LG Aimers 6기

난임 치료 데이터 기반 임신 성공 예측 모델링:
CatBoost, XGBoost, BERT 스태킹 및 Iterative Imputation 기법의 통합적 적용

2025.3.5

권휘준 이도은 최지호

목차

1. 프로젝트 개요

문제제기

1

고령화와 늦은 출산으로 인해
난임 치료를 받는
환자의 수가 빠르게
증가하고 있음

2

저출산이 심각한
사회적 문제로 대두되는 만큼
난임 치료가 출산율 향상에
중요한 역할을 할 수 있음

3

난임 치료에 대한
높은 비용과 심리적 부담으로
인해 치료 시기를 놓치거나
포기하는 사례가
늘어나고 있음

팀 구성 및 역할

최지호

- 데이터 전처리
- 파생변수 생성
- Catboost
  하이퍼파라미터 튜닝

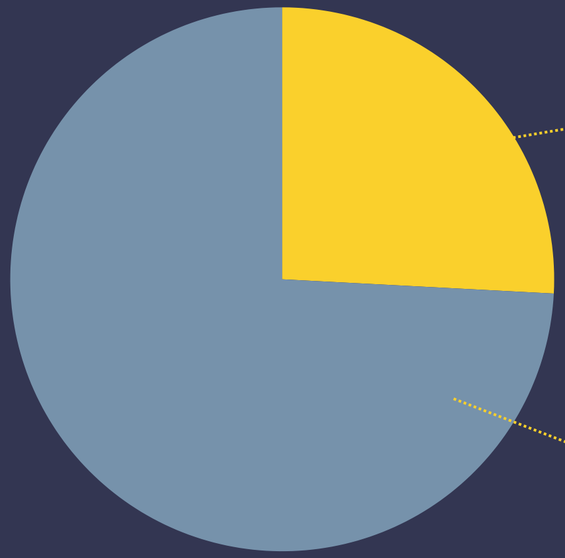
이도은

- 파생변수 생성
- Xgboost, BERT 모델링
- Stacking

권휘준

- 데이터 전처리
- 파생변수 생성
- 하이퍼파라미터 튜닝
- GitHub & Notion 관리

데이터 개요



[Target : 임신 성공여부]

25.8%

임신 성공 데이터 66228개 (25.8%)

74.2%

임신 실패 데이터 190123개 (74.2%)

- HFEA 데이터셋 (2010~2016)을 기반
- 총 데이터의 수 256351개
- 칼럼의 수 69개 (Target 포함)

2. 데이터 탐색 및 전처리

개요

1
특정 변수 처리

2
이상치 제거

3
순서형 변수 인코딩

4
결측치 처리

5
분산 0인 변수 제거

6
파생변수 생성

1. 특정 변수 처리

배아 생성 주요 이유
[object]

1. 현재 시술용
2. 배아 저장용
3. nan
4. 기증용, 현재 시술용
5. 난자 저장용
6. 기증용
7. 기증용, 배아 저장용
8. 배아 저장용, 현재 시술용
9. 기증용, 난자 저장용
10. 기증용, 배아 저장용, 현재 시술용
11. other

“,”를 기준으로 나눌 필요가 있음

특정 시술 유형
[object]

1. ICSI
2. IVF
3. Unknown
4. IUI
5. ICSI:ICSI
6. ICSI / BLASTOCYST
7. IVF / BLASTOCYST
8. IVF:IVF
9. ICSI:IVF
10. ICSI / AH
11. other

“:”, “/”를 기준으로 나눌 필요가 있음

1. 특정 변수 처리

배아 생성 주요 이유

배아 생성 주요 이유
[object]

1. 현재 시술용
2. 배아 저장용
3. nan
4. 기증용, 현재 시술용
5. 난자 저장용
6. 기증용
7. 기증용, 배아 저장용
8. 배아 저장용, 현재 시술용
9. 기증용, 난자 저장용
10. 기증용, 배아 저장용, 현재 시술용
11. other

'현재 시술용', '난자 저장용', '배아 저장용', '기증용'
칼럼을 만들어 해당하는 값이 있으면 1, 없으면 0으로 처리
'연구용'은 수가 너무 적어 모델에 유의미한 영향을 주지 않을 것이라 판단하였음

