위스콘신 유방암 데이터의 자동 시스템 만들기

학습 목표

- 자동 시스템을 구현해 본다.

목차

01 라이브러리 설치 및 불러오기

목차로 이동하기

- 설치
  - 설치의 약간의 시간이 걸린다.

```
pip install pycaret
```

- 만약, 머신러닝 알고리즘 XGBoost, LightGBM, CatBoost 등의 알고리즘을 사용하려면 해당 라이브러리도 설치되어 있어야 한다.

```
pip install xgboost
pip install lightgbm
pip install catboost
```

In [1]:
```
from pycaret.classification import *
from pycaret.datasets import get_data
import pycaret

from sklearn.datasets import load_breast_cancer
import pandas as pd
import xgboost, lightgbm, catboost

import warnings
warnings.filterwarnings(action='ignore')
```

In [2]:
```
print(f"pycaret version : {pycaret.__version__}")
print(f"xgboost version : {xgboost.__version__}")
print(f"lightgbm version : {lightgbm.__version__}")
print(f"catboost version : {catboost.__version__}")
```

```
pycaret version : 3.0.1
xgboost version : 1.7.5
lightgbm version : 3.3.5
catboost version : 1.2
```

02 pycaret 기본 입문

목차로 이동하기

데이터 불러오기

In [3]:
```
# 데이터 로드 - get_data로 가져올 수 없음.(유방암 데이터)
cancer = load_breast_cancer()

cancer_df = pd.DataFrame(cancer.data, columns=cancer.feature_names)
cancer_df['target'] = cancer.target
cancer_df
```

Out[3]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.241
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.181
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.206
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.259
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	0.180
...	...	...	...	...	...	...	...	...	...
564	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.172
565	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.175
566	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	0.159
567	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	0.239
568	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	0.158

569 rows × 31 columns

데이터 분석 및 머신러닝을 위한 초기 설정 수행

data: 분석에 사용할 데이터셋을 지정합니다.
target: 예측할 대상 변수를 지정합니다.
normalize: 데이터를 정규화할지 여부를 설정합니다.
transform_target: 대상 변수를 변환할지 여부를 설정합니다.
transform_method: 대상 변수를 변환하는 방법을 지정합니다.
numeric_features: 숫자형 변수로 간주할 특성을 지정합니다.
categorical_features: 범주형 변수로 간주할 특성을 지정합니다.
ignore_features: 분석에서 제외할 특성을 지정합니다.
feature_selection: 피처 선택을 수행할지 여부를 설정합니다.
feature_selection_threshold: 피처 선택에 사용할 임계값을 지정합니다.
normalize_method: 데이터 정규화에 사용할 방법을 지정합니다.
transformation: 데이터 변환에 사용할 방법을 지정합니다.
session_id: 실행 세션의 고유 식별자를 지정합니다.

In [4]:
```
# 분류기 설정 및 데이터셋 준비
clf = setup(cancer_df, target='target', session_id=123)
```

	Description	Value
0	Session id	123
1	Target	target
2	Target type	Binary
3	Original data shape	(569, 31)
4	Transformed data shape	(569, 31)
5	Transformed train set shape	(398, 31)
6	Transformed test set shape	(171, 31)
7	Numeric features	30
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	mode
12	Fold Generator	StratifiedKFold
13	Fold Number	10
14	CPU Jobs	-1
15	Use GPU	False
16	Log Experiment	False
17	Experiment Name	clf-default-name
18	USI	8b1f

- Preprocess : True(전처리 수행)
  - 결측치 처리
  - 데이터 인코딩 - 범주형 변수를 숫자 변환(원핫 인코딩, 레이블 인코딩)
  - 정규화 - 수치형 변수의 분포를 정규화하여 스케일링
  - 피처 선택(중요한 변수 선택 및 불필요한 피처 제거) 등을 수행.
- Imputation type : 결측치 처리 방법
  - simple : 평균(mean) 또는 최빈값(mode)로 사용하여 간단하게 결측치 대체
- Numeric imputation : 수치형 변수 대체 방법
- Categorical imputation : 범주형 변수 대체 방법
- Fold Generator: 교차 검증을 위한 폴드(fold) 생성 방법
- Use GPU : GPU 사용 여부
- Experiment Name : 실험의 이름

모델 비교

- 모델을 비교한 이후에 결과가 출력된다. 각 metric별 가장 성능이 좋은 위치에 노락색으로 하이라이트 되어 출력이 된다.
- fold : cross_validation의 fold를 지정(default=10)

- sort : 정렬기준 지표 설정
- n_select : 상위 n개의 모델 결과만 출력.

