

Ensemble

November 30, 2025

1 Import data and libraries

```
[ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import make_scorer, matthews_corrcoef
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from imblearn.over_sampling import SMOTE
from imblearn.under_sampling import RandomUnderSampler

tree_data = pd.read_csv('../data/covtype.data', header=None)
tree_cols = ['Elevation', 'Aspect', 'Slope', 'Horizontal_To_Hydrology',
             'Vertical_To_Hydrology', 'Horizontal_To_Roadways',
             'Hillshade_9am', 'Hillshade_Noon', 'Hillshade_3pm',
             'Horizontal_To_Fire'] + \
            [f'Wilderness_Area_{i}' for i in range(4)] + \
            [f'Soil_Type_{i}' for i in range(40)] + \
            ['Cover_Type']

# print(tree_data.describe())
tree_data.columns = tree_cols
print("Nan values in dataset:", tree_data.isna().sum().sum())
print(tree_data.head())
```

Nan values in dataset: 0

	Elevation	Aspect	Slope	Horizontal_To_Hydrology	Vertical_To_Hydrology	Horizontal_To_Roadways	Hillshade_9am	Hillshade_Noon	Hillshade_3pm
0	2596	51	3	258	0	510	221	232	148
1	2590	56	2	212	-6				
2	2804	139	9	268	65				
3	2785	155	18	242	118				
4	2595	45	2	153	-1				

```

1          390        220        235        151
2          3180       234        238        135
3          3090       238        238        122
4          391        220        234        150

   Horizontal_To_Fire ... Soil_Type_31  Soil_Type_32  Soil_Type_33 \
0            6279    ...          0          0          0
1            6225    ...          0          0          0
2            6121    ...          0          0          0
3            6211    ...          0          0          0
4            6172    ...          0          0          0

Soil_Type_34  Soil_Type_35  Soil_Type_36  Soil_Type_37  Soil_Type_38 \
0            0          0          0          0          0
1            0          0          0          0          0
2            0          0          0          0          0
3            0          0          0          0          0
4            0          0          0          0          0

Soil_Type_39  Cover_Type
0            0          5
1            0          5
2            0          2
3            0          2
4            0          5

```

[5 rows x 55 columns]

