비정형 데이터 분석 기말 레포트

20171481 박은총

목표 : 1. mimic ii의 데이터를 가지고 패혈증을 예측하기

2. 예측을 머신러닝, 딥러닝으로 해서 비교하기

사용데이터 목록

chartevents	진료차트
Icd9	질병 코드
admissions	환자 입원정보
D_codeitems	아이템 코드
drgevents	drg정보
patients	환자정보
Icustay_days	입.퇴원 일자
Demographic_detail	인구통계_상세

📌기준 테이블 정의 및 학습에 필요한 변수 생성

제일 먼저 기준 테이블을 잡고 그 테이블에 계속 변수를 추가하는 방식으로 진행하였습니다.

기준 테이블이 필요할 것 같아 amissions와 patients를 merge한 pa_am테이블을 생성하고 이것을 기준으로 삼았습니다.
Admissions에 최초 입원 기간, 마지막 입원 기간을 활용해서 환자들의 나이를 유추 할 수 있었습니다.

	hadm_id	subject_id	admit_dt	disch_dt	sex	dob	dod	is_dead	first_admit_age	life_span	Age_band
0	2	24807	3033-07-08 00:00:00	3033-07-17 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.63		adult
1	1223	24807	3033-06-13 00:00:00	3033-06-17 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.56		adult
2	16385	24807	3033-06-25 00:00:00	3033-06-28 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.59		adult
3	17810	24807	3033-09-02 00:00:00	3033-09-17 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.78		adult
4	17898	24807	3033-08-24 00:00:00	3033-08-30 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.76		adult
...	...	...	...	...	...	...	...	...	...	...	...
5069	36005	29309	2541-12-11 00:00:00	2541-12-20 00:00:00	M	2457-12-11 00:00:00	2541-12-20 00:00:00	Y	84.00	84.02	adult
5070	36069	32711	3143-05-20 00:00:00	3143-05-22 00:00:00	F	3057-12-21 00:00:00	3143-05-22 00:00:00	Y	85.41	85.42	adult
5071	36071	32667	2866-02-18 00:00:00	2866-02-27 00:00:00	M	2778-01-19 00:00:00	2866-02-27 00:00:00	Y	88.08	88.11	adult
5072	36077	31134	2724-04-24 00:00:00	2724-04-27 00:00:00	F	2632-08-08 00:00:00	2724-04-27 00:00:00	Y	91.71	91.72	>89
5073	36103	28191	2675-04-27 00:00:00	2675-04-28 00:00:00	F	2618-02-08 00:00:00	2675-04-28 00:00:00	Y	57.21	57.22	adult

5074 rows × 11 columns

Icd9코드를 이용해서 환자들이 패혈증인지 아닌지를 판단 할 수 있는 변수를 생성했습니다.(0/1 이진분류)

그 다음에 환자가 입원한 병실에서 얼마나 머물렀는지 확인 할 수 있는 icustay_diff라는 변수를 만들어서 입원기간을 확인 할 수 있게 되었습니다.

	hadm_id	subject_id	admit_dt	disch_dt	sex	dob	dod	is_dead	first_admit_age	life_span	Age_band	sequence	code	description	issepis	icustay_id	icustay_diff
0	2	24807	3033-07-08 00:00:00	3033-07-17 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.63		adult	NaN	NaN	NaN	0.0	30800	1
1	1223	24807	3033-06-13 00:00:00	3033-06-17 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.56		adult	NaN	NaN	NaN	0.0	30798	3
2	16385	24807	3033-06-25 00:00:00	3033-06-28 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.59		adult	NaN	NaN	NaN	0.0	30799	0
3	17810	24807	3033-09-02 00:00:00	3033-09-17 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.78		adult	NaN	NaN	NaN	0.0	30804	2
4	17898	24807	3033-08-24 00:00:00	3033-08-30 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.76		adult	NaN	NaN	NaN	0.0	30802	5
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5380	36005	29309	2541-12-11 00:00:00	2541-12-20 00:00:00	M	2457-12-11 00:00:00	2541-12-20 00:00:00	Y	84.00	84.02	adult	NaN	NaN	NaN	0.0	43679	9
5381	36069	32711	3143-05-20 00:00:00	3143-05-22 00:00:00	F	3057-12-21 00:00:00	3143-05-22 00:00:00	Y	85.41	85.42	adult	NaN	NaN	NaN	0.0	47421	0
5382	36071	32667	2866-02-18 00:00:00	2866-02-27 00:00:00	M	2778-01-19 00:00:00	2866-02-27 00:00:00	Y	88.08	88.11	adult	NaN	NaN	NaN	0.0	47373	2
5383	36077	31134	2724-04-24 00:00:00	2724-04-27 00:00:00	F	2632-08-08 00:00:00	2724-04-27 00:00:00	Y	91.71	91.72	>89	NaN	NaN	NaN	0.0	45690	1
5384	36103	28191	2675-04-27 00:00:00	2675-04-28 00:00:00	F	2618-02-08 00:00:00	2675-04-28 00:00:00	Y	57.21	57.22	adult	NaN	NaN	NaN	0.0	42406	0

5385 rows × 17 columns

📌통계분석

본격적인 분석에 들어가는 파트입니다.

```
# 맥박수
search_detail(d_chartitems, 'pulse')
# 81, 188, 209, 546, 603, 1332, 1341, 1438, 1456, 1725, 1740, 1909, 2406, 2529,2530,2702,2858,3097,3124,6160,7090,7243,7925,8144
print("--"*100)

# 이완기
search_detail(d_chartitems, 'diastolic')
# 153

print("--"*100)

# 체온 관련인데,,,
search_detail(d_chartitems, 'temperature')
# 676, 677, 678, 679
```

교수님께서 공유해주신 노션에 어떤 변수를 주로 분석했는지 나와서 그대로 찾아서 통계 변수로 추출 해보려고 합니다.

총 4번을 추출 하려고 하는데 1. 호흡률, 2. 혈압, 3. 맥박수, 4. 체온 부분으로 mean, max, min, std를 환자별로 추출했습니다.(이것이 없는 환자들이 많은데 이것들은 null값으로 두었습니다.) 추출한 다음 기존의 pa_am이랑 합쳤습니다.

	hadm_id	subject_id	admit_dt	disch_dt	sex	dob	dod	is_dead	first_admit_age	life_span	...	con2_min	con2_std	con3_mean	con3_max	con3_min	con3_std	con4_mean	con4_max	con4_min	con4_std
0	2	24807	3033-07-08 00:00:00	3033-07-17 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.63		...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1223	24807	3033-06-13 00:00:00	3033-06-17 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.56		...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	16385	24807	3033-06-25 00:00:00	3033-06-28 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.59		...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	17810	24807	3033-09-02 00:00:00	3033-09-17 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.78		...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	17898	24807	3033-08-24 00:00:00	3033-08-30 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.76		...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5380	36005	29309	2541-12-11 00:00:00	2541-12-20 00:00:00	M	2457-12-11 00:00:00	2541-12-20 00:00:00	Y	84.00	84.02	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5381	36069	32711	3143-05-20 00:00:00	3143-05-22 00:00:00	F	3057-12-21 00:00:00	3143-05-22 00:00:00	Y	85.41	85.42	...	71.0	9.389028	NaN	NaN	NaN	NaN	97.150002	97.500000	96.800003	0.494973
5382	36071	32667	2866-02-18 00:00:00	2866-02-27 00:00:00	M	2778-01-19 00:00:00	2866-02-27 00:00:00	Y	88.08	88.11	...	-1.0	39.468162	NaN	NaN	NaN	NaN	97.614287	98.900002	96.800003	0.779804
5383	36077	31134	2724-04-24 00:00:00	2724-04-27 00:00:00	F	2632-08-08 00:00:00	2724-04-27 00:00:00	Y	91.71	91.72	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5384	36103	28191	2675-04-27 00:00:00	2675-04-28 00:00:00	F	2618-02-08 00:00:00	2675-04-28 00:00:00	Y	57.21	57.22	...	-1.0	39.999892	NaN	NaN	NaN	NaN	99.099999	101.000000	94.900002	1.916376

5385 rows × 33 columns

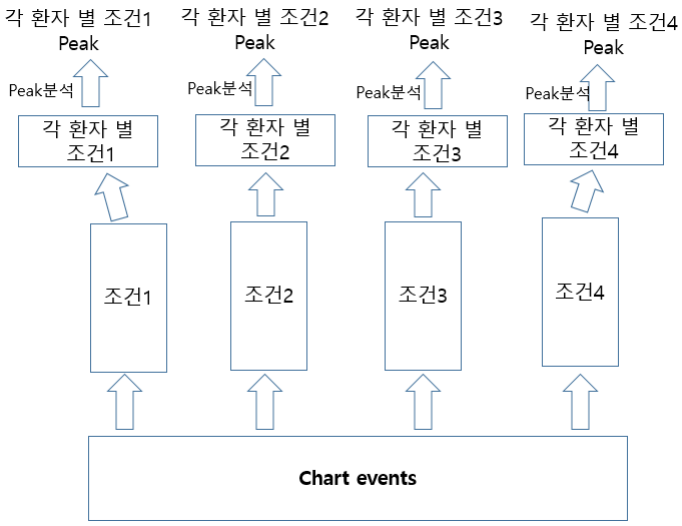
Mean, max, min, std를 모두 추출 했을 때는 변수의 개수가 33개로 늘어났습니다.

📌피크분석

위에서 호흡률, 혈압, 맥박수, 체온의 통계부분을 추출했다면 여기서는 이 네가지의 피크를 추출하려고 합니다.

파이썬에서는 scipy에 find_peaks라는 함수가 있으므로 불러와서 사용했습니다.

chartevents에서 조건1(호흡률), 조건2(혈압), 조건3(맥박수), 조건4(체온) 환자별로 value값(체온값or 혈압값 or 맥박수값 or 체온값)을 추출해서 이것들의 피크 개수를 추출했습니다.



도식화를 하면 이런 모습입니다.

이것도 조건마다 진행을 하게 되었고 변수의 개수는 4개 늘어나서 총 37개가 되었습니다.

	hadm_id	subject_id	admit_dt	disch_dt	sex	dob	dod	is_dead	first_admit_age	life_span	...	con3_min	con3_std	con4_mean	con4_max	con4_min	con4_std	con1_peaks	con2_peaks	con3_peaks	con4_peaks
0	2	24807	3033-07-08 00:00:00	3033-07-17 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.63		...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1223	24807	3033-06-13 00:00:00	3033-06-17 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.56		...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	16385	24807	3033-06-25 00:00:00	3033-06-28 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.59		...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	17810	24807	3033-09-02 00:00:00	3033-09-17 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.78		...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	17898	24807	3033-08-24 00:00:00	3033-08-30 00:00:00	F	2992-11-20 00:00:00	3033-09-23 00:00:00	N	40.76		...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5438	36005	29309	2541-12-11 00:00:00	2541-12-20 00:00:00	M	2457-12-11 00:00:00	2541-12-20 00:00:00	Y	84.00	84.02	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5439	36069	32711	3143-05-20 00:00:00	3143-05-22 00:00:00	F	3057-12-21 00:00:00	3143-05-22 00:00:00	Y	85.41	85.42	...	NaN	NaN	97.150002	97.500000	96.800003	0.494973	NaN	NaN	NaN	NaN
5440	36071	32667	2866-02-18 00:00:00	2866-02-27 00:00:00	M	2778-01-19 00:00:00	2866-02-27 00:00:00	Y	88.08	88.11	...	NaN	NaN	97.614287	98.900002	96.800003	0.779804	NaN	NaN	NaN	NaN
5441	36077	31134	2724-04-24 00:00:00	2724-04-27 00:00:00	F	2632-08-08 00:00:00	2724-04-27 00:00:00	Y	91.71	91.72	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
5442	36103	28191	2675-04-27 00:00:00	2675-04-28 00:00:00	F	2618-02-08 00:00:00	2675-04-28 00:00:00	Y	57.21	57.22	...	NaN	NaN	99.099999	101.000000	94.900002	1.916376	NaN	NaN	NaN	NaN

5443 rows × 37 columns

추가적인 분석을 하면서 넣지 못했던 변수들을 추가

➔ drg, demographic_detail

📌ML모델링

모델링 하기에 앞서 통계나, 피크를 결합하면서 계속 NULL값이 누적되었습니다. 이것들은 수치상 없는 값이기 때문에 -1로 처리했습니다.(-1은 현재 수집된 데이터에서 없는 값이다.)

target값을 issepis로 정했고 categorical데이터는 다 수치형으로 변경해주었습니다.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5433 entries, 0 to 5432
Data columns (total 28 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   sex                     5433 non-null   int64
 1   first_admit_age         5433 non-null   float64
 2   is_dead                 5433 non-null   int64
 3   icustay_diff            5433 non-null   int64
 4   con1_mean               5433 non-null   float64
 5   con1_max                5433 non-null   float64
 6   con1_min                5433 non-null   float64
 7   con1_std                5433 non-null   float64
 8   con2_mean               5433 non-null   float64
 9   con2_max                5433 non-null   float64
 10  con2_min                5433 non-null   float64
 11  con2_std                5433 non-null   float64
 12  con3_mean               5433 non-null   float64
 13  con3_max                5433 non-null   float64
 14  con3_min                5433 non-null   float64
 15  con3_std                5433 non-null   float64
 16  con4_mean               5433 non-null   float64
 17  con4_max                5433 non-null   float64
 18  con4_min                5433 non-null   float64
 19  con4_std                5433 non-null   float64
 20  con1_peaks              5433 non-null   float64
 21  con2_peaks              5433 non-null   float64
 22  con3_peaks              5433 non-null   float64
 23  con4_peaks              5433 non-null   float64
 24  cost_weight             5433 non-null   float64
 25  itemid                  5433 non-null   int64
 26  admission_source_itemid  5433 non-null   int64
 27  ethnicity_itemid        5433 non-null   int64
dtypes: float64(22), int64(6)
memory usage: 1.2 MB
```

제가 사용한 변수이고 이것으로 모델링을 진행했습니다.

Sklearn에 있는 train_test_split을 사용해서 train과 test비율을 8:2로 맞추었습니다. (stratify=True) 하지만 전체 환자중에서 패혈증을 가지고 있는 사람이 많이 없었기 때문에 클래스 불균형이 있습니다. 그래서 SMOTE 알고리즘을 사용하여 Oversampling해줘서 train의 클래스 불균형을 맞춰주었습니다.

```
[26] print('x_train_over shape', x_train_over.shape)
     print('y_trin_over shape', y_train_over.shape)

     x_train_over shape (7486, 28)
     y_trin_over shape (7486,)

[27] y_train_over.describe()
     # oversampleing이 잘 된듯하다.

     count    7486.000000
     mean        0.500000
     std         0.500033
     min         0.000000
     25%         0.000000
     50%         0.500000
     75%         1.000000
     max         1.000000
     Name: issepis, dtype: float64
```

데이터 개수는 4346->7486로 늘어났고 이진분류이므로 평균 0.5이면 1과 0의 비율이 같습니다.

ML모델은 LGBM Classifier를 사용했습니다. 우선 아무것도 안하고 알고리즘만 적용했을 때의 K-Fold Cross Validation은 K=5기준으로 약 90.4%가 나왔습니다.

이것을 그대로 Test 데이터에 넣었더니 약 90.7%의 성능이 나왔습니다.,

```
[28] import lightgbm as lgb
     from sklearn.model_selection import cross_val_score

     lgbm = lgb.LGBMClassifier(random_state = 123456789)

     scores = cross_val_score(lgbm, x_train, y_train, cv = 5)
     print('K-Fold Average Accuarcy :',np.mean(scores))

     K-Fold Average Accuarcy : 0.9042813115881646
```

```
[29] from sklearn.metrics import confusion_matrix, accuracy_score

     clf = lgbm.fit(x_train, y_train)
     pred = clf.predict(x_test)

     print('Accuarcy :', accuracy_score(y_test, pred))
     print('*' * 150)
     print('₩n','Confusion matrix','₩n',confusion_matrix(y_test, pred))

     Accuarcy : 0.9070837166513339
     ******************************************************************************************************************************************************

      Confusion matrix
      [[905  31]
      [ 70  81]]
```

언듯보면 잘나온 것 같지만 confusion matrix를 보면 1부분 즉, 패혈증 151개중 81개만 맞췄습니다. 성능이 그렇게 좋지 못하다는 뜻으로 해석이 됩니다. (불균형이 심한 데이터여서 기본적으로 정확도는 잘 나옵니다.)

다음으로는 optuna를 이용해서 hyper parameter tuning을 진행했습니다.
** optuna란 최적의 하이퍼 파라메터를 찾아주는 알고리즘으로 하이퍼 파라메터들의 분포(log scale, linear scale 등)를 정해서 가져올 수 있습니다.

```
[31] trial = study_lgb.best_trial
     trial_params = trial.params
     print('Best Trial: score {},₩nparams {}'.format(trial.value, trial_params))

     Best Trial: score 0.9162833486660533,
     params {'max_depth': 13, 'learning_rate': 0.009863854510035084, 'n_estimators': 2516, 'min_child_samples': 19, 'subsample': 0.5521347742722199}
```

```
[32] lgbm = lgb.LGBMClassifier(**trial_params)

     lgbm.fit(x_train, y_train)
     pred = lgbm.predict(x_test)
     print('Accuarcy :', accuracy_score(y_test, pred))
     print('*' * 150)
     print('₩n','Confusion matrix','₩n',confusion_matrix(y_test, pred))

     Accuarcy : 0.9162833486660533
     ******************************************************************************************************************************************************

      Confusion matrix
      [[898  38]
      [ 53  98]]
```

Optuna로 최적의 하이퍼 파라메터를 찾을 수 있었고 최적의 하이퍼 파라메터 조합으로 다시 테스트 해봤더니 약 91.6%의 성능과 함께 패혈증 부분을 조금 더 잘 맞추게 되었습니다.

패혈증 부분은 81->98개로 약간의 성능 향상을 보여줬습니다.

📌DL모델링

딥러닝 모델링을 하려고 생 데이터(raw data)로 구성하려고 했지만 환자들마다 진단 받은 횟수나 투여받은 약물의 횟수가 일정하지 않습니다..(학습하려면 각 환자에 대해서 동일한 shape가 나와야한다.) 그래서 위 머신러닝에서 썼던 데이터를 그대로 활용해서 딥러닝을 구축해보겠습니다.

딥러닝은 모델링할 때 트리 기반 모형과 다르게 Scaling을 해줘야 학습에 좋습니다.
그래서 MinMax Scaling을 진행했습니다.

```
Model: "model_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_4 (InputLayer)         [(None, 28)]              0
_________________________________________________________________
dense_12 (Dense)             (None, 128)               3712
_________________________________________________________________
dense_13 (Dense)             (None, 64)                8256
_________________________________________________________________
dense_14 (Dense)             (None, 32)                2080
_________________________________________________________________
dense_15 (Dense)             (None, 1)                 33
=================================================================
Total params: 14,081
Trainable params: 14,081
Non-trainable params: 0
_________________________________________________________________
```

Hidden Layer 3개인 DNN모델을 만들었고 이진 분류이므로

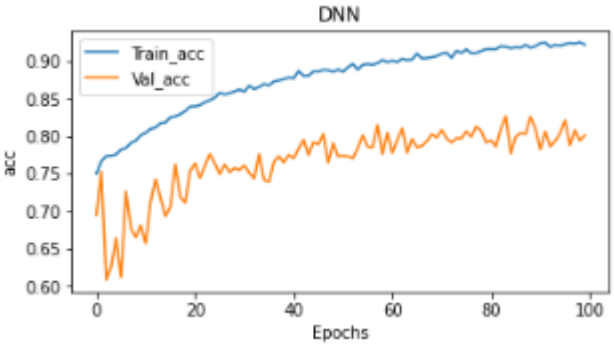
Loss는 'binary_crossentropy'를 사용,

Optimaizer는 'Adam',

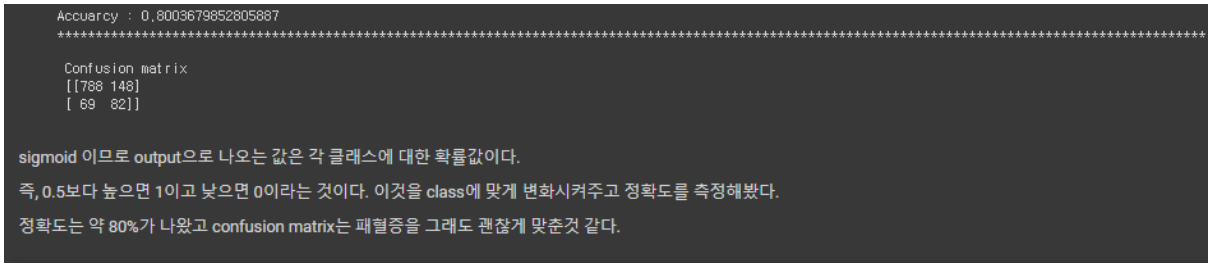
Metrics은 'Acc'사용

Hideen Layer1 의 units = 128, 2의 units = 64, 3의 units = 32

위 사진처럼 Hidden Layer 3개의 DNN모델을 만들었고 각각의 파라메터를 지정해줬습니다.
Loss : binary_crossentropy,
Optimaizer : 'Adam',
Metrics : 'Acc'
Hideen Layer1 의 units = 128, 2의 units = 64, 3의 units = 32
epochs : 100으로 설정해서 학습을 진행했습니다.



Train의 정확도는 올라가는데 validation의 정확도는 증가하지 않고 있습니다. 여기서는 보이지않지만 val_loss는 계속해서 올라가는 것을 확안 할 수 있었습니다.
** 아마도 데이터 구성이 학습을 하기에 적절한 구성이 아니라서 그런 것 같습니다.

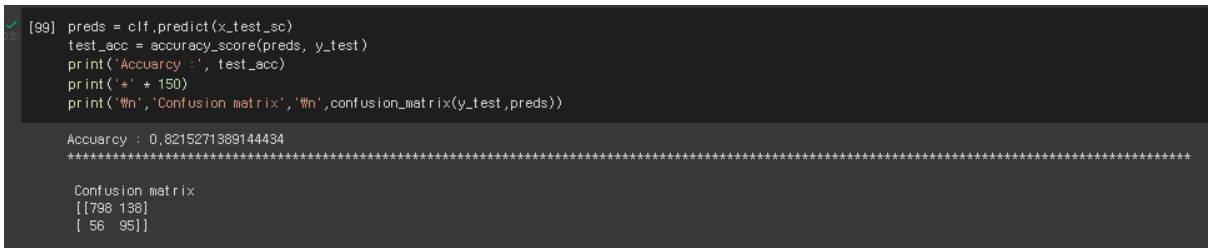


정확도는 약 80%정도 나왔고 confusion matrix를 보니까 생각보다 패혈증을 괜찮게 맞춘느낌이 드는데 성능이 낮긴 합니다.

TabNet

TabNet으로 해당데이터를 다시 학습을 시도했습니다.(이것도 데이터의 문제로 성능이 안나올 것 같지만 시도)

** TabNet은 이미지나 텍스트, 음성과 같이 비정형데이터 비해 성능이 상대적으로 낮은 Taular데이터를 뉴럴 네트워크에 특화시켜서 나온 알고리즘(Google Research 2019)



TabNet 역시 크게 성능이 좋지는 않았지만 패혈증 부분은 위에 ML부분의 LGBM과 비슷하게 나타났습니다.

데이터 셋을 어떻게 구성하느냐에 따라 성능이 많이 달라질 듯 합니다.

PS. 딥러닝 모델링 할 때 데이터 구성을 어떻게 해야할지 조언 부탁드립니다.(제출한 Ipynb에도 썼지만 raw데이터로 진행하려고 하는데 쉽게 되지않아서요 ㅎㅎ...)