PREDICTING ICU Mortality

A Machine Learning Approach

2023 Fall - Data Analytics

정보융합학부 김주혁 장원재

TABLE OF CONTENTS

01.

INTRODUCTION

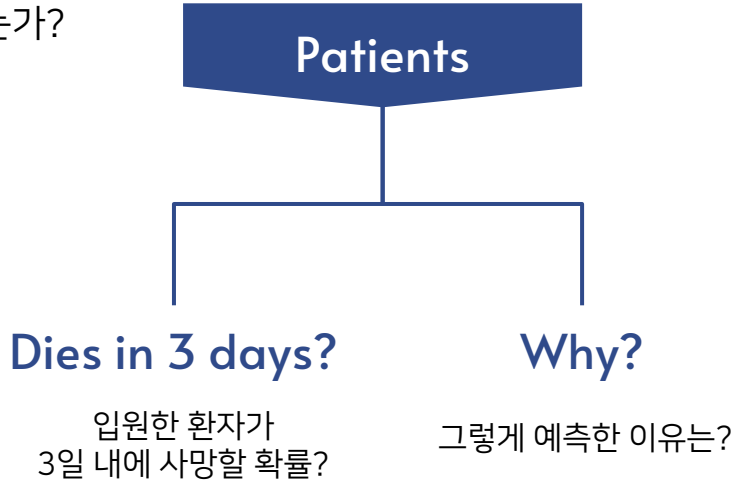
프로젝트 주제

프로젝트 소개 – Predicting ICU Mortality

ICU : Intensive Care Unit, 중환자실

머신러닝 기반 중환자 중증도 추정

- 중환자실에 입원한 환자가 3일 내 사망할 확률은 어떻게 되는가?
- 사망 영향 요인 식별



데이터 확보 과정

중환자실에 입원한 환자가 **3일 내 사망할 확률** 추정

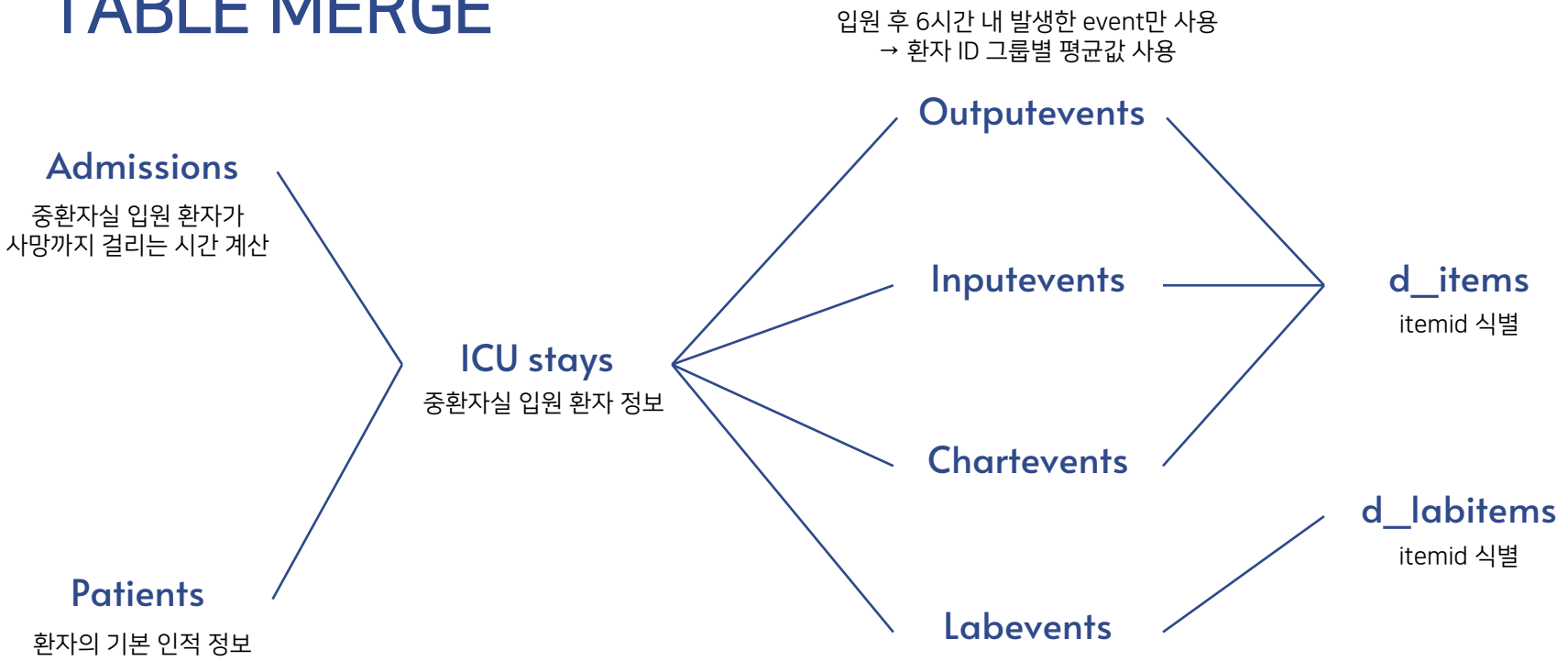
→ MIMIC-IV 데이터셋의 Chartevents, Inputevents, Outputevents, Labevents 데이터 사용