1.1 Split training/testing data

```

[ ]: x_tree = tree_data.drop('Cover_Type', axis=1)
y_tree = tree_data['Cover_Type']

x_train, x_test, y_train, y_test = train_test_split(x_tree, y_tree, test_size=0.
                                                    ↪2)

x_underSampled, y_underSampled = RandomUnderSampler(random_state=42, ↪
                                                    ↪sampling_strategy='majority').fit_resample(x_train, y_train)

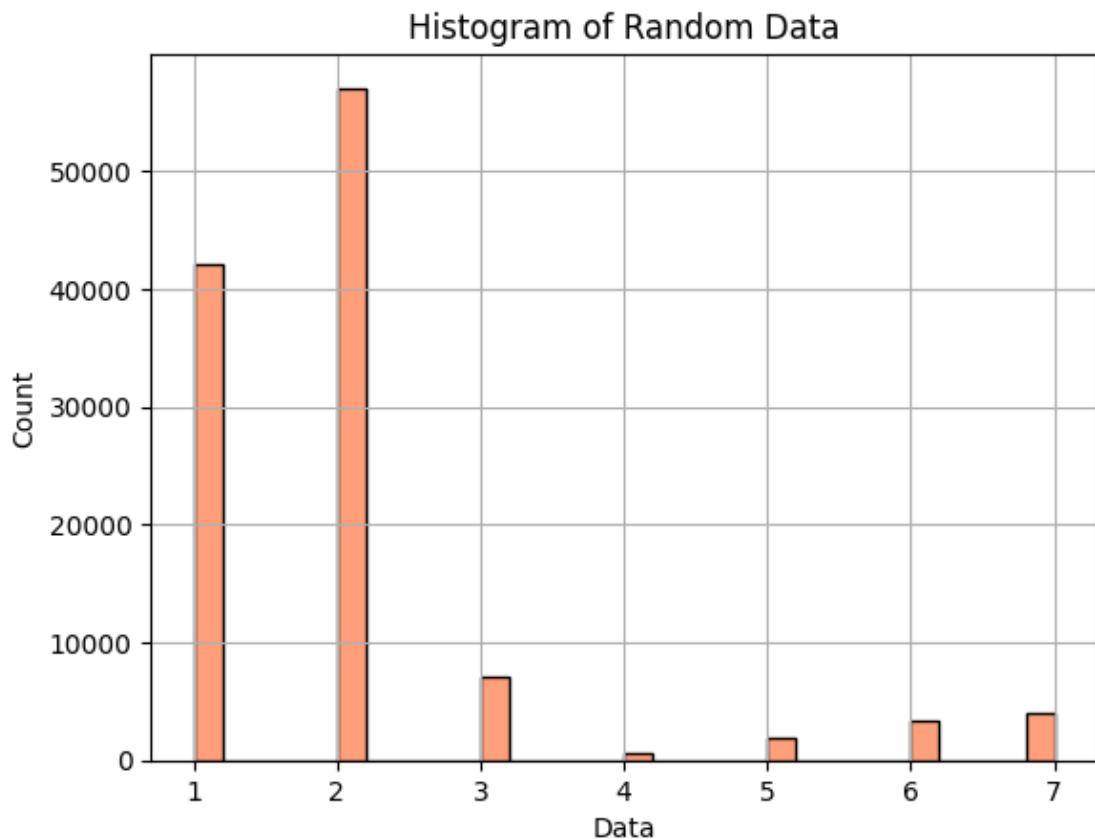
x_underSampled = x_underSampled.to_numpy()
y_underSampled = y_underSampled.to_numpy()

data = tree_data.to_numpy()

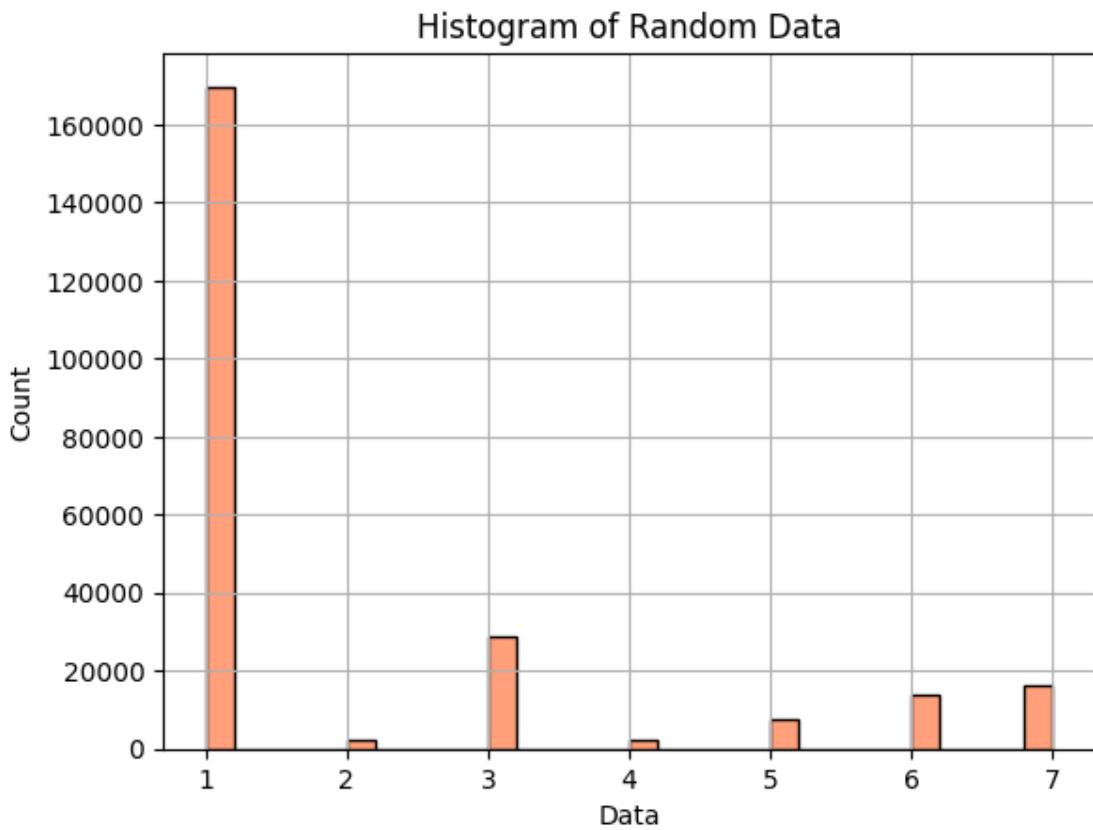
```

