세종대학교

모바일 게임 로그 데이터를 이용한 고객 LTV 예측

팀이름 | User 진심 알아주슈

당담교수 | 김미숙

학과 | 데이터사이언스

팀원 | 심소민 17011720

윤영주 17011718

이유진 17011730

진현영 18011724

최시우 17011724

목 차

1 서론

1.1 프로젝트 목적 및 필요성

많은 기업이 고객의 선호 경향과 행동을 이해하는 쪽으로 변함으로써 맞춤형 마케팅으로 변하고 있고, 동일 업계의 기업 간 경쟁은 더욱 치열해지고 있다. 이러한 상황 속에서, 고객의 특성을 파악하고 고객의 가치를 측정하여 적절한 마케팅 전략을 수립하는 것은 고객가치를 상승시키고 지속적 경쟁 우위를 유지하는 데 있어서 매우 중요하다. 여기서 고객 가치는 고객이 기업에 가져다주는 이익과 그 고객이 기업의 상품을 사용하는 시간의 함수로 정의할 수 있다. 이러한 관점에서 고객의 가치를 측정하는 모형을 고객 생애 가치(Customer Lifetime Value; LTV) 모형이라고 한다.

본 프로젝트에 데이터를 제공한 게임업체 'ALOHA factory' 에서는 LTV 를 고객 개별적인 특성을 반영하지 않고, 일반화된 통계 모형으로 계산하였다. 이는 LTV 를 간단하게 계산할 수 있지만, 고객 행동의 변동 및 현재 가치 등과 같은 개별적 특성을 고려하지 못하여 오차가 발생한다는 단점이 있다. 따라서 이러한 단점을 보완한 LTV 모형을 통해 더욱 정확한 고객 가치를 계산하고, LTV 값의 패턴을 통한 더 세밀한 고객 세분화가 필요하다. 이에 본 프로젝트는 머신러닝 기법을 활용한 LTV 를 예측 모델을 제시하고, 고객 세분화를 기반으로 효율적인 마케팅 전략을 제시하고자 한다.

게임업체 'ALOHA factory' 의 'draw-hammer' 게임의 경우, 인앱 결제가 없고 오로지 광고로 수익을 창출하는 구조이며, 평균적으로 한 user 가 게임 설치 후 7 일이 손익분기점이다. 이에 본 프로젝트는 업체에서 받은 'draw-hammer' 게임 로그 데이터를 이용하여 각 사용자의 7 일의 광고 노출 수를 예측하는 것을 목표로 설정하였다.

1.2 프로젝트 수행 일정

No	일시	학습목표 및 내용	실천과제
1	3/5-7	Customer Lifetime Value Study & project plan	관련 서적 & 논문 리딩
2	3/8-14	project plan & data preprocessing	관련 서적 & 논문 리딩
3	3/15-21	data preprocessing	데이터 분석에 맞게 형태 변환
4	3/22-28	data preprocessing & EDA	EDA를 통한 게임 분석 및 이해
5	3/29-4/4	EDA & target 선정	EDA를 통한 게임 분석 및 이해
6	4/5-11	feature engineering & extraction	feature 정리 및 추출
7	4/12-18	feature extraction - correlation	상관분석을 통한 feature 선정
8	4/19-23	중간고사	.
9	4/24-30	outlier 제거 & regression modeling	feature별 outlier 제거 및 impression 예측
10	5/1-6	feature selection & classification modeling	day 5,6,7 잔존 여부 예측
11	5/7-14	model development	예측 모델 개선 및 성능 분석
12	5/15-22	brazil data data preprocessing & feature extraction	brazil data - 데이터 가공, 전처리 및 피쳐 선정
13	5/23-30	brazil data modeling & model evaluation	brazil data - impression 예측 및 성능 개선
14	5/31-6/4	user segment & marketing strategy	brazil data - user segment 별 특징 분석 및 마케팅 전략 구축
15	06/5-11	project finish	최종 보고서 작성 및 제출

2 본론

2.1 데이터 설명

게임업체 'ALOHA factory'로부터 두 차례 데이터를 제공받았으며, 각각 USA 와 Brazil user log data 이다. 아래는 각 데이터의 공통된 특징을 설명한다.

- 파일 형식: JSON
- 전처리 nested (JSON) to flatten(CSV) :
- target user
  프로젝트의 목적에 맞게 day 7 자체가 없거나 신규 가입일을 알 수 없는 user 는 분석 대상에서 제외하였다.
  - USA : 2021.02.18 ~ 03.02 (약 2 주) (25,237 명)
  - Brazil : 2021.04.01 ~ 05.8 (약 5 주) (241,130 명)

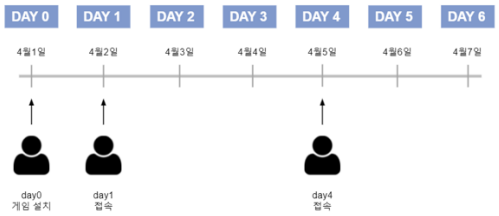
총 21 개 event 중 firebase 에서 자동 생성되는 것을 제외한 10 개의 event 를 채택하였다.

	event_name		first_open
1	event_name	12	first_open
2	draw_start	13	cross
3	play_end	14	activity
4	ads	15	tutorial
5	stage_start	16	item
6	screen_view	17	session_start
7	user_engagement	18	app_remove
8	stage_end	19	firebase_campaign
9	draw_end	20	app_clear_data
10	play_start	21	os_update
11	asset	22	app_update

<그림 1> event 종류

'day' 정의

사용자가 게임 설치일로부터 흐른 날을 의미한다. 예를 들어 한 사용자가 4 월 1 일에 신규 가입을 했다면 그날은 day0, 4 일 뒤 재접속을 했다면 그날은 day 4 가 된다.