→ **입원 후 6시간 내에 발생한 이벤트**만을 고려

중환자실에 입원한 환자 한 명을 하나의 샘플(Row, Instance)로 생각하여 분류 모델 학습

→ 샘플의 특성(Column, Feature)은 6시간 내 발생한 이벤트들의 평균값

TABLE MERGE


Admissions
중환자실 입원 환자가
사망까지 걸리는 시간 계산
Patients
환자의 기본 인적 정보
ICU stays
중환자실 입원 환자 정보
입원 후 6시간 내 발생한 event만 사용
→ 환자 ID 그룹별 평균값 사용
Outputevents
Inputevents
Chartevents
Labevents
d_items
itemid 식별
d_labitems
itemid 식별

```
data = pd.read_csv('./DA_data.csv', index_col=0)
data
```

✓ 22.0s Python

	hadm_id	intime	gender	anchor_age	mortality_in_second	mortality_in_3days	Anderson (gastric)	Blakemore	Cath Lab	...	WBC_y	WBCApacheIIValue	WBCScore_ApacheIV	WBC_ApacheIV	Warming Device	Warming Device Status	WbcApacheIIScore	Yawning	Yawning (COWS)
0	24528534	2154-03-03 04:11:00	M	25	NaN	False	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	28960964	2150-06-19 17:57:00	M	42	NaN	False	NaN	NaN	NaN	...	13.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	27385897	2138-02-05 18:54:00	M	70	835560.0	False	NaN	NaN	NaN	...	17.8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	23483021	2123-10-25 10:35:00	M	87	NaN	False	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	20817525	2200-07-12 00:33:00	M	72	NaN	False	NaN	NaN	NaN	...	7.8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
69180	21944963	2152-08-01 17:53:56	F	75	NaN	False	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
69181	27299174	2126-06-13 01:00:00	F	47	NaN	False	NaN	NaN	NaN	...	8.8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
69182	28911582	2177-11-08 14:09:00	M	60	NaN	False	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
69183	22562812	2182-08-15 09:37:33	M	72	NaN	False	NaN	NaN	NaN	...	8.0	NaN	NaN	NaN	0.0	0.0	NaN	NaN	NaN
69184	22695803	2115-12-01 00:37:00	M	55	NaN	False	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

69185 rows × 2921 columns

69,185개의 샘플과 2,921개의 특성 확보

02.

PROJECT PIPELINE

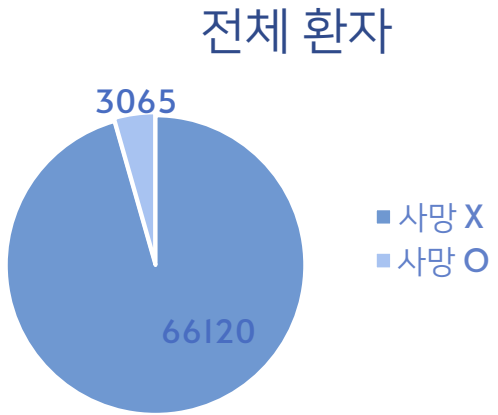
프로젝트 과정

Binary Classification with Class Imbalance

환자의 3일 내 사망 여부를 예측하는 **이진 분류 문제**

분류 문제에 적용할 수 있는 다음 5개의 모델 사용

- ResNet
- XGBoost
- LightGBM
- RandomForest
- LogisticRegression



추가로, 전체 중환자실 입원 환자 중 3일 내 사망한 환자의 수가 적은 **클래스 불균형 문제** 고려
→ 오버샘플링 사용 (sklearn.resample)

PIPELINE

중환자실 입원 데이터

SHAP 기반 특성 선택
결측치 대체 → 0
표준화
오버샘플링

모델 훈련
→ 각 모델의 예측 확률
soft voting

기본 하이퍼파라미터
조합 사용

새로운 환자 예측
→ 테스트 데이터 예측

SHAP 기반 모델 해석
→ 사망 예측 이유 해석

03.

