2024.08.26

**회원 만료 시점 예측을 위한**

**머신러닝 모델 개발 보고**

**빅데이터 9기 신유라**

**[목차]**

1. [요약](#요약)
2. [서론](#서론)
3. [개발 환경](#개발환경)
4. [개발 프로세스](#개발프로세스)
5. [사용된 데이터](#사용된데이터)
6. [Result](#result)
7. [Discussion](#discussion)
8. [Conclusion](#conclusion)
9. [소스코드](#소스코드)
10. [약어](#약어)

**1. 요약**

본 보고서는 회원 상태가 "만료"로 전환되는 시점을 예측하기 위해 세 가지 회귀 모델을 적용한 결과를 다룬다. 천재교육 서비스 회원 활동 및 상태 변화 데이터(2022년 9월)를 활용하였으며, 모델의 성능 평가는 RMSE(Root Mean Square Error; 평균 제곱근 오차), 회귀 잔차, 학습 곡선을 기준으로 이루어졌다. 그 결과, 랜덤포레스트 회귀 모델이 가장 우수한 성능을 보이는 모델로 선정되었다. 이 모델은 회원 상태가 만료로 전환되기 전에 맞춤형 마케팅 제공, 회원 피드백 시스템 도입 등의 대응을 가능하게 하여 회원 유지에 효과적으로 활용될 수 있다.

**2. 서론**

**1) 배경**

회원 서비스 운영에서 회원 유지율은 비즈니스의 성과에 직결되는 중요한 요소이다. 특히, 회원이 "만료" 상태로 전환되는 것은 곧 서비스 이용 중단으로 이어지며, 이는 고객 이탈로 직결된다. 고객 이탈은 매출 감소뿐만 아니라, 신규 고객을 확보하는 데 드는 비용과 시간 측면에서도 기업에 큰 부담을 안겨준다. 따라서, 회원이 "만료" 상태로 전환되기 전에 이를 사전에 예측하고, 적절한 대응을 통해 이탈을 방지하는 것은 매우 중요하다.