<그림 2> day 정의 예시

Stickiness

Stickiness 는 월간 활성 사용자 대비 일간 활성 사용자의 비율 (DAU/MAU)로 계산하며 월간 순수 사용자가 어제 얼마나 접속했는지 알 수 있다. Stickiness 가 50%라는 것은 앱의 한 달 사용자 중 반이 어제 재접속했다는 것을 의미한다. Draw hammer 의 경우 받은 데이터 기간 내 stickiness 가 2% 채 되지 않는다. 이것은 꾸준히 접속하는 유저가 많지 않음을 의미한다.

the_month_utc	the_day_utc	DAU	MAU	stickiness
2	1	23	53291	0.04315925766076824
2	2	60	53291	0.11258936781069974
2	3	751	53291	1.4092435870972584
2	4	953	53291	1.7882944587266143
2	5	1009	53291	1.8933778686832672
2	6	1360	53291	2.552025670375861
2	7	985	53291	1.8483421215589875
2	8	457	53291	0.8575556848248298

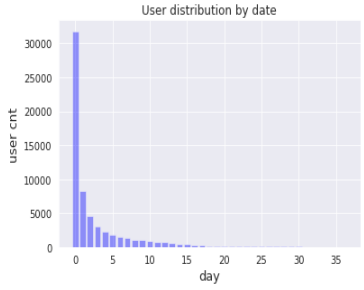
<그림 3> USA dada stickiness 계산 결과

2.2 분석 절차

2.2.1 EDA (Exploratory Data Analysis)

[USA]

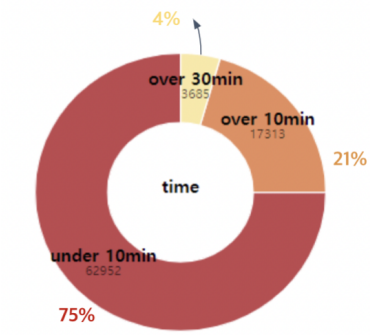
<그림 4> 와 같이 day 에 따른 사용자 분포를 봤을 때, day0 과 비해 day1 에서 절반 이상 사용자가 줄어들었다. 또한, 전체적으로 접속자가 줄어들지만, day3 부터는 줄어드는 정도가 완만해지는 것을 알 수 있다. 이를 통해 초기 day0, day1 에서 사용자 확보가 중요하고, 며칠 뒤의 사용자 수가 초기에 결정 날 가능성이 높아 보인다.



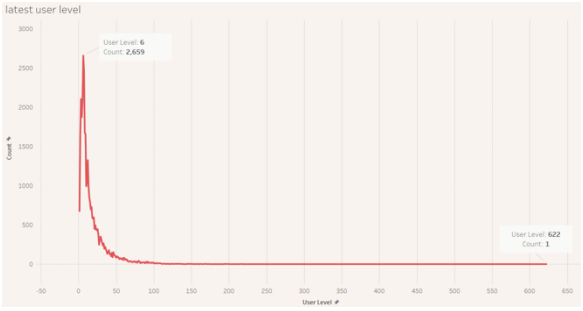
<그림 4> day에 따른 user 분포

전체 user 의 session 당 접속 시간을 초 단위로 계산한 후 분포를 살펴보았을 때 <그림 5> 와 같은 결과를 보였다. 한번 접속했을 때 체류 시간은 10 분 미만으로 접속해있는 세션의 비율이 전체의 75%를 차지했으며 10 분 이상 30 분 미만으로 접속해 있는 세션의 비율은 21%를 차지했다. 30 분 이상 머무른 세션의 비율은 4%로 가장 낮았고 1 시간을 넘어가는 세션은 없었다.

User 별 가장 최근에 도달한 level 을 시각화해 본 결과 <그림 6> 과 같은 결과를 보였다. level6 에 user 가 가장 많이 분포해 있으며, 가장 높은 level 은 622 였다. 대부분 낮은 레벨에 분포해 있음을 알 수 있었다.



<그림 5> session당 접속 시간 비율



<그림 6> user별 가장 최근 level 분포

Target variable

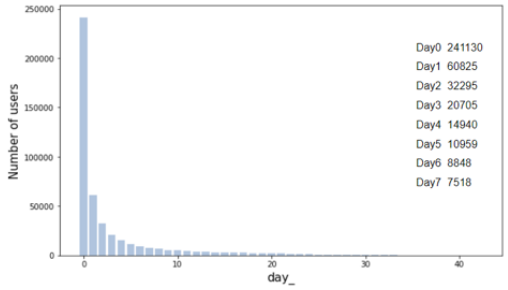
day 7 로 광고 노출 수를 종속 변수로 설정했을 때, 예측 성능이 좋지 않았다.

매일 연속적으로 접속하는 유저가 아닌 일정 간격을 두고 접속하는 유저의 경우 day7 에 접속하지 않았을 때 배제될 수 있다. 예를 들어 day7 전후 날에 접속하여 광고를 시청하였지만 day7 에는 접속하지 않으면 종속변수 값은 0 이기 때문에 이를 예측하기는 어렵다. 이때 발생하는 오차를 줄이기 위해 종속 변수를 day5,6,7 의 광고 노출 수의 합으로 변경하였다. 그 결과 종속변수의 값이 0 이 아닌 사용자가 약 1,800 명으로 3 배 증가하였다.

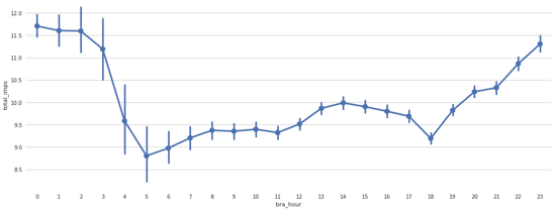
[Brazil]

<그림 7> Brazil 도 USA 와 같이 day0 에 비해 day1 에서 절반 이상 사용자가 줄어드는 것을 확인할 수 있다. <그림 8> 은 시간의 흐름에 따라 광고 노출 수 변화를 나타낸 것이다. 사람들이 활동을 시작하는 오전 6 시부터 자정까지 점차 증가하고 이후에는 줄어드는 일반적인 생활 패턴이 보인다.

