**<본선 발표 대본>**

안녕하십니까. 2019 빅콘테스트 챔피언리그 리니지 유저이탈 예측 모형 공모전에 참여한 ‘우리팀화이팅’입니다. 저는 발표를 맡은 명재성입니다.

p2

발표순서는 다음과 같습니다.  
먼저 저희에게 주어진 문제를 정의하고,  
주어진 데이터를 살펴보고 이해하는 eda를 진행하였으며,  
데이터 전처리후에 모델링,  
그리고 모델링 결과 해석을 통한 사후 분석의 순서로 진행하였습니다.

*### Part 1. 문제 정의 ##############################################*

p4

우선 저희에게 주어진 문제 상황은 다음과 같습니다.  
게임 유저 이탈예측에 있어서 중요한 것은 단순히 해당 유저가 이탈을 할 것인지 아닌 지가 아니라,  
잔존가치가 높은 유저를 타겟팅하고, 해당 유저에게 인센티브를 제공함으로써 회사의 매출을 극대화할 수 있는 해지 방어 전략이 필요합니다.   
따라서 저희는 단순히 유저의 이탈을 예측하는 것이 아니라,  
유저의 이전 결제 행태와 게임 내 활동을 토대로  
유저의 생존기간을 파악하고, 유저의 생존기간에 따른 평균 결제 금액을 예측하여  
고객의 잔존가치의 합을 최대화하는 유저 이탈 예측 모형을 구축하였습니다.

*### Part 2. EDA ####################################################*

p6

다음은 EDA입니다. 주어진 데이터는 총 6개로,  
각각 label, 일별 거래 기록, 게임 내 활동 및 결제에 관한 데이터입니다.  
각각의 table은 캐릭터의 각 항목별 일일 활동 집계로써,  
한 명의 유저는 하나 이상의 캐릭터를 가질 수 있으며,  
각 유저별로 시간의 흐름에 따른 활동을 기록하고 있습니다.   
그러나 최종적으로 예측해야 할 대상은 유저의 생존 시간과 그 기간에 따른 일평균결제금액 이므로, 유저기준으로 데이터를 정리하였습니다.

p7

또한 제공된 train, test1, test2는 각각 수집된 시점이 다른 time-shift된 특성을 가지기 때문에, 보다 시간의 변화에 강건한 모델을 만들려고 하였습니다.

p8

train set에서 잔존/이탈 고객의 비율은 55:45로 balanced한 편이지만,  
생존시간으로 본다면 1에서 64일중 64일의 비율이 55%로 큰 비중을 차지합니다.  
또한 이탈고객의 경우, 생존시간이 1,8,29일인 사람이 많습니다.

p9

일평균결제금액이 0인 무과금 유저의 비율은 59%로 전체의 절반 이상이며,  
결제 금액과 생존 시간의 관계를 보았을 때,   
평균적으로 큰 금액을 결제한 유저는 빨리 이탈한다는 사실을 알 수 있었습니다.

p10

다음으로 combat 테이블은 전체 row중 약 28%만이 정보를 담고 있는 테이블이었습니다.

p11

이러한 combat 기록이 있는 유저들 중 이탈/잔존 유저,   
그리고 과금/무과금 유저 간의 기록을 비교해보았을 때,  
pvp를 활발하게 하는 유저들이 이탈하지 않고 과금도 많이 하는 경향을 보였습니다.  
다만 여기서 주목할 만한 점은 막피를 당했을 때는 그 경향이 반대로 나타났는데,  
오히려 이탈을 하고 과금을 하지 않는 경향을 보였습니다.

p12

다음으로 혈맹전투 데이터의 경우 일주일 간격으로 비슷한 경향성이 나타나는 시계열적 특성을 보였습니다.  
각 주차별로 유저들을 살펴보았을 때, ‘주간 혈맹전투 참여일수’를 보여주는  
pledge\_combat\_day가 높아질수록 유저들이 오래 생존하는 경향을 확인할 수 있었습니다.

*### Part 3. 데이터 전처리 ##############################################*

다음으로 데이터 전처리 과정에 대하여 설명드리겠습니다.

p14 ~ 17

우선 각 테이블에서 기존 변수의 28일간 활동량의 합, 평균, 비율을 계산하여 기본적인 변수들을 만들었고, EDA 결과를 기반으로 파생변수를 만들었습니다.

저희가 만든 파생변수는 다음과 같습니다.

p18

또한 주어진 table은 각 id 별로 시간에 따라 반복 측정된 자료이므로,  
28일간 총 활동뿐만 아니라 ‘일별 활동’이 이탈시기와 평균 결제 금액에 주는 영향을 반영하기 위하여 각 day별 변수를 생성하였습니다.  
해당 day에 활동기록이 없는 것들은 모두 0으로 대체하였습니다.

p19

다음으로 Feature selection 부분입니다. 이 그래프에서 왼쪽은 일별 결제 금액의 중위값, 오른쪽은 평균입니다.  
보시는 바와 같이 train과 test1,2의 중위값 및 평균의 일별 패턴이 다른 것을 확인할 수 있었습니다.  
그 이유는 첫 번째, 각 셋에서 집계를 시작한 날짜의 ‘요일’이 다르기 때문이고  
두 번째, 정기점검일의 변동과 이벤트 진행으로 인하여, 유저들의 결제 금액이 평소와 달라졌기 때문이라고 판단하였습니다.  
이러한 train과 test1,2의 일별 차이를 줄이기 위하여,   
각 셋에서 평균 결제 금액이 ‘안정적인’ 날들의 변수를 선택하여, 모델에 포함하였습니다.

p20

또한 최종적으로 선택한 변수들이 대부분 skewed되어있고 outlier가 존재하기 때문에,  
이를 보정하기 위한 Robust Scaling을 진행해주었습니다.

p21

결론적으로, 유저 별 28일간의 활동 정보를 반영하는 통합집계변수와 일별집계변수를 합한 총 683개의 변수를 input으로 사용하였습니다.

*### Part 4. Modeling ###############################################*

이제 본격적으로, 모델링 과정에 대하여 말씀드리겠습니다.

p 23

우선 간단한 Deep Learning approach를 통해서 모델링 진행 방향을 잡았고,   
최종적인 모델로는 Tree 기반 Ensemble 모델을 적합하였습니다

p24

먼저 별다른 데이터 전처리를 하지 않은 상태에서 DNN model을 적합 해보았습니다.

p 25

초기값을 변경해 가면서 DNN의 예측 결과를 확인하던 중, 가장 높은 score를 보인 부분의 생존시간과 일평균결제금액이 각각 하나의 상수 값으로 예측된 것을 확인할 수 있었습니다.

구체적으로 상수모델에서의 생존시간은 13.37로 비교적 낮은 값,   
일평균결제금액은 0.98로 상위 5% 이내의 매우 높은 값이 나타났습니다.   
비록 DNN model을 최종 model로 사용하지는 않았지만, 저희는 이 결과 자체는 모델링의 baseline으로 활용할 수 있을 것이라고 생각하였습니다.

p26

즉, 유저의 생존시간과 결제금액 각각을 정확히 맞추지는 못하더라도,  
하루에 많은 금액을 결제하고 빨리 이탈하는 유저를 예측해내는 것이  
기대이익 최대화에 있어 중요한 방향이라는 insight를 얻을 수 있었고,   
추후 modeling에서도 이를 활용하였습니다.

p27

다음으로 저희의 최종 모델인 Tree 기반 ensemble model에 대하여 말씀드리겠습니다.  
Tree 기반 ensemble model은 정형 데이터를 예측하는 데 가장 뛰어난 성능을 보이는 model입니다.  
물론 Hyperparameter에 따라 성능이 민감하게 반응하지만,   
최근 Grid search, random search 외에 Bayesian Optimization 등 다양한 방법론들이 등장하고 모듈화 되어 그 활용성이 더 높아졌습니다.

p28

최종 모델의 process는 다음과 같습니다. 여러 step에서의 ensemble 모델을 결합한 후,   
몇 가지 threshold를 통한 조정을 거쳐 최종 예측 값을 도출하였습니다.

p29

생존시간과 일평균결제금액을 예측하는 데 각각 회귀모델과 분류모델을 적합하였습니다.

p30

자세한 modeling 과정을 설명하기 이전에, model 최적화에 사용한 Bayesian optimization에 대하여 설명 드리겠습니다.  
저희가 사용한 boosting 모델의 특징은, 적절한 hyperparameter를 설정했을 때 높은 예측력을 보여준다는 장점이 있습니다.  
그러나 설정해주어야 할 hyperparameter의 dimension이 높다는 단점이 있는데,  
grid search와 random search의 경우 각각의 parameter 조합에 대해서 loss를 직접 계산해야 하므로 시간이 많이 소요된다는 문제가 있습니다.  
따라서 이들보다 상대적으로 계산 속도가 빠른 Bayesian Optimization을 사용했습니다.

<모델 1. 생존시간 Regression>

p31

다음으로 생존시간 예측 모델에 대한 설명입니다.  
생존 시간은 수치형 변수이긴 하지만, 55%의 비중을 차지하는 '64'로 인해   
전체적인 생존 시간이 과대 추정되게 됩니다.   
그러나 모든 머신 러닝 모델은 loss function을 최소화하는 방향으로 예측이 이루어지기 때문에, hyperparameter tuning만으로는 이 문제를 해결할 수가 없습니다.  
물론 주어진 score metric을 활용해서 loss function을 customize 할 수 있다면 가장 좋은 해결책이 될 수 있지만, score metric의 gradient와 hessian matrix를 구할 수 없었기 때문에 이에 대한 차선책을 생각해 보았습니다.

p32

그 차선책으로, 생존시간이 64보다 작은 '이탈한 유저'의 정보만을 input으로 하여 생존시간 예측 모형을 구축하였습니다.  
여러 ensemble model 중 같은 parameter boundary내에서 가장 좋은 성능을 보인 XGBoost model을 채택하였으며,  
생존시간은 일종의 ‘게임에 방문한 횟수’라 볼 수 있으므로,  
단위 구간 내의 사건의 발생 횟수와 관련되어 있고, 실제 적합 성능도 가장 좋았던 poisson regression 목적 함수를 선택하였습니다.

<모델 2. 일평균결제금액 Regression>

p33

다음으로 일평균결제금액의 경우 1보다 작은 값의 비중이 99% 정도이지만,   
최댓값은 약 39로 outlier가 심하게 나타나는 분포를 보입니다.  
따라서 전체적인 예측에 있어 outlier에 강건한 model을 적용해야 하며,   
동시에 기대 이익에 큰 플러스를 주는 outlier 유저를 담아낼 수 있는 접근 또한 필요합니다.

p34

이를 위하여, 우선 전체적인 경향을 반영하는 model로 outlier에 robust하며 fitting 속도 및 높은 예측 정확도 모두를 보장할 수 있는 LightGBM 모형을 채택하였습니다.

p35 ~ 36

그러나 LightGBM의 예측값을 보면 1보다 큰 금액을 결제한 outlier 유저를 제대로 예측하지 못한 것을 알 수 있는데,  
이러한 Outlier 유저의 경향 또한 최대한 반영하고자 앞선 LightGBM의 예측값을 그 표준편차로 나누어주는 scaling step을 도입하여 과소 추정 경향을 보정하였습니다.

<모델 3. 이탈/잔존 classification>

p37

앞서 살펴보았던 생존시간을 예측하는 모델 1의 결과, 실제 잔존 고객임에도 tree model의 한계 상 64를 정확히 예측하지 못하는 문제가 발생하였습니다.   
이러한 문제를 해결하기 위하여 이탈 유저와 잔존 유저로 binary classification을 진행한 후 이 때의 분류결과가 잔존이면 64, 이탈이면 모델 1의 예측 결과로 할당하였습니다.

p38

모델 3을 활용하여 생존시간 예측을 보정한 예시입니다.

<모델 4. 과금/무과금 classification>

p39

일평균결제금액 또한 모델 3과 같은 원리로,   
과금 대 무과금으로 binary classification을 진행한 후 모델 4의 분류 결과가 과금이면 모델2의 예측 결과로(모델 3은 오타입니다), 무과금이면 0을 할당하였습니다.

p40

모델 4를 활용하여 일평균결제금액 예측을 보정한 예시입니다.

p41

그러나 보정한 생존 시간과 일평균결제금액의 예측값으로 score를 구한 결과,   
보정하기 전의 score보다 오히려 감소하였습니다.  
저희는 주어진 score metric에서 그 원인을 찾을 수 있었습니다.  
즉, 지나치게 많은 유저들의 생존시간이 64로, 일평균결제금액이 0으로 보정되어 기대이익이 0이 되어버리는 문제가 발생한 것입니다.  
이 문제를 해결하기 위하여 저희는 binary classification에서 ‘아주 높은 확률’을 보이는,   
즉 확실한 증거를 가지는 obs만을 64 혹은 0으로 보정해주어야 한다고 생각하였습니다.

p42

앞선 이탈/잔존, 과금/무과금 분류에 있어서는 여러 binary classification model들 중   
가장 높은 score를 보인 Random Forest model을 사용하였습니다.  
그리고 앞서 보았듯이, 확실하게 생존할 유저와 확실하게 무과금할 유저를 분류해줄 예측 확률의 threshold 조합을 찾는 것이 중요한데,  
저희는 이를 Grid search를 통해 score를 maximize하는 threshold 조합으로 찾아주었습니다.

p43, 44

다음은 확률 최적 조합을 찾는 예시입니다.

p45

기대 이익의 최대화를 위한 또 하나의 필수 요소는   
바로 상위 3% 이내의 많은 금액을 결제한 유저의 결제 금액을 정확히 예측하는 것입니다.  
그러나 앞선 model process에서 이들의 경향을 전부 담아내는 것에는 한계가 있었기에,  
상위 3% 이내 예측값에 대하여 ‘몇 이상 몇 이하일 때 몇 배를 곱하는’ 과정을 최종 step으로 추가하였으며,  
필요한 3가지 parameter는 Grid search를 통해 score metric을 maximize하는 것으로 탐색, 최적화하였습니다. (예시는 다음과 같습니다.)

*### Part 5. 사후분석 #################################################*

사후분석에서는 최종 모델에서 얻은 insight를 중심으로 살펴보도록 하겠습니다.

[슬라이드 넘김] p49

LIME은 관심 변수의 값을 중심으로 임의의 값을 생성한 뒤, 그에 따른 예측 값이 바뀌는 정도를 확인하여 해당 변수의 중요도를 파악하는 해석 알고리즘으로,  
어떤 예측 모델에도 적용 가능하여 복잡한 블랙 박스 모델의 설명도 가능하게 해줍니다.  
저희는 앞서 사용한 4 가지 종류의 모델에 LIME을 적용하여 사후분석을 진행해보았습니다.

[슬라이드 넘김] p50

다음은 실제 생존여부가 1인 observation을 저희의 모델이 생존으로 예측한 경우입니다.

[슬라이드 넘김 – 빨간박스 생김]

이 결과에서 저희가 눈 여겨 본 feature는 바로 ‘pledge\_days’입니다  
pledge\_days는 유저가 속한 혈맹의 활동이 pledge 테이블에 기록된 일 수 입니다.  
즉, 28일의 관측 기간 중 pledge\_days가 26일 이상이면  
생존시간을 64로 예측할 유인이 높다는 것을 알 수 있습니다.

(예상질문) 혈맹 활동은 어떤 것들을 말하나요 ? :혈맹활동이란 pledge 테이블에서 기록이 있는 날짜들의 합 입니다.

[슬라이드 넘김 – 그래프]

실제로 data set을 살펴보았을 때,  
pledge\_days와 생존시간 사이에 강한 양의 상관관계가 나타나는 것을 확인할 수 있었습니다.

[슬라이드 넘김 – ‘혈맹활동’]

즉, 활발한 혈맹에 참여하는 유저일수록 생존시간이 증가하며, 이는 곧  
게임 내 사회활동이 지속적인 플레이의 원동력이 되고 있다고 할 수 있습니다.  
따라서 혈맹 활동의 이점들을 다양화하거나 관련 이벤트를 진행하는 등   
혈맹 관련 지원 및 프로모션을 통해 많은 유저들의 혈맹활동 참여를 이끌어낸다면,  
유저들의 이탈을 방지하고 이를 통해 잠재적인 가치를 얻을 수 있으리라 예상됩니다.

[슬라이드 넘김 – ‘total max level’ & ‘빨간 박스 쳐짐’]

다음은 과금 여부에 대해 살펴보겠습니다.  
과금 여부는 1인 경우 과금, 0인 경우 무과금 유저를 나타내는 변수입니다.  
저희가 눈 여겨 본 feature는 본캐의 레벨이라고 볼 수 있는 ‘total\_max\_level’입니다.

[슬라이드 넘김 – ‘막대 그래프’]

이 그래프의 가로 축은 ‘total\_max\_level’, 세로 축은 일평균결제금액입니다.

[슬라이드 넘김 – 빨간 박스]

모든 레벨 구간 중에서 40~44레벨 유저들이 가장 많은 금액을 결제하였는데, 이를 고려한 맞춤형 프로모션이 필요할 것입니다.  
또한 현재 리니지 육성 속도를 고려하였을 때, 25~29레벨은 신규 유저로 볼 수 있습니다.  
plot에 나와 있듯이, 25~29레벨 유저들 또한 고 레벨 유저들 못지 않게 많은 금액을 결제하고 있는 것을 확인할 수 있습니다.  
따라서 본캐의 레벨, 특히 40~44레벨 구간의 유저들과 신규 유저들을 대상으로 하는   
맞춤형 이벤트 및 상품 개발에 집중한다면   
결제 금액 향상을 통한 기대이익 증진에 기여할 수 있을 것이라 생각됩니다.

[슬라이드 넘김 – 막피 이슈]

p58

저희가 발견한 또 하나의 중요한 변수는 혈맹의 막피 경향을 보여주는 ‘def ratio’입니다

[슬라이드 넘김 – 빨간 박스]

여기서 알 수 있는 것은 막피를 가한 횟수에 비하여 막피를 당한 횟수가 많은 혈맹원의 경우 이탈 속도가 빠르게 증가한다는 것입니다.  
특히 def ratio가 90%를 넘어갈 경우 평균 생존시간이 급격하게 줄어들며,  
100%에 근접할 때 평균 생존시간이 약 39일 정도로 감소한다는 사실을 알 수 있었습니다.

[슬라이드 넘김 – 막피 이슈 설명]

즉, 무차별적인 PK로 인하여 게임 플레이에 지장이 생기는 경우,   
이로 인하여 게임을 그만두게 되는 플레이어들이 많다는 것을 알 수 있습니다.

[슬라이드 넘김 – 정리]

궁극적으로 잔존가치가 높은 이탈 유저를 예측하고,   
그들을 타겟으로 프로모션을 하는 것과 더불어,  
혈맹 활동, 특정레벨구간, 막피 이슈 등을 고려한 육성 가이드라인 개편 및 프로모션이 필요하다고 할 수 있습니다.

이상 발표를 마치도록 하겠습니다. 감사합니다.

**<본선 예상 질문>**

**##################### EDA & Feature Selection 관련 질문 ######################**

**Q. EDA를 바탕으로 만든 변수 & 강조하고 싶은 변수는 무엇이었나요?**

**A.** EDA를 통해서 파생변수를 새로 만든다고 해도, 이미 주어져있는 데이터를 활용한다면, 파생변수를 늘릴 필요없이 tree의 depth를 늘려 변수 간 교호작용을 반영함으로써 해결할 수 있다고 생각하여, 특별히 추가한 변수는 없습니다.   
(파생변수는 결국 서로 다른 변수들 간의 관계를 고려하여 만드는 것이므로)  
강조하고 싶은 변수의 경우는 저희가 사전조사과정에서 리니지 커뮤니티에 가장 많이 보였던 막피 관련 이슈입니다. 실제로 커뮤니티의 여론 뿐만 아니라 데이터를 통해서도 막피를 많이 당하는 유저들이 게임이탈을 많이 하는 것으로 확인되었습니다. 따라서 이러한 점이 개선되어야할 필요가 있다고 느꼈습니다.

Q. 어떻게 개선할 수 있을까?

A. 막피관련 이슈의 경우 막피 피로도 라는 개념을 도입하여 일간 가능한 막피 횟수를 제한하고 이를 초과하여 막피를 하기 위해서 재화를 소모하게 한다면, 무차별적인 막피의 문제를 해결함과 동시에 매출 증대도 노릴 수 있을 것으로 생각합니다.

**Q. EDA의 경우 발표자료에 그려진 것이 전부인가요? eda를 통해서 향후 모델링에 영향을 끼친 부분이 있다면 어느 부분인가요?**

**A.** 막피 관련 변수

**Q. 변수 select, dimensional reduction 어떻게 했나요?**

**A.** tree 기반 모델의 경우 모델 자체가 다중공선성에 robust 하기 때문에 최대한 가용한 변수를 모두 사용하였습니다. 다만 day별 변수들중 ppt에서 말씀드린 것처럼 일간 소비 패턴이 비정상적인 날짜의 경우 결과에 부정적인 영향을 주어서 제거 하였습니다.

**Q. feature selection 부분에서 안정적인 날들만을 선택하였다고 했는데 무슨 기준…?**

A. 실제 게임을 해보면서 느낀 경험은 매달 1일 결제 한도가 초기화되는 날이나, 특별한 이벤트가 진행될때 많은 결제를 진행하게 됩니다.   
따라서 저희는 결제금액이 급격하게 높아지는 날들을 이러한 noise로 생각하여서, 제외하고 모델링을 진행하였습니다

**Q. playtime같은 변수도 있었는데, 왜 feature selection을 일별 결제 금액 기준으로 하셨나요?**

**A.** 분석의 목표가 기대 이익을 최대화 하는 것이었고, 이러한 목표에 있어 가장 중요하게 작용했던 요소가 결제 금액이었기 때문에 결제 금액을 최우선 기준으로 삼고 feature selection을 진행하게 되었습니다.

**Q. Scaling된 데이터를 저희가 드렸는데, 왜 다시 Scaling하신 건가요? 그 효과가 무엇이었나요?**

**A.** 기존에 제공된 Scaling의 경우 평균이나 중앙값등의 대푯값을 반영하지 않고 단순히 표준편차로 나누어진 Scaling이었기 때문에, 변수간 척도의 차이를 다 상쇄하지 못했다고 생각하여서 진행하게 되었고, 실제로 Scaling 결과 안정적으로 더 높은 score를 얻을 수 있었습니다.

**############################ Modeling 관련 질문 ############################**

**Q. 왜 굳이 딥러닝 모델로 constant를...?**

**A.** DNN 모형을 적합했을 때 다른 tree based model에 비하여 성능이 매우 안좋아 최종 모델로 사용하지는 못하였지만,   
Constant model을 통해 낮은 생존 시간과 높은 결제 금액을 보이는 유저의 예측이 중요하다는 핵심적인 insight를 얻을 수 있었으므로 발표에 수록하게 되었습니다.

**Q. Hyperparameter는 어떤 것들을 바꾸셨나요?**

**A.** boosting 모델의 경우 l1,l2 정규화 부분과 트리갯수, 그리고 learning rate 를 튜닝 했습니다.

**Q. threshold 조합을 찾을 때 각 grid의 간격은 어떻게 설정하였으며, grid search를 사용했다고 했는데 계산 속도가 느리지 않았나요?**

**A.** 0.01단위로 grid를 주었으며, threshold의 dimension이 2차원으로 크지 않아서 local 환경에서 충분히 빠르게 최적화되었습니다.

**Q. Classifier는 파라미터 최적화를 왜 하지 않았는가.**

**A.** 마찬가지로 Bayesian Optimization을 사용하여 최적화를 해 보았지만 empirical 하게 hyperparameter를 설정했을 때보다 score가 오히려 좋지 않게 나와서, 따로 최적화를 하지 않았습니다.

**Q. 분류를 먼저 하고 regression을 하면서 threshold 조정을 할 수도 있었을 텐데 순서를 왜 이렇게?**

**A.** 그러면 각 threshold 조합마다 regression model을 모두 fitting 해주어야 하기 때문에 모델링에 너무 오랜 시간이 걸려서, 저희는 빠르게 모델링을 하고 예측 결과의 추이를 살펴보기 위해 이러한 방식을 채택하였습니다.

(또한 실제 모델을 게임 운영에 적용한다면 기간이 바뀜에 따라서 계속 모델의 수정이 필요할 수 있는데 이러한 부분에서 계산속도가 문제가 될 수 있다고 판단하였습니다)

**Q. 시간이 더 주어진다면 어떻게 본인들의 모델을 발전시킬 수 있을까요?**

**A.** 분류를 먼저 진행하고 0이 아니라고 예측하는 값들에 대해서만 regression

**Q. 본인들의 모델이 다른 팀 모델들에 비하여 차별되는 점이 무엇이라 생각하나요?**

**A.**

**Q. survival\_time과 amount\_spent간의 상관관계가 있었을 수도 있는데 이를 고려해보았는가?**

**A.** 상관관계를 고려하기 위해서 먼저 생존시간을 예측하고 예측된 생존시간을 설명변수로 넣어서 결제금액을 예측해보는 방식으로 상관관계를 고려한 모델링도 진행해 보았습니다. 그러나 각각을 독립적으로 예측한 모델의 성능이 더 좋았기 때문에 상관관계를 반영한 모델을 결과로 사용하지는 않았습니다.

**Q. amount\_spent를 std으로도 나누고 multiply하는 작업을 거쳤었는데, 이렇게 되면 전체적으로 amount\_spent가 너무 커져 오히려 과대 추정의 문제가 발생하는 것이 아닌가요?**

**A.** 그러한 문제가 있다는 것을 저희도 인지하였지만, 주어진 score metric을 바탕으로 일평균결제금액을 과대 추정 하는 것이 과소 추정 하는 것보다 기대이익을 최대화하는데 더 유리하였기 때문에 이러한 방식을 채택하였습니다.  
실제로 과대 추정할 경우 각 obs별 손실이 발생하였는데 손실의 최대치가 -10~20인것에 반해 이익은 300~500 정도로 훨씬 큰 값이 도출되어서, 과대추정으로 인한 손해보다 이익이 더 큰 것을 알 수 있었습니다.

**Q. amount\_spent multiply 과정에서 ceiling을 어떻게 주었으며, ceiling을 줄 시 오히려 원래 추정된 최대값보다 작은 값들이 오히려 최대값보다 더 커지게 되는데, 이렇게 되면 원래 amount\_spent를 더 왜곡하는 문제가 발생하는 것이 아닌가요?**

**A.** Ceiling은 상위 0.01~0.1% 정도의 예측 값으로 주었습니다.  
물론 합리적으로 생각하였을 때에는 곱한 후의 값에 대한 최댓값의 상한을 정해주는 것이 맞다고 생각하여 예측값의 max 등으로 이러한 상한을 부여해 보았으나, 실제 score가 ceiling을 주고, 상한은 부여하지 않는 것이 하는게 가장 좋은 결과를 내서 이러한 방식을 채택하였습니다.

**Q. 총 4가지 모델을 스코어 기준으로 좋은 모델을 선택하였다고 하는데, 스코어가 유의미할 정도로 차이가 있었나요? 그렇지 않다면 모델을 통일성 있게 짜는 것이 좋지 않나요?**(lightgbm, xgboost)

**A.** 네. 확실히 생존시간, 일평균결제금액 그리고 binary target 별로 score가 제일 높게 나오는 모델이 달랐기 때문에 각기 서로 다른 ensemble model을 결합하는 방식을 채택하게 되었습니다.

**Q. threshold를 5-fold에서 나온 값들의 평균을 사용했다는 것에 대한 설명과 이유가 무엇인가요?**

A. threshold를 정하는데 있어서도 testset에 과적합되는 문제가 발생 할 수 있다고 생각하여서, 데이터셋을 5-fold로 나누어서 나온 threshold값의 평균을 사용하였습니다.

**########################### 사후분석 관련 질문 ###########################**

**Q. 왜 pdp를 안쓰고 lime을 썼죠?**

**A.** PDP는 상관관계가 매우 강한 변수들이 많을 때 변수의 effect가 심하게 왜곡되는 단점이 있습니다. 반면에 lime의 경우, Lasso를 통해 개별적인 feature selection을 진행하기 때문에 다중공선성 문제에서 보다 자유롭다는 장점이 있습니다.

그런데 저희가 분석에 사용한 변수들이 상관관계가 매우 높았기 때문에, PDP보다는 LIME이 적절하다고 생각하여 사용하게 되었습니다.

**Q. lime은 unstable한 경향이 있지 않나요?**

**A.** lime과 대척점에 있는 사후분석 방법으로는 pdp가 존재합니다. 그러나 저희가 사용한 모델에서는 변수간 상관관계가 높은 문제가 있어서, pdp로는 유의미한 결과를 도출 할 수 없는 문제가 있었습니다. 따라서 lime을 사용했습니다.

**Q. lime의 경우 각각의 obs별로 변수가 뽑히는데 obs별로 다른 결과가 나오진 않는지? 그렇다면 obs를 선택한 기준은?**

**A.** classification을 한 후 적용하는 threshold가 0.8~0.9 정도로 나왔었기 때문에 lime을 통해 모델을 해석할 때, 예측 확률이 그 정도에 해당하는 observation들을 주로 보았습니다. 그 observation들에서 공통적으로 유의한 변수가 많이 등장하는 결과를 대표적으로 보여드린 것입니다.

**Q. 사후분석 파트에서, total\_max\_level이 구간 14 ( 70-74 )이하일 때 무과금을 무과금으로 분류할 확률이 높은데 그림상에서는 돈을 14이하에서 더 많이 쓰는 것으로 보이는데...?**

**A.** 저희가 그려놓은 히스토그램의 경우에는 유저들 중에서 결제 금액이 0인 사람들의 비율이 아니라, 결제 금액의 평균을 그린 히스토그램이기 때문에 그러한 모습이 보이는것 같습니다.

**Q. pledge\_days가 단순히 playtime이나 num\_of\_days와 상관관계가 높아서 survival\_time과 양의 상관관계를 보이게 된 것이 아닌가**

**A.**pledge\_days는 단순히 개인이 혈맹활동에 참여한 날들을 센 변수가 아니라, 자신이 속한 혈맹이 활동을 한 날을 센 것이기 때문에 다른 의미가 있다고 생각하였습니다.