배아 생성 주요 이유	현재 시술용	난자 저장용	배아 저장용	기증용
현재 시술용	1	0	0	0
NaN	0	0	0	0
난자 저장용	0	1	0	0
배아 저장용	0	0	1	0
기증용, 현재 시술용	1	0	0	1
기증용, 배아 저장용	0	0	1	1
기증용	0	0	0	1

1. 특정 변수 처리

특정 시술 유형

특정 시술 유형
[object]

1. ICSI
2. IVF
3. Unknown
4. IUI
5. ICSI:ICSI
6. ICSI / BLASTOCYST
7. IVF / BLASTOCYST
8. IVF:IVF
9. ICSI:IVF
10. ICSI / AH
11. other

“FER”, “GIFT”는 수가 매우 적어 “Unknown”으로 대체
“:”, “/”을 기준으로 split하여 원핫인코딩을 진행하였음

특정 시술 유형	ICI	IVF	GenericDI	BLASTOCYST	IUI	AH	ICSI	Unknown	IVI
ICSI	0	0	0	0	0	0	1	0	0
IVF	0	1	0	0	0	0	0	0	0
Unknown	0	0	0	0	0	0	0	1	0
IUI	0	0	0	0	1	0	0	0	0
IVF:IVF	0	1	0	0	0	0	0	0	0
IVF / BLASTOCYST	0	1	0	1	0	0	0	0	0
ICSI:IVF	0	1	0	0	0	0	1	0	0
ICSI / AH	0	0	0	0	0	1	1	0	0
ICSI:ICSI	0	0	0	0	0	0	1	0	0
IVF:ICSI	0	1	0	0	0	0	1	0	0
ICSI / BLASTOCYST	0	0	0	1	0	0	1	0	0
ICSI:Unknown	0	0	0	0	0	0	1	1	0

2. 이상치 제거

‘시술 유형'이 ‘IVF’인 행 중 앞뒤가 맞지 않는 이상치 제거

- ‘특정 시술 유형'에 ‘ICSI’가 포함되지만 ‘미세' 포함 변수 값이 0이 아닌 행 삭제
- ‘이식된 배아 수'가 0인데 ‘배아 이식 경과일' > 0인 행 삭제
- ‘동결 배아 사용 여부'가 0인데 ‘해동된 배아 수’ > 0인 행 삭제
- ‘동결 배아 사용 여부'가 0이고 ‘배아 해동 경과일'이 결측인데 ‘해동된 배아 수'가 0이 아닌 행 삭제
- ‘동결 배아 사용 여부'가 0인데 ‘배아 해동 경과일'이 결측이 아닌 행 삭제

3. 순서형 변수 인코딩

총 시술 횟수	클리닉 내 총 시술 횟수	IVF 시술 횟수	DI 시술 횟수	총 임신 횟수	IVF 임신 횟수	DI 임신 횟수	총 출산 횟수	IVF 출산 횟수	DI 출산 횟수
0회	0회	0회	0회	0회	0회	0회	0회	0회	0회
1회	0회	1회	0회	0회	0회	0회	0회	0회	0회
1회	1회	1회	0회	0회	0회	0회	0회	0회	0회
2회	2회	2회	0회	0회	0회	0회	0회	0회	0회

횟수 관련 변수들은 범주형 변수로 사용하기보다 ordinal encoding후 순서 정보를 반영하고자 하였음

"n회" → n

3. 순서형 변수 인코딩

총 시술 횟수	IVF 시술 횟수	DI 시술 횟수
0회	0회	0회
1회	1회	0회
2회	2회	0회
6회 이상	3회	5회
3회	3회	0회

총 임신 횟수	IVF 임신 횟수	DI 임신 횟수
0회	0회	0회
1회	1회	0회
2회	2회	0회
1회	0회	1회
3회	3회	0회

총 출산 횟수	IVF 출산 횟수	DI 출산 횟수
0회	0회	0회
1회	1회	0회
1회	0회	1회
2회	2회	0회
2회	0회	2회

“총 ~ 횟수" = “IVF ~ 횟수" + “DI ~ 횟수"

“총 ~ 횟수" 변수들은 좀 더 구체적으로 처리하였음.

