

2021

파이썬, R을 활용한 빅데이터 시각화 구현

# CAVITIES<sup>+</sup>

(충치)

# INDEX



개요



데이터 수집



데이터 전처리



모델 구성 & 학습



결과

# 개요

주제 및 목적

일정관리

개발환경

목표설정

# 주제 & 목적

SUBJECT & PURPOSE

Subject :

건강검진 후, **충치 유무**(有無)를  
알려주는 친절한 **AI**

Purpose :

**AI**, 충치있는 사람은 **구강검진**을  
필수항목으로 변경해줘



		 계획	 완료	8/30		8/31		9/1		9/2		9/3	
기획 및 설계	선정 및 요구사항, 일정관리												
	설계, 기능정의, 데이터수집												
서론 및 이론적 배경	연구 배경 및 목적												
	연구 내용 및 방법												
	프로젝트의 구성												
	이론적 배경												
연구 모형 및 조사 설계	연구모형 및 연구가설												
	변수의 조작적 정의 및 측정 항목												
실증 분석 및 결론	자료 수집 및 표본의 특성												
	측정항목의 기술통계분석 및 시각화												
	신뢰도 및 타당성 검증 및 시각화												
	각자에 맞는 데이터분석 및 시각화												
	추가 분석 및 시각화												
테스트 및 수정													

**OS**

Windows 10 Pro

**Language**

Python 3.8.8

**IDE**

Jupyter Notebook 6.3.0

**Open Source**

Tensorflow, pycaret



## 첫 번째.

필수사항인 건강검진을 통해,  
AI가 충치 유무를 예측



## 두 번째.

AI의 예측에 따라,  
구강검진을 필수사항으로 변경



# 데이터 수집

자료 출처

자료 수집





## 검진정보

공공데이터포털 (다운로드)



## 내 용

국민건강보험공단  
건강검진대상자 100만명 (2019년)

# CAVITIES

데이터 세계 | 공공데이터포털

국립중앙도서관 | 데이터 찾기 | 국가데이터맵 | 데이터요청 | 데이터활용 | 정보공유 | 이용안내

## 데이터 세계

국민건강보험공단 건강검진정보

건강검진정보란 국민건강보험의 직장가입자와 40세 이상의 피부양자, 세대주인 지역가입자와 40세 이상의 지역가입자의 일반건강검진 결과와 이를 일반건강검진 대상자 중에 만40세와 만66세에 도달한 이월이 받게 되는 생애전환기건강검진 수검이력이 있는 각 연도별 수검자 100만 명에 대한 기본정보(성, 연령대, 시도코드 등)와 검진내역(신장, 체중, 총콜레스테롤, 혈색소 등)으로 구성된 개방데이터입니다.

파일데이터 | 오픈API

공공데이터활용지원센터는 공공데이터포털에 개방되는 3단계 이상의 오픈 포맷 파일데이터를 오픈 API(RestAPI 기반의 JSON/XML)로 자동변환하여 제공합니다. 오픈 API를 활용하기 위해서는 공공데이터포털 회원 가입 및 활용신청이 필요하며, 활용 관련 문의는 공공데이터활용지원센터로 연락주시기 바랍니다. 파일데이터는 로그인 없이 다운로드를 통해 이용하실 수 있습니다.

CSV | 국민건강보험공단\_건강검진정보

다ownload | 다운로드 | 데이터 열람

파일데이터 정보 | 데이터 열람

파일데이터명	국민건강보험공단_건강검진정보_20191231	제공기관	국민건강보험공단
분류체계	보건 - 보건의료	관리부서 전화번호	033-736-3444
관리부서명	빅데이터운영실 데이터관리팀	수집방법	2021-12-31
보유근거	공공데이터의 제공 및 이용 활성화에 관한 법률	차기 등록 예정일	1
업데이트 주기	연간	전체 행	53845
매체유형	텍스트	다운로드(바로그가)	키워드
원장자	CSV	수정	2021-01-19
데이터 한계	2021-01-06	다운로드(바로그가)	53845
등록	국공데이터포털에서 다운로드(원문파일용)		
제공형태			

```
1 raw_data = pd.read_csv("data/국민건강보험공단_건강검진정보_20191231.csv", encoding='cp949')
2 raw_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000000 entries, 0 to 999999
Data columns (total 34 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   기준년도                             1000000 non-null  int64
1   가입자 일련번호                       1000000 non-null  int64
2   시도코드                             1000000 non-null  int64
3   성별코드                             1000000 non-null  int64
4   연령대 코드(5세단위)                 1000000 non-null  int64
5   신장(50cm단위)                       1000000 non-null  int64
6   체중(5kg 단위)                       1000000 non-null  int64
7   허리둘레                             999597 non-null   float64
8   시력(좌)                             999805 non-null   float64
9   시력(우)                             999812 non-null   float64
10  청력(좌)                             999819 non-null   float64
11  청력(우)                             999822 non-null   float64
12  수축기 혈압                          994576 non-null   float64
13  이완기 혈압                          994575 non-null   float64
14  식전혈당(공복혈당)                   994477 non-null   float64
15  총 콜레스테롤                        333549 non-null   float64
16  트리글리세라이드                    333544 non-null   float64
17  HDL 콜레스테롤                       333541 non-null   float64
18  LDL 콜레스테롤                       327148 non-null   float64
19  혈색소                               994468 non-null   float64
20  요단백                               999694 non-null   float64
21  혈청크레아티닌                       994474 non-null   float64
22  (혈청치오티)AST                      994478 non-null   float64
23  (혈청치오티)ALT                      994477 non-null   float64
24  갈마 지티피                          994470 non-null   float64
25  흡연상태                             999834 non-null   float64
26  음주여부                             644918 non-null   float64
27  구강검진 수검여부                    1000000 non-null  int64
28  치아우식증유무                       397680 non-null   float64
29  결손치 유무                          1000000 non-null  object
30  치아마모증유무                       1000000 non-null  object
31  제3대구치(사랑니) 이상              1000000 non-null  object
32  치석                                  397680 non-null   float64
33  데이터 공개일자                      1000000 non-null  int64
dtypes: float64(22), int64(9), object(3)
memory usage: 259.4+ MB
```

```
1 raw_data.isnull().sum()

기준년도                                0
가입자 일련번호                          0
시도코드                                0
성별코드                                0
연령대 코드(5세단위)                    0
신장(50cm단위)                          0
체중(5kg 단위)                          0
허리둘레                                403
시력(좌)                                195
시력(우)                                188
청력(좌)                                181
청력(우)                                178
수축기 혈압                             5424
이완기 혈압                             5425
식전혈당(공복혈당)                     5523
총 콜레스테롤                           666451
트리글리세라이드                       666456
HDL 콜레스테롤                         666459
LDL 콜레스테롤                         672852
혈색소                                  5532
요단백                                  10306
혈청크레아티닌                         5526
(혈청치오티)AST                        5522
(혈청치오티)ALT                        5523
갈마 지티피                            5530
흡연상태                               166
음주여부                               355082
구강검진 수검여부                      0
치아우식증유무                         602320
결손치 유무                             0
치아마모증유무                         0
제3대구치(사랑니) 이상                 0
치석                                    602320
데이터 공개일자                        0
dtype: int64
```

# 데이터 전처리

컨텐츠  
전처리

# Python Contents

## Contents ↻ ⚙

- 1 데이터 전처리 및 저장
- ▼ 2 전체 데이터 (유: 96,324명 / 무:24,100명)
  - ▼ 2.1 tensorflow
    - 2.1.1 keras / binary\_crossentropy
  - ▼ 2.2 pycaret - Binary Classification
    - 2.2.1 ridge
    - 2.2.2 lda
- ▼ 3 샘플데이터 (유: 20,000명 / 무:20,000명)
  - ▼ 3.1 tensorflow
    - 3.1.1 keras / binary\_crossentropy
  - ▼ 3.2 pycaret - Binary Classification
    - 3.2.1 ridge
    - 3.2.2 lda

# Data Preprocessing

```
1 raw_data = pd.read_csv("data/국민건강보험공단_건강검진정보_20191231.csv", encoding='cp949')
2 raw_data = raw_data[raw_data['구강검진 수검여부'] == 1]
3 raw_data['음주여부'].fillna(0, inplace=True)
4 raw_data.dropna(how='any', inplace=True)
5 raw_data = raw_data.loc[:, '시도코드' : '치아우식증유무']
6 raw_data['치아우식증유무'] = raw_data['치아우식증유무'].astype(int)
7 raw_data.drop('구강검진 수검여부', axis=1, inplace=True)
8 raw_data.index = range(raw_data.shape[0])
9 raw_data.head(1)
```

시 도 코 드	성 별 코 드	연령대 코드(5 세단위)	신장 (5Cm단 위)	체중 (5Kg 단 위)	허리 둘레	시력 (좌)	시력 (우)	청력 (좌)	청력 (우)	...	LDL 콜 레스테 롤	혈색 소	요 단 백	혈청크 레아티 닌	(혈청지오 티)AST	(혈청지오 티)ALT	감마 지티 피	흡연 상태	음주 여부	치아 우식 증유 무
0	41	1	6	175	70	81.3	0.9	1.0	1.0	...	93.0	13.1	1.0	0.9	14.0	11.0	30.0	2.0	1.0	0

```
1 raw_data.isnull().sum()

시도코드      0
성별코드      0
연령대 코드(5세단위)  0
신장(5Cm단위)  0
체중(5Kg 단위)  0
허리둘레      0
시력(좌)      0
시력(우)      0
청력(좌)      0
청력(우)      0
수축기 혈압    0
이완기 혈압    0
식전혈당(공복혈당)  0
총 콜레스테롤  0
트리글리세라이드  0
HDL 콜레스테롤  0
LDL 콜레스테롤  0
혈색소        0
요단백        0
혈청크레아티닌  0
(혈청지오티)AST  0
(혈청지오티)ALT  0
감마 지티피    0
흡연상태      0
음주여부      0
치아우식증유무  0
dtype: int64
```

```
1 raw_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 120424 entries, 0 to 120423
Data columns (total 26 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   시도코드                                 120424 non-null  int64
1   성별코드                                 120424 non-null  int64
2   연령대 코드(5세단위)                   120424 non-null  int64
3   신장(5Cm단위)                           120424 non-null  int64
4   체중(5Kg 단위)                           120424 non-null  int64
5   허리둘레                                120424 non-null  float64
6   시력(좌)                                120424 non-null  float64
7   시력(우)                                120424 non-null  float64
8   청력(좌)                                120424 non-null  float64
9   청력(우)                                120424 non-null  float64
10  수축기 혈압                             120424 non-null  float64
11  이완기 혈압                             120424 non-null  float64
12  식전혈당(공복혈당)                     120424 non-null  float64
13  총 콜레스테롤                           120424 non-null  float64
14  트리글리세라이드                       120424 non-null  float64
15  HDL 콜레스테롤                         120424 non-null  float64
16  LDL 콜레스테롤                         120424 non-null  float64
17  혈색소                                  120424 non-null  float64
18  요단백                                  120424 non-null  float64
19  혈청크레아티닌                         120424 non-null  float64
20  (혈청지오티)AST                       120424 non-null  float64
21  (혈청지오티)ALT                       120424 non-null  float64
22  감마 지티피                             120424 non-null  float64
23  흡연상태                               120424 non-null  float64
24  음주여부                               120424 non-null  float64
25  치아우식증유무                         120424 non-null  int32
dtypes: float64(20), int32(1), int64(5)
memory usage: 23.4 MB
```

치아우식증유무: 충치유무(有:1 / 無:0)

- 문제점 : 높은 accuracy(정확도)에 비해 **recall(재현률)**이 매우 낮음
- 원인 : 낮은 충치 비율  
(120,424명中 24,100명, 약20%)
- 해결 : 충치無 20,000명 / 충치有 20,000명 샘플링



```
1 score = model.evaluate(X_test, Y_test, verbose=0)
2 print('model loss :', score[0])
3 print('model accuracy :', score[1])
4 print('model recall :', score[2])
5 print('model precision :', score[3])

model loss : 0.49374809861183167
model accuracy : 0.7973040342330933
model recall : 0.0
model precision : 0.0
```

```
1 predict_model(final_ridge)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Ridge Classifier	0.7988	0.5001	0.0002	1.0000	0.0003	0.0003	0.0118

```
1 predict_model(final_lda)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Linear Discriminant Analysis	0.7988	0.6087	0.0005	0.6000	0.0010	0.0007	0.0132

# Preprocessed Data

```
1 result_df = pd.read_csv('data/result_df.csv', encoding='utf-8')
2 result_df_0 = result_df[result_df['치아우식증유무']==0].sample(20000)
3 result_df_1 = result_df[result_df['치아우식증유무']==1].sample(20000)
4 result_df = pd.concat([result_df_0, result_df_1])
5 result_df.sort_index(inplace=True)
6 print('총치無 : ', len(result_df[result_df['치아우식증유무']==0]))
7 print('총치有 : ', len(result_df[result_df['치아우식증유무']==1]))
```

총치無 : 20000  
총치有 : 20000

```
1 result_df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 40000 entries, 2 to 120421
Data columns (total 26 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   시도코드                             40000 non-null  int64
1   성별코드                             40000 non-null  int64
2   연령대 코드(5세단위)                 40000 non-null  int64
3   신장(50cm단위)                       40000 non-null  int64
4   체중(5Kg 단위)                       40000 non-null  int64
5   허리둘레                             40000 non-null  float64
6   시력(좌)                             40000 non-null  float64
7   시력(우)                             40000 non-null  float64
8   청력(좌)                             40000 non-null  float64
9   청력(우)                             40000 non-null  float64
10  수축기 혈압                          40000 non-null  float64
11  이완기 혈압                          40000 non-null  float64
12  식전혈당(공복혈당)                   40000 non-null  float64
13  총 콜레스테롤                        40000 non-null  float64
14  트리글리세라이드                    40000 non-null  float64
15  HDL 콜레스테롤                      40000 non-null  float64
16  LDL 콜레스테롤                      40000 non-null  float64
17  혈색소                              40000 non-null  float64
18  요단백                              40000 non-null  float64
19  혈청크레아티닌                      40000 non-null  float64
20  (혈청지오티)AST                      40000 non-null  float64
21  (혈청지오티)ALT                      40000 non-null  float64
22  감마 지티피                          40000 non-null  float64
23  흡연상태                             40000 non-null  float64
24  음주여부                             40000 non-null  float64
25  치아우식증유무                       40000 non-null  int64
dtypes: float64(20), int64(6)
memory usage: 8.2 MB
```

