하고 사용된 데이터에 대한 분석을 진행한다. 또한, 본 연구에서 제안한 예측 모델과 방법론을 구체적으로 살펴본다. 4장에서는 모델의 구현 방법과 구체적인 실험 조건을 설명하고, 예측 성능에 대한 평가와 분석을 진행한다.

5장에서는 본 연구를 통해 달성한 성과와 연구의 한계에 대해 설명하고, 향후 연구 계획을 제시한다.

제 2 장

배경 지식과 관련 연구

2.1 머신러닝 방법론

머신러닝 (Machine Learning)은 데이터를 분석하고, 분석 결과를 학습한 후, 이를 기반으로 판단이나 예측을 수행하는 알고리즘을 의미한다. 본 절에서 다루는 Logistic Regression, Random Forest, Gradient Boosting은 선행 연구에서 우수한 예측 성능을 보여주는 대표적인 머신러닝 알고리즘이다. [4]

2.1.1 Logistic Regression

로지스틱 회귀분석은 이항분류 모델에 주로 활용되는 방법론이다. 양성과 음성, 합격과 불합격 등 종속변수의 값이 두 개의 범주로 나누어지는 경우 회귀분석 결과가 항상 $[0,1]$ 사이에 있도록 조정해 준다. 일반적인 로지스틱 회귀분석은 오즈비 (Odds Ratio)를 구하고, 로짓 (Logit) 변환을 수행한 뒤, 로지스틱 함수를 도출하는 순서로 진행된다. [5]

오즈비는 임의의 사건 A가 발생할 확률과 발생하지 않을 확률 간의 비율을 의미한다. 예를 들어, 양성일 확률이 음성일 확률 보다 몇 배나 더 높은지를 나타낸다.

$$odds\ ratio = \frac{P(A)}{1-P(A)} \tag{2.1}$$

오즈비는 두 확률의 비율에 따라 $[0,\infty)$ 범위의 값을 갖게 된다. 여기서 오즈비에 로그를 취하게 되면 입력값이 $[0,1]$일 때 출력값이 $(-\infty,+\infty)$인 로짓 함수가 만들어 진다.

$$z = logit(odds\ ratio) = log(\frac{P(A)}{1-P(A)}) \tag{2.2}$$

로지스틱 함수는 로짓 함수의 역함수이다. 로지스틱 함수는 입력값이 $(-\infty,+\infty)$일 때 출력값이 $[0,1]$ 사이에 있도록 변환해 주는 작업을 수행한다. $[0,1]$ 사이에 양성과 음성을 구분하는 임의의 임계값을 설정하면 로지스틱 함수를 이용해 이항분류 작업을 수행할 수 있다.

$$logistic(z) = \frac{1}{1+exp(-z)} \tag{2.3}$$

모델의 학습에는 Log Loss가 사용된다. 예측값과 실제값의 차이를 계산하고 경사하강법을 이용해 Log Loss를 최소화 하는 계수를 찾는 과정을 통해 학습이 진행된다.

2.1.2 Random Forest

랜덤포레스트 (Random Forest)는 다수의 의사결정나무를 이용해 얻은 결과를 투표에 부쳐 결론을 내는 앙상블 (Ensemble) 방법론의 일종이다. 단일한 의사결정나무 (Decision Tree)가 갖고 있는 과적합의 위험성을 낮추고 Unseen 데이터에 대해서도 보다 일반화된 성능 (Generalized performance)을 보여준다. [6, 7]

랜덤포레스트는 동일한 데이터로부터 무작위 복원추출을 수행한 후 각각의 하위 데이터셋에 의사결정나무를 적용하는 방식으로 동작한다. 데이터셋과 의사결정나무를 무작위로 생성하는 과정에서 개별 나무들의 다양성이 높아지고, 일부의 나무가 특정 데이터셋에 과적합 되더라도 나무들 사이의 투표를 통해 결과적으로 숲 차원에서 올바른 결정이 이루어질 수 있도록 한다.

위의 무작위 복원추출과 의사결정나무의 결합 기법을 Bootstrap aggregating (Bagging)이라 부른다. 구체적으로 Bagging은 다음의 절차를 통해 이루어진다. 1) 전체 데이터셋으로부터 무작위 복원추출을 통해 T개의 학습용 하위 데이터셋을 생성한다. 2) 각각의 학습 데이터셋에 대응하는 T개의 의사결정나무를 훈련시킨다. 이때 개별 의사결정나무가 이용하는 변수의 개수에 제한을 두어 나무의 다양성을 높인다. 3) 학습이 완료된 의사결정나무를 결합하여 투표를 진행한다. 투표는 다수결 또는 평균 방식으로 이루어진다.

2.1.3 Gradient Boosting

그래디언트 부스팅 (Gradient Boosting)은 다수의 의사결정나무를 이용하는 또 다른 앙상블 방법론이다. 무작위 복원추출을 통해 상관관계가 없는 의사결정나무들을 이용하는 랜덤포레스트와 달리 앞서 사용된 나무의 오차를 보완하는 나무를 순차적으로 생성하는 방식을 통해 정확도를 높인다. [8]

부스팅은 약한 분류기를 순차적으로 연결하여 강한 분류기를 만드는 방법을 의미한다. A 분류기를 만든 후 결과를 예측하고, 그 정보를 바탕으로 B 분류기를 만든다. 다시 예측을 진행하고, 그 정보를 바탕으로 C 분류기를 만든다. 이렇게 만들어진 분류기를 모두 결합하면 최종적인 모

델이 만들어진다.

각각의 분류기가 만들어낸 예측값과 실제값과의 차이를 잔차 (Residual)이라고 부른다. 전반적인 그래디언트 부스팅 알고리즘은 Residual fitting 과정으로 이해할 수 있다. A 분류기가 예측하지 못한 잔차를 B 분류기가 이어서 예측하는 과정을 반복하게 되면 모델 전체의 Bias가 줄어드는 원리이다.

하지만 그래디언트 부스팅 알고리즘에서는 앞선 분류기의 잔차를 얼마나 강하게 보정할 것인지에 따라 과적합이 발생할 수 있다는 단점이 있다. 이를 제어하기 위해 다양한 하이퍼파라미터와 정규화 방법이 활용된다. 학습률 (Learning rate)은 대표적인 하이퍼파라미터이다. 학습률이 크면 이전 분류기의 오차를 강하게 보정하기 때문에 과적합이 발생할 가능성도 높아진다. 또한, 랜덤포레스트와 달리 분류기의 개수가 많아질수록 잔차를 더 많이 보정하게 되기 때문에 역시 과적합 가능성이 커지게 된다.

부스팅 기법을 보완하고 발전시킨 알고리즘에는 AdaBoost [9], XGBoost [10], LightGBM [11] CatBoost [12] 등이 존재하며 본 연구에서는 Gradient Boosting, AdaBoost, XGBoost, LightGBM 알고리즘으로 머신러닝 비교모델을 구현했다.

2.2 딥러닝 방법론

딥러닝 (Deep Learning)은 인간의 뇌를 모사한 신경망 알고리즘을 이용하는 머신러닝의 하위 분야이다. 오차역전파 (Back Propagation) 알고리즘과 경사하강법 (Gradient Descent)을 이용해 데이터로부터 스스로 학습을 수행하며 특정한 문제에 대한 판단과 예측이 가능한 심층신경망을

구축한다. 본 절에서는 다루는 다층 퍼셉트론 (MLP), 순환신경망 (RNN), 트랜스포머 (Transformer)는 대표적인 인공신경망 알고리즘이다. [13]

2.2.1 다층 퍼셉트론 (MLP)

MLP (Multilayer Perceptron)는 단층 퍼셉트론의 한계를 넘어서기 위해 제안된 방법이다. 단층 퍼셉트론은 선형적인 결정 경계를 갖고 있고, 이를 통해서는 비선형적인 XOR 문제를 해결할 수 없다는 한계를 갖고 있다. MLP는 입력층과 출력층 사이에 은닉층 (Hidden Layer)을 추가하는 방식으로 만들어진다. AND, NAND, OR 게이트를 조합하면 XOR 게이트를 만들 수 있기 때문에 은닉층을 쌓아가는 방식을 통해 비선형 문제를 풀 수 있게 된다. [14]

1개의 은닉층을 가진 MLP를 수식으로 나타내면 다음과 같다. n개의 관측치와 d개의 변수를 가진 행렬 $\mathbf{X} \in \mathbb{R}^{n \times d}$와 h개의 Unit을 가진 은닉층 $\mathbf{H} \in \mathbb{R}^{n \times h}$ 그리고 q개의 Unit을 가진 출력층 $\mathbf{O} \in \mathbb{R}^{n \times q}$이 있다고 가정하자. 이때 은닉층의 Weights는 $\mathbf{W}^{(\mathbf{1})} \in \mathbb{R}^{d \times h}$, Biases는 $\mathbf{b}^{(\mathbf{1})} \in \mathbb{R}^{1 \times h}$이 되고 출력층의 Weights는 $\mathbf{W}^{(\mathbf{2})} \in \mathbb{R}^{h \times q}$, Biases는 $\mathbf{b}^{(\mathbf{2})} \in \mathbb{R}^{1 \times q}$가 된다.

$$\begin{aligned} \mathbf{H} &= \mathbf{X}\mathbf{W}^{(\mathbf{1})} + \mathbf{b}^{(\mathbf{1})} \\ \mathbf{O} &= \mathbf{H}\mathbf{W}^{(\mathbf{2})} + \mathbf{b}^{(\mathbf{2})} \end{aligned} \tag{2.4}$$

퍼셉트론의 출력값은 활성화 함수 (Activation Function)를 통해 비선형화된다. 어떤 활성화 함수가 효과적인지에 대해서는 많은 연구가 이루어지고 있는데 Sigmoid, ReLU, Softmax 함수 등이 이용된다. 활성화 함수 σ를 적용한 MLP의 구조는 다음과 같다.

$$
\begin{aligned}
\mathbf{H} &= \sigma(\mathbf{X}\mathbf{W}^{(\mathbf{1})} + \mathbf{b}^{(\mathbf{1})}) \\
\mathbf{O} &= \mathbf{H}\mathbf{W}^{(\mathbf{2})} + \mathbf{b}^{(\mathbf{2})}
\end{aligned}
\tag{2.5}
$$

심층신경망 (Deep Neural Network)은 은닉층을 2개 이상 쌓은 신경망이다. 심층신경망이 고안 되었던 초기에는 신경망의 Weights를 학습시킬 수 있는 방법을 찾지 못해 제대로 동작하는 모델을 구현할 수 없었다. 1986년에 출력층에서 계산된 오차를 입력층 방향으로 내려보내면서 가중치를 재조정하는 오차역전파 (Back Propagation) 알고리즘이 제안되었으며 이후 딥러닝 (Deep Learning) 연구가 활성화되는 계기가 되었다. [15]

2.2.2 순환신경망 (RNN)

RNN (Recurrent Neural Network)은 시퀀스 (Sequence) 단위로 입력과 출력을 처리하는 네트워크다. RNN에서는 은닉층을 RNN 셀이라고 표현하며 t 시점에서 $t+1$ 시점의 자신에게 보내는 Weights를 은닉 상태 (Hidden State)라고 부른다. RNN 셀은 시퀀스의 길이에 관계없이 입력과 출력을 받아들일 수 있기 때문에 번역과 같이 상황에 따라 길이라 달라지는 데이터를 처리하기에 적합한 구조를 갖고 있다. [16]

하지만 시퀀스의 길이가 너무 길어질 경우에는 오차역전파 시 그래디언트가 소멸 되어 학습이 이루어지지 않는 문제 (Vanishing Gradient Problem)가 발생한다. Long Short-Term Memory (LSTM)는 그래디언트 소멸 문제를 극복하기 위해 제안된 네트워크 구조이다. 단기 정보를 기억하는 은닉 상태와 장기 정보를 기억하는 셀 상태 (Cell state)를 이용하여 시퀀스를 처리하는 과정에서 최종 출력층과 먼 위치에 있는 정보가 소실되는 단점을 보완했다. [17]

RNN 기반의 네트워크는 추천과 예측 분야에서 활발하게 활용되고 있다. GRU4Rec [18]은 GRU 레이어를 통해 사용자와 아이템 간의 상호작용 시퀀스를 학습함으로써 사용자의 앞선 선택이 뒤의 선택에 미치는 영향을 고려하여 추천이 이루어지는 모델을 제안했다. NARM [19]은 GRU 아키텍처에 Attention 메커니즘을 통합하여 시퀀스 내에서도 각각의 상호작용에 대한 중요도를 구분할 수 있는 구조를 제안했다. 이를 통해 멀리 떨어져 있는 상호작용 간의 관계를 더 효과적으로 학습할 수 있는 모델이 구현되었다.

2.2.3 트랜스포머 (Transformer)

트랜스포머 (Transformer)는 인코더 - 디코더 구조와 멀티 헤드 어텐션 (Multi-head Attention)을 이용하는 시퀀스 모델이다. RNN처럼 순차적으로 데이터를 입력 받는 대신 위치 인코딩 (Positional Encoding)을 이용하며 시퀀스 전체를 한 번에 처리한다.

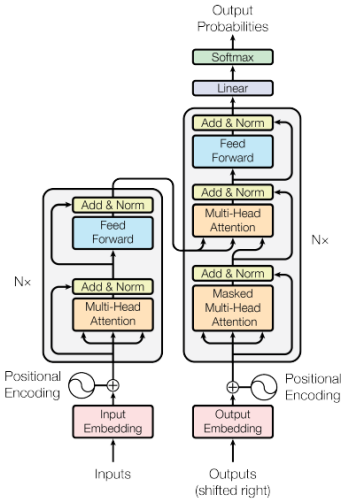

그림 1: 트랜스포머 아키텍처 [2, p. 3]

트랜스포머는 자연어 처리 분야 (Natural Language Processing)에서 처음으로 제안되었으며, 트랜스포머를 이용한 BERT [20], GPT-3 [21]와 같은 모델은 기존의 방법론을 뛰어넘는 탁월한 성능을 보여주었다. 이후 자연어 처리 이외의 분야에서도 트랜스포머의 구조를 활용한 다양한 연구가 진행되었으며 예측과 추천 모델에도 적용되어 좋은 성과를 보여주고 있다.

구체적으로 SASRec [22]은 트랜스포머 아키텍처를 이용해 대용량 모델을 구현했다. BERT4Rec [23]은 양방향 인코더와 가면 훈련 기법 (Masked Training Techniques)을 이용해 일방향 트랜스포머 모델의 성능을 개선했다. Transformers4Rec [24]은 Auto Regressive (AR)와 Auto Encoding (AE) 학습 방식의 장점을 결합한 XLNet [25]의 구조를 이용하여 시퀀스 내의 정보를 더 효과적으로 모델링 하는 방법을 제안했다.

세계 최대의 B2B 전자상거래 플랫폼을 운영하는 알리바바에서 2019년에 발표한 Behavior Sequence Transformer (BST) [3]는 트랜스포머 아키텍처를 이용하여 구현된 추천시스템 모델이다. 기존의 연구에서는 사용자의 행동 시퀀스를 고려하지 않고 서로 다른 사용자 변수들을 병렬적으로 학습하는 경우가 많았다. BST 논문에서는 트랜스포머를 이용해 조회나 클릭 등의 행동 시퀀스를 모델링하고 이를 성별이나 나이 등의 보조적인 정보와 결합 (Concatenate)함으로써 사용자 고유의 Latent Representations을 학습한다. 이를 통해, 예측 대상의 특징을 효과적으로 표현함으로써 개인화된 추천 성능을 높일 수 있었다.

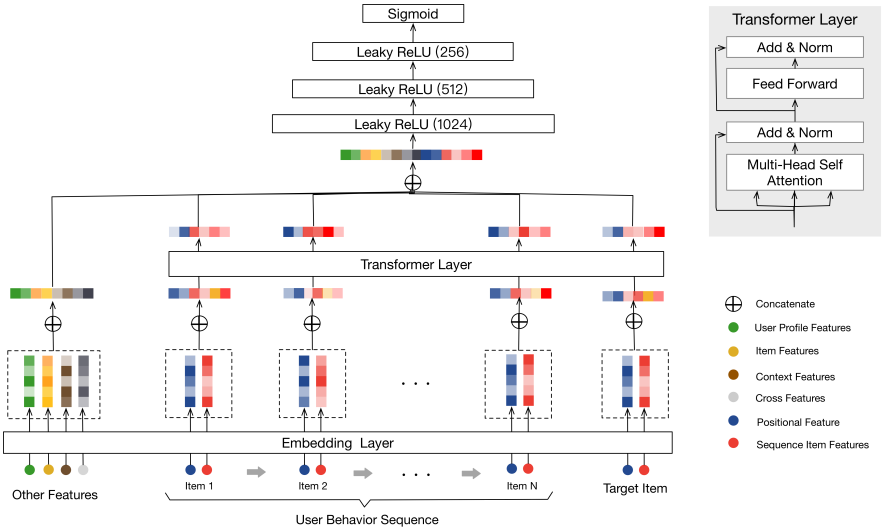

그림 2: 행동 시퀀스 트랜스포머 아키텍처 [3, p. 2]

2.3 이탈 예측 분야

이탈 예측 분야는 예측의 대상이 되는 데이터셋과 예측 모델의 목표에 따라 구분된다. 본 절에서는 전자상거래와 다른 구독 비즈니스 데이터셋의 특징과 선행 연구에서 다루어진 대표적인 이탈 예측 모델에 대해 살펴본다.

2.3.1 전자상거래와 구독 비즈니스

이탈 예측 연구는 구매의 정기성을 기준으로 비정기적인 구매가 대부분인 전자상거래 비즈니스와 체결한 약정에 따라 정기적인 구매가 발생하는 구독 비즈니스로 분야가 구분된다. 구독 계약의 유무에 따라 고객의 행동 패턴이 달라지기 때문에 각각의 분야에서는 서로 다른 방식으로 고객 데이터를 정의하고 예측하는 연구를 수행해 왔다.

전자상거래 비즈니스 분야의 연구는 앞으로 고객이 무엇을 얼마나

구매할지 또는 구매하지 않을지를 예측하여 특정 고객의 고객생애가치 (Customer Lifetime Value)를 계산하는 것을 목표로 한다. RFM으로 통칭되는 Recency (최근 구매일), Frequency (구매 빈도), Monetary (구매 금액)는 전자상거래 분야의 예측 모델이 이용하는 대표적인 고객 데이터 유형이다. Recency는 마지막으로 발생한 고객과의 상호작용 일시를 의미하며 해당 고객이 다시 구매할 시점을 알려준다. Frequency는 고객이 얼마나 자주 구매를 해왔는지를 의미하며 충성도를 파악하는데 이용된다. Monetary는 고객이 얼마를 지불해 왔는지를 통해 해당 고객의 가치를 금액으로 계산할 수 있도록 해준다. [26]

한편, 구독 비즈니스 분야의 연구는 사전에 체결된 약정을 기준으로 구독 계약의 미납과 해지를 방지하는 것을 목표로 한다. 정기적으로 제품을 사용하고 비용을 지불한다는 약속에 기반하여 약정 사항, 납입 사항, 제품 사용량, 고객 만족도의 4가지 데이터가 대표적인 유형으로 이용된다 [27]. 약정 사항은 구독 비즈니스에서만 존재하는 특수한 데이터 유형이다. 구체적인 약정 사항에는 구독 플랜, 플랜 변동내역, 할인 혜택, 구독 기간, 갱신일자 등이 있다. 납입 사항은 청구 일자, 납입 일자, 결제수단, 신용카드 만료일 등의 데이터를 의미한다. 제품 사용량에는 구독 대상이 되는 제품에 대한 일별/주별/월별 활성 사용자 수, 로그인 빈도, 기능별 사용 빈도 등이 포함된다. 또한, 사용자가 특정되어 있는 구독 비즈니스에서는 고객 만족도 지표를 전자상거래 비즈니스에서 보다 더 적극적으로 활용할 수 있다. Net Promotor Score (NPS) [28], 평균 문의량 등의 정보는 구독 고객의 현재 상태를 반영하며 예측 정확도를 높이는데 기여한다.

2.3.2 이탈 예측 모델

이탈 예측 모델은 목표로 하는 예측 대상에 따라 Sentiment Model (감정 모델), Event Model (사건 모델), Survival Model (생존 모델), Treatment Model (처치 모델)의 4가지 유형으로 구분된다. 각각의 모델은 데이터셋의 특징과 따라 비즈니스 목표에 따라 통계, 머신러닝, 딥러닝 등 다양한 알고리즘의 결합을 통해 구현된다. [29]

Sentiment Model (감정 모델)은 이탈을 촉발시킨 주요 요인 (이탈 사유)을 식별하고 이탈 신호를 감지하는 것을 목표로 한다. 텍스트나 음성과 같은 비정형 데이터를 적극적으로 활용하며 감정 분석과 주제 모델링을 수행하하기 위해 BERT와 같은 자연어처리 (Natural Language Processing) 모델이 이용되고 있다.

Event Model (사건 모델)은 단기적인 (예. 1개월 이내) 이탈 위험을 예측하는 것을 목표로 한다. Gradient Boosting, Random Forest와 같은 머신러닝 방법론과 시계열 분석에서 좋은 성능을 보여주는 RNN, 트랜스포머 기반의 방법론이 제안되고 있다.

Survival Model (생존 모델)은 장기적인 (예. 6개월 이상) 이탈 위험을 예측하는 것을 목표로 한다. 전통적으로 Kaplan-Meier 추정법과 Cox 비례위험모형과 같은 통계적인 방법론이 활용된다. Kaplan-Meier 추정법은 사건이 발생한 시점의 구간생존율을 이용해 최종적인 누적생존율을 산출하는 방법이며, Cox 비례위험모형은 다변량 회귀분석을 통해 생존율에 영향을 미치는 위험 요소를 파악하는 방법이다. 최근에는 Sentiment Model과 Event Model의 출력을 이용하는 머신러닝과 딥러닝 기반의 이진 분류기도 좋은 성능을 보여주고 있다.

Treatment Model (처치 모델)은 예측된 이탈 대상에게 가장 효과적

인 이탈 방지 방안을 판별하는 것을 목표로 한다. Event Model과 Survival Model의 출력을 이용하며 Uplift 모델링으로 가장 효과적인 이탈 방지 대안을 결정한다. 이탈할 것으로 예측된 대상에게 특정한 조치를 취했을 때 실제로 이탈을 방지할 수 있는지의 여부를 판단할 수 있기 때문에 비용과 편익에 기반하여 예측 모델의 실효성을 높여주는 역할을 한다. Treatment Model (처치 모델)의 정확도를 높이기 위해 다양한 Causal Inference (인과 추론) 방법이 연구되고 있다.

본 보고서에서는 단기적인 이탈 위험을 예측하는 Event Model (사건 모델)에 대한 연구를 수행하였다. 4가지 유형의 모델은 전체적인 이탈 예측 프레임워크의 구성 요소로서 비즈니스의 목표에 따라 상호보완적으로 배치되며 예측 모델의 기대효과를 높여준다.

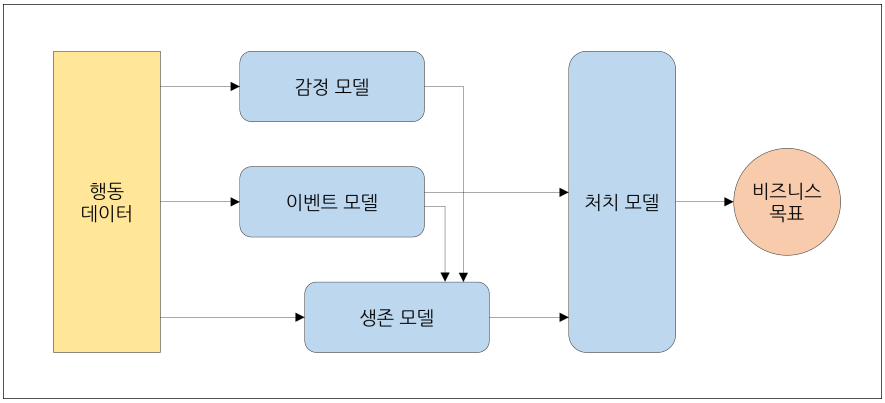

그림 3: 이탈 예측 프레임워크

제 3 장

문제 정의 및 방법론

3.1 문제 정의

본 절에서는 이번 연구에서 사용된 데이터셋의 특징을 분석하고, 이탈 행동을 수학적으로 규정함으로써 예측 모델이 풀어야 하는 문제를 구체적으로 정의한다.

3.1.1 데이터셋 분석

탐색적 데이터 분석 (Exploratory Data Analysis) [30]은 학습에 사용할 데이터셋에 대한 이해도를 높여 문제에 새로운 가설을 수립하고 효과적으로 접근할 수 있도록 도와준다. 본 항에서는 변수의 구성과 기술통계량을 포함하여 다양한 각도에서 데이터셋을 관찰하였으며 차트를 이용해 데이터의 분포와 패턴을 파악하였다.

3.1.1.1 변수의 구성

데이터셋에 포함된 변수의 상세한 구성 정보는 표 1, 2에 나타나 있다. 특정 조직 (org_id)에 대한 구독자 (user_id)의 정보를 중심으로 총 27개의 변수를 포함하고 있다. 변수의 유형은 이산형 9개, 연속형 6개, 범주형 12개로 구분된다.

표 1: 변수의 구성 (이산형 및 연속형)

번호	변수명	설명	유형	값
1	org_id	조직 id	이산형	1∼5 사이의 값
2	user_id	구독자 id	이산형	1∼213156 사이의 값
3	age	연령	이산형	0: null, 1∼90 사이의 값
4	total_num_pledge	총 약정개수	이산형	
5	total_num_payments	총 납입횟수	이산형	
6	total_engagements	총 참여횟수	이산형	
7	recent_engagements	시퀀스 내 참여횟수 합계	이산형	
8	user_writedate	첫 등록일부터의 기간	연속형	일수로 표기(Day)
9	first_paymentdate	첫 납입일부터의 기간	연속형	일수로 표기 (Day)
10	total_payment_amount	총 납입액	연속형	
11	payment_amount	시퀀스 내 납입액 합계	연속형	
12	payments	시퀀스 내 월별 납입액	연속형	리스트로 구분
13	months_position	시퀀스 내 월별 납입순번	이산형	리스트로 구분
14	org_churn_rate	조직의 평균 미납률	연속형	
15	churn	시퀀스의 미납 여부	이산형	1: 미납, 0: 납입

표 2: 변수의 구성 (범주형)

번호	변수명	설명	유형	값
16	sex	성별	범주형	0: null, 1: 남성, 2: 여성, 3: 비공개
17	region	지역	범주형	0: null, 1: 서울 ∼18: 기타
18	cellphone	휴대전화번호 확보 여부	범주형	1: 확보, 0: 미확보
19	email	이메일 주소 확보 여부	범주형	1: 확보, 0: 미확보
20	account	로그인 계정 확보 여부	범주형	1: 확보, 0: 미확보
21	zipcode	거주지 우편번호 확보 여부	범주형	1: 확보, 0: 미확보
22	receipt	영수증 발급 신청 여부	범주형	0: null, 1: 신청, 2: 미신청
23	is_text	문자 수신동의 여부	범주형	1: 동의, 0: 미동의
24	is_email	이메일 수신동의 여부	범주형	1: 동의, 0: 미동의
25	is_post	우편 수신동의 여부	범주형	1: 동의, 0: 미동의
26	is_call	전화 수신동의 여부	범주형	1: 동의, 0: 미동의
27	payment_methods	시퀀스 내 월별 결제수단	범주형	리스트로 구분

3.1.1.2 기술통계량

학습에 사용된 데이터셋은 구독 플랫폼을 통해 2021년 5월부터 2022년 8월까지 총 16개월 동안 수집한 5개 조직의 구독 정보를 포함하고 있다. 구독자의 납입 이력은 12개월 단위로 분절되었으며 12개월 동안의 시퀀스 정보를 바탕으로 13개월 째의 미납 여부를 예측하는 방식으로 학습이 진행되었다. 전체 시퀀스의 길이는 16개월이므로 1명의 구독자는 최대 4개의 관측치를 가질 수 있다.

학습 데이터셋 전체의 관측치는 841,191개이고 평균 미납률은 7.94%이지만 관측치의 개수와 미납률은 조직마다 큰 차이를 보인다. 소규모 조직인 D의 경우 45,114개의 관측치를 갖고 있으며, 대형 조직인 E는 D의 10배인 449,099개의 관측치를 보유했다. 조직별 미납률의 최소값은 4.65%, 최대값은 10.02%으로 2.2배의 차이를 보여준다. 조직별로 상세한 납입 및 미납 분포는 표 3과 그림 4에 정리되어 있다.

표 3: 조직별 납입 및 미납 분포

조직명	구독자 수	관측치의 개수 (A+B)	납입 수 (A)	미납 수 (B)	미납률
A	20448	78048	74422	3626	4.65%
B	18928	99859	95100	4759	4.77%
C	44262	169071	157643	11428	6.76%
D	11697	45114	43173	1941	4.30%
E	117821	449099	404078	45021	10.02%
합계	213156	841191	774416	66775	7.94%

시퀀스 단위로 분할한 개별 변수들의 기술통계량은 표 4, 5에 정리되어 있다. 각각의 시퀀스가 어떤 객체에 포함되어 있는지는 조직 id (org_id)와 구독자 id (user_id) 변수를 통해 구분 된다.

구독자들의 성별, 연령별, 지역별 인구통계학적 특성에 따른 납입 및 미납 분포는 그림 5, 6, 7에서 자세하게 확인할 수 있다. 성별에 있어서

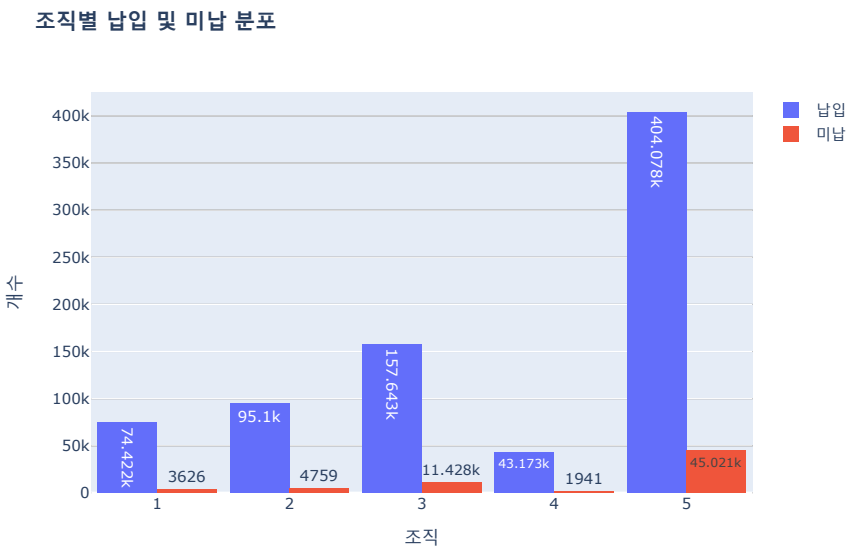

그림 4: 조직별 납입 및 미납 분포

표 4: 개별 변수의 기술통계량 (이산형 및 연속형)

번호	변수명	고유값의 개수	평균	표준편차	최소값	최대값
1	org_id	5	3.88	1.40	1.00	5.00
2	user_id	213156	106577.50	61532.98	0.00	213155.00
3	age	213156	42.83	17.90	0.00	90.00
4	total_num_pledge	213156	1.22	1.36	1.00	315.00
5	total_num_payments	213156	42.65	30.61	1.00	520.00
6	total_engagements	213156	0.21	2.36	0.00	387.00
7	recent_engagements	213156	0.03	0.25	0.00	27.00
8	user_writedate	213156	1269.55	932.34	1.00	9800.00
9	first_paymentdate	213156	1268.59	903.60	0.00	6386.00
10	total_payment_amount	213156	839470.58	845039.12	1.00	10000000.00
11	payment_amount	841191	242063.07	218308.47	0.00	12000000.00
12	payments	841191	-	-	-	-
13	months_position	841191	-	-	-	-
14	org_churn_rate	5	0.08	0.02	0.04	1.00
15	churn	841191	0.08	0.27	0.00	1.00

표 5: 개별 변수의 기술통계량 (범주형)

번호	변수명	고유값의 개수	평균	표준편차	최소값	최대값
16	sex	213156	1.81	0.65	0.00	3.00
17	region	213156	3.85	4.94	0.00	18.00
18	is_cellphone	213156	0.98	0.13	0.00	1.00
19	is_email	213156	0.60	0.49	0.00	1.00
20	is_account	213156	0.04	0.20	0.00	1.00
21	is_zipcode	213156	0.60	0.49	0.00	1.00
22	is_receipt	213156	1.13	0.33	0.00	2.00
23	is_text	213156	1.27	0.44	0.00	1.00
24	is_email	213156	1.44	0.50	0.00	1.00
25	is_post	213156	1.44	0.50	0.00	1.00
26	is_call	213156	1.06	0.23	0.00	1.00
27	payment_methods	841191	-	-	-	-

는 여성이 차지하는 비중이 56.96%로 가장 컸다. 결측치 5.66%를 제외한 구독자들의 평균 연령은 42.83세였으며 표준편차는 17.9를 나타냈다. 거주지는 정보를 수집하지 못한 구독자의 비중이 30.12%로 가장 컸고, 정보를 알고 있는 경우에는 서울과 경기도에 거주하는 비중이 35.71%로 가장 많았다.

피어슨 상관 계수로 살펴본 이산형 및 연속형 변수들은 대부분 낮은 상관 관계를 보여주었다. user_writedate와 total_num_payment 변수 간의 상관 계수가 0.38로 가장 컸으며, age와 total_num_pledge 간의 상관 계수가 0.01로 가장 낮았다. 다른 변수들의 피어슨 상관 계수는 그림 8에서 확인할 수 있다.

그림 5: 성별 납입 및 미납 분포

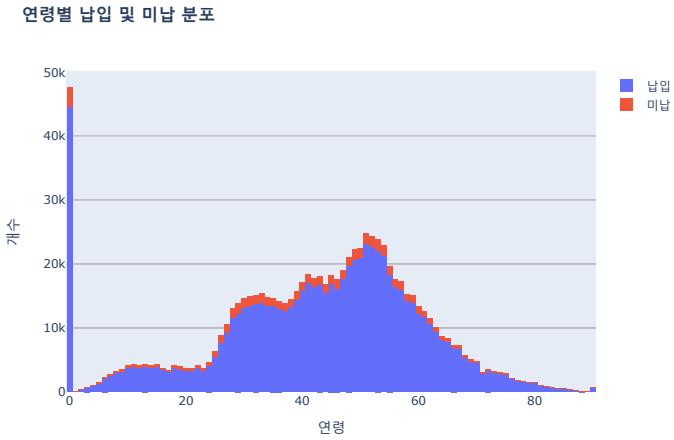

그림 6: 연령별 납입 및 미납 분포

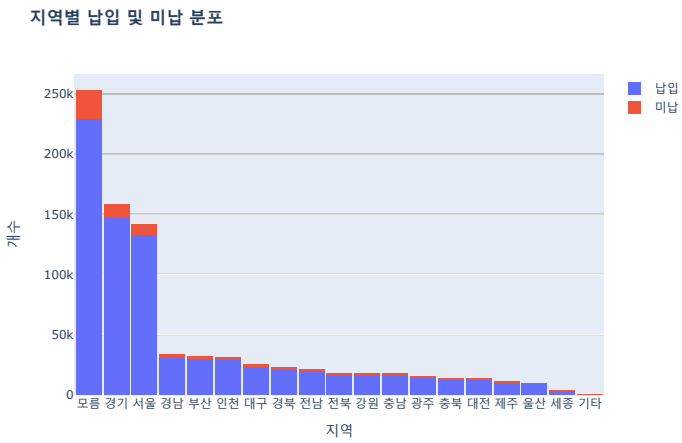

그림 7: 지역별 납입 및 미납 분포

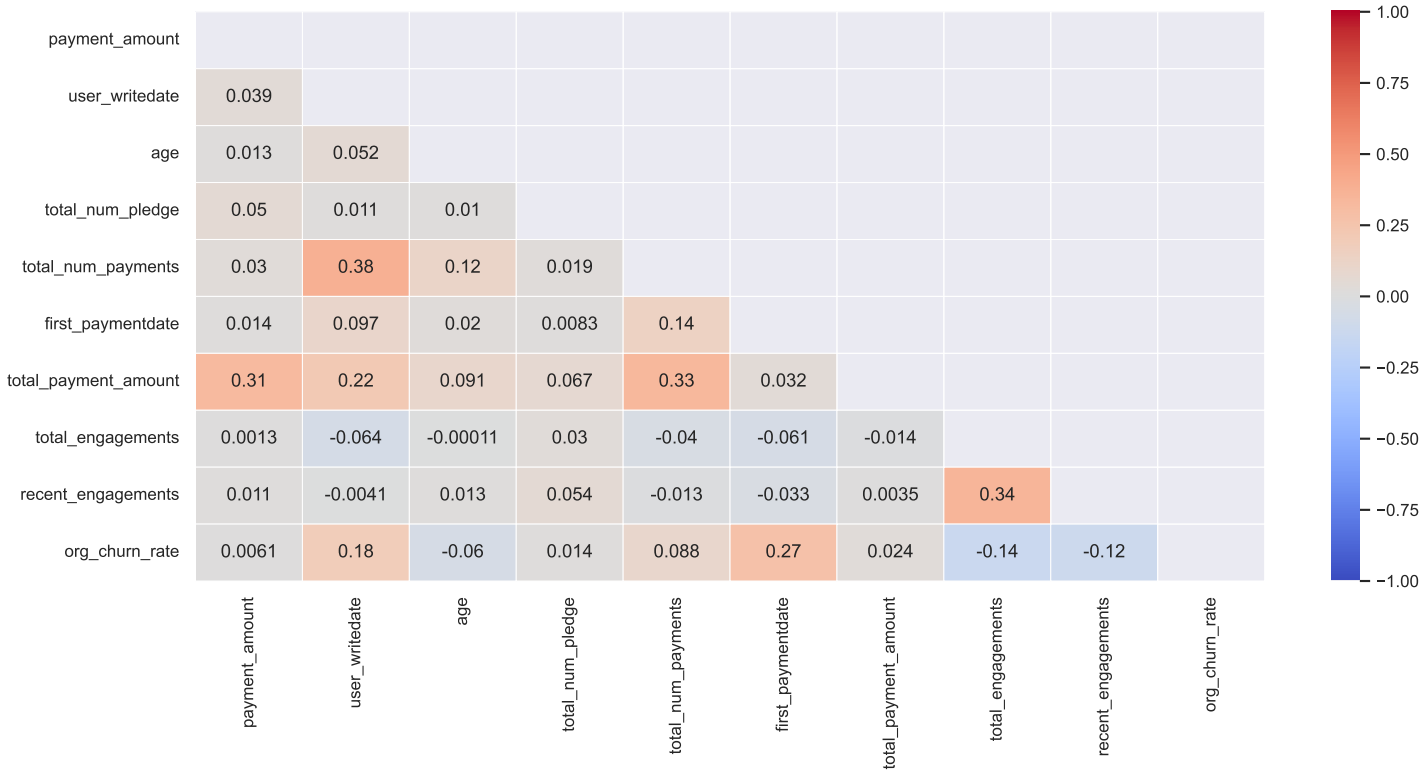

그림 8: 이산형 및 연속형 변수들의 피어슨 상관 계수

3.1.2 이탈 예측 문제의 정의

본 연구에서 정의한 이탈 예측 문제는 다음과 같다. 우선, 구독자 그룹을 $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$으로 가정하고, 납입액을 $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$으로 가정한다. n과 m은 각각 데이터셋에 포함된 구독자와 납입액 고유값의 개수를 나타낸다. 납입액의 고유값은 납입액을 구간별로 묶어서 구간 내의 납입액을 동일하게 셈하는 방식으로 산출된다. 구독자 $u_i \in \mathcal{U}$에 의해 발생한 납입액 시퀀스는 $\mathrm{X}_i = \{v_1^i, v_2^i, \ldots, v_t^i\}$으로 표현되며 t는 구독자가 납입을 한 달을 의미한다. 시퀀스가 12개월로 구성되어 있다면 자연수 $t \in \mathbb{N}$는 [1,12] 범위에 포함되는 값을 가진다.

$\mathrm{Y}_i = \{y_1^i, y_2^i, \cdots, y_t^i\}$는 납입액 $v^i \in \mathrm{X}_i$에 대한 각 달의 미납 여부를 나타낸다. 납입액이 0보다 큰 경우에는 납입을 의미하는 0, 납입액이 0인 경우에는 미납을 의미하는 1이 부여된다. 본 연구의 목표는 과거 12개월 동안의 시퀀스 X_i 정보를 바탕으로 다음 달의 미납 여부 y_{t+1}^i를 예측하는 것이다.

3.2 연구 방법론

본 절에서는 이번 연구에서 제안한 Churn BST 모델의 구조와 구체적인 구현 방법을 살펴보고 선행연구인 BST 모델과의 비교를 통해 구독 비즈니스를 대상으로 Churn BST 모델이 어떤 차별점을 갖는지를 살펴본다.

3.2.1 Churn BST 모델의 구조

Churn BST 모델은 1) Embedding Layer와 Transformer Layer를 이용한 Behavior Encoding 단계, 2) Information Aggregation 단계, 3) Feed-

forward Layer와 Sigmoid Function을 이용하는 Decision Making 단계로 구성된다.

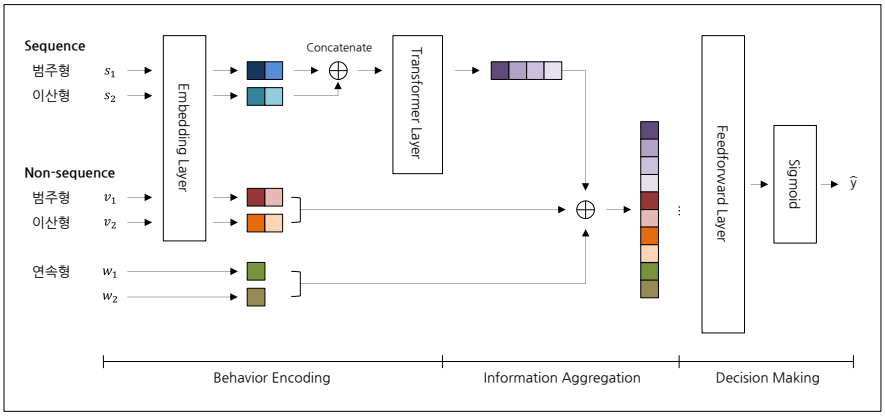

그림 9: Churn BST 아키텍처

가장 처음의 Behavior Encoding 단계에서는 효과적인 학습을 위해 행동 시퀀스와 보조 정보를 더 낮은 차원으로 임베딩한다. Churn BST의 학습 데이터는 3가지 유형의 Sequence 데이터와 Non-sequence 데이터로 구분되며 연속형을 제외한 데이터는 Embedding Layer를 거치게 된다.

수식 3.1에서 $|V|$는 범주형 데이터로 전처리한 월별 납입액의 개수이고 d_v는 임베딩 차원의 크기이다. 이를 트랜스포머 인코더에 입력하여 구독자의 납입 이력을 모델링한다.

$$\mathbf{W_v} \in \mathbb{R}^{|V| \times d_v} \tag{3.1}$$

수식 3.2에서 $|D|$는 보조 정보의 개수이고d_o는 임베딩 차원의 크기이다. Information Aggregation 단계에서는 최종적으로 행렬 W_v와 W_o를 결합하여 새로운 행렬 S를 만든다.

$$\mathbf{W_o} \in \mathbb{R}^{|D| \times d_o} \tag{3.2}$$

행렬 S는 Feedforward Network (FFN)의 입력으로 이용되며 Decision Making 단계에서는 4개의 Hidden Layer를 이용해 이진 분류 작업을 수행한다.

$$F = LeakyReLU(\mathbf{SW^{(1)}} + b^{(1)})\mathbf{W^{(2)}} + b^{(2)} \tag{3.3}$$

마지막으로 Cross-entropy Loss를 이용해 손실값을 계산하며 Embedding, Transformer, FFN Layer의 파라미터를 업데이트한다.

$$L = -\frac{1}{N} \sum_{(x,y) \in D} (ylogp(x) + (1-y)log(1-p(x)) \tag{3.4}$$

3.2.2 Churn BST 모델의 구현 상세

Churn BST는 딥러닝 프레임워크인 PyTorch 1.10.2 버전으로 구현되었다. 임베딩 레이어는 17개의 변수를 각각 학습하며, 4개의 인코더 레이어로 구성된 트랜스포머는 활성화 함수로 GELU를 이용한다. FFN은 4개의 Hidden 레이어와 1개의 출력 레이어로 구성된다. 모든 레이어를 합친 모델 전체의 학습 파라미터는 총 140만 개이다.

표 6: Churn BST 임베딩 레이어의 구성

번호	변수명	설명	파라미터 수	차원
1	org_id	조직 id	12	2
2	user_id	구독자 id	426,000	2
2	sex	성별	8	2
3	age_group	연령 그룹	33	3
4	region	지역	76	4
5	cellphone	휴대전화번호 확보 여부	2	1
6	email	이메일 확보 여부	2	1
7	account	로그인 계정 확보 여부	2	1
8	zipcode	거주지 우편번호 확보 여부	2	1
9	receipt	영수증 발급신청 여부	3	1
10	is_text	문자 수신동의 여부	3	1
11	is_email	이메일 수신동의 여부	3	1
12	is_post	우편 수신동의 여부	3	1
13	is_call	전화 수신동의 여부	3	1
14	payments	시퀀스 내 월별 납입액	180	6
15	payment_methods	시퀀스 내 월별 결제수단	3	1
16	months_position	시퀀스 내 월별 납입순번	34	2

표 7: Churn BST 네트워크의 구성

네트워크	크기	비고	파라미터 수
Transformer	Encoder 4개	GELU, d_model: 9, nhead: 3, Dropout: 0.1	157,000
Feedforward	Hidden Layer 4개	LeakyReLU	840,000

3.2.3 Churn BST와 BST 모델의 비교

본 연구에서는 선행 연구인 BST에서 효과가 입증된 트랜스포머를 이용한 시퀀스 데이터 모델링 방법과 보조 데이터의 결합 방식을 토대로 구독 비즈니스의 이탈 예측 과업을 효과적으로 수행하는 Churn BST 모델을 제안하였다.

전자상거래 비즈니스를 대상으로 제안된 BST 모델은 사용자와 상품 사이에 일어나는 상호작용 시퀀스를 모델링하고, 이를 연령과 성별 같은 사용자의 인구통계학적 데이터와 결합하여, 최종적으로 상품 추천 과업을 수행하는 알고리즘을 수행한다. 하지만 상품의 종류가 단일하거나 소수의 상품만이 존재할 경우에는 상품과 사용자 간의 상호작용 패턴을 효과적으로 학습하지 못하는 문제가 발생하며 이로 인해 정기적인 약정과 납입 패턴을 포착하는 것이 중요한 구독 비즈니스를 대상으로는 행동 데이터에 대한 모델링 성능이 감소하게 된다.

Churn BST 모델은 정기 납입자에 대한 행동 데이터 모델링 방법을 제안하여 모델이 더 다양한 유형의 데이터를 입력 받고 정기 납입자의 행동을 구독이라는 맥락 안에서 종합적으로 해석할 수 있는 방법을 제안하였다. 또한, 플랫폼 관점에서 모델이 복수의 구독 비즈니스를 대상으로 보다 강건한 (Robust) 예측 성능을 낼 수 있도록 네트워크의 크기를 키우고 세부 구조를 변경함으로써 Unseen 데이터에 대한 성능을 높일 수 있었다.

표 8은 이용 목적, 행동 데이터 모델링, 뉴럴 네트워크 구조를 중심으로 정리된 Churn BST와 BST 모델 간의 차이점을 보여준다.

표 8: Churn BST와 BST 모델의 비교

비교 항목	Churn BST의 특징	BST의 특징
1. 이용 목적		
- 적용 대상	구독 비즈니스	전자상거래 비즈니스
- 수행 과업	이탈 예측	상품 추천
2. 행동 데이터 모델링		
- Sequence 데이터	범주화된 납입액 패턴 학습	Item id 학습
- Non-sequence 데이터	범주형, 이산형, 연속형	범주형, 이산형
- 불균형 데이터 학습법	WeightedRandomSampler	없음
- 데이터 Aggregation	Concatenate	Concatenate
3. 뉴럴 네트워크 구조		
3.1 Embedding Layer		
- Embedding Module	nn.Embedding	nn.Embedding
3.2 Transformer Layer		
- Transformer Block	4	1
- Positional Embedding	Fixed Window	$pos(v_i) = t(v_t) - t(v_i)$
- Activation Function	GELU	ReLU
- Dropout	0.1	0.2
3.3 Feed Forward Layer		
- Hidden Layer Shape	1024 * 512 * 256 * 128 * 64	1024 * 512 * 256
3.4 Loss Function	Cross-entropy	Cross-entropy
3.5 Optimizer	Adadelta	Adagrad

제 4 장

실험 및 평가

4.1 실험 조건

본 절에서는 이번 연구에서 수행된 실험의 설계조건을 살펴본다. 구독 데이터셋의 전처리 방법, 예측 모델의 정확도를 평가하는 성능 지표의 종류와 의미, 비교 실험을 위해 준비된 다른 예측 모델의 구현 조건을 각각의 항으로 분리하여 기술하였다.

4.1.1 데이터 전처리

그림 10에는 학습 데이터에 적용된 Sliding Windows 방법이 기술되어 있다. 2021년 5월∼2022년 8월 (총 16개월) 동안 취합된 학습 데이터를 Window size: 12개월, Step size: 1개월로 슬라이싱하여 User_id를 기준으로 반복적인 학습이 이루어지도록 했다. 가장 최근의 미납 여부가 훈련 과정에 노출되지 않도록 가장 마지막 Window에 해당하는 2021년 8월∼2022년 8월 기간의 데이터는 모두 Test set으로 이용하였다. 나머지 데이터는 Train set 80%, Validation set 20%로 구분하여 훈련을 진행했다.

연속형 변수에서 특정 입력값이 학습을 저해하는 문제를 방지하기 위해 입력값의 분포를 기준으로 단위 (Scale)를 조정하는 표준화 작업을 수행했다. 전처리 작업에는 Scikit-learn 1.1.3 [31]을 이용하였으며 StandardScaler과 RobustScaler의 2가지 방법을 적용했다. StandardScaler는 대상 관측치 전체를 평균 0, 표준편차 1로 변환하여 정규분포에 가깝게 만

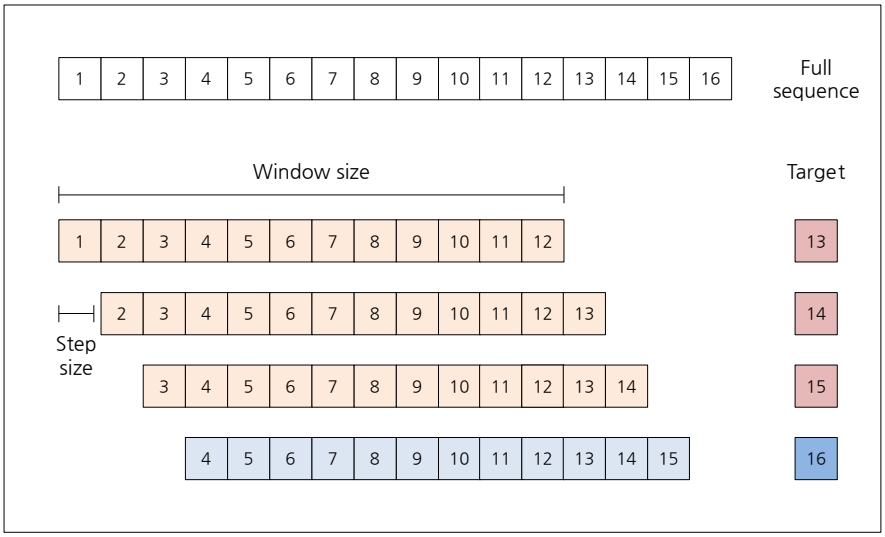

그림 10: Sliding Windows 방법

드는 방법이다. 이상치가 존재할 경우 관측치들의 상대적인 거리가 크게 왜곡되는 단점이 있다.

$$Y = \frac{X - X_{mean}}{\sigma_Y} \tag{4.1}$$

RobustScaler는 대상 관측치를 중앙값 (Median)으로 빼고 사분위값 (Quartile)의 1분위 (25%) ∼ 3분위 (75%) 범위로 나누어 변환하는 방법이다. 단순히 최대값 빼기 최소값을 분모로 사용하는 방법에 비해 이상치의 영향력이 줄어든다는 특징이 있다.

$$Y = \frac{X - X_{median}}{X_{IRQ,75\%} - X_{IRQ,25\%}} \tag{4.2}$$

StandardScaler, RobustScaler 방법으로 연속형 변수를 표준화한 결과는 표 9, 10와 그림 11에 나타나 있다.

표 9: 연속형 변수의 표준화 (StandardScaler)

순번	변수명	개수	평균	표준편차	최소값	최대값
1	age	841191	0	1	-2.30	2.61
2	total_num_pledge	841191	0	1	-0.17	241.52
3	total_num_payments	841191	0	1	-1.37	15.58
4	total_engagements	841191	0	1	-0.09	163.76
5	recent_engagements	841191	0	1	-0.14	102.31
6	user_writedate	841191	0	1	-1.38	9.20
7	first_paymentdate	841191	0	1	-1.43	5.69
8	total_payment_amount	841191	0	1	-0.99	10.77
9	payment_amount	841191	0	1	-1.11	53.86

표 10: 연속형 변수의 표준화 (RobustScaler)

순번	변수명	개수	평균	표준편차	최소값	최대값
1	age	841191	-0.12	0.80	-1.96	1.96
2	total_num_pledge	841191	0.22	1.30	0	314.00
3	total_num_payments	841191	0.30	0.93	-0.97	14.76
4	total_engagements	841191	0.22	2.36	0	387.00
5	recent_engagements	841191	0.04	0.26	0	27.00
6	user_writedate	841191	0.32	0.90	-0.93	8.60
7	first_paymentdate	841191	0.34	0.91	-0.96	5.52
8	total_payment_amount	841191	0.34	1.18	-0.83	13.06
9	payment_amount	841191	0.02	1.82	-2.00	98.86

4.1.2 성능 지표

본 항에서는 본 연구에서 진행한 실험을 평가하기 위해 사용한 성능 지표의 종류와 의미에 대해 설명한다.

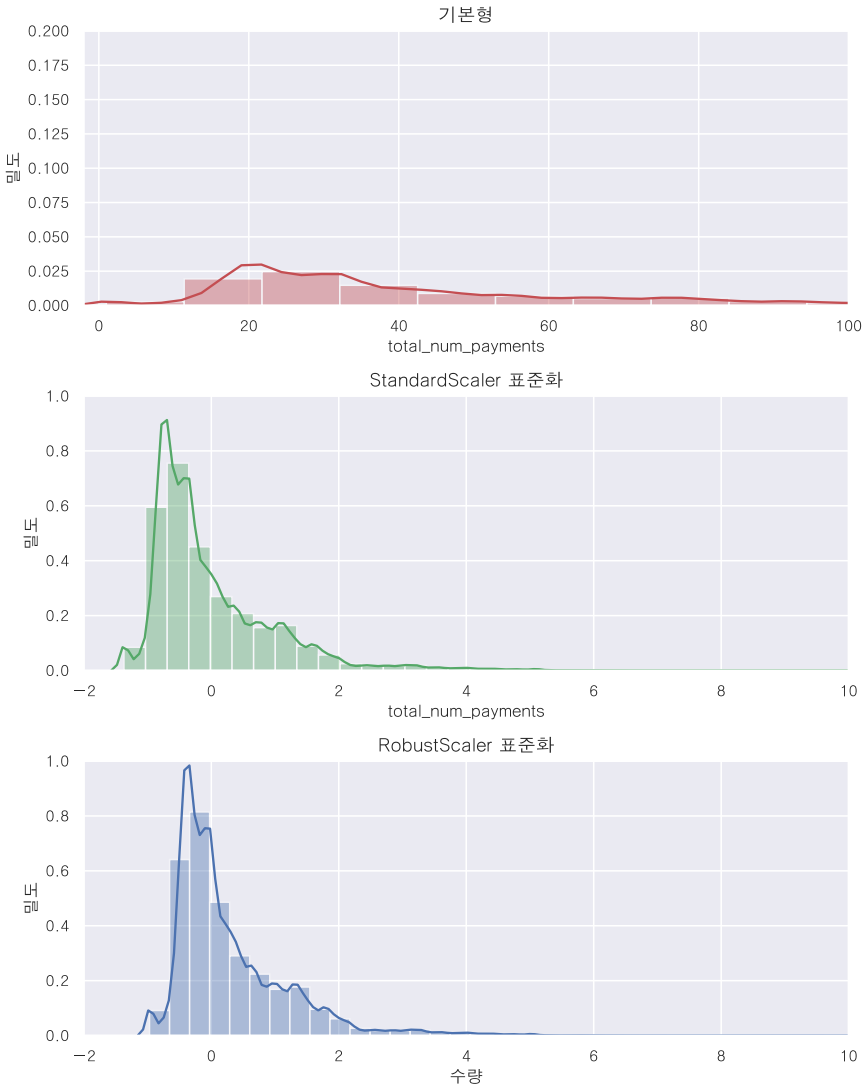

그림 11: 연속형 변수 total_num_payments의 표준화

4.1.2.1 오차 행렬

오차 행렬 (Confusion Matrix)은 분류 모델이 예측값과 실제값 사이의 차이를 True Positive (TP), False Positive (FP), False Negative (FN), True

Negative (TN) 4개의 요소로 구분한 표를 의미한다. 이진 분류 모델의 평가 지표로 널리 활용되고 있다.

본 연구에서 TP는 모델이 1 (미납)로 예측했는데 실제값도 1 (미납)인 경우를 의미하고, FP는 모델이 1 (미납)로 예측했는데 실제값은 0 (납입)인 경우를 의미한다. FN은 모델이 0 (납입)으로 예측했는데 실제값은 1 (미납)인 경우를 나타내고, TN은 모델이 0 (납입)으로 예측했는데 실제값도 0 (납입)인 경우를 나타낸다.

		Predict	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

그림 12: 오차 행렬 (Confusion Matrix)

오차 행렬을 통해 도출된 TP, FP, FN, TN 4개의 요소를 조합하면 정확도 (Accuracy), 재현율 (Recall), 정밀도 (Precision), F-Score 4개의 성능 지표를 정의할 수 있다.

정확도 (Accuracy)는 예측값의 긍정 및 부정 여부와 관계 없이 예측한 값이 참이었는지만을 측정하는 지표이다. 예측 과정은 무시하고 예측 결과를 평가하는 지표로 볼 수 있다. 실제값이 비교적 균등하게 분포해 있는 상황에서는 정확도가 중요한 의미를 갖는다. 하지만 실제값이 불균형하게 분포되어 있는 상황에서는 정확도 만으로는 변별력이 떨어지는 약점이 있다. 예를 들어, 99%의 정상제품과 1%의 불량제품이 존재하는 상황에서 불량제품을 예측하는 모델의 성능을 평가한다고 했을 때 모든 제품을 정상제품으로 예측한다면 모델이 99%의 정확도를 달성하지만 불량제품의 예측에는 모두 실패하는 상황이 발생한다.

$$Accuracy = \frac{TP+FP}{TP+FP+FN+TN} \tag{4.3}$$

정밀도 (Precision)는 참으로 예측하였을 때 실제값도 참인 확률을 의미한다. 참에 대한 예측이 틀릴 경우 큰 비용이 발생하는 상황에 적합한 성능 지표라고 할 수 있다. 예를 들어, 금융 사기를 수사하는데 많은 인력과 시간이 소요된다고 했을 때 실제로는 정상적이었던 거래를 사기로 예측한 경우 불필요한 비용이 발생한다.

$$Precision = \frac{TP}{TP+FP} \tag{4.4}$$

재현율 (Recall)은 실제값이 참인 경우를 참으로 정확하게 예측한 확률을 의미한다. 실제값이 참인 경우를 틀리지 않고 예측해야 하는 상황에 적합한 성능 지표라고 할 수 있다. 예를 들어, 코로나에 양성인 사람을 최대한 많이 판별하는 것이 중요한 상황에서는 모델의 재현율이 충분히 높은지를 확인해야 한다. 반대로 실제값이 거짓인 상황에 대해서는 측정을 하지 않기 때문에 모든 값을 참으로 예측하면 임의로 재현율을 높일 수 있다는 단점이 있다.

$$Recall = \frac{TP}{TP+FN} \tag{4.5}$$

F-Score는 정밀도와 재현율을 조화평균으로 계산한 지표를 의미한다. 정밀도와 재현율은 실제값이 참인 경우 또는 예측값이 참인 경우와

같이 어느 한 쪽만을 측정하기 때문에 다른 쪽의 성능을 고려하지 못한다는 약점이 있다. 분류 모델에서는 재현율이 높아지는 상황에서 정밀도가 떨어지는 등의 트레이드 오프 관계가 발생하는 경우가 있는데 F-Score를 이용하면 재현율과 정밀도 중 어느 한 쪽으로 성능이 치우는 경우를 방지할 수 있다.

조화평균은 두 값의 합은 일정한 상황에서 각각의 값이 이질적일수록 전체적으로는 더 낮은 값을 출력하게 된다. 예를 들어, 재현율이 0.9, 정밀도가 0.1인 경우와 각각의 값이 0.5, 0.5인 경우가 있을 경우 분모의 값은 1로 동일하지만 전자의 조화평균 값은 0.18, 후자의 조화평균 값은 0.5가 된다는 점을 확인할 수 있다.

$$FScore = \frac{2 * Recall * Precision}{Recall + Precision} \tag{4.6}$$

4.1.2.2 ROC Curve

Receiver Operating Characteristic (ROC) Curve는 False Positive Rate가 변할 때 True Positive Rate가 어떻게 따라서 변하는지를 나타내는 곡선이다. 곡선을 통해 모든 임계값 (Threshold)에 대한 예측 모델의 성능을 표현한다.

False Positive Rate가 낮고 True Positive Rate가 높을수록 곡선이 좌측 상단에 위치하며 성능이 좋은 모델이라고 할 수 있다. 하지만 좌표상에 표시된 곡선의 변화를 육안으로 식별하는 것은 쉽지 않다.

ROC Area Under the Curve (AUC)는 단일한 값으로 모델의 성능을 비교하기 위해 도입된 개념이다. ROC 곡선 아래의 면적을 구함으로써

곡선을 그리지 않고도 정량적인 값으로 모델의 성능을 비교할 수 있도록 했다. ROC AUC가 1에 가까울수록 좋은 예측 모델이라고 할 수 있으며, 0.5에 가까우면 제대로 된 예측을 수행하지 못한다고 볼 수 있다.

ROC AUC는 True Positive와 False Positive를 모두 반영하여 산출하기 때문에 Positive와 Negative를 모두 골고루 잘 예측해야 하는 경우에 적합한 성능 지표라고 할 수 있다. 예를 들어, 개와 고양이의 이미지를 보고 어떤 동물인지를 분류해야 하는 상황이라면 ROC AUC를 이용해 모델의 성능을 측정하는 것이 좋다.

4.1.2.3 PR Curve

Precision-Recall (PR) Curve는 임계값 (Threshold)에 따라 변하는 정밀도 (Precision)와 재현율 (Recall)을 표현한 성능 지표이다. PR Curve에서 X축은 재현율, Y축은 정밀도를 나타낸다.

정밀도와 정확도가 모두 높을수록 좋은 예측 모델이라고 할 수 있으며 곡선이 우측 상단으로 이동한다. 하지만 일반적으로 0에서 1사이의 값을 가지는 임계값의 크기에 따라 정밀도와 정확도 사이에 트레이드오프 관계가 발생하며 PR Curve는 완만한 곡선을 이루게 된다.

PR Area Under the Curve (AUC)는 PR Curve의 크기를 정량적으로 비교하기 위해 곡선 아래의 면적을 구한 지표이다. PR AUC를 계산할 때는 적분을 이용하는데 X축 재현율 각각의 임계값에 대한 Y축 정밀도의 평균 (Average Precision)으로 이해할 수 있다.

정밀도와 재현율의 계산식을 살펴보면 True Negative가 영향을 주지 않는다는 점을 알 수 있다. 즉, PR AUC는 Positive 클래스를 예측하는 중요도가 Negative 클래스를 예측하는 중요도 보다 높을 때 적절한 역할을 하는 성능 지표이다. 예를 들어, 암을 진단하는 상황에서는 암이 발병하지

않았다는 것을 정확히 예측하는 것 보다는 암이 발병했다는 것을 정확히 예측하는 것이 더 중요하다고 할 수 있다.

본 연구에서 사용하는 데이터셋은 Positive 클래스의 비율이 평균적으로 7.94% 밖에 되지 않는 불균형한 분포를 갖고 있다. 또한, 구독 비즈니스의 특성상 미납자를 예측하여 예방 조치를 취하는 것이 미납자를 예측하지 못 해서 손해를 보는 것 보다 더 중요하다. 따라서 본 연구에서는 데이터셋의 불균형한 분포를 고려하면서 Positive 클래스에 대한 예측 정확도를 측정하는데 적합한 PR AUC를 가장 중요한 성능 지표로 선정했다.

4.1.3 비교 모델

본 연구에서는 기존 머신러닝 방법론과의 비교를 위해 가장 널리 이용되는 머신러닝 패키지 중 하나인 Scikit-learn 1.1.3 [31]을 활용해 총 8개의 비교 모델을 구현했다. Scikit-learn을 이용하면 통일된 인터페이스를 이용해 선행 연구를 통해 검증된 방식으로 머신러닝 알고리즘을 구현할 수 있다는 장점이 있다.

1. **K-Nearest Neighbor** [32]: KNeighborsClassifier API를 이용해 구현했으며 하이퍼파라미터는 $n_neighbors = 5$으로 설정했다.

2. **Logistic Regression** [5]: LogisticRegression API를 이용해 모델을 구현했으며 하이퍼파라미터는 기본값을 이용했다.

3. **Decision Tree** [7]: DecisionTreeClassifier API를 이용해 구현했으며 하이퍼파라미터는 기본값을 이용했다.

4. **Random Forest** [6]: RandomForestClassifier API를 이용해 구현했으며 기본값 이외의 하이퍼파라미터는 $n_estimators = 500$,

$oob_score = True, \quad n_jobs = -1, \quad max_features =$ "$auto$", $class_weight =$ "$balanced$"로 설정했다.