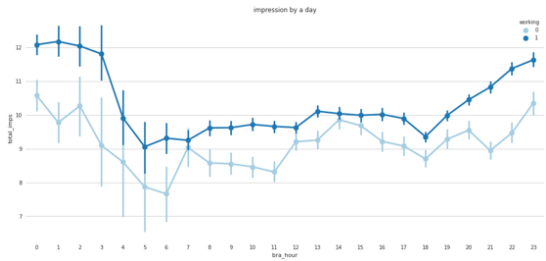
<그림 9> 는 평일과 휴일을 나누어 본 그래프로, 휴일에 더 많을 것으로 예상한 것과는 다르게, 이 게임 사용자들은 평일에 더 많은 광고 노출 수가 나타났다. <그림 10> 는 요일별, 시간대별 접속 횟수를 히트맵으로 나타낸 결과이다. 진한 빨간색일수록 활성 시간대, 진한 초록색일수록 비활성 시간대이다. 요일에 따라 저녁 시간대의 이용률이 차이가 나는 것이 눈에 띈다. 이 히트맵에서 알 수 있듯이 오전보다 오후 시간대에 게임을 하는 횟수가 더 많다. 이를 이용하여 요일 및 시간대 별 광고 노출 수를 조정하는 전략적인 방식으로 접근하면 좋을 것으로 보인다.



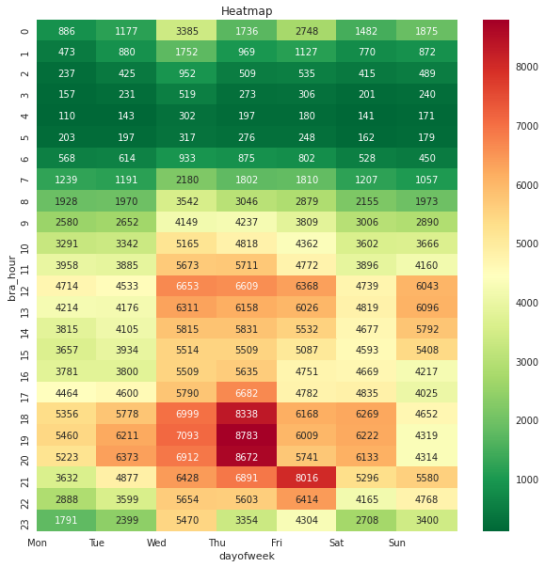
<그림 7> day에 따른 사용자 분포



<그림 8> 일일 광고 노출 수



<그림 9> 일일 광고 노출 수 by 평일, 휴일



<그림 10> 요일/시간 별 접속 횟수

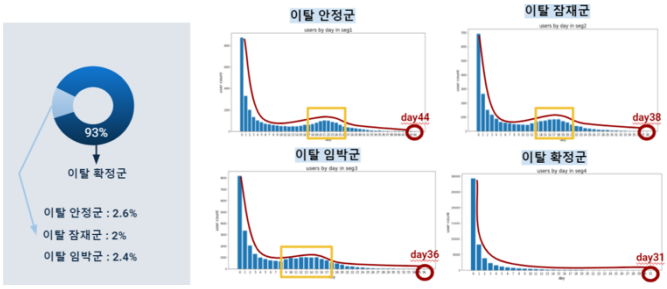
이탈 위험 비율 (risk ratio)

개별 유저의 특성을 무시하고 하나의 기준을 만드는 데 그치는 문제점을 해결하기 위해, 사용자마다 접속 주기가 다를 수 있으므로 이를 고려한 이탈 기준을 찾는다면 더 나은 사용자 관리가 가능하다. 이용자가 평균 접속 주기보다 긴 기간 접속하지 않을 시 이탈하였을 가능성이 크다. 따라서 이탈 위험 비율을 최종 접속 경과일 / 평균 접속 주기로 측정하였고 그에 따른 유저 구분 기준은 <그림 11>과 같다.

이탈 확정 군이 전체 유저의 93%로 가장 높은 비율을 차지하고, 이탈 위험 비율이 높아질수록 게임을 지속하는 기간이 짧아지는 것을 확인할 수 있다. 또한, 이탈 확정 군만 유일하게 day0 이후로 사용자 수의 상승 변화 없이 지속해서 줄어드는 경향이 있다.

구분	측정식
이탈 안정군	0.0<Risk Ratio≤1.0
이탈 잠재군	1.0<Risk Ratio≤2.0
이탈 임박군	2.0<Risk Ratio≤3.5
이탈 확정군	Risk Ratio>3.5

<그림 11> 이탈 위험 비율 구분 기준



<그림 12> 이탈 위험 비율 군집 별 day 분포

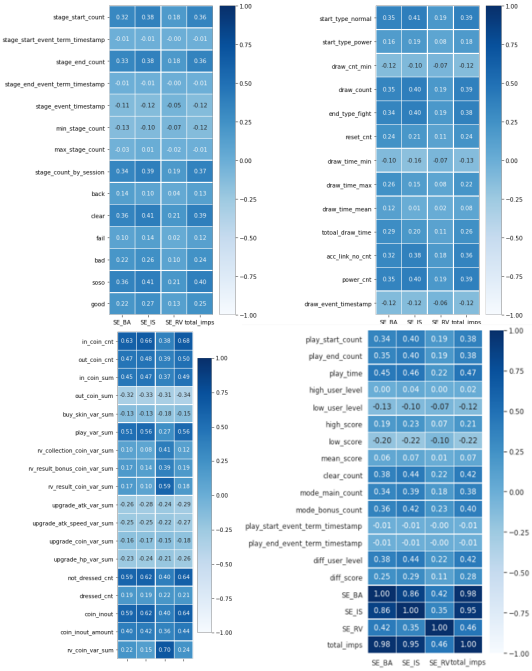
2.2.2 Feature Engineering / Extraction

<표 1> 과 같이 앱 설치 후 지속적으로 앱을 활발하게 사용하는 유저를 구별할 수 있을 것이라 생각되는 feature 와 인앱 광고와 관련된 feature 를 선정해 추출했다. 그 후 <그림 13>과 같이 상관분석을 통해 너무 낮은 상관관계를 보이는 feature 들은 1 차적으로 제거했다.

모델을 훈련시키기 위해 ad_id 별 기본 정보, 일별 feature, 그리고 종속 변수 day 5,6,7 의 impression <그림 14>과 같이 구조를 변경하였다.

