Experiment Objective :

- XGBoost + Random Search

Hasil dan Analisa :

- Iterasi 100 dan 50 pada RS menghasilkan nilai AUC yang sama yaitu 0.695. Hanya berbeda sedikit dibandingkan menggunakan GS yaitu 0.738. Waktu komputasi berbeda cukup jauh, untuk GS diperlukan 82 menit dengan nilai AUC 0.738 sedangkan dengan RS diperlukan 22 menit dengan nilai AUC 0.695
- Nilai RS tidak mencapai nilai GS karena nilai hyperparameter dengan AUC tertinggi pada GS tidak masuk ke dalam hyperparameter search space RS

Code:

In [17]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import cross_val_score
from xgboost import XGBClassifier
```

In [18]:
```
X_bank=pd.read_csv('dataset/X_bank_preprocessed.csv').to_numpy()
y_bank=pd.read_csv('dataset/y_bank_preprocessed.csv').to_numpy().ravel()
```

In [19]:
```
model=XGBClassifier()
```

In [29]:
```
params={
    'eta': list(np.linspace(0.001,1,10)), # learning rate
    'subsample': list(np.linspace(0,1,10)),
    'max_depth': [int(i) for i in list(np.linspace(5,50,10))],
    'gamma': list(np.linspace(0,1,10)),
    'min_child_weight': [int(i) for i in list(np.linspace(0,15,15))]
}
```

n_iter = 50

In [30]:
```
rnd_srch_clf=RandomizedSearchCV(
                    model,
                    params,
                    n_iter=50,
                    scoring='roc_auc',
                    n_jobs=-1, cv=5,
                    random_state=1)
```

In [31]:
```
rnd_srch_clf.fit(X_bank,y_bank)
```

Out[31]:
```
RandomizedSearchCV(cv=5, error_score=nan,
                   estimator=XGBClassifier(base_score=None, booster=None,
                                           colsample_bylevel=None,
                                           colsample_bynode=None,
                                           colsample_bytree=None, gamma=None,
                                           gpu_id=None, importance_type='gai
n',
                                           interaction_constraints=None,
                                           learning_rate=None,
                                           max_delta_step=None, max_depth=Non
e,
                                           min_child_weight=None, missing=na
n,
                                           monotone_constraints=None,
                                           n...
                                          'max_depth': [5, 10, 15, 20, 25, 30,
35,
                                                        40, 45, 50],
                                          'min_child_weight': [0, 1, 2, 3, 4,
5,
                                                               6, 7, 8, 9, 10,
11,
                                                               12, 13, 15],
                                          'subsample': [0.0, 0.111111111111111
1,
                                                        0.2222222222222222,
                                                        0.3333333333333333,
                                                        0.4444444444444444,
                                                        0.5555555555555556,
                                                        0.6666666666666666,
                                                        0.7777777777777777,
                                                        0.8888888888888888,
                                                        1.0]},
                   pre_dispatch='2*n_jobs', random_state=1, refit=True,
                   return_train_score=False, scoring='roc_auc', verbose=0)
```

In [32]:
```
index=rnd_srch_clf.best_index_
print("Best params: ")
print(rnd_srch_clf.best_params_)
print("AUC: ")
print(rnd_srch_clf.cv_results_['mean_test_score'][index])
print("std: ")
print(rnd_srch_clf.cv_results_['std_test_score'][index])
```

```
Best params:
{'subsample': 0.2222222222222222, 'min_child_weight': 10, 'max_depth': 20, 'g
amma': 0.6666666666666666, 'eta': 0.001}
AUC:
0.6957506053674708
std:
0.15371171277770676
```

n_iter = 100

In [35]:
```
rnd_srch_clf_2=RandomizedSearchCV(
                    model,
                    params,
                    n_iter=100,
                    scoring='roc_auc',
                    n_jobs=-1, cv=5,
                    random_state=1)
```

In [36]:
```
rnd_srch_clf_2.fit(X_bank,y_bank)
```

Out[36]:
```
RandomizedSearchCV(cv=5, error_score=nan,
                   estimator=XGBClassifier(base_score=None, booster=None,
                                           colsample_bylevel=None,
                                           colsample_bynode=None,
                                           colsample_bytree=None, gamma=None,
                                           gpu_id=None, importance_type='gai
n',
                                           interaction_constraints=None,
                                           learning_rate=None,
                                           max_delta_step=None, max_depth=Non
e,
                                           min_child_weight=None, missing=na
n,
                                           monotone_constraints=None,
                                           n...
                                         'max_depth': [5, 10, 15, 20, 25, 30,
35,
                                                       40, 45, 50],
                                         'min_child_weight': [0, 1, 2, 3, 4,
5,
                                                              6, 7, 8, 9, 10,
11,
                                                              12, 13, 15],
                                         'subsample': [0.0, 0.111111111111111
1,
                                                       0.2222222222222222,
                                                       0.3333333333333333,
                                                       0.4444444444444444,
                                                       0.5555555555555556,
                                                       0.6666666666666666,
                                                       0.7777777777777777,
                                                       0.8888888888888888,
                                                       1.0]},
                   pre_dispatch='2*n_jobs', random_state=1, refit=True,
                   return_train_score=False, scoring='roc_auc', verbose=0)
```

In [37]:
```
index=rnd_srch_clf_2.best_index_
print("Best params: ")
print(rnd_srch_clf_2.best_params_)
print("AUC: ")
print(rnd_srch_clf_2.cv_results_['mean_test_score'][index])
print("std: ")
print(rnd_srch_clf_2.cv_results_['std_test_score'][index])
```

```
Best params:
{'subsample': 0.2222222222222222, 'min_child_weight': 10, 'max_depth': 20, 'g
amma': 0.6666666666666666, 'eta': 0.001}
AUC:
0.6957506053674708
std:
0.15371171277770676
```

Hyperparameter Search Space

In [40]:
```
print('eta : ',list(np.linspace(0.001,1,10))) # learning rate
print('subsample : ',list(np.linspace(0,1,10)))
print('max_depth : ',[int(i) for i in list(np.linspace(5,50,10))])
print('gamma : ',list(np.linspace(0,1,10)))
print('min_child_weight : ', [int(i) for i in list(np.linspace(0,15,15))])
```

```
eta :  [0.001, 0.112, 0.223, 0.334, 0.445, 0.556, 0.667, 0.778, 0.889, 1.0]
subsample :  [0.0, 0.1111111111111111, 0.2222222222222222, 0.333333333333333
3, 0.4444444444444444, 0.5555555555555556, 0.6666666666666666, 0.777777777777
7777, 0.8888888888888888, 1.0]
max_depth :  [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
gamma :  [0.0, 0.1111111111111111, 0.2222222222222222, 0.3333333333333333, 0.
4444444444444444, 0.5555555555555556, 0.6666666666666666, 0.7777777777777777,
0.8888888888888888, 1.0]
min_child_weight :  [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15]
```

In [ ]: