

Problem Set 5: Trees, Forests, and Fairness in ML

Your mission for this problem set is to use your knowledge of tree-based methods and supervised learning to -- among other things! -- explore issues of fairness in machine learning (ML). Unlike in previous psets, where we were implementing algorithms from scratch, this pset will rely more heavily on sklearn. Unless explicitly noted otherwise, you are allowed (encouraged!) to make good use of this wonderful library.

This problem set will also rely on the resources provided by the [folktables](#) project. Before getting started, take some time to understand what folktables is about and how it can be used to benchmark ML algorithms in social science. If you don't understand the basics, this pset will be very challenging!!

Some resources you might find useful:

[Folktables paper](#)

[Folktables video](#)

[PUMS_Data_Dictionary_2018](#)

[Fairness in ML](#)

Part 0

Question 1 - upload your ipynb to bcourses

Question 2 - upload your pdf to bcourses

Question 3 - please provide a summary of any resources consulted, and people with whom you worked in the completion of the problem set.

Highlight any specific LLM prompts/resources used at the point of use.

Part I: Setup of prediction task

Question 4: Define the income prediction task

For this first question we are providing the code. Run it to download the data you'll be using throughout this problem set.

Begin by answering the following questions:

- Where do the data come from?
- What is the sample?
- What are we trying to predict?
- What are the features that we will be using?
- What is the "group" feature?

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import folktables
import seaborn as sns
```

```
In [131]: from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV, StratifiedKFold, cross_val
from sklearn.metrics import precision_recall_curve
from scipy.optimize import minimize_scalar
```

```
In [3]: from folktables import ACSDataSource, generate_categories

def adult_filter(data):
    """Mimic the filters in place for Adult data.

    Adult documentation notes: Extraction was done by Barry Becker from
    the 1994 Census database. A set of reasonably clean records was extracted
    using the following conditions:
    ((AAGE>16) && (AGI>100) && (AFNLWGT>1)&& (HRSWK>0))
    """
    df = data
    df = df[df['AGEP'] > 16]
    df = df[df['PINC'] > 100]
    df = df[df['WKHP'] > 0]
    df = df[df['PWGTP'] >= 1]
    return df

ACSIIncome = folktables.BasicProblem(
    features=[
        'AGEP',
        'COW',
        'SCHL',
```

```

        'MAR',
        'POBP',
        'RELP',
        'WKHP',
        'SEX',
        'RAC1P',
    ],
    target='PINCP',
    target_transform=Lambda x: x > 50000,
    group='RAC1P',
    preprocess=adult_filter,
    postprocess=Lambda x: np.nan_to_num(x, -1),
)

data_source = ACSDataSource(survey_year='2018', horizon='1-Year', survey='pe
acs_data = data_source.get_data(states=["CA"], download=True)
definition_df = data_source.get_definitions(download=True)
categories = generate_categories(features=ACSIncome.features, definition_df=
features, target, group = ACSIncome.df_to_pandas(acs_data)

feature_names = ACSIncome.features
target_name = ACSIncome.target
group_name = ACSIncome.group

```

Question 5: Split into train and test

Split your data into 80%-20% train and test splits.

```

In [4]: X_train, X_test, y_train, y_test = train_test_split(features, target, test_s
        random_state=42, stratify=target)
# X_train, X_test, y_train, y_test, group_train, group_test = train_test_spl
#     random_state=42, stratify=target)

```

Question 6: Data Cleaning

Remember to identify which features are best represented as numerical data types, and which ones are best represented as categorical data types.

- Set each feature to its desired data type in both train and test splits. Make sure that the target variable in both splits is numeric before moving ahead.

```

In [5]: num_feats = ['AGEP', 'WKHP']
        cat_feats = ['COW', 'SCHL', 'MAR', 'POBP', 'RELP', 'SEX', 'RAC1P']

```

```

In [6]: for col in num_feats:
        X_train[col] = X_train[col].astype(float)
        X_test[col] = X_test[col].astype(float)

        for col in cat_feats:
            X_train[col] = X_train[col].astype('category')
            X_test[col] = X_test[col].astype('category')

```

```
In [7]: y_train = y_train.astype(int)
        y_test = y_test.astype(int)
```

Question 7: Entropy and Information Gain

As we discussed in lecture, decision trees (and their variants) recursively split your dataset into smaller and smaller samples. Let's review how these splits are placed!

For this exercise, consider a really simple decision tree, with only 1 feature: WKHP. Given our income prediction task (defined above), your task is to determine whether it would be better to split this feature at WKHP = 40, or WKHP = 50.

Your tasks:

- Complete the function *weighted_entropy*. The function takes as input a feature, a split point for the feature, and the target. The function returns the entropy (as defined and discussed in the lectures) for that particular split.
- Using this function, calculate and report the entropy gain from 1) placing a split at WKHP = 40 and 2) placing a split at WKHP = 50.
- State your conclusion: is it better (in terms of information gain) to split the feature WKHP at the value 40 or the value 50?

Hint: review pages 25 - 30 in Lecture 12: Decision Trees

```
In [8]: def weighted_entropy(x, y, split_point):
        """
        x: a column vector of dimensions N X 1 (the feature)
        y: a column vector of dimensions N x 1 (the target variable)
        split_point: an integer in the range [min(x), max(x)], which indicates w
        """
        left_mask = x <= split_point
        right_mask = x > split_point

        y_left = y[left_mask]
        y_right = y[right_mask]

        n = len(y)
        n_left = len(y_left)
        n_right = len(y_right)

        ent_left = compute_entropy(y_left)
        ent_right = compute_entropy(y_right)

        weighted_ent = (n_left / n) * ent_left + (n_right / n) * ent_right

        return weighted_ent

def compute_entropy(x):
```

```

if len(x) == 0:
    return 0
p1 = np.mean(x)
p0 = 1 - p1
h = 0
for p in [p0, p1]:
    if p > 0:
        h -= p * np.log2(p)
return h

```

```

In [9]: base_h = compute_entropy(y_train)

h_40 = weighted_entropy(X_train['WKHP'], y_train, 40) #split==40
gain_40 = base_h - h_40

h_50 = weighted_entropy(X_train['WKHP'], y_train, 50)
gain_50 = base_h - h_50

print(f"Information gain at split=40: {gain_40}")
print(f"Information gain at split=50: {gain_50}")

```

Information gain at split=40: 0.054740648125942926
Information gain at split=50: 0.014932182601868571

Part II: ML pipeline for a Classification Tree

Next, you'll build an ML pipeline using sklearn. Take some time to familiarize yourself with [sklearn.pipeline.Pipeline](#) before proceeding.

Question 8: Pre-processing of features

The first step of the pipeline will standardize the numeric features and one-hot encode the categorical features. Write the code for this preprocessing step below.

Hint: Take a look at sklearn's ColumnTransformer, OneHotEncoder and StandardScaler for this task.

```

In [10]: #preprocessing
num_transformer = Pipeline(steps=[('scaler', StandardScaler())])
cat_transformer = Pipeline(steps=[('onehot', OneHotEncoder(handle_unknown='ignore'))])

#in the next step, I transform them all into a single ColumnTransformer
preprocessor = ColumnTransformer(transformers=[('num', num_transformer, num_features),
                                              ('cat', cat_transformer, cat_features)])

```

Question 9: Use your pre-processed data to fit a basic DecisionTreeClassifier

Now, bring together the pre-processing step with a Decision Tree Classifier in a pipeline. For now, use the default values; we'll take care of hyperparameter optimization later.

- Fit the pipeline on the training data.
- How deep is the resulting tree? How many leafs are in the tree?
- Print the 10 variables with highest feature importance and tell us what those importances mean in plain English.
- Plot the top 3 levels of the fitted decision tree. Make sure that the nodes are appropriately labeled.

```
In [11]: #full pipeline:: >> ::preprocessing + ::decision tree
clf_pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('classifier',
clf_pipeline.fit(X_train, y_train)

tree_model = clf_pipeline.named_steps['classifier']
```

```
In [12]: print("Tree depth:", tree_model.get_depth())
print("No of leaves:", tree_model.get_n_leaves())
```

Tree depth: 78
No of leaves: 38134

```
In [13]: #feature names
ohe_feats = clf_pipeline.named_steps['preprocessor'].named_transformers_['ca
ohe_names = ohe_feats.get_feature_names_out(cat_feats)

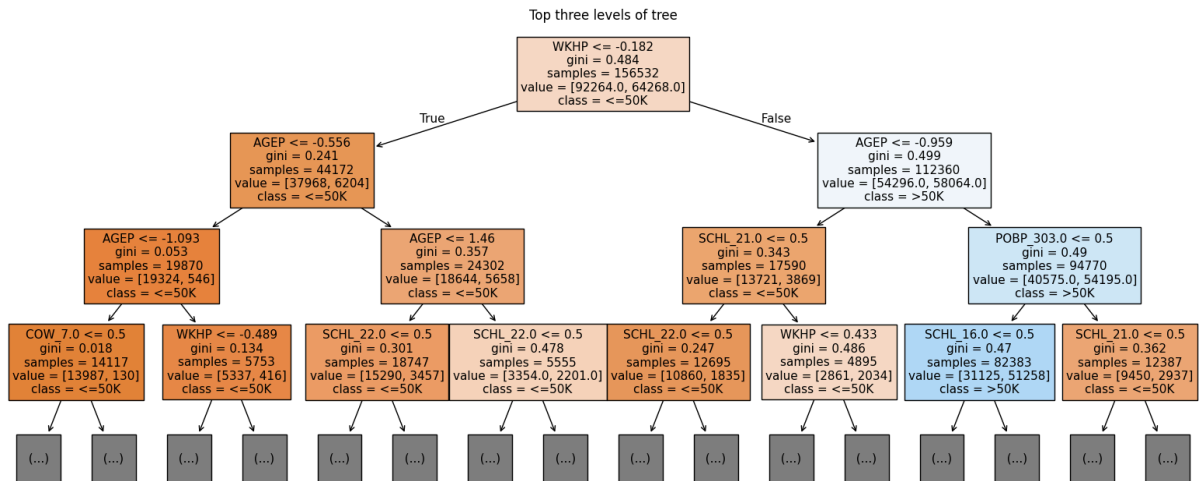
all_feat_names = np.concatenate([num_feats, ohe_names])
```

```
In [14]: importances = tree_model.feature_importances_
top_idx = np.argsort(importances)[::-1][:10]

print("Top 10 variables by importance:")
for idx in top_idx:
    print(f"{all_feat_names[idx]}: {importances[idx]:.4f}")
```

Top 10 variables by importance:
AGEP: 0.2350
WKHP: 0.2009
POBP_303.0: 0.0482
SCHL_21.0: 0.0323
SCHL_22.0: 0.0270
SCHL_16.0: 0.0245
SCHL_23.0: 0.0179
POBP_6.0: 0.0161
RELP_0.0: 0.0158
COW_1.0: 0.0155

```
In [15]: plt.figure(figsize=(20, 8))
tree.plot_tree(tree_model,max_depth=3, feature_names=all_feat_names, class_r
plt.title("Top three levels of tree")
plt.show()
```



Question 10: Performance Metrics

Let's evaluate the performance of your classifier. Generate predictions from your fitted model, and calculate the following metrics (on the training dataset):

- accuracy
- precision
- recall
- f1-score

Do not use any pre-packaged or canned functions from sklearn to calculate these metrics. Rather, implement the necessary code to calculate these from scratch. You can continue using DecisionTreeClassifier to generate predictions etc.

```
In [16]: y_pred = clf_pipeline.predict(X_train)
print('Done with the prediction >> Next, confusion matrices')
```

Done with the prediction >> Next, confusion matrices

```
In [ ]: #first i used this but it took lots of time so in the next cell i tried using

y_true = np.array(y_train)
y_pred = np.array(y_pred)

TP = np.sum((y_true == 1)*(y_pred == 1))
TN = np.sum((y_true == 0)*(y_pred == 0))
FP = np.sum((y_true == 0)*(y_pred == 1))
FN = np.sum((y_true == 1)*(y_pred == 0))

accuracy = (TP + TN) / len(y_true)
precision = TP / (TP + FP) if (TP + FP) > 0 else 0
recall = TP / (TP + FN) if (TP + FN) > 0 else 0
f1 = 2 * (precision * recall) / (precision + recall) if (precision + recall) > 0 else 0
```

```
In [21]: y_true = np.array(y_train).ravel()
y_pred = np.array(y_pred).ravel()
```

```

conf_mat = y_true * 2 + y_pred
t_counts = np.bincount(conf_mat, minlength=4)

TN, FP, FN, TP = t_counts[0], t_counts[1], t_counts[2], t_counts[3]

total = len(y_true)
accuracy = (TP + TN) / total
precision = TP / (TP + FP) if TP + FP else 0
recall = TP / (TP + FN) if TP + FN else 0
f1 = 2 * precision * recall / (precision + recall) if precision + recall else 0

```

```

In [22]: print(f"Accuracy : {accuracy}")
         print(f"Precision: {precision}")
         print(f"Recall   : {recall}")
         print(f"F1-score : {f1}")

```

```

Accuracy : 0.964857026039404
Precision: 0.974179805380283
Recall   : 0.9393010518453974
F1-score : 0.9564225452529013

```

Question 11: Estimate AUC using K-Fold CV

Calculate the 5-fold cross-validated AUC-ROC for this simple tree pipeline.

- Report the score for each fold as well as the average across all folds.

Hint 1: `sklearn.model_selection.cross_val_score` is your friend. Setting the verbose option to 3 is useful.

```

In [23]: cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
         scores = cross_val_score(clf_pipeline, X_train, y_train, cv=cv, scoring='roc_auc')

```

```

[CV] END ..... score: (test=0.735) total time=7.9s
[CV] END ..... score: (test=0.737) total time=8.0s
[CV] END ..... score: (test=0.742) total time=8.0s
[CV] END ..... score: (test=0.737) total time=7.8s
[CV] END ..... score: (test=0.739) total time=7.9s

```

```

In [24]: for i, score in enumerate(scores, 1):
         print(f"Fold {i}: AUC-ROC = {score:.4f}")

         print(f"Average AUC-ROC: {np.mean(scores):.4f}")

```

```

Fold 1: AUC-ROC = 0.7346
Fold 2: AUC-ROC = 0.7367
Fold 3: AUC-ROC = 0.7417
Fold 4: AUC-ROC = 0.7367
Fold 5: AUC-ROC = 0.7391
Average AUC-ROC: 0.7378

```


Question 12: Estimating AUC using nested CV [extra-credit]

Let's try to improve on the results of the decision tree by tweaking its hyperparameters. Since you already are an expert using nested CV, this is going to be a walk in the park! Compute the nested cv AUC-ROC of a decision tree pipeline. Use 5 folds for the inner loop and 3 folds for the outer loop. Include different values for "max_depth", "max_features" and "max_leave_nodes" in your hyperparameter grid.

- Ensure that you print out the inner and outer fold scores for all combinations of hyperparams.
- Report the AUC-ROC of a decision tree pipeline (average over outer fold scores)
- Report the best hyperparameters used in each outer fold.
- Interpret your results:
 - Are they stable?
 - To what extent do you think specific hyperparameters might lead to overfitting?
 - How is run-time impacted by your choice of hyperparameters?
 - Is hyperparameter optimization worth the trouble?

Hint 1: The pipeline remains the same. You can access the list of all the hyperparams in your pipeline with `pipeline.get_params_keys()`. Notice the naming convention.

Hint 2: `GridSearchCV` and `cross_val_score` are your friends-- especially if you run them with `verbose = 3`.

Hint 3: This gets computationally expensive quickly. Be intelligent about the hyperparameter values you include in the grid and the number of outer and inner folds you use (we use 5 inner folds and 3 outer folds).

```
In [25]: param_grid = {'classifier__max_depth': [3, 5, 10],
                      'classifier__max_features': [None, 'sqrt', 'log2'],
                      'classifier__max_leaf_nodes': [10, 20, 50]}

inner_cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
outer_cv = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)

grid_search = GridSearchCV(estimator=clf_pipeline, param_grid=param_grid, cv=
                          n_jobs=-1, return_train_score=True)
```

```
In [26]: outer_scores = []
best_params = []
for i, (train_idx, test_idx) in enumerate(outer_cv.split(X_train, y_train),
    print(f"\nOuter fold {i} is running-----")

    X_tr, X_te = X_train.iloc[train_idx], X_train.iloc[test_idx]
    y_tr, y_te = y_train.iloc[train_idx], y_train.iloc[test_idx]
    grid_search.fit(X_tr, y_tr)
```

```

score = grid_search.score(X_te, y_te)
outer_scores.append(score)
best_params.append(grid_search.best_params_)

print(f"Outer fold {i} AUC-ROC score: {score:.4f}")
print(f"Best params: {grid_search.best_params_}")

```

Outer fold 1 is running-----

Fitting 5 folds for each of 27 candidates, totalling 135 fits

/new/benpyenv/lib/python3.10/site-packages/numpy/ma/core.py:2846: RuntimeWarning: invalid value encountered in cast

```
_data = np.array(data, dtype=dtype, copy=copy,
```

Outer fold 1 AUC-ROC score: 0.8475

Best params: {'classifier__max_depth': 10, 'classifier__max_features': None, 'classifier__max_leaf_nodes': 50}

Outer fold 2 is running-----

Fitting 5 folds for each of 27 candidates, totalling 135 fits

Outer fold 2 AUC-ROC score: 0.8505

Best params: {'classifier__max_depth': 10, 'classifier__max_features': None, 'classifier__max_leaf_nodes': 50}

Outer fold 3 is running-----

Fitting 5 folds for each of 27 candidates, totalling 135 fits

Outer fold 3 AUC-ROC score: 0.8485

Best params: {'classifier__max_depth': 10, 'classifier__max_features': None, 'classifier__max_leaf_nodes': 50}

```

In [28]: print("\nResults of nested CV")
         for i, (score, params) in enumerate(zip(outer_scores, best_params), 1):
             print(f"Fold {i}: AUC = {score:.4f}, Best params = {params}")

         print(f"\nAverage AUC-ROC over outer //folds// >> {np.mean(outer_scores)}")

```

Results of nested CV

Fold 1: AUC = 0.8475, Best params = {'classifier__max_depth': 10, 'classifier__max_features': None, 'classifier__max_leaf_nodes': 50}
 Fold 2: AUC = 0.8505, Best params = {'classifier__max_depth': 10, 'classifier__max_features': None, 'classifier__max_leaf_nodes': 50}
 Fold 3: AUC = 0.8485, Best params = {'classifier__max_depth': 10, 'classifier__max_features': None, 'classifier__max_leaf_nodes': 50}

Average AUC-ROC over outer //folds// >> 0.8488224245741188

```
[CV 1/5] END classifier__max_depth=3, classifier__max_features=None, classifier__max_leaf_nodes=20;; score=(train=0.761, test=0.761) total time= 1.2s
[CV 2/5] END classifier__max_depth=3, classifier__max_features=sqrt, classifier__max_leaf_nodes=20;; score=(train=0.593, test=0.594) total time= 0.5s
[CV 4/5] END classifier__max_depth=3, classifier__max_features=log2, classifier__max_leaf_nodes=20;; score=(train=0.530, test=0.530) total time= 0.5s
[CV 5/5] END classifier__max_depth=5, classifier__max_features=None, classifier__max_leaf_nodes=20;; score=(train=0.804, test=0.806) total time= 1.6s
[CV 5/5] END classifier__max_depth=5, classifier__max_features=sqrt, classifier__max_leaf_nodes=20;; score=(train=0.707, test=0.709) total time= 0.4s
[CV 5/5] END classifier__max_depth=5, classifier__max_features=log2, classifier__max_leaf_nodes=20;; score=(train=0.602, test=0.598) total time= 0.6s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50;; score=(train=0.850, test=0.846) total time= 2.8s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20;; score=(train=0.703, test=0.708) total time= 0.6s
[CV 3/5] END classifier__max_depth=3, classifier__max_features=None, classifier__max_leaf_nodes=20;; score=(train=0.759, test=0.762) total time= 1.0s
[CV 4/5] END classifier__max_depth=3, classifier__max_features=sqrt, classifier__max_leaf_nodes=20;; score=(train=0.675, test=0.672) total time= 0.5s
[CV 4/5] END classifier__max_depth=3, classifier__max_features=log2, classifier__max_leaf_nodes=20;; score=(train=0.530, test=0.527) total time= 0.6s
[CV 1/5] END classifier__max_depth=5, classifier__max_features=None, classifier__max_leaf_nodes=50;; score=(train=0.807, test=0.805) total time= 1.9s
[CV 2/5] END classifier__max_depth=5, classifier__max_features=log2, classifier__max_leaf_nodes=10;; score=(train=0.629, test=0.624) total time= 0.6s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20;; score=(train=0.833, test=0.831) total time= 1.9s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20;; score=(train=0.770, test=0.774) total time= 0.7s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=10;; score=(train=0.595, test=0.588) total time= 0.6s
[CV 2/5] END classifier__max_depth=3, classifier__max_features=None, classifier__max_leaf_nodes=20;; score=(train=0.761, test=0.759) total time= 1.1s
[CV 1/5] END classifier__max_depth=3, classifier__max_features=sqrt, classifier__max_leaf_nodes=50;; score=(train=0.631, test=0.637) total time= 0.6s
[CV 2/5] END classifier__max_depth=3, classifier__max_features=log2, classifier__max_leaf_nodes=50;; score=(train=0.646, test=0.646) total time= 0.5s
[CV 2/5] END classifier__max_depth=5, classifier__max_features=None, classifier__max_leaf_nodes=50;; score=(train=0.807, test=0.804) total time= 1.7s
[CV 2/5] END classifier__max_depth=5, classifier__max_features=log2, classifier__max_leaf_nodes=10;; score=(train=0.667, test=0.669) total time= 0.6s
[CV 4/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=10;; score=(train=0.810, test=0.808) total time= 1.6s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20;; score=(train=0.728, test=0.732) total time= 0.7s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20;; score=(train=0.728, test=0.732) total time= 0.7s
```

```
fier_max_leaf_nodes=10;; score=(train=0.669, test=0.671) total time= 0.6s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.762, test=0.758) total time= 1.2s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.530, test=0.529) total time= 0.5s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.797, test=0.794) total time= 1.4s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.755, test=0.752) total time= 0.7s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.652, test=0.653) total time= 0.6s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=20;; score=(train=0.835, test=0.830) total time= 2.2s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=50;; score=(train=0.784, test=0.783) total time= 0.6s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=50;; score=(train=0.695, test=0.690) total time= 0.5s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.760, test=0.760) total time= 1.3s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.572, test=0.574) total time= 0.6s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.794, test=0.797) total time= 1.3s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.708, test=0.710) total time= 0.6s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.707, test=0.704) total time= 0.6s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.654, test=0.651) total time= 0.6s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=10;; score=(train=0.735, test=0.735) total time= 0.6s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=10;; score=(train=0.737, test=0.738) total time= 0.7s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=10;; score=(train=0.592, test=0.589) total time= 0.6s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=50;; score=(train=0.576, test=0.569) total time= 0.3s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.761, test=0.759) total time= 1.2s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.679, test=0.685) total time= 0.6s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.637, test=0.639) total time= 0.6s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.811, test=0.805) total time= 1.9s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.763, test=0.770) total time= 0.6s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=20;; score=(train=0.837, test=0.836) total time= 2.2s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=50;; score=(train=0.785, test=0.784) total time= 0.8s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.762, test=0.758) total time= 1.1s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.717, test=0.718) total time= 0.6s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=log2, classif
```

```
ier_max_leaf_nodes=50;; score=(train=0.530, test=0.529) total time= 0.5s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.807, test=0.802) total time= 1.7s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.754, test=0.756) total time= 0.6s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.603, test=0.599) total time= 0.6s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=10;; score=(train=0.753, test=0.750) total time= 0.5s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=10;; score=(train=0.758, test=0.761) total time= 0.4s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=20;; score=(train=0.769, test=0.771) total time= 0.6s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=10;; score=(train=0.657, test=0.658) total time= 0.6s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.760, test=0.760) total time= 0.9s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.680, test=0.681) total time= 0.6s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.572, test=0.574) total time= 0.5s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.803, test=0.797) total time= 1.5s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.709, test=0.711) total time= 0.7s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.549, test=0.545) total time= 0.6s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=20;; score=(train=0.833, test=0.832) total time= 1.8s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=50;; score=(train=0.767, test=0.767) total time= 0.7s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.756, test=0.762) total time= 0.5s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.647, test=0.645) total time= 0.6s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.633, test=0.627) total time= 0.5s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.681, test=0.675) total time= 0.6s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.658, test=0.663) total time= 0.5s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.674, test=0.678) total time= 0.6s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.668, test=0.670) total time= 0.6s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=20;; score=(train=0.835, test=0.832) total time= 2.1s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=50;; score=(train=0.759, test=0.757) total time= 0.9s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.761, test=0.761) total time= 1.3s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.719, test=0.712) total time= 0.6s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.530, test=0.530) total time= 0.5s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=None, classif
```

```

ier_max_leaf_nodes=50;; score=(train=0.806, test=0.804) total time= 1.8s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.601, test=0.598) total time= 0.6s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=20;; score=(train=0.834, test=0.831) total time= 2.1s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.780, test=0.775) total time= 0.7s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.745, test=0.751) total time= 0.5s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.760, test=0.760) total time= 1.1s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.649, test=0.645) total time= 0.5s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.630, test=0.629) total time= 0.4s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.802, test=0.801) total time= 1.5s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.672, test=0.667) total time= 0.6s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.629, test=0.624) total time= 0.6s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=50;; score=(train=0.849, test=0.850) total time= 3.0s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.761, test=0.759) total time= 0.8s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.679, test=0.685) total time= 0.6s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.637, test=0.639) total time= 0.6s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.800, test=0.795) total time= 1.4s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.727, test=0.723) total time= 0.6s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.745, test=0.741) total time= 0.6s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=10;; score=(train=0.808, test=0.801) total time= 1.6s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=20;; score=(train=0.776, test=0.779) total time= 0.7s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=10;; score=(train=0.667, test=0.674) total time= 0.6s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.762, test=0.758) total time= 1.2s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.593, test=0.594) total time= 0.4s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.593, test=0.589) total time= 0.5s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.807, test=0.802) total time= 1.6s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.731, test=0.729) total time= 0.6s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.653, test=0.648) total time= 0.6s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=50;; score=(train=0.851, test=0.852) total time= 3.2s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=None, classif

```

```
ier_max_leaf_nodes=10;; score=(train=0.761, test=0.755) total time= 0.8s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=sqrt, classifier_max_leaf_nodes=10;; score=(train=0.675, test=0.672) total time= 0.5s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=log2, classifier_max_leaf_nodes=10;; score=(train=0.530, test=0.527) total time= 0.6s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=None, classifier_max_leaf_nodes=20;; score=(train=0.804, test=0.801) total time= 1.6s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=sqrt, classifier_max_leaf_nodes=20;; score=(train=0.707, test=0.704) total time= 0.6s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=log2, classifier_max_leaf_nodes=20;; score=(train=0.654, test=0.651) total time= 0.5s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=50;; score=(train=0.850, test=0.847) total time= 2.5s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=10;; score=(train=0.573, test=0.567) total time= 0.6s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=None, classifier_max_leaf_nodes=10;; score=(train=0.761, test=0.759) total time= 1.0s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=sqrt, classifier_max_leaf_nodes=20;; score=(train=0.631, test=0.637) total time= 0.5s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=log2, classifier_max_leaf_nodes=20;; score=(train=0.679, test=0.685) total time= 0.6s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=None, classifier_max_leaf_nodes=20;; score=(train=0.804, test=0.802) total time= 1.5s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=sqrt, classifier_max_leaf_nodes=20;; score=(train=0.698, test=0.700) total time= 0.7s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=log2, classifier_max_leaf_nodes=50;; score=(train=0.764, test=0.770) total time= 0.7s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=50;; score=(train=0.850, test=0.849) total time= 2.4s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=10;; score=(train=0.684, test=0.686) total time= 0.6s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=None, classifier_max_leaf_nodes=10;; score=(train=0.762, test=0.758) total time= 1.1s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=sqrt, classifier_max_leaf_nodes=20;; score=(train=0.724, test=0.726) total time= 0.4s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=log2, classifier_max_leaf_nodes=20;; score=(train=0.530, test=0.529) total time= 0.5s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=None, classifier_max_leaf_nodes=20;; score=(train=0.803, test=0.808) total time= 1.5s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=sqrt, classifier_max_leaf_nodes=20;; score=(train=0.757, test=0.754) total time= 0.6s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=log2, classifier_max_leaf_nodes=50;; score=(train=0.652, test=0.653) total time= 0.6s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=50;; score=(train=0.851, test=0.847) total time= 2.9s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=20;; score=(train=0.697, test=0.697) total time= 0.5s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=None, classifier_max_leaf_nodes=50;; score=(train=0.762, test=0.761) total time= 1.4s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=log2, classifier_max_leaf_nodes=10;; score=(train=0.715, test=0.718) total time= 0.7s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=None, classifier_max_leaf_nodes=10;; score=(train=0.795, test=0.795) total time= 1.4s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=sqrt, classifier_max_leaf_nodes=10;; score=(train=0.734, test=0.735) total time= 0.6s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=log2, classifier_max_leaf_nodes=10;; score=(train=0.734, test=0.735) total time= 0.6s
```

```

ier_max_leaf_nodes=10;; score=(train=0.747, test=0.751) total time= 0.7s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.806, test=0.808) total time= 1.5s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.778, test=0.779) total time= 0.6s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.594, test=0.594) total time= 0.7s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.760, test=0.761) total time= 0.9s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.639, test=0.638) total time= 0.6s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.681, test=0.675) total time= 0.6s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.800, test=0.803) total time= 1.5s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.737, test=0.738) total time= 0.7s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.649, test=0.656) total time= 0.5s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.834, test=0.834) total time= 1.9s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.771, test=0.771) total time= 0.6s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.714, test=0.704) total time= 0.6s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.761, test=0.763) total time= 1.1s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.593, test=0.594) total time= 0.6s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.593, test=0.589) total time= 0.5s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.753, test=0.749) total time= 0.5s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.752, test=0.751) total time= 0.5s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.652, test=0.653) total time= 0.6s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.809, test=0.815) total time= 1.7s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.712, test=0.715) total time= 0.6s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.657, test=0.657) total time= 0.6s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.762, test=0.761) total time= 0.8s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.688, test=0.690) total time= 0.7s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.551, test=0.552) total time= 0.6s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.801, test=0.804) total time= 1.6s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.741, test=0.740) total time= 0.7s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.752, test=0.755) total time= 0.6s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=None, classif

```



```
fier_max_leaf_nodes=50;; score=(train=0.850, test=0.848) total time= 2.7s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=log2, classif
fier_max_leaf_nodes=50;; score=(train=0.720, test=0.715) total time= 0.4s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.761, test=0.759) total time= 1.2s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.532, test=0.536) total time= 0.5s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.794, test=0.791) total time= 1.5s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.703, test=0.703) total time= 0.7s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.682, test=0.681) total time= 0.7s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.849, test=0.851) total time= 2.6s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.733, test=0.739) total time= 0.7s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.762, test=0.758) total time= 1.3s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.530, test=0.529) total time= 0.5s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.797, test=0.793) total time= 1.2s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.751, test=0.754) total time= 0.6s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.757, test=0.754) total time= 0.5s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.652, test=0.652) total time= 0.5s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.851, test=0.849) total time= 2.8s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.676, test=0.671) total time= 0.4s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.759, test=0.762) total time= 1.0s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.681, test=0.677) total time= 0.6s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.796, test=0.796) total time= 1.4s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.671, test=0.666) total time= 0.5s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.709, test=0.711) total time= 0.6s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.549, test=0.545) total time= 0.6s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.849, test=0.846) total time= 2.7s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.626, test=0.621) total time= 0.5s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.756, test=0.762) total time= 1.2s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.639, test=0.638) total time= 0.7s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.793, test=0.798) total time= 1.4s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
```

```
ier_max_leaf_nodes=10;; score=(train=0.703, test=0.703) total time= 0.6s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.737, test=0.738) total time= 0.6s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.682, test=0.681) total time= 0.6s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=10;; score=(train=0.688, test=0.692) total time= 0.5s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=10;; score=(train=0.735, test=0.734) total time= 0.6s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=50;; score=(train=0.759, test=0.762) total time= 0.6s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=20;; score=(train=0.709, test=0.716) total time= 0.6s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.761, test=0.763) total time= 1.0s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.717, test=0.718) total time= 0.4s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.530, test=0.529) total time= 0.5s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.806, test=0.808) total time= 1.8s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.653, test=0.648) total time= 0.6s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=20;; score=(train=0.834, test=0.838) total time= 1.8s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=20;; score=(train=0.762, test=0.761) total time= 0.7s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=20;; score=(train=0.678, test=0.676) total time= 0.6s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.761, test=0.755) total time= 1.1s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.680, test=0.681) total time= 0.6s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.572, test=0.574) total time= 0.6s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.805, test=0.802) total time= 1.9s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.751, test=0.755) total time= 0.6s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=20;; score=(train=0.833, test=0.833) total time= 2.0s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=50;; score=(train=0.777, test=0.778) total time= 0.7s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=50;; score=(train=0.785, test=0.786) total time= 0.3s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.760, test=0.761) total time= 1.2s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.679, test=0.685) total time= 0.6s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.796, test=0.795) total time= 1.4s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.735, test=0.737) total time= 0.5s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.728, test=0.724) total time= 0.6s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=None, classi
```

```

fier_max_leaf_nodes=10;; score=(train=0.808, test=0.813) total time= 1.6s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
fier_max_leaf_nodes=10;; score=(train=0.735, test=0.737) total time= 0.5s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
fier_max_leaf_nodes=20;; score=(train=0.750, test=0.748) total time= 0.6s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=log2, classif
fier_max_leaf_nodes=20;; score=(train=0.726, test=0.728) total time= 0.7s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.762, test=0.758) total time= 0.9s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.593, test=0.594) total time= 0.4s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.724, test=0.726) total time= 0.5s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.555, test=0.555) total time= 0.4s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.806, test=0.810) total time= 1.8s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.653, test=0.648) total time= 0.4s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.810, test=0.807) total time= 1.4s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.764, test=0.760) total time= 0.6s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.746, test=0.749) total time= 0.5s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.618, test=0.615) total time= 0.6s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.762, test=0.761) total time= 1.2s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.681, test=0.677) total time= 0.7s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.715, test=0.718) total time= 0.5s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.804, test=0.805) total time= 1.6s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.736, test=0.736) total time= 0.6s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.803, test=0.803) total time= 1.6s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.731, test=0.732) total time= 0.5s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.792, test=0.791) total time= 0.7s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.575, test=0.569) total time= 0.5s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.761, test=0.759) total time= 1.0s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.647, test=0.645) total time= 0.6s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.532, test=0.536) total time= 0.6s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.805, test=0.801) total time= 1.8s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.679, test=0.680) total time= 0.7s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=None, classif

```

```

fier_max_leaf_nodes=20;; score=(train=0.835, test=0.838) total time= 2.0s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
fier_max_leaf_nodes=20;; score=(train=0.765, test=0.766) total time= 0.7s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=log2, classif
fier_max_leaf_nodes=20;; score=(train=0.703, test=0.698) total time= 0.6s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.761, test=0.763) total time= 0.9s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.717, test=0.718) total time= 0.4s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.593, test=0.589) total time= 0.5s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.796, test=0.799) total time= 1.4s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.730, test=0.729) total time= 0.6s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.657, test=0.662) total time= 0.6s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.835, test=0.833) total time= 2.3s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.751, test=0.754) total time= 0.7s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.705, test=0.703) total time= 0.3s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.680, test=0.681) total time= 0.5s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.649, test=0.645) total time= 0.6s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.715, test=0.718) total time= 0.6s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.802, test=0.802) total time= 1.5s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.736, test=0.736) total time= 0.6s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.670, test=0.666) total time= 0.6s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.850, test=0.845) total time= 2.7s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.777, test=0.778) total time= 0.5s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.759, test=0.755) total time= 1.0s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.679, test=0.685) total time= 0.5s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.637, test=0.639) total time= 0.6s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.808, test=0.803) total time= 1.5s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.674, test=0.678) total time= 0.6s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.668, test=0.670) total time= 0.6s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.852, test=0.851) total time= 2.9s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.762, test=0.768) total time= 0.4s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif

```

```

ier_max_leaf_nodes=10;; score=(train=0.719, test=0.712) total time= 0.6s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.719, test=0.712) total time= 0.4s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.555, test=0.555) total time= 0.6s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.804, test=0.802) total time= 1.6s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.755, test=0.752) total time= 0.5s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.657, test=0.662) total time= 0.6s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=50;; score=(train=0.851, test=0.851) total time= 2.7s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=20;; score=(train=0.670, test=0.668) total time= 0.5s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.760, test=0.760) total time= 1.1s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.688, test=0.690) total time= 0.7s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.530, test=0.527) total time= 0.5s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.804, test=0.804) total time= 1.9s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.670, test=0.666) total time= 0.6s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=20;; score=(train=0.835, test=0.828) total time= 2.1s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=50;; score=(train=0.734, test=0.725) total time= 0.7s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=50;; score=(train=0.697, test=0.694) total time= 0.4s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.759, test=0.755) total time= 1.2s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.646, test=0.646) total time= 0.6s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.805, test=0.806) total time= 1.4s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.696, test=0.698) total time= 0.6s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.742, test=0.748) total time= 0.6s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=10;; score=(train=0.806, test=0.804) total time= 1.5s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=10;; score=(train=0.715, test=0.714) total time= 0.5s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=50;; score=(train=0.790, test=0.793) total time= 0.7s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=20;; score=(train=0.759, test=0.751) total time= 0.5s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.760, test=0.764) total time= 0.9s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.724, test=0.726) total time= 0.5s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.530, test=0.530) total time= 0.6s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=None, classif

```

```

ier_max_leaf_nodes=20;; score=(train=0.804, test=0.800) total time= 1.9s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.707, test=0.709) total time= 0.5s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=10;; score=(train=0.812, test=0.804) total time= 1.5s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=10;; score=(train=0.752, test=0.750) total time= 0.6s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=50;; score=(train=0.799, test=0.794) total time= 0.7s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=20;; score=(train=0.669, test=0.666) total time= 0.5s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.760, test=0.760) total time= 1.0s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.688, test=0.690) total time= 0.7s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.551, test=0.552) total time= 0.6s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.805, test=0.800) total time= 1.9s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.575, test=0.569) total time= 0.5s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=10;; score=(train=0.806, test=0.803) total time= 1.6s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=20;; score=(train=0.729, test=0.720) total time= 0.7s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=10;; score=(train=0.746, test=0.750) total time= 0.6s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.759, test=0.755) total time= 0.9s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.633, test=0.627) total time= 0.6s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.646, test=0.646) total time= 0.6s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.814, test=0.814) total time= 1.5s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.728, test=0.723) total time= 0.6s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.766, test=0.762) total time= 0.5s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=20;; score=(train=0.830, test=0.824) total time= 1.9s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=50;; score=(train=0.785, test=0.783) total time= 0.7s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=50;; score=(train=0.710, test=0.716) total time= 0.3s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.760, test=0.764) total time= 1.2s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.593, test=0.594) total time= 0.5s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.796, test=0.792) total time= 1.3s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.730, test=0.729) total time= 0.6s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.656, test=0.661) total time= 0.5s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=None, classi

```

```
fier_max_leaf_nodes=10;; score=(train=0.811, test=0.809) total time= 1.8s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
fier_max_leaf_nodes=20;; score=(train=0.775, test=0.772) total time= 0.7s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=log2, classif
fier_max_leaf_nodes=10;; score=(train=0.664, test=0.661) total time= 0.6s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.760, test=0.760) total time= 0.9s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.681, test=0.677) total time= 0.6s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.630, test=0.629) total time= 0.5s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.795, test=0.794) total time= 1.4s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.741, test=0.740) total time= 0.6s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.549, test=0.544) total time= 0.4s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.807, test=0.800) total time= 1.7s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.716, test=0.714) total time= 0.6s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.763, test=0.760) total time= 0.7s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.741, test=0.740) total time= 0.3s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.631, test=0.637) total time= 0.5s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.647, test=0.645) total time= 0.6s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.532, test=0.536) total time= 0.5s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.802, test=0.799) total time= 1.6s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.704, test=0.703) total time= 0.7s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.649, test=0.656) total time= 0.6s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.851, test=0.847) total time= 2.9s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.744, test=0.745) total time= 0.5s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.760, test=0.764) total time= 0.9s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.593, test=0.594) total time= 0.5s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.530, test=0.529) total time= 0.5s
[CV 2/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.804, test=0.799) total time= 1.5s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.754, test=0.756) total time= 0.6s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.652, test=0.652) total time= 0.6s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.834, test=0.837) total time= 2.2s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=sqrt, classif
```

```

fier_max_leaf_nodes=50;; score=(train=0.798, test=0.794) total time= 0.7s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=log2, classif
fier_max_leaf_nodes=50;; score=(train=0.674, test=0.672) total time= 0.3s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.761, test=0.755) total time= 1.1s
[CV 4/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.675, test=0.672) total time= 0.7s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.551, test=0.552) total time= 0.5s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.736, test=0.735) total time= 0.5s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.706, test=0.704) total time= 0.7s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.592, test=0.588) total time= 0.6s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=10;; score=(train=0.804, test=0.809) total time= 1.5s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=10;; score=(train=0.699, test=0.693) total time= 0.5s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=20;; score=(train=0.744, test=0.745) total time= 0.5s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=20;; score=(train=0.596, test=0.592) total time= 0.5s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=20;; score=(train=0.756, test=0.762) total time= 1.1s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=20;; score=(train=0.633, test=0.627) total time= 0.6s
[CV 5/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=20;; score=(train=0.681, test=0.675) total time= 0.6s
[CV 1/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.803, test=0.805) total time= 1.7s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=50;; score=(train=0.698, test=0.700) total time= 0.6s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=50;; score=(train=0.766, test=0.762) total time= 0.6s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=50;; score=(train=0.850, test=0.844) total time= 2.9s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=50;; score=(train=0.709, test=0.702) total time= 0.4s
[CV 1/5] END classifier_max_depth=3, classifier_max_features=None, classif
ier_max_leaf_nodes=50;; score=(train=0.761, test=0.761) total time= 1.2s
[CV 2/5] END classifier_max_depth=3, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.555, test=0.555) total time= 0.6s
[CV 3/5] END classifier_max_depth=5, classifier_max_features=None, classif
ier_max_leaf_nodes=10;; score=(train=0.795, test=0.801) total time= 1.4s
[CV 5/5] END classifier_max_depth=5, classifier_max_features=sqrt, classif
ier_max_leaf_nodes=10;; score=(train=0.707, test=0.709) total time= 0.7s
[CV 4/5] END classifier_max_depth=5, classifier_max_features=log2, classif
ier_max_leaf_nodes=10;; score=(train=0.652, test=0.652) total time= 0.6s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=None, classi
fier_max_leaf_nodes=10;; score=(train=0.810, test=0.814) total time= 1.7s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=sqrt, classi
fier_max_leaf_nodes=20;; score=(train=0.782, test=0.777) total time= 0.6s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=log2, classi
fier_max_leaf_nodes=10;; score=(train=0.662, test=0.668) total time= 0.6s
[CV 3/5] END classifier_max_depth=3, classifier_max_features=None, classif

```



```

ier__max_leaf_nodes=10;; score=(train=0.759, test=0.762) total time= 0.6s
[CV 2/5] END classifier__max_depth=3, classifier__max_features=sqrt, classif
ier__max_leaf_nodes=10;; score=(train=0.649, test=0.645) total time= 0.6s
[CV 1/5] END classifier__max_depth=3, classifier__max_features=log2, classif
ier__max_leaf_nodes=10;; score=(train=0.630, test=0.629) total time= 0.6s
[CV 2/5] END classifier__max_depth=5, classifier__max_features=None, classif
ier__max_leaf_nodes=10;; score=(train=0.796, test=0.791) total time= 1.6s
[CV 2/5] END classifier__max_depth=5, classifier__max_features=sqrt, classif
ier__max_leaf_nodes=20;; score=(train=0.672, test=0.667) total time= 0.7s
[CV 2/5] END classifier__max_depth=5, classifier__max_features=log2, classif
ier__max_leaf_nodes=20;; score=(train=0.629, test=0.624) total time= 0.6s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=None, classi
fier__max_leaf_nodes=20;; score=(train=0.833, test=0.836) total time= 2.0s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=sqrt, classi
fier__max_leaf_nodes=20;; score=(train=0.757, test=0.754) total time= 0.6s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classi
fier__max_leaf_nodes=20;; score=(train=0.698, test=0.696) total time= 0.6s
[CV 3/5] END classifier__max_depth=3, classifier__max_features=None, classif
ier__max_leaf_nodes=20;; score=(train=0.760, test=0.761) total time= 1.0s
[CV 4/5] END classifier__max_depth=3, classifier__max_features=sqrt, classif
ier__max_leaf_nodes=20;; score=(train=0.639, test=0.638) total time= 0.6s
[CV 1/5] END classifier__max_depth=3, classifier__max_features=log2, classif
ier__max_leaf_nodes=50;; score=(train=0.679, test=0.685) total time= 0.6s
[CV 3/5] END classifier__max_depth=5, classifier__max_features=None, classif
ier__max_leaf_nodes=50;; score=(train=0.816, test=0.815) total time= 1.8s
[CV 3/5] END classifier__max_depth=5, classifier__max_features=log2, classif
ier__max_leaf_nodes=10;; score=(train=0.649, test=0.656) total time= 0.5s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=None, classi
fier__max_leaf_nodes=10;; score=(train=0.809, test=0.808) total time= 1.6s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=sqrt, classi
fier__max_leaf_nodes=10;; score=(train=0.721, test=0.721) total time= 0.6s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classi
fier__max_leaf_nodes=10;; score=(train=0.711, test=0.716) total time= 0.6s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=log2, classi
fier__max_leaf_nodes=50;; score=(train=0.785, test=0.778) total time= 0.3s

```

Part III: ML pipeline for a random forest

Question 13: Bare bones random forest

Just as you did for the classification tree, start by training a bare bones random forest using your training data. We will take care of the hyperparameter optimization later, but for the time being, use a maximum tree depth of 15. Hopefully is it clear by now why this is a good idea, since we are going to be training a bunch of trees. You should continue to use the same preprocessing step as before.

- How many trees are in the forest?
- Which are the most important features?
- Explain the meaning of variable importance in the case of a random forest classifier.

```
In [31]: rf_clf = RandomForestClassifier(max_depth=15, random_state=42, n_jobs=-1)
rf_pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('classifier',
rf_pipeline.fit(X_train, np.asarray(y_train).ravel())
rf_model = rf_pipeline.named_steps['classifier']
print(f"No. of trees in the forest: {len(rf_model.estimators_)}")
```

No. of trees in the forest: 100

```
In [32]: ohe = rf_pipeline.named_steps['preprocessor'].named_transformers_['cat'].named_steps['ohe']
ohe_names = ohe.get_feature_names_out(cat_feats)
all_feature_names = np.concatenate([num_feats, ohe_names])

importances = rf_model.feature_importances_
top_idx = np.argsort(importances)[::-1][:10]
```

```
In [33]: print("Top 10 features by importance:")
for idx in top_idx:
    print(f"{all_feature_names[idx]}: {importances[idx]:.4f}")
```

Top 10 features by importance:

WKHP: 0.2050
 AGE: 0.1192
 SCHL_22.0: 0.0670
 SCHL_21.0: 0.0592
 RELP_0.0: 0.0559
 MAR_5.0: 0.0529
 POBP_303.0: 0.0528
 MAR_1.0: 0.0525
 RELP_2.0: 0.0301
 SCHL_16.0: 0.0293

Question 14: Estimating AUC using K-fold CV

Calculate the 5-fold cross-validated AUC-ROC for the random forests pipeline.

- Report the score for each fold as well as the average across all folds.

Hint 1: `sklearn.model_selection.cross_val_score` is your friend. Setting the verbose option to 3 is useful.

```
In [35]: cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
rf_cv_scores = cross_val_score(rf_pipeline, X_train, np.asarray(y_train).ravel(), cv=cv, n_jobs=-1)
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 16 concurrent workers.
[Parallel(n_jobs=-1)]: Done 2 out of 5 | elapsed: 34.4s remaining: 51.6s
[Parallel(n_jobs=-1)]: Done 5 out of 5 | elapsed: 34.4s finished
```

```
In [36]: for i, score in enumerate(rf_cv_scores, 1):
    print(f"Fold {i}: AUC-ROC = {score}")
```

```
Fold 1: AUC-ROC = 0.8684340093336425
Fold 2: AUC-ROC = 0.8736474844046158
Fold 3: AUC-ROC = 0.8751826171401669
Fold 4: AUC-ROC = 0.8718618406099571
Fold 5: AUC-ROC = 0.8687820908022938
```

```
In [37]: print(f"\nAve. AUC-ROC across al 5 folds: {np.mean(rf_cv_scores)}")
```

```
Ave. AUC-ROC across al 5 folds: 0.8715816084581351
```

Question 15: Estimating AUC using nested CV [extra-credit]

Let's try to improve on the results of the random forest by tweaking its hyperparameters. Build a pipeline that computes the nested cv AUC-ROC. Include different values for "n_estimators", "max_samples", "max_features", "max_depth" and "max_leaf_nodes" in your hyperparameter grid.

- Print out the inner and outer fold scores for all combinations of hyperparams.
- Report the AUC-ROC of a decision tree pipeline (average over outer fold scores)
- Report the best hyperparams used in each outer fold.
- Interpret your results

What do you notice about these results? How are these results related to those from the decision tree pipeline? Is hyperparameter optimization worth the trouble?

Hint 1: The pipeline remains the same. You can access the list of all the hyperparams in your pipeline with `pipeline.get_params_keys()`. Notice the naming convention.

Hint 2: `GridSearchCV` and `cross_val_score` are your friends. Specially if you run them with `verbose = 3`.

Hint 3: This gets computationally expensive quickly. Be intelligent about the hyperparameter values you include in the grid.

```
In [38]: rf_clf = RandomForestClassifier(random_state=42)

rf_pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('classifier',
param_grid = {'classifier__n_estimators': [50, 100],
               'classifier__max_samples': [0.6, 0.8],
               'classifier__max_features': ['sqrt', 'log2', None],
               'classifier__max_depth': [10, 15],
               'classifier__max_leaf_nodes': [20, 50]})
```

```
In [39]: inner_cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
outer_cv = StratifiedKFold(n_splits=3, shuffle=True, random_state=42)
```

```
In [41]: outer_scores = []
best_params_ = []

for i, (train_idx, test_idx) in enumerate(outer_cv.split(X_train, y_train),
```

```
print(f"\nOuter fold {i}")

X_tr, X_te = X_train.iloc[train_idx], X_train.iloc[test_idx]
y_tr, y_te = y_train.iloc[train_idx], y_train.iloc[test_idx]

grid_search = GridSearchCV(estimator=rf_pipeline, param_grid=param_grid,
grid_search.fit(X_tr, np.asarray(y_tr).ravel())

outer_score = grid_search.score(X_te, np.asarray(y_te).ravel())
outer_scores.append(outer_score)
best_params_.append(grid_search.best_params_)

print(f"Outer fold {i} AUC-ROC is: {outer_score:.4f}")
print(f"Best params is: {grid_search.best_params_}")
```

Outer fold 1

Fitting 5 folds for each of 48 candidates, totalling 240 fits

```
/new/benpyenv/lib/python3.10/site-packages/numpy/ma/core.py:2846: RuntimeWarning: invalid value encountered in cast
  _data = np.array(data, dtype=dtype, copy=copy,
```

Outer fold 1 AUC-ROC is: 0.8615

Best params is: {'classifier__max_depth': 15, 'classifier__max_features': 'sqrt', 'classifier__max_leaf_nodes': 50, 'classifier__max_samples': 0.8, 'classifier__n_estimators': 100}

Outer fold 2

Fitting 5 folds for each of 48 candidates, totalling 240 fits

[CV 5/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.860 total time= 11.3s

[CV 2/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.861 total time= 15.2s

[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.855 total time= 5.1s

[CV 3/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.845 total time= 5.6s

[CV 4/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.843 total time= 5.9s

[CV 4/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.844 total time= 40.2s

[CV 4/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.843 total time= 1.5min

[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.848 total time= 5.2s

[CV 2/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.850 total time= 5.3s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.857 total time= 5.0s

[CV 4/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.855 total time= 5.4s

[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.854 total time= 10.3s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.859 total time= 10.7s

[CV 2/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.853 total time= 6.0s

[CV 4/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.857 total time= 11.3s

[CV 5/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.865 total time= 7.8s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classi

```

fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.865 total time= 9.0s
[CV 5/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.867 total time= 18.6s
[CV 4/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.850 total time= 5.9s
[CV 5/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.865 total time= 11.4s
[CV 5/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.847 total time= 42.5s
[CV 1/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.860 total time= 1.1min
[CV 2/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.859 total time= 1.1min
[CV 3/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.854 total time= 6.4s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.858 total time= 13.7s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.852 total time= 4.2s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.853 total time= 8.6s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.855 total time= 5.8s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.857 total time= 6.4s
[CV 4/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.844 total time= 41.4s
[CV 4/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.843 total time= 1.6min
[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.854 total time= 5.2s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.845 total time= 5.0s
[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.854 total time= 5.2s
[CV 4/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.856 total time= 5.1s

```

```
[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.855 total time= 10.9s
[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.853 total time= 10.2s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.844 total time= 6.3s
[CV 5/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.858 total time= 12.1s
[CV 4/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.866 total time= 15.8s
[CV 5/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.841 total time= 4.4s
[CV 1/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.849 total time= 5.2s
Outer fold 2 AUC-ROC is: 0.8667
Best params is: {'classifier__max_depth': 15, 'classifier__max_features': 'sqrt', 'classifier__max_leaf_nodes': 50, 'classifier__max_samples': 0.8, 'classifier__n_estimators': 100}
```

Outer fold 3

Fitting 5 folds for each of 48 candidates, totalling 240 fits

```
[CV 2/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.854 total time= 6.4s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.857 total time= 7.5s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.864 total time= 8.5s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.861 total time= 4.3s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.856 total time= 5.2s
[CV 4/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.848 total time= 5.5s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.854 total time= 11.5s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.839 total time= 46.3s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.861 total time= 1.0min
[CV 5/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.861 total time= 1.0min
```

```

fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.860 total time= 1.1min
[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.852 total time= 12.1s
[CV 4/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.862 total time= 8.2s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.862 total time= 9.1s
[CV 4/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.864 total time= 18.8s
[CV 5/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.865 total time= 6.3s
[CV 3/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.852 total time= 7.0s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.840 total time= 40.0s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.840 total time= 1.5min
[CV 1/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.859 total time= 1.3min
[CV 2/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.846 total time= 7.3s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.857 total time= 8.1s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.860 total time= 17.6s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.849 total time= 6.3s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.852 total time= 11.9s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.846 total time= 46.2s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.855 total time= 58.3s
[CV 4/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.857 total time= 1.1min
[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.851 total time= 6.2s

```


[CV 4/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.860 total time= 12.1s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.861 total time= 15.8s

[CV 4/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.849 total time= 4.4s

[CV 5/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.850 total time= 8.5s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.857 total time= 9.6s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.858 total time= 12.0s

[CV 5/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.844 total time= 41.4s

[CV 5/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.843 total time= 1.6min

[CV 3/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.862 total time= 1.3min

[CV 3/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.860 total time= 6.9s

[CV 2/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.860 total time= 8.1s

[CV 4/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.858 total time= 9.3s

[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.852 total time= 9.5s

[CV 4/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.841 total time= 6.7s

[CV 1/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.848 total time= 6.0s

[CV 4/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.857 total time= 12.7s

[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.843 total time= 4.1s

[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.848 total time= 8.3s

[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.848 total time= 8.3s

```

rs=100;; score=0.858 total time= 9.5s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.852 total time= 11.1s
[CV 4/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.843 total time= 47.2s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.857 total time= 2.0min
[CV 4/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.853 total time= 6.2s
[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.858 total time= 7.8s
[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.863 total time= 7.9s
[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.859 total time= 8.9s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.865 total time= 18.7s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=log2, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.862 total time= 5.9s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=log2, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.863 total time= 12.4s
[CV 4/5] END classifier__max_depth=15, classifier__max_features=None, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.844 total time= 42.0s
[CV 4/5] END classifier__max_depth=15, classifier__max_features=None, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.844 total time= 1.6min
[CV 5/5] END classifier__max_depth=15, classifier__max_features=None, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.864 total time= 1.2min
[CV 1/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.856 total time= 13.3s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.852 total time= 8.9s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.845 total time= 4.4s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.846 total time= 4.9s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.848 total time= 9.5s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=log2, classifi

```

fier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.860 total time= 11.7s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.844 total time= 46.4s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.858 total time= 58.9s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.857 total time= 1.2min
[CV 3/5] END classifier_max_depth=15, classifier_max_features=sqrt, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.854 total time= 12.0s
[CV 2/5] END classifier_max_depth=15, classifier_max_features=sqrt, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.856 total time= 15.3s
[CV 5/5] END classifier_max_depth=15, classifier_max_features=sqrt, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.863 total time= 18.8s
[CV 5/5] END classifier_max_depth=15, classifier_max_features=log2, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.850 total time= 6.2s
[CV 3/5] END classifier_max_depth=15, classifier_max_features=log2, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.858 total time= 7.1s
[CV 1/5] END classifier_max_depth=15, classifier_max_features=None, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.842 total time= 41.4s
[CV 2/5] END classifier_max_depth=15, classifier_max_features=None, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.837 total time= 1.5min
[CV 5/5] END classifier_max_depth=15, classifier_max_features=None, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.859 total time= 1.0min
[CV 5/5] END classifier_max_depth=10, classifier_max_features=sqrt, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.850 total time= 5.9s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=sqrt, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.852 total time= 14.2s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.858 total time= 4.0s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.859 total time= 8.7s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.855 total time= 10.0s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.854 total time= 11.9s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.841 total time= 46.8s

[CV 4/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.858 total time= 11.7s

[CV 5/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.864 total time= 15.3s

[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.852 total time= 9.5s

[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.858 total time= 6.4s

[CV 3/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.849 total time= 42.1s

[CV 1/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.857 total time= 1.0min

[CV 2/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.857 total time= 1.1min

[CV 5/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.861 total time= 10.6s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.859 total time= 11.8s

[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.862 total time= 15.1s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.865 total time= 18.4s

[CV 1/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.854 total time= 6.2s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.859 total time= 12.2s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.849 total time= 42.9s

[CV 5/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.846 total time= 1.6min

[CV 4/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.862 total time= 1.2min

[CV 5/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.854 total time= 7.1s

[CV 4/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.862 total time= 8.8s

[CV 2/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.854 total time= 7.1s

```

rs=100;; score=0.855 total time= 18.6s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.853 total time= 11.5s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.847 total time= 1.4min
[CV 1/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.856 total time= 1.1min
[CV 5/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.854 total time= 5.5s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.849 total time= 10.3s
[CV 4/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.860 total time= 10.3s
[CV 4/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.859 total time= 6.5s
[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.859 total time= 8.3s
[CV 5/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.860 total time= 8.4s
[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.861 total time= 19.1s
[CV 3/5] END classifier__max_depth=15, classifier__max_features=log2, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.852 total time= 5.2s
[CV 5/5] END classifier__max_depth=15, classifier__max_features=log2, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.854 total time= 9.9s
[CV 1/5] END classifier__max_depth=15, classifier__max_features=log2, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.856 total time= 13.3s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=None, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.839 total time= 1.4min
[CV 3/5] END classifier__max_depth=15, classifier__max_features=None, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.863 total time= 1.7min
[CV 3/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.859 total time= 11.7s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.864 total time= 15.4s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.856 total time= 5.3s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classifi

```

```
fier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.854 total time= 9.9s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.858 total time= 12.1s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.843 total time= 48.8s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.863 total time= 2.1min
[CV 2/5] END classifier_max_depth=15, classifier_max_features=sqrt, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.855 total time= 6.6s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=sqrt, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.848 total time= 6.7s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=sqrt, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.860 total time= 7.8s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=sqrt, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.858 total time= 17.6s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.864 total time= 5.6s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.860 total time= 11.6s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.849 total time= 47.5s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.860 total time= 1.1min
[CV 2/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.857 total time= 2.4min
[CV 3/5] END classifier_max_depth=15, classifier_max_features=None, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.849 total time= 48.2s
[CV 4/5] END classifier_max_depth=15, classifier_max_features=None, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.862 total time= 1.1min
[CV 2/5] END classifier_max_depth=15, classifier_max_features=None, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.859 total time= 1.3min
[CV 3/5] END classifier_max_depth=10, classifier_max_features=sqrt, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.851 total time= 7.4s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=sqrt, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.859 total time= 16.0s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.855 total time= 8.9s
```

[CV 4/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.856 total time= 9.7s

[CV 4/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.862 total time= 11.8s

[CV 1/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.843 total time= 1.6min

[CV 3/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.860 total time= 1.2min

[CV 5/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.855 total time= 6.4s

[CV 2/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.854 total time= 8.3s

[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.861 total time= 16.2s

[CV 2/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.847 total time= 4.5s

[CV 2/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.850 total time= 8.6s

[CV 5/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.849 total time= 4.9s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.853 total time= 6.3s

[CV 1/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.854 total time= 6.8s

[CV 2/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.853 total time= 12.9s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.847 total time= 1.5min

[CV 1/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.858 total time= 1.2min

[CV 2/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.857 total time= 5.9s

[CV 3/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.860 total time= 13.1s

[CV 3/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.865 total time= 19.8s

[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.856 total time= 9.7s

```

rs=50;; score=0.861 total time= 6.3s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.847 total time= 42.3s
[CV 4/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.844 total time= 1.6min
[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.860 total time= 5.6s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.857 total time= 5.5s
[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.859 total time= 5.1s
[CV 4/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.854 total time= 5.7s
[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.858 total time= 10.6s
[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.859 total time= 11.1s
[CV 4/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.854 total time= 6.7s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.852 total time= 12.9s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.860 total time= 8.6s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.842 total time= 4.2s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.863 total time= 8.3s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.853 total time= 10.5s
[CV 4/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.844 total time= 1.3min
[CV 3/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.862 total time= 1.9min
[CV 5/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.867 total time= 15.4s
[CV 4/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.842 total time= 4.4s
[CV 5/5] END classifier__max_depth=15, classifier__max_features=log2, classi

```



```

fier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.863 total time= 8.4s
[CV 1/5] END classifier_max_depth=15, classifier_max_features=log2, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.853 total time= 9.6s
[CV 1/5] END classifier_max_depth=15, classifier_max_features=log2, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.857 total time= 6.8s
[CV 3/5] END classifier_max_depth=15, classifier_max_features=log2, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.858 total time= 12.9s
[CV 5/5] END classifier_max_depth=15, classifier_max_features=None, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.847 total time= 1.4min
[CV 5/5] END classifier_max_depth=15, classifier_max_features=None, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.865 total time= 1.7min
[CV 4/5] END classifier_max_depth=10, classifier_max_features=sqrt, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.860 total time= 11.5s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=sqrt, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.855 total time= 15.8s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.844 total time= 5.0s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.857 total time= 10.2s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.854 total time= 11.0s
[CV 5/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.843 total time= 48.6s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.857 total time= 1.9min
[CV 5/5] END classifier_max_depth=15, classifier_max_features=sqrt, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.857 total time= 10.4s
[CV 1/5] END classifier_max_depth=15, classifier_max_features=sqrt, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.856 total time= 12.1s
[CV 3/5] END classifier_max_depth=15, classifier_max_features=sqrt, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.861 total time= 8.3s
[CV 3/5] END classifier_max_depth=15, classifier_max_features=sqrt, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.860 total time= 9.0s
[CV 4/5] END classifier_max_depth=15, classifier_max_features=sqrt, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.867 total time= 20.0s
[CV 1/5] END classifier_max_depth=15, classifier_max_features=log2, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.858 total time= 12.4s

```

[CV 3/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.861 total time= 13.6s

[CV 1/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.842 total time= 49.7s

[CV 2/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.856 total time= 1.1min

[CV 4/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.860 total time= 1.1min

[CV 1/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.858 total time= 11.8s

[CV 4/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.859 total time= 16.7s

[CV 5/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.842 total time= 4.9s

[CV 5/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.849 total time= 10.2s

[CV 5/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.853 total time= 11.4s

[CV 1/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.850 total time= 1.6min

[CV 3/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.860 total time= 1.2min

[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.860 total time= 6.4s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.860 total time= 6.5s

[CV 5/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.852 total time= 13.0s

[CV 4/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.854 total time= 6.8s

[CV 1/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.859 total time= 15.3s

[CV 4/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.841 total time= 4.3s

[CV 3/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.843 total time= 4.9s

[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.843 total time= 4.9s

```

rs=50;; score=0.848 total time= 5.7s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.860 total time= 6.3s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.840 total time= 1.4min
[CV 5/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.862 total time= 2.1min
[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.860 total time= 20.0s
[CV 5/5] END classifier__max_depth=15, classifier__max_features=log2, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.861 total time= 10.3s
[CV 1/5] END classifier__max_depth=15, classifier__max_features=log2, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.860 total time= 13.1s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=None, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.841 total time= 1.4min
[CV 4/5] END classifier__max_depth=15, classifier__max_features=None, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.862 total time= 1.7min
[CV 1/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.854 total time= 6.8s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.852 total time= 8.1s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.860 total time= 8.8s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.849 total time= 9.1s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.851 total time= 5.9s
[CV 4/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.856 total time= 6.4s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.844 total time= 43.0s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.857 total time= 1.0min
[CV 2/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.854 total time= 1.2min
[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.853 total time= 6.2s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi

```

```
fier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.849 total time= 12.3s
[CV 4/5] END classifier_max_depth=15, classifier_max_features=sqrt, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.864 total time= 8.4s
[CV 5/5] END classifier_max_depth=15, classifier_max_features=sqrt, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.860 total time= 9.1s
[CV 3/5] END classifier_max_depth=15, classifier_max_features=log2, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.846 total time= 4.5s
[CV 4/5] END classifier_max_depth=15, classifier_max_features=log2, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.854 total time= 8.5s
[CV 2/5] END classifier_max_depth=15, classifier_max_features=log2, classifier_max_leaf_nodes=20, classifier_max_samples=0.8, classifier_n_estimators=100;; score=0.848 total time= 10.4s
[CV 5/5] END classifier_max_depth=15, classifier_max_features=log2, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.857 total time= 12.2s
[CV 1/5] END classifier_max_depth=15, classifier_max_features=None, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.843 total time= 1.4min
[CV 2/5] END classifier_max_depth=15, classifier_max_features=None, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.857 total time= 1.8min
[CV 4/5] END classifier_max_depth=10, classifier_max_features=sqrt, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.854 total time= 12.4s
[CV 1/5] END classifier_max_depth=10, classifier_max_features=sqrt, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.865 total time= 8.8s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.846 total time= 4.1s
[CV 4/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.848 total time= 10.5s
[CV 2/5] END classifier_max_depth=10, classifier_max_features=log2, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.854 total time= 11.4s
[CV 3/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.847 total time= 1.4min
[CV 3/5] END classifier_max_depth=10, classifier_max_features=None, classifier_max_leaf_nodes=50, classifier_max_samples=0.6, classifier_n_estimators=100;; score=0.861 total time= 2.1min
[CV 2/5] END classifier_max_depth=15, classifier_max_features=sqrt, classifier_max_leaf_nodes=50, classifier_max_samples=0.8, classifier_n_estimators=50;; score=0.862 total time= 9.3s
[CV 1/5] END classifier_max_depth=15, classifier_max_features=log2, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.857 total time= 4.3s
[CV 5/5] END classifier_max_depth=15, classifier_max_features=log2, classifier_max_leaf_nodes=20, classifier_max_samples=0.6, classifier_n_estimators=50;; score=0.847 total time= 4.4s
```

[CV 1/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.854 total time= 11.8s

[CV 4/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.862 total time= 14.9s

[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.848 total time= 4.8s

[CV 3/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.849 total time= 9.8s

[CV 5/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.862 total time= 11.6s

[CV 1/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.843 total time= 1.6min

[CV 3/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.861 total time= 1.2min

[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.849 total time= 6.1s

[CV 5/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.857 total time= 6.4s

[CV 2/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.859 total time= 8.8s

[CV 4/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.865 total time= 16.4s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.848 total time= 4.3s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.853 total time= 8.4s

[CV 5/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.860 total time= 5.0s

[CV 1/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.856 total time= 11.9s

[CV 4/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.857 total time= 12.6s

[CV 4/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.845 total time= 1.4min

[CV 1/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.859 total time= 1.1min

[CV 2/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.854 total time= 11.8s

```
rs=50;; score=0.843 total time= 6.1s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.849 total time= 13.5s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.861 total time= 18.3s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.852 total time= 6.4s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.839 total time= 42.7s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.843 total time= 1.5min
[CV 5/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.857 total time= 2.4min
[CV 1/5] END classifier__max_depth=15, classifier__max_features=None, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.843 total time= 1.6min
[CV 1/5] END classifier__max_depth=15, classifier__max_features=None, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.858 total time= 1.4min
[CV 4/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.853 total time= 7.3s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=100;; score=0.864 total time= 15.9s
[CV 4/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.835 total time= 4.9s
[CV 4/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.838 total time= 5.4s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.848 total time= 6.2s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.862 total time= 12.1s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.847 total time= 48.5s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimato
rs=50;; score=0.860 total time= 1.0min
[CV 5/5] END classifier__max_depth=10, classifier__max_features=None, classifi
fier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimato
rs=50;; score=0.855 total time= 1.2min
[CV 2/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
fier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimato
rs=100;; score=0.854 total time= 12.8s
[CV 5/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifi
```

```

fier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.857 total time= 8.6s
[CV 4/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.860 total time= 9.2s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.850 total time= 4.3s
[CV 1/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.859 total time= 8.6s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.854 total time= 5.3s
[CV 3/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.856 total time= 6.0s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.861 total time= 12.4s
[CV 4/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.863 total time= 17.8s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.865 total time= 10.7s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.842 total time= 48.2s
[CV 2/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.857 total time= 56.4s
[CV 4/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.859 total time= 1.1min
[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.860 total time= 6.3s
[CV 5/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.862 total time= 12.8s
[CV 3/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.865 total time= 16.5s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.858 total time= 4.6s
[CV 4/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.848 total time= 8.7s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.858 total time= 9.6s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.861 total time= 6.5s

```

[CV 2/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.862 total time= 12.3s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.850 total time= 1.3min

[CV 3/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.864 total time= 1.7min

[CV 5/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.856 total time= 12.4s

[CV 1/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.859 total time= 8.9s

[CV 2/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.847 total time= 4.5s

[CV 4/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.853 total time= 8.6s

[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.857 total time= 11.0s

[CV 2/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.838 total time= 1.4min

[CV 5/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.858 total time= 2.0min

[CV 5/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.862 total time= 16.6s

[CV 3/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.853 total time= 8.3s

[CV 4/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.850 total time= 5.0s

[CV 4/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.857 total time= 6.5s

[CV 4/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=50;; score=0.857 total time= 6.8s

[CV 4/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.861 total time= 12.8s

[CV 4/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.844 total time= 1.4min

[CV 5/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.860 total time= 1.7min

[CV 5/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.856 total time= 12.4s


```

rs=100;; score=0.852 total time= 11.8s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.858 total time= 16.7s
[CV 1/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.859 total time= 9.2s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.853 total time= 11.0s
[CV 5/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.842 total time= 1.4min
[CV 5/5] END classifier__max_depth=10, classifier__max_features=None, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.856 total time= 2.1min
[CV 1/5] END classifier__max_depth=15, classifier__max_features=sqrt, classifier__max_leaf_nodes=50, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.866 total time= 18.8s
[CV 1/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=20, classifier__max_samples=0.8, classifier__n_estimators=100;; score=0.860 total time= 9.6s
[CV 2/5] END classifier__max_depth=15, classifier__max_features=log2, classifier__max_leaf_nodes=50, classifier__max_samples=0.6, classifier__n_estimators=100;; score=0.858 total time= 12.0s
[CV 3/5] END classifier__max_depth=15, classifier__max_features=None, classifier__max_leaf_nodes=20, classifier__max_samples=0.6, classifier__n_estimators=50;; score=0.847 total time= 43.7s
Outer fold 3 AUC-ROC is: 0.8645
Best params is: {'classifier__max_depth': 15, 'classifier__max_features': 'sqrt', 'classifier__max_leaf_nodes': 50, 'classifier__max_samples': 0.6, 'classifier__n_estimators': 100}

```

```

In [42]: for i, (score, params) in enumerate(zip(outer_scores, best_params_), 1):
          print(f"Fold {i}: AUC-ROC = {score} $ Best params = {params}")

          print(f"\nAverage AUC-ROC across outer folds >> {np.mean(outer_scores)}")

```

```

Fold 1: AUC-ROC = 0.8615088531916582 $ Best params = {'classifier__max_depth': 15, 'classifier__max_features': 'sqrt', 'classifier__max_leaf_nodes': 50, 'classifier__max_samples': 0.8, 'classifier__n_estimators': 100}
Fold 2: AUC-ROC = 0.866740152465654 $ Best params = {'classifier__max_depth': 15, 'classifier__max_features': 'sqrt', 'classifier__max_leaf_nodes': 50, 'classifier__max_samples': 0.8, 'classifier__n_estimators': 100}
Fold 3: AUC-ROC = 0.8645285059151733 $ Best params = {'classifier__max_depth': 15, 'classifier__max_features': 'sqrt', 'classifier__max_leaf_nodes': 50, 'classifier__max_samples': 0.6, 'classifier__n_estimators': 100}

```

Average AUC-ROC across outer folds >> 0.8642591705241619

Question 16: Take stock of parts II and III

If you were to choose between a decision tree and a random forest for this classification task, what would be your choice and why?

In []: # Your response here

Part IV: Test data, ROC curves and fairness thresholds

Note that until now, we have not used the test data for any purpose. Let's now incorporate it into our analysis and use it to find a "fair" threshold value.

Question 17: Find the optimal hyperparameters and fit a decision tree pipeline

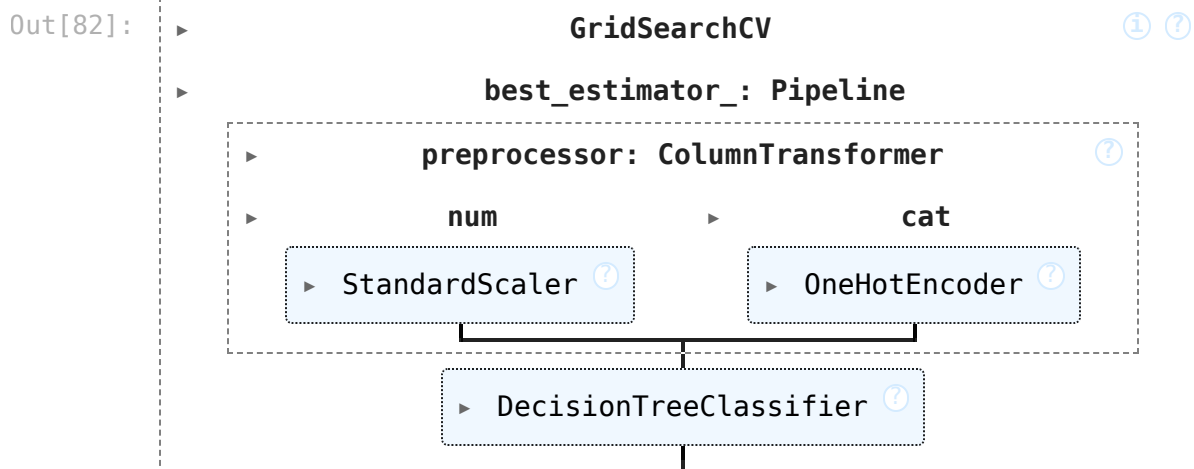
Use all of your training data to find the optimal hyperparameters and fit a decision tree pipeline.

- Report the optimal hyperparameters.

Note: Recall that nested cross-validation is only providing you with an estimate of the out-of-sample performance of the model finding procedure that involves hyperparameter optimization. At training time you should carry out the entire model finding procedure, including hyperparameter optimization.

```
In [82]: dt_pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('classifier',
dt_param_grid = {'classifier__max_depth': [3, 5, 10],
'classifier__max_features': [None, 'sqrt', 'log2'],
'classifier__max_leaf_nodes': [10, 20, 50]}
dt_grid_search = GridSearchCV(estimator=dt_pipeline, param_grid=dt_param_grid)
dt_grid_search.fit(X_train, y_train)
```

Fitting 5 folds for each of 27 candidates, totalling 135 fits



```
In [83]: print("The optimal Hyperparameters are as shown below:")
print(dt_grid_search.best_params_)
```

The optimal Hyperparameters are as shown below:

```
{'classifier__max_depth': 10, 'classifier__max_features': None, 'classifier__max_leaf_nodes': 50}
```

```
In [84]: y_proba = dt_grid_search.predict_proba(X_test)[: , 1]
```

```
In [85]: precisions, recalls, thresholds = precision_recall_curve(y_test, y_proba)
f1_scores = 2 * (precisions * recalls) / (precisions + recalls + 1e-8)
```

```
In [86]: best_idx = f1_scores.argmax()
best_threshold = thresholds[best_idx]
best_f1 = f1_scores[best_idx]
```

```
In [87]: print(f"Best threshold: {best_threshold:.4f} with f1 score of: {best_f1}")
```

Best threshold: 0.4040 with f1 score of: 0.7367298334523492

Question 18: Scores and the ROC curve

Use the fitted pipeline to compute score values for all instances in the test set.

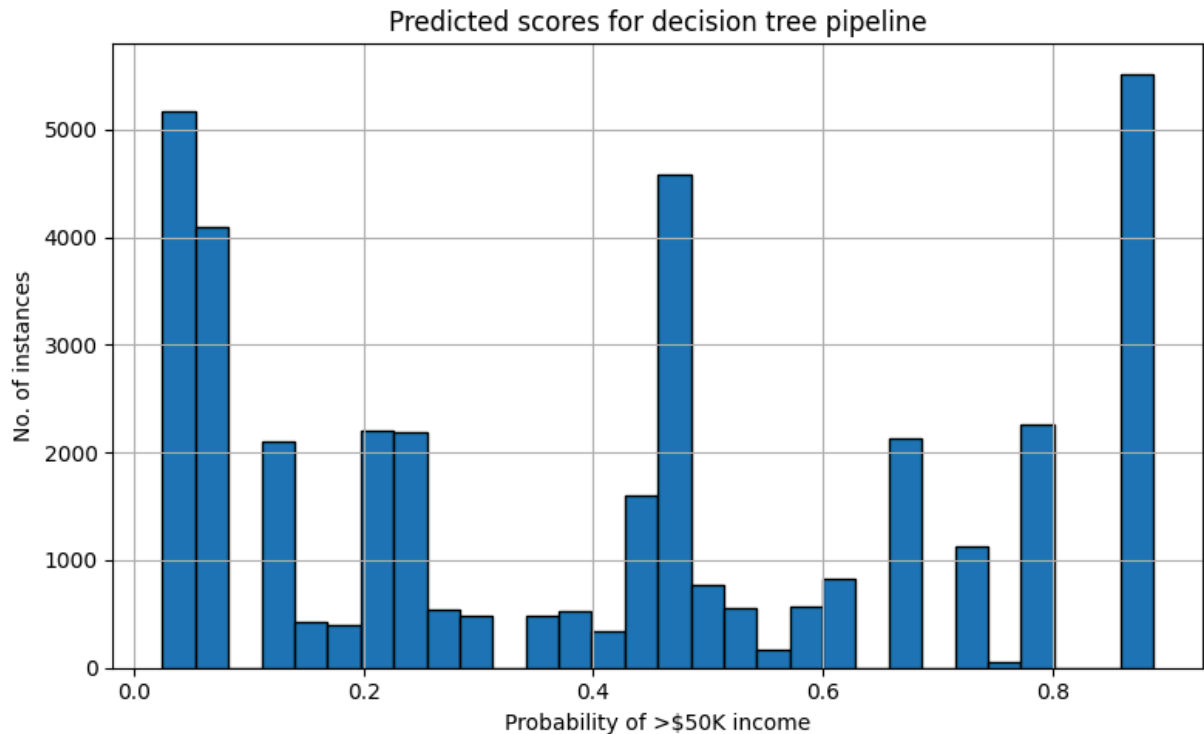
- Show the distribution of the scores in a histogram.
- Additionally, compute the (FPR,TPR) combinations for all relevant threshold values and use them to plot the ROC curve in a different figure. Following the convention, include a dashed line along the diagonal. Remember to label the axes and to make the figures as polished as possible.

Note 1: You should NOT be using `sklearn.metrics.auc` or `sklearn.metrics.RocCurveDisplay` to calculate or display the ROC curve. Please code this part yourself.

```
In [91]: best_dt = dt_grid_search.best_estimator_
```

```
In [92]: #for income > $50K
y_test_true = y_test.values.ravel()
y_test_scores = best_dt.predict_proba(X_test)[: , 1] #::this is the prob of p
```

```
In [151]: plt.figure(figsize=(8, 5))
plt.hist(y_test_scores, bins=30, edgecolor='k')
plt.title("Predicted scores for decision tree pipeline")
plt.xlabel("Probability of >$50K income")
plt.ylabel("No. of instances")
plt.grid(True)
plt.tight_layout()
plt.show()
```



```
In [94]: thresholds = np.sort(np.unique(y_test_scores))

P = np.sum(y_test_true == 1)
N = np.sum(y_test_true == 0)

tpr_list = []
fpr_list = []

for thresh in thresholds:
    y_pred = (y_test_scores >= thresh).astype(int)

    TP = np.sum((y_test_true == 1) & (y_pred == 1))
    FP = np.sum((y_test_true == 0) & (y_pred == 1))

    TPR = TP / P if P else 0
    FPR = FP / N if N else 0

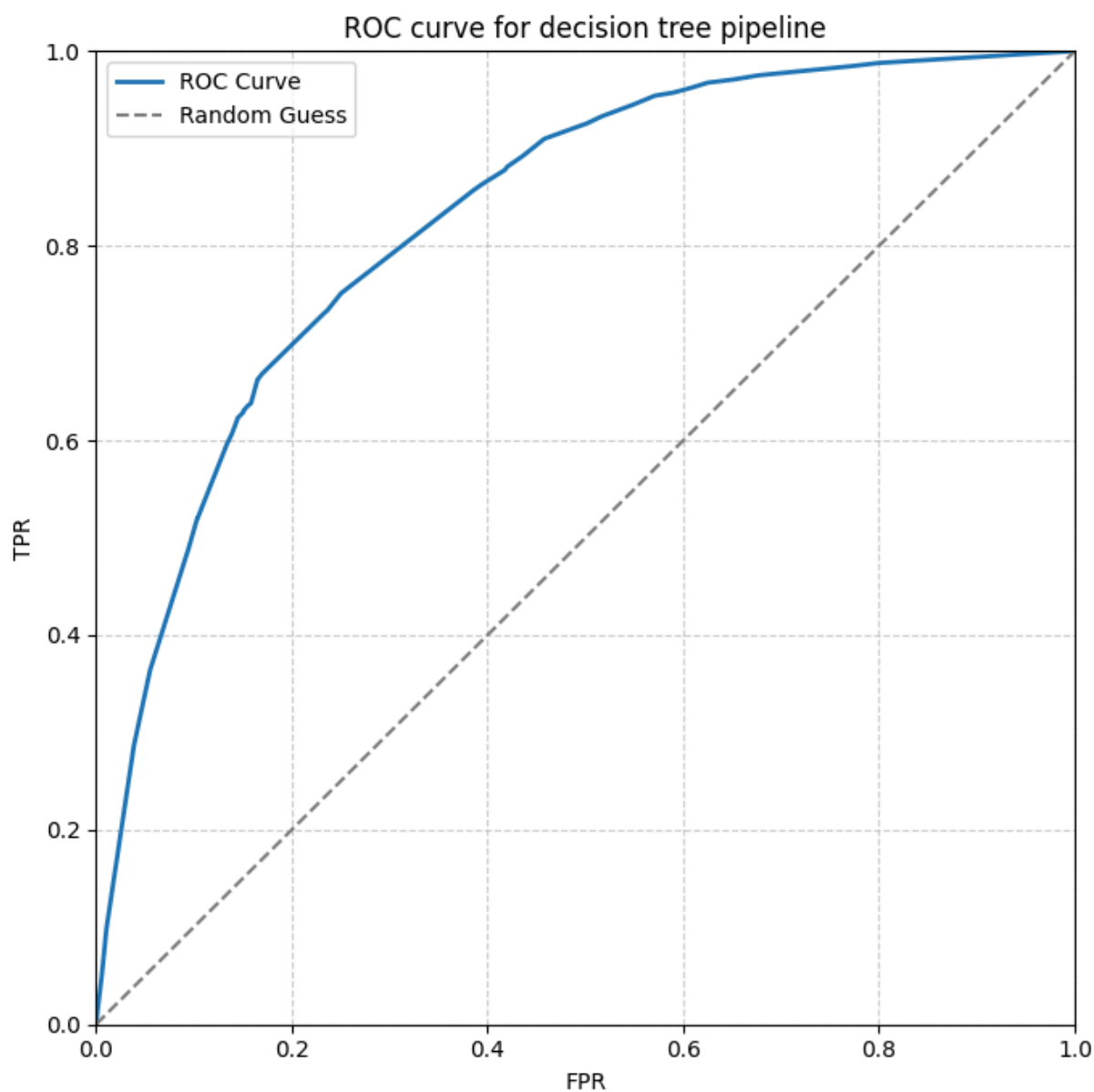
    tpr_list.append(TPR)
    fpr_list.append(FPR)
```

```
In [150... plt.figure(figsize=(7, 7))

plt.plot(fpr_list, tpr_list, label='ROC Curve', linewidth=2)
plt.plot([0, 1], [0, 1], linestyle='--', color='gray', label='Random Guess')

plt.title('ROC curve for decision tree pipeline')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
# plt.axis('square')
plt.xlim(0, 1)
```

```
plt.ylim(0, 1)
plt.tight_layout()
plt.show()
```



Question 19: ROC curves by racial groups

Compute and plot in the same figure the ROC curves for the following racial groups identified by the ACS:

- White (RAC1P==1) (plot in blue)
- African American (RAC1P==2) (plot in orange)
- Asian American (RAC1P==6) (plot in green)

Interpret the results.

```
In [181]: plt.figure(figsize=(10, 8))
```

```

g_test_values = group_test.values

for (name, code), color in zip(groups.items(), ['blue', 'orange', 'green']):
    mask = (g_test_values.flatten() == code)

    print(f"TESTING::: mask shape: {mask.shape}, y_proba shape: {y_proba.shape}")

    # first filter then running roc
    y_true_group = y_test[mask]
    y_proba_group = y_proba[mask]

    fpr, tpr, _ = roc_curve(y_true_group, y_proba_group)
    auc_score = roc_auc_score(y_true_group, y_proba_group)

    plt.plot(fpr, tpr, label=f'{name} (AUC={auc_score:.2f})', color=color)

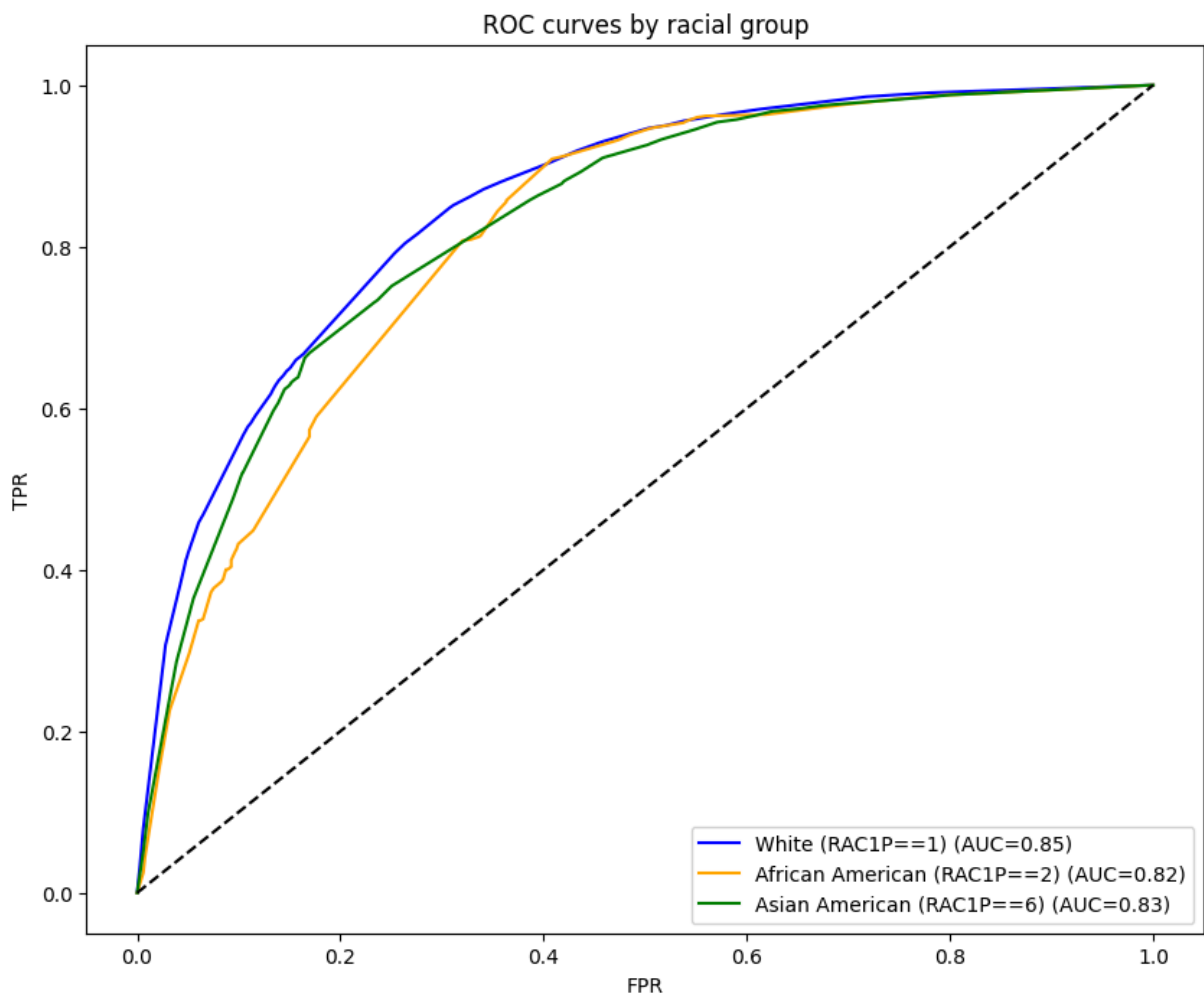
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC curves by racial group')
plt.legend()
plt.show()

```

TESTING::: mask shape: (39133,), y_proba shape: (39133,)

TESTING::: mask shape: (39133,), y_proba shape: (39133,)

TESTING::: mask shape: (39133,), y_proba shape: (39133,)



Question 20: Achieving error parity

Implement a thresholding strategy that satisfies error parity for all racial groups with $\text{FPR} = 0.25$, $\text{TPR} = 0.7$ and $\epsilon = 0.025$. In plain english, find a way of setting thresholds for the members of each group in the test data that, when evaluated on the test data, delivers FPR and TPR values that differ at most from the objective values by ϵ .

- Plot the estimated TPRs and FPRs of the racial groups in the ROC plot. Use star markers, colored accordingly.

Hint: Consider using group-specific stochastic thresholds.

```
In [196... def find_threshold(y_true, y_proba, target_fpr=0.25, target_tpr=0.7, epsilon=0.025):
    y_true = np.ravel(y_true)

    def objective(thresh):
        y_pred = (y_proba >= thresh).astype(int)
        fpr = np.mean(y_pred[y_true == 0])
        tpr = np.mean(y_pred[y_true == 1])
        return max(abs(fpr - target_fpr), abs(tpr - target_tpr))

    res = minimize_scalar(objective, bounds=(0, 1), method='bounded')
    return res.x, res.fun
```

```
In [218... plt.figure(figsize=(10, 8))
g_test_np = group_test.values.flatten()

for (name, code), color in zip(groups.items(), ['blue', 'orange', 'green']):
    mask = (g_test_np == code)
    y_true_group = np.ravel(y_test[mask])
    y_proba_group = y_proba[mask]

    fpr_curve, tpr_curve, _ = roc_curve(y_true_group, y_proba_group)
    auc = roc_auc_score(y_true_group, y_proba_group)
    plt.plot(fpr_curve, tpr_curve, color=color, label=f'{name} ROC (AUC={auc:.2f})')

    threshold, error = find_threshold(y_true_group, y_proba_group)
    y_pred = (y_proba_group >= threshold).astype(int)
    fpr = np.mean(y_pred[y_true_group == 0])
    tpr = np.mean(y_pred[y_true_group == 1])

    plt.scatter(fpr, tpr, marker='*', s=75, color=color, label=f'{name} threshold')
    print(f'{name} >> Threshold = {threshold:.4f}, FPR = {fpr:.3f}, TPR = {tpr:.3f}')

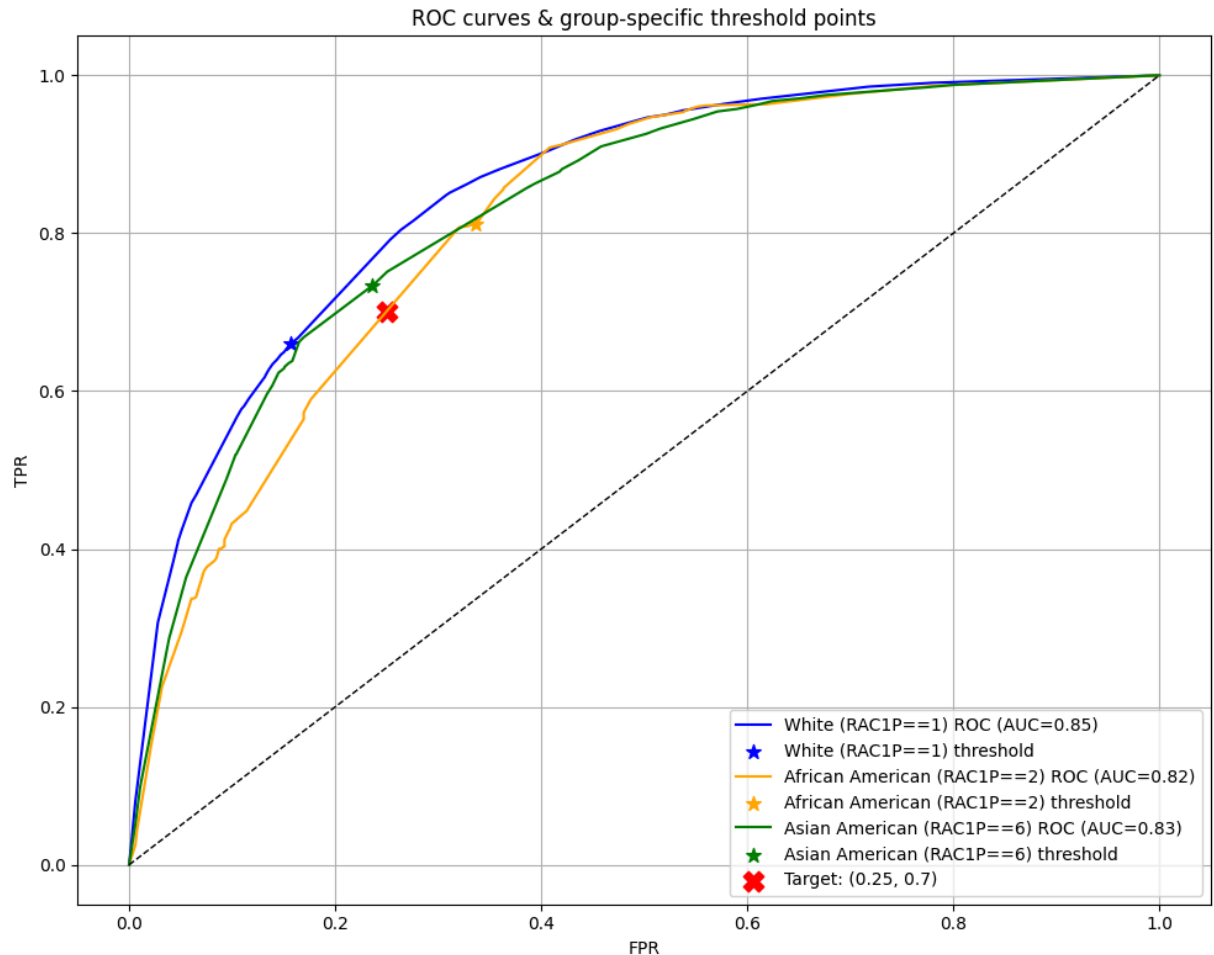
plt.scatter(0.25, 0.7, marker='X', s=140, color='red', label='Target: (0.25, 0.7)')
plt.plot([0, 1], [0, 1], 'k--', linewidth=1)
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('ROC curves & group-specific threshold points')
plt.legend()
plt.grid(True)
```

```
plt.tight_layout()
plt.show()
```

White (RAC1P==1) >> Threshold = 0.5059, FPR = 0.156, TPR = 0.660

African American (RAC1P==2) >> Threshold = 0.4270, FPR = 0.337, TPR = 0.812

Asian American (RAC1P==6) >> Threshold = 0.4754, FPR = 0.236, TPR = 0.733



Question 21: Improving the results

Can you improve the results from Question 20? That is, can you tweak your algorithm to deliver a higher TPR and lower FPR while still satisfying error parity with $\epsilon = 0.025$?

- What is the best result that you are able to achieve?
- Is there a hard limit on how much you can improve?

```
In [215... def check_curve(y_true, y_scores, fpr_range=(0.225, 0.275), tpr_range=(0.675, 0.725)):
    fpr, tpr, _ = roc_curve(y_true, y_scores)

    for f, t in zip(fpr, tpr):
        if fpr_range[0] <= f <= fpr_range[1] and tpr_range[0] <= t <= tpr_range[1]:
            return True
    return False

for (name, code) in groups.items():
    mask = (group_test.values.flatten() == code)
    y_true_group = y_test[mask].values.ravel()
```



```
y_scores_group = y_proba[mask].ravel()

passes_box = check_curve_crosses_fair_region(y_true_group, y_scores_group)

print(f"{name}: {'Passes target box' if passes_box else ' Does NOT pass'}
```

White (RAC1P==1): Does NOT pass target box

African American (RAC1P==2): Does NOT pass target box

Asian American (RAC1P==6): Does NOT pass target box

Question 22: Conclusion

What can you say about welfare and error parity? Some things to consider:

- In this income prediction task, is enforcing error parity costly?
- Would you expect this results to generalize to other predictions problems?
- Is the group definition relevant?
- Who benefits from enforcing error parity? Who doesn't?