4.7 LightGBM

4.8 분류실습

- 캐글 산탄데르 고객 만족 예측

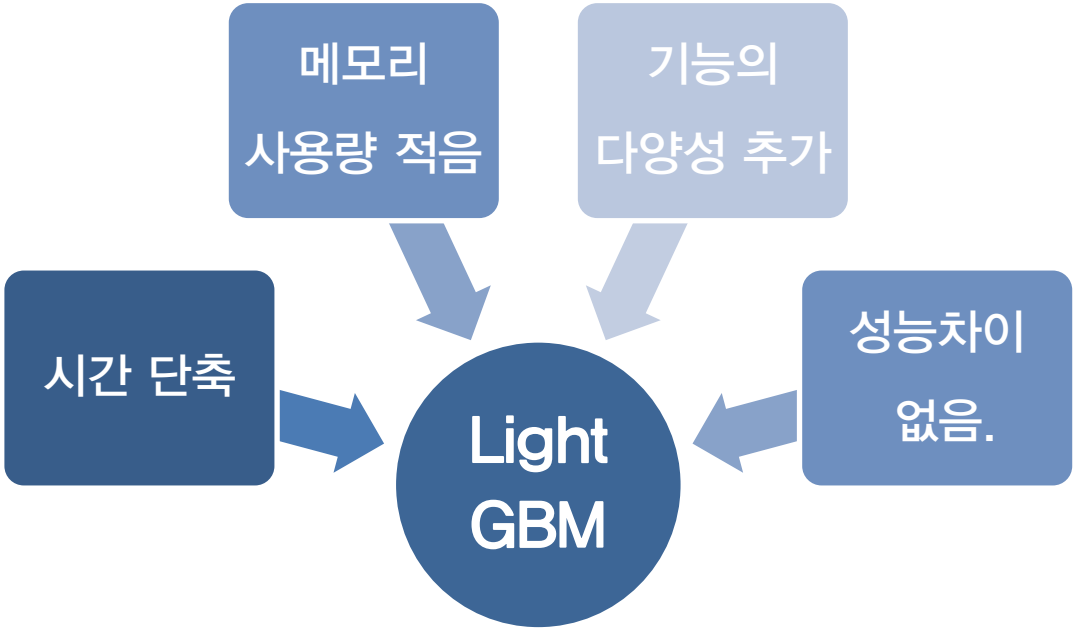
최보라

목차

4.7 LightGBM

4.7.1 소개

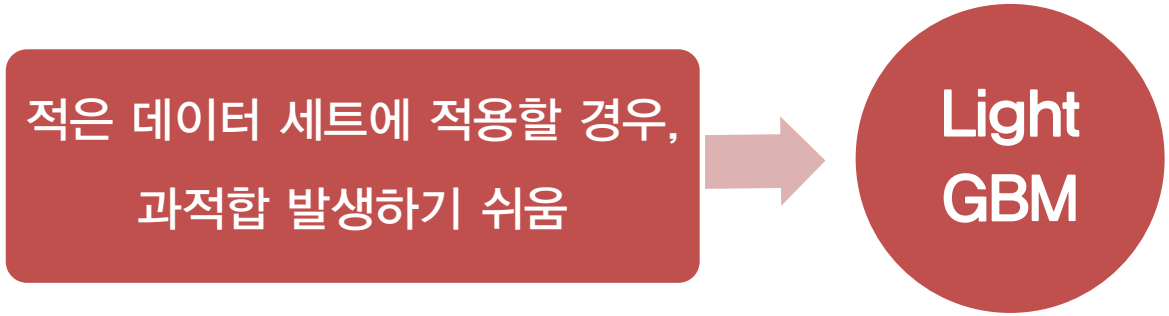
: XGBoost와 함께 부스팅 계열 알고리즘에서 각광 받고 있음.
: from lightgbm import LGBMClassifier



[XGBoost 대비 LightGBM의 장점]

4.7 LightGBM

4.7.1 소개



[XGBoost 대비 LightGBM의 단점]

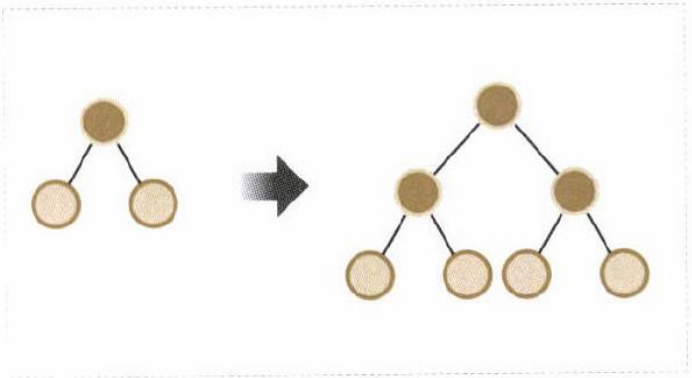
* 적은 데이터 – 10,000개 이하

4.7 LightGBM

4.7.1 소개

: LightGBM과 일반 GBM 계열의 **트리분할 방법의 차이점**

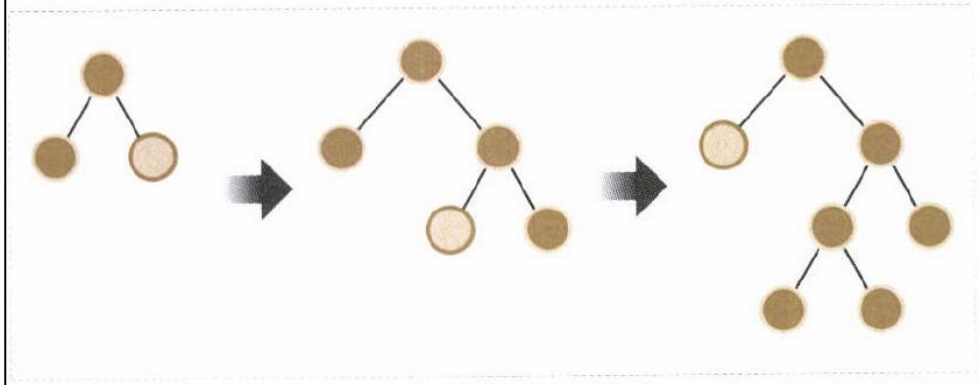
균형 트리 분할(Level Wise)



–〉 일반 GBM 계열

– 트리의 깊이 최소화
– 오버피팅에 강한구조
– 균형을 맞추기 위한 시간 필요

리프 중심 트리 분할(Leaf Wise)



–〉 LightGBM

– 최대손실값을 가지는
리프노드를 지속적으로 분할
– 예측오류손실 최소화 가능

4.7 LightGBM

4.7.2 하이퍼 파라미터

: LightGBM의 하이퍼 파라미터는 **XGBoost와 많은 부분이 유사**

: 유의할점 – 리프노드가 계속 분할되며 트리가 깊어지므로,
이에 맞는 하이퍼 파라미터 설정 필요.

(하이퍼 파라미터 튜닝방안)

– 방안1. num_leaves의 개수를 중심으로 min_child_samples, max_depth를 함께 조정하면서 모델의 복잡도를 줄이는 것

: num_leaves – 개별 트리가 가질 수 있는 최대리프의 개수
: min_child_samples – min_sample_leaf 인데 이름만 바뀐 것.
: max_depth – 명시적으로 깊이의 크기를 제한

– 방안2. learning_rate를 작게하면서 , n_estimators를 크게 하는 것.
(부스팅계열 튜닝에서 가장 기본적인 튜닝방안)

4.7 LightGBM

4.7.3 파이선 래퍼 LightGBM/사이킷런 래퍼 XGBoost, LightGBM 하이퍼 파라미터 비교

유형	파이썬 래퍼 LightGBM	사이킷런 래퍼 LightGBM	사이킷런 래퍼 XGBoost
파라미터명	num_iterations	n_estimators	n_estimators
	learning_rate	learning_rate	learning_rate
	max_depth	max_depth	max_depth
	min_data_in_leaf	min_child_samples	N/A
	bagging_fraction	subsample	subsample
	feature_fraction	colsample_bytree	colsample_bytree
	lambda_l2	reg_lambda	reg_lambda
	lambda_l1	reg_alpha	reg_alpha
	early_stopping_round	early_stopping_rounds	early_stopping_rounds
	num_leaves	num_leaves	N/A
	min_sum_hessian_in_leaf	min_child_weight	min_child_weight

- 유형마다 **파라미터명이 다르구나!** 정도만 알아두면 됨.

4.7 LightGBM

4.7.4 LightGBM 적용 – 위스콘신 유방암 예측

1. 데이터 전처리

- Sklearn.datasets 에서 load_breast_cancer 데이터 이용
- 학습용 데이터(80%) 테스트용 데이터(20%)추출

2.LightGBM 모델학습

- LightGBM 객체 생성 - n_estimators는 400으로 설정
- 조기중단 수행가능하도록 early_stopping_rounds, eval_metric, eval_set 설정
- fit()과 predict() 시행

3. 예측 성능 평가

- get_clf_eval로 오차행렬, 정확도, 정밀도, 재현율, F1스코어, AUC 알아보기

4. 시각화

- plot_importance()를 이용해 피처 중요도 시각화(from lightgbm import plot_importance)

4.7 LightGBM

4.7.4 LightGBM 적용 – 위스콘신 유방암 예측

1. 데이터 전처리

- Sklearn.datasets 에서 load_breast_cancer 데이터 이용
- 학습용 데이터(80%) 테스트용 데이터(20%)추출

```
#LightGBM의 파이썬 패키지인 lightgbm에서 LGBMClassifier 임포트
from lightgbm import LGBMClassifier

import pandas as pd
import numpy as np
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split

dataset=load_breast_cancer()
ftr=dataset.data
target=dataset.target
#전체 데이터 중 80%는 학습용 데이터, 20%는 테스트용 데이터 추출
X_train,X_test,y_train,y_test=train_test_split(ftr,target,test_size=0.2,random_state=156)
```

4.7 LightGBM

4.7.4 LightGBM 적용 – 위스콘신 유방암 예측

2.LightGBM 모델학습

- LightGBM 객체 생성 - n_estimators는 400으로 설정
- 조기중단 수행가능하도록 early_stopping_rounds, eval_metric, eval_set 설정
- fit()과 predict() 시행

```
#앞서 XGBoost와 동일하기 n_estimators는 400설정
lgbm_wrapper=LGBMClassifier(n_estimators=400)

#LightGBM도 XGBoost와 동일하게중단 수행 가능.
evals=[(X_test,y_test)]
lgbm_wrapper.fit(X_train,y_train,early_stopping_rounds=100,eval_metric='logloss',eval_set=evals,verbose=True)
preds=lgbm_wrapper.predict(X_test)
```

4.7 LightGBM

4.7.4 LightGBM 적용 – 위스콘신 유방암 예측

3. 예측 성능 평가

•get_clf_eval로 오차행렬, 정확도, 정밀도, 재현율, F1스코어, AUC 알아보기

In [4]:
```
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.metrics import precision_score, recall_score
from sklearn.metrics import f1_score, roc_auc_score

def get_clf_eval(y_test , pred):
    confusion = confusion_matrix( y_test, pred)
    accuracy = accuracy_score(y_test , pred)
    precision = precision_score(y_test , pred)
    recall = recall_score(y_test , pred)
    f1 = f1_score(y_test,pred)
    roc_auc = roc_auc_score(y_test, pred)
    print('오차 행렬')
    print(confusion)
    print('정확도: {0:.4f}, 정밀도: {1:.4f}, 재현율: {2:.4f},₩
    F1: {3:.4f}, AUC:{4:.4f}'.format(accuracy, precision, recall, f1, roc_auc))
```

In [6]:
```
get_clf_eval(y_test, preds)
```

```
오차행렬
 [[33  4]
 [ 2 75]]
정확도 : 0.9474, 정밀도 : 0.9494, 재현율 : 0.9740,f1스코어 : 0.9615, AUC : 0.9330
```

4.7 LightGBM

4.7.4 LightGBM 적용 – 위스콘신 유방암 예측

4. 시각화

- plot_importance()를 이용해 피처 중요도 시각화(from lightgbm import plot_importance)

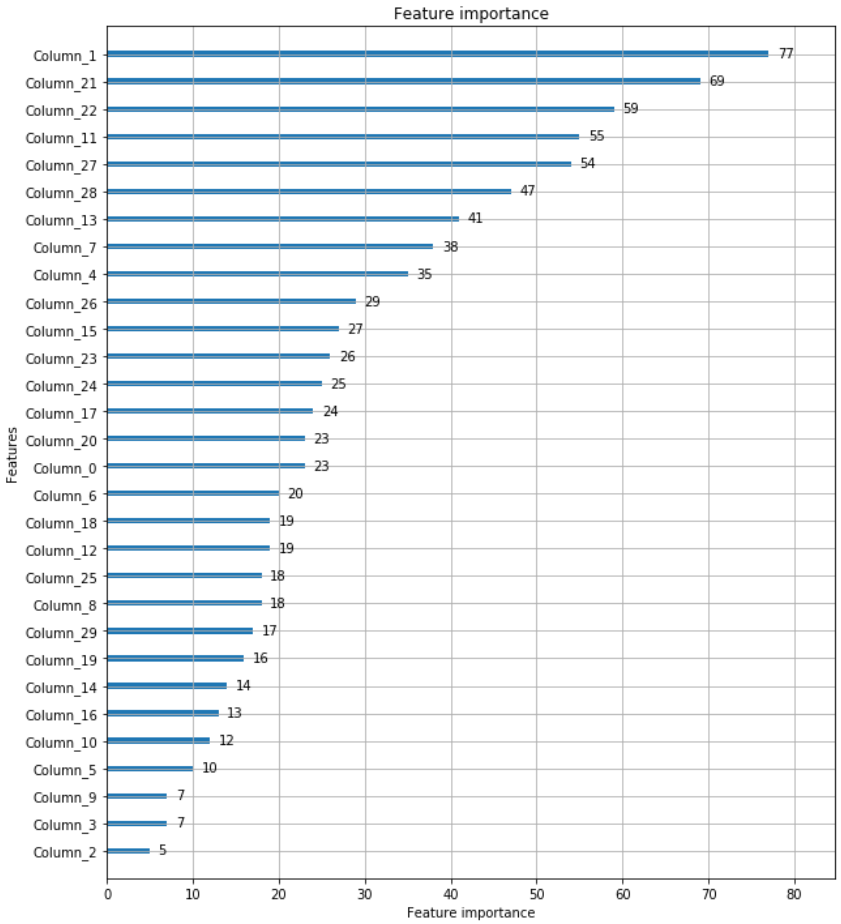
```
#plot_importance()를 이용하여 feature 중요도 시각화
from lightgbm import plot_importance
import matplotlib.pyplot as plt
%matplotlib inline

fig,ax=plt.subplots(figsize=(10,12))
plot_importance(lgbm_wrapper,ax=ax)
```

4.7 LightGBM

4.7.4 LightGBM 적용 – 위스콘신 유방암 예측

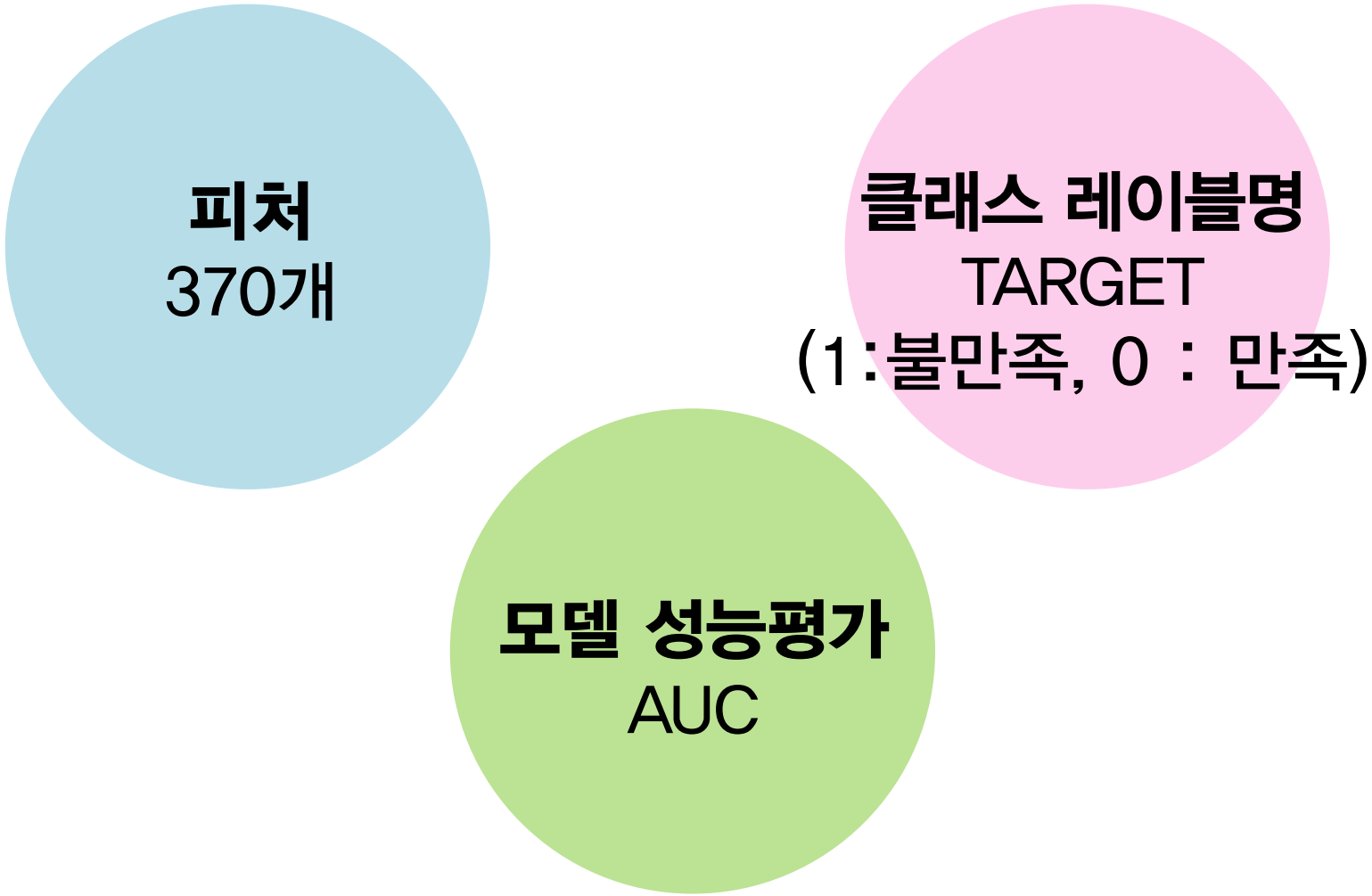
4. 시각화

•plot_importance()를 이용해 피처 중요도 시각화(from lightgbm import plot_importance)



4.8 분류실습 – 캐글 산탄데르 고객 만족예측

4.8.1 데이터 설명



4.8 분류실습 – 캐글 산탄데르 고객 만족예측

4.8.2 목표

: 고객 만족 여부를 XGBoost와 LightGBM을 활용하여 예측하자!

https://www.kaggle.com/c/santander-customer-satisfaction/data

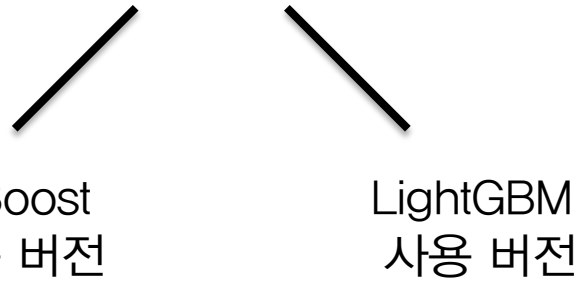
에서 데이터 santander 다운받고 시작하기!

4.8 분류실습 – 캐글 산탄데르 고객 만족예측

1차설계 – 대략적 flow만 정하기

Step1. 데이터 전처리

Step2.모델 학습과 하이퍼 파라미터 튜닝



Step3. 시각화

4.8 분류실습 – 캐글 산탄데르 고객 만족예측

2차설계 – 구체적 flow 정하기

Step1. 데이터 전처리

1. 데이터 전처리에 필요한 패키지, 모듈 import 하기
(numpy, pandas, matplotlib.pyplot, matplotlib 필요)

2. train_scatander.csv 파일 로딩하기(변수명 : cust_df)

3. data의 정보 파악(.shape , info(), value_counts(), describe()사용)
   3-1. shape사용
   3-2. info 사용 – Null 값 확인을 위하여
   3-3. value_counts() 사용 – 클래스레이블의 분포 파악을 위해
   3-4. describe() 사용 – 각 피처의 값 분포확인을 위해
   (결측값이 있는 피처 결측값 대체, 필요없는 피처 드롭하기)

4. 피처세트와 레이블 세트 분리.
(피처 세트 변수명 : X_features, 레이블 세트 변수명 : y_labels)

5. 학습 데이터세트와 테스트 데이터 세트 분리

4.8 분류실습 – 캐글 산탄데르 고객 만족예측

2차설계 – 구체적 flow 정하기

Step2.모델 학습과 하이퍼 파라미터 튜닝

모델학습
1. 필요한모듈 import하기
(XGBClassifier와 roc_auc_score 필요)

2. XGBoost 학습 모델 생성(변수명 : xgb_clf)
(n_estimators는 500으로, random_state는 156으로 설정)

3.성능평가지표를 auc로, 조기중단 파라미터는 100, eval_metric은 auc로, eval_set는 학습데이터와 테스트 데이터로 설정하고 학습 수행

4. auc 파악(변수명 : xgb_roc_score)

4.8 분류실습 – 캐글 산탄데르 고객 만족예측

2차설계 – 구체적 flow 정하기

Step2.모델 학습과 하이퍼 파라미터 튜닝

하이퍼 파라미터 수행

1. 필요한 모듈 import 하기(GridSearchCV)
하이퍼 파라미터 테스트의 수행속도를 향상시키기 위해 n_estimators=100으로 감소

2. 파라미터 설정(변수명 params – max_depth, min_child_weight, colsample_bytree 지정)

3. GridSearchCV 객체 만들기 (수행 속도 향상을 위해cv를 지정하지 않음.)
(변수명 : gridcv)

4. 학습 수행

5. 위에서 GridSearchCV로 구한 최적 하이퍼 파라미터를 이용하여 다시 최적화 진행

4.8 분류실습 – 캐글 산탄데르 고객 만족예측

2차설계 – 구체적 flow 정하기

Step3. 시각화

1. 필요한 모듈 import 하기 (plot_importance, matplotlib)

```
#plot_importance()를 이용하여 feature 중요도 시각화
from lightgbm import plot_importance
import matplotlib.pyplot as plt
%matplotlib inline
```