RESULT

프로젝트 결과

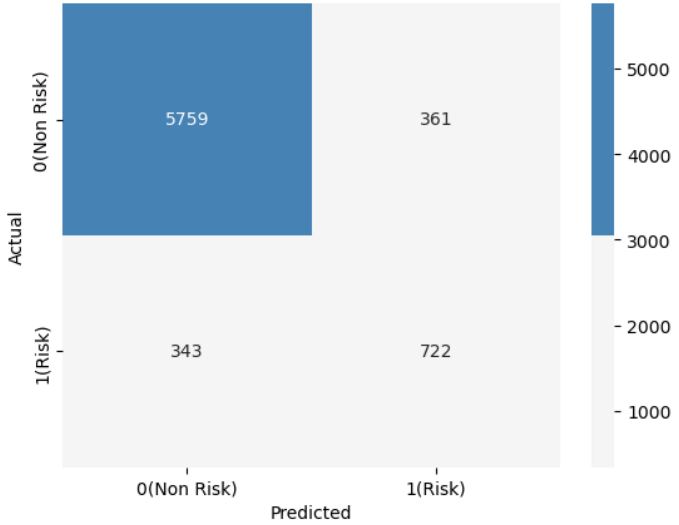
모델 훈련 결과

AUROC(AUC-ROC)를 사용하여 모델 예측 성능 평가

```
models = [xgb, lgbm, rf, lr, resnet]
model_name = ['XGBoost', 'LightGBM', 'RandomForest', 'LogisticRegression', 'ResNet']

for i, model in enumerate(models):
    print(model_name[i])
    model_auc_roc(model)
    print()
```

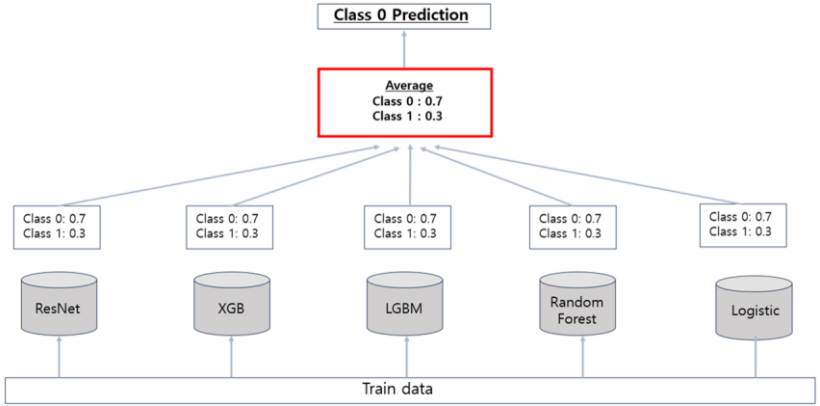
	XGBoost	LightGBM	Random Forest	Logisitc Regression	ResNet
AUROC (binary)	0.718	0.809	0.533	0.812	0.690
AUROC (probability)	0.896	0.922	0.908	0.898	0.848

<LGBMClassifier>




```
Accuracy: 0.9020180932498261
Recall: 0.6779342723004694
Precision: 0.6666666666666666
F1: 0.6722532588454375
```

모델 훈련 결과



모델 예측 확률(predict_proba)을 모두 더하여 soft voting 수행

```
xgb_preds_proba = xgb.predict_proba(X_test.values)[:, 1]
lgbm_preds_proba = lgbm.predict_proba(X_test.values)[:, 1]
rf_preds_proba = rf.predict_proba(X_test.values)[:, 1]
lr_preds_proba = lr.predict_proba(X_test.values)[:, 1]
resnet_preds_proba = torch.sigmoid(resnet(torch.tensor(X_test.values, dtype=torch.float))).detach().numpy().reshape(-1, )

preds_proba = xgb_preds_proba + lgbm_preds_proba + rf_preds_proba + resnet_preds_proba + lr_preds_proba
preds_proba = pd.Series(preds_proba)
```

preds_proba 값이 [0, 1] 사이에 분포하도록 정규화

```
# normalize into [0, 1]
preds_proba_normalize = preds_proba / 5
preds_proba_normalize.describe()
```

<preds_proba_normalize>

```
count    7185.000000
mean        0.152253
std         0.206912
min         0.000722
25%         0.017825
50%         0.061375
75%         0.193323
max         0.939831
dtype: float64
```

모델 훈련 결과

soft voting 수행 결과

```
print('Soft Voting AUROC (probability):', roc_auc_score(y_test, preds_proba_normalize))
```

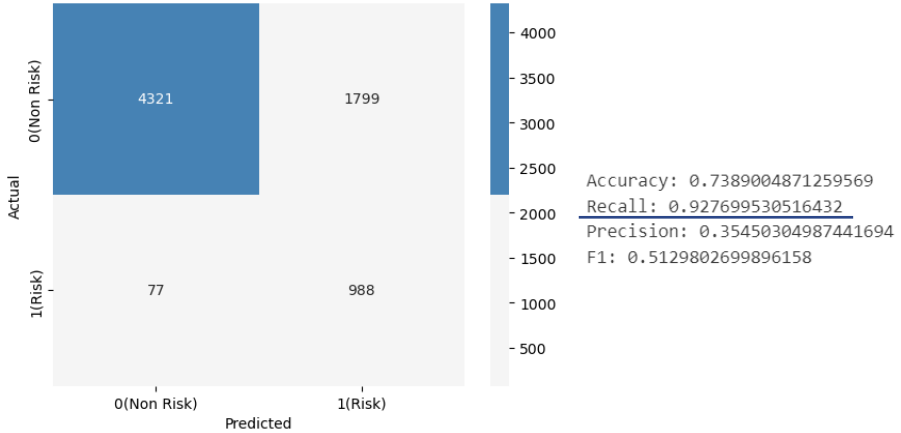
```
Soft Voting AUROC (probability): 0.9230151278038603
```

```
LightGBM
Model AUROC (binary): 0.8094736720979471
Model AUROC (probability): 0.9215344134523918
```

클래스 별 평균 예측 확률 평균

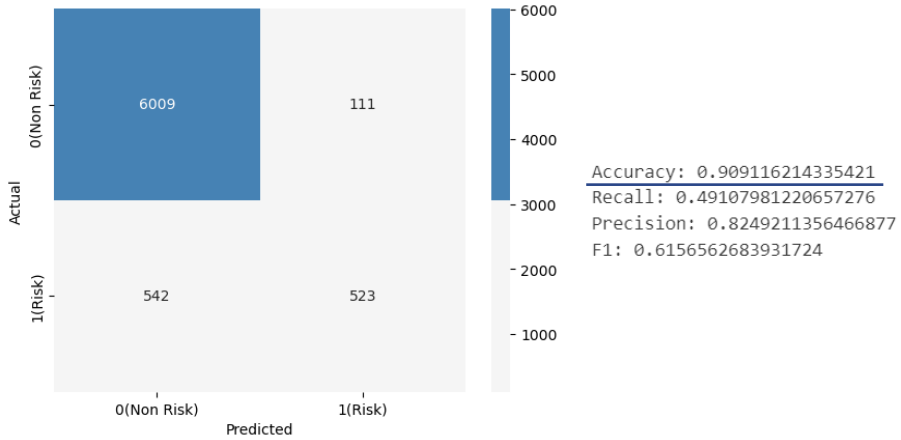
```
# probability groupby class label
preds_df = pd.concat([preds_proba_normalize, y_test.reset_index(drop=True)], axis=1)
preds_df.columns = ['Preds', 'Label']

preds_df.groupby('Label')['Preds'].describe()
```

Label	count	mean	std	min	25%	50%	75%	max
0	6120.0	0.093481	0.122832	0.000722	0.014640	0.044580	0.120526	0.871560
1	1065.0	0.489985	0.261264	0.001853	0.260562	0.490417	0.719647	0.939831



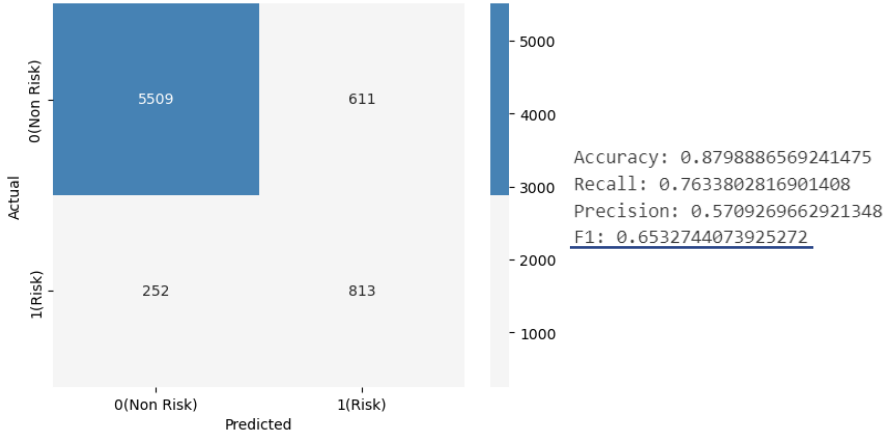
Accuracy: 0.7389004871259569
Recall: 0.927699530516432
Precision: 0.35450304987441694
F1: 0.5129802699896158

<Threshold = 0.1>



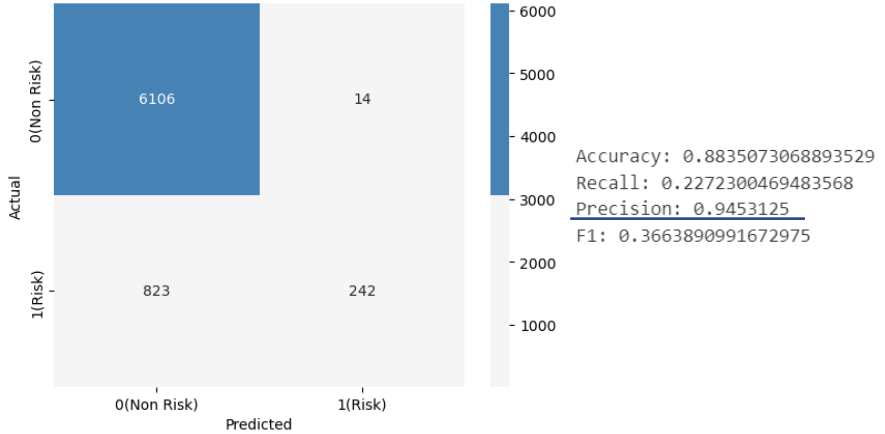
Accuracy: 0.909116214335421
Recall: 0.49107981220657276
Precision: 0.8249211356466877
F1: 0.6156562683931724

<Threshold = 0.5>



Accuracy: 0.8798886569241475
Recall: 0.7633802816901408
Precision: 0.5709269662921348
F1: 0.6532744073925272

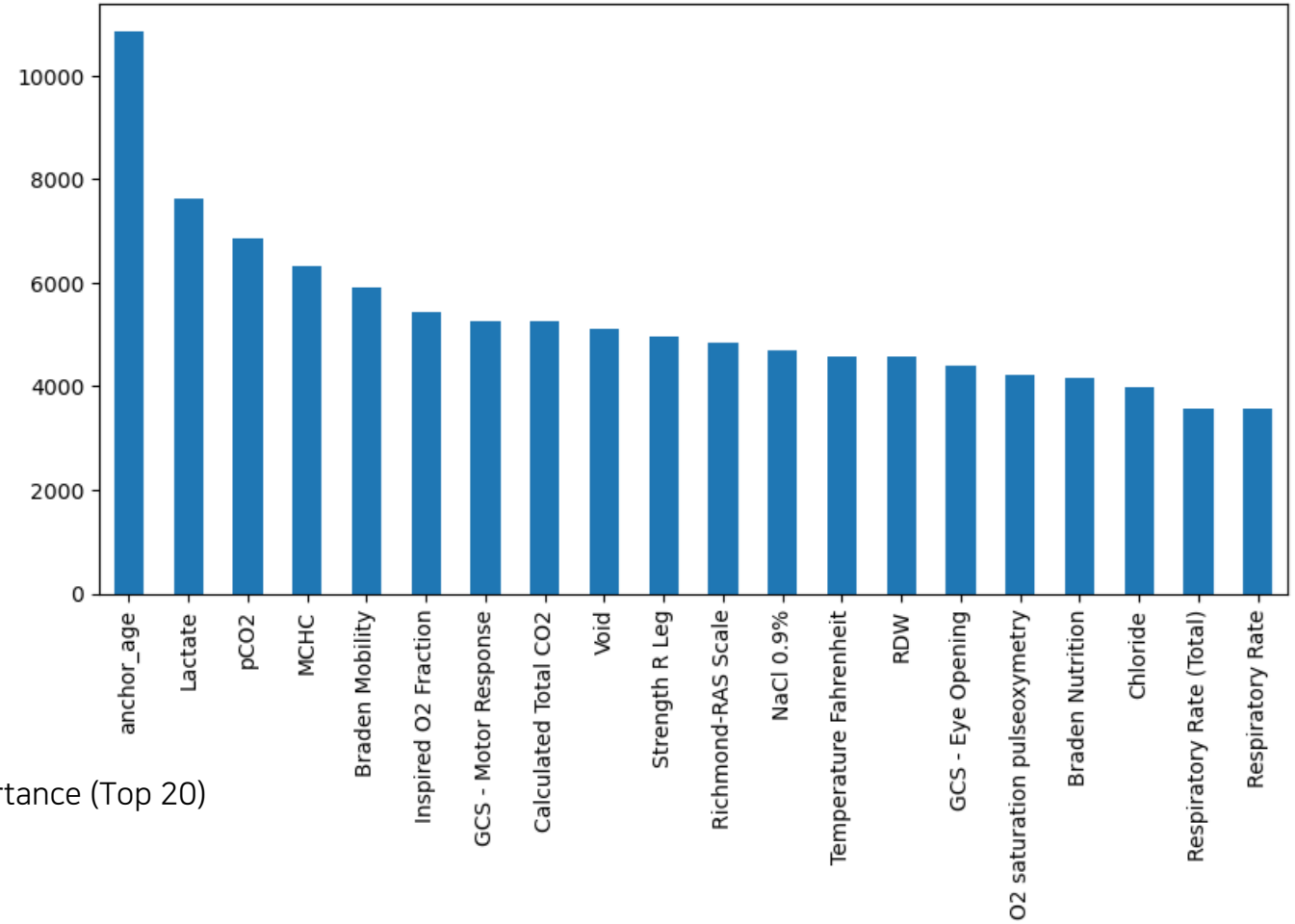
<Threshold = 0.25>



Accuracy: 0.8835073068893529
Recall: 0.2272300469483568
Precision: 0.9453125
F1: 0.3663890991672975

<Threshold = 0.75>

모델 해석 결과



SHAP 기반 Global Feature Importance (Top 20)

중증도 추정에 큰 영향을 준 특성

04.