# 모델구성 & 학습

Tensorflow

Pycaret



## 구분 : tensorflow / binary\_crossentropy

```
# 독립변수와 타겟변수 분리 & scale 조정 & 학습셋과 테스트셋 분리
Input = result_df.iloc[:, :-1]
Target = result_df.iloc[:, [-1]]
scaler = MinMaxScaler()
scaler.fit(Input)
scaled_input = pd.DataFrame(scaler.transform(Input))
X_train, X_test, Y_train, Y_test = train_test_split(scaled_input, Target, test_size=0.3, random_state=5)

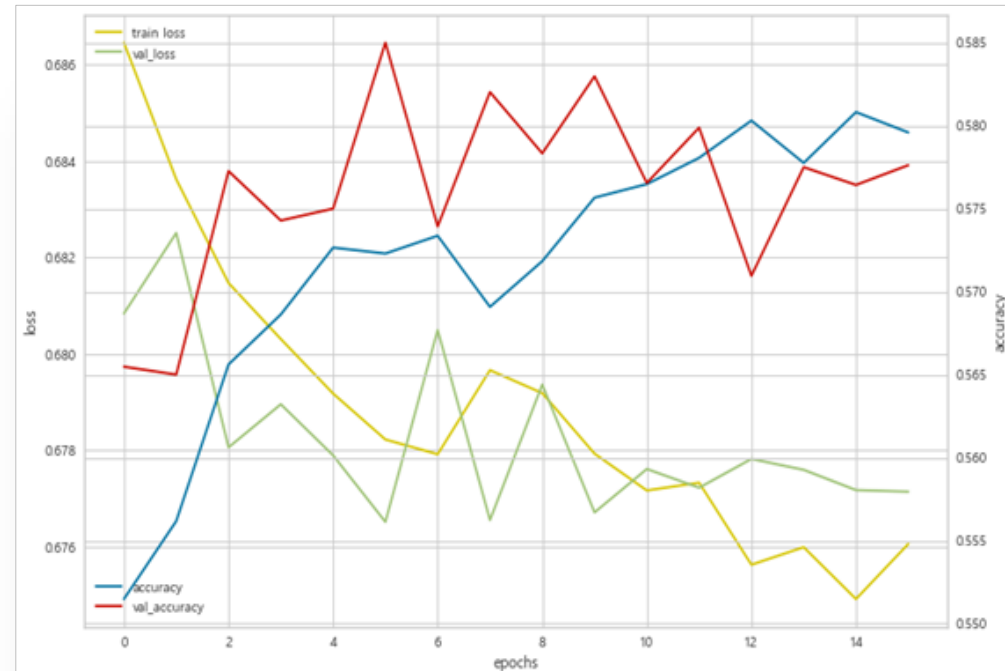
# 모델 구성
model = Sequential()
model.add(Dense(units=1000, input_dim=25, activation='relu'))
model.add(Dense(units=600, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(units=300, activation='relu'))
model.add(Dense(units=100, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(units=1, activation='sigmoid'))

# 학습 및 학습과정 그래프
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy', metrics.Recall(), metrics.Precision()])
earlyStopping = EarlyStopping(patience=10)
hist = model.fit(X_train, Y_train, epochs=50, batch_size=1000, verbose=1, validation_split=0.3, callbacks=[earlyStopping])

fig, loss_ax = plt.subplots(figsize=(12,8))
loss_ax.plot(hist.history['loss'], 'y', label='train loss')
loss_ax.plot(hist.history['val_loss'], 'g', label='val_loss')
loss_ax.set_xlabel('epochs')
loss_ax.set_ylabel('loss')

acc_ax = loss_ax.twinx()
acc_ax.plot(hist.history['accuracy'], 'b', label='accuracy')
acc_ax.plot(hist.history['val_accuracy'], 'r', label='val_accuracy')
acc_ax.set_ylabel('accuracy')

loss_ax.legend(loc='upper left')
acc_ax.legend(loc='lower left')
plt.show()
```



## 구분 : pycaret / Compare & Create & Tune a Model

```
# 학습셋과 테스트셋 분리
data = result_df.sample(frac=0.95, random_state=786)
data_unseen = result_df.drop(data.index)
data.reset_index(inplace=True, drop=True)
data_unseen.reset_index(inplace=True, drop=True)

# PyCaret 환경설정 및 모델 비교하기
from pycaret.classification import *
exp_clf101 = setup(data = data, target = '치아우식증유무', session_id=123)
best_model = compare_models()
```

```
1 best_model = compare_models()
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	lda	0.5761	0.6039	0.5552	0.5793	0.5670	0.1522	0.1524	0.1970
1	ridge	0.5759	0.0000	0.5551	0.5791	0.5668	0.1518	0.1519	0.0550
2	lr	0.5743	0.6037	0.5551	0.5772	0.5659	0.1487	0.1488	2.0640
3	gbc	0.5697	0.5996	0.5555	0.5715	0.5634	0.1394	0.1394	2.1700
4	ada	0.5682	0.5960	0.5557	0.5699	0.5626	0.1364	0.1365	0.5220
5	lightgbm	0.5645	0.5908	0.5366	0.5681	0.5518	0.1291	0.1293	0.2960
6	nb	0.5575	0.5857	0.3493	0.5983	0.4399	0.1149	0.1265	0.0440
7	rf	0.5564	0.5814	0.5320	0.5591	0.5452	0.1127	0.1129	1.7620
8	et	0.5507	0.5738	0.5257	0.5532	0.5390	0.1013	0.1015	1.7700
9	xgboost	0.5504	0.5699	0.5405	0.5513	0.5457	0.1008	0.1008	1.9280
10	catboost	0.5153	0.5406	0.5011	0.5173	0.5091	0.1306	0.1307	5.9490
11	dt	0.5100	0.5109	0.5095	0.5108	0.5101	0.0218	0.0218	0.2120
12	svm	0.5070	0.0000	0.5213	0.5275	0.3793	0.0140	0.0272	0.8240
13	knn	0.5063	0.5063	0.4994	0.5063	0.5028	0.0127	0.0127	1.3660
14	qda	0.5038	0.5038	0.6115	0.5020	0.5408	0.0076	0.0087	0.1530