5. **Gradient Boosting** [8]: GradientBoostingClassifier API를 이용해 구현했으며 하이퍼파라미터는 기본값을 이용했다.

6. **AdaBoost** [9]: AdaBoostClassifier API를 이용해 구현했으며 하이퍼파라미터는 기본값을 이용했다.

7. **XGBoost** [10]: XGBClassifier API를 이용해 구현했으며 기본값 이외의 하이퍼파라미터는 $n_estimators = 1000, \quad learning_rate = 0.1$ $max_depth = 50, \quad subsample = 0.2, \quad early_stopping_rounds = 100,$ $eval_metric =$ "auc"로 설정했다.

8. **LightGBM** [11]: LGBMClassifier API를 이용해 구현했으며 GridSearchCV API를 이용해 기본값 이외의 하이퍼파라미터에 대한 최적화를 진행했다. 최종적인 하이퍼파라미터는 $n_estimators = 1000,$ $learning_rate = 0.1, \quad num_leaves = 400, \quad max_depth = 50,$ $min_child_samples = 50, \quad subsample = 0.05, \quad eval_metric =$ "auc", $early_stopping_rounds = 100$로 설정되었다.

4.2 평가 및 분석

본 절에서는 이번 연구에서 제안한 Churn BST 모델의 예측 결과를 분석하고, 머신러닝 및 딥러닝 모델과의 비교를 통해 성능을 평가한다. 또한, 학습에 이용되지 않았던 Unseen 데이터셋을 이용해 모델의 일반화가 잘 이루어졌는지를 검증한다.

4.2.1 기본 실험

기본 실험은 학습에 사용된 조직 A~E가 포함된 데이터셋을 이용해 진행했다. 전체 2021년 5월부터 2022년 8월까지의 데이터 중 마지막 시퀀스에 해당하는 2021년 8월부터 2022년 8월까지의 데이터를 Test set으로 분할했고, 나머지 시퀀스 중 80%는 Train set으로 이용 20%는 Validation set으로 무작위 분할했다.

Chun BST 모델은 Batch size 256을 기준으로 총 40,000 Step에 걸쳐 학습이 진행되었다. 그림 13은 모델의 Train loss를 나타낸다. 그림 14은 8,000 Step 이후 Validation set의 PR AUC 지표가 감소하여 과적합이 발생한 모습을 보여준다.

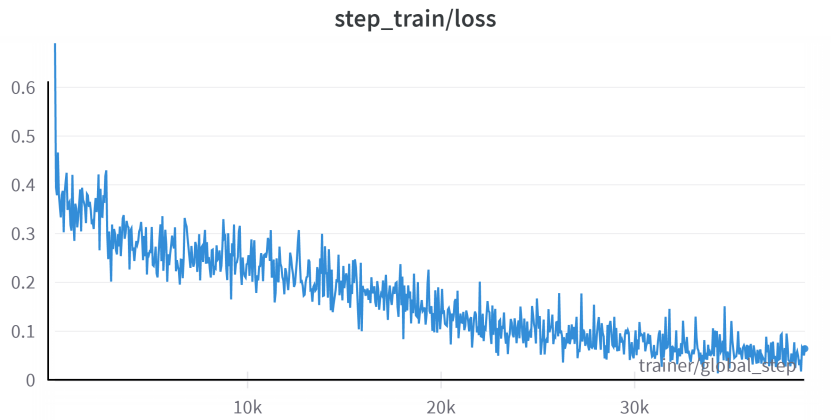

그림 13: Churn BST의 Train Loss

Test set을 기준으로 성능을 평가했을 때 학습이 완료된 Churn BST 모델은 정확도 (Accuracy) 0.9594, 정밀도 (Precision) 0.9127, 재현율 (Recall) 0.6357, F-Score 0.7495, ROC AUC 0.9508, PR AUC 0.8302를 달성하는 준수한 성능을 보여주었다.

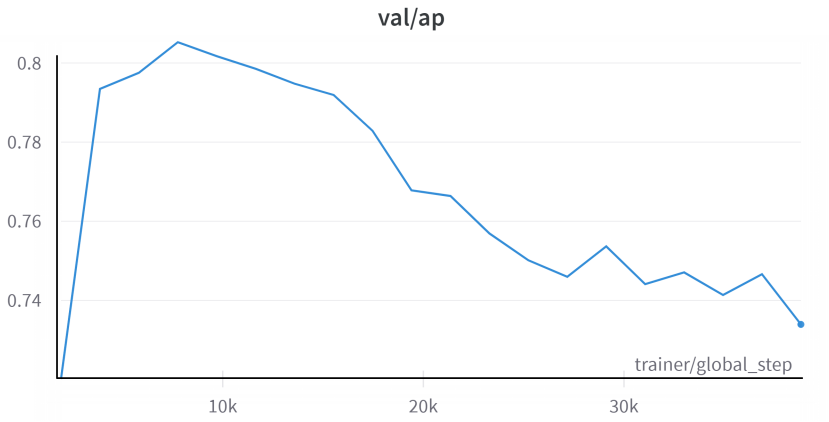

그림 14: Churn BST의 Validation PR AUC

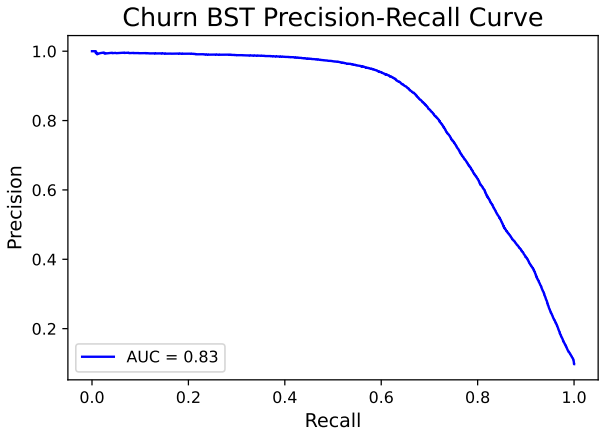

그림 15: Churn BST의 Precision-Recall Curve

Test set에 대한 상세한 예측 결과는 Confusion Matrix 형식으로 그림 17에 나타나 있다. Confusion Matrix에서 X축은 모델이 예측한 값이고, Y축은 실제 값을 의미한다. 본 연구에서는 미납을 1, 납입을 0으로 표기했다.

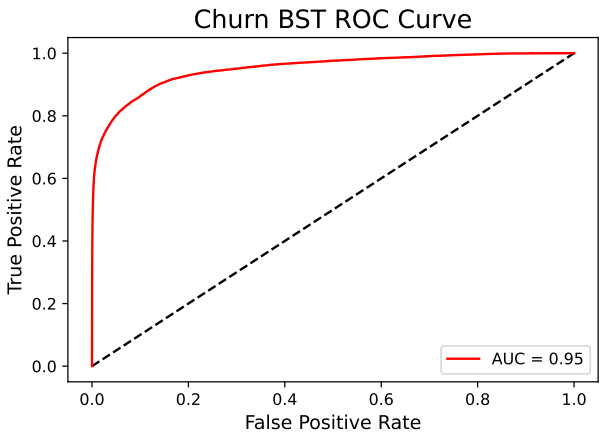

그림 16: Churn BST의 ROC Curve

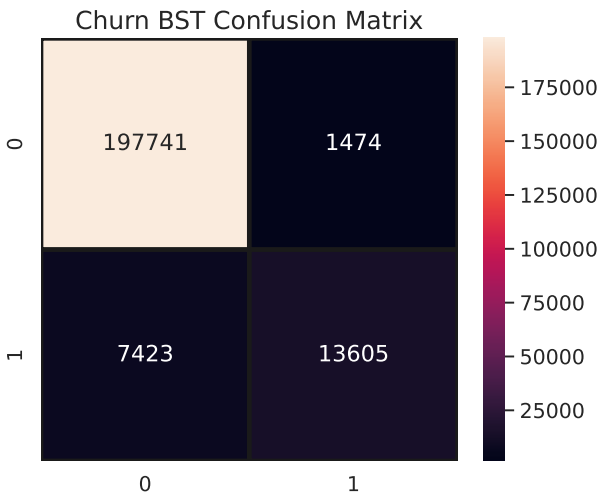

그림 17: Churn BST의 Confusion Matrix

Churn BST 모델은 머신러닝 방법론으로 구현한 총 9개의 모델과 비교했을 대부분의 지표에서 더 뛰어난 성능을 보여주었다. 특히, 비교 모델 중 가장 높은 Precision 0.9127을 유지하면서도 Recall을 0.6357까지 개선함으로써 Chrun BST 모델이 비교 모델에 비해 더 많은 미납자를 정확하게 예측할 수 있다는 점을 보여주었다. 개별 모델의 자세한 성능은 표 15에서 확인할 수 있다. 해당 지표에서 가장 높은 성능을 달성한 경우 수치를 굵게 표시했으며, 두 번째로 높은 성능을 달성한 경우에는 밑줄로 표시했다.

표 11: 머신러닝 모델과의 성능 비교

예측 모델	Accuracy	Precision	Recall	F-Score	ROC AUC	PR AUC
KNN	0.9385	0.7779	0.4981	0.6074	0.8127	0.5583
Logistic Regression	0.9038	0.4448	0.0322	0.0600	0.8050	0.3480
Decision Tree	0.9374	0.7172	0.5691	0.6346	0.7728	0.4496
Random Forest	0.9569	0.9028	0.6143	0.7311	0.9320	0.7989
Gradient Boosting	0.9581	0.9019	<u>0.6302</u>	0.7420	0.9370	0.8063
AdaBoost	0.9522	0.9032	0.5597	0.6911	0.9196	0.7697
XGBoost	0.9581	0.9045	0.6277	0.7411	0.9471	0.8196
LightGBM	<u>0.9587</u>	<u>0.9100</u>	0.6295	<u>0.7442</u>	**0.9529**	<u>0.8285</u>
Churn BST	**0.9594**	**0.9127**	**0.6357**	**0.7495**	<u>0.9508</u>	**0.8302**
개선율	0.07%	0.30%	0.87%	0.71%	-0.22%	0.21%

머신러닝 비교 모델의 PR Curve, ROC Curve, Confusion Matrix는 그림 18, 19, 20에서 확인할 수 있다. 머신러닝 비교 모델 중 가장 높은 PR AUC와 ROC AUC는 보인 방법론은 LightGBM이었으며 각각 0.8285, 0.9529를 달성했다.

