MLNN End of Term Assignment

March 26, 2022

0.1 Introduction

This coursework aims to implement the universal workflow from Francois Chollet's Deep Learning with Python and apply it to classify mushrooms as either edible or not edible.

We are using the Mushroom Dataset from UCI's Machine Learning Repository to implement our model.

The overall workflow will reflect the literature:

1. Define the problem and assemble a dataset
2. Choose a measure of success
3. Decide on an evaluation protocol
4. Prepare the data
5. Develop a model that does better than a baseline
6. Develop a model that overfits
7. Regularize the model and tune its hyperparameters

1 Define the problem and assemble a dataset

Our hypothetical problem is the classification of mushrooms as edible or non-edible, with the purpose of selecting mushrooms for use in human food production. Since mushrooms can be poisonous, the goal is to prevent humans from getting sick by eating poisonous mushrooms.

Let's frame the problem in more detail. Our input data is the aforementioned dataset, which is a multivariate set of data about mushrooms and 22 features about the sample mushrooms. Based on these features, we are trying to predict if the mushrooms are suitable for human consumption.

The task at hand is binary classification - we determine if a mushroom edible or not.

Using the data information file, we can load up a list of features for our data.

[1]:
```
column_names = ['class',
                'cap-shape',
                'cap-surface',
                'cap-color',
                'bruises?',
                'odor',
                'gill-attachment',
                'gill-spacing',
                'gill-size',
```

```
                'gill-color',
                'stalk-shape',
                'stalk-root',
                'stalk-surface-above-ring',
                'stalk-surface-below-ring',
                'stalk-color-above-ring',
                'stalk-color-below-ring',
                'veil-type',
                'veil-color',
                'ring-number',
                'ring-type',
                'spore-print-color',
                'population',
                'habitat']
```

We can import our dataset using `pandas`.

[2]:
```
import pandas as pd

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/
 ↪agaricus-lepiota.data'

mushrooms = pd.read_csv(url, header=None, names=column_names)
```

2 Choose a measure of success

Since our problem involves the prevention of human sickness or death, we will use precision as a measure of success. Precision is the ability of the classifier not to label mushrooms which are poisonous as edible. This places a preference for correct predictions of poisonous mushrooms, as we prefer if edible mushrooms are misidentified as poisonous rather than poisonous mushrooms being misidentified as edible.

[4]:
```
from sklearn.metrics import precision_score
```

3 Decide on an evaluation protocol

I'll apply 10-fold cross validation to evaluate our model. While a simple holdout validation set might be enough, our dataset of 8.000 samples might not be enough.

[14]:
```
from sklearn.model_selection import train_test_split, cross_validate
```

First lets split our data into a feature matrix (`X`), and a target vector (`y`). We will use `OneHotEncoder` to encode our categorical variables.

[33]:
```
import category_encoders as ce

#Drop target feature
X = mushrooms.drop(columns='class') #Encode categorical features
```

```
X = ce.OneHotEncoder(use_cat_names=True).fit_transform(X)
y = mushrooms['class'].replace({'p':0, 'e':1})

print('Feature matrix size:',X.shape)
print('Target vector size:',len(y))
```

```
Feature matrix size: (8124, 117)
Target vector size: 8124
```

[34]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42,␣
 ↪test_size=.2, stratify=y)

print('Training feature matrix size:',X_train.shape)
print('Training target vector size:',y_train.shape)
print('Test feature matrix size:',X_test.shape)
print('Test target vector size:',y_test.shape)
```

```
Training feature matrix size: (6499, 117)
Training target vector size: (6499,)
Test feature matrix size: (1625, 117)
Test target vector size: (1625,)
```

[ ]:
```

```

[35]:
```
print(mushrooms.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8124 entries, 0 to 8123
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   class                     8124 non-null   object
 1   cap-shape                 8124 non-null   object
 2   cap-surface               8124 non-null   object
 3   cap-color                 8124 non-null   object
 4   bruises?                  8124 non-null   object
 5   odor                      8124 non-null   object
 6   gill-attachment           8124 non-null   object
 7   gill-spacing              8124 non-null   object
 8   gill-size                 8124 non-null   object
 9   gill-color                8124 non-null   object
 10  stalk-shape               8124 non-null   object
 11  stalk-root                8124 non-null   object
 12  stalk-surface-above-ring  8124 non-null   object
 13  stalk-surface-below-ring  8124 non-null   object
 14  stalk-color-above-ring    8124 non-null   object
 15  stalk-color-below-ring    8124 non-null   object
 16  veil-type                 8124 non-null   object
 17  veil-color                8124 non-null   object
```

```
 18  ring-number               8124 non-null   object
 19  ring-type                 8124 non-null   object
 20  spore-print-color         8124 non-null   object
 21  population                8124 non-null   object
 22  habitat                   8124 non-null   object
dtypes: object(23)
memory usage: 1.4+ MB
None
```

[ ]:

[36]:
```
print(mushrooms.isna().sum())
```

```
class                       0
cap-shape                   0
cap-surface                 0
cap-color                   0
bruises?                    0
odor                        0
gill-attachment             0
gill-spacing                0
gill-size                   0
gill-color                  0
stalk-shape                 0
stalk-root                  0
stalk-surface-above-ring    0
stalk-surface-below-ring    0
stalk-color-above-ring      0
stalk-color-below-ring      0
veil-type                   0
veil-color                  0
ring-number                 0
ring-type                   0
spore-print-color           0
population                  0
habitat                     0
dtype: int64
```

[ ]:

[37]:
```
import numpy as np
mushrooms = mushrooms.replace({'?':np.NaN})
print(mushrooms.isna().sum())
```

```
class                       0
cap-shape                   0
cap-surface                 0
cap-color                   0
bruises?                    0
```

```
odor                      0
gill-attachment           0
gill-spacing              0
gill-size                 0
gill-color                0
stalk-shape               0
stalk-root                0
stalk-surface-above-ring  0
stalk-surface-below-ring  0
stalk-color-above-ring    0
stalk-color-below-ring    0
veil-type                 0
veil-color                0
ring-number               0
ring-type                 0
spore-print-color         0
population                0
habitat                   0
dtype: int64
```

[ ]:

[38]:
```
mushrooms['stalk-root'] = mushrooms['stalk-root'].replace(np.NaN,'m')

print(mushrooms['stalk-root'].value_counts())
```

```
b    3776
m    2480
e    1120
c     556
r     192
Name: stalk-root, dtype: int64
```

[ ]:

[39]:
```
mushrooms['class'].value_counts(normalize=True)
```

[39]:
```
e    0.517971
p    0.482029
Name: class, dtype: float64
```

[ ]:

[40]:
```
majority_class = y_train.mode()[0]

baseline_predictions = [majority_class] * len(y_train)
```

[41]:
```
print(X_train)
```

```
      cap-shape_x  cap-shape_b  cap-shape_s  cap-shape_f  cap-shape_k  \
972             0            0            0            1            0
6451            1            0            0            0            0
3869            1            0            0            0            0
3732            0            0            0            1            0
482             0            0            0            1            0
…             …            …            …            …            …
3275            0            0            0            1            0
4913            1            0            0            0            0
5338            0            0            0            1            0
3933            1            0            0            0            0
731             1            0            0            0            0

      cap-shape_c  cap-surface_s  cap-surface_y  cap-surface_f  cap-surface_g  \
972             0              0              0              1              0
6451            0              1              0              0              0
3869            0              1              0              0              0
3732            0              0              1              0              0
482             0              1              0              0              0
…             …              …              …              …              …
3275            0              0              0              1              0
4913            0              0              1              0              0
5338            0              0              1              0              0
3933            0              0              1              0              0
731             0              0              0              1              0

      …  population_v  population_y  population_c  habitat_u  habitat_g  \
972   …              1             0             0          0          0
6451  …              1             0             0          0          0
3869  …              0             0             0          0          0
3732  …              0             1             0          0          0
482   …              1             0             0          0          0
…   …              …             …             …          …          …
3275  …              0             1             0          0          0
4913  …              0             1             0          0          0
5338  …              1             0             0          0          1
3933  …              0             1             0          0          0
731   …              0             0             0          0          1

      habitat_m  habitat_d  habitat_p  habitat_w  habitat_l
972           0          1          0          0          0
6451          0          0          0          0          1
3869          0          1          0          0          0
3732          0          1          0          0          0
482           0          1          0          0          0
…           …          …          …          …          …
3275          0          1          0          0          0
4913          0          0          1          0          0
```

```
5338          0          0          0          0          0
3933          0          0          1          0          0
731           0          0          0          0          0

[6499 rows x 117 columns]
```

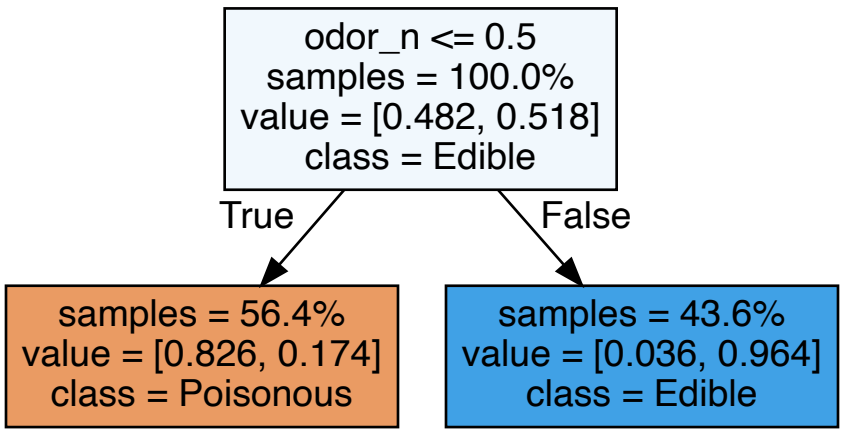
[42]:
```
from sklearn.metrics import accuracy_score

majority_class_accuracy = accuracy_score(baseline_predictions, y_train)
print(majority_class_accuracy)
```

```
0.5179258347438067
```

[ ]:

[44]:
```
from sklearn.tree import DecisionTreeClassifier
import graphviz
from sklearn.tree import export_graphviz

tree = DecisionTreeClassifier(max_depth=1)

# Fit the model
tree.fit(X_train, y_train)# Visualize the tree
dot_data = export_graphviz(tree, out_file=None, feature_names=X_train.columns,␣
 ↪class_names=['Poisonous', 'Edible'], filled=True, impurity=False,␣
 ↪proportion=True)