```
1 lda = create_model('lda')
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.5756	0.6019	0.5519	0.5793	0.5653	0.1511	0.1513
1	0.5797						
2	0.5684						
3	0.5793						
4	0.5714						
5	0.5647						
6	0.5835						
7	0.5782						
8	0.5733						
9	0.5871						
Mean	0.5761						
SD	0.0065						

```
1 tuned_lda = tune_model(lda)
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.5771	0.5998	0.5474	0.5819	0.5641	0.1541	0.1544
1	0.5774	0.6125	0.5662	0.5792	0.5726	0.1549	0.1549
2	0.5699	0.5851	0.5376	0.5748	0.5556	0.1398	0.1401
3	0.5786	0.5973	0.5594	0.5817	0.5703	0.1571	0.1573
4	0.5684	0.6015	0.5556	0.5702	0.5628	0.1368	0.1369
5	0.5692	0.5952	0.5421	0.5731	0.5572	0.1383	0.1385
6	0.5850	0.6054	0.5681	0.5875	0.5777	0.1699	0.1700
7	0.5793	0.6124	0.5561	0.5828	0.5691	0.1586	0.1588
8	0.5748	0.5944	0.5425	0.5796	0.5604	0.1496	0.1499
9	0.5927	0.6298	0.5561	0.5998	0.5771	0.1854	0.1859
Mean	0.5772	0.6033	0.5531	0.5811	0.5667	0.1545	0.1547
SD	0.0071	0.0118	0.0098	0.0079	0.0075	0.0142	0.0143

```
1 ridge = create_model('ridge')
```

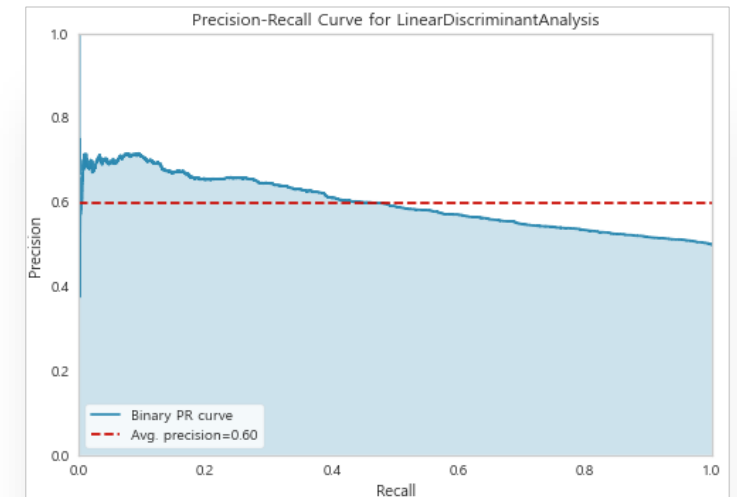
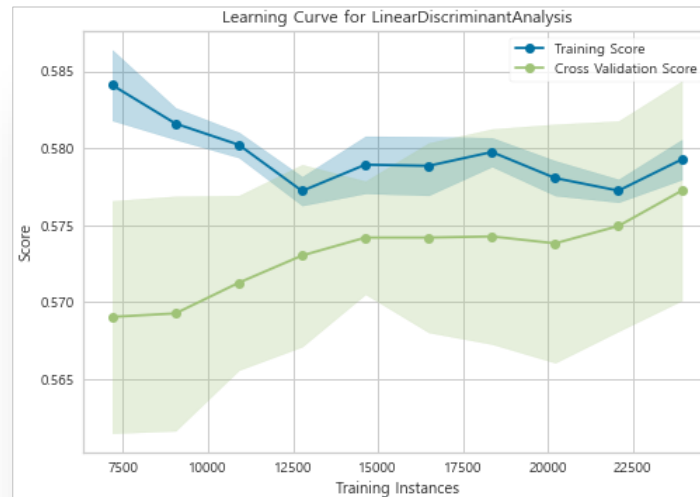
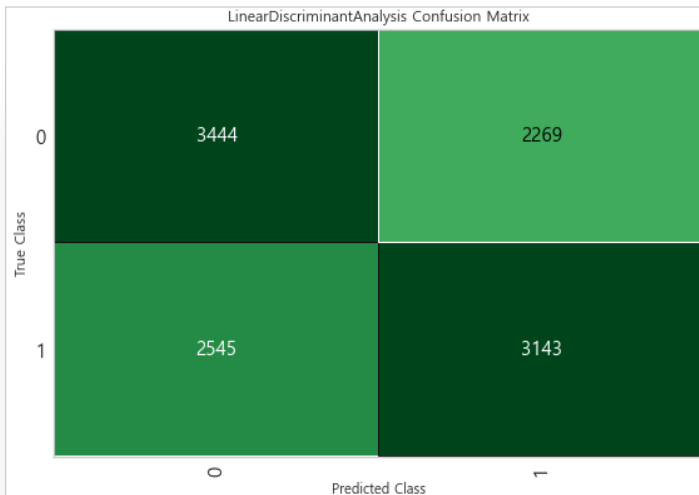
	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.5752	0.0000	0.5519	0.5789	0.5651	0.1504	0.1505
1	0.5797						
2	0.5680						
3	0.5789						
4	0.5714						
5	0.5647						
6	0.5827						
7	0.5786						
8	0.5729						
9	0.5867						
Mean	0.5759						
SD	0.0064						