APPLICATION / LIMITATION

적용 방안 및 한계

Mortality Prediction

Model Input

Choose a file (Support for Excel files)

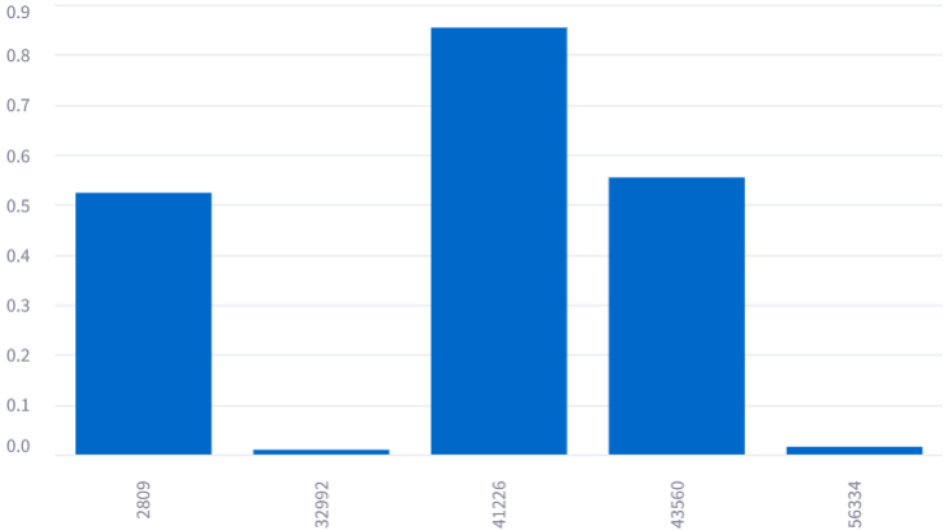
Drag and drop file here
Limit 200MB per file • CSV

Browse files

Web_test.csv 58.7KB

	mortality_in_3days	Model Predict
2,809	1	0.5237
32,992	0	0.0094
41,226	1	0.8542
43,560	0	0.5545
56,334	0	0.0154

Model Output



Streamlit 기반 웹페이지 구현

Mortality Interpretation

Model Input

Choose a file (Support for Excel files)

Drag and drop file here

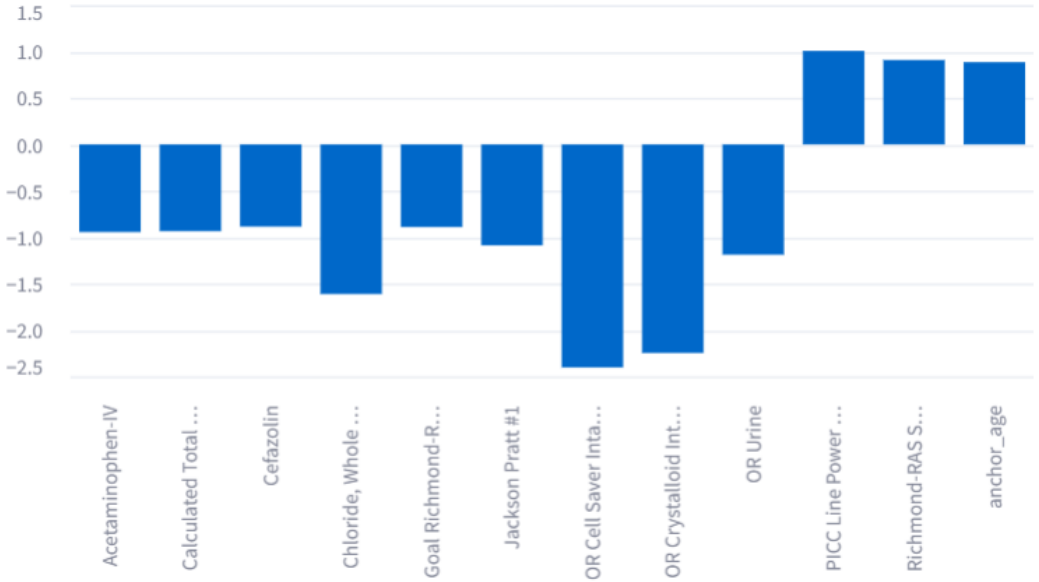
Limit 200MB per file • CSV

Browse files

Web_test_sample1.csv 19.1KB

	mortality_in_3days	Model Predict
56,334	0	0.0154

Model Output



SHAP 기반 local interpretation (Top12)

환자 56334의 예측 확률이 0.0154로 계산된 이유 설명

Mortality Interpretation

Model Input

Choose a file (Support for Excel files)

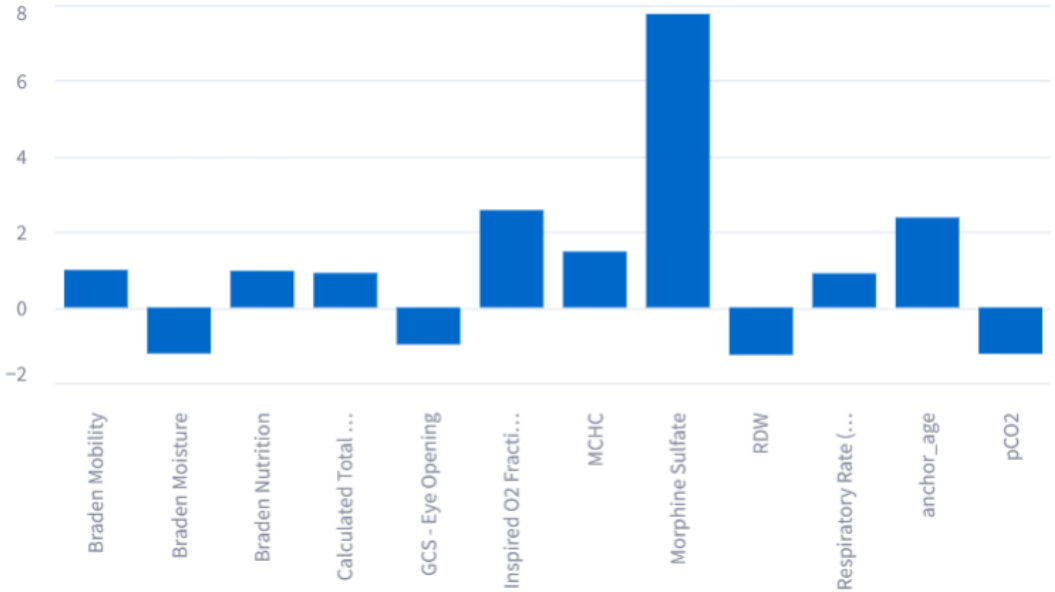
Drag and drop file here
Limit 200MB per file • CSV

Browse files

Web_test_sample2.csv 19.2KB

	mortality_in_3days	Model Predict
41,226	1	0.8542

Model Output



SHAP 기반 local interpretation (Top12)

환자 41226의 예측 확률이 0.8542로 계산된 이유 설명

→ global 뿐만 아니라 각 환자에 대한 정보를 확인함으로 사망 영향 요인을 식별하고 추가적인 처치 가능

프로젝트 적용 방안

정밀한 중증도 추정으로 실제 위중한 환자를 식별하고, 해당 환자에 대한 자원 집중

→ 의료서비스 품질 향상

특정 기간, 특정 지역에 대한 평균적인 중환자 중증도를 계산하여 적정 간호 인력 수요 산정

→ 효율적인 자원 배분

입원 후 12시간, 1일 내 사망 예측 등 전체 프로세스 조절 가능

→ 각 시스템 환경에 맞는 커스터마이징

프로젝트 한계

하드웨어 성능 한계

- 데이터의 전체 특성을 사용할 수 없었음
- 하이퍼파라미터 튜닝에 어려움이 있음

→ 모델 적합 수준의 한계가 존재하였음

임계값 설정

- 임계값 설정에 따라 위양성, 위음성 샘플의 수가 매우 달라짐

→ 위양성, 위음성 (정밀도, 재현율) 간의 트레이드오프를 고려하여 적절한 임계값을 찾는 방법 필요

Q&A

Q. 입원 데이터의 기준이 6시간인 이유가 궁금합니다. 3일 내 사망이라면 12시간 혹은 24시간이 더 적합하지 않을까 생각이 들었습니다.

A. 중환자실의 고질적인 문제 중 하나인 병상 부족을 고려해 보았을 때, 가능한 한 빨리 환자의 중증도를 추정하고 위중 정도를 판단하는 것이 중요하다고 판단하였습니다. 빠른 시간 내에 위중 정도를 판단하고 비교적 덜 위중한 환자는 일반 병실로 배치함으로써 중환자실 병상 부족 문제를 어느 정도 완화할 수 있을 것이라 생각하였습니다. 해당 부분을 고려하여 6시간을 기준으로 잡았습니다.