4. 결측치 처리

‘시술 유형'이 ‘IVF’인 경우 결측치 처리

- ‘이식된 배아 수'가 0이면 ‘배아 이식 경과일'의 결측값을 0으로 대체
- ‘동결 배아 사용 여부'가 0이면 ‘배아 해동 경과일’의 결측값을 0으로 대체

배아 이식 경과일 결측치 처리

- ‘배아 이식 경과일' 변수와 상관계수가 0.5 이상인 변수들로 IterativeImputer() 함수 사용
- ‘배아 이식 경과일'의 변수 중요도가 비교적 높았기 때문에 이러한 방법을 사용함

5. 분산 0인 변수 처리

```
train['불임 원인 - 여성 요인'].nunique()
✓  0.0s

1
```

‘불임 원인 - 여성 요인' 제거

6. 파생 변수 생성

‘~ 시술 실패 횟수' & ‘~ 임신 실패 횟수'

- ‘~ 시술 실패 횟수' = ‘~ 시술 횟수' – ‘~ 임신 횟수'
- ‘~ 임신 실패 횟수' = ‘~ 임신 횟수' – ‘~ 출산 횟수'

‘~ 임신 성공률' & ‘~ 출산 성공률’ & ‘~ 시술 경험 여부'

- ‘~ 임신 성공률' = ‘~ 임신 횟수' / ‘~ 시술 횟수'
- ‘~ 출산 성공률' = ‘~ 출산 횟수’ / ‘~ 임신 횟수'
- ‘~ 시술 경험 여부' : 해당 시술 경험이 있을 경우 1, 없을 경우 0

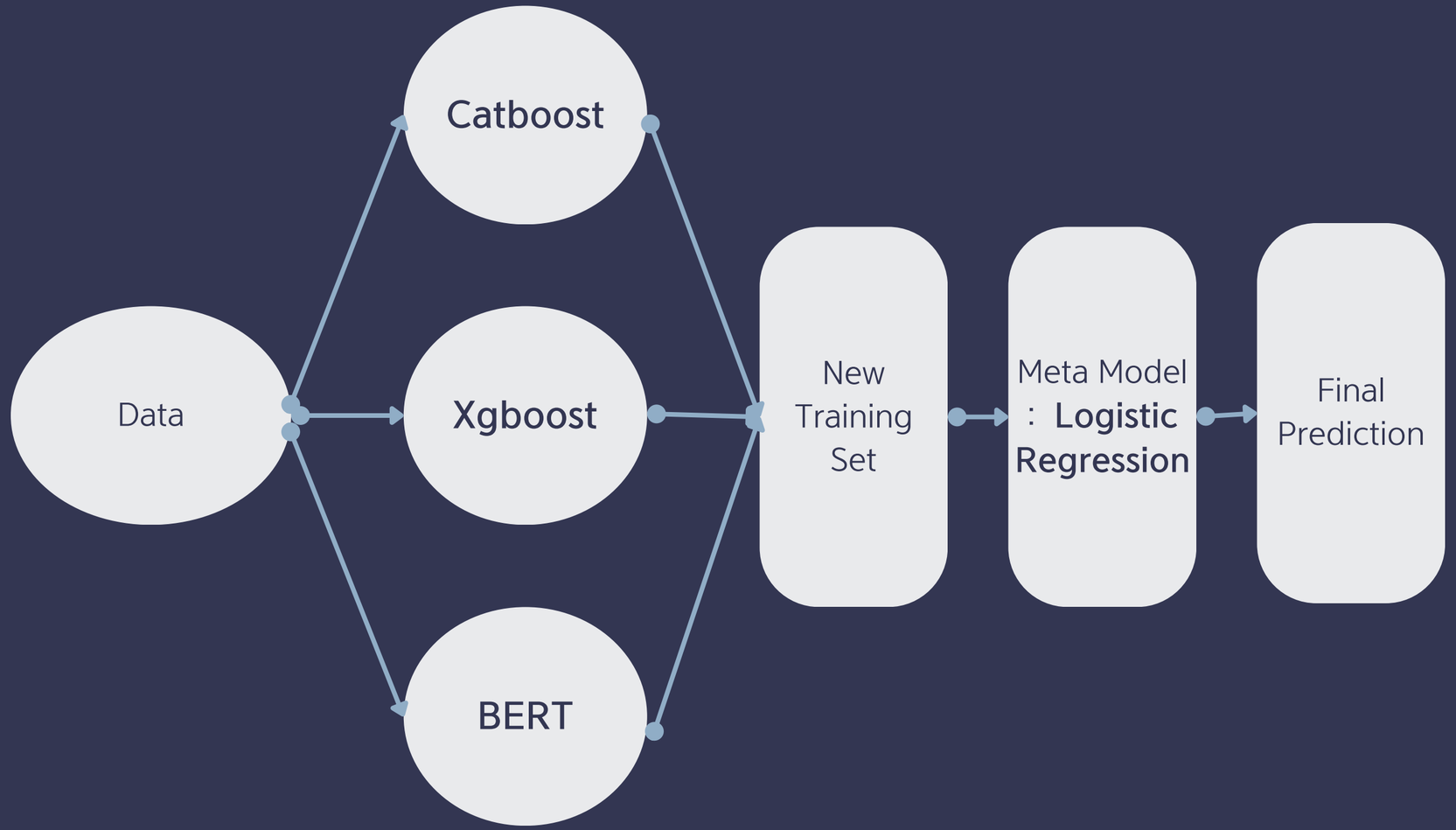
6. 파생 변수 생성

‘자연 수정 배아 비율'

- 1 – ‘미세 주입 배아 이식 수' / ‘이식된 배아 수'

3. 모델링

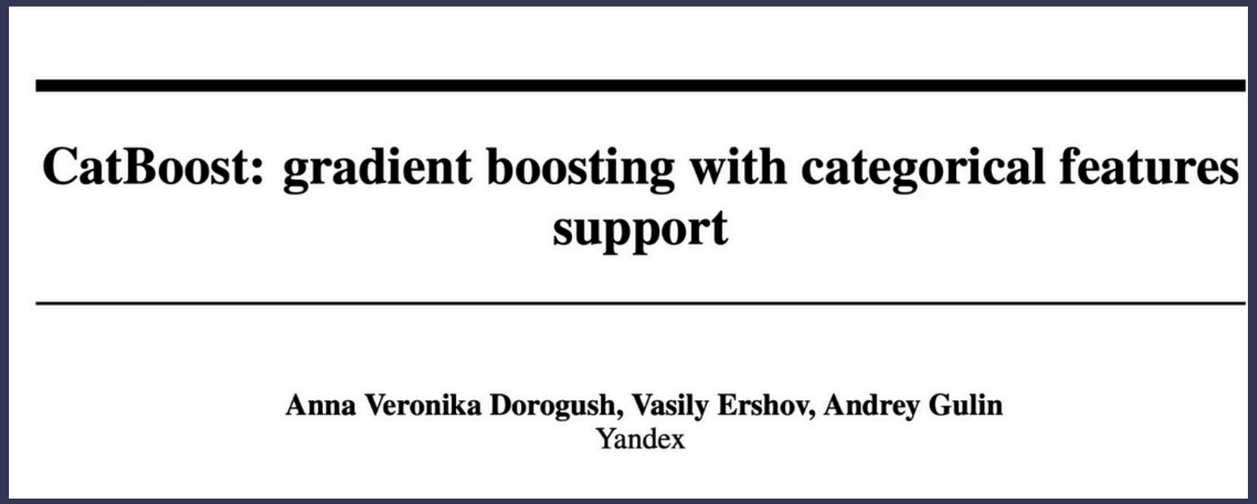
Ensemble : Stacking



Catboost의 강건성을 위해서 Xgboost와 BERT 모델을 추가적으로 **Stacking** 하였음.

Base Learner로 Catboost, Xgboost, BERT를 사용하였고, Base Learner의 예측값을 입력으로 받아 Logistic Regression을 Meta Modeler로 사용하였음