```
1 tuned_ridge = tune_model(ridge)
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.5767	0.0000	0.5534	0.5804	0.5666	0.1534	0.1536
1	0.5801	0.0000	0.5722	0.5814	0.5767	0.1602	0.1602
2	0.5695	0.0000	0.5383	0.5742	0.5557	0.1391	0.1394
3	0.5778	0.0000	0.5632	0.5802	0.5715	0.1556	0.1557
4	0.5711	0.0000	0.5632	0.5722	0.5676	0.1421	0.1421
5	0.5635	0.0000	0.5398	0.5667	0.5529	0.1271	0.1272
6	0.5850	0.0000	0.5673	0.5877	0.5773	0.1699	0.1700
7	0.5767	0.0000	0.5553	0.5797	0.5673	0.1534	0.1535
8	0.5737	0.0000	0.5515	0.5767	0.5638	0.1473	0.1475
9	0.5904	0.0000	0.5568	0.5968	0.5761	0.1809	0.1813
Mean	0.5765	0.0000	0.5561	0.5796	0.5676	0.1529	0.1530
SD	0.0073	0.0000	0.0105	0.0079	0.0080	0.0146	0.0147

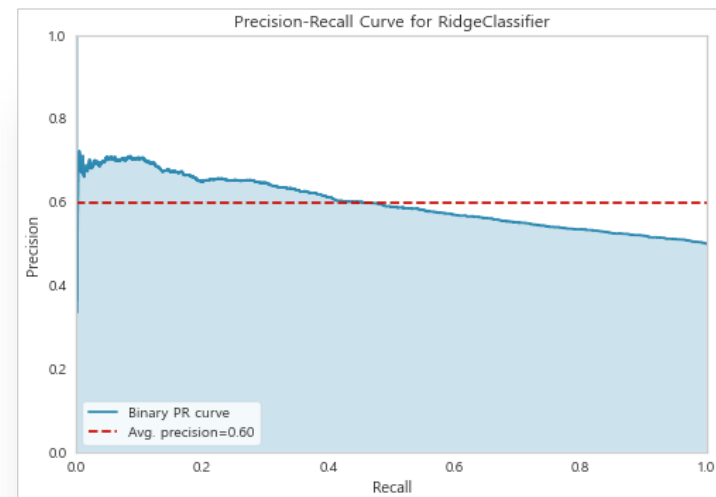
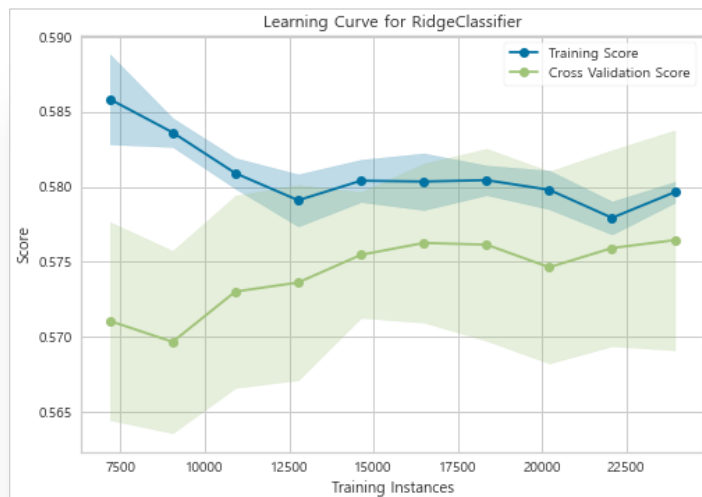
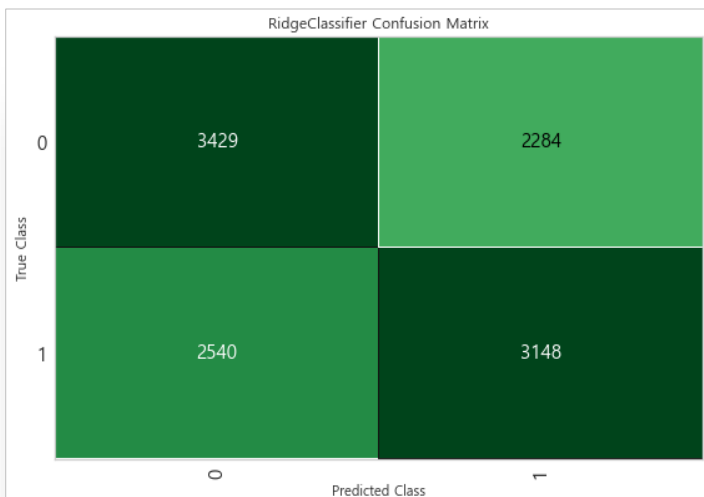
## 구분 : pycaret / Ida

1	evaluate_model(tuned_Ida)				
Plot Type:	Hyperparameters	AUC	Confusion Matrix	Threshold	Precision Recall
	Prediction Error	Class Report	Feature Selection	Learning Curve	Manifold Learning
	Calibration Curve	Validation Curve	Dimensions	Feature Importance	Feature Importance...
	Decision Boundary	Lift Chart	Gain Chart	Decision Tree	KS Statistic Plot



## 구분 : pycaret / ridge

1	evaluate_model(tuned_ridge)				
Plot Type:	Hyperparameters	AUC	Confusion Matrix	Threshold	Precision Recall
	Prediction Error	Class Report	Feature Selection	Learning Curve	Manifold Learning
	Calibration Curve	Validation Curve	Dimensions	Feature Importance	Feature Importance...
	Decision Boundary	Lift Chart	Gain Chart	Decision Tree	KS Statistic Plot



## 구분 : pycaret / Finalize Model

```
1 predict_model(tuned_lda):
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Linear Discriminant Analysis	0.5778	0.6052	0.5526	0.5807	0.5663	0.1554	0.1556

```
1 final_lda = finalize_model(tuned_lda)
```

```
1 print(final_lda)
```

LinearDiscriminantAnalysis(n\_components=None, priors=None, shrinkage='auto', solver='lsqr', store\_covariance=False, tol=0.0001)

```
1 predict_model(final_lda)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Linear Discriminant Analysis	0.5799	0.6094	0.5411	0.5854	0.5624	0.1596	0.1600

	체중 (5Kg 단위)	허리 둘레	시력 (좌)	시력 (우)	수축기 혈압	이완기 혈압	식전 혈당 (공복 혈당)	총 콜레스테롤	트리글리세라이드	HDL 콜레스테롤	...	신장 (5Cm단위)_180	신장 (5Cm단위)_185	신장 (5Cm단위)_190	청력 (좌)_1.0	청력 (우)_1.0	음연 상태_2.0	음주 여부_0.0	지아우식증 유무	Label	Score
0	55.0	77.5	0.9	1.2	110.0	71.0	98.0	222.0	90.0	51.0	...	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0	1	0.5153
1	90.0	102.0	1.0	0.9	120.0	85.0	111.0	201.0	273.0	39.0	...	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0	1	0.5567
2	70.0	89.0	0.4	0.4	130.0	81.0	305.0	142.0	201.0	30.0	...	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0	0	0.5385
3	75.0	81.0	1.5	1.5	116.0	84.0	86.0	214.0	84.0	52.0	...	0.0	1.0	0.0	1.0	1.0	1.0	0.0	0	1	0.5343
4	50.0	72.0	0.6	0.7	104.0	70.0	83.0	213.0	113.0	56.0	...	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0	0	0.6115
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11396	65.0	86.0	1.0	1.2	131.0	70.0	86.0	238.0	177.0	56.0	...	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1	1	0.6235
11397	75.0	80.0	1.5	1.2	110.0	70.0	96.0	189.0	66.0	54.0	...	0.0	0.0	0.0	1.0	1.0	0.0	1.0	0	1	0.5692
11398	80.0	98.0	0.5	0.7	142.0	72.0	104.0	125.0	196.0	32.0	...	0.0	0.0	0.0	1.0	1.0	1.0	0.0	0	0	0.5523
11399	75.0	79.0	1.0	0.9	138.0	76.0	148.0	162.0	82.0	41.0	...	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0	1	0.5839
11400	65.0	74.0	2.0	2.0	110.0	72.0	98.0	162.0	65.0	69.0	...	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1	1	0.5830

11401 rows x 69 columns

```
1 predict_model(tuned_ridge):
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Ridge Classifier	0.5769	0.5768	0.5534	0.5795	0.5662	0.1537	0.1538

```
1 final_ridge = finalize_model(tuned_ridge)
```

```
1 print(final_ridge)
```

RidgeClassifier(alpha=9.17, class\_weight=None, copy\_X=True, fit\_intercept=False, max\_iter=None, normalize=False, random\_state=123, solver='auto', tol=0.001)

```
1 predict_model(final_ridge)
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Ridge Classifier	0.5791	0.5790	0.5424	0.5842	0.5625	0.1580	0.1584

	체중 (5Kg 단위)	허리 둘레	시력 (좌)	시력 (우)	수축기 혈압	이완기 혈압	식전 혈당 (공복 혈당)	총 콜레스테롤	트리글리세라이드	HDL 콜레스테롤	...	신장 (5Cm단위)_175	신장 (5Cm단위)_180	신장 (5Cm단위)_185	신장 (5Cm단위)_190	청력 (좌)_1.0	청력 (우)_1.0	음연 상태_2.0	음주 여부_0.0	지아우식증 유무	Label	Score
0	55.0	77.5	0.9	1.2	110.0	71.0	98.0	222.0	90.0	51.0	...	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0	1	0.5153
1	90.0	102.0	1.0	0.9	120.0	85.0	111.0	201.0	273.0	39.0	...	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0	1	0.5567
2	70.0	89.0	0.4	0.4	130.0	81.0	305.0	142.0	201.0	30.0	...	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0	0	0.5385
3	75.0	81.0	1.5	1.5	116.0	84.0	86.0	214.0	84.0	52.0	...	0.0	0.0	1.0	0.0	1.0	1.0	1.0	0.0	0	1	0.5343
4	50.0	72.0	0.6	0.7	104.0	70.0	83.0	213.0	113.0	56.0	...	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0	0	0.6115
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11396	65.0	86.0	1.0	1.2	131.0	70.0	86.0	238.0	177.0	56.0	...	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1	1	0.6235
11397	75.0	80.0	1.5	1.2	110.0	70.0	96.0	189.0	66.0	54.0	...	1.0	0.0	0.0	0.0	1.0	1.0	0.0	1.0	0	1	0.5692
11398	80.0	98.0	0.5	0.7	142.0	72.0	104.0	125.0	196.0	32.0	...	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	0	0	0.5523
11399	75.0	79.0	1.0	0.9	138.0	76.0	148.0	162.0	82.0	41.0	...	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0	1	0.5839
11400	65.0	74.0	2.0	2.0	110.0	72.0	98.0	162.0	65.0	69.0	...	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1	1	0.5830

