Case Study 4: Bankruptcy Prediction

Matt Farrow

June 17, 2022

1 Introduction

In this case study, my goal is to predict whether a company may declare bankruptcy using data collecting from Polish companies between 2000-2012 for companies that went bankrupt, and 2007 to 2013 for those that did not.

The objective of this case study is to use Random Forest and XGBoost to accurately predict bankruptcy so that the company has an opportunity to potentiall divest their investments and save, or at least now lose as much, money.

2 Methods

2.1 Data Examination

The initial data set contained in five files, each of which spans a year of information.

	Attr1	Attr2	Attr3	Attr4	Attr5
count	43397.00000	43397.00000	43397.00000	43271.00000	43316.00000
mean	0.03516	0.59021	0.11443	6.31470	-385.34660
std	2.99411	5.84275	5.43943	295.43443	61243.02587
min	-463.89000	-430.87000	-479.96000	-0.40311	-11903000.00000
25%	0.00343	0.26898	0.02152	1.04950	-49.08000
50%	0.04966	0.47190	0.19661	1.56980	-1.03450
75%	0.12958	0.68832	0.40339	2.78745	50.63425
max	94.28000	480.96000	28.33600	53433.00000	1250100.00000

Figure 1: Sample of Original Data

I next looked at the response variable, class, to see what the distribution of bankruptcies to non-bankruptcies was.

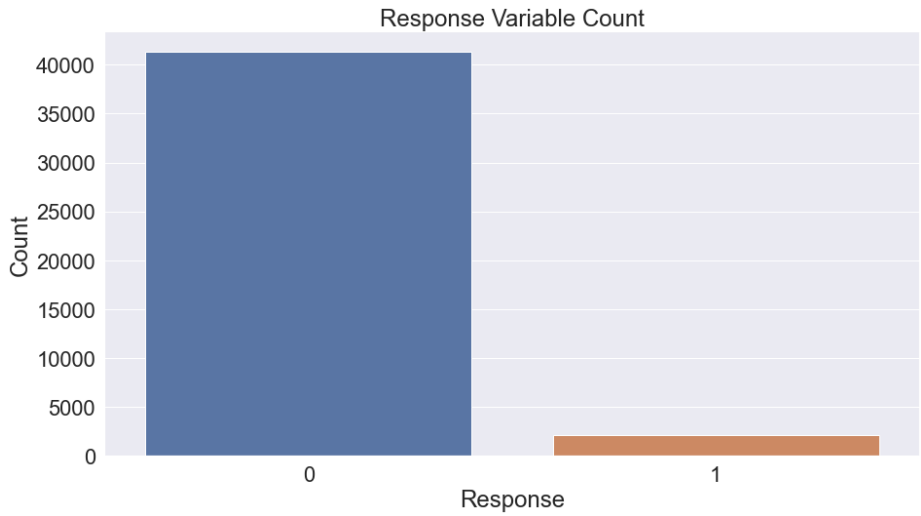


Figure 2: Distribution of Response Variable

Unfortunately, on intial observation the data appears to be significantly imbalanced. That said, the imbalanced nature of the data confirms that our company is making far more investments with companies who do not go bankrupt than with those who do. However, this observation only applies to the number of investments, not the size of them. I decided that I was to address the scaling I would do so in the model building pipeline.

After examining the response variable, I next looked at how many missing values were in the data and noted attributes varied between no missing values and almost 19,000. I made the decision to drop the attributes that contained more than 1,000 missing values; the rest would be imputed during model building.

Finally, before proceeding to model building, I checked attribute correlation using a number of different thresholds for isolating which attributes were most closely correlated. Starting at a level of 0.8 and progressing to 0.9, 0.95, and 0.98, I made the decision to drop only the attributes that had a 98% or great correlation (Figure 3: **Final Correlation Heatmap After Removing Highly-Correlated Variables**).