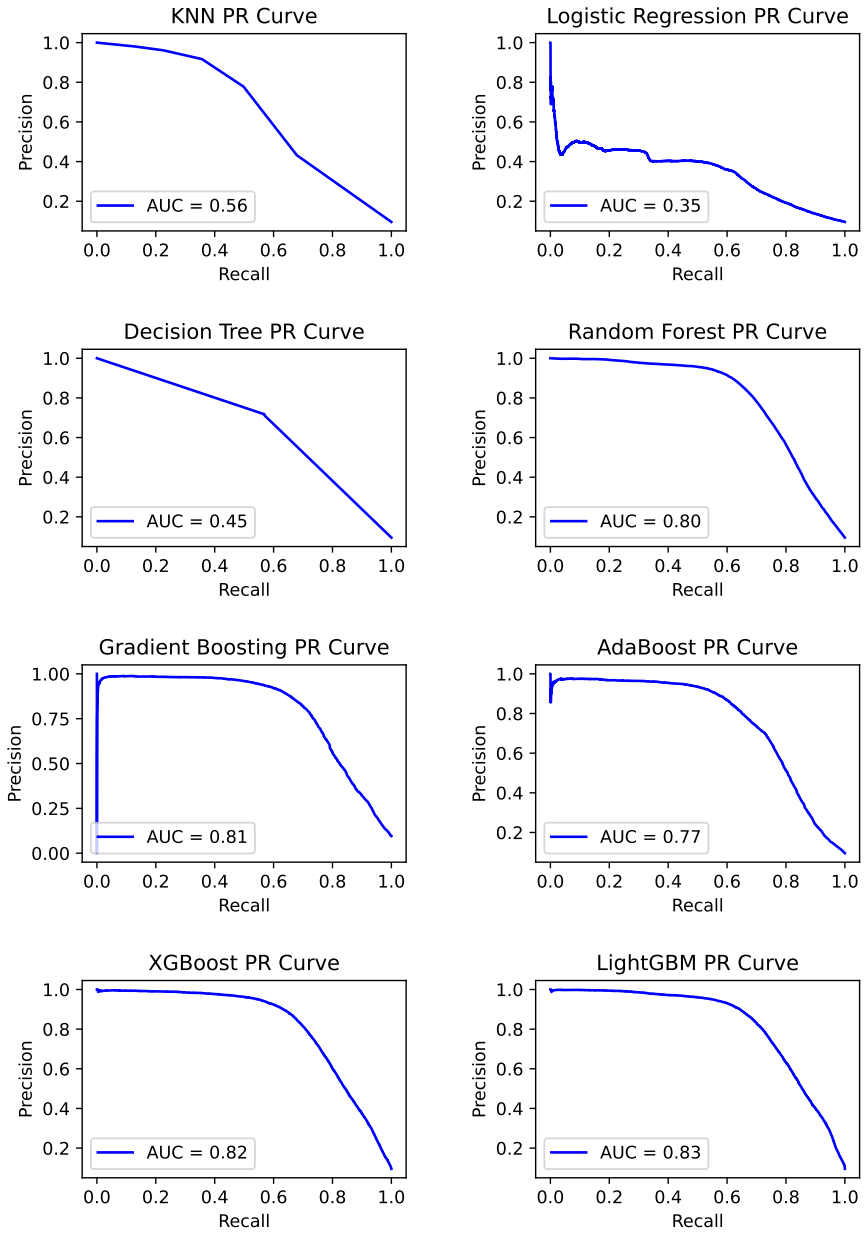

그림 18: 머신러닝 비교 모델의 Precision-Recall Curve

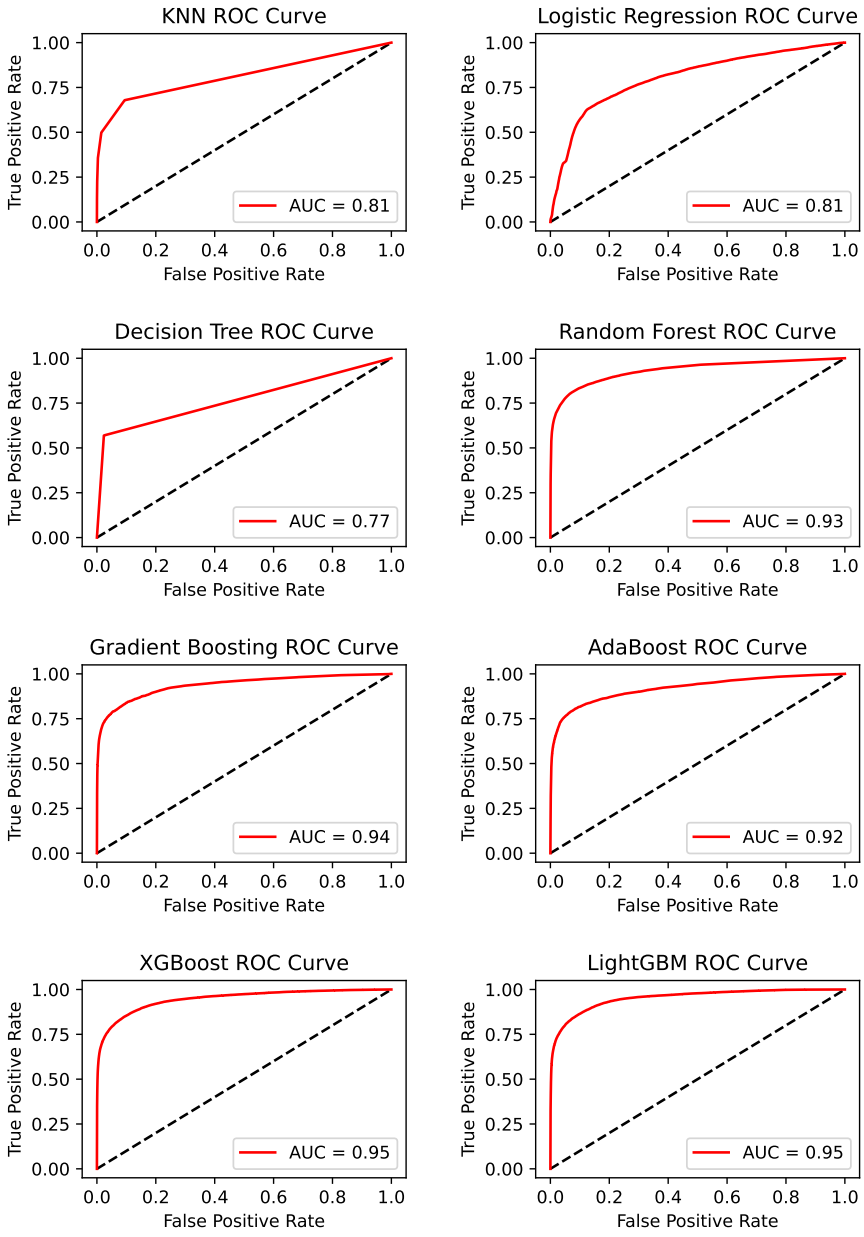

그림 19: 머신러닝 비교 모델의 ROC Curve

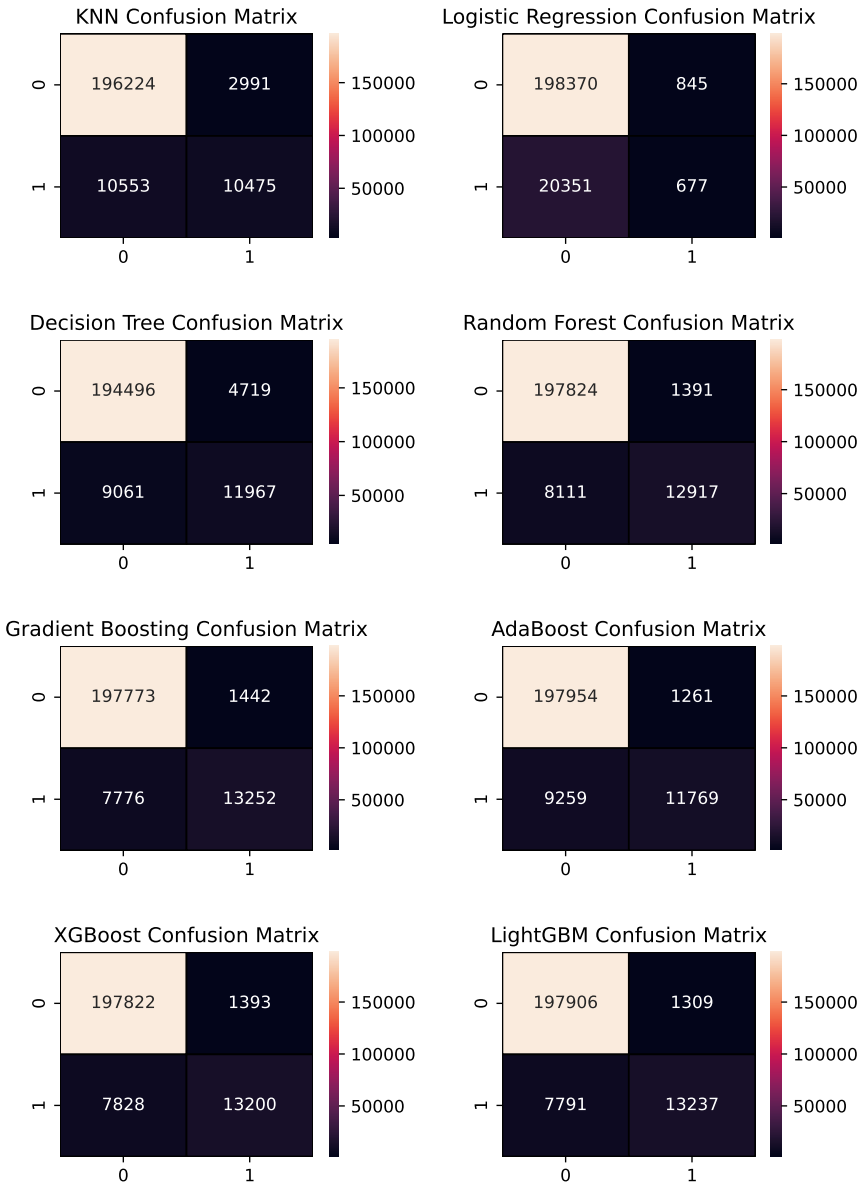

그림 20: 머신러닝 비교 모델의 Confusion Matrix

Churn BST 모델에 사용된 트랜스포머의 시퀀스 모델링 능력을 확인하기 위해 딥러닝 방법으로 구현한 MLP, RNN, LSTM 모델과의 성능 비교를 진행했다. MLP 모델에서는 시퀀스를 모델링 하는 네트워크가 이용되지 않았으며, RNN과 LSTM 모델에서는 Churn BST 모델에서 이용된 트랜스포머 네트워크가 각각 RNN과 LSTM으로 대체되었다.

표 12의 PR AUC 지표를 살펴보면 시퀀스 모델링의 효과와 트랜스포머 네트워크의 기여도를 판단할 수 있다. 먼저 시퀀스 네트워크를 제거함으로써 납입 순서를 고려하지 않았던 MLP 모델은 0.7992의 가장 낮은 PR AUC를 보여주었다. 납입 순서를 모델링 한 RNN과 LSTM 모델은 MLP 모델보다 더 나은 성능을 보여주었으나 트랜스포머를 이용한 Churn BST 보다는 떨어지는 결과가 나왔다.

표 12: 딥러닝 모델과의 성능 비교

예측 모델	Accuracy	Precision	Recall	F-Score	ROC AUC	PR AUC
MLP	0.9579	0.9082	0.6223	0.7385	0.9243	0.7992
RNN	0.9571	0.9080	0.6131	0.7319	<u>0.9454</u>	<u>0.8120</u>
LSTM	<u>0.9583</u>	<u>0.9094</u>	<u>0.6260</u>	<u>0.7416</u>	0.9384	0.8059
Churn BST	**0.9594**	**0.9127**	**0.6357**	**0.7495**	**0.9508**	**0.8302**
개선율	0.11%	0.36%	1.55%	1.07%	0.57%	2.24%

딥러닝 비교 모델의 PR Curve, ROC Curve, Confusion Matrix는 그림 21, 22, 23에서 확인할 수 있다. 딥러닝 비교 모델 중 가장 높은 PR AUC와 ROC AUC는 보인 방법론은 RNN이었으며 각각 0.8120와 0.9454를 달성했다.

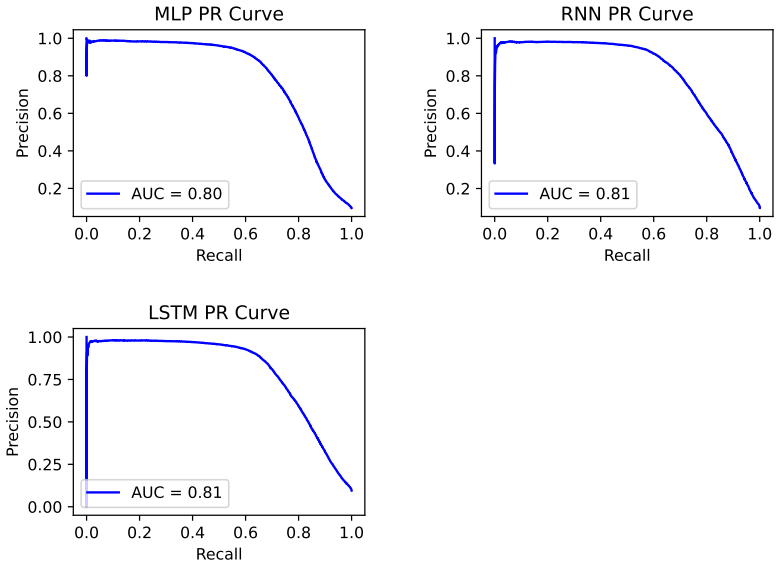

그림 21: 딥러닝 비교 모델의 Precision-Recall Curve

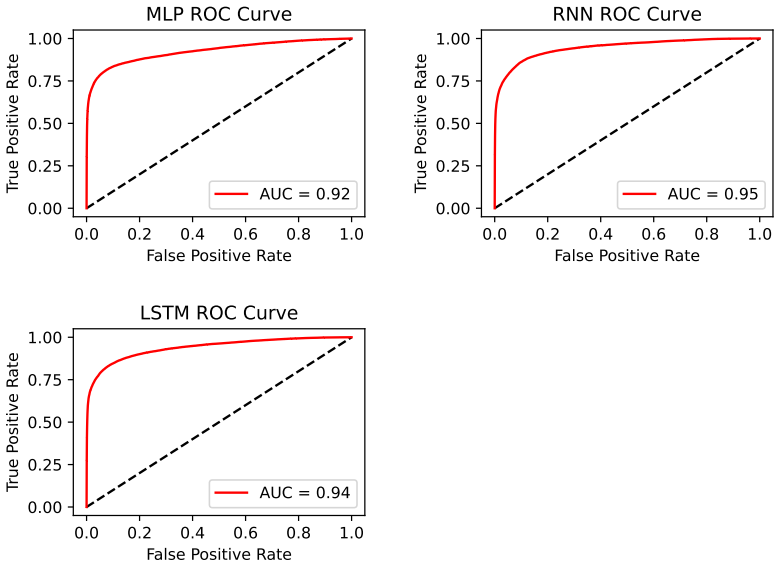

그림 22: 딥러닝 비교 모델의 ROC Curve

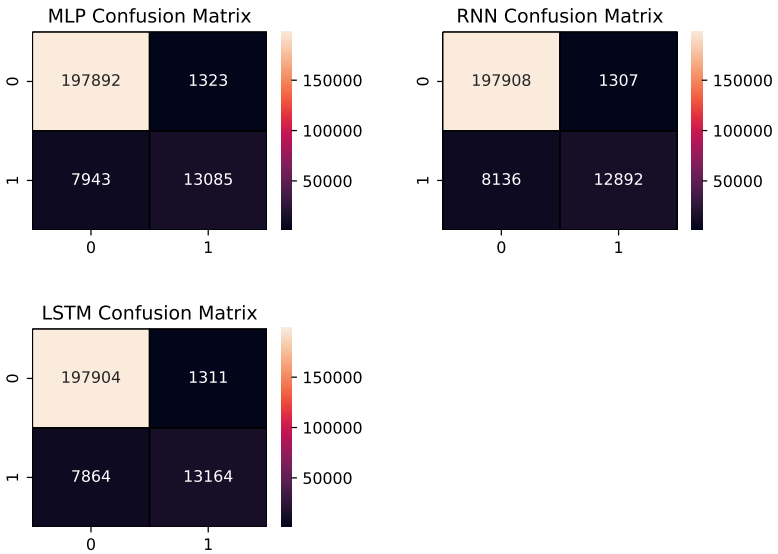

그림 23: 딥러닝 비교 모델의 Confusion Matrix

4.2.2 확장 실험

이탈 예측 모델의 일반화된 성능 (Generalized performance)을 검증하기 위해 학습에 이용되지 않은 Unseen 데이터셋으로 확장 실험을 진행했다. Unseen 데이터셋은 구독 플랫폼 상의 다른 조직인 F와 G의 2021년 8월부터 2022년 8월까지의 시퀀스로 구성되었다. 조직 F는 학습에 이용되었던 데이터셋의 평균 미납률인 7.94% 보다 높은 9.6%의 미납률을 갖고 있으며, 조직 G는 그보다 낮은 5.79%의 미납률을 보여준다.

표 13: Unseen 데이터셋의 조직별 납입 및 미납 분포

조직명	구독자 수	관측치의 개수 (A+B)	납입 수 (A)	미납 수 (B)	미납률
A∼E	213156	841191	774416	66775	7.94%
F	21525	21525	19458	2067	9.60%
G	28910	28910	27235	1675	5.79%

Unseen 데이터셋에서는 대부분의 지표에서 Churn BST가 다른 비교 모델을 앞도하는 성능을 보여주었다. Churn BST의 PR AUC는 0.8921인 반면 두 번째로 높은 PR AUC를 달성한 Random Forest 모델은 0.7082로 25.97%의 성능 차이가 발생했다. 특히, 학습 데이터셋을 대상으로 비교 모델 중 가장 좋은 성능을 보여준 LightGBM은 성능이 현격하게 감소했다. Churn BST와 LightGBM의 PR AUC는 각각 0.8921과 0.4793으로 차이가 86.13%까지 벌어졌다.

확장 실험을 통해 우리는 Churn BST 모델이 새로운 분포를 가진 Unseen 데이터셋에서도 기존에 학습했던 지식을 충분히 활용하여 예측 작업을 수행한다는 사실을 추론할 수 있다. 반대로 LightGBM을 포함한 기존 모델들은 학습 데이터의 패턴에 쉽게 과적합 되며, 특히 확장 실험의 조건처럼 구독자의 시퀀스가 단 1개만 존재하는 상황에서 취약한 모습을 보여주었다.

표 14: Unseen 데이터셋 조직 F에 대한 성능 비교

예측 모델	Accuracy	Precision	Recall	F-Score	ROC AUC	PR AUC
KNN	0.9040	0.0000	0.0000	0.0000	0.5000	0.0960
Logistic Regression	0.8986	0.1081	0.0077	0.0144	0.4948	0.0960
Decision Tree	0.8783	0.3893	0.4698	0.4258	0.6912	0.2401
Random Forest	0.9542	0.8572	0.6275	0.7246	0.9044	0.7082
Gradient Boosting	0.9028	0.4515	0.0585	0.1036	0.8608	0.3850
AdaBoost	0.8881	0.0704	0.0135	0.0227	0.2096	0.0628
XGBoost	0.9067	0.5782	0.1055	0.1784	0.8273	0.3565
LightGBM	0.9104	0.6725	0.1311	0.2194	0.8805	0.4793
Churn BST	**0.9704**	**0.9143**	**0.7634**	**0.8321**	**0.9705**	**0.8921**
개선율	1.70%	6.66%	21.66%	14.84%	7.31%	25.97%

확장 실험의 PR Curve, ROC Curve, Confusion Matrix는 그림 24, 25, 26에서 확인할 수 있다. Churn BST 모델의 조직 F와 G에 대한 PR AUC는

표 15: Unseen 데이터셋 조직 G에 대한 성능 비교

예측 모델	Accuracy	Precision	Recall	F-Score	ROC AUC	PR AUC
KNN	0.9421	0.0000	0.0000	0.0000	0.5000	0.0579
Logistic Regression	0.9331	0.0519	0.0090	0.0153	0.5010	0.0582
Decision Tree	0.8970	0.2334	0.3403	0.2769	0.6308	0.1236
Random Forest	0.9605	0.7785	0.4448	0.5661	0.9177	0.6568
Gradient Boosting	0.9377	0.3167	0.0645	0.1071	0.8513	0.2500
AdaBoost	0.9248	0.0421	0.0137	0.0207	0.2532	0.0419
XGBoost	0.9422	0.5037	0.1618	0.2449	0.8756	0.3296
LightGBM	0.9461	0.5918	0.2233	0.3242	0.9204	0.4728
Churn BST	**0.9755**	**0.9094**	**0.6412**	**0.7521**	**0.9680**	**0.8288**
개선율	1.56%	16.81%	44.15%	32.86%	5.17%	26.19%

각각 0.8921과 0.9399이었으며, LightGBM 모델의 조직 F와 G에 대한 PR AUC는 각각 0.4793과 0.4728을 보여주었다.

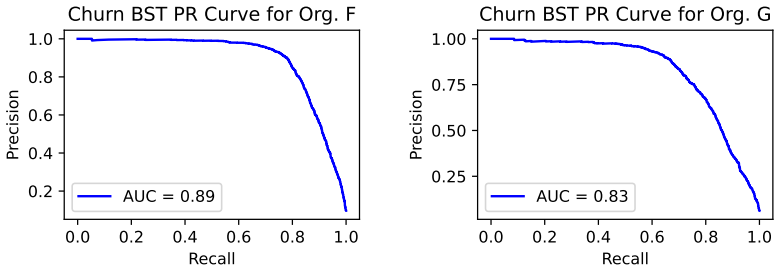

그림 24: Churn BST 확장 실험의 Precision-Recall Curve

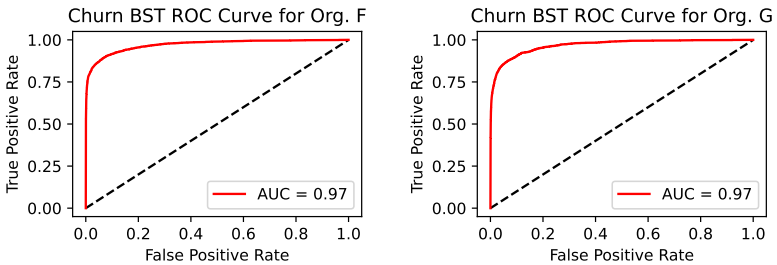

그림 25: Churn BST 확장 실험의 ROC Curve

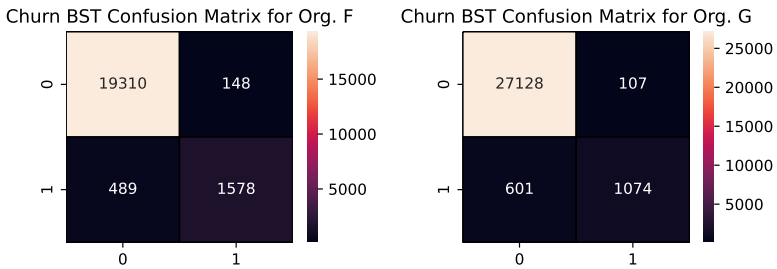

그림 26: Churn BST 확장 실험의 Confusion Matrix

제 5 장

결론

본 연구에서는 트랜스포머를 이용하여 복수의 구독 비즈니스에 적용 가능한 플랫폼 관점의 이탈 예측 모델을 제안하였다. 구독 비즈니스 플랫폼은 여러 비즈니스 조직의 혼재된 구독자 데이터를 포함하고 있으며, 실시간으로 입력되는 새로운 구독자의 정보로 인해 기존 데이터의 분포와 특성이 지속적으로 변경된다는 특징이 있다. 따라서 주로 고정되어 있는 벤치마크 데이터셋에 대한 State-of-the-Art (SOTA) 연구를 진행하는 실험실 환경을 벗어나 Unseen 데이터셋에 대해서도 안정적인 성능을 보여주는 일반화된 모델을 연구하고 적용하는 일이 중요하다.

본 연구를 통해 제안된 Churn BST 모델의 알고리즘은 데이터 전처리, 변수 임베딩, Sequence 데이터의 모델링과 Non-sequence 데이터의 결합 과정으로 구성되어 있다. 보다 엄밀한 성능 실험을 진행하기 위해 8개의 머신러닝 방법론과 3개의 딥러닝 방법론으로 대규모의 비교 모델을 구현하였고, 5개의 조직으로 구성된 84만 개의 학습 데이터셋과 2개의 조직으로 구성된 5만 개의 Unseen 데이터셋을 구성하였다. 비교 실험 결과 Churn BST 모델은 기존의 이탈 예측 방법론보다 Unseen 데이터셋에서의 PR AUC 성능을 25.97% 개선하는 성과를 달성했다.

Churn BST 모델을 현장에 적용하는 과정에서의 향후 연구 과제로는 모델의 경량화를 꼽을 수 있다. 변수의 임베딩과 모델링에 많은 수의 파라미터를 이용하는 트랜스포머 네트워크의 특성상 데이터의 학습과 추론에

상대적으로 많은 컴퓨팅 자원과 시간이 소요된다는 단점이 있다. 더욱 효율적인 임베딩 방법과 파라미터 공유 방법에 대한 연구를 통해 경량화된 모델을 구현할 수 있다면 현장에서의 활용도를 더욱 높일 수 있을 것으로 기대된다. 또한, 단순한 미납 여부 예측을 넘어 금액을 기준으로 한 가중치 예측 모델을 개발한다면 손실액을 기준으로 대응 방안을 수립하는 보다 지능적인 접근과 경제성을 고려한 조치가 가능해 진다. 이를 위해 앞으로 이탈, 유지, 증액으로 구성된 Multi-task Learning 환경을 구성하여 보다 고도화된 모델에 대한 연구를 진행할 계획이다.

참고 문헌

[1] "Market size of the digital subscription economy worldwide in 2020, with a forecast for 2025, by segment." https://www.statista.com/statistics/1295064/market-size-digital-subscription-economy-worldwide-by-segment/. Accessed: 2022-10-11.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[3] Q. Chen, H. Zhao, W. Li, P. Huang, and W. Ou, "Behavior sequence transformer for e-commerce recommendation in alibaba," in *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, pp. 1–4, 2019.

[4] "Machine learning." https://en.wikipedia.org/wiki/Machine_learning. Accessed: 2023-01-30.

[5] J. S. Cramer, "The origins of logistic regression," 2002.

[6] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1, pp. 278–282, IEEE, 1995.

[7] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Routledge, 2017.

[8] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[9] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[10] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[11] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

[12] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," *Advances in neural information processing systems*, vol. 31, 2018.

[13] "Deep learning." `https://en.wikipedia.org/wiki/Deep_learning`. Accessed: 2023-01-30.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[16] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[19] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1419–1428, 2017.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[21] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[22] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206, IEEE, 2018.

[23] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1441–1450, 2019.

[24] G. de Souza Pereira Moreira, S. Rabhi, J. M. Lee, R. Ak, and E. Oldridge, "Transformers4rec: Bridging the gap between nlp and sequential/session-based recommendation," in *Fifteenth ACM Conference on Recommender Systems*, pp. 143–153, 2021.

[25] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[26] P. S. Fader, B. G. Hardie, and K. L. Lee, "Rfm and clv: Using iso-value curves for customer base analysis," *Journal of marketing research*, vol. 42, no. 4, pp. 415–430, 2005.

[27] "Subscription business model." `https://en.wikipedia.org/wiki/Subscription_business_model`. Accessed: 2023-01-30.

[28] F. F. Reichheld, "The one number you need to grow," *Harvard business review*, vol. 81, no. 12, pp. 46–55, 2003.

[29] "Customer churn prevention: A prescriptive solution using deep learning." `https://blog.griddynamics.com/customer-churn-prevention-prescriptive-solution-using-deep-learning/`. Accessed: 2023-01-30.

[30] J. W. Tukey *et al.*, *Exploratory data analysis*, vol. 2. Reading, MA, 1977.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[32] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

Abstract

Churn Prediction with Behavior Sequence Transformer for Subscription Business

Minchang Kim
Graduate School of Practical Engineering
Seoul National University

The subscription business market has been growing continuously in recent years. According to data from the statistics platform Statista, the global digital subscription market, which was valued at $650 billion in 2020, is expected to reach $1.5 trillion in 2025, growing at an average annual rate of 18.2% [1]. From traditional subscription business models (e.g., water purifiers, study books, and mobile communication) to digital subscription business models (e.g., content, music, and software), subscription services are now everywhere.

In order to increase the sustainability of the subscription business, it is crucial to maintain the subscribers' satisfaction. When new customers use the service for a short period, the acquisition cost could rise above the revenue. Therefore, predicting the churn rates and preventing churn is a pivotal activity for creating a sustainable subscription business model.

In this work, we propose the Churn Behavior Sequence Transformer (Churn BST) based on the transformer architecture, which predicts the possibility of nonpayment of subscribers. Churn BST models the interaction sequence of the subscriber through the transformer network and captures the dynamic changes in the subscriber behavior. In addition, Churn BST learns latent representations of unpaid subscribers by concatenating the sequence and non-sequence data. Therefore, it solves the overfitting problem and increases generalized predictive performance.

Churn BST shows higher average prediction accuracy than traditional machine learning and deep learning methodologies. In particular, Churn BST improves Precision-Recall AUC (PR AUC) by 25.97% on the Unseen datasets for which the existing methodologies cause severe performance degradation due to the overfitting problem. The robustness of Churn BST presents a more practical alternative in real-world applications where the data distribution of subscribers continuously changes.

Keywords : Deep Learning, Transformer, Churn Prediction, Imbalanced Classification, Subscription Business

Student Number : 2021-23150