Experiment Objective :

- XGBoost + Grid Search

Hasil dan Analisa :

- XGBoost + GS lebih baik dibandingkan XGBoost menggunakan default hyperparameter
- Nilai AUC dari XGBoost + GS -> 0.738
- Nilai AUC dari XGBoost + default hyperparameter -> 0.483

In [26]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import cross_val_score
from xgboost import XGBClassifier
```

In [5]:
```
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()
```

XGBoost default hyperparameter

In [7]:
```
model=XGBClassifier()
```

In [9]:
```
skf=StratifiedKFold(n_splits=5)
results=cross_val_score(model,X_bank,y_bank,cv=skf,scoring="roc_auc")
print(f"AUC: {round(results.mean(),4)}, std: {round(results.std(),4)}")
```

```
AUC: 0.4837, std: 0.1006
```

XGBoost + GridSearch

In [11]:
```
params={
    'eta': [0.001,0.01,0.1], # learning rate
    'subsample': [0.1,0.4,0.8],
    'max_depth': [10,20,30],
    'gamma':[0.1,0.4,0.8],
    'min_child_weight':[2,5,11]
}
```

In [12]:
```
grid_search_clf = GridSearchCV(
    estimator=model,
    param_grid=params,
    scoring = 'roc_auc',
    n_jobs = -1,
    cv = 5,
    verbose= 1
)
```

Bersumber dari https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html), default metode cv yang digunakan ketika input value berupa integer adalah StratifiedKFold

In [13]:
```
grid_search_clf.fit(X_bank,y_bank)
```

```
Fitting 5 folds for each of 243 candidates, totalling 1215 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  34 tasks      | elapsed:  1.7min
[Parallel(n_jobs=-1)]: Done 184 tasks      | elapsed: 12.0min
[Parallel(n_jobs=-1)]: Done 434 tasks      | elapsed: 30.0min
[Parallel(n_jobs=-1)]: Done 784 tasks      | elapsed: 53.4min
[Parallel(n_jobs=-1)]: Done 1215 out of 1215 | elapsed: 82.1min finished
```

Out[13]:
```
GridSearchCV(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='gain',
                                     interaction_constraints=None,
                                     learning_rate=None, max_delta_step=None,
                                     max_depth=None, min_child_weight=None,
                                     missing=nan, monotone_constraints=None,
                                     n_estim...
                                     reg_lambda=None, scale_pos_weight=None,
                                     subsample=None, tree_method=None,
                                     validate_parameters=None, verbosity=Non
e),
             iid='deprecated', n_jobs=-1,
             param_grid={'eta': [0.001, 0.01, 0.1], 'gamma': [0.1, 0.4, 0.8],
                         'max_depth': [10, 20, 30],
                         'min_child_weight': [2, 5, 11],
                         'subsample': [0.1, 0.4, 0.8]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring='roc_auc', verbose=1)
```

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

```
Best params:
{'eta': 0.01, 'gamma': 0.4, 'max_depth': 10, 'min_child_weight': 11, 'subsamp
le': 0.1}
AUC:
0.7382674790807361
std:
0.12269998614087489
```