1.2 Class Distribution

```
[ ]: plt.hist(y_test, bins=30, color='lightsalmon', edgecolor='black')
plt.xlabel('Data')
plt.ylabel('Count')
plt.title('Histogram of Random Data')
plt.grid(True)
plt.show()
```



```
[ ]: plt.hist(y_underSampled, bins=30, color='lightsalmon', edgecolor='black')
plt.xlabel('Data')
plt.ylabel('Count')
plt.title('Histogram of Random Data')
plt.grid(True)
plt.show()
```



1.3 Scaling Dataset

```
[ ]: scalar = StandardScaler()

scaled = scalar.fit_transform(x_underSampled[:, :10])

dataScaled = np.concatenate((scaled, x_underSampled[:,10:54]), axis=1)

x_Scaled_train, x_Scaled_test, y_Scaled_train, y_Scaled_test = train_test_split(dataScaled, y_underSampled.reshape(-1, 1), test_size=0.2)
```

1.4 AdaBoost with balanced class weights

```
[ ]: ab = AdaBoostClassifier(estimator=DecisionTreeClassifier(max_depth=13,
    class_weight='balanced'), n_estimators=100)
ab.fit(x_Scaled_train, y_Scaled_train.ravel())
print("Z Scaled Test Accuracy: ", ab.score(x_Scaled_test, y_Scaled_test))
```

Z Scaled Test Accuracy: 0.9803730165323585

1.5 AdaBoost

```
[ ]: ab = AdaBoostClassifier(estimator=DecisionTreeClassifier(max_depth=13),  
    ↪n_estimators=100)  
ab.fit(x_Scaled_train, y_Scaled_train.ravel())  
print("Z Scaled Test Accuracy: ", ab.score(x_Scaled_test, y_Scaled_test))
```

Z Scaled Test Accuracy: 0.9805807094791061

1.6 GridSearch with balanced class weights

```
[ ]: mcc_scorer = make_scorer(matthews_corrcoef, greater_is_better=True)  
param_grid = {"estimator": [DecisionTreeClassifier(max_depth=7,  
    ↪class_weight='balanced'), DecisionTreeClassifier(max_depth=9,  
    ↪class_weight='balanced'), \  
        DecisionTreeClassifier(max_depth=11,  
        ↪class_weight='balanced'), DecisionTreeClassifier(max_depth=13,  
        ↪class_weight='balanced')], \  
    "n_estimators": [25, 50, 75, 100]}  
  
model_ABB = AdaBoostClassifier()  
  
grid_searchABB = GridSearchCV(model_ABB, param_grid, cv=3, scoring=mcc_scorer,  
    ↪verbose=10)  
grid_searchABB.fit(x_Scaled_train, y_Scaled_train.ravel())  
  
print(grid_searchABB.best_params_, grid_searchABB.best_score_)
```

Fitting 3 folds for each of 16 candidates, totalling 48 fits
[CV 1/3; 1/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7, n_estimators=25
[CV 1/3; 1/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7, n_estimators=25;, score=0.658 total time= 9.2s
[CV 2/3; 1/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7, n_estimators=25
[CV 2/3; 1/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7, n_estimators=25;, score=0.647 total time= 9.2s
[CV 3/3; 1/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7, n_estimators=25
[CV 3/3; 1/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7, n_estimators=25;, score=0.662 total time= 9.3s
[CV 1/3; 2/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7, n_estimators=50
[CV 1/3; 2/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7, n_estimators=50;, score=0.694 total time= 18.5s
[CV 2/3; 2/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7, n_estimators=50
[CV 2/3; 2/16] END estimator=DecisionTreeClassifier(class_weight='balanced',

```

max_depth=7), n_estimators=50;, score=0.694 total time= 18.4s
[CV 3/3; 2/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7), n_estimators=50
[CV 3/3; 2/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7), n_estimators=50;, score=0.697 total time= 18.4s
[CV 1/3; 3/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7), n_estimators=75
[CV 1/3; 3/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7), n_estimators=75;, score=0.708 total time= 27.6s
[CV 2/3; 3/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7), n_estimators=75
[CV 2/3; 3/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7), n_estimators=75;, score=0.713 total time= 27.5s
[CV 3/3; 3/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7), n_estimators=75
[CV 3/3; 3/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7), n_estimators=75;, score=0.725 total time= 27.6s
[CV 1/3; 4/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7), n_estimators=100
[CV 1/3; 4/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7), n_estimators=100;, score=0.751 total time= 36.8s
[CV 2/3; 4/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7), n_estimators=100
[CV 2/3; 4/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7), n_estimators=100;, score=0.737 total time= 36.7s
[CV 3/3; 4/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7), n_estimators=100
[CV 3/3; 4/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=7), n_estimators=100;, score=0.743 total time= 36.8s
[CV 1/3; 5/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=25
[CV 1/3; 5/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=25;, score=0.812 total time= 11.6s
[CV 2/3; 5/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=25
[CV 2/3; 5/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=25;, score=0.822 total time= 11.7s
[CV 3/3; 5/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=25
[CV 3/3; 5/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=25;, score=0.817 total time= 11.8s
[CV 1/3; 6/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=50
[CV 1/3; 6/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=50;, score=0.850 total time= 23.3s
[CV 2/3; 6/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=50
[CV 2/3; 6/16] END estimator=DecisionTreeClassifier(class_weight='balanced',

