ASSIGNMENT 2

Christos Eleftheriou

1009537

Data Preparation:

For this exercise I removed NAs and duplicates from the datasets.

I encoded Salary and Sex using 1 and 0, I left hours per week as it was and I used ordinal encoding to encode education. I removed the rest of the columns.

Part A – Preparation

finetune and fit_and_evaluate

```
def finetune(clf, grid_param, rand_param, X, Y):
    search_model = GridSearchCV(clf, grid_param,scoring='roc_auc', cv=2)
    search_model.fit(X,Y)
    grid_best=search_model.best_params_
    print("GridSearch - Best Hyperparameters:",grid_best)

    search_model2 = RandomizedSearchCV(estimator = clf,param_distributions = rand_param, scoring="roc_auc",cv = 2, n_iter=50)
    search_model2.fit(X,Y)
    rand_best=search_model2.best_params_
    print("RandomizedSearch - Best Hyperparameters:",rand_best)
    return grid_best , rand_best

def fit_and_evaluate(clf, X_train, y_train, X_test, y_test):
    clf.fit(X_train, y_train)
    prediction=clf.predict(X_test)
    clf_pred_proba=clf.predict_proba(X_test)[:,1]
    fpr, tpr, _ = metrics.roc_curve(y_test,  clf_pred_proba)
    auc = metrics.roc_auc_score(y_test, clf_pred_proba)
    return fpr, tpr, auc
```

For Bagging Classifier

```
GridSearch - Best Hyperparameters: {'n_estimators': 100}
```

```
RandomizedSearch - Best Hyperparameters: {'n_estimators': 45}
```

For Random Forest Classifier

```
GridSearch - Best Hyperparameters: {'n_estimators': 150}
```

```
RandomizedSearch - Best Hyperparameters: {'n_estimators': 20}
```

For XGB Classifier

```
GridSearch - Best Hyperparameters: {'n_estimators': 200}
```

```
RandomizedSearch - Best Hyperparameters: {'n_estimators': 41}
```

Model Settings

```
search_model = GridSearchCV(clf, grid_param,scoring='roc_auc', cv=2)
```

```
search_model2 = RandomizedSearchCV(estimator = clf,param_distributions = rand_param, scoring="roc_auc",cv = 2, n_iter=50)
```

Part B – Model Selection

B1.

1.

a)
```
n_estimators = np.linspace(start=1, stop=50, num=50, dtype=int)
g_parameters = {'n_estimators':[5,10,25,50,100,150,200]}
r_parameters = {'n_estimators':n_estimators}
```

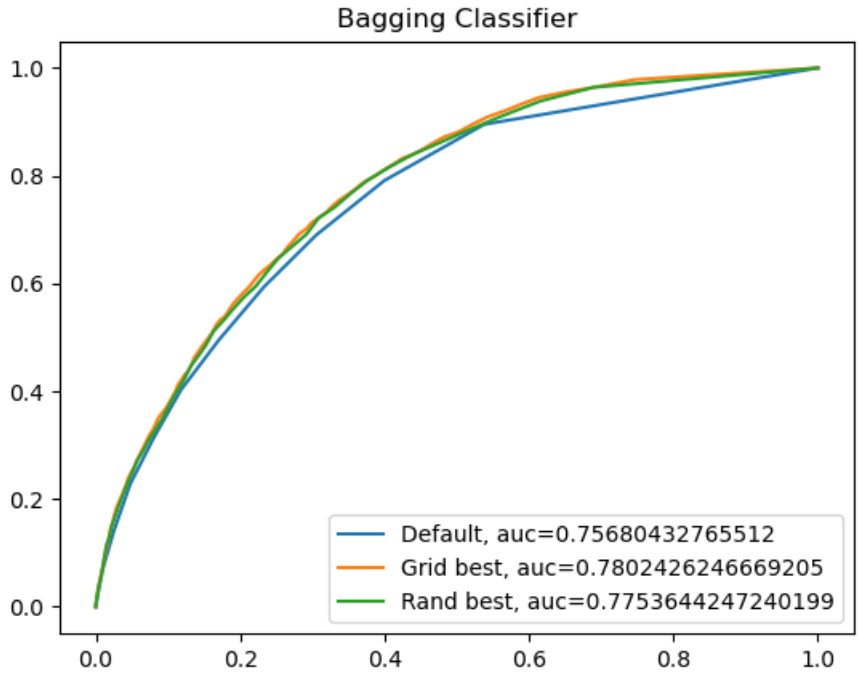
For this exercise I chose to investigate different numbers of estimators for the Grid Search and Randomized Search.