graphviz.Source(dot_data)
```

[44]:

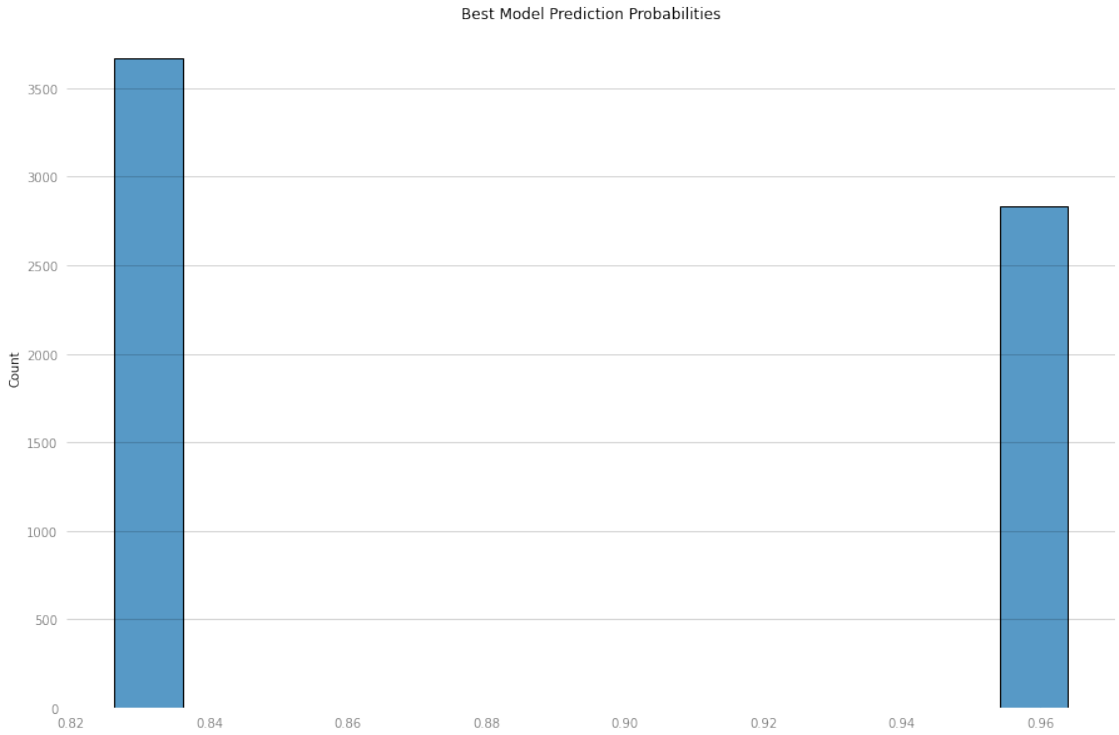


[ ]:

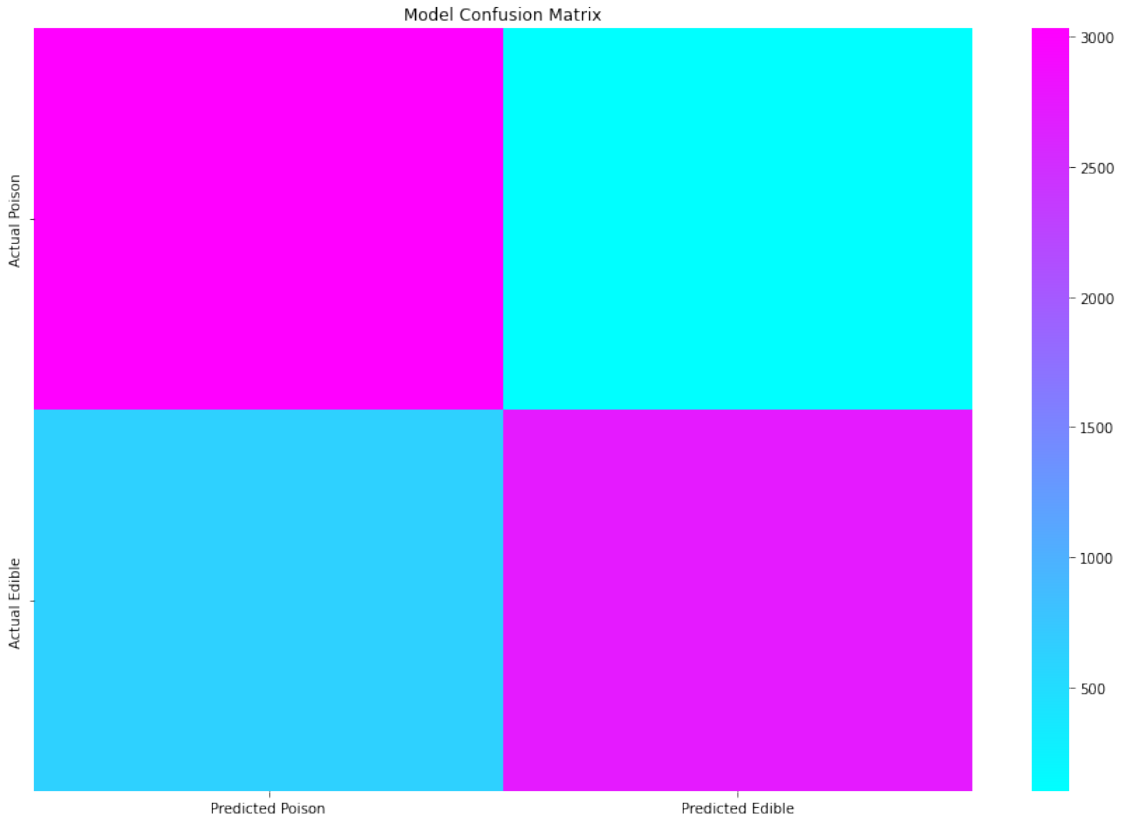
[55]:
```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report, confusion_matrix

def model_analysis(model, train_X, train_y):
        model_probabilities = model.predict_proba(train_X)

        Model_Prediction_Probability = []

        for _ in range(len(train_X)):
                x = max(model_probabilities[_])
                Model_Prediction_Probability.append(x)

        plt.figure(figsize=(15,10))

        sns.histplot(Model_Prediction_Probability)

        plt.title('Best Model Prediction Probabilities')

        # Set x and y ticks
        plt.xticks(color='gray')
        plt.yticks(color='gray')

        # Create axes object with plt. get current axes
        ax = plt.gca()

        # Set grid lines
        ax.grid(visible=True, which='major', axis='y', color='black', alpha=.2)

        # Set facecolor
        ax.set_facecolor('white')

        # Remove box
        ax.spines['top'].set_visible(False)
        ax.spines['right'].set_visible(False)
        ax.spines['bottom'].set_visible(False)
        ax.spines['left'].set_visible(False)
        ax.tick_params(color='white')

        plt.show();

        model_predictions = model.predict(train_X)

        # Classification Report
        print('\n\n', classification_report(train_y, model_predictions,␣
↪target_names=['0-Poisonous', '1-Edible']))
```

```
        # Confusion Matrix
        con_matrix = pd.DataFrame(confusion_matrix(train_y, model_predictions),␣
↪columns=['Predicted Poison', 'Predicted Edible'], index=['Actual Poison',␣
↪'Actual Edible'])

        plt.figure(figsize=(15,10))
        sns.heatmap(data=con_matrix, cmap='cool');
        plt.title('Model Confusion Matrix')
        plt.show();

        return con_matrix
```

[56]:
```
model_analysis(tree, X_train, y_train)
```



	precision	recall	f1-score	support
0-Poisonous	0.83	0.97	0.89	3133
1-Edible	0.96	0.81	0.88	3366
accuracy			0.89	6499
macro avg	0.90	0.89	0.89	6499
weighted avg	0.90	0.89	0.89	6499



[56]:

	Predicted Poison	Predicted Edible
Actual Poison	3031	102
Actual Edible	637	2729

[57]:
```
tree_predictions = tree.predict(X_train)

accuracy_score(y_train, tree_predictions)
```

[57]: 0.8862901984920757

[58]:
```
from sklearn.ensemble import RandomForestClassifier

random_forest = RandomForestClassifier(n_estimators=100, max_depth=5)

cv = cross_validate(estimator = random_forest, X = X_train, y = y_train,␣
 ↪scoring='accuracy', n_jobs=-1, cv=10, verbose=10, return_train_score=True)
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 10 concurrent workers.

[CV] START …
[CV] START
```

```
…[CV] START
…

[CV] START …
[CV] START …
[CV] START …
[CV] START …
[CV] START …
[CV] START …
[CV] START …
[CV] END …, score=(train=0.991, test=0.985) total time=   0.2s
[CV] END …, score=(train=0.990, test=0.991) total time=   0.2s
[CV] END …, score=(train=0.991, test=0.997) total time=   0.2s
[CV] END …, score=(train=0.992, test=0.989) total time=   0.2s
[CV] END …, score=(train=0.990, test=0.997) total time=   0.2s
[CV] END …, score=(train=0.990, test=0.994) total time=   0.2s
[CV] END …, score=(train=0.991, test=0.983) total time=   0.2s
[CV] END …, score=(train=0.990, test=0.992) total time=   0.2s
[CV] END …, score=(train=0.990, test=0.986) total time=   0.2s
[CV] END …, score=(train=0.990, test=0.991) total time=   0.2s

[Parallel(n_jobs=-1)]: Done   3 out of  10 | elapsed:    1.0s remaining:    2.4s
[Parallel(n_jobs=-1)]: Done   5 out of  10 | elapsed:    1.0s remaining:    1.0s
[Parallel(n_jobs=-1)]: Done   7 out of  10 | elapsed:    1.0s remaining:    0.4s
[Parallel(n_jobs=-1)]: Done  10 out of  10 | elapsed:    1.0s finished
```

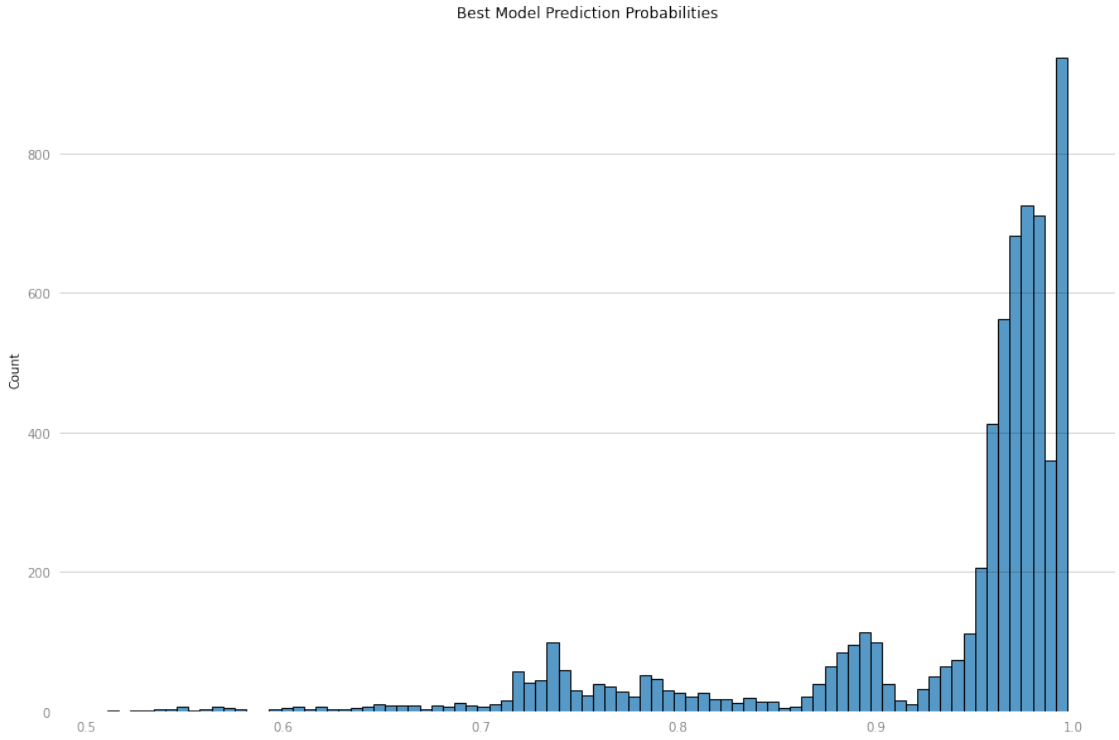
[ ]:

```
[59]: random_forest.fit(X_test, y_test)

      test_predictions = random_forest.predict(X_train)

      accuracy_score(y_train, test_predictions)
```

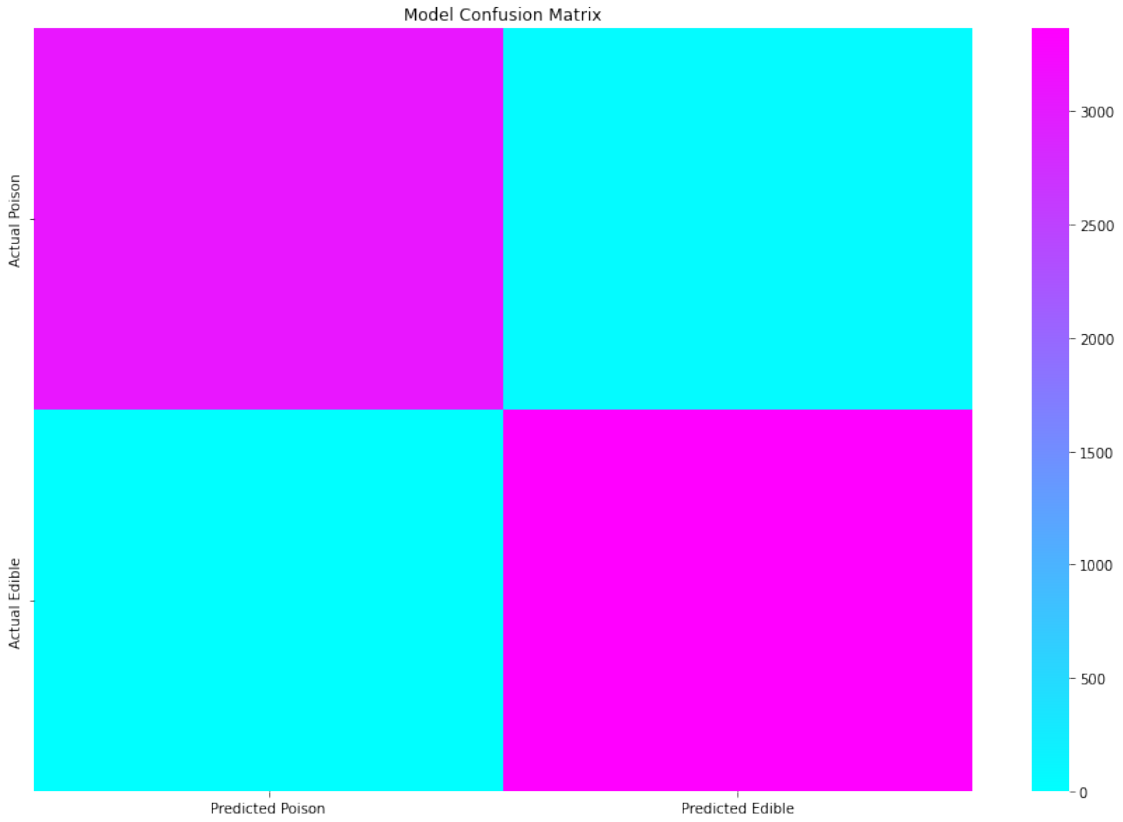
```
[59]: 0.9906139406062471
```

[ ]:

```
[60]: model_analysis(random_forest, X_train, y_train)
```



	precision	recall	f1-score	support
0-Poisonous	1.00	0.98	0.99	3133
1-Edible	0.98	1.00	0.99	3366
accuracy			0.99	6499
macro avg	0.99	0.99	0.99	6499
weighted avg	0.99	0.99	0.99	6499



[60]:

	Predicted Poison	Predicted Edible
Actual Poison	3072	61
Actual Edible	0	3366

[ ]:

[61]:
```
from sklearn.model_selection import RandomizedSearchCV

param_distributions = {
    'max_depth':[1, 2, 3, 4, 5],
    'n_estimators': [10, 25, 50, 100, 150, 200]}

search = RandomizedSearchCV(estimator = RandomForestClassifier(),␣
 ↪param_distributions = param_distributions, n_iter=100, scoring='precision',␣
 ↪n_jobs=-1, cv=10, verbose=10, return_train_score=True)

search.fit(X_train, y_train)
```

```
/opt/homebrew/lib/python3.9/site-
packages/sklearn/model_selection/_search.py:292: UserWarning: The total space of
parameters 30 is smaller than n_iter=100. Running 30 iterations. For exhaustive
searches, use GridSearchCV.
```

```
  warnings.warn(

Fitting 10 folds for each of 30 candidates, totalling 300 fits
[CV 2/10; 1/30] START max_depth=1, n_estimators=10…
[CV 1/10; 1/30] START max_depth=1, n_estimators=10…
[CV 3/10; 1/30] START max_depth=1, n_estimators=10…
[CV 5/10; 1/30] START max_depth=1, n_estimators=10…
[CV 4/10; 1/30] START max_depth=1, n_estimators=10…
[CV 6/10; 1/30] START max_depth=1, n_estimators=10…
[CV 5/10; 1/30] END max_depth=1, n_estimators=10;, score=(train=0.915,
test=0.929) total time=   0.0s
[CV 4/10; 1/30] END max_depth=1, n_estimators=10;, score=(train=0.856,
test=0.889) total time=   0.0s
[CV 1/10; 1/30] END max_depth=1, n_estimators=10;, score=(train=0.869,
test=0.844) total time=   0.0s
[CV 3/10; 1/30] END max_depth=1, n_estimators=10;, score=(train=0.898,
test=0.889) total time=   0.0s
[CV 2/10; 1/30] END max_depth=1, n_estimators=10;, score=(train=0.881,
test=0.851) total time=   0.0s
[CV 6/10; 1/30] END max_depth=1, n_estimators=10;, score=(train=0.845,
test=0.861) total time=   0.0s
[CV 7/10; 1/30] START max_depth=1, n_estimators=10…
[CV 8/10; 1/30] START max_depth=1, n_estimators=10…
[CV 9/10; 1/30] START max_depth=1, n_estimators=10…
[CV 10/10; 1/30] START max_depth=1, n_estimators=10…
[CV 1/10; 2/30] START max_depth=1, n_estimators=25…
[CV 2/10; 2/30] START max_depth=1, n_estimators=25…
[CV 3/10; 2/30] START max_depth=1, n_estimators=25…
[CV 4/10; 2/30] START max_depth=1, n_estimators=25…
[CV 7/10; 1/30] END max_depth=1, n_estimators=10;, score=(train=0.933,
test=0.942) total time=   0.0s
[CV 8/10; 1/30] END max_depth=1, n_estimators=10;, score=(train=0.847,
test=0.855) total time=   0.0s
[CV 9/10; 1/30] END max_depth=1, n_estimators=10;, score=(train=0.827,
test=0.814) total time=   0.0s
[CV 5/10; 2/30] START max_depth=1, n_estimators=25…
[CV 6/10; 2/30] START max_depth=1, n_estimators=25…
[CV 10/10; 1/30] END max_depth=1, n_estimators=10;, score=(train=0.802,
test=0.805) total time=   0.0s
[CV 7/10; 2/30] START max_depth=1, n_estimators=25…
[CV 8/10; 2/30] START max_depth=1, n_estimators=25…
[CV 1/10; 2/30] END max_depth=1, n_estimators=25;, score=(train=0.853,
test=0.804) total time=   0.0s
[CV 9/10; 2/30] START max_depth=1, n_estimators=25…
[CV 2/10; 2/30] END max_depth=1, n_estimators=25;, score=(train=0.881,
test=0.851) total time=   0.0s
[CV 3/10; 2/30] END max_depth=1, n_estimators=25;, score=(train=0.871,
test=0.867) total time=   0.0s
```

```
[CV 10/10; 2/30] START max_depth=1, n_estimators=25…
[CV 4/10; 2/30] END max_depth=1, n_estimators=25;, score=(train=0.970,
test=0.973) total time=   0.0s
[CV 1/10; 3/30] START max_depth=1, n_estimators=50…
[CV 6/10; 2/30] END max_depth=1, n_estimators=25;, score=(train=0.869,
test=0.885) total time=   0.0s
[CV 5/10; 2/30] END max_depth=1, n_estimators=25;, score=(train=0.928,
test=0.948) total time=   0.0s
[CV 7/10; 2/30] END max_depth=1, n_estimators=25;, score=(train=0.827,
test=0.834) total time=   0.0s
[CV 8/10; 2/30] END max_depth=1, n_estimators=25;, score=(train=0.856,
test=0.866) total time=   0.0s
[CV 9/10; 2/30] END max_depth=1, n_estimators=25;, score=(train=0.923,
test=0.898) total time=   0.0s
[CV 10/10; 2/30] END max_depth=1, n_estimators=25;, score=(train=0.825,
test=0.816) total time=   0.0s
[CV 3/10; 3/30] START max_depth=1, n_estimators=50…
[CV 1/10; 4/30] START max_depth=1, n_estimators=100…
[CV 7/10; 4/30] START max_depth=1, n_estimators=100…
[CV 5/10; 3/30] START max_depth=1, n_estimators=50…
[CV 7/10; 3/30] START max_depth=1, n_estimators=50…
[CV 5/10; 4/30] START max_depth=1, n_estimators=100…
[CV 9/10; 3/30] START max_depth=1, n_estimators=50…
[CV 3/10; 4/30] START max_depth=1, n_estimators=100…
[CV 1/10; 3/30] END max_depth=1, n_estimators=50;, score=(train=0.843,
test=0.793) total time=   0.1s
[CV 2/10; 3/30] START max_depth=1, n_estimators=50…
[CV 9/10; 4/30] START max_depth=1, n_estimators=100…
[CV 3/10; 3/30] END max_depth=1, n_estimators=50;, score=(train=0.830,
test=0.827) total time=   0.1s
[CV 4/10; 3/30] START max_depth=1, n_estimators=50…
[CV 9/10; 3/30] END max_depth=1, n_estimators=50;, score=(train=0.835,
test=0.823) total time=   0.1s
[CV 10/10; 3/30] START max_depth=1, n_estimators=50…
[CV 5/10; 3/30] END max_depth=1, n_estimators=50;, score=(train=0.825,
test=0.857) total time=   0.1s
[CV 6/10; 3/30] START max_depth=1, n_estimators=50…
[CV 7/10; 3/30] END max_depth=1, n_estimators=50;, score=(train=0.826,
test=0.838) total time=   0.1s
[CV 8/10; 3/30] START max_depth=1, n_estimators=50…
[CV 4/10; 3/30] END max_depth=1, n_estimators=50;, score=(train=0.844,
test=0.875) total time=   0.1s
[CV 1/10; 5/30] START max_depth=1, n_estimators=150…
[CV 5/10; 4/30] END max_depth=1, n_estimators=100;, score=(train=0.880,
test=0.911) total time=   0.1s
[CV 6/10; 4/30] START max_depth=1, n_estimators=100…
[CV 2/10; 3/30] END max_depth=1, n_estimators=50;, score=(train=0.852,
test=0.822) total time=   0.1s
```

```
[CV 3/10; 5/30] START max_depth=1, n_estimators=150…
[CV 6/10; 3/30] END max_depth=1, n_estimators=50;, score=(train=0.906,
test=0.931) total time=   0.1s
[CV 5/10; 5/30] START max_depth=1, n_estimators=150…
[CV 10/10; 3/30] END max_depth=1, n_estimators=50;, score=(train=0.827,
test=0.824) total time=   0.1s
[CV 8/10; 3/30] END max_depth=1, n_estimators=50;, score=(train=0.830,
test=0.838) total time=   0.1s
[CV 7/10; 5/30] START max_depth=1, n_estimators=150…
[CV 9/10; 5/30] START max_depth=1, n_estimators=150…
[CV 9/10; 4/30] END max_depth=1, n_estimators=100;, score=(train=0.863,
test=0.851) total time=   0.1s
[CV 10/10; 4/30] START max_depth=1, n_estimators=100…
[CV 1/10; 4/30] END max_depth=1, n_estimators=100;, score=(train=0.835,
test=0.787) total time=   0.1s
[CV 2/10; 4/30] START max_depth=1, n_estimators=100…
[CV 7/10; 4/30] END max_depth=1, n_estimators=100;, score=(train=0.878,
test=0.875) total time=   0.2s
[CV 8/10; 4/30] START max_depth=1, n_estimators=100…
[CV 3/10; 4/30] END max_depth=1, n_estimators=100;, score=(train=0.852,
test=0.849) total time=   0.2s
[CV 4/10; 4/30] START max_depth=1, n_estimators=100…
[CV 6/10; 4/30] END max_depth=1, n_estimators=100;, score=(train=0.834,
test=0.849) total time=   0.1s
[CV 1/10; 6/30] START max_depth=1, n_estimators=200…
[CV 2/10; 4/30] END max_depth=1, n_estimators=100;, score=(train=0.844,
test=0.820) total time=   0.1s
[CV 3/10; 6/30] START max_depth=1, n_estimators=200…
[CV 10/10; 4/30] END max_depth=1, n_estimators=100;, score=(train=0.868,
test=0.868) total time=   0.2s
[CV 8/10; 4/30] END max_depth=1, n_estimators=100;, score=(train=0.837,
test=0.851) total time=   0.1s
[CV 5/10; 6/30] START max_depth=1, n_estimators=200…
[CV 7/10; 6/30] START max_depth=1, n_estimators=200…
[CV 4/10; 4/30] END max_depth=1, n_estimators=100;, score=(train=0.840,
test=0.885) total time=   0.2s
[CV 3/10; 5/30] END max_depth=1, n_estimators=150;, score=(train=0.833,
test=0.830) total time=   0.2s
[CV 4/10; 5/30] START max_depth=1, n_estimators=150…
[CV 9/10; 6/30] START max_depth=1, n_estimators=200…
[CV 5/10; 5/30] END max_depth=1, n_estimators=150;, score=(train=0.848,
test=0.877) total time=   0.2s
[CV 6/10; 5/30] START max_depth=1, n_estimators=150…
[CV 1/10; 5/30] END max_depth=1, n_estimators=150;, score=(train=0.864,
test=0.814) total time=   0.2s
[CV 2/10; 5/30] START max_depth=1, n_estimators=150…
[CV 7/10; 5/30] END max_depth=1, n_estimators=150;, score=(train=0.837,
test=0.844) total time=   0.2s
```