11401 rows x 68 columns

# 결과

Tensorflow

Pycaret

연구의 한계점

## 구분 : tensorflow

```

1 score = model.evaluate(X_test, Y_test, verbose=0)
2 print('model loss :', score[0])
3 print('model accuracy :', score[1])
4 print('model recall : ', score[2])
5 print('model precision :', score[3])

```

model loss : 0.6803668737411499

model accuracy : 0.5715833306312561

model recall : 0.5064250230789185

model precision : 0.5890017151832581

```

1 pred = model.predict(X_test)
2 pred = (pred>0.5)
3 print(confusion_matrix(Y_test, pred), end='\n')
4 print('f1_score : ', f1_score(Y_test, pred))

```

[[3785 2145]

[2996 3074]]

f1\_score : 0.5446009389671362

## 구분 : pycaret

Accuracy

0.578

```

1 unseen_predictions = predict_model(final_lda, data=data_unseen)
2 unseen_predictions.head()

```

	시도 코드	성별 코드	연령대 코드(5세단위)	신장 (5Cm단위)	체중 (5Kg단위)	허리둘레	시력 (좌)	시력 (우)	청력 (좌)	청력 (우)	...	요단백	혈청크레아티닌	(혈청지오티)AST	(혈청지오티)ALT	감마지티피	흡연상태	음주여부	치아우식증유무	Label	Score
0	42	1	9	165	90	94.3	0.8	0.5	1.0	1.0	...	1.0	0.9	30.0	46.0	95.0	2.0	1.0	1	1	0.6156
1	41	1	9	175	100	102.0	1.2	1.2	1.0	1.0	...	4.0	1.2	47.0	72.0	55.0	2.0	0.0	0	1	0.5992
2	41	1	10	170	70	85.0	0.7	0.6	1.0	1.0	...	1.0	1.0	26.0	31.0	25.0	2.0	1.0	0	1	0.5278
3	29	1	6	170	80	97.0	1.0	1.5	1.0	1.0	...	1.0	0.8	56.0	118.0	123.0	2.0	1.0	0	1	0.6090
4	48	1	7	175	75	80.0	1.0	1.0	1.0	1.0	...	1.0	1.1	15.0	28.0	20.0	2.0	1.0	1	1	0.6371

5 rows x 28 columns

```

1 from pycaret.utils import check_metric
2 check_metric(unseen_predictions['치아우식증유무'], unseen_predictions['Label'], metric = 'Accuracy')

```

Accuracy

0.5835

```

1 unseen_predictions = predict_model(final_ridge, data=data_unseen)
2 unseen_predictions.head()

```

	시도 코드	성별 코드	연령대 코드(5세단위)	신장 (5Cm단위)	체중 (5Kg단위)	허리둘레	시력 (좌)	시력 (우)	청력 (좌)	청력 (우)	...	혈색소	요단백	혈청크레아티닌	(혈청지오티)AST	(혈청지오티)ALT	감마지티피	흡연상태	음주여부	치아우식증유무	Label
0	42	1	9	165	90	94.3	0.8	0.5	1.0	1.0	...	15.6	1.0	0.9	30.0	46.0	95.0	2.0	1.0	1	1
1	41	1	9	175	100	102.0	1.2	1.2	1.0	1.0	...	15.1	4.0	1.2	47.0	72.0	55.0	2.0	0.0	0	1
2	41	1	10	170	70	85.0	0.7	0.6	1.0	1.0	...	15.4	1.0	1.0	26.0	31.0	25.0	2.0	1.0	0	1
3	29	1	6	170	80	97.0	1.0	1.5	1.0	1.0	...	14.6	1.0	0.8	56.0	118.0	123.0	2.0	1.0	0	1
4	48	1	7	175	75	80.0	1.0	1.0	1.0	1.0	...	15.4	1.0	1.1	15.0	28.0	20.0	2.0	1.0	1	1

5 rows x 27 columns

```

1 from pycaret.utils import check_metric
2 check_metric(unseen_predictions['치아우식증유무'], unseen_predictions['Label'], metric = 'Accuracy')

```





**충치 치료여부** 에 대한 정보 없음



낮은 **Accuracy**(정확도) 와 **Precision**(정밀도)

# Q & A