Q&A

Q. 중증도 하나만으로 위급한 환자들에게 적당한 치료기법을 적용하기에는 조금 무리가 있어 보이는데 이를 해결할 수 있는 방안이 있을까요?

A. 저희는 모델의 예측값(환자가 3일 내 사망할 확률 = 추정된 환자의 중증도)과 함께 SHAP 기반 local interpretation(feature importances)을 제공합니다. 예측값을 기반으로 해당 환자가 얼마나 위중한지 확인하며, local interpretation 기반으로 어떤 특성이 예측에 얼마나 영향을 주었는지 확인할 수 있습니다. 예를 들어 한 환자의 예측값이 0.8(사망 확률 높음)이고, 이때 가장 큰 영향을 주었던 특성이 Heart Rate라면, 해당 환자가 심장과 관련된 적절한 진단 및 치료를 받도록 조치할 수 있습니다.

Q&A

Q. 임계값에 따라 정확도가 달라진다고 했는데 어떤 영향(변수?)에 의해 달라지는 것인지?

A. 임계값이 변함에 따라 정확도 및 다른 지표(recall, precision, f1)가 달라지는 것은 임계값에 따라 분류 결과가 달라지기 때문입니다. 예를 들어 임계값이 0.5인 경우를 생각해 보면, 모델의 예측값(범위: [0, 1])이 0.5 보다 큰 경우 양성 샘플(3일 내 사망으로 예측)로 분류됩니다. 만약 임계값이 0.1로 줄게 되면, 각 샘플의 예측값이 0.1 보다 클 때 양성 샘플로 분류됩니다. 임계값이 0.1로 작아질 때, 임계값이 0.5일 때는 음성이었던 샘플이 양성인 샘플로 분류될 수 있습니다(예측값이 (0.1, 0.5]에 속하는 경우). 이렇게 임계값이 변함에 따라 분류 결과가 달라지게 되고, 이에 따라 위음성 및 위양성 샘플의 수가 달라져 정확도 등의 지표가 변할 수 있습니다.

APPENDIX

TABLE MERGE

Outputevents

hadm_id : 환자 입원 식별 id
storetime : 데이터 저장 시각
itemid : Event 식별 id
value : Event 측정 값

6시간 이내의 데이터 추출

```
outputevents['intime'] = pd.to_datetime(outputevents['intime'])
outputevents['storetime'] = pd.to_datetime(outputevents['storetime'])

outputevents['time_to_store'] = outputevents['storetime'] - outputevents['intime']
outputevents['time_to_store'] = outputevents['time_to_store'].dt.total_seconds()

# 6시간 이내의 데이터
outputevents['time_to_store_in_day'] = (outputevents['time_to_store'] < 86400 / 4) & (outputevents['time_to_store'] > 0)
```

itemid 식별

```
outputevents_in_6hour = outputevents[outputevents.time_to_store_in_day]
outputevents_in_6hour = pd.merge(outputevents_in_6hour, d_items[['itemid', 'label']], on=['itemid'], how='left')
```

hadm_id로 group, mean 및 pivot

```
# 6시간 내 value의 평균을 사용
tmp = outputevents_in_6hour.groupby(['hadm_id', 'label'])['value'].mean()
tmp = pd.DataFrame(tmp).reset_index()

outputevents_in_6hour_pivot = tmp.pivot(index='hadm_id', columns='label', values='value')
outputevents_in_6hour_pivot = outputevents_in_6hour_pivot.reset_index()
outputevents_in_6hour_pivot.to_csv('outputevents_in_row_mean.csv')
```

입원 후 6시간 이내의 데이터

Events

ID	storetime	itemid	valuenum
1	19:00:35	3	60
1	19:01:35	1	5
2	19:01:48	2	10
3	19:05:20	3	58
1	19:10:05	7	0.82
1	19:11:27	8	0.02
4	19:21:35	3	630
1	19:22:00	1	7
2	19:22:11	3	88
1	19:24:39	2	32
1	19:33:12	3	420
5	19:48:11	5	600
…			
1	23:01:11	9	100
2	23:01:23	9	110
1	23:02:04	3	610
3	23:03:21	1	9
1	23:04:47	2	11

d_items
itemid 식별

1 : 동공 반사 정도
2 : 혈압 변동
3 : 심박수
…

groupby('ID') / Mean / Pivot

Pivot Table

ID	itemid_1	itemid_2	itemid_3	itemid_4	itemit_5	…
1	4.8	Null	77	0.77	610	
2	4.2	11	80	0.12	Null	
3	4.9	19	91	0.31	590	
4	Null	18	50	0.14	614	
5	5.2	10	61	0.58	615	
…						

6시간 내 해당 이벤트가 발생하지 않으면 Null
→ 이후 0으로 대체

THANKS

2023 Fall - Data Analytics

정보융합학부 김주혁 장원재