IAB 실무응용 연구1 프로젝트

정기후원자의 이탈 예측모형 구현

2021. 07. 28

응용공학과 김민창

선정 사유

정기후원자를 기반으로 수입을 얻는 비영리조직의 경우
이탈 비용을 고려하여 후원자를 모집해야 순수입을 늘릴 수 있습니다.

프로젝트 주제

정기후원자의 행동 데이터를 활용한 이탈 예측모형을 구현하여
이탈 가능성이 높은 후원자에 대한 선제적인 방어 전략을 수행합니다.

구현 과정

1. Direction

2. Implementation

   a. Data Extraction

   b. Exploratory Data Analysis

   c. Model Comparison

   d. Feature Engineering

3. Conclusion

- 2021년 6월의 Snapshot 데이터로 예측 대상(Churn)을 정의하였습니다.
- Churn의 비중은 2.9%이며, Numeric data는 다양한 분포를 갖고 있습니다.
- 기계학습 Classifiers 중 Decision Tree 모델의 예측률이 가장 높았습니다.
- Outlier, 최근성(Recency)을 고려한 데이터 가공을 진행했습니다.

정기후원자 44,508 명의 데이터를 추출하였습니다

데이터 현황

- 2021년 5월 납입자: 44,508 명
    - 2021년 6월 이탈자: 1,248 명 (2.9%)
    - 2021년 6월 유지자: 43,260 명 (97.1%)

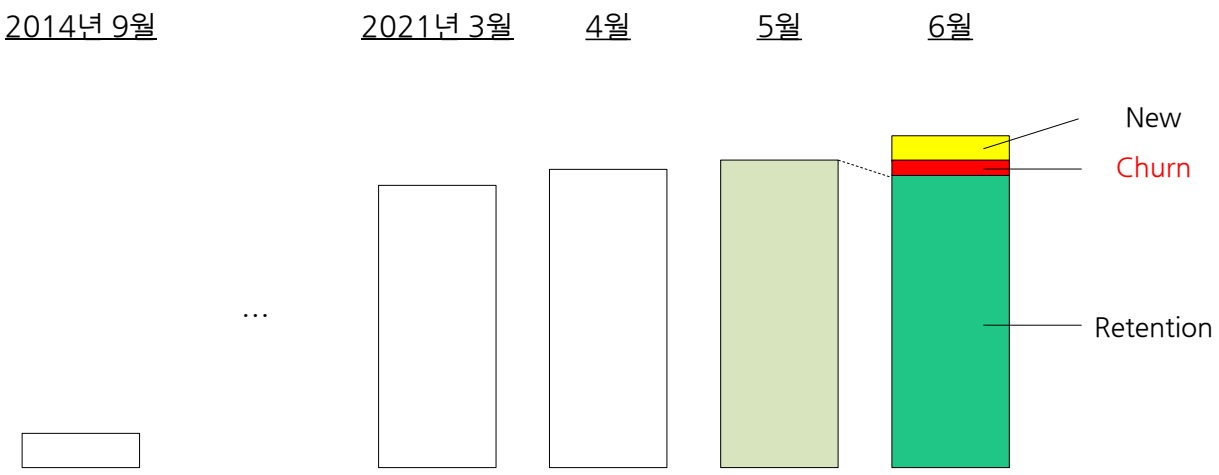
변수 목록

- 독립변수: 22개, 종속변수: 1개
- 레코드: 44,508 개

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45469 entries, 0 to 45468
Data columns (total 24 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   is_payment_2021-06                    45469 non-null  object
 1   is_payment_2021-05                    45469 non-null  object
 2   is_payment_2021-04                    44088 non-null  object
 3   is_payment_2021-03                    42598 non-null  object
 4   is_payment_2021-02                    41073 non-null  object
 5   is_payment_2021-01                    39545 non-null  object
 6   is_payment_2020-12                    37669 non-null  object
 7   is_payment_2020-11                    36144 non-null  object
 8   is_payment_2020-10                    35028 non-null  object
 9   is_payment_2020-09                    34037 non-null  object
 10  is_payment_2020-08                    32810 non-null  object
 11  is_payment_2020-07                    31584 non-null  object
 12  duration_month_until_2021-05          45469 non-null  int64
 13  total_payments                        45469 non-null  int64
 14  monthly_amount                        45469 non-null  int64
 15  total_amount                          45469 non-null  int64
 16  payment_method                        45469 non-null  object
 17  total_promises                        45469 non-null  int64
 18  total_engagements                     45469 non-null  int64
 19  total_logins                          45469 non-null  int64
 20  general_interactions_2020-07_2021-06  45469 non-null  int64
 21  is_receipt                            45469 non-null  object
 22  sex                                   45469 non-null  object
 23  age                                   44508 non-null  float64
```

* 출처: 내부자료

월 단위 예측을 목표로 대상을 정의하였습니다



예측 대상

1. 2021년 5월 납입자 중에서
2. 2021년 6월에 납입할 사람과 그렇지 않은 사람을 예측

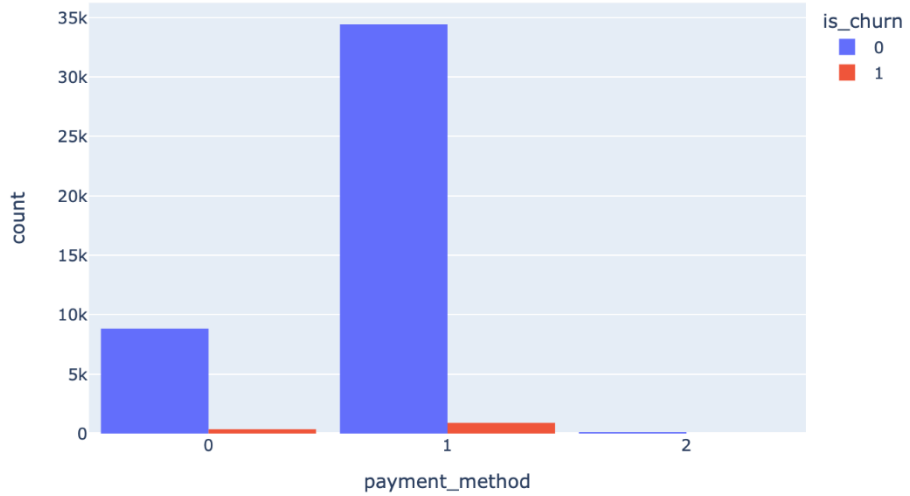
제약 사항

- 2021년 5월에 납입을 하지 않은 사람은 예측대상에서 제외됨
- 2021년 6월의 Snapshot 으로만 Training, Validation을 진행하였기 때문에 시간의 변화에 따른 정확도를 보장할 수 없음

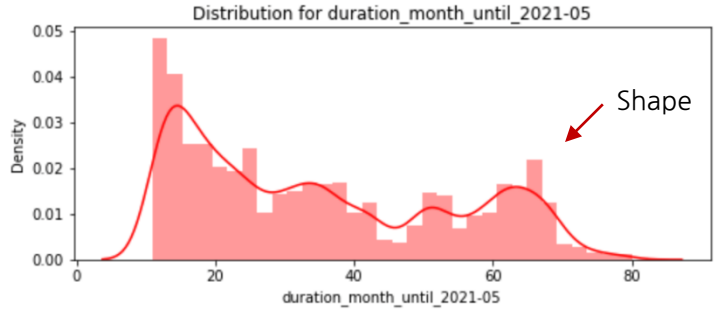
Churn의 비중은 2.9%로 낮으며, Numeric data는 다양한 분포를 갖고 있습니다

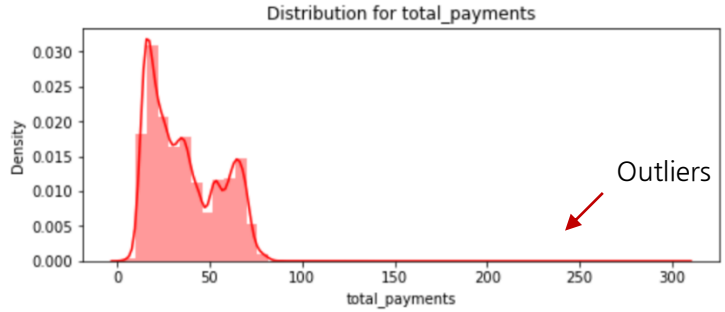
결제수단별 이탈여부

0: card, 1: cms, 3: 수동이체



후원기간, 총납입횟수



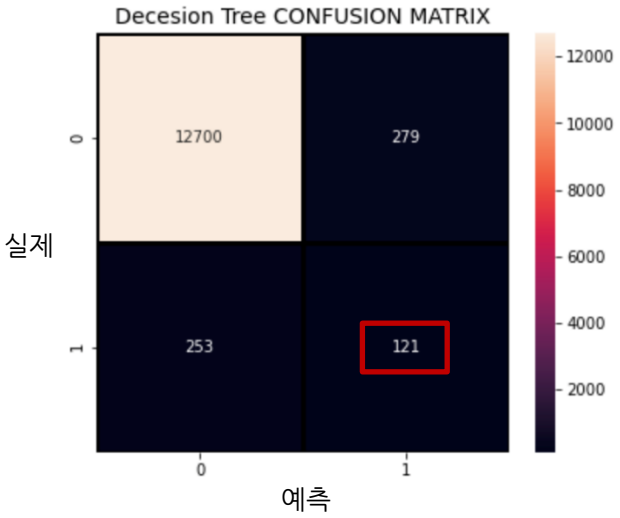


Decision Tree Classifier의 Recall이 가장 높았습니다

Decision Tree (Best)

- Recall: 0.32
- F1-score: 0.31

	precision	recall	f1-score	support
0	0.98	0.98	0.98	12979
1	0.30	0.32	0.31	374
accuracy			0.96	13353
macro avg	0.64	0.65	0.65	13353
weighted avg	0.96	0.96	0.96	13353



KNN

- Recall: 0.01
- F1-score: 0.01

Logistic Regression

- Recall: 0.05
- F1-score: 0.09

SVM

- Recall: 0.00
- F1-score: 0.00

Ada Boost

- Recall: 0.05
- F1-score: 0.08

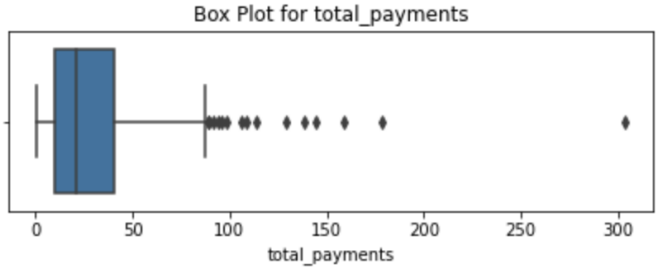
Random Forest

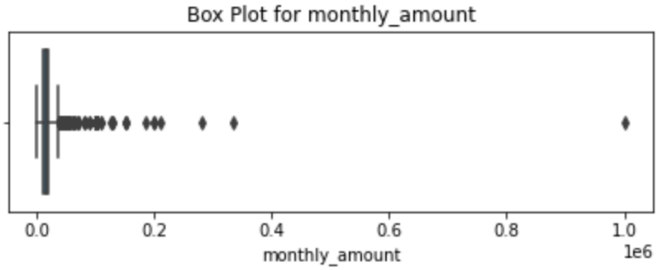
- Recall: 0.16
- F1-score: 0.15

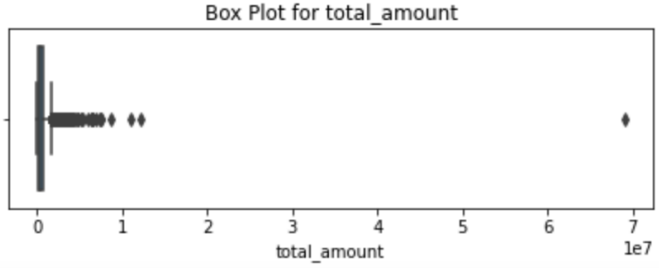
Gradient Boosting

- Recall: 0.16
- F1-score: 0.25

Z-score Method를 활용해 Outlier data를 통폐합하였습니다

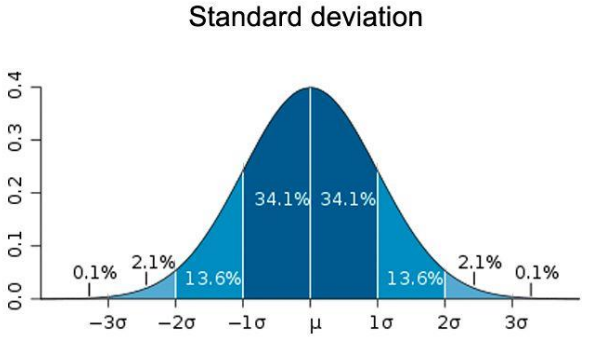







```
In [26]: df.groupby('monthly_amount').size()

Out[26]: monthly_amount
         500          2
         1000         3
         2000         4
         3000        12
         4000         3
                     ..
         200000       2
         210000       1
         280000       1
         333000       1
         1000000      1
         Length: 69, dtype: int64
```



최근성(Recency)을 고려해 새로운 Feature를 가공하였습니다

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45469 entries, 0 to 45468
Data columns (total 24 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   is_payment_2021-06                    45469 non-null  object
 1   is_payment_2021-05                    45469 non-null  object
 2   is_payment_2021-04                    44088 non-null  object
 3   is_payment_2021-03                    42598 non-null  object
 4   is_payment_2021-02                    41073 non-null  object
 5   is_payment_2021-01                    39545 non-null  object
 6   is_payment_2020-12                    37669 non-null  object
 7   is_payment_2020-11                    36144 non-null  object
 8   is_payment_2020-10                    35028 non-null  object
 9   is_payment_2020-09                    34037 non-null  object
 10  is_payment_2020-08                    32810 non-null  object
 11  is_payment_2020-07                    31584 non-null  object
 12  duration_month_until_2021-05          45469 non-null  int64
 13  total_payments                        45469 non-null  int64
 14  monthly_amount                        45469 non-null  int64
 15  total_amount                          45469 non-null  int64
 16  payment_method                        45469 non-null  object
 17  total_promises                        45469 non-null  int64
 18  total_engagements                     45469 non-null  int64
 19  total_logins                          45469 non-null  int64
 20  general_interactions_2020-07_2021-06  45469 non-null  int64
 21  is_receipt                            45469 non-null  object
 22  sex                                   45469 non-null  object
 23  age                                   44508 non-null  float64
dtypes: float64(1), int64(8), object(15)
memory usage: 8.3+ MB
```

예. 단순한 One hot encoding 방식 대신 최근 11개월 동안의 납입률로 가공하여 사용

- 최근 납입률 = 납입한 개월 수 / 11개월

```
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   is_payment_2021-06              44508 non-null  int64
 1   full_payment_rate_11months      44508 non-null  float64
 2   duration_month_until_2021-05    44508 non-null  int64
 3   total_payments                  44508 non-null  int64
 4   monthly_amount                  44508 non-null  int64
 5   total_amount                    44508 non-null  int64
 6   payment_method                  44508 non-null  int64
 7   total_promises                  44508 non-null  int64
 8   total_engagements               44508 non-null  int64
 9   total_logins                    44508 non-null  int64
 10  is_receipt                      44508 non-null  int64
 11  sex                             44508 non-null  int64
 12  age                             44508 non-null  int64
dtypes: float64(1), int64(12)
memory usage: 4.8 MB
```

Feature Engineering을 통해 Decision Tree 방식의 Recall이 0.63으로 증가했습니다

Decision Tree

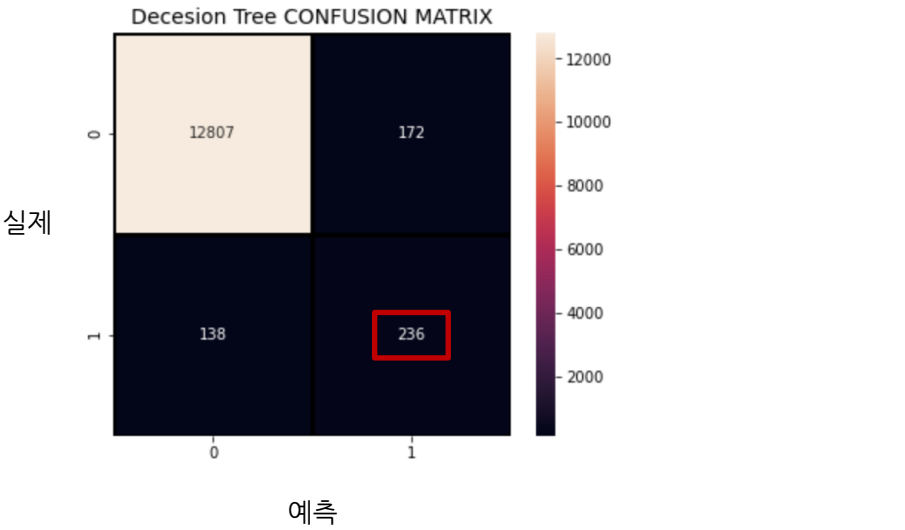
- Recall(0.32 → 0.63)
- F1-score(0.31 → 0.60)

```
In [36]: print(classification_report(y_test, predictdt_y))
```

	precision	recall	f1-score	support
0	0.99	0.99	0.99	12979
1	0.58	0.63	0.60	374
accuracy			0.98	13353
macro avg	0.78	0.81	0.80	13353
weighted avg	0.98	0.98	0.98	13353

```
In [37]: plt.figure(figsize=(6,5))
sns.heatmap(confusion_matrix(y_test, predictdt_y),
                annot=True, fmt = "d",linecolor="k",linewidths=3)

plt.title("Decesion Tree CONFUSION MATRIX",fontsize=14)
plt.show()
```



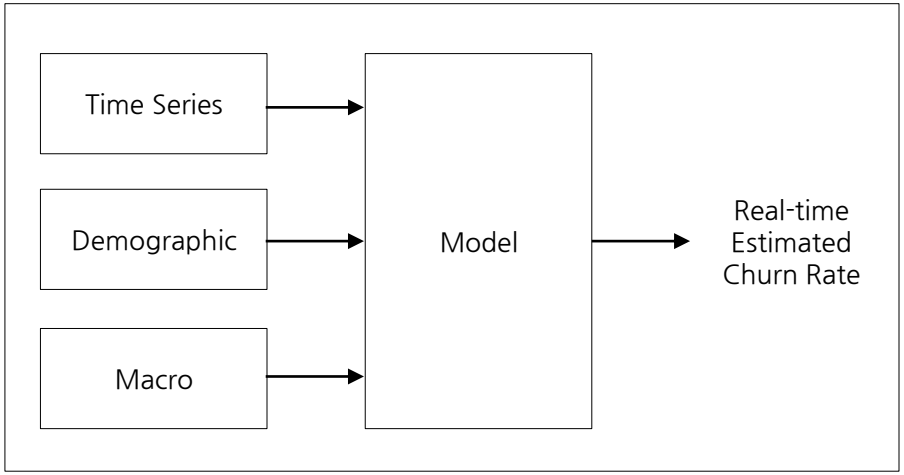
향후 과제

1. 시간변수 고려
   - 시간의 변화에 따라 예측 정확도가 달라지는 문제가 발생할 수 있습니다 (예. 월별 이탈률이 다른 경우). 시간변수를 고려한 모델이 필요합니다.

2. 외부요인 고려
   - 자체 데이터 만으로는 사건사고 등 외부 요인의 영향력을 반영하기가 어렵습니다. 평시가 아닌 긴급상황에서도 Robust 한 예측률을 보여주는 모델이 필요합니다.

3. 예측속도 고려
   - 후원자 행동 데이터는 월 단위가 아닌 수시로 수집됩니다. 월 단위 '예측'을 넘어 실시간으로 '반응' 하는 모델이 필요합니다.

예측모델 고도화



- Time Series: Payments, Interactions
- Demographic: Age, Sex
- Macro: GDP, Income, Unemployment Rate

감사합니다

Thank You