```

```

max_depth=9), n_estimators=50;, score=0.848 total time= 23.2s
[CV 3/3; 6/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=50
[CV 3/3; 6/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=50, score=0.854 total time= 23.4s
[CV 1/3; 7/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=75
[CV 1/3; 7/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=75, score=0.866 total time= 35.0s
[CV 2/3; 7/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=75
[CV 2/3; 7/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=75, score=0.872 total time= 34.8s
[CV 3/3; 7/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=75
[CV 3/3; 7/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=75, score=0.867 total time= 35.0s
[CV 1/3; 8/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=100
[CV 1/3; 8/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=100, score=0.882 total time= 47.0s
[CV 2/3; 8/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=100
[CV 2/3; 8/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=100, score=0.882 total time= 46.9s
[CV 3/3; 8/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=100
[CV 3/3; 8/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=9), n_estimators=100, score=0.875 total time= 46.6s
[CV 1/3; 9/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=25
[CV 1/3; 9/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=25, score=0.913 total time= 14.1s
[CV 2/3; 9/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=25
[CV 2/3; 9/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=25, score=0.915 total time= 14.0s
[CV 3/3; 9/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=25
[CV 3/3; 9/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=25, score=0.907 total time= 14.1s
[CV 1/3; 10/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=50
[CV 1/3; 10/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=50, score=0.932 total time= 28.2s
[CV 2/3; 10/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=50
[CV 2/3; 10/16] END estimator=DecisionTreeClassifier(class_weight='balanced',

```

```
max_depth=11), n_estimators=50;, score=0.934 total time= 28.1s
[CV 3/3; 10/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=50
[CV 3/3; 10/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=50;, score=0.935 total time= 28.2s
[CV 1/3; 11/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=75
[CV 1/3; 11/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=75;, score=0.941 total time= 42.2s
[CV 2/3; 11/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=75
[CV 2/3; 11/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=75;, score=0.940 total time= 42.1s
[CV 3/3; 11/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=75
[CV 3/3; 11/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=75;, score=0.940 total time= 42.2s
[CV 1/3; 12/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=100
[CV 1/3; 12/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=100;, score=0.942 total time= 1.1min
[CV 2/3; 12/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=100
[CV 2/3; 12/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=100;, score=0.943 total time= 56.1s
[CV 3/3; 12/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=100
[CV 3/3; 12/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=11), n_estimators=100;, score=0.946 total time= 1.3min
[CV 1/3; 13/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=25
[CV 1/3; 13/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=25;, score=0.939 total time= 27.1s
[CV 2/3; 13/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=25
[CV 2/3; 13/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=25;, score=0.939 total time= 22.0s
[CV 3/3; 13/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=25
[CV 3/3; 13/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=25;, score=0.944 total time= 17.6s
[CV 1/3; 14/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=50
[CV 1/3; 14/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=50;, score=0.947 total time= 35.4s
[CV 2/3; 14/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=50
[CV 2/3; 14/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
```

```

max_depth=13), n_estimators=50;, score=0.948 total time= 34.9s
[CV 3/3; 14/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=50
[CV 3/3; 14/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=50;, score=0.947 total time= 34.9s
[CV 1/3; 15/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=75
[CV 1/3; 15/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=75;, score=0.947 total time= 52.1s
[CV 2/3; 15/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=75
[CV 2/3; 15/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=75;, score=0.949 total time= 52.2s
[CV 3/3; 15/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=75
[CV 3/3; 15/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=75;, score=0.951 total time= 52.4s
[CV 1/3; 16/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=100
[CV 1/3; 16/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=100;, score=0.948 total time= 1.2min
[CV 2/3; 16/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=100
[CV 2/3; 16/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=100;, score=0.952 total time= 1.2min
[CV 3/3; 16/16] START estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=100
[CV 3/3; 16/16] END estimator=DecisionTreeClassifier(class_weight='balanced',
max_depth=13), n_estimators=100;, score=0.952 total time= 1.2min
{'estimator': DecisionTreeClassifier(class_weight='balanced', max_depth=13),
'n_estimators': 100} 0.9506170942498722

```

1.7 GridSearch without class weights

```

[ ]: mcc_scorer = make_scorer(matthews_corrcoef, greater_is_better=True)
param_grid = {"estimator": [DecisionTreeClassifier(max_depth=7), \
                           DecisionTreeClassifier(max_depth=9), \
                           DecisionTreeClassifier(max_depth=11), \
                           DecisionTreeClassifier(max_depth=13)], \
              "n_estimators": [25, 50, 75, 100]}

model_ab = AdaBoostClassifier()

grid_searchAB = GridSearchCV(model_ab, param_grid, cv=3, scoring=mcc_scorer, \
                             verbose=10)
grid_searchAB.fit(x_Scaled_train, y_Scaled_train.ravel())

```

```
print(grid_searchAB.best_params_, grid_searchAB.best_score_)
```

```
Fitting 3 folds for each of 16 candidates, totalling 48 fits
[CV 1/3; 1/16] START estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=25
[CV 1/3; 1/16] END estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=25;, score=0.739 total time= 9.8s
[CV 2/3; 1/16] START estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=25
[CV 2/3; 1/16] END estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=25;, score=0.706 total time= 9.8s
[CV 3/3; 1/16] START estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=25
[CV 3/3; 1/16] END estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=25;, score=0.731 total time= 9.8s
[CV 1/3; 2/16] START estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=50
[CV 1/3; 2/16] END estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=50;, score=0.756 total time= 19.8s
[CV 2/3; 2/16] START estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=50
[CV 2/3; 2/16] END estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=50;, score=0.772 total time= 19.5s
[CV 3/3; 2/16] START estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=50
[CV 3/3; 2/16] END estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=50;, score=0.771 total time= 19.8s
[CV 1/3; 3/16] START estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=75
[CV 1/3; 3/16] END estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=75;, score=0.771 total time= 29.5s
[CV 2/3; 3/16] START estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=75
[CV 2/3; 3/16] END estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=75;, score=0.764 total time= 29.6s
[CV 3/3; 3/16] START estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=75
[CV 3/3; 3/16] END estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=75;, score=0.764 total time= 29.4s
[CV 1/3; 4/16] START estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=100
[CV 1/3; 4/16] END estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=100;, score=0.776 total time= 39.4s
[CV 2/3; 4/16] START estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=100
[CV 2/3; 4/16] END estimator=DecisionTreeClassifier(max_depth=7),
n_estimators=100;, score=0.782 total time= 39.3s
```

```

[CV 3/3; 4/16] START estimator=DecisionTreeClassifier(max_depth=7) ,
n_estimators=100
[CV 3/3; 4/16] END estimator=DecisionTreeClassifier(max_depth=7) ,
n_estimators=100;, score=0.777 total time= 39.4s
[CV 1/3; 5/16] START estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=25
[CV 1/3; 5/16] END estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=25;, score=0.849 total time= 12.3s
[CV 2/3; 5/16] START estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=25
[CV 2/3; 5/16] END estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=25;, score=0.852 total time= 12.2s
[CV 3/3; 5/16] START estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=25
[CV 3/3; 5/16] END estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=25;, score=0.842 total time= 12.3s
[CV 1/3; 6/16] START estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=50
[CV 1/3; 6/16] END estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=50;, score=0.869 total time= 24.5s
[CV 2/3; 6/16] START estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=50
[CV 2/3; 6/16] END estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=50;, score=0.879 total time= 24.6s
[CV 3/3; 6/16] START estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=50
[CV 3/3; 6/16] END estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=50;, score=0.874 total time= 24.7s
[CV 1/3; 7/16] START estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=75
[CV 1/3; 7/16] END estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=75;, score=0.891 total time= 52.5s
[CV 2/3; 7/16] START estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=75
[CV 2/3; 7/16] END estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=75;, score=0.883 total time= 39.3s
[CV 3/3; 7/16] START estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=75
[CV 3/3; 7/16] END estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=75;, score=0.877 total time= 37.2s
[CV 1/3; 8/16] START estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=100
[CV 1/3; 8/16] END estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=100;, score=0.888 total time= 49.6s
[CV 2/3; 8/16] START estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=100
[CV 2/3; 8/16] END estimator=DecisionTreeClassifier(max_depth=9) ,
n_estimators=100;, score=0.896 total time= 49.5s

```

```

[CV 3/3; 8/16] START estimator=DecisionTreeClassifier(max_depth=9),
n_estimators=100
[CV 3/3; 8/16] END estimator=DecisionTreeClassifier(max_depth=9),
n_estimators=100;, score=0.886 total time= 49.5s
[CV 1/3; 9/16] START estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=25
[CV 1/3; 9/16] END estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=25;, score=0.920 total time= 14.9s
[CV 2/3; 9/16] START estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=25
[CV 2/3; 9/16] END estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=25;, score=0.921 total time= 14.9s
[CV 3/3; 9/16] START estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=25
[CV 3/3; 9/16] END estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=25;, score=0.921 total time= 14.9s
[CV 1/3; 10/16] START estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=50
[CV 1/3; 10/16] END estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=50;, score=0.933 total time= 29.8s
[CV 2/3; 10/16] START estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=50
[CV 2/3; 10/16] END estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=50;, score=0.937 total time= 29.7s
[CV 3/3; 10/16] START estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=50
[CV 3/3; 10/16] END estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=50;, score=0.935 total time= 29.8s
[CV 1/3; 11/16] START estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=75
[CV 1/3; 11/16] END estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=75;, score=0.937 total time= 44.6s
[CV 2/3; 11/16] START estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=75
[CV 2/3; 11/16] END estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=75;, score=0.941 total time= 44.5s
[CV 3/3; 11/16] START estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=75
[CV 3/3; 11/16] END estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=75;, score=0.942 total time= 44.1s
[CV 1/3; 12/16] START estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=100
[CV 1/3; 12/16] END estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=100;, score=0.942 total time= 58.8s
[CV 2/3; 12/16] START estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=100
[CV 2/3; 12/16] END estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=100;, score=0.943 total time= 58.6s

```

```

[CV 3/3; 12/16] START estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=100
[CV 3/3; 12/16] END estimator=DecisionTreeClassifier(max_depth=11),
n_estimators=100;, score=0.944 total time= 58.8s
[CV 1/3; 13/16] START estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=25
[CV 1/3; 13/16] END estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=25;, score=0.940 total time= 17.0s
[CV 2/3; 13/16] START estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=25
[CV 2/3; 13/16] END estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=25;, score=0.942 total time= 16.9s
[CV 3/3; 13/16] START estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=25
[CV 3/3; 13/16] END estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=25;, score=0.941 total time= 16.9s
[CV 1/3; 14/16] START estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=50
[CV 1/3; 14/16] END estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=50;, score=0.947 total time= 34.2s
[CV 2/3; 14/16] START estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=50
[CV 2/3; 14/16] END estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=50;, score=0.949 total time= 34.2s
[CV 3/3; 14/16] START estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=50
[CV 3/3; 14/16] END estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=50;, score=0.948 total time= 34.3s
[CV 1/3; 15/16] START estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=75
[CV 1/3; 15/16] END estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=75;, score=0.948 total time= 51.4s
[CV 2/3; 15/16] START estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=75
[CV 2/3; 15/16] END estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=75;, score=0.950 total time= 51.3s
[CV 3/3; 15/16] START estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=75
[CV 3/3; 15/16] END estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=75;, score=0.952 total time= 51.2s
[CV 1/3; 16/16] START estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=100
[CV 1/3; 16/16] END estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=100;, score=0.947 total time= 1.1min
[CV 2/3; 16/16] START estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=100
[CV 2/3; 16/16] END estimator=DecisionTreeClassifier(max_depth=13),
n_estimators=100;, score=0.950 total time= 1.1min

```

```
[CV 3/3; 16/16] START estimator=DecisionTreeClassifier(max_depth=13),  
n_estimators=100  
[CV 3/3; 16/16] END estimator=DecisionTreeClassifier(max_depth=13),  
n_estimators=100;, score=0.952 total time= 1.1min  
{'estimator': DecisionTreeClassifier(max_depth=13), 'n_estimators': 75}  
0.9497520640715948
```