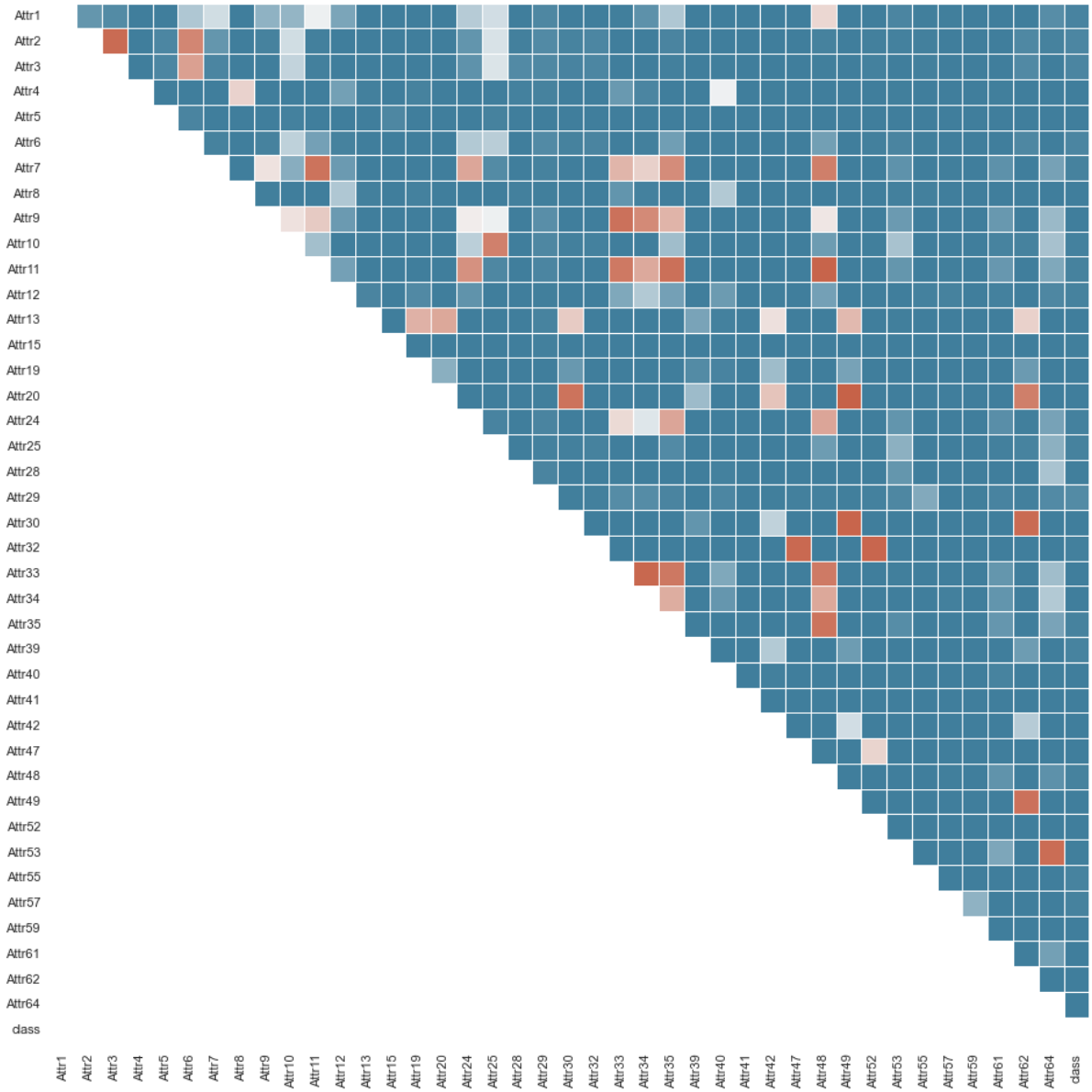


Figure 3: Final Correlation Heatmap After Removing Highly-Correlated Variables

2.2 Model Preparation & Execution

Two approaches were taken to model building. In both, the original data was first split into test and training sets with missing values imputed using the attribute's mean value. In addition, the data was scaled ahead of model building in order to correct for the class imbalance that appears in the data. The RobustScaler was chosen because of its ability to handle outliers.

For both the Random Forest as well as the XGBoost models, a list of hyperparameters was defined along with cross-validation to be used in the grid search process of each model.

3 Results

3.1 Random Forest Results

The Random Forest model did perform quite admirably (Table 1: **Random Forest Performance Metrics**) with an accuracy score of 0.9455. In addition to the performance metrics, the ROC curve (Figure 4: **ROC Curve for Random Forest Model**) and confusion matrix (Figure 5: **Random Forest Confusion Matrix**) give additional clarity to the model's performance.