```
[CV 8/10; 5/30] START max_depth=1, n_estimators=150…
[CV 9/10; 5/30] END max_depth=1, n_estimators=150;, score=(train=0.850,
test=0.838) total time=   0.2s
[CV 10/10; 5/30] START max_depth=1, n_estimators=150…
[CV 6/10; 5/30] END max_depth=1, n_estimators=150;, score=(train=0.874,
test=0.895) total time=   0.2s
[CV 4/10; 5/30] END max_depth=1, n_estimators=150;, score=(train=0.835,
test=0.869) total time=   0.2s
[CV 8/10; 5/30] END max_depth=1, n_estimators=150;, score=(train=0.855,
test=0.866) total time=   0.2s
[CV 2/10; 5/30] END max_depth=1, n_estimators=150;, score=(train=0.840,
test=0.815) total time=   0.2s
[CV 1/10; 7/30] START max_depth=2,
n_estimators=10…[CV 3/10; 7/30] START max_depth=2,
n_estimators=10…

[CV 7/10; 7/30] START max_depth=2,
n_estimators=10…[CV 10/10; 5/30] END max_depth=1,
n_estimators=150;, score=(train=0.837, test=0.836) total time=   0.2s

[CV 5/10; 7/30] START max_depth=2, n_estimators=10…
[CV 9/10; 7/30] START max_depth=2, n_estimators=10…
[CV 1/10; 7/30] END max_depth=2, n_estimators=10;, score=(train=0.879,
test=0.843) total time=   0.0s
[CV 2/10; 7/30] START max_depth=2, n_estimators=10…
[CV 3/10; 7/30] END max_depth=2, n_estimators=10;, score=(train=0.881,
test=0.875) total time=   0.0s
[CV 4/10; 7/30] START max_depth=2, n_estimators=10…
[CV 7/10; 6/30] END max_depth=1, n_estimators=200;, score=(train=0.863,
test=0.857) total time=   0.2s
[CV 8/10; 6/30] START max_depth=1, n_estimators=200…
[CV 3/10; 6/30] END max_depth=1, n_estimators=200;, score=(train=0.858,
test=0.849) total time=   0.2s
[CV 4/10; 6/30] START max_depth=1, n_estimators=200…
[CV 5/10; 7/30] END max_depth=2, n_estimators=10;, score=(train=0.948,
test=0.954) total time=   0.0s
[CV 6/10; 7/30] START max_depth=2, n_estimators=10…
[CV 7/10; 7/30] END max_depth=2, n_estimators=10;, score=(train=0.921,
test=0.933) total time=   0.0s
[CV 8/10; 7/30] START max_depth=2, n_estimators=10…
[CV 9/10; 7/30] END max_depth=2, n_estimators=10;, score=(train=0.973,
test=0.957) total time=   0.0s
[CV 10/10; 7/30] START max_depth=2, n_estimators=10…
[CV 1/10; 6/30] END max_depth=1, n_estimators=200;, score=(train=0.870,
test=0.824) total time=   0.3s
[CV 2/10; 6/30] START max_depth=1, n_estimators=200…
[CV 2/10; 7/30] END max_depth=2, n_estimators=10;, score=(train=0.915,
test=0.880) total time=   0.0s
```

```
[CV 4/10; 7/30] END max_depth=2, n_estimators=10;, score=(train=0.898,
test=0.911) total time=   0.0s
[CV 1/10; 8/30] START max_depth=2, n_estimators=25…
[CV 5/10; 6/30] END max_depth=1, n_estimators=200;, score=(train=0.829,
test=0.862) total time=   0.2s
[CV 6/10; 6/30] START max_depth=1, n_estimators=200…
[CV 3/10; 8/30] START max_depth=2, n_estimators=25…
[CV 6/10; 7/30] END max_depth=2, n_estimators=10;, score=(train=0.855,
test=0.853) total time=   0.0s
[CV 5/10; 8/30] START max_depth=2, n_estimators=25…
[CV 8/10; 7/30] END max_depth=2, n_estimators=10;, score=(train=0.952,
test=0.968) total time=   0.0s
[CV 10/10; 7/30] END max_depth=2, n_estimators=10;, score=(train=0.946,
test=0.938) total time=   0.0s
[CV 9/10; 6/30] END max_depth=1, n_estimators=200;, score=(train=0.834,
test=0.820) total time=   0.2s
[CV 10/10; 6/30] START max_depth=1, n_estimators=200…
[CV 7/10; 8/30] START max_depth=2, n_estimators=25…
[CV 9/10; 8/30] START max_depth=2, n_estimators=25…
[CV 3/10; 8/30] END max_depth=2, n_estimators=25;, score=(train=0.909,
test=0.895) total time=   0.0s
[CV 4/10; 8/30] START max_depth=2, n_estimators=25…
[CV 1/10; 8/30] END max_depth=2, n_estimators=25;, score=(train=0.956,
test=0.943) total time=   0.0s
[CV 2/10; 8/30] START max_depth=2, n_estimators=25…
[CV 7/10; 8/30] END max_depth=2, n_estimators=25;, score=(train=0.929,
test=0.936) total time=   0.0s
[CV 8/10; 8/30] START max_depth=2, n_estimators=25…
[CV 9/10; 8/30] END max_depth=2, n_estimators=25;, score=(train=0.902,
test=0.869) total time=   0.0s
[CV 10/10; 8/30] START max_depth=2, n_estimators=25…
[CV 5/10; 8/30] END max_depth=2, n_estimators=25;, score=(train=0.891,
test=0.922) total time=   0.1s
[CV 6/10; 8/30] START max_depth=2, n_estimators=25…
[CV 2/10; 8/30] END max_depth=2, n_estimators=25;, score=(train=0.887,
test=0.856) total time=   0.0s
[CV 10/10; 8/30] END max_depth=2, n_estimators=25;, score=(train=0.846,
test=0.842) total time=   0.0s
[CV 8/10; 8/30] END max_depth=2, n_estimators=25;, score=(train=0.857,
test=0.862) total time=   0.0s
[CV 4/10; 8/30] END max_depth=2, n_estimators=25;, score=(train=0.896,
test=0.916) total time=   0.1s
[CV 7/10; 9/30] START max_depth=2, n_estimators=50…
[CV 3/10; 9/30] START max_depth=2, n_estimators=50…
[CV 1/10; 9/30] START max_depth=2, n_estimators=50…
[CV 6/10; 8/30] END max_depth=2, n_estimators=25;, score=(train=0.891,
test=0.908) total time=   0.0s
[CV 9/10; 9/30] START max_depth=2, n_estimators=50…
```

```
[CV 5/10; 9/30] START max_depth=2, n_estimators=50…
[CV 1/10; 9/30] END max_depth=2, n_estimators=50;, score=(train=0.951,
test=0.949) total time=   0.1s
[CV 8/10; 6/30] END max_depth=1, n_estimators=200;, score=(train=0.835,
test=0.848) total time=   0.2s
[CV 2/10; 9/30] START max_depth=2, n_estimators=50…
[CV 7/10; 9/30] END max_depth=2, n_estimators=50;, score=(train=0.876,
test=0.874) total time=   0.1s
[CV 8/10; 9/30] START max_depth=2, n_estimators=50…
[CV 1/10; 10/30] START max_depth=2, n_estimators=100…
[CV 3/10; 9/30] END max_depth=2, n_estimators=50;, score=(train=0.857,
test=0.856) total time=   0.1s
[CV 4/10; 9/30] START max_depth=2, n_estimators=50…
[CV 2/10; 6/30] END max_depth=1, n_estimators=200;, score=(train=0.862,
test=0.826) total time=   0.2s
[CV 9/10; 9/30] END max_depth=2, n_estimators=50;, score=(train=0.888,
test=0.866) total time=   0.1s
[CV 10/10; 9/30] START max_depth=2, n_estimators=50…
[CV 5/10; 9/30] END max_depth=2, n_estimators=50;, score=(train=0.871,
test=0.896) total time=   0.1s
[CV 6/10; 9/30] START max_depth=2, n_estimators=50…
[CV 3/10; 10/30] START max_depth=2, n_estimators=100…
[CV 6/10; 6/30] END max_depth=1, n_estimators=200;, score=(train=0.836,
test=0.850) total time=   0.2s
[CV 2/10; 9/30] END max_depth=2, n_estimators=50;, score=(train=0.910,
test=0.896) total time=   0.1s
[CV 4/10; 6/30] END max_depth=1, n_estimators=200;, score=(train=0.854,
test=0.885) total time=   0.3s
[CV 5/10; 10/30] START max_depth=2, n_estimators=100…
[CV 7/10; 10/30] START max_depth=2, n_estimators=100…
[CV 4/10; 9/30] END max_depth=2, n_estimators=50;, score=(train=0.923,
test=0.947) total time=   0.1s
[CV 8/10; 9/30] END max_depth=2, n_estimators=50;, score=(train=0.866,
test=0.886) total time=   0.1s
[CV 9/10; 10/30] START max_depth=2, n_estimators=100…
[CV 6/10; 9/30] END max_depth=2, n_estimators=50;, score=(train=0.933,
test=0.966) total time=   0.1s
[CV 10/10; 9/30] END max_depth=2, n_estimators=50;, score=(train=0.858,
test=0.851) total time=   0.1s
[CV 10/10; 6/30] END max_depth=1, n_estimators=200;, score=(train=0.846,
test=0.841) total time=   0.3s
[CV 1/10; 11/30] START max_depth=2, n_estimators=150…
[CV 1/10; 10/30] END max_depth=2, n_estimators=100;, score=(train=0.875,
test=0.848) total time=   0.1s
[CV 2/10; 10/30] START max_depth=2, n_estimators=100…
[CV 5/10; 11/30] START max_depth=2, n_estimators=150…
[CV 9/10; 11/30] START max_depth=2, n_estimators=150…
[CV 3/10; 11/30] START max_depth=2, n_estimators=150…
```

```
[CV 7/10; 11/30] START max_depth=2, n_estimators=150…
[CV 3/10; 10/30] END max_depth=2, n_estimators=100;, score=(train=0.896,
test=0.880) total time=   0.1s
[CV 4/10; 10/30] START max_depth=2, n_estimators=100…
[CV 7/10; 10/30] END max_depth=2, n_estimators=100;, score=(train=0.872,
test=0.873) total time=   0.1s
[CV 8/10; 10/30] START max_depth=2, n_estimators=100…
[CV 9/10; 10/30] END max_depth=2, n_estimators=100;, score=(train=0.951,
test=0.928) total time=   0.1s
[CV 10/10; 10/30] START max_depth=2, n_estimators=100…
[CV 5/10; 10/30] END max_depth=2, n_estimators=100;, score=(train=0.904,
test=0.921) total time=   0.2s
[CV 6/10; 10/30] START max_depth=2, n_estimators=100…
[CV 2/10; 10/30] END max_depth=2, n_estimators=100;, score=(train=0.891,
test=0.880) total time=   0.1s
[CV 4/10; 10/30] END max_depth=2, n_estimators=100;, score=(train=0.933,
test=0.954) total time=   0.2s
[CV 1/10; 12/30] START max_depth=2, n_estimators=200…
[CV 5/10; 11/30] END max_depth=2, n_estimators=150;, score=(train=0.883,
test=0.906) total time=   0.2s
[CV 6/10; 11/30] START max_depth=2, n_estimators=150…
[CV 1/10; 11/30] END max_depth=2, n_estimators=150;, score=(train=0.900,
test=0.882) total time=   0.2s
[CV 2/10; 11/30] START max_depth=2, n_estimators=150…
[CV 3/10; 11/30] END max_depth=2, n_estimators=150;, score=(train=0.918,
test=0.918) total time=   0.2s
[CV 4/10; 11/30] START max_depth=2, n_estimators=150…
[CV 9/10; 11/30] END max_depth=2, n_estimators=150;, score=(train=0.914,
test=0.887) total time=   0.2s
[CV 10/10; 11/30] START max_depth=2, n_estimators=150…
[CV 10/10; 10/30] END max_depth=2, n_estimators=100;, score=(train=0.880,
test=0.873) total time=   0.1s
[CV 3/10; 12/30] START max_depth=2, n_estimators=200…
[CV 8/10; 10/30] END max_depth=2, n_estimators=100;, score=(train=0.939,
test=0.957) total time=   0.2s
[CV 6/10; 10/30] END max_depth=2, n_estimators=100;, score=(train=0.902,
test=0.921) total time=   0.1s
[CV 5/10; 12/30] START max_depth=2, n_estimators=200…
[CV 7/10; 11/30] END max_depth=2, n_estimators=150;, score=(train=0.862,
test=0.857) total time=   0.2s
[CV 8/10; 11/30] START max_depth=2, n_estimators=150…
[CV 7/10; 12/30] START max_depth=2, n_estimators=200…
[CV 9/10; 12/30] START max_depth=2, n_estimators=200…
[CV 6/10; 11/30] END max_depth=2, n_estimators=150;, score=(train=0.912,
test=0.926) total time=   0.2s
[CV 10/10; 11/30] END max_depth=2, n_estimators=150;, score=(train=0.909,
test=0.908) total time=   0.2s
[CV 2/10; 11/30] END max_depth=2, n_estimators=150;, score=(train=0.880,
```

```
test=0.851) total time=   0.2s
[CV 4/10; 11/30] END max_depth=2, n_estimators=150;, score=(train=0.941,
test=0.947) total time=   0.2s
[CV 3/10; 13/30] START max_depth=3, n_estimators=10…
[CV 8/10; 11/30] END max_depth=2, n_estimators=150;, score=(train=0.923,
test=0.936) total time=   0.2s
[CV 1/10; 13/30] START max_depth=3, n_estimators=10…
[CV 5/10; 13/30] START max_depth=3, n_estimators=10…
[CV 7/10; 13/30] START max_depth=3, n_estimators=10…
[CV 9/10; 13/30] START max_depth=3, n_estimators=10…
[CV 3/10; 13/30] END max_depth=3, n_estimators=10;, score=(train=0.922,
test=0.911) total time=   0.0s
[CV 4/10; 13/30] START max_depth=3, n_estimators=10…
[CV 1/10; 13/30] END max_depth=3, n_estimators=10;, score=(train=0.982,
test=0.974) total time=   0.0s
[CV 2/10; 13/30] START max_depth=3, n_estimators=10…
[CV 5/10; 13/30] END max_depth=3, n_estimators=10;, score=(train=0.911,
test=0.933) total time=   0.0s
[CV 6/10; 13/30] START max_depth=3, n_estimators=10…
[CV 7/10; 13/30] END max_depth=3, n_estimators=10;, score=(train=0.917,
test=0.926) total time=   0.0s
[CV 8/10; 13/30] START max_depth=3, n_estimators=10…
[CV 1/10; 12/30] END max_depth=2, n_estimators=200;, score=(train=0.897,
test=0.882) total time=   0.3s
[CV 2/10; 12/30] START max_depth=2, n_estimators=200…
[CV 9/10; 13/30] END max_depth=3, n_estimators=10;, score=(train=0.903,
test=0.898) total time=   0.0s
[CV 10/10; 13/30] START max_depth=3, n_estimators=10…
[CV 4/10; 13/30] END max_depth=3, n_estimators=10;, score=(train=0.932,
test=0.947) total time=   0.0s
[CV 6/10; 13/30] END max_depth=3, n_estimators=10;, score=(train=0.974,
test=0.988) total time=   0.0s
[CV 2/10; 13/30] END max_depth=3, n_estimators=10;, score=(train=0.953,
test=0.929) total time=   0.0s
[CV 8/10; 13/30] END max_depth=3, n_estimators=10;, score=(train=0.913,
test=0.939) total time=   0.0s
[CV 10/10; 13/30] END max_depth=3, n_estimators=10;, score=(train=0.964,
test=0.948) total time=   0.0s
[CV 3/10; 12/30] END max_depth=2, n_estimators=200;, score=(train=0.884,
test=0.885) total time=   0.3s
[CV 4/10; 12/30] START max_depth=2, n_estimators=200…
[CV 3/10; 14/30] START max_depth=3, n_estimators=25…
[CV 1/10; 14/30] START max_depth=3, n_estimators=25…
[CV 9/10; 14/30] START max_depth=3, n_estimators=25…
[CV 5/10; 14/30] START max_depth=3, n_estimators=25…
[CV 7/10; 14/30] START max_depth=3, n_estimators=25…
[CV 9/10; 12/30] END max_depth=2, n_estimators=200;, score=(train=0.875,
test=0.856) total time=   0.3s
```

```
[CV 10/10; 12/30] START max_depth=2, n_estimators=200…
[CV 5/10; 12/30] END max_depth=2, n_estimators=200;, score=(train=0.902,
test=0.918) total time=   0.3s
[CV 6/10; 12/30] START max_depth=2, n_estimators=200…
[CV 1/10; 14/30] END max_depth=3, n_estimators=25;, score=(train=0.978,
test=0.968) total time=   0.0s
[CV 2/10; 14/30] START max_depth=3, n_estimators=25…
[CV 5/10; 14/30] END max_depth=3, n_estimators=25;, score=(train=0.897,
test=0.918) total time=   0.0s
[CV 6/10; 14/30] START max_depth=3, n_estimators=25…
[CV 7/10; 14/30] END max_depth=3, n_estimators=25;, score=(train=0.980,
test=0.985) total time=   0.0s
[CV 8/10; 14/30] START max_depth=3, n_estimators=25…
[CV 9/10; 14/30] END max_depth=3, n_estimators=25;, score=(train=0.973,
test=0.957) total time=   0.0s
[CV 10/10; 14/30] START max_depth=3, n_estimators=25…
[CV 3/10; 14/30] END max_depth=3, n_estimators=25;, score=(train=0.980,
test=0.982) total time=   0.1s
[CV 4/10; 14/30] START max_depth=3, n_estimators=25…
[CV 7/10; 12/30] END max_depth=2, n_estimators=200;, score=(train=0.936,
test=0.931) total time=   0.3s
[CV 8/10; 12/30] START max_depth=2, n_estimators=200…
[CV 6/10; 14/30] END max_depth=3, n_estimators=25;, score=(train=0.923,
test=0.949) total time=   0.0s
[CV 8/10; 14/30] END max_depth=3, n_estimators=25;, score=(train=0.935,
test=0.946) total time=   0.0s
[CV 2/10; 14/30] END max_depth=3, n_estimators=25;, score=(train=0.980,
test=0.971) total time=   0.0s
[CV 4/10; 14/30] END max_depth=3, n_estimators=25;, score=(train=0.974,
test=0.971) total time=   0.0s
[CV 10/10; 14/30] END max_depth=3, n_estimators=25;, score=(train=0.957,
test=0.944) total time=   0.0s
[CV 1/10; 15/30] START max_depth=3, n_estimators=50…
[CV 3/10; 15/30] START max_depth=3, n_estimators=50…
[CV 9/10; 15/30] START max_depth=3, n_estimators=50…
[CV 5/10; 15/30] START max_depth=3, n_estimators=50…
[CV 7/10; 15/30] START max_depth=3, n_estimators=50…
[CV 1/10; 15/30] END max_depth=3, n_estimators=50;, score=(train=0.969,
test=0.966) total time=   0.1s
[CV 2/10; 15/30] START max_depth=3, n_estimators=50…
[CV 5/10; 15/30] END max_depth=3, n_estimators=50;, score=(train=0.973,
test=0.977) total time=   0.1s
[CV 6/10; 15/30] START max_depth=3, n_estimators=50…
[CV 3/10; 15/30] END max_depth=3, n_estimators=50;, score=(train=0.980,
test=0.983) total time=   0.1s
[CV 4/10; 15/30] START max_depth=3, n_estimators=50…
[CV 7/10; 15/30] END max_depth=3, n_estimators=50;, score=(train=0.981,
test=0.985) total time=   0.1s
```

```
[CV 8/10; 15/30] START max_depth=3, n_estimators=50…
[CV 9/10; 15/30] END max_depth=3, n_estimators=50;, score=(train=0.981,
test=0.963) total time=   0.1s
[CV 10/10; 15/30] START max_depth=3, n_estimators=50…
[CV 2/10; 15/30] END max_depth=3, n_estimators=50;, score=(train=0.916,
test=0.880) total time=   0.1s
[CV 4/10; 12/30] END max_depth=2, n_estimators=200;, score=(train=0.941,
test=0.939) total time=   0.3s
[CV 4/10; 15/30] END max_depth=3, n_estimators=50;, score=(train=0.980,
test=0.980) total time=   0.1s
[CV 10/10; 12/30] END max_depth=2, n_estimators=200;, score=(train=0.892,
test=0.906) total time=   0.3s
[CV 6/10; 15/30] END max_depth=3, n_estimators=50;, score=(train=0.932,
test=0.935) total time=   0.1s
[CV 2/10; 12/30] END max_depth=2, n_estimators=200;, score=(train=0.891,
test=0.862) total time=   0.3s
[CV 1/10; 16/30] START max_depth=3, n_estimators=100…
[CV 8/10; 15/30] END max_depth=3, n_estimators=50;, score=(train=0.963,
test=0.982) total time=   0.1s
[CV 10/10; 15/30] END max_depth=3, n_estimators=50;, score=(train=0.967,
test=0.963) total time=   0.1s
[CV 7/10; 16/30] START max_depth=3, n_estimators=100…
[CV 3/10; 16/30] START max_depth=3,
n_estimators=100…[CV 1/10; 17/30] START max_depth=3,
n_estimators=150…

[CV 9/10; 16/30] START max_depth=3, n_estimators=100…
[CV 5/10; 16/30] START max_depth=3, n_estimators=100…
[CV 6/10; 12/30] END max_depth=2, n_estimators=200;, score=(train=0.927,
test=0.947) total time=   0.3s
[CV 5/10; 17/30] START max_depth=3, n_estimators=150…
[CV 3/10; 17/30] START max_depth=3, n_estimators=150…
[CV 8/10; 12/30] END max_depth=2, n_estimators=200;, score=(train=0.927,
test=0.944) total time=   0.3s
[CV 7/10; 17/30] START max_depth=3, n_estimators=150…
[CV 9/10; 17/30] START max_depth=3, n_estimators=150…
[CV 7/10; 16/30] END max_depth=3, n_estimators=100;, score=(train=0.981,
test=0.985) total time=   0.1s
[CV 8/10; 16/30] START max_depth=3, n_estimators=100…
[CV 9/10; 16/30] END max_depth=3, n_estimators=100;, score=(train=0.976,
test=0.963) total time=   0.1s
[CV 10/10; 16/30] START max_depth=3, n_estimators=100…
[CV 3/10; 16/30] END max_depth=3, n_estimators=100;, score=(train=0.981,
test=0.983) total time=   0.1s
[CV 4/10; 16/30] START max_depth=3, n_estimators=100…
[CV 1/10; 16/30] END max_depth=3, n_estimators=100;, score=(train=0.954,
test=0.955) total time=   0.2s
[CV 2/10; 16/30] START max_depth=3, n_estimators=100…
```