CatBoost



CatBoost: gradient boosting with categorical features support

Anna Veronika Dorogush, Vasily Ershov, Andrey Gulin
Yandex

Catboost는 Gradient Boosting Decision Tree 모델로 범주형 변수를 효과적으로 처리하도록 최적화되어 있음

자동으로 범주형 변수를 숫자형으로 변환하고,
Ordered Boosting 기법으로
Overfitting 위험을 낮춤

하이퍼파라미터 튜닝

Grid Search와 3-fold Cross Validation을 결합해 탐색 진행

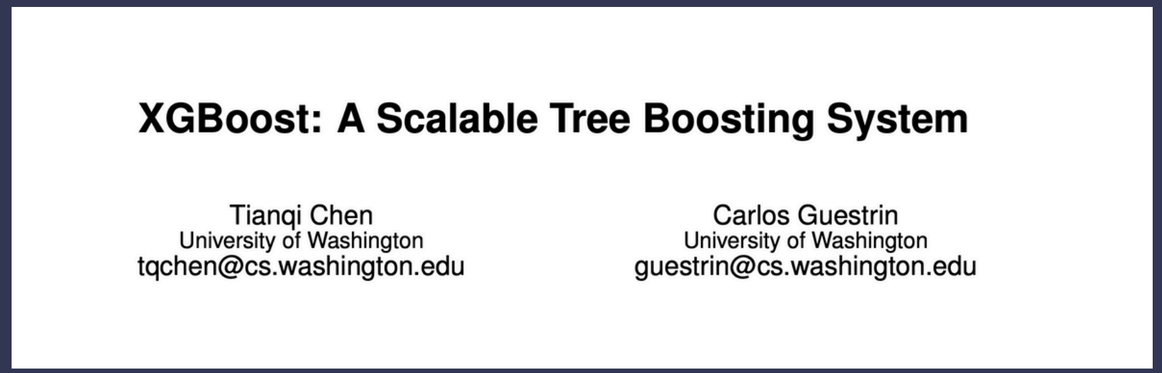
튜닝 대상 파라미터
depth, l2_leaf_reg, learning_rate, iterations, scale_pos_weight

범주형 변수(cat_features) :
'['시술 시기 코드', '시술 당시 나이', '시술 유형', '배란 유도 유형', '해동 난자 수', '난자 출처', '정자 출처', '난자 기증자 나이', '정자 기증자 나이']

하이퍼파라미터 튜닝 결과 - Catboost

Parameter	Description	Value
depth	트리의 최대 깊이를 설정해 모델의 복잡도를 조절하고 과적합을 방지함.	7
l2_leaf_reg	리프의 가중치를 억제하여 과적합을 방지	145
learning_rate	학습률 조절	0.032
iterations	반복 횟수 : 트리의 개수	920
border_count	연속형 변수를 이산화하는 경계 수를 지정	64
scale_pos_weight	불균형 클래스 문제 해결을 위해 양성 클래스의 가중치를 강화	3.5106

XGBoost



XGBoost: A Scalable Tree Boosting System

Tianqi Chen
University of Washington
tqchen@cs.washington.edu

Carlos Guestrin
University of Washington
guestrin@cs.washington.edu

Catboost와의 차이점 : Catboost와 달리 범주형 변수를 사전 인코딩 필요(One-hot, Label encoding)

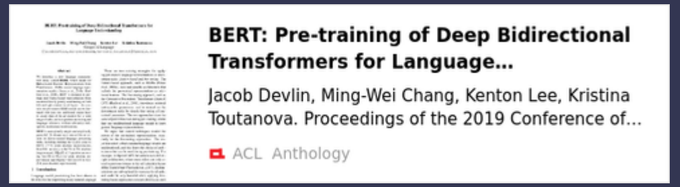
CatBoost는 범주형 변수를 자동으로 처리하고 순서형(Target-based) 인코딩을 기본 적용하나, XGBoost는 수치형 변수 처리에 더 최적화됨

XGBoost의 장점 : 과적합 방지를 위해 다양한 규제 항목(L1, L2)를 세부적으로 조정 가능

하이퍼파라미터 튜닝 결과 – XGBoost

Parameter	Description	Value	Parameter	Description	Value
learning_rate	모델 학습 시 각 트리의 기여도를 조절하여 과적합을 방지	0.0365	colsample_bytree	트리 생성 시 사용할 피처(특성)의 비율을 지정하여 과적합 방지 및 다양성 확보	0.7904
n_estimators	모델 학습 시 각 트리의 기여도를 조절하여 과적합을 방지	434	lambda	L2 정규화를 통해 트리 가중치에 대한 규제를 적용하여 모델의 복잡도 감소	4
max_depth	트리의 최대 깊이를 제한하여 모델의 복잡성을 제어하고 과적합 방지	4	alpha	L1 정규화를 적용해 트리의 복잡성을 제어하고 중요하지 않은 특성을 제거	10
min_child_weight	리프 노드의 최소 가중치 합을 설정해 과적합을 방지하고 일반화 성능 향상	1	scale_pos_weight	불균형 클래스 문제 해결을 위해 양성 클래스의 가중치를 높여 모델 성능 개선	6
gamma	노드를 추가적으로 분할할 때 필요한 최소 손실 감소량을 설정하여 트리 구조 단순화	4	tree_method	히스토그램 기반 알고리즘을 사용해 빠르고 효율적으로 트리를 학습	“hist”
subsample	각 트리를 학습할 때 사용할 데이터의 샘플 비율을 정해 과적합 감소	0.5879			

BERT *Devlin et al. (2019) – BERT: Bidirectional Transformers for NLP



모델의 강건성을 위해 Base Model로 BERT* 사용

➡ 데이터의 대부분이 범주형 변수기 때문에,
한 행을 문장으로 변환하여 target 변수를 예측해야하는 다음 단어로 간주

- CatBoost, XGBoost와 동일하게 이상치 제거
- 결측치의 경우 범주형 : '알 수 없음', 이진형 : 'False', 연속형 : 평균 대체
- 연속형 변수는 범주화 하여 텍스트화

예시)

시술 시기 코드	시술 당시 나이	시술 유형	배란 자극 여부	배란 유도 유형	임신 성공 여부
TRZKPL	만18-34세	IVF	1	기록되지 않은 시행	1
TRYBLT	만45-50세	None	0	알 수 없음	0

➡ {"시술 시기 코드 : TRZKPL, 시술 당시 나이 : 만18-34세 , 시술 유형 : IVF, 배란 자극 여부 : True, ..., 임신 성공 여부 : True"
"시술 시기 코드 : TRZKPL, 시술 당시 나이 : 만18-34세 , 시술 유형 : IVF, 배란 자극 여부 : True, ..., 임신 성공 여부 : True"}

BERT

- BERT의 ‘BertTokenizerFast’ 이용하여 텍스트 토큰화
- DistilBERT model 사용
  - Optimizer : AdamW (lr =2e-5, eps=1e-8)
  - Linear Learning rate Scheduler를 사용하여 학습률 점진적 감소
- Epoch 5회 동안 AUC 점수를 기준으로 성능이 좋은 모델을 최적의 모델로 선정

Meta Model : Logistic Regression

- Stacking의 Meta Model로써 해석이 용이하며 과적합을 방지할 수 있는
  ➡ Logistic Regressor 선택

$$logit(p_i) = ln\frac{p_i}{1-p_i} = \beta_1 \cdot \mathrm{CatBoost} + \beta_2 \cdot \mathrm{XGBoost} + \beta_3 \cdot \mathrm{BERT}$$

Coefficient	β_1	β_2	β_3
추정값	**5.22813311**	**0.1662358**	**-0.03656457**

- 임신 성공 여부 예측시,
  CatBoost에 비해 XGBoost는 약 32배, BERT는 173배 더 적은 영향을 미침
  ➡ CatBoost를 주된 예측모델로 사용하되,
  XGBoost, BERT에 의해 다양한 방식으로 데이터를 학습하여 과적합 방지

4. 결과

Feature Importance 상위 6개의 변수

Feature	Importance	Value	Feature	Importance	Value
38	이식된 배아 수	56.974119	35	총 생성 배아 수	2.665002
1	시술 당시 나이	6.243503	49	난자 출처	2.054706
62	배아 이식 경과일	5.805183	0	시술 시기 코드	1.062556
59	난자 채취 경과일	4.648658	92	자연 수정 배아 비율	0.981425
40	저장된 배아 수	4.391952	5	배란 유도 유형	0.703907
44	수집된 신선 난자 수	3.031707	50	정자 출처	0.665666

성능 향상에 기여한 핵심 전략
: 데이터 전처리

1 이상치 제거 :
논리적으로 맞지 않는 이상치 제거가 성능 향상에 기여함

2 Iterative Imputer :
배아 이식 경과일에 Iterative Imputer를 통한 데이터 결측 대체가 효과적이었음.

성능 향상에 기여한 핵심 전략
: 모델링

1 모델 앙상블 : 3가지 모델을 Stacking하여 데이터의 다양한 패턴에 대해 예측을 통합

2 GPU 기반 튜닝이 학습 속도 향상에 도움이 되었음

3 Catboost 사용시, cat_features 지정이 성능에 중요한 역할

성능 향상에 미흡한 접근

1

DI와 IVF 모델링 분리 :
별도 모델링이 성능 향상에 미비한 효과

2

Under & Oversampling :
두 샘플링 기법 모두 Catboost의 scale_pos_weight 도입하는 것보다 성능이 낮았음

3

Feature Importance 기반 변수 선택 :
Importance가 높은 변수를 기반으로 모델링하는 것이 성능향상에 도움이 되지 않음

Reference

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers) (pp. 4171-4186).

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. Advances in neural information processing systems, 31.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

thank you!