Metric	Value
Accuracy	0.9455
Recall	0.1531
Precision	0.3497

Table 1: Random Forest Performance Metrics

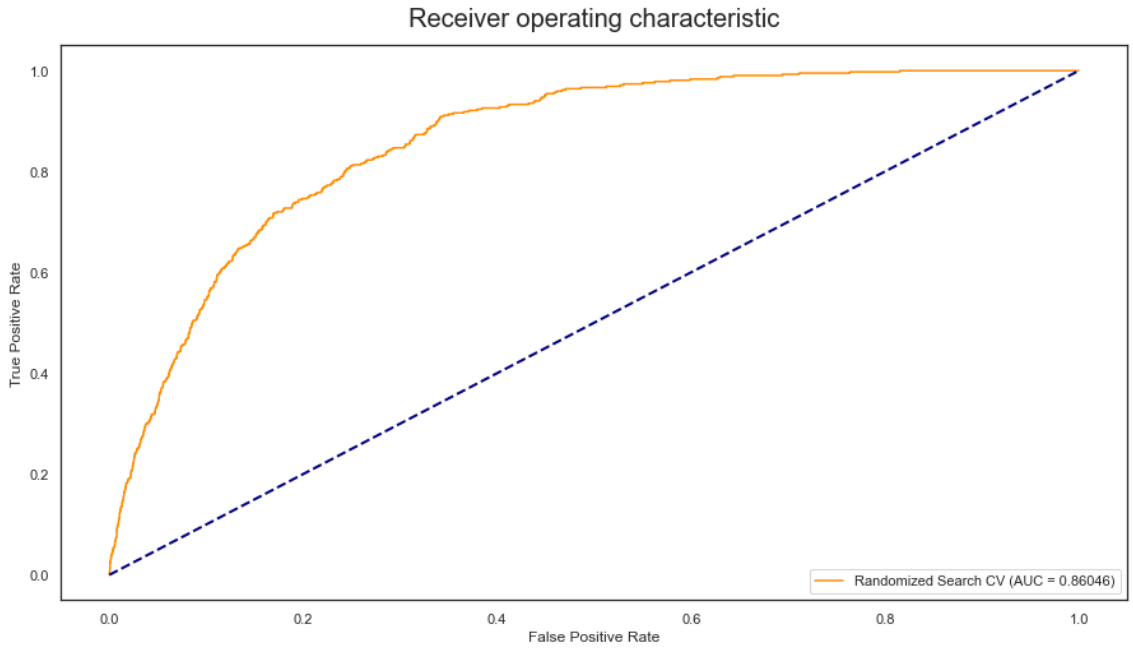


Figure 4: ROC Curve for Random Forest Model

In the confusion matrix, we can see that the model did misclassify over 300 companies as being ones that are likely to file for bankruptcy when in face they did not. That error rate would need additional evaluation to determine whether that is an acceptable risk to reward ratio for putting this model into production.

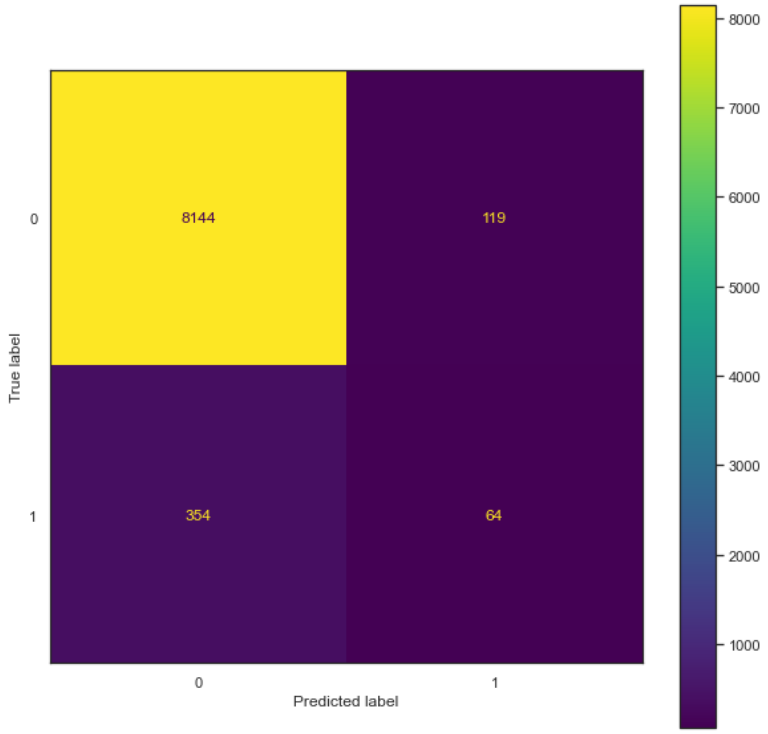


Figure 5: Random Forest Confusion Matrix

3.1 XGBoost Results

The XGBoost model was configured to run 1,000 rounds with an early stopping rounds value of five. As a result, the final trained model stopped at 632 trees (Figure 6: **Results of XGBoost Log Loss Error for Train & Test Data**). The best estimator occurred at 193 trees.