```
[CV 5/10; 16/30] END max_depth=3, n_estimators=100;, score=(train=0.980,
test=0.988) total time=   0.1s
[CV 6/10; 16/30] START max_depth=3, n_estimators=100…
[CV 1/10; 17/30] END max_depth=3, n_estimators=150;, score=(train=0.982,
test=0.974) total time=   0.2s
[CV 2/10; 17/30] START max_depth=3, n_estimators=150…
[CV 5/10; 17/30] END max_depth=3, n_estimators=150;, score=(train=0.980,
test=0.988) total time=   0.2s
[CV 6/10; 17/30] START max_depth=3, n_estimators=150…
[CV 7/10; 17/30] END max_depth=3, n_estimators=150;, score=(train=0.976,
test=0.985) total time=   0.2s
[CV 8/10; 17/30] START max_depth=3, n_estimators=150…
[CV 3/10; 17/30] END max_depth=3, n_estimators=150;, score=(train=0.981,
test=0.983) total time=   0.3s
[CV 4/10; 17/30] START max_depth=3, n_estimators=150…
[CV 10/10; 16/30] END max_depth=3, n_estimators=100;, score=(train=0.967,
test=0.966) total time=   0.2s
[CV 9/10; 17/30] END max_depth=3, n_estimators=150;, score=(train=0.981,
test=0.963) total time=   0.2s
[CV 10/10; 17/30] START max_depth=3, n_estimators=150…
[CV 4/10; 16/30] END max_depth=3, n_estimators=100;, score=(train=0.981,
test=0.983) total time=   0.2s
[CV 2/10; 16/30] END max_depth=3, n_estimators=100;, score=(train=0.969,
test=0.966) total time=   0.2s
[CV 6/10; 16/30] END max_depth=3, n_estimators=100;, score=(train=0.974,
test=0.991) total time=   0.2s
[CV 8/10; 16/30] END max_depth=3, n_estimators=100;, score=(train=0.972,
test=0.994) total time=   0.2s
[CV 3/10; 18/30] START max_depth=3, n_estimators=200…
[CV 1/10; 18/30] START max_depth=3, n_estimators=200…
[CV 5/10; 18/30] START max_depth=3, n_estimators=200…
[CV 7/10; 18/30] START max_depth=3, n_estimators=200…
[CV 9/10; 18/30] START max_depth=3, n_estimators=200…
[CV 2/10; 17/30] END max_depth=3, n_estimators=150;, score=(train=0.982,
test=0.974) total time=   0.2s
[CV 1/10; 19/30] START max_depth=4, n_estimators=10…
[CV 8/10; 17/30] END max_depth=3, n_estimators=150;, score=(train=0.978,
test=0.994) total time=   0.2s
[CV 3/10; 19/30] START max_depth=4, n_estimators=10…
[CV 1/10; 19/30] END max_depth=4, n_estimators=10;, score=(train=0.949,
test=0.947) total time=   0.0s
[CV 2/10; 19/30] START max_depth=4, n_estimators=10…
[CV 6/10; 17/30] END max_depth=3, n_estimators=150;, score=(train=0.959,
test=0.977) total time=   0.3s
[CV 4/10; 17/30] END max_depth=3, n_estimators=150;, score=(train=0.981,
test=0.983) total time=   0.2s
[CV 3/10; 19/30] END max_depth=4, n_estimators=10;, score=(train=0.965,
test=0.974) total time=   0.0s
```

```
[CV 4/10; 19/30] START max_depth=4, n_estimators=10…
[CV 5/10; 19/30] START max_depth=4, n_estimators=10…
[CV 2/10; 19/30] END max_depth=4, n_estimators=10;, score=(train=0.973,
test=0.955) total time=   0.0s
[CV 10/10; 17/30] END max_depth=3, n_estimators=150;, score=(train=0.974,
test=0.968) total time=   0.2s
[CV 7/10; 19/30] START max_depth=4, n_estimators=10…
[CV 1/10; 20/30] START max_depth=4,
n_estimators=25…[CV 9/10; 19/30] START max_depth=4,
n_estimators=10…

[CV 4/10; 19/30] END max_depth=4, n_estimators=10;, score=(train=0.979,
test=0.983) total time=   0.0s
[CV 5/10; 19/30] END max_depth=4, n_estimators=10;, score=(train=0.911,
test=0.933) total time=   0.0s
[CV 6/10; 19/30] START max_depth=4, n_estimators=10…
[CV 3/10; 20/30] START max_depth=4, n_estimators=25…
[CV 7/10; 19/30] END max_depth=4, n_estimators=10;, score=(train=0.894,
test=0.901) total time=   0.0s
[CV 8/10; 19/30] START max_depth=4, n_estimators=10…
[CV 9/10; 19/30] END max_depth=4, n_estimators=10;, score=(train=0.965,
test=0.946) total time=   0.0s
[CV 10/10; 19/30] START max_depth=4, n_estimators=10…
[CV 6/10; 19/30] END max_depth=4, n_estimators=10;, score=(train=0.950,
test=0.968) total time=   0.0s
[CV 8/10; 19/30] END max_depth=4, n_estimators=10;, score=(train=0.981,
test=0.994) total time=   0.0s
[CV 5/10; 20/30] START max_depth=4, n_estimators=25…
[CV 7/10; 20/30] START max_depth=4, n_estimators=25…
[CV 10/10; 19/30] END max_depth=4, n_estimators=10;, score=(train=0.951,
test=0.924) total time=   0.0s
[CV 9/10; 20/30] START max_depth=4, n_estimators=25…
[CV 1/10; 20/30] END max_depth=4, n_estimators=25;, score=(train=0.982,
test=0.974) total time=   0.0s
[CV 2/10; 20/30] START max_depth=4, n_estimators=25…
[CV 1/10; 18/30] END max_depth=3, n_estimators=200;, score=(train=0.969,
test=0.963) total time=   0.3s
[CV 2/10; 18/30] START max_depth=3, n_estimators=200…
[CV 5/10; 18/30] END max_depth=3, n_estimators=200;, score=(train=0.980,
test=0.988) total time=   0.3s
[CV 6/10; 18/30] START max_depth=3, n_estimators=200…
[CV 3/10; 20/30] END max_depth=4, n_estimators=25;, score=(train=0.981,
test=0.983) total time=   0.1s
[CV 4/10; 20/30] START max_depth=4, n_estimators=25…
[CV 5/10; 20/30] END max_depth=4, n_estimators=25;, score=(train=0.974,
test=0.983) total time=   0.0s
[CV 6/10; 20/30] START max_depth=4, n_estimators=25…
[CV 9/10; 20/30] END max_depth=4, n_estimators=25;, score=(train=0.974,
```

```
test=0.960) total time=   0.0s
[CV 10/10; 20/30] START max_depth=4, n_estimators=25…
[CV 9/10; 18/30] END max_depth=3, n_estimators=200;, score=(train=0.971,
test=0.941) total time=   0.3s
[CV 10/10; 18/30] START max_depth=3, n_estimators=200…
[CV 3/10; 18/30] END max_depth=3, n_estimators=200;, score=(train=0.981,
test=0.983) total time=   0.3s
[CV 2/10; 20/30] END max_depth=4, n_estimators=25;, score=(train=0.982,
test=0.974) total time=   0.0s
[CV 4/10; 18/30] START max_depth=3, n_estimators=200…
[CV 7/10; 20/30] END max_depth=4, n_estimators=25;, score=(train=0.973,
test=0.985) total time=   0.1s
[CV 8/10; 20/30] START max_depth=4, n_estimators=25…
[CV 1/10; 21/30] START max_depth=4, n_estimators=50…
[CV 7/10; 18/30] END max_depth=3, n_estimators=200;, score=(train=0.978,
test=0.985) total time=   0.3s
[CV 8/10; 18/30] START max_depth=3, n_estimators=200…
[CV 6/10; 20/30] END max_depth=4, n_estimators=25;, score=(train=0.978,
test=0.988) total time=   0.0s
[CV 4/10; 20/30] END max_depth=4, n_estimators=25;, score=(train=0.981,
test=0.983) total time=   0.1s
[CV 3/10; 21/30] START max_depth=4, n_estimators=50…
[CV 5/10; 21/30] START max_depth=4, n_estimators=50…
[CV 8/10; 20/30] END max_depth=4, n_estimators=25;, score=(train=0.977,
test=0.994) total time=   0.0s
[CV 10/10; 20/30] END max_depth=4, n_estimators=25;, score=(train=0.977,
test=0.968) total time=   0.1s
[CV 7/10; 21/30] START max_depth=4, n_estimators=50…
[CV 1/10; 21/30] END max_depth=4, n_estimators=50;, score=(train=0.982,
test=0.974) total time=   0.1s
[CV 9/10; 21/30] START max_depth=4, n_estimators=50…
[CV 2/10; 21/30] START max_depth=4, n_estimators=50…
[CV 3/10; 21/30] END max_depth=4, n_estimators=50;, score=(train=0.981,
test=0.983) total time=   0.1s
[CV 4/10; 21/30] START max_depth=4, n_estimators=50…
[CV 5/10; 21/30] END max_depth=4, n_estimators=50;, score=(train=0.980,
test=0.988) total time=   0.1s
[CV 6/10; 21/30] START max_depth=4, n_estimators=50…
[CV 7/10; 21/30] END max_depth=4, n_estimators=50;, score=(train=0.981,
test=0.985) total time=   0.1s
[CV 8/10; 21/30] START max_depth=4, n_estimators=50…
[CV 9/10; 21/30] END max_depth=4, n_estimators=50;, score=(train=0.982,
test=0.971) total time=   0.1s
[CV 10/10; 21/30] START max_depth=4, n_estimators=50…
[CV 2/10; 21/30] END max_depth=4, n_estimators=50;, score=(train=0.967,
test=0.952) total time=   0.1s
[CV 4/10; 21/30] END max_depth=4, n_estimators=50;, score=(train=0.981,
test=0.983) total time=   0.1s
```

```
[CV 6/10; 21/30] END max_depth=4, n_estimators=50;, score=(train=0.980,
test=0.991) total time=   0.1s
[CV 1/10; 22/30] START max_depth=4,
n_estimators=100…[CV 3/10; 22/30] START max_depth=4,
n_estimators=100…

[CV 8/10; 21/30] END max_depth=4, n_estimators=50;, score=(train=0.980,
test=0.994) total time=   0.1s
[CV 5/10; 22/30] START max_depth=4, n_estimators=100…
[CV 10/10; 21/30] END max_depth=4, n_estimators=50;, score=(train=0.982,
test=0.968) total time=   0.1s
[CV 7/10; 22/30] START max_depth=4, n_estimators=100…
[CV 10/10; 18/30] END max_depth=3, n_estimators=200;, score=(train=0.980,
test=0.968) total time=   0.3s
[CV 6/10; 18/30] END max_depth=3, n_estimators=200;, score=(train=0.980,
test=0.991) total time=   0.3s
[CV 2/10; 18/30] END max_depth=3, n_estimators=200;, score=(train=0.982,
test=0.974) total time=   0.3s
[CV 8/10; 18/30] END max_depth=3, n_estimators=200;, score=(train=0.980,
test=0.994) total time=   0.3s
[CV 3/10; 23/30] START max_depth=4, n_estimators=150…
[CV 1/10; 23/30] START max_depth=4, n_estimators=150…
[CV 9/10; 22/30] START max_depth=4, n_estimators=100…
[CV 7/10; 23/30] START max_depth=4, n_estimators=150…
[CV 5/10; 23/30] START max_depth=4, n_estimators=150…
[CV 4/10; 18/30] END max_depth=3, n_estimators=200;, score=(train=0.980,
test=0.983) total time=   0.3s
[CV 9/10; 23/30] START max_depth=4, n_estimators=150…
[CV 1/10; 22/30] END max_depth=4, n_estimators=100;, score=(train=0.982,
test=0.974) total time=   0.2s
[CV 2/10; 22/30] START max_depth=4, n_estimators=100…
[CV 3/10; 22/30] END max_depth=4, n_estimators=100;, score=(train=0.981,
test=0.983) total time=   0.2s
[CV 4/10; 22/30] START max_depth=4, n_estimators=100…
[CV 5/10; 22/30] END max_depth=4, n_estimators=100;, score=(train=0.980,
test=0.988) total time=   0.2s
[CV 6/10; 22/30] START max_depth=4, n_estimators=100…
[CV 7/10; 22/30] END max_depth=4, n_estimators=100;, score=(train=0.981,
test=0.985) total time=   0.2s
[CV 8/10; 22/30] START max_depth=4, n_estimators=100…
[CV 9/10; 22/30] END max_depth=4, n_estimators=100;, score=(train=0.982,
test=0.971) total time=   0.2s
[CV 10/10; 22/30] START max_depth=4, n_estimators=100…
[CV 2/10; 22/30] END max_depth=4, n_estimators=100;, score=(train=0.982,
test=0.974) total time=   0.2s
[CV 5/10; 23/30] END max_depth=4, n_estimators=150;, score=(train=0.980,
test=0.988) total time=   0.2s
[CV 6/10; 23/30] START max_depth=4, n_estimators=150…
```