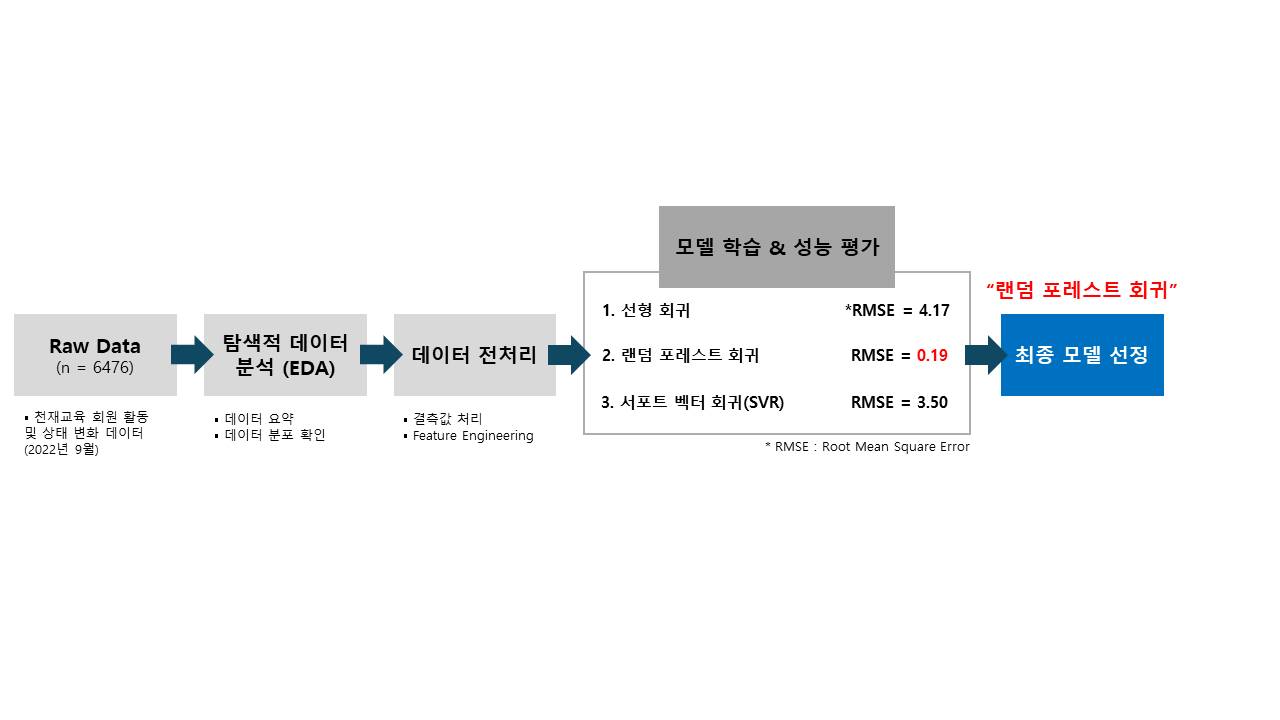
**2) 프로젝트 기획의도**

본 프로젝트의 목적은 회원 상태가 "만료"로 전환되는 시점을 사전에 예측하여 고객 이탈을 최소화하고, 회원 유지율을 향상시키기 위한 기반을 마련하는 데 있다. 이를 위해 다양한 회원 데이터를 분석하여 최적의 독립변수를 도출하고, 그 결과를 바탕으로 세 가지 회귀 모델을 개발하고자 한다. 이 중 가장 성능이 우수한 모델의 예측 결과를 활용하여 개인 맞춤형 혜택 제공, 성과 배지 부여, 회원 피드백 시스템 도입 등 다양한 대응 전략을 강화할 수 있을 것이다. 궁극적으로는 고객 만족도와 충성도를 높여 장기적인 회원 가치를 극대화하는 데 기여하고자 한다.

**3. 개발 환경**

|  |  |  |
| --- | --- | --- |
| 언어 및  라이브러리 | 버전 | 라이선스 |
| Python | 3.12.4 | Python Software Foundation License |
| Pandas | |  | | --- | |  |   2.1.2 | BSD 3-Clause License |
| Seaborn | 0.12.2 | BSD 3-Clause License |
| NumPy | 1.26.0 | BSD 3-Clause License |
| Matplotlib | |  | | --- | |  |   3.8.0 | Matplotlib License |
| scikit-learn | 1.3.1 | BSD 3-Clause License |
| imblearn | 0.11.0 | MIT License |
| SciPy | 1.11.3 | BSD 3-Clause License |

**4. 개발 프로세스**

****

**5. 사용된 데이터**

*\*천재교육 서비스 회원 활동 및 상태 변화 데이터(2022년 9월)*

▪ 데이터 셋 구성

▫ 총 Feature 수 : 44개

▫ 총 Sample 수 : 6,476개

▪ 주요 Feature의 고유값 수

▫ "userid" : 30개

▫ "event\_type" : 3개

▫ "change\_date" : 12개

▪ 데이터 전처리

▫ 결측값 처리

|  |  |  |
| --- | --- | --- |
|  | Feature | |
| **"day\_NN\_status"** | **"memberstatus\_change"** |
| 결측값 처리 방법 | 0으로 대체 | - 결측값이 맨 앞에 존재할 경우, 0으로 대체  - 결측값이 중간에 존재할 경우, 바로 뒤의 값으로 대체 |
| 처리 전 데이터 수 (개) | 4,876 | 4,876 |
| 처리 후 데이터 수 (개) | 6,476 | 6,476 |

▪ Feature Engineering

▫ 새로운 Feature 생성

- "days\_between\_mean" : 사용자별 평균 접속일 간격

- "days\_between\_slope" : 사용자별 접속일 간격 증감 추이

- "mcode\_count" : 사용자별 컨텐츠 이용 개수 총 합

▫ 로그 변환 : "days\_between\_mean", "days\_between\_slope", "mcode\_count"

▫ Min-Max 스케일링(0~1) : "days\_between\_mean", "days\_between\_slope", "mcode\_ count"

▪ 독립변수, 종속변수 선정

▫ **독립변수 : "days\_between\_mean", "days\_between\_slope", "mcode\_count"**

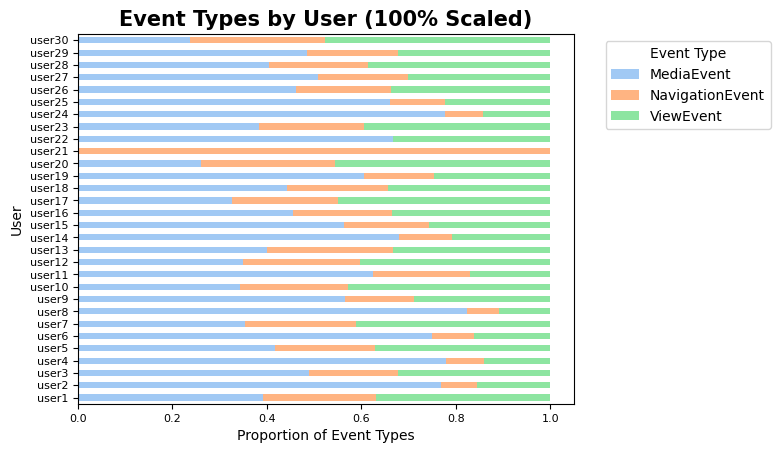
▫ **종속변수 : "change\_date**("만료" 상태 변환 일)"

**6. Result**

**1) 탐색적 데이터 분석(EDA) 및 데이터 전처리**

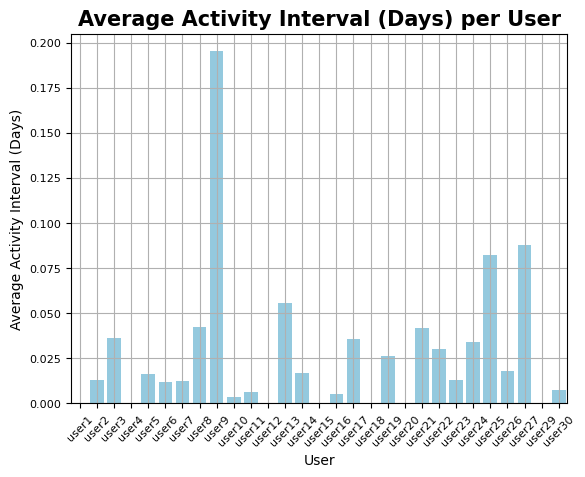
① 데이터 패턴 파악

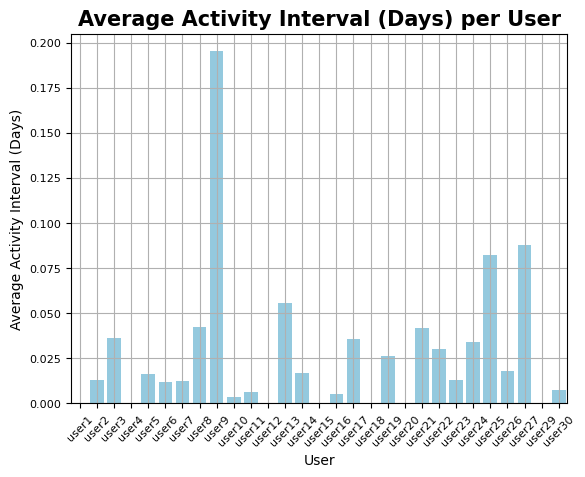
▪ 100% Scaled Bar Plot : 각 사용자별 이벤트 타입의 상대적인 참여 비율을 파악하였다.



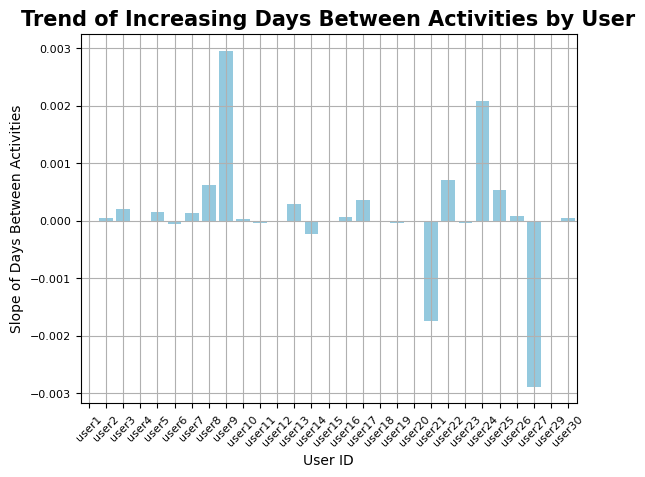
▪ Bar Plot : 사용자들 간의 활동 패턴을 확인하였다.

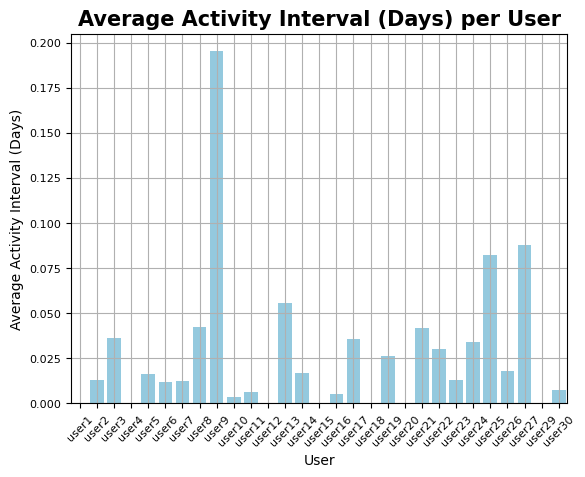
▫ 각 사용자별 평균 접속일 간격(Average Activity Interval)



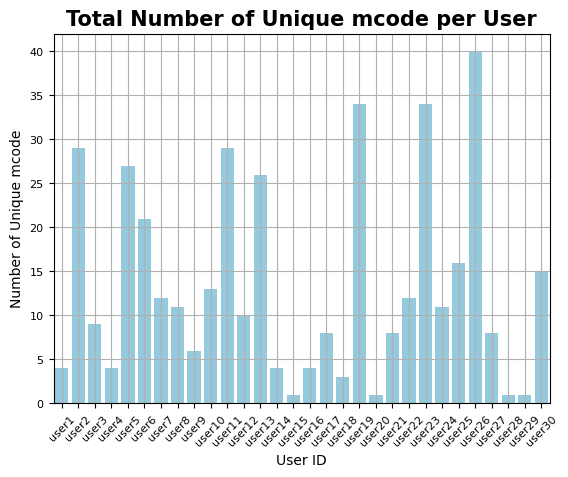


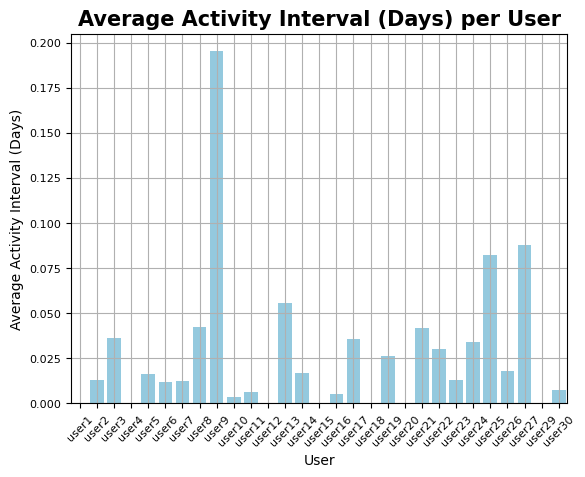
▫ 각 사용자별 접속일 간격 증감 추이(Trend of Increasing Days Between Activites)





▫ 각 사용자 별 컨텐츠 이용 개수 총 합(Total Number of Unique mcode)

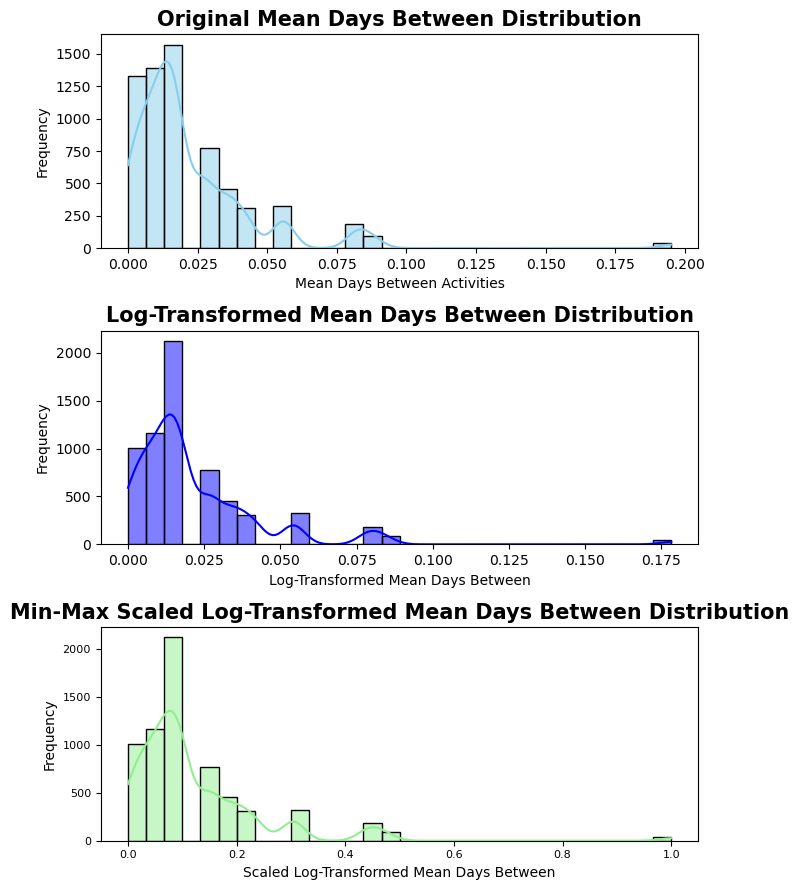




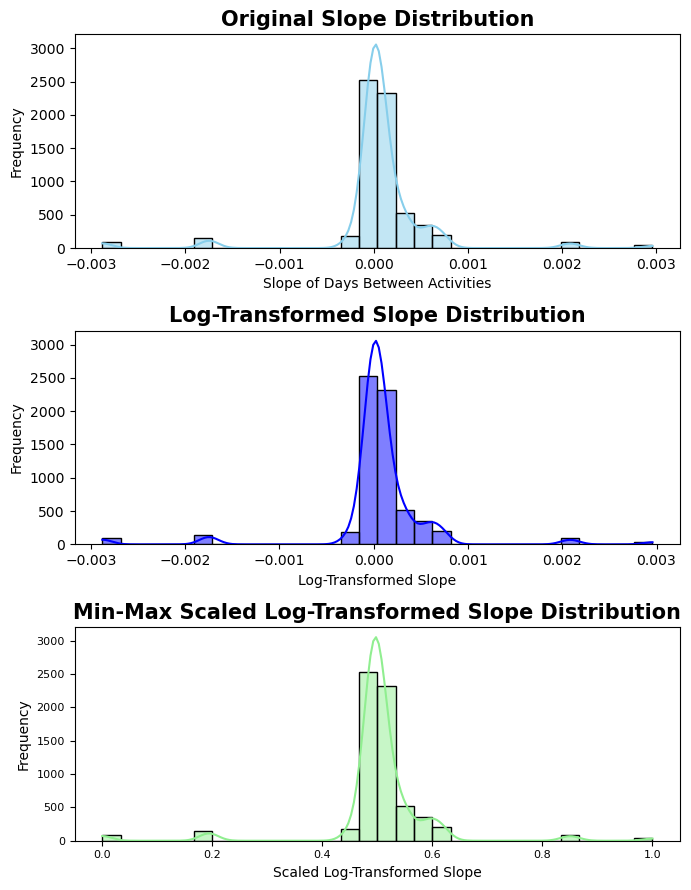
② Feature Engineering : 로그 변환 후, Min-Max Scaling(0~1)을 수행하였다.

▪ 히스토그램 : Feature Engineering이 완료된 Feature들의 분포를 확인하였다.

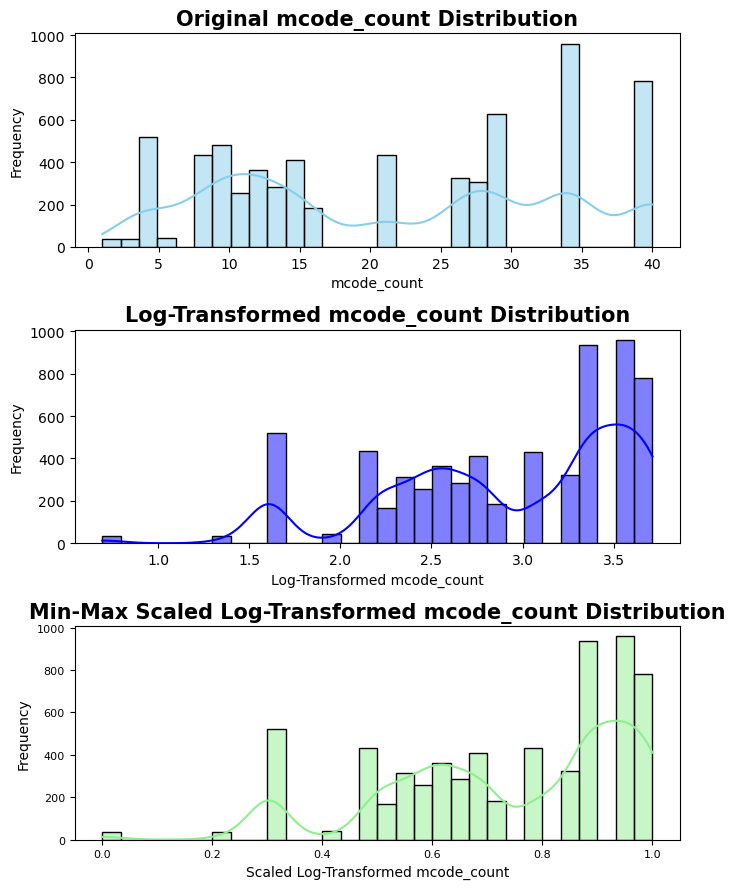
▫ Feature "days\_between\_mean"



▫ Feature "days\_between\_slope"



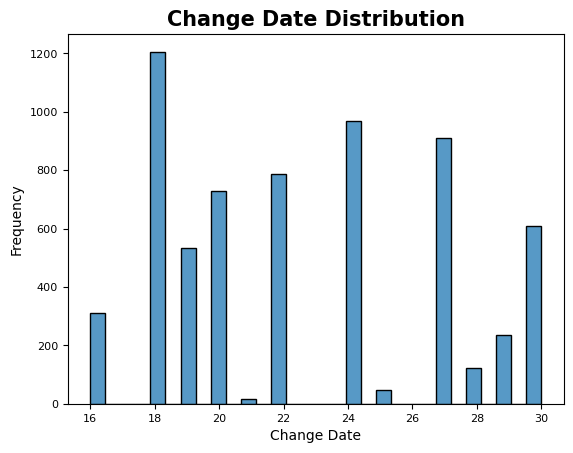
▫ Feature "mcode\_count"



**2) 모델 학습 및 성능 평가**

① 오버샘플링 : 데이터의 불균형을 해결하기 위해 다수 클래스의 수를 기준으로 소수 클래스 데이터를 오버샘플링(SMOTE; Synthetic Minority Over-sampling Technique)하였다.

▪ 히스토그램 : 오버샘플링 전, Feature "change\_date"의 클래스 별 분포를 확인하였다.



▪ 오버샘플링 전∙후의 데이터 수 변화를 확인하였다.

|  |  |  |
| --- | --- | --- |
| "change\_date"  클래스 | 데이터 수 (개) | |
| **오버샘플링 전** | **오버샘플링 후** |
| 16  기준 | 312 | 1,206 |
| 18 | 1,206 | 1,206 |
| 19 | 532 | 1,206 |
| 20 | 729 | 1,206 |
| 21 | 17 | 1,206 |
| 22 | 788 | 1,206 |
| 24 | 968 | 1,206 |
| 25 | 46 | 1,206 |
| 27 | 911 | 1,206 |
| 28 | 124 | 1,206 |
| 29 | 234 | 1,206 |
| 30 | 609 | 1,206 |

② 모델 학습

▪ 머신러닝 모델

▫ 선형 회귀 모델

▫ 랜덤 포레스트 회귀 모델

▫ 서포트 벡터 회귀 모델(SVR)

▪ Train set, Test set 분포

▫ 비율

- Train set : Test set = 8 : 2

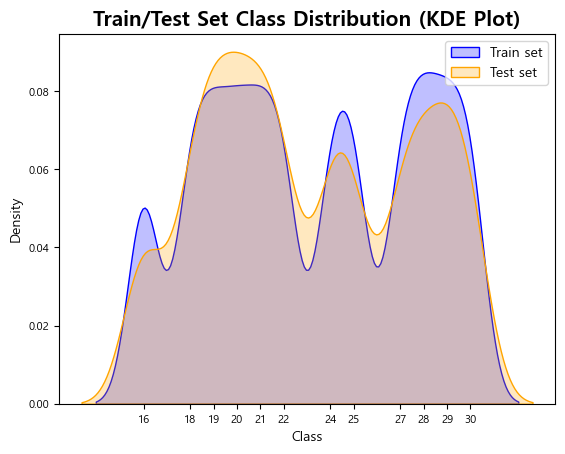
▫ 크기

- 전체 데이터 셋 크기 : 14,472 개

|  |  |  |
| --- | --- | --- |
| 클래스 | Train set (개) | Test set (개) |
| 16 | 980 | 226 |
| 18 | 968 | 238 |
| 19 | 936 | 270 |
| 20 | 944 | 262 |
| 21 | 950 | 256 |
| 22 | 952 | 254 |
| 24 | 952 | 254 |
| 25 | 987 | 219 |
| 27 | 967 | 239 |
| 28 | 1,002 | 204 |
| 29 | 962 | 244 |
| 30 | 977 | 229 |
| 총합 | **11,577** | **2,895** |

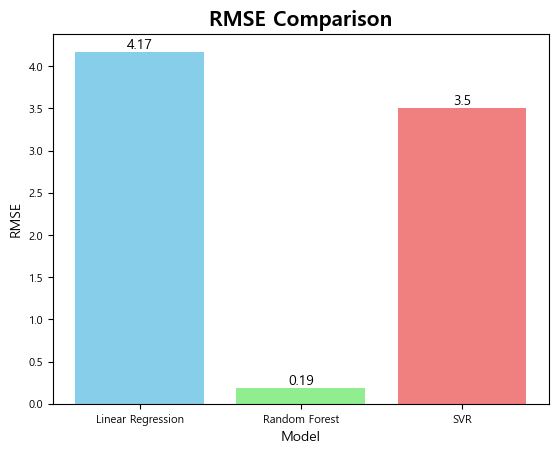
\*KDE : Kernel Density Estimation

▫ KDE\* Plot : Train set과 Test set의 클래스별 분포를 시각화였다.



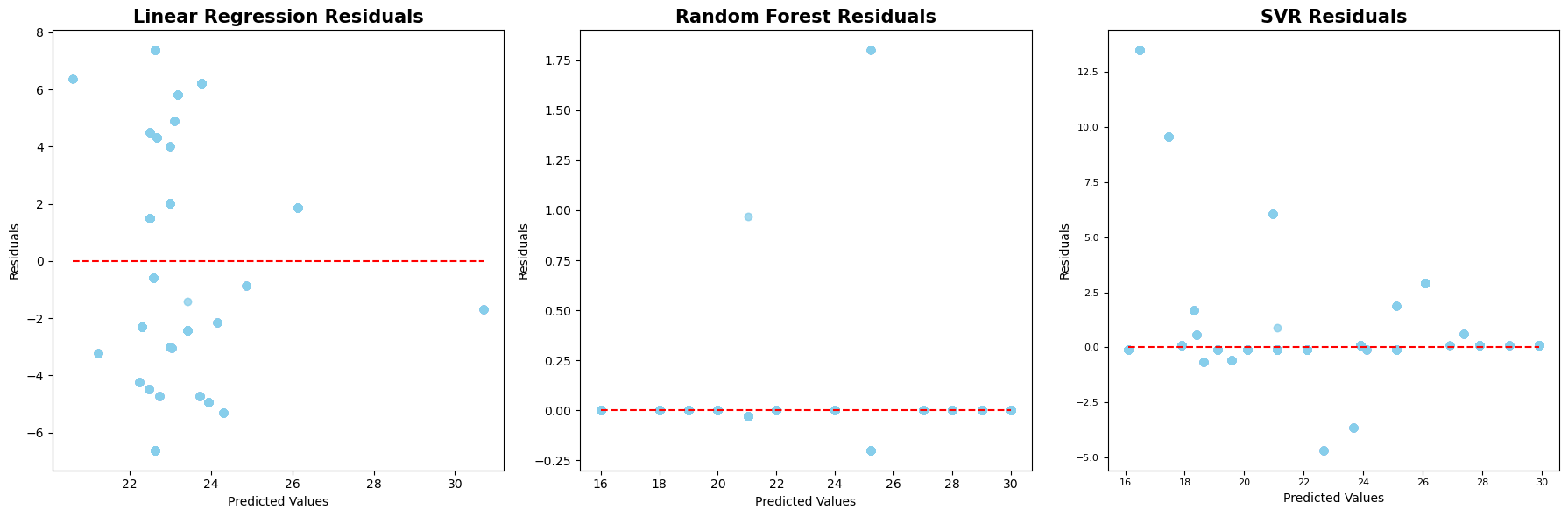
③ 모델 성능 비교

▪ Root Mean Square Error (RMSE) : 모델의 예측값과 실제값 간의 평균적 차이를 제곱근 평균으로 측정하였다.

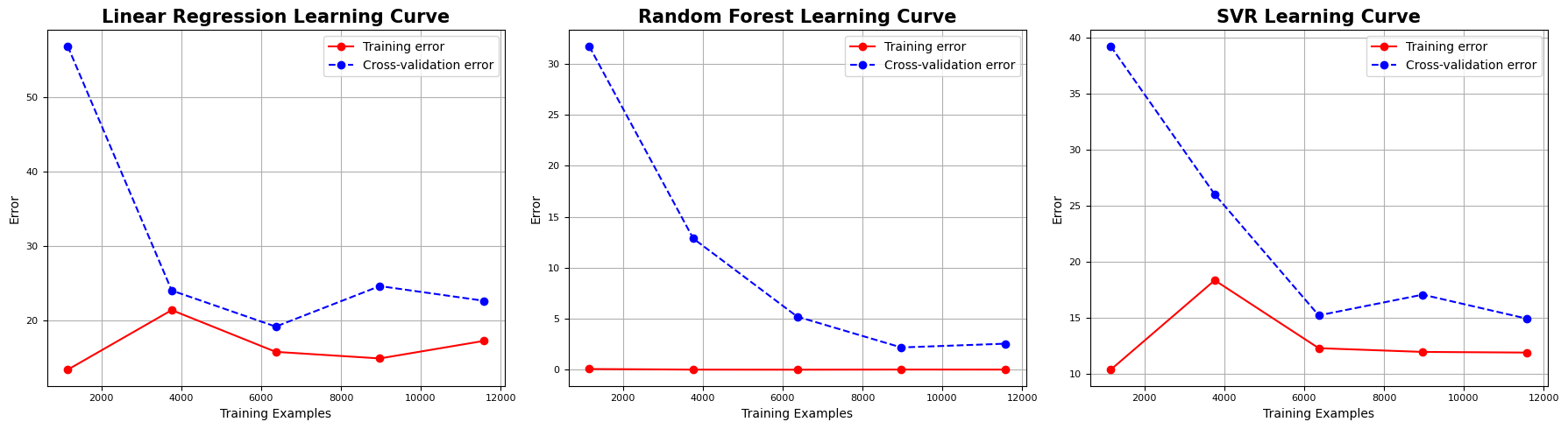


|  |  |  |  |
| --- | --- | --- | --- |
|  | 선형 회귀 모델 | 랜덤 포레스트  회귀 모델 | 서포트 벡터 회귀 모델(SVR) |
| RMSE | 4.17 | 0.19 | 3.50 |

▪ Residual Plot (회귀 잔차) : 모델의 예측 오류 패턴을 확인하였다.



▪ Learning Curve (학습 곡선) : 훈련 데이터와 검증 데이터에 대한 성능 변화를 나타내었다.

****

**7. Discussion**

본 프로젝트에서는 회원의 "만료" 상태로의 전환 시점을 예측하기 위해 세 가지 회귀 모델을 비교∙분석하였다. 랜덤 포레스트 회귀 모델이 회원의 "만료" 상태 전환 시점을 예측하는 데 가장 우수한 모델임을 확인할 수 있었다.

각 모델의 성능은 Root Mean Square Error(RMSE)로 평가되었으며, 랜덤 포레스트 회귀 모델이 가장 낮은 RMSE 값을 보여 최상의 성능을 나타내었다. 구체적으로, 랜덤 포레스트 회귀 모델의 RMSE는 0.19로, 이는 선형 회귀 모델(RMSE 4.17), 서포트 벡터 회귀 모델(SVR)(RMSE 3.50)보다 현저히 낮았다. 즉, 랜덤 포레스트 모델이 다른 모델들에 비해 예측 정확도가 우수함을 확인할 수 있었다.

이 결과를 뒷받침하는 중요한 요소는 회귀 잔차(Residual Plot)와 학습 곡선(Learning Curve)이다. 회귀 잔차 분석을 통해 랜덤 포레스트 회귀 모델이 예측에서 발생하는 오차를 최소화했음을 확인할 수 있었다. 랜덤 포레스트 회귀 모델의 잔차는 다른 모델들에 비해 균일하게 분포하며, 이는 모델이 데이터의 다양한 패턴을 잘 학습하고 있음을 보여준다. 반면, 선형 회귀 모델과 서포트 벡터 회귀 모델(SVR)은 잔차의 오차가 비교적 크게 분포하는 경향을 보여, 데이터의 복잡한 패턴을 제대로 반영하지 못했음을 나타낸다.

또한, 학습 곡선을 통해 랜덤 포레스트 회귀 모델이 훈련 데이터와 검증 데이터 모두에서 안정적인 성능을 유지하면서 과적합을 방지하고 있음을 확인하였다. 랜덤 포레스트 회귀 모델의 학습 곡선은 학습 데이터와 검증 데이터 간의 오차 차이가 가장 적으며, 학습이 진행됨에 따라 두 곡선이 수렴하는 양상을 보였다. 이는 모델이 과적합 없이 데이터의 패턴을 효과적으로 학습했다는 것을 의미한다. 반면, 선형 회귀 모델 과 서포트 벡터 회귀 모델(SVR)은 학습 곡선에서 과적합의 징후를 보이지는 않았으나, 검증 데이터에 대한 에러가 일정하지 않고, 비교적 높은 수준을 유지했다.

**8. Conclusion**

이번 프로젝트를 통해 **회원의 "만료" 상태로의 전환 시점을 예측하는 데 있어 가장 적합한 모델로 랜덤 포레스트 회귀 모델이 선정되었다.** 이 모델은 가장 낮은 RMSE를 기록하였으며, 이는 본 데이터셋의 비선형적 특성을 효과적으로 반영한 결과로 해석된다.

선형 회귀 모델과 서포트 벡터 회귀 모델은 랜덤 포레스트 모델만큼의 예측 성능을 보이지는 못했다. 이는 **단순한 선형적 접근보다는 데이터의 다양한 패턴을 반영할 수 있는 비선형 모델이 더 적합하다**는 점을 강조한다.

결과적으로, 본 프로젝트에서는 랜덤 포레스트 모델을 기반으로 한 예측 시스템을 구축함으로써, 회원이 "만료" 상태로 전환되기 전 다음과 같은 선제적 대응이 가능할 것으로 기대된다:

1) 맞춤형 리마인더 발송 : 회원의 "만료" 시점이 예측되면, 개인화된 이메일 또는 푸시 알림을 통해 구독 연장을 유도하는 리마인더를 발송한다.

2) 특별 할인 제공 : "만료" 예정 회원에게만 제공되는 특별 할인 쿠폰이나 프로모션을 제안하여 구독 연장을 유도한다.

3) 개인화된 학습 리포트 제공 : 회원이 그동안 이용한 학습 자료나 성과를 종합한 리포트를 제공하여 구독 연장의 필요성을 느끼게 한다.

4) 고객 지원 강화 : "만료" 예정 회원에게 직접 연락하여 구독 연장의 이유나 필요성을 설명하고, 고객의 질문이나 불만 사항에 즉각적인 피드백을 제공한다.

**9. 소스코드**

▪ 프로젝트의 소스코드는 별첨 자료('만료상태변환일예측\_소스코드\_240826\_신유라\_수정.ipynb')로 제공되며, 다음의 주요 섹션으로 구성되어 있다:

▫ 탐색적 데이터 분석(EDA)

▫ 데이터 전처리

▫ 모델 학습

▫ 성능 평가 및 시각화

**10. 약어**

▪ **SMOTE** : Synthetic Minority Over-sampling Technique (합성 소수 클래스 오버샘플링 기법)

▪ **EDA** : Exploratory Data Analysis (탐색적 데이터 분석)

▪ **SVR** : Support Vector Regression (서포트 벡터 회귀)

▪ **RMSE** : Root Mean Square Error (평균 제곱근 오차)

▪ **KDE** : Kernel Density Estimation (커널 밀도 추정)