- 총 19개의 모델 사용(pyCaret 3.0.1 버전) - catboost 추가.

```
선형 모델: Linear Regression, Lasso Regression, Ridge Regression,
Elastic Net Regression
결정 트리 기반 모델: Decision Tree, Random Forest, Extra Trees
부스팅 모델: Gradient Boosting, Extreme Gradient Boosting
(XGBoost), Light Gradient Boosting Machine (LGBM), CatBoost
K-최근접 이웃 (K-Nearest Neighbors, KNN)
서포트 벡터 머신 (Support Vector Machines, SVM)
나이브 베이즈 (Naive Bayes)
다층 퍼셉트론 (Multi-Layer Perceptron, MLP)
Ridge 분류기
Quadratic Discriminant Analysis (QDA)
Linear Discriminant Analysis (LDA)
Logistic Regression
AdaBoost
Light Gradient Boosting on Trees (LightGBM)
Gradient Boosting Classifier (CatBoost)
Extra Trees Classifier
Random Forest Classifier
Decision Tree Classifier
K-Nearest Neighbors Classifier
Extreme Gradient Boosting Classifier (XGBoost)
```

In [6]:
```
# 모델 비교 및 성능 평가
best_model = compare_models()
best_model
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.9674	0.9901	0.9840	0.9657	0.9745	0.9295	0.9306	0.7890
catboost	CatBoost Classifier	0.9674	0.9899	0.9840	0.9654	0.9743	0.9298	0.9310	1.5280
lightgbm	Light Gradient Boosting Machine	0.9650	0.9885	0.9840	0.9620	0.9724	0.9246	0.9270	1.0900
rf	Random Forest Classifier	0.9574	0.9864	0.9680	0.9652	0.9662	0.9087	0.9101	1.0440
gbc	Gradient Boosting Classifier	0.9549	0.9829	0.9760	0.9540	0.9644	0.9026	0.9047	0.7260
ridge	Ridge Classifier	0.9524	0.0000	1.0000	0.9317	0.9641	0.8940	0.9012	0.7960
xgboost	Extreme Gradient Boosting	0.9524	0.9861	0.9680	0.9577	0.9623	0.8975	0.8993	0.6820
ada	Ada Boost Classifier	0.9499	0.9881	0.9800	0.9444	0.9614	0.8902	0.8935	0.7760
qda	Quadratic Discriminant Analysis	0.9497	0.9874	0.9520	0.9682	0.9597	0.8930	0.8944	0.8270
lda	Linear Discriminant Analysis	0.9474	0.9880	0.9960	0.9274	0.9601	0.8834	0.8897	0.7460
lr	Logistic Regression	0.9473	0.9919	0.9640	0.9544	0.9580	0.8871	0.8908	2.7430
knn	K Neighbors Classifier	0.9423	0.9660	0.9680	0.9436	0.9549	0.8749	0.8783	0.6920
nb	Naive Bayes	0.9348	0.9821	0.9720	0.9297	0.9497	0.8572	0.8614	0.7560
dt	Decision Tree Classifier	0.9272	0.9218	0.9440	0.9421	0.9421	0.8442	0.8475	0.7300
svm	SVM - Linear Kernel	0.8797	0.0000	0.9040	0.9212	0.9032	0.7392	0.7658	0.6050
dummy	Dummy Classifier	0.6282	0.5000	1.0000	0.6282	0.7716	0.0000	0.0000	1.2080

```
Processing:   0%|          | 0/69 [00:00<?, ?it/s]
```

Out[6]:
```
ExtraTreesClassifier(bootstrap=False, ccp_alpha=0.0, class_weight=None,
                     criterion='gini', max_depth=None, max_features='auto',
                     max_leaf_nodes=None, max_samples=None,
                     min_impurity_decrease=0.0, min_samples_leaf=1,
                     min_samples_split=2, min_weight_fraction_leaf=0.0,
                     n_estimators=100, n_jobs=-1, oob_score=False,
                     random_state=123, verbose=0, warm_start=False)
```

모델의 성능 최적화 - 하이퍼 파라미터 튜닝 진행

- GridSearch 또는 랜덤 서치(Random Search)와 같은 방법 수행.
- 다양한 조합 시도 및 교차 검증을 통해 각 조합의 성능을 평가

- tune_model([model], optimize=[]) 함수를 사용하여 모델의 하이퍼 파라미터 튜닝 진행
  - optimize : 평가 metric 지정

In [7]:
```
# 최적 모델 튜닝
tuned_model = tune_model(best_model)
```

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9000	0.9493	0.9200	0.9200	0.9200	0.7867	0.7867
1	0.9750	0.9973	1.0000	0.9615	0.9804	0.9459	0.9473
2	0.9750	1.0000	0.9600	1.0000	0.9796	0.9474	0.9487
3	0.9750	0.9973	0.9600	1.0000	0.9796	0.9474	0.9487
4	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
5	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
6	0.9250	0.9653	0.9600	0.9231	0.9412	0.8378	0.8391
7	0.9750	0.9893	1.0000	0.9615	0.9804	0.9459	0.9473
8	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
9	0.9487	0.9971	0.9600	0.9600	0.9600	0.8886	0.8886
Mean	0.9674	0.9896	0.9760	0.9726	0.9741	0.9300	0.9306
Std	0.0318	0.0168	0.0265	0.0306	0.0252	0.0682	0.0682

```
Processing:   0%|          | 0/7 [00:00<?, ?it/s]
Fitting 10 folds for each of 10 candidates, totalling 100 fits
Original model was better than the tuned model, hence it will be returned. NOTE: The
display metrics are for the tuned model (not the original one).
```

In [8]:
```
tuned_model
```

Out[8]:
```
ExtraTreesClassifier(bootstrap=False, ccp_alpha=0.0, class_weight=None,
                     criterion='gini', max_depth=None, max_features='auto',
                     max_leaf_nodes=None, max_samples=None,
                     min_impurity_decrease=0.0, min_samples_leaf=1,
                     min_samples_split=2, min_weight_fraction_leaf=0.0,
                     n_estimators=100, n_jobs=-1, oob_score=False,
                     random_state=123, verbose=0, warm_start=False)
```

튜닝된 모델을 이용하여 평가 및 새로운 데이터 예측 수행

In [9]:
```
# 모델 예측
predictions = predict_model(tuned_model)

# 모델 저장
save_model(tuned_model, 'breast_cancer_model')
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Extra Trees Classifier	0.9708	0.9977	0.9720	0.9811	0.9765	0.9378	0.9378

```
Transformation Pipeline and Model Successfully Saved
```

Out[9]:
```
(Pipeline(memory=FastMemory(location=C:₩Users₩DANIEL~1₩AppData₩Local₩Temp₩joblib),
          steps=[('numerical_imputer',
                  TransformerWrapper(exclude=None,
                                     include=['mean radius', 'mean texture',
                                              'mean perimeter', 'mean area',
                                              'mean smoothness',
                                              'mean compactness',
                                              'mean concavity',
                                              'mean concave points',
                                              'mean symmetry',
                                              'mean fractal dimension',
                                              'radius error', 'texture error',
                                              'perimet...
                  ExtraTreesClassifier(bootstrap=False, ccp_alpha=0.0,
                                       class_weight=None, criterion='gini',
                                       max_depth=None, max_features='auto',
                                       max_leaf_nodes=None, max_samples=None,
                                       min_impurity_decrease=0.0,
                                       min_samples_leaf=1, min_samples_split=2,
                                       min_weight_fraction_leaf=0.0,
                                       n_estimators=100, n_jobs=-1,
                                       oob_score=False, random_state=123,
                                       verbose=0, warm_start=False))],
          verbose=False),
 'breast_cancer_model.pkl')
```