

# Ensemble 퀴즈

주어진 데이터를 활용하여 모델을 적합하고, 최적의 모델을 찾으세요.

- Ensemble 실습 퀴즈 (bagging / boosting / BGM)
- 모델은, RandomForestClassifier, GradientBoostingClassifier, lightgbm 을 사용하세요.
- 각 모델별로 최적의 파라미터를 찾으세요. (평가는 f1 score)
- 모델간의 평가는 ROC 값으로 하고, ROC 커브로 비교하세요.

In [23]:

```
# import library

import sklearn
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import confusion_matrix, roc_curve
```

```
import warnings
warnings.simplefilter("ignore", UserWarning)
```

In [5]:

```
filename = './data/pima-indians-diabetes.csv'
dataframe = pd.read_csv(filename, header=None)
dataframe.columns = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesP<

array = dataframe.values

X = array[:,0:8].astype(float) # 0 - 7 column은 독립변수
Y = array[:,8].astype(int) # 마지막 column은 종속변수

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=0)
```

---

## RandomForestClassifier

In [6]:

```
'''
hyperparameter를 정의하세요
'''
param_grid = {'n_estimators': [100, 200],
              'oob_score': [True], # compute out of bag error
              'n_jobs': [-1],
              'max_depth': [3, 5]
              }
```

In [7]:

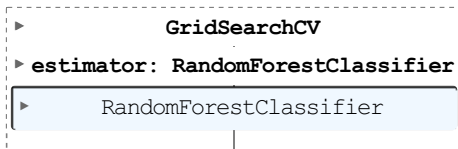
```
'''
모델을 정의하세요
'''
from sklearn.ensemble import RandomForestClassifier

rf_model = RandomForestClassifier()
```

In [8]:

```
'''
hyperparameter를 search 하세요
'''
grid_search = GridSearchCV(rf_model, param_grid=param_grid, cv=5, scoring='f1')
grid_search.fit(X_train, y_train)
```

Out[8]:



In [9]:

```
'''
최적의 파라미터로 최적의 모델을 결정하세요
'''
rf_opt = grid_search.best_estimator_
```

## GradientBoostingClassifier

In [10]:

```
'''
hyperparameter를 정의하세요
'''
param_grid = {'n_estimators': [100, 200],
              'learning_rate': [0.01, 0.001, 0.0001],
              'max_depth': [1, 3, 5]
              }
```

In [11]:

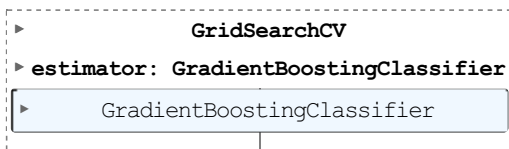
```
'''
모델을 정의하세요
'''
from sklearn.ensemble import GradientBoostingClassifier

gb_model = GradientBoostingClassifier()
```

In [12]:

```
'''
hyperparameter를 search 하세요
'''
grid_search = GridSearchCV(gb_model, param_grid=param_grid, cv=5, scoring='f1')
grid_search.fit(X_train, y_train)
```

Out[12]:



In [13]:

```
'''
최적의 파라미터로 최적의 모델을 결정하세요
'''
gb_opt = grid_search.best_estimator_
```

## lightgbm

In [15]:

```
'''
hyperparameter를 정의하세요
'''

param_dict = { "objective":['binary'], # multiclass, regression
               "max_depth": [25,50, 75],
               "learning_rate" : [0.01,0.05,0.1],
               "num_leaves": [300,900,1200],
               "n_estimators": [200]
               }
```

In [16]:

```
'''
모델을 정의하세요
```

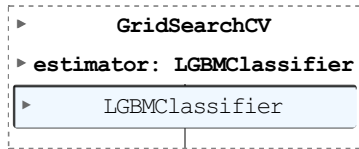
```
'''
import lightgbm as lgb

lg = lgb.LGBMClassifier(silent=True)
```

In [17]:

```
'''
hyperparameter를 search 하세요
'''
grid_search = GridSearchCV(lg, n_jobs=-1, param_grid=param_dict, cv = 3, scoring="accuracy")
grid_search.fit(X_train,y_train)
```

Out[17]:



In [18]:

```
'''
최적의 파라미터로 최적의 모델을 결정하세요
'''
lg_opt = grid_search.best_estimator_
```

## ROC 커브로 모델 비교

In [22]:

```
'''
ROC 커브로 최적의 모델을 산출하세요
'''
from sklearn.metrics import plot_roc_curve
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.gca()
plot_roc_curve(rf_opt, X_test, y_test, ax=ax)
plot_roc_curve(gb_opt, X_test, y_test, ax=ax)
plot_roc_curve(lg_opt, X_test, y_test, ax=ax)
plt.show()
```

/home/restful3/anaconda3/envs/trading/lib/python3.8/site-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot\_roc\_curve is deprecated; Function :func:`plot\_roc\_curve` is deprecated in 1.0 and will be removed in 1.2. Use one of the class methods: :meth:`sklearn.metrics.RocCurveDisplay.from\_predictions` or :meth:`sklearn.metrics.RocCurveDisplay.from\_estimator`.

warnings.warn(msg, category=FutureWarning)

/home/restful3/anaconda3/envs/trading/lib/python3.8/site-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot\_roc\_curve is deprecated; Function :func:`plot\_roc\_curve` is deprecated in 1.0 and will be removed in 1.2. Use one of the class methods: :meth:`sklearn.metrics.RocCurveDisplay.from\_predictions` or :meth:`sklearn.metrics.RocCurveDisplay.from\_estimator`.

warnings.warn(msg, category=FutureWarning)

/home/restful3/anaconda3/envs/trading/lib/python3.8/site-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function plot\_roc\_curve is deprecated; Function :func:`plot\_roc\_curve` is deprecated in 1.0 and will be removed in 1.2. Use one of the class methods: :meth:`sklearn.metrics.RocCurveDisplay.from\_predictions` or :meth:`sklearn.metrics.RocCurveDisplay.from\_estimator`.

warnings.warn(msg, category=FutureWarning)