공통	- 접속 시간 - 접속 횟수 - 세션 평균 체류 시간
play_start / play_end	- play end 횟수 - play 한 시간 - user min level, high level
activity	- activity한 횟수 - activity_id 별 누른 횟수 - action_id별 누른 횟수
ads	- placement-id
asset	- 실제 재화 변동량 합계 - 코인 지급 in, out count
stage_start / stage_end	- 실제 stage 횟수 - is_pinch, is_perfect를 이용한 게임 수행 능력 점수 환산
draw_start / draw_end	- draw start type 별 횟수 - draw 시행 횟수 - power_id 시행 횟수 - reset 횟수 - draw 소요 시간 - draw end type 횟수 - color_id 변경 횟수 - is_power_success 횟수 - acc_link_no 횟수

<표 1> 각 event 별 추출된 feature



<그림 13> 각 event에서 impression의 상관관계 히트맵

<그림 14> model 적용 data 구조

2.2.3 Preprocessing

missing values

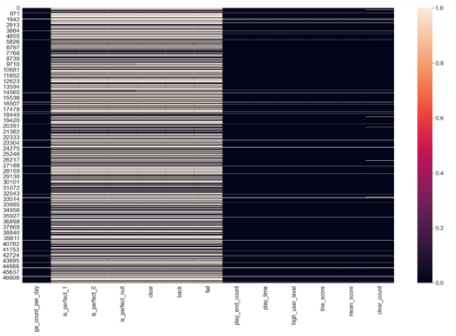
train dataset 생성 전, 전체 데이터(71 columns)의 결측값을 시각화 한 결과, 열과 행 각각의 기준으로 확인하였을 때, 결측값이 대부분인 feature 가 다수 존재했다.

일부 피처를 시각화 한 결과 <그림 15-2>에서 흰 띠와 같은 부분은 해당 column 들이 모두 결측치라는 의미이다. 따라서 유저의 기본 정보와 같은 column 을 제외한 모든 칼럼이 연속적으로 결측치를 갖는 데이터는 제거하였다. 또한 각 피처 별 box plot 을 확인하여 노이즈 데이터도 제거하였다.

그 결과 class 변수와의 상관계수가 소폭 상승하였고, stepwise method 로 선택된 피처가 일부 변경되었음을 <그림 16>에서 확인할 수 있다. 또한, 로지스틱 회귀 모델의 재현율이 향상되는 양상을 보였다.

```
20  stage_count_per_day      46646 non-null
21  is_perfect_1             22435 non-null
22  is_perfect_0             22435 non-null
23  is_perfect_null          22435 non-null
24  clear                    22342 non-null
25  back                     22342 non-null
26  fail                     22342 non-null
27  play_end_count           46208 non-null
28  play_time                46208 non-null
29  high_user_level          46208 non-null
30  low_score                46208 non-null
31  mean_score               46208 non-null
32  clear_count              46001 non-null
```

<그림 15-1> column 20~32



<그림 15-2> column 20~32의 결측치

결측치 처리 전

```
class                      1.000000
usage_time_sec             0.122843
in_coin_cnt                0.122718
rv_result_bonus_coin       0.122526
clear_count                0.121245
rv_result_coin             0.120851
stage_count_per_day        0.119636
play_start_count           0.119284
type_fight                 0.119188
play_end_count             0.119132
draw_cnt                   0.118749
DAY_RV                     0.117938
reset                      0.117886
type_normal                0.117640
draw_count                 0.116860
DAY_BA                     0.116256
play_time                  0.116205
rv_color                   0.116078
acc_link_no                0.114744
ba_ingame                  0.110676
is_game_end                0.110353
DAY_IS                     0.109714
play_var_sum               0.109151
ba_home                    0.107094
tot_dt                     0.105793
day1_visit                 0.103982
out_coin_cnt               0.093563
ba_result                  0.081753
power_id                   0.077674
type_power                 0.077146
is_power_success           0.075712
rv_up_1                    0.073204
rv_up_2                    0.069704
open_col                   0.068767
rv_up_3                    0.068390
rv_collection_coin         0.055215
rv_up_4                    0.049453
rv_result_bonus_coin_var_sum  0.027433
upgrade_coin_var_sum      -0.056209
upgrade_hp_var_sum        -0.071353
upgrade_atk_speed_var_sum -0.074306
upgrade_atk_var_sum       -0.083444
Name: class, dtype: float64
```

결측치 처리 후

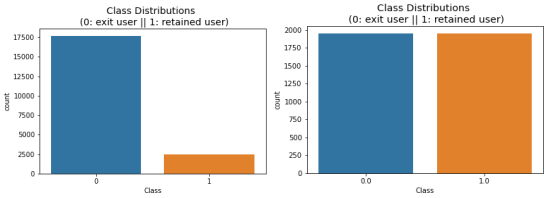
```
class                      1.000000
in_coin_cnt                0.124613
rv_result_bonus_coin       0.124451
usage_time_sec             0.123696
clear_count                0.123115
rv_result_coin             0.122687
stage_count_per_day        0.121498
play_start_count           0.121097
type_fight                 0.121046
play_end_count             0.120942
draw_cnt                   0.120560
type_normal                0.119776
reset                      0.119761
DAY_RV                     0.119699
draw_count                 0.118737
rv_color                   0.118087
play_time                  0.117811
DAY_BA                     0.117574
acc_link_no                0.116676
is_game_end                0.112555
ba_ingame                  0.112098
DAY_IS                     0.111928
play_var_sum               0.110991
ba_home                    0.108235
tot_dt                     0.106909
day1_visit                 0.104050
ba_result                  0.082646
power_id                   0.077190
type_power                 0.076587
is_power_success           0.075359
rv_up_1                    0.074551
rv_up_2                    0.070258
rv_up_3                    0.067618
rv_collection_coin         0.056293
rv_up_4                    0.049494
Name: class, dtype: float64
```

<그림 16> 결측 값 처리 후 상관계수 변화

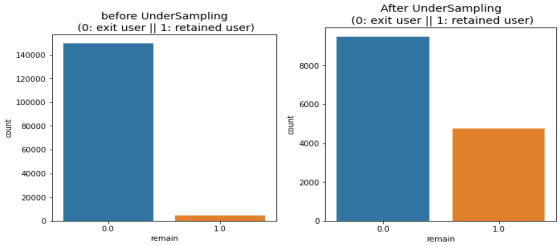
Under sampling

예측 모델을 구축하기 위해서는 충분한 학습 데이터를 확보하여야 한다. 그러나 제공받은 데이터에서 day 5,6,7 의 광고 노출 수가 0 인 고객과 0 이상의 값을 가지는 고객의 분포가 불균형하다. 이러한 불균형 데이터로 모델을 학습시킨다면 모든 값을 0 으로 예측하는 문제가 발생한다. 따라서 불균형 문제를 해결하여 광고 노출 수를 제대로 예측할 수 있도록 해야 한다.

데이터 불균형을 해결하는 방법으로는 크게 Oversampling 과 Under sampling 이 있다. Oversampling 의 경우 정보가 손실되지 않는다는 장점이 있지만 새로 추가된 데이터들 때문에 overfitting 이 될 우려가 있다. 반면에 Under sampling 은 정상 데이터의 손실이 커질 수 있다는 단점이 있다. 본 프로젝트에서는 대용량 데이터 분석에 메모리가 충분하지 않기 때문에 USA data 와 Brazil data 는 랜덤 언더샘플링을 진행하였다. USA data 는 <그림 17>와 같이 1:1 로, Brazil data 는 <그림 18>과 같이 day 5,6,7 의 광고 노출 수가 0 인 user 가 0 이 아닌 user 의 약 14 배로 비정상적으로 불균형하기 때문에 2:1 로 언더 샘플링하여 정보의 손실을 줄여 불균형 데이터를 보정하였다.



<그림 17> USA data Under sampling적용 전, 후



<그림 18> Brazil data Under sampling적용 전, 후

feature scaling

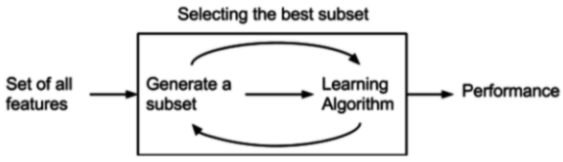
독립 변수와 종속 변수를 히스토그램으로 시각화하고 왜도(skew)를 확인한 다음 왜도가 큰 변수에 대해 로그 변환을 적용하였다. 이렇게 피처를 변형할 경우 데이터가 너무 많거나 너무 적어 모델에 부정적인 영향을 줄 수 있는 피처를 안정적으로 변환하고 성능 향상에 도움을 줄 수 있다. 그리고 회귀 분석 후에는 예측값에 역으로 지수 처리하여 RMSE 를 확인하였다.

2.2.4 Modeling

feature selection

본 프로젝트에서는 feature selection 에서 아래 두 방법론을 적용해 분석을 진행하였다.

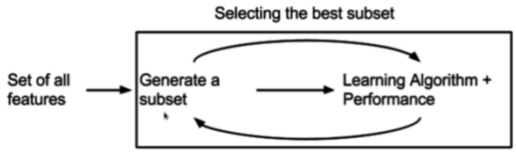
[stepwise method]



<그림 19> Wrapper method알고리즘

Stepwise 방법은 feature selection 의 3 가지 방법론 중 wrapper method 의 하나의 방법으로, Forward Selection 과 Backward Elimination 을 결합한 방법이다. 모든 변수를 가지고 시작하여 가장 도움이 되지 않는 변수를 삭제하거나, 모델에서 빠져있는 변수 중에서 가장 중요한 변수를 추가하는 방식이다. 이처럼 변수를 추가 또는 삭제를 반복한다. 반대로 아무것도 없는 모델에서 출발해 변수를 추가, 삭제를 반복할 수 있다.

[embedded method]



<그림 20> Embedded method알고리즘

Embedded method 는 Filtering 과 Wrapper 의 장점을 결합한 방법으로, 각각의 Feature 를 직접 학습하며, 모델의 정확도에 기여하는 Feature 를 선택한다. 계수가 0 이 아닌 Feature 가 선택되어, 더 낮은 복잡성으로 모델을 훈련하며, 학습 절차를 최적화한다.

2.3 분석 결과

2.3.1 USA Data

[impression prediction]

<표 2> 는 stepwise 방법을 이용하여 feature selection 을 한 후 각 모델에 적용해 day5,6,7 의 impression 을 예측한 결과이다. day0 과 day1 의 feature 를 합산하여 feature 로 사용하였다. 그 결과 전반적으로 banner 의 예측 성능이 가장 좋았으며 모델별 성능의 차이는 크게 나지 않았다.

Feature selection + model	banner	interstitial	reward	총 impression
Stepwise + Linear	3.30	3.56	10.32	15.59
Stepwise + Random forest	3.25	3.83	10.30	15.54
Stepwise + Xgboost	3.40	4.31	11.76	15.65
Stepwise + Lgbm	3.16	3.59	10.33	15.57

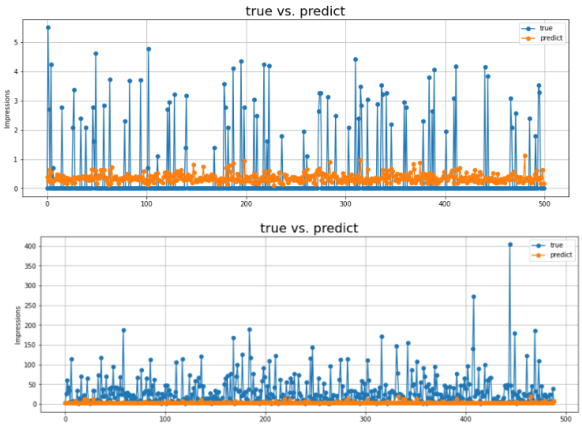
<표 2> day0, 1합한 feature로 적용한 모델 RMSE

<표 3> 은 위의 방법과 동일하지만 day0 과 day1 에 해당하는 feature 를 각각 적용했다. 그 결과 인터스티셜 예측 성능이 가장 좋았다.

총 impression 예측에서는 day0 과 day1 의 합산으로 feature 를 선택했을 때가 더 성능이 좋음을 알 수 있다. RMSE 값은 낮지만 정확한 성능 분석을 위해 위의 분석 결과를 각각 plot 해 보았다. 결과는 대부분 <그림 21>과 같이 대부분을 0 으로 예측하였다. day5,6,7 의 impression 가 있는 user 와 없는 user 의 feature 값의 차이가 뚜렷하지 않기 때문이라고 생각된다.

Feature selection + model	banner	Interstitial	reward	총 impression
Stepwise +SVM	4.30	3.63	10.05	40.51

<표 3> day0, 1 따로 feature로 적용한 모델 RMSE



<그림 21> day 5,6,7 impression예측 결과

예측 성능을 개선하기 위해 먼저 day5,6,7 에 남아있을 user 를 예측하는 classification 과정을 거쳐 남아있을 것으로 예상되는 user 로 regression 을 진행해보기로 하였다.

[Classification]

<표 4> 는 stepwise 를 이용하여 feature selection 후 모델을 학습시킨 결과이다. GridSearchCV 로 각 모델의 하이퍼 파라미터를 조정 후에는 DecisionTreeClassifier 모델에서만 0.02 의 유의미한 성능 향상을 보였다. 이후 테스트 셋 예측 성능은 Logistic Regression 모델이 0.6 으로 가장 좋았다. 하지만 여전히 낮은 성능으로 판단된다.

Feature selection method / classification	Train Score	Train Score (GridSearchCV)	Test Score
Stepwise + Logistic Regression	0.55	0.55	0.60
Stepwise + KNeighborsClassifier	0.52	0.52	0.49
Stepwise + DecisionTreeClassifier	0.54	0.56	0.59

<표 4> USA data Classification결과

2.3.2 Brazil Data

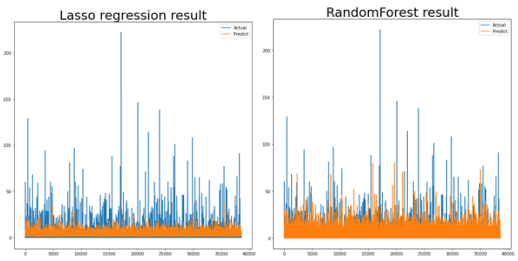
USA data 의 모델링 결과가 만족스럽지 않았다. 데이터의 양이 부족하여 예측 결과가 나오지 않았을 가능성도 있기 때문에 Aloha factory 측에서 추가로 제공한 Brazil data 로 위와 동일한 과정을 적용해보았다.

[impression prediction]

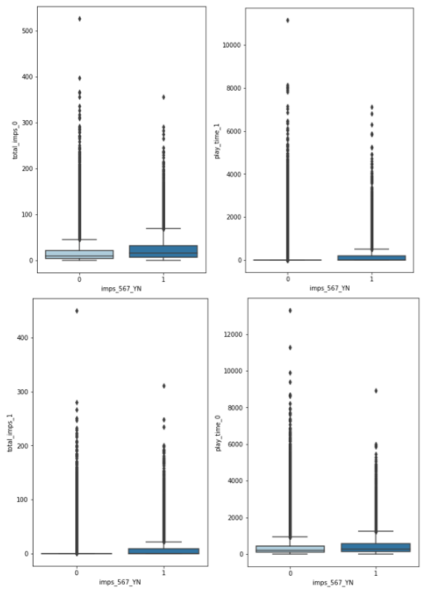
day0,1~ day5,6,7 impression

<그림 22>를 보면 day5,6,7 으로 종속변수를 설정하여 위의 과정을 진행해보았음에도 불구하고 예측이 잘되지 않는 것을 알 수 있다.

모델의 성능이 좋지 않은 원인을 파악하기 위해 day5,6,7 의 광고 노출 수의 합계가 0 이 아닌 user 와 0 인 user 를 구분하여 impression 과의 Correlation 이 큰 상위 5 개 feature 들의 분포를 비교분석을 해보았으나 아래의 <그림 23> 와 같이 두 집단에서 큰 차이가 없었다.



<그림 22> Brazil day0,1 ~ day5,6,7 imps예측 결과



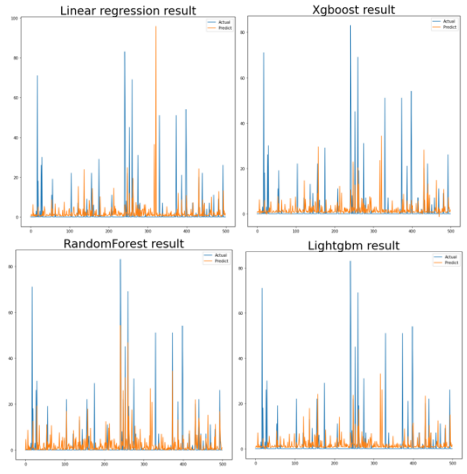
<그림 23> imps0인user와 아닌 user의feature

다양한 방법을 적용해봤음에도 불구하고 모델의 성능이 좋지 않은 이유를 분석해보자면 draw hammer 게임의 특성상 user 가 게임을 하면서 선택할 수 있는 폭이 넓지가 않다. 앱을 실행하면 draw start → draw end → stage start → play start → play end → stage end → Interstitial ads 과 같은 정해진 과정들이 진행된다. 또한 게임 초기에 user 는 게임 속 재화로 skin 을 사거나 능력치를 향상시키는 것도 제한이 되기 때문에 잔존 고객으로 특정지을 만한 뚜렷한 행동 양상을 찾아보기 힘든 것으로 판단된다.

<그림 7>의 day 에 따른 user 분포도를 보면, 신규가입자 중 day1 에 재접속하는 비율이 절반 이하이다. 신규 가입일로부터 시간이 흐를수록 feature 와의 상관성이 낮아지기 때문에 비교적 가까운 날인 day2 를 종속변수로 하여 분석을 다시 진행하였다.

- **day0,1~ day2 impression**

<표 5>는 day2 impression 을 예측하기 위해 4 가지 regression 모델에 적용한 결과이다. day567impression 을 예측했을 때의 RMSE 와 비교해보았을 때 RMSE 값이 줄어들었음을 알 수 있었다. 훨씬 0 으로 예측하는 비율도 줄어들었고, 실제로 잘 예측된 경우도 있었다. 하지만 이것을 그래프로 보며 실제값과 예측값을 비교해보면 예측이 잘 되었다고 볼 수 없다.



<그림 24> Brazil day0,1 ~ day2 imps예측 결과

Feature selection + model	Linear	Xgboost	Random Forest	lgbm
Embedded + 총 impression	7.92	7.75	5.90	7.34

<표 5> day0,1~day2 imps예측 결과 (RMSE)

[Classification]

이진 분류의 경우 정밀도와 재현율 사이에는 trade-off 관계가 존재한다. 본 프로젝트의 경우, 잔존하는 user 의 수가 매우 적고 잔존 고객에 대한 관리가 더 중요하기 때문에 재현율 즉 실제 잔존 고객을 이탈 고객으로 잘못 예측하면 상대적으로 큰 손실이 있다. 따라서 재현율을 높이는 것이 더 중요하다고 판단하여 분류 임계 값(threshold)을 낮추었다. 하지만 <표 6> 결과를 보면, 재현율은 높아지지만, 정밀도가 현저히 낮아 전체적인 분류 모델의 성능이 좋다고 할 수 없다.

Feature selection method / classification		Threshold = 0.5	Threshold = 0.35	Threshold = 0.3	Threshold = 0.25
Embedded + Logistic Regression	정확도	0.89	0.70	0.48	0.10
	정밀도	0.17	0.11	0.09	0.07
	재현율	0.20	0.50	0.74	0.99
Embedded + DecisionTreeClassifier	정확도	0.84	0.72	0.53	0.14
	정밀도	0.15	0.12	0.09	0.07
	재현율	0.32	0.52	0.71	0.97

<표 6> day0,1~day2 imps예측 결과 (RMSE)

2.3.3 Conclusion

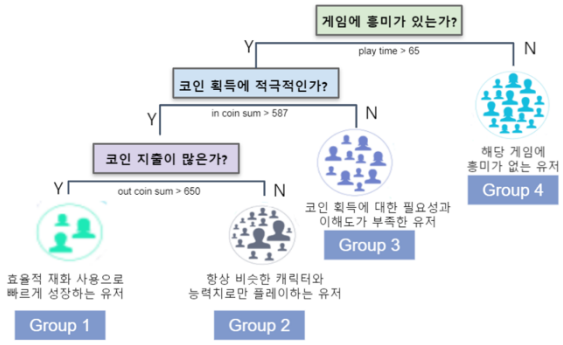
위의 과정을 통해서도 성능 개선이 유의미하지 못했기 때문에 User Segmentation 을 통해 각 그룹에 맞는 마케팅을 기업에 추천하여 고객 잔존율을 높이는 방안으로 전환하였다.

2.4 User Segmentation

2.4.1 By User Activity

[유저의 day0 의 플레이 패턴을 통해 우수고객의 기준을 정의]

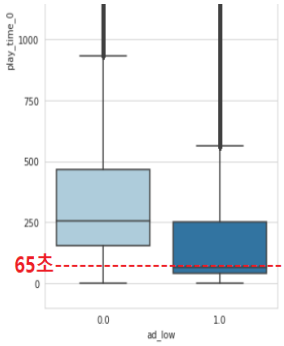
본 게임의 특성에 따라 아래와 같이 tree 방식으로 게임 흥미 유무, 재화 획득 적극성, 재화 사용 적극성의 세 가지 측면으로 분류했다.



<그림 25> 고객 세분화 트리

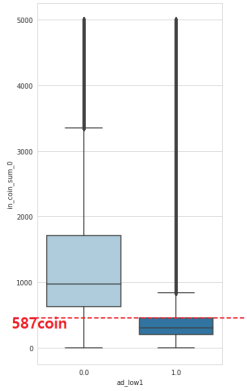
[분류 기준 수립]

각각의 분류 기준은 이전의 회귀모델에서 중요한 피처로 뽑힌 'play time', 'in coin sum', 'out coin sum'을 이용했다. 먼저 본 게임에서의 게임 흥미 유무는 다음 전투를 위한 광고 시청 의향의 유무로 해석 할 수 있다. 따라서 전체 유저의 day0 총 광고 노출 수의 사분위 수를 확인했고, 첫날 total 광고 수 상위 75%와 하위 25%의 play 시간을 비교했다.



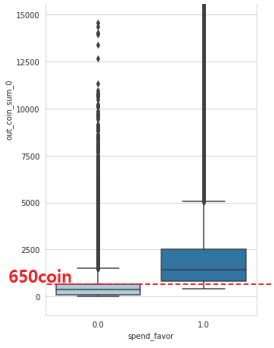
<그림 26> day0 전체 광고 수 상위 75% & 하위 25%의play시간

그 결과 하위 25% 클래스의 50%가 play 시간이 65 초 미만임을 확인했고, 해당 수치를 게임 흥미의 기준으로 설정했다. 게임에 흥미가 있는 유저라고 판단되었을 때 다음 세분화 기준을 '광고 시청 적극성'으로 수립했고 이는 즉 '코인 획득'과 직결된다. 게임에 흥미가 있는 유저 중 코인 획득과 관련 있는 is+reward 광고 노출 수가 하위 50%인 유저와 상위 50%인 유저를 나누었고, 각 클래스의 코인 획득량 분포를 확인했다.



<그림 27> day0 is + rv 광고 수 상위 50%
& 하위 50%의 코인 획득량

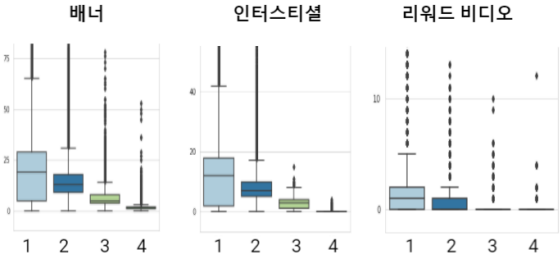
box plot 을 통해 하위 50% 클래스의 75%가 코인 획득량이 587 coin 미만임을 확인했고, 해당 수치를 코인 획득 적극성의 기준으로 설정했다. 그 결과 게임에 흥미는 있지만 코인 획득에는 필요성을 느끼지 못해 광고 시청 동기가 부족한 그룹 3 이 분류됐다. 위의 두 기준에 따라 '광고 시청에 적극적인 진성 유저'라고 판단되었을 때, 해당 유저들을 재화 사용률에 따라 두 패턴의 그룹으로 한 번 더 세분화하고자 했다.



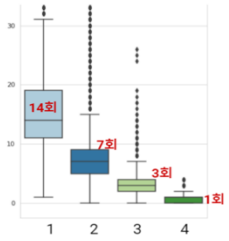
<그림 28> day0재화 사용률 상위 50%
& 하위 50%의 코인 소비량

수입 대비 지출 비율이 하위 50%인 유저의 코인 소비량의 75%를 설명할 수 있는 기준인 650 coin 으로 그 기준을 확립했다.

분류된 각 그룹의 첫날 광고 노출 수를 살펴보면 계층적으로 분포되어있고, 코인 소비 경향이 높은 그룹 1 의 상위 50%는 세 광고 노출 수에서 모두 최상위권에 위치한다. 특히 리워드 광고 시청 수는 그룹 1, 2 가 대부분을 차지하였으며, 그룹별 첫날 전투 횟수의 중앙값을 살펴보면 그룹 1 이 그 횟수가 확연히 높았다.



<그림 29> 그룹 별 day0 광고 시청 수 비교



<그림 30> 그룹별 day0 전투 횟수 비교

2.4.2 Marketing Strategy

최종적으로 결정된 기준에 따라 유저를 분류한 뒤, 그룹별 이탈 원인에 대해 파악하고 이탈 방지를 위한 마케팅 전략을 수립하였다.

[그룹 1] 효율적인 재화 사용으로 빠르게 성장하는 유저

첫 번째 그룹은 효율적인 코인 사용으로 빠르게 성장하며, 리워드 비디오 수익을 가장 많이 창출해주는 유저이다. 하지만 그들이 상위권 레벨에 도달하였음에도, 재화 사용으로 얻을 수 있는 아이템의 종류가 한정적이기 때문에 이탈이 촉진되었다고 분석하였다. 이를 방지하기 위해서는 특정 레벨 이상이 되어야 획득할 수 있는 아이템을 추가하는 등 게임을 지속하기 위한 동기부여 촉진이 필요하다.

<그림 31> 하위레벨과 상위레벨 능력치 및 스킨 비교

<그림 32> 타 게임의 로드맵 예시

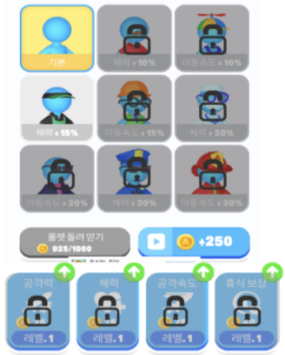
[그룹 2] 항상 비슷한 캐릭터와 능력치로만 플레이하는 유저

두 번째 그룹은 이들은 능력치나 스킨을 업그레이드 하지 않아서 항상 비슷한 캐릭터와 능력치로만 플레이 하므로 레벨이 올라감에도 승률이 높아지지 않고 권태에 빠지는 것으로 분석하였다. 이들을 관리하기 위해서는 능력치 업그레이드를 권고하는 게임 팁 관련 팝업을 띄워주거나 퀘스트를 통한 캐릭터 보상을 통해 새로운 캐릭터로 플레이하도록 유도하는 전략을 취할 수 있다.

<그림 33> 타 게임의 팁 및 보상 예시

[그룹 3] 코인 획득에 대한 필요성과 이해도가 부족한 유저

세 번째 그룹은 게임에는 흥미가 있지만 코인 획득에 대한 필요성과 이해도가 부족하여 게임 흥미 대비 광고 시청에 메리트를 느끼지 못하고 이탈하는 유저이다. 이들의 50%는 첫날 전투 횟수가 2 회에서 4 회 사이이지만, 상점은 레벨 7 부터 확인할 수 있기 때문에 이들은 해당 레벨에 도달하기 전까지는 광고를 시청해야 하는 이유와 코인 사용처에 대해 알 수 없다. 따라서 향후에 레벨이 높아졌을 때 구매할 수 있는 능력치와 스킨을 미리 보여줌으로써 광고 시청을 통한 코인 저축을 유도할 수 있다.



<그림 34> 게임 UI변화 예시

[그룹 4] 해당 게임에 흥미가 없는 유저

마지막 그룹은 모든 유저의 첫날 전투 횟수가 2 회 이하인, 해당 게임에 흥미가 없는 유저이다. 이들의 이탈을 방지하기 위해서는 지루할 수 있는 튜토리얼 및 레벨 초기 단계에서 화려한 전투와 가장 좋은 캐릭터를 먼저 보여주는 등 게임플레이의 역동적인 부분을 어필하여 몰입할 수 있는 콘텐츠임을 보여주어야 한다.

<그림 35> 타 게임의 예시.
초반 튜토리얼에서 보스 모드를 보여주어 흥미 유발

3 결론

초기 프로젝트 목표인 8 일차(day7) 광고 노출 수 예측하는 데에는 다소 아쉬움이 있지만 본 프로젝트는 현업에서 사용되는 게임 로그 데이터를 가공하고 모델링했다는 데에 의의가 있다.

처음 다루는 JSON 데이터를 분석하기 좋은 형태로 변환하고 로컬 노트북으로 감당할 수 없는 데이터 용량을 처리하는 경험을 통해 대용량 데이터 핸들링 경험도 체득할 수 있었다. 분석하며 느낀 바로는 분석에 앞서 서비스 이해가 필수적으로 선행되어야 한다는 것이다. 이 과정에서 앱에 대한 이해도를 높일 수 있고, 향후 분석을 위한 방향과 프레임을 잡는 데 유용한 정보를 얻을 수 있기 때문이다. 설계된 스키마대로 쌓이는 로그 데이터를 이해하기 위해 앱의 모든 화면과 이벤트를 상세히 파악하며 시나리오와 서비스의 흐름을 이해하기 위해 노력하였다. 또한 고객 생애 가치의 개념과 다양한 마케팅 기법에 관한 도메인 지식을 쌓고 해당 분야에 데이터 분석이 미치는 영향에 대해서 고찰해 보았다. 예측 과정에서는 이상치 및 결측치, 왜곡이 심한 변수에 대해 어떤 처리 방식이 모델의 성능 개선을 끌어낼 수 있을지 특성 공학적으로 접근할 수 있었다.

4 참고 문헌 및 자료

- 전희주 (2011). 고객 세분화에 기반한 생존분석을 활용한 고객수명 예측 모델, 한국 통계학회 논문집 687-695
- 모바일 게임 로그데이터를 이용한 게임플레이어 이탈 예측
- 고객 생애가치증대를 위한 CRM 마케팅 기법의 고찰
- Feature importance analysis for User Lifetime Value prediction in games using Machine Learning-an exploratory approach