```
[CV 1/10; 24/30] START max_depth=4, n_estimators=200…
[CV 3/10; 23/30] END max_depth=4, n_estimators=150;, score=(train=0.981,
test=0.983) total time=   0.3s
[CV 4/10; 23/30] START max_depth=4, n_estimators=150…
[CV 1/10; 23/30] END max_depth=4, n_estimators=150;, score=(train=0.982,
test=0.974) total time=   0.3s
[CV 2/10; 23/30] START max_depth=4, n_estimators=150…
[CV 4/10; 22/30] END max_depth=4, n_estimators=100;, score=(train=0.981,
test=0.983) total time=   0.2s
[CV 9/10; 23/30] END max_depth=4, n_estimators=150;, score=(train=0.982,
test=0.971) total time=   0.2s
[CV 10/10; 23/30] START max_depth=4, n_estimators=150…
[CV 6/10; 22/30] END max_depth=4, n_estimators=100;, score=(train=0.980,
test=0.991) total time=   0.2s
[CV 3/10; 24/30] START max_depth=4, n_estimators=200…
[CV 5/10; 24/30] START max_depth=4, n_estimators=200…
[CV 7/10; 23/30] END max_depth=4, n_estimators=150;, score=(train=0.981,
test=0.985) total time=   0.3s
[CV 8/10; 23/30] START max_depth=4, n_estimators=150…
[CV 8/10; 22/30] END max_depth=4, n_estimators=100;, score=(train=0.980,
test=0.994) total time=   0.2s
[CV 7/10; 24/30] START max_depth=4, n_estimators=200…
[CV 10/10; 22/30] END max_depth=4, n_estimators=100;, score=(train=0.982,
test=0.968) total time=   0.2s
[CV 9/10; 24/30] START max_depth=4, n_estimators=200…
[CV 2/10; 23/30] END max_depth=4, n_estimators=150;, score=(train=0.982,
test=0.974) total time=   0.3s
[CV 4/10; 23/30] END max_depth=4, n_estimators=150;, score=(train=0.981,
test=0.983) total time=   0.3s
[CV 6/10; 23/30] END max_depth=4, n_estimators=150;, score=(train=0.980,
test=0.991) total time=   0.3s
[CV 10/10; 23/30] END max_depth=4, n_estimators=150;, score=(train=0.982,
test=0.968) total time=   0.3s
[CV 8/10; 23/30] END max_depth=4, n_estimators=150;, score=(train=0.980,
test=0.994) total time=   0.2s
[CV 5/10; 25/30] START max_depth=5, n_estimators=10…
[CV 1/10; 25/30] START max_depth=5, n_estimators=10…
[CV 3/10; 25/30] START max_depth=5, n_estimators=10…
[CV 7/10; 25/30] START max_depth=5, n_estimators=10…
[CV 9/10; 25/30] START max_depth=5, n_estimators=10…
[CV 1/10; 25/30] END max_depth=5, n_estimators=10;, score=(train=0.987,
test=0.983) total time=   0.0s
[CV 2/10; 25/30] START max_depth=5, n_estimators=10…
[CV 3/10; 25/30] END max_depth=5, n_estimators=10;, score=(train=0.993,
test=0.994) total time=   0.0s
[CV 4/10; 25/30] START max_depth=5, n_estimators=10…
[CV 5/10; 25/30] END max_depth=5, n_estimators=10;, score=(train=0.986,
test=0.988) total time=   0.0s
```

```
[CV 6/10; 25/30] START max_depth=5, n_estimators=10…
[CV 7/10; 25/30] END max_depth=5, n_estimators=10;, score=(train=0.983,
test=0.985) total time=   0.0s
[CV 8/10; 25/30] START max_depth=5, n_estimators=10…
[CV 2/10; 25/30] END max_depth=5, n_estimators=10;, score=(train=0.982,
test=0.971) total time=   0.0s
[CV 1/10; 26/30] START max_depth=5, n_estimators=25…
[CV 4/10; 25/30] END max_depth=5, n_estimators=10;, score=(train=0.983,
test=0.983) total time=   0.0s
[CV 6/10; 25/30] END max_depth=5, n_estimators=10;, score=(train=0.990,
test=0.994) total time=   0.0s
[CV 9/10; 25/30] END max_depth=5, n_estimators=10;, score=(train=0.990,
test=0.985) total time=   0.0s
[CV 10/10; 25/30] START max_depth=5, n_estimators=10…
[CV 3/10; 26/30] START max_depth=5,
n_estimators=25…[CV 1/10; 24/30] END max_depth=4,
n_estimators=200;, score=(train=0.982, test=0.974) total time=   0.4s
[CV 2/10; 24/30] START max_depth=4, n_estimators=200…

[CV 8/10; 25/30] END max_depth=5, n_estimators=10;, score=(train=0.980,
test=0.994) total time=   0.0s
[CV 3/10; 24/30] END max_depth=4, n_estimators=200;, score=(train=0.981,
test=0.983) total time=   0.3s
[CV 4/10; 24/30] START max_depth=4, n_estimators=200…
[CV 10/10; 25/30] END max_depth=5, n_estimators=10;, score=(train=0.995,
test=0.994) total time=   0.0s
[CV 5/10; 26/30] START max_depth=5, n_estimators=25…
[CV 1/10; 26/30] END max_depth=5, n_estimators=25;, score=(train=0.982,
test=0.974) total time=   0.0s
[CV 2/10; 26/30] START max_depth=5, n_estimators=25…
[CV 5/10; 24/30] END max_depth=4, n_estimators=200;, score=(train=0.980,
test=0.988) total time=   0.4s
[CV 6/10; 24/30] START max_depth=4, n_estimators=200…
[CV 3/10; 26/30] END max_depth=5, n_estimators=25;, score=(train=0.986,
test=0.985) total time=   0.0s
[CV 4/10; 26/30] START max_depth=5, n_estimators=25…
[CV 9/10; 26/30] START max_depth=5, n_estimators=25…
[CV 7/10; 26/30] START max_depth=5, n_estimators=25…
[CV 7/10; 24/30] END max_depth=4, n_estimators=200;, score=(train=0.981,
test=0.985) total time=   0.4s
[CV 8/10; 24/30] START max_depth=4, n_estimators=200…
[CV 5/10; 26/30] END max_depth=5, n_estimators=25;, score=(train=0.983,
test=0.988) total time=   0.0s
[CV 6/10; 26/30] START max_depth=5, n_estimators=25…
[CV 2/10; 26/30] END max_depth=5, n_estimators=25;, score=(train=0.991,
test=0.983) total time=   0.0s
[CV 9/10; 26/30] END max_depth=5, n_estimators=25;, score=(train=0.983,
test=0.974) total time=   0.0s
```

```
[CV 10/10; 26/30] START max_depth=5, n_estimators=25…
[CV 4/10; 26/30] END max_depth=5, n_estimators=25;, score=(train=0.996,
test=0.994) total time=   0.0s
[CV 7/10; 26/30] END max_depth=5, n_estimators=25;, score=(train=0.987,
test=0.991) total time=   0.1s
[CV 8/10; 26/30] START max_depth=5, n_estimators=25…
[CV 9/10; 24/30] END max_depth=4, n_estimators=200;, score=(train=0.982,
test=0.971) total time=   0.3s
[CV 10/10; 24/30] START max_depth=4, n_estimators=200…
[CV 1/10; 27/30] START max_depth=5, n_estimators=50…
[CV 3/10; 27/30] START max_depth=5, n_estimators=50…
[CV 6/10; 26/30] END max_depth=5, n_estimators=25;, score=(train=0.983,
test=0.994) total time=   0.0s
[CV 5/10; 27/30] START max_depth=5, n_estimators=50…
[CV 10/10; 26/30] END max_depth=5, n_estimators=25;, score=(train=0.986,
test=0.977) total time=   0.0s
[CV 8/10; 26/30] END max_depth=5, n_estimators=25;, score=(train=0.980,
test=0.994) total time=   0.0s
[CV 7/10; 27/30] START max_depth=5, n_estimators=50…
[CV 9/10; 27/30] START max_depth=5, n_estimators=50…
[CV 1/10; 27/30] END max_depth=5, n_estimators=50;, score=(train=0.982,
test=0.974) total time=   0.1s
[CV 2/10; 27/30] START max_depth=5, n_estimators=50…
[CV 3/10; 27/30] END max_depth=5, n_estimators=50;, score=(train=0.981,
test=0.983) total time=   0.1s
[CV 4/10; 27/30] START max_depth=5, n_estimators=50…
[CV 5/10; 27/30] END max_depth=5, n_estimators=50;, score=(train=0.982,
test=0.988) total time=   0.1s
[CV 6/10; 27/30] START max_depth=5, n_estimators=50…
[CV 7/10; 27/30] END max_depth=5, n_estimators=50;, score=(train=0.981,
test=0.985) total time=   0.1s
[CV 8/10; 27/30] START max_depth=5, n_estimators=50…
[CV 9/10; 27/30] END max_depth=5, n_estimators=50;, score=(train=0.982,
test=0.971) total time=   0.1s
[CV 10/10; 27/30] START max_depth=5, n_estimators=50…
[CV 2/10; 27/30] END max_depth=5, n_estimators=50;, score=(train=0.982,
test=0.980) total time=   0.1s
[CV 4/10; 27/30] END max_depth=5, n_estimators=50;, score=(train=0.981,
test=0.983) total time=   0.1s
[CV 1/10; 28/30] START max_depth=5, n_estimators=100…
[CV 3/10; 28/30] START max_depth=5, n_estimators=100…
[CV 4/10; 24/30] END max_depth=4, n_estimators=200;, score=(train=0.981,
test=0.983) total time=   0.3s
[CV 6/10; 27/30] END max_depth=5, n_estimators=50;, score=(train=0.980,
test=0.991) total time=   0.1s
[CV 2/10; 24/30] END max_depth=4, n_estimators=200;, score=(train=0.982,
test=0.974) total time=   0.4s
[CV 8/10; 27/30] END max_depth=5, n_estimators=50;, score=(train=0.981,
```

```
test=0.994) total time=   0.1s
[CV 9/10; 28/30] START max_depth=5, n_estimators=100…
[CV 5/10; 28/30] START max_depth=5, n_estimators=100…
[CV 7/10; 28/30] START max_depth=5, n_estimators=100…
[CV 6/10; 24/30] END max_depth=4, n_estimators=200;, score=(train=0.980,
test=0.991) total time=   0.3s
[CV 1/10; 29/30] START max_depth=5, n_estimators=150…
[CV 3/10; 29/30] START max_depth=5, n_estimators=150…
[CV 10/10; 27/30] END max_depth=5, n_estimators=50;, score=(train=0.982,
test=0.968) total time=   0.1s
[CV 5/10; 29/30] START max_depth=5, n_estimators=150…
[CV 10/10; 24/30] END max_depth=4, n_estimators=200;, score=(train=0.982,
test=0.968) total time=   0.3s
[CV 7/10; 29/30] START max_depth=5, n_estimators=150…
[CV 8/10; 24/30] END max_depth=4, n_estimators=200;, score=(train=0.980,
test=0.994) total time=   0.4s
[CV 9/10; 29/30] START max_depth=5, n_estimators=150…
[CV 1/10; 28/30] END max_depth=5, n_estimators=100;, score=(train=0.982,
test=0.974) total time=   0.2s
[CV 2/10; 28/30] START max_depth=5, n_estimators=100…
[CV 3/10; 28/30] END max_depth=5, n_estimators=100;, score=(train=0.981,
test=0.983) total time=   0.2s
[CV 4/10; 28/30] START max_depth=5, n_estimators=100…
[CV 9/10; 28/30] END max_depth=5, n_estimators=100;, score=(train=0.982,
test=0.971) total time=   0.2s
[CV 10/10; 28/30] START max_depth=5, n_estimators=100…
[CV 7/10; 28/30] END max_depth=5, n_estimators=100;, score=(train=0.981,
test=0.985) total time=   0.2s
[CV 8/10; 28/30] START max_depth=5, n_estimators=100…
[CV 5/10; 28/30] END max_depth=5, n_estimators=100;, score=(train=0.981,
test=0.988) total time=   0.2s
[CV 6/10; 28/30] START max_depth=5, n_estimators=100…
[CV 2/10; 28/30] END max_depth=5, n_estimators=100;, score=(train=0.982,
test=0.974) total time=   0.2s
[CV 1/10; 30/30] START max_depth=5, n_estimators=200…
[CV 3/10; 29/30] END max_depth=5, n_estimators=150;, score=(train=0.983,
test=0.983) total time=   0.3s
[CV 4/10; 28/30] END max_depth=5, n_estimators=100;, score=(train=0.981,
test=0.983) total time=   0.2s
[CV 4/10; 29/30] START max_depth=5, n_estimators=150…
[CV 7/10; 29/30] END max_depth=5, n_estimators=150;, score=(train=0.981,
test=0.985) total time=   0.3s
[CV 8/10; 29/30] START max_depth=5, n_estimators=150…
[CV 1/10; 29/30] END max_depth=5, n_estimators=150;, score=(train=0.982,
test=0.974) total time=   0.3s
[CV 2/10; 29/30] START max_depth=5, n_estimators=150…
[CV 3/10; 30/30] START max_depth=5, n_estimators=200…
[CV 6/10; 28/30] END max_depth=5, n_estimators=100;, score=(train=0.980,
```

```
test=0.991) total time=   0.1s
[CV 9/10; 29/30] END max_depth=5, n_estimators=150;, score=(train=0.982,
test=0.974) total time=   0.3s
[CV 10/10; 29/30] START max_depth=5, n_estimators=150…
[CV 5/10; 29/30] END max_depth=5, n_estimators=150;, score=(train=0.980,
test=0.988) total time=   0.3s
[CV 6/10; 29/30] START max_depth=5, n_estimators=150…
[CV 8/10; 28/30] END max_depth=5, n_estimators=100;, score=(train=0.980,
test=0.994) total time=   0.2s
[CV 5/10; 30/30] START max_depth=5, n_estimators=200…
[CV 10/10; 28/30] END max_depth=5, n_estimators=100;, score=(train=0.982,
test=0.968) total time=   0.2s
[CV 7/10; 30/30] START max_depth=5, n_estimators=200…
[CV 9/10; 30/30] START max_depth=5, n_estimators=200…
[CV 8/10; 29/30] END max_depth=5, n_estimators=150;, score=(train=0.981,
test=0.994) total time=   0.2s
[CV 4/10; 29/30] END max_depth=5, n_estimators=150;, score=(train=0.981,
test=0.983) total time=   0.3s
[CV 2/10; 29/30] END max_depth=5, n_estimators=150;, score=(train=0.983,
test=0.974) total time=   0.3s
[CV 1/10; 30/30] END max_depth=5, n_estimators=200;, score=(train=0.982,
test=0.974) total time=   0.3s
[CV 2/10; 30/30] START max_depth=5, n_estimators=200…
[CV 6/10; 29/30] END max_depth=5, n_estimators=150;, score=(train=0.980,
test=0.991) total time=   0.3s
[CV 10/10; 29/30] END max_depth=5, n_estimators=150;, score=(train=0.983,
test=0.971) total time=   0.3s
[CV 3/10; 30/30] END max_depth=5, n_estimators=200;, score=(train=0.982,
test=0.983) total time=   0.4s
[CV 4/10; 30/30] START max_depth=5, n_estimators=200…
[CV 7/10; 30/30] END max_depth=5, n_estimators=200;, score=(train=0.981,
test=0.985) total time=   0.3s
[CV 8/10; 30/30] START max_depth=5, n_estimators=200…
[CV 5/10; 30/30] END max_depth=5, n_estimators=200;, score=(train=0.981,
test=0.988) total time=   0.4s
[CV 6/10; 30/30] START max_depth=5, n_estimators=200…
[CV 9/10; 30/30] END max_depth=5, n_estimators=200;, score=(train=0.982,
test=0.971) total time=   0.3s
[CV 10/10; 30/30] START max_depth=5, n_estimators=200…
[CV 2/10; 30/30] END max_depth=5, n_estimators=200;, score=(train=0.982,
test=0.974) total time=   0.3s
[CV 4/10; 30/30] END max_depth=5, n_estimators=200;, score=(train=0.982,
test=0.983) total time=   0.3s
[CV 8/10; 30/30] END max_depth=5, n_estimators=200;, score=(train=0.980,
test=0.994) total time=   0.3s
[CV 6/10; 30/30] END max_depth=5, n_estimators=200;, score=(train=0.980,
test=0.991) total time=   0.3s
[CV 10/10; 30/30] END max_depth=5, n_estimators=200;, score=(train=0.982,
```

```
test=0.968) total time=   0.3s
```

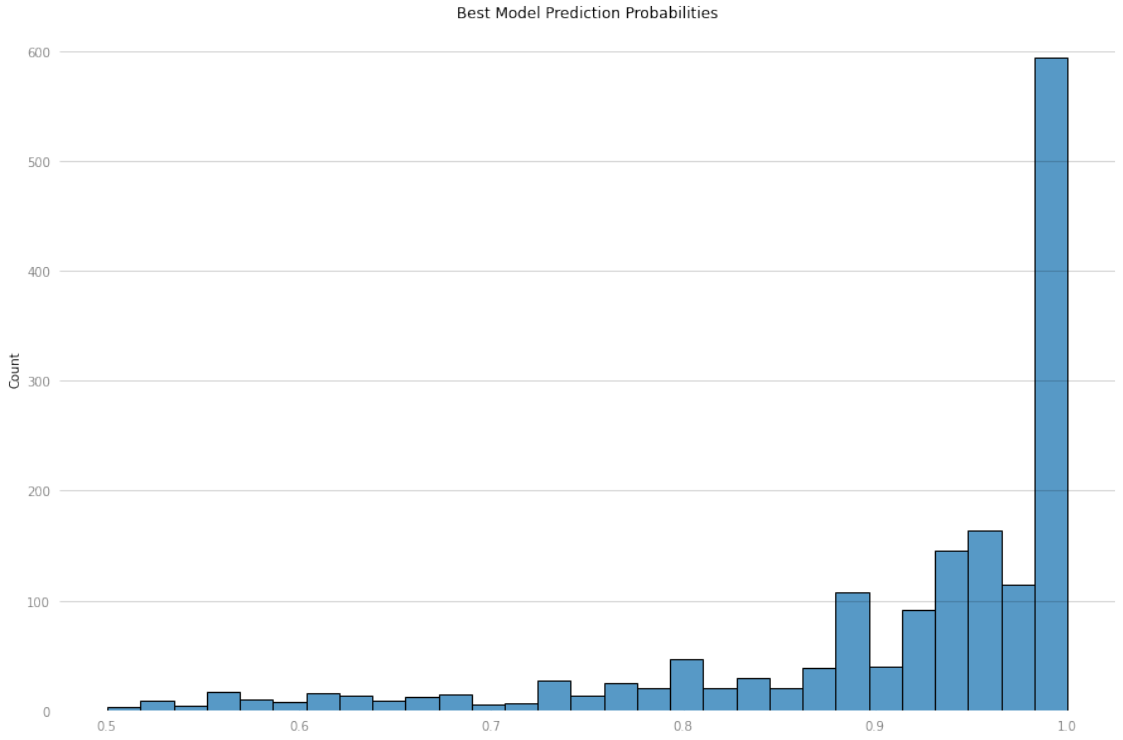
[61]:
```
RandomizedSearchCV(cv=10, estimator=RandomForestClassifier(), n_iter=100,
                   n_jobs=-1,
                   param_distributions={'max_depth': [1, 2, 3, 4, 5],
                                        'n_estimators': [10, 25, 50, 100, 150,
                                                         200]},
                   return_train_score=True, scoring='precision', verbose=10)
```

[ ]:

[62]:
```
best_model = search.best_estimator_

best_model
```

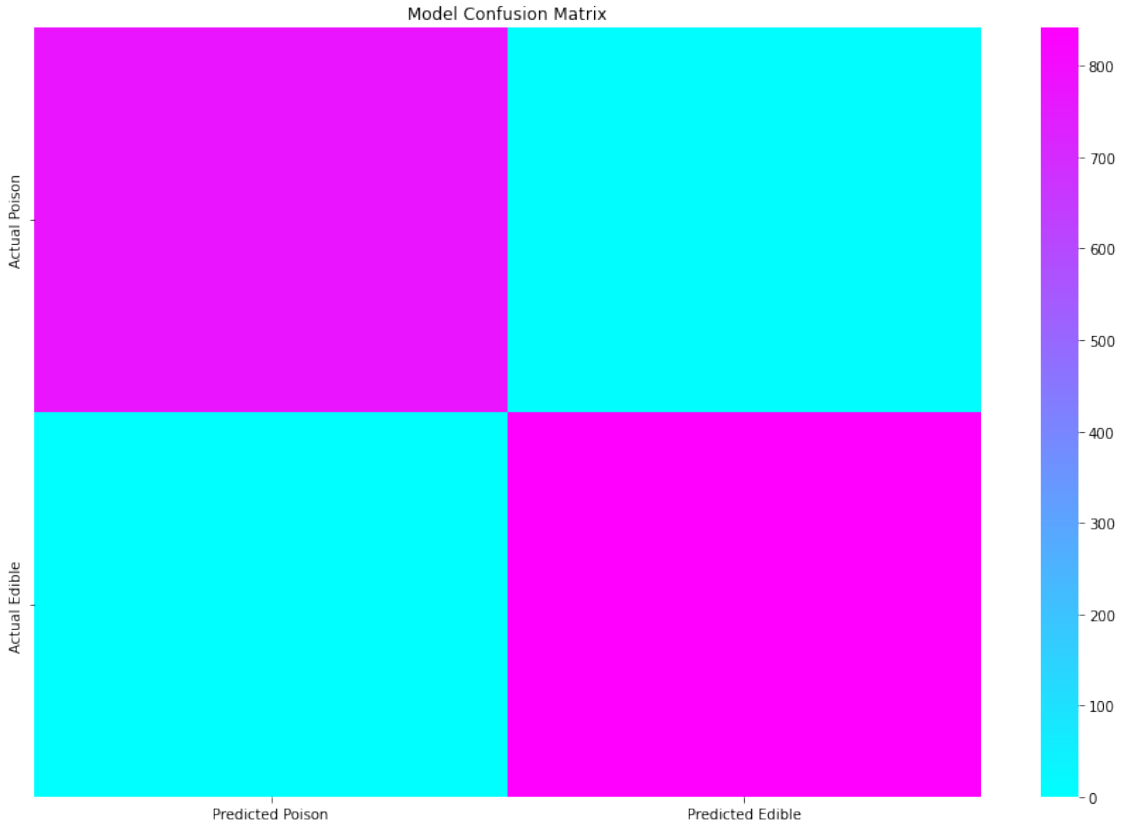
[62]:
```
RandomForestClassifier(max_depth=5, n_estimators=10)
```

[63]:
```
model_analysis(best_model, X_test, y_test)
```



	precision	recall	f1-score	support
0-Poisonous	1.00	0.99	0.99	783

```
              1-Edible       0.99      1.00      1.00       842

              accuracy                           1.00      1625
             macro avg       1.00      0.99      1.00      1625
          weighted avg       1.00      1.00      1.00      1625
```



[63]:

	Predicted Poison	Predicted Edible
Actual Poison	775	8
Actual Edible	0	842

[ ]: