University of Cyprus

MAI612 - MACHINE LEARNING
Assignment 2 – Model Selection, Trees & Kernels

Chrysis Andreou (UC1366020)

B1.1.B_These parameters are chosen to balance the trade-off between model complexity and computational efficiency, allowing the model to learn effectively from the data while avoiding overfitting. The grid search uses a smaller, more focused range to thoroughly explore specific values, while the randomized search uses a broader range to quickly explore a wider space.

B1.1.C_Grid Search is expected to find the best hyperparameter combinations due to its exhaustive evaluation of all specified options, but it trades off efficiency and computational resources compared to Randomized Search.

Tuning Random Forest:
Best parameters from GridSearchCV: {'max_depth': 10, 'min_samples_split': 20, 'n_estimators': 100}
Best parameters from RandomizedSearchCV: {'n_estimators': 75, 'min_samples_split': 15, 'max_depth': 10}

Tuning Bagging Decision Tree:
Best parameters from GridSearchCV: {'bootstrap': True, 'max_features': 0.5, 'max_samples': 0.5, 'n_estimators': 200}
Best parameters from RandomizedSearchCV: {'n_estimators': 100, 'max_samples': 0.3, 'max_features': 0.3, 'bootstrap': True}

Tuning XGBoost:
Best parameters from GridSearchCV: {'learning_rate': 0.1, 'max_depth': 3, 'min_child_weight': 1, 'n_estimators': 100}
Best parameters from RandomizedSearchCV: {'n_estimators': 100, 'min_child_weight': 1, 'max_depth': 3, 'learning_rate': 0.05}

B1.2-3_FIT AND EVALUATE AND PLOT
Comparison of different settings for each model:

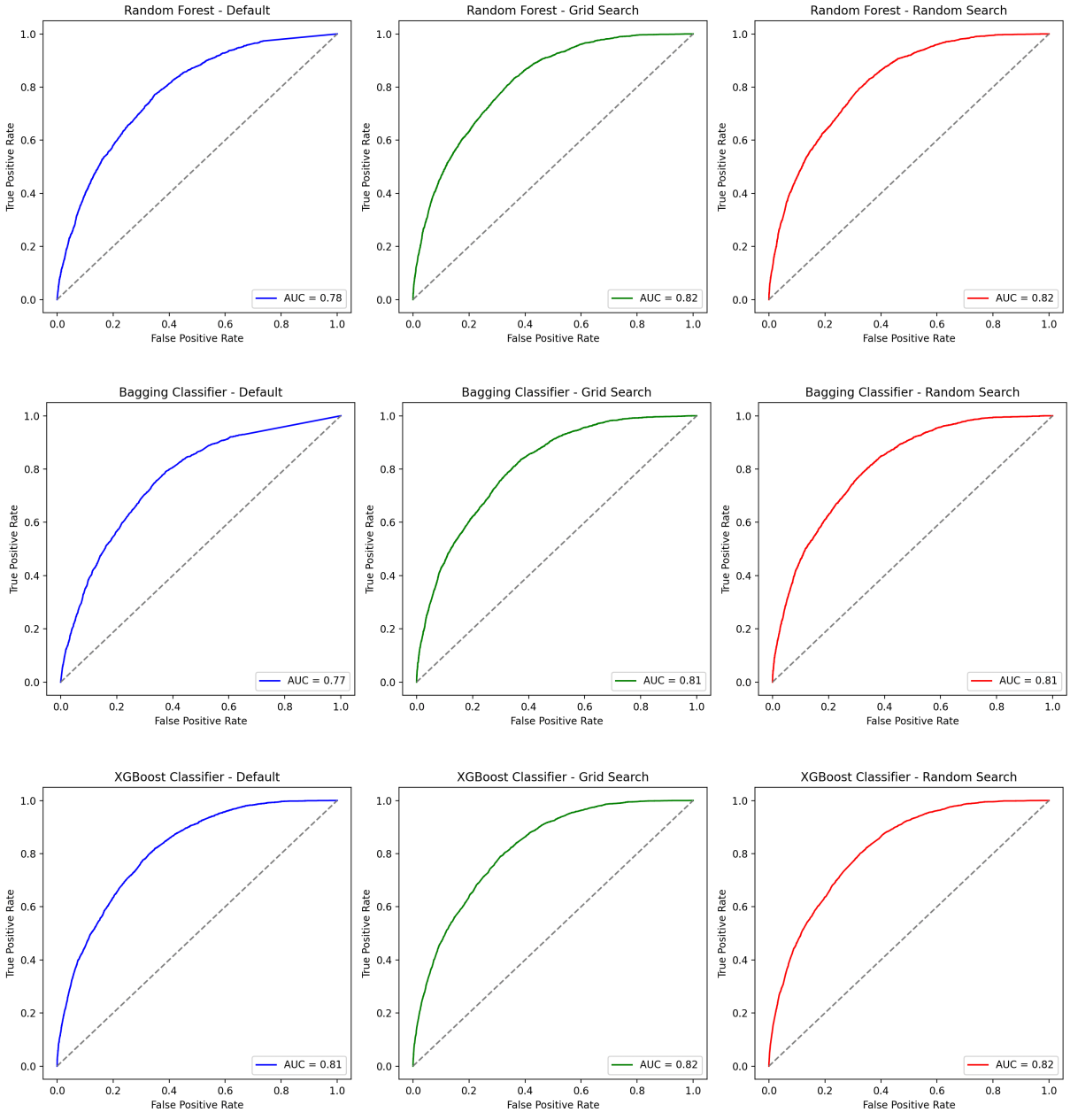
Random Forest: Default AUC: 0.7803, Grid Search AUC: 0.8163, Random Search AUC: 0.8161
Random Forest: Grid Search improved performance by 4.61% over default.
Random Forest: Random Search improved performance by 4.58% over default.

Bagging Decision Tree: Default AUC: 0.7674, Grid Search AUC: 0.8101, Random Search AUC: 0.8120
Bagging Decision Tree: Grid Search improved performance by 5.57% over default.
Bagging Decision Tree: Random Search improved performance by 5.81% over default.

XGBoost: Default AUC: 0.8142, Grid Search AUC: 0.8193, Random Search AUC: 0.8191
XGBoost: Grid Search improved performance by 0.63% over default.

XGBoost: Random Search improved performance by 0.60% over default.

Comparison of the best version of each model:
Best Random Forest AUC: 0.8163
Best Bagging Decision Tree AUC: 0.8120
Best XGBoost AUC: 0.8193
XGBoost performed best overall.
The best model outperformed the second-best by 0.38%



B.2 - Measuring training and evaluation time
Random Forest - Training time: 0.61s, Evaluation time: 0.06s
Bagging Decision Tree - Training time: 0.42s, Evaluation time: 0.07s
XGBoost - Training time: 0.12s, Evaluation time: 0.01s
SVC - Training time: 117.51s, Evaluation time: 15.88s

Explanation of findings regarding execution time:

1. XGBoost: Fastest overall (train: 0.12s, eval: 0.01s)
  - Highly optimized, leveraging gradient boosting and parallel processing

2. Bagging Decision Tree: Second fastest (train: 0.42s, eval: 0.07s)
  - Efficient due to parallel training of base estimators

3. Random Forest: Close third (train: 0.61s, eval: 0.06s)
  - Slightly slower in training, faster in evaluation than BDT

4. SVC: Significantly slower (train: 117.51s, eval: 15.88s)
  - About 964x slower in training and 2658x slower in evaluation than XGBoost
  - Higher time complexity, especially for larger datasets

Key Takeaway: Tree-based ensemble methods (XGBoost, RF, BDT) are significantly faster than SVC for this dataset, with XGBoost being the most time-efficient. Consider this trade-off between performance and speed when selecting classifiers for large-scale or time-sensitive applications.