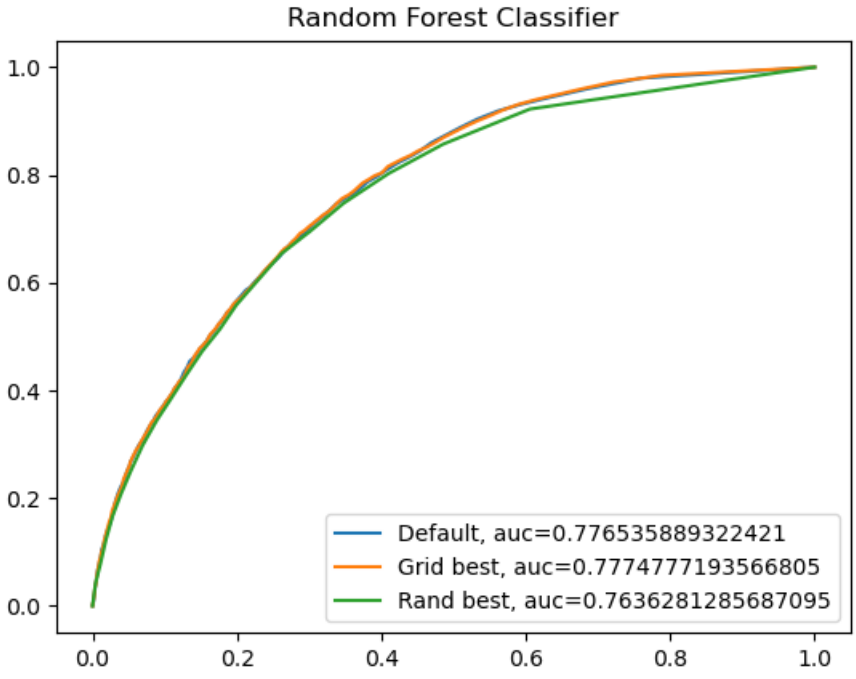
b) For the GridSearch I chose to test 7 numbers of estimators ([5, 10, 25, 50, 100, 150, 200]), so that a wide range of values is covered without checking every possible value in between. These values were selected because they are reasonable and commonly used numbers for this purpose that usually provide good results. Also, by including values from 5 to 200 we can see and measure the performance of the model when it is simple and when it is more complicated, and find the value for which the model has the best performance. For the RandomizedSearch values between 1-50 will be randomly selected and tested.

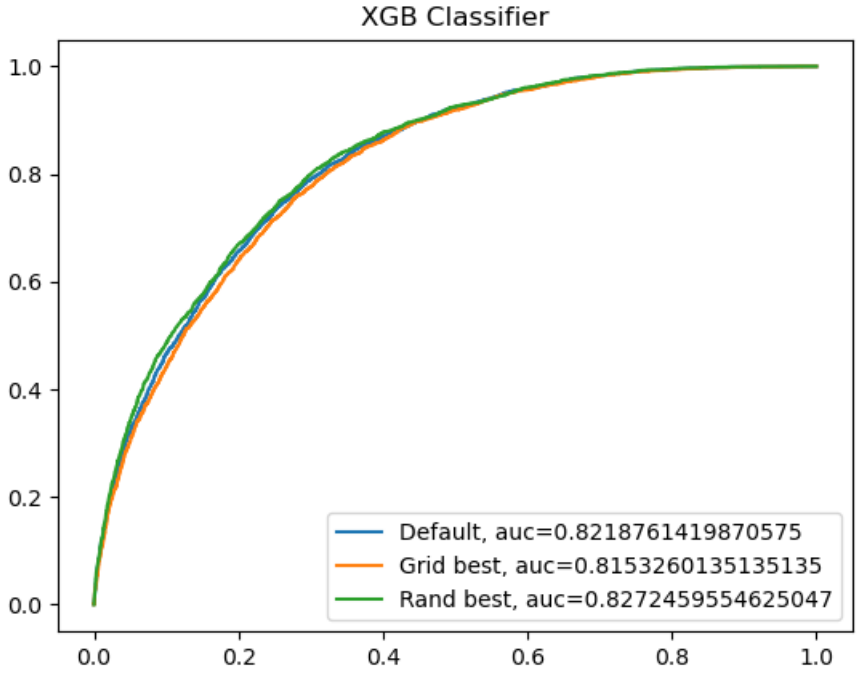
c) In most cases I would expect the Randomized Search to find the best hyperparameter because it checks values that we are usually not able to predict are the best. In this specific case, in Randomized Search values between 1-50 are checked randomly, whereas Grid Search checks specific values above 50 (100, 150, 200). If the best hyperparameter combination is in the space of 1-50 then Randomized Search will most likely find the better one, but if the best hyperparameter is higher then this might not be the case.The trade-off between Grid Search and Randomized Search mainly revolves around the balance between exhaustiveness and computational efficiency, meaning that we can check many

values with Grid Search but have less computational efficiency, or check randomly fewer values hoping to find the best hyperparameter.

3.





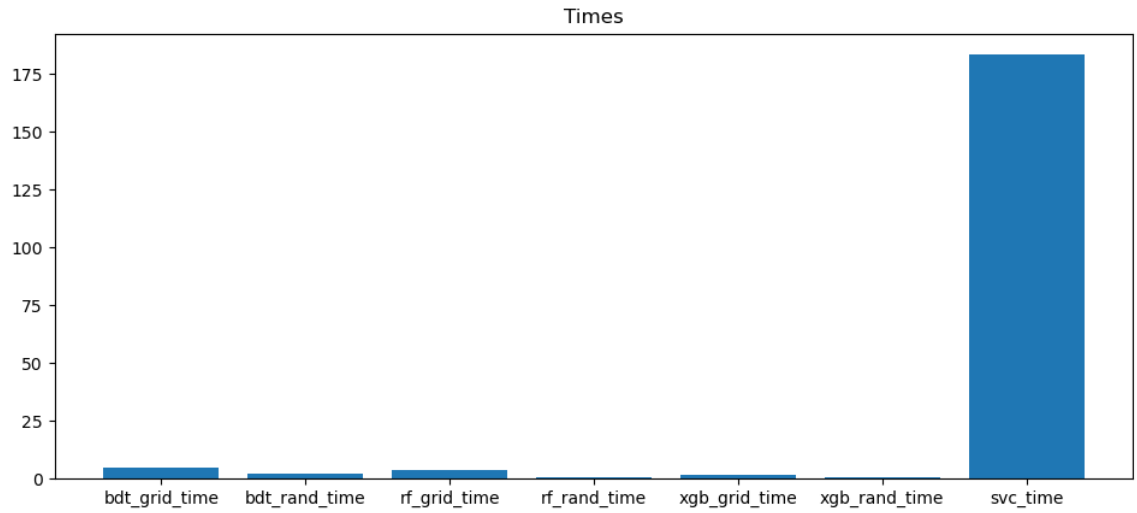


We can see that, for the Bagging Classifier the GridSearch hyperparameters provide the best result as the AUC is bigger than the other choices. For the Random Forrest Classifier, the GridSearch provides the best hyperparameters by a small margin, and for the XGB Classifier the RandomizedSearch gives the best hyperparameters. Generally, the XGB Classifier better results with the default settings, the GridSearch settings and the RandomizedSearch settings than the other two classifiers.

B2.

Based on B1, the best performing hyperparameters for BDT, RF and XGB come from Grid Search, Grid Search and Randomized Search respectively.

```
Best bdt:  grid  Best rf:  grid  Best xgb:  rand
```



The first thing that we can see on the plot is that the execution time of the SVC Classifier is much higher than the rest. We can also see that, generally, the execution time of the classifiers using hyperparameters from the Grid Search is higher than them using hyperparameters from the Randomized Search. Finally, we can see that the XGB Classifier has the smallest computational time of them all.