안녕하십니까 >\_< 2019 빅콘테스트 챔피언스리그 리니지 유저이탈 예측 모형 공모전에 참여한 ‘우리팀화이팅’입니다. 저는 발표를 맡은 (***명재성*** ) 입니다. ^-^

p2

발표순서는 다음과 같습니다. 먼저 저희에게 주어진 문제를 정의하고, 주어진 데이터를 살펴보는 eda를 진행하였고, **데이터 전처리** 후에 모델링 그리고 결과 해석 순서로 진행하였습니다.

p4

우선 저희에게 주어진 문제 상황은 다음과 같습니다.

게임 유저 이탈예측에 있어서 중요한 것은 단순히 해당 유저가 이탈을 할 것인지 아닌지가 아니라, 잔존가치가 높은 유저를 타겟팅하고, 해당 유저에게 인센티브를 제공함으로써 회사의 매출을 극대화 할 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 합이 최대가 되도록 하는 유저 이탈 예측 모형을 만들었습니다.

1:00

### Part 2

p6

주어진 데이터는 총 6개의 table로 각각 정답, 일별 거래 기록, 게임 내 활동 및 결제에 관한 데이터입니다. 각각의 table은 캐릭터의 각 항목별 일일 활동 집계로 한명의 유저는 하나 이상의 캐릭터를 가질 수 있으며, 각 유저별로 시간의 흐름에 따른 활동기록을 보이고 있습니다.

그러나 최종적으로 예측해야 할 대상은 유저의 생존 시간과 그 기간에 따른 일 평균 결제금액이므로 유저기준으로 데이터를 정리하였습니다.

p7

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

p8

train set에서 잔존과 이탈 고객의 비율은 55:45로 balanced한 편이지만, 생존시간으로 본다면 1에서 64일중 64일의 비율이 55%로 큰 비중을 차지합니다. 이탈고객은 생존기간이 1,8,29일인 이탈자가 많습니다.

p9

결제 금액이 0인, 무과금 유저는 59%로 절반 이상입니다. 결제 금액과 생존 시간의 관계를 보았을 때, 평균적으로 큰 금액을 결제한 유저는 이탈한다는 사실을 알 수 있습니다.

2:10

p10

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

p11

#(상) random\_attacker\_cnt / (하)random\_defender\_cnt

combat기록이 있는 유저들 중에서 이탈/잔존 유저 그리고 과금/무과금 유저간의 기록을 비교해보았을 때 pvp 활동이 활발한 유저들이 이탈하지 않고 과금도 많이 하는 경향을 보였습니다. 다만 여기서 주목할만한 점은 막피를 당했을 때는 그 경향이 반대로 나타났는데, 오히려 이탈을 하고 과금하지 않는 경향을 보였습니다.

p12

혈맹전투 데이터의 경우 일주일 간격으로 비슷한 모습을 보이는 시계열적인 특성을 보였습니다.

각 주차별로 유저들을 살펴보았을 때 주간 혈맹전투 참여일수를 보여주는 pledge combat day가 높아질수록 유저들의 생존시간이 높아지는 모습을 확인 할 수 있었습니다.

3:00

### Part 3

데이터 전처리에 대하여 설명하겠습니다.

우선 각 테이블의 기존 변수를 28일간 활동량의 합, 평균, 비율을 계산하여 기본적인 변수들을 만들었고, EDA 결과를 기반으로 파생변수를 만들었습니다. 저희가 만든 파생변수는 다음과 같습니다.

p18

주어진 table은 각 id 별로 시간에 따라 반복 측정된 자료로, 28일간 총 활동 뿐만 아니라 일간 활동이 이탈시기와 평균 결제 금액에 주는 영향을 반영하기 위하여 각 day별 변수를 생성하였습니다. 해당 day에 활동기록이 없는 것은 모두 0으로 대체하였습니다.

3:40

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 확인 할 수 있었습니다. 그 이유는 첫 번째, 각 셋에서 집계를 시작한 날짜의 요일이 다르기 때문이고 두 번째, 정기점검일의 변동과 이벤트 진행으로 인하여, 유저들의 결제 금액이 평소와 달라졌기 때문이라고 판단하였습니다. train과 test1,2의 일별 차이를 줄이기 위하여, 각 셋에서 평균 결제 금액이 안정적인 날들의 변수를 선택하여, 모델에 포함하였습니다.

p 20

다음으로 최종적으로 선택한 변수들 중 대부분의 변수들이 skewed되어있고 outlier가 존재하기 때문에 이를 보정하기 위한 Robust Scaling을 진행해주었습니다.

p 21

결론적으로, 28일간의 활동 정보를 반영하는 통합집계변수와 일별집계변수를 합한 683개의 변수를 사용하였습니다.

4:30

### Part 4 Modeling

다음으로, 본격적인 모델링 과정에 대하여 말씀드리겠습니다.

p 23

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

p24

먼저 최종 modeling 과정에는 포함되지 않았지만, 별다른 데이터 전처리를 하지 않은 상태로 DNN model을 적합해보았습니다.

p 25

초기값을 변경해 가면서 DNN의 예측 결과를 확인하던 중, 생존시간과 평균결제금액이 각각 하나의 상수 값으로 예측된 것을 확인할 수 있었습니다.

가장 높은 score를 보인 상수모델에서의 생존시간은 13.37로 비교적 낮은 값, 평균결제금액은 0.98로 상위 5% 이내의 매우 높은 값이 나타났으며, 저희는 이를 모델링의 baseline으로 활용할 수 있을 것이라고 생각하였습니다.

p 26

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

5:40

p 27

다음으로 최종 모형인 Tree-based ensemble model에 대하여 말씀드리겠습니다.

Tree 기반 모델의 경우 정형데이터를 예측하는데 가장 뛰어난 성능을 보이는 모델입니다.

물론 Hyperparameter에 따라 성능이 민감하게 반응하지만, 최근 Grid, random search 외에 Bayesian Optimization 등 다양한 방법론들이 등장하고 모듈화되어 그 활용성이 더 높아졌습니다.

----------------------------------------------------------------------------------------------------------

p 28

최종 모델 process는 다음과 같습니다. 여러 step에서의 ensemble 모형을 결합한 후, 몇 가지 threshold를 통한 조정을 거쳐 최종 예측값을 도출하였습니다.

----------------------------------------------------------------------------------------------------------

p 29

생존시간과 일평균결제금액을 예측하는 데 각각 회귀모델과 분류모델을 적합하였습니다.

6:20

p30

자세한 modeling 과정을 설명하기 이전에, model의 최적화에 사용한 Bayesian optimization에 대하여 설명드리겠습니다.

저희가 사용한 boosting 모델의 특징은, 적절한 hyper parameter를 설정했을 때 높은 예측력을 보여준다는 장점이 있습니다.

그러나 설정해주어야 할 hyper parameter의 dimension이 높다는 단점이 있는데, grid search와 random search의 경우 각각의 parameter 조합에 대해서 loss를 직접 계산해야하므로 시간이 많이 소요된다는 문제가 있습니다. 따라서 이들보다 상대적으로 계산 속도가 빠른 Bayesian Optimization을 사용했습니다.

따라서 저희가 hyper parameter를 찾기 위해 사용한 방법은 Bayesian optimization 방법입니다. parameter조합에 대해서 loss function값을 계산해서 비교하는 grid search 혹은 random search와 다르게 acquisition function을 통해 제안된 parameter조합의 loss 값을 surrogate model로 근사시키는 과정을 거치게 됩니다. 이를 통해서 loss function를 approximate하고 loss를 가장 낮출수 있는 parameter를 찾게 됩니다.

최종 parameter 설정에 있어서는 classification 모델 2가지의 경우 bayes opt를 사용한 모델보다 초기에 empirical하게 설정하였던 parameter에서 더 좋은 결과를 보여주었기 때문에 empirical하게 설정한 hyper parameter를 그대로 사용하였습니다.

6:50

p31

다음으로 생존시간 예측 모형에 대한 설명입니다.

생존 시간은 수치형 변수이긴 하지만, 다른 값들에 비해 '64'가 55%의 비중을 차지하여, 이로 인해 생존 시간 자체를 과대 추정하게 됩니다.

그러나 저희가 사용한 앙상블 모델은 loss function 을 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, 이때 score metric의 gradient와 hessian matrix를 구할수 없었기 때문에 차선책을 생각해 보았습니다.

p32

이러한 차선책으로, 생존시간이 64보다 작은 '이탈한 유저'의 정보만을 input으로 하여 생존시간 예측 모형을 구축하였습니다.

같은 parameter boundary안에서 가장 좋은 성능을 보인 XGBoost model을 채택하였으며,

생존시간은 게임에 ‘방문한 횟수’라 볼 수 있으므로,   
단위 구간 안에서 사건의 발생 횟수를 예측하며, 실제 적합 성능도 가장 좋았던 poisson regression 목적 함수를 선택하였습니다.

7:50

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.

(전체적인 경향을 잘 반영하되, 결제 금액이 매우 높은 유저 또한 탐지할 수 있는 step을 추가해주어야 한다.)

p34

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

p35~36  
(Plot에 대한 설명)  
이러한 Outlier 유저의 경향 또한 최대한 반영하고자 앞선 LightGBM의 예측값을 그 표준편차로 나누어주는 scaling step을 도입하여 과소 추정 경향을 보정하였습니다.   
8:40

p37

< 모델 3. 이탈/잔존 classification >

앞서 살펴보았던 survival\_time(생존시간)을 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 predict 결과로 할당하였습니다.

p38

모델 3을 활용하여 생존시간 예측을 보정한 결과 예시입니다.

9:20

p39

< 모델 4. 과금/무과금 classification >

다음은 일평균결제금액에 대해 과금 vs 무과금으로 분류하는 모델에 대한 설명입니다.

따라서 과금 대 무과금으로 binary의 classification을 진행한 후 모델 4의 분류 결과가 과금이면 모델2의 예측결과로 무과금이면 0을 할당하였습니다.

p40

모델 4를 활용하여 일평균결제금액 예측을 보정한 결과 예시입니다.

(과정은 앞선 결과와 같습니다)

10:10

p41

그러나 보정한 생존 시간과 일평균결제금액 예측값으로 score를 구한 결과, 보정하기 전의 score보다 오히려 감소하였습니다. 이유가 무엇일지 고민해보았을 때, 주어진 score function으로부터 그 원인을 찾을 수 있었습니다.

즉, 문제를 간단하게 설명하자면 지나치게 많은 유저들의 생존시간을 64로 그리고 일평균류결제금액을 0으로 보내는 문제가 발생하였습니다.

따라서 저희는 확실한 증거 즉 binary classification에서 높은 확률을 보여주는 obs만을 64 혹은 0으로 보정해주어야 한다고 생각하였습니다.

p42

앞선 이탈/잔존, 과금/무과금 분류에 있어서 사용한 모델은 Random Forest입니다. binary classification이 가능한 여러 model을 모두 사용해 본 결과 가장 높은 score를 얻은 Random Forest를 선택하였습니다.

그리고 앞서 보았듯이 확실하게 생존할 유저와 확실하게 무과금할 유저를 분류해줄 예측 확률의 threshold 조합을 찾는 것이 중요합니다.   
따라서 저희는 Grid search를 통해 score를 maximize하는 threshold 조합을 찾아주었습니다.

11:00

p43, 44

다음은 최적 조합 찾는 예시입니다.

p45

앞서부터 보았듯이, 기대 이익 metric의 최적화를 위해서는 상위 3% 이내의 많은 금액을 결제한 유저의 결제 금액을 정확히 예측하여야 합니다.

그러나 앞선 model process에서 이들의 경향을 전부 담아내는 것에는 한계가 있었으며, 따라서 상위 3% 이내 예측값에 대해서 몇 이상 몇 이하일 때 몇 배를 곱하는 과정을 최종 step으로 추가하였으며,   
필요한 3가지 parameter는 Grid search를 통해 score metric을 maximize하는 것으로 탐색, 최적화하였습니다.

11:40

##############

5번. 사후분석에서는 저희의 모델을 통하여 얻은 insight를 중심으로 살펴보도록 하겠습니다.

[슬라이드 넘김] p 49

LIME은 관심 변수의 값을 중심으로 임의의 값을 생성한 뒤, 그에 따른 예측 값이 바뀌는 정도를 확인하여 해당 변수의 중요도를 파악합니다. 따라서 어떤 예측 모델이더라도 적용 가능하고 이때문에 딥러닝 혹은 복잡한 머신 러닝 모델 설명 시 사용 가능합니다. 저희는 앞서 사용한 4 가지 종류의 모델에 LIME을 적용하여 사후분석을 진행해보았습니다.

[슬라이드 넘김] p 50

실제 생존여부가 1인, 즉 생존시간이 64인 observation을 저희의 모델이 생존으로 예측한 경우입니다.

[슬라이드 넘김 – 빨간박스 생김]

이 결과에서 저희가 눈 여겨 본 feature는 바로 ‘pledge\_days’입니다. 28일의 관측 기간 중, 혈맹 활동에 참여한 날의 합이 26일 이상이면 생존시간을 64로 예측할 유인이 높다는 것을 알 수 있습니다.

[슬라이드 넘김 – 그래프]

실제로 training data를 살펴보았을 때, pledge\_days와 생존시간이 강한 양의 상관관계를 보이는 것을 확인할 수 있었습니다.

[슬라이드 넘김 – ‘혈맹활동’]

즉, 혈맹활동에 많이 참가하는 user일수록 생존시간이 증가한다는 것을 알 수 있습니다. 다시말해서 게임 내 사회활동이 지속적인 플레이의 원동력이 되고 있는 것입니다.

따라서 유저들의 혈맹활동을 장려함으로써 유저들의 이탈을 방지 할수 있을것으로 예상됩니다. 현재 혈맹에 가입할 경우 60레벨 미만의 캐릭터는 경험치 보너스 효과가 있는 혈맹 버프를 얻고 있습니다. 이러한 혈맹 활동의 이점들을 상향하거나, 혈맹 활동과 관련된 이벤트를 진행하는 등 혈맹 활동 관련 지원 및 프로모션을 통하여 혈맹활동에 많은 유저들이 참여 할 수 있도록 장려해야합니다.

13:10

[슬라이드 넘김 – ‘total max level’]

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 – ‘빨간 박스 쳐짐’]

LIME을 통해 확인하였을 때, observation의 total\_max\_level이 14이하인 경우 해당 observation을 무과금 유저로 예측할 유인이 높다는 사실을 확인할 수 있었습니다.

[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 소모하였는데, 이를 고려한 맞춤형 프로모션이 필요할 것입니다. 또한 현재 리니지 육성 속도를 고려하였을 때, 25~29레벨은 신규 유저로 볼 수 있습니다. plot을 보시면 알 수 있듯이, 25~29레벨 유저들 또한 고레벨 유저들에 못지 않게 많은 금액을 결제하고 있는 것을 확인할 수 있습니다. 만약 본캐의 레벨에 따라 맞춤형 프로모션을 진행한다면 기존 유저들, 특히 40~44레벨 구간의 유저들과 신규 유저들을 대상으로 하는 이벤트 및 상품 개발에 집중하면 좋을 것입니다.

14:20

[슬라이드 넘김 – 막피 이슈]

p 58

저희가 발견한 또 하나의 중요한 변수는 바로 ‘def ratio’ 입니다. 여기서 ‘def ratio’는 전체 막피활동중 막피를 당한 비율입니다. ( 막피당함/ (당함+가함))

[슬라이드 넘김 – 빨간 박스]

여기서 알 수 있는 것은 막피를 가한 횟수에 비하여 막피를 당한 횟수가 많은 경우 이탈률이 빠르게 증가한다는 사실을 알 수 있었습니다. 특히 def ratio가 90%가 넘어갈 경우 평균 생존기간이 급격하게 줄어들며, 100%에 근접할 경우 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.

14:50

[슬라이드 넘김 – 정리]

결론적으로 잔존가치가 높은 이탈 유저를 예측하고, 그들을 타겟으로 프로모션을 하는것과 더불어 혈맹 활동, 특정레벨구간, 막피이슈등을 고려한 육성 가이드라인 개편 및 프로모션이 필요합니다.

이상 발표를 마치도록 하겠습니다. 감사합니다.

15:10

##### 예상 질문 #####

Q. 왜 pdp를 안쓰고 lime을 썼죠?

A. pdp는 단위에 민감(영향이 크고), lime은 그런 거 없이 얘네들의 magnitude를 같은 스케일로 비교하여

블랙박스 모델 해석에 있어 많이 사용되는 지표로는 LIME, PDP 등이 있습니다. 보편적으로 많이 사용되는 PDP의 경우, 특정 변수의 중요도를 확인하기 위하여 해당 변수 값의 일괄적인 변화에 따라 predicted value의 평균적인 변화를 시각화한 Plot입니다. 블랙박스 모형 해석시에 자주 사용되지만 상관관계가 매우 강한 변수들이 많을 때 변수의 effect가 왜곡되는 경우가 발생할 수 있는 단점이 존재합니다.

하지만 저희가 사용하는 변수들은 상관관계가 강한 변수들이 분명 존재하였으며, 따라서 PDP를 사용하기보다는 LIME을 사용하는 것이 적절하다고 생각했습니다.

Q. 이러한 데이터는?

A. longitudinal data. random effect 모델. random effect를 다루기 위해 day별로 쪼개서 전처리

Q. EDA를 바탕으로 만든 변수 & 강조하고 싶은 변수

A. ??????????????????

Q. EDA를 통해 발견 -> 넣었을 때 ->

Q. 변수 select, dimensional reduction 어떻게 했나요?

Q. 왜 굳이 딥러닝 모델로 constant를...?

Q. bayesian opt의 장점

A. grid search와, 계산 소요시간 감소, 계산 cost 감소

Q. threshold 조합

A. 0.01단위, dimension이 크지 않아서 local 환경에서 충분히 돌아갔다.

Q. lime은 unstable한 경향이 있다

A.

Q. 하이퍼파라미터 어떤 거 바꿨는지

Q. playtime같은 변수도 있는데, 왜 feature selection을 일별 결제 금액기준으로 했는가.

분석의 목표가 가치를 최대화 하는 과정이였기때문에 가치 = 돈 따라서 금액으로 설정하였습니다.

Q. Classifier는 파라미터 최적화를 왜 하지 않았는가.

마찬가지로 베이즈 옵트로 최적화를 해 보았는데 score점수가 emphirical 설정한 하이퍼 파라미터보다 성증이 좋지 않게 나와서 그랬다

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

시간과 비용 떄문

Q. 사후분석 파트에서, total\_max\_level이 14이하일 때 무과금을 무과금으로 분류할 확률이 높은데 그림상에서는 돈을 14이하에서 더 많이 쓰는 것으로 보이는데...?

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

어떻게 하면 좋을까요?? // 분류를 먼저 진행하고 0이 아니라고 예측하는 값들에대해서만 regression

Q 스케일링된 데이터를 저희가 드렸는데, 왜 다시 스케일링 한건가요? 그 효과가 무엇이었나요?

기존에 제공된 스케일링의 경우 평균이나 중앙값등의 대푯값을 반영하지 않고 단순히 분산으로 나누어진 스케일링이였기 떄문에, 변수간 척도의 차이가 발생 할수 있다고 생각하여서

했따

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

다른팀에는 관심이없어서...

(질문)

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

그러한 문제가 있다는 것을 저희도 인지하였지만, 주어진 score를 바탕으로 값을 과대추정하는것이 과소추정하는것보다 잔존가치를 보존하는데 더 유리하였기 떄문에

실제로 과대추정할 경우 각 obs별 손실이 발생하였는데 손실의 최대치는 -10~20 이익은 500떳어용 따라서 과대추정으로 인해 손해보는것보다 이익이 더커서 아시겠어요? 에?

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

합리적으로 생각하였을때는 최댓값의 상한을 정해주는 것이 맞다고 생각하여 그렇게 진행하였으나, 실제 score의 경우는 요렇게 하는게 더 좋아용

3. Robust scaling을 할 때 train, test1, test2 모두를 활용하여 scaling을 했다고 들었다. 그런데 modeling을 할 때는 training data만을 사용하는데, 왜 training data와 test data 모두를 통합하여 scaling을 진행하고 이를 modeling 과정에 활용하였다면 이는 문제가 있는 것 아닌가?

test data의 아웃풋 받기위해서 들어가는 input test 데이터도 scaling된게 들어가기 떄문에 영향이 없어요 에?

[ eda의 경우 발표자료에 그려진 것이 전부인지? eda를 통해서 햐후 모델링에 영향을 끼친 부분이 있다면 어느 뷰분인지?

막피 관련 변수

[ 총 4가지 모델을 스코어 기준으로 좋은 모델을 선택하였다고 하는데 스코어가 유의미할 정도로 차이가 있었는지 그렇지 않다면 모델을 통일성 있게 짜는게 좋은게 아닌지? (lightgbm xgboost)

네 유의미합니다!

[ lime의 경우 각각의 obs별로 변수가 뽑히는데 obs별로 다른 결과가 나오진 않는지? 그렇다면 obs를 선택한 기준은?

threshold가 0.9 정도니까/// 공통적으로 유의한 변수가 많이 등장하는 것으로 분류를 잘 한 모델중에서 random하게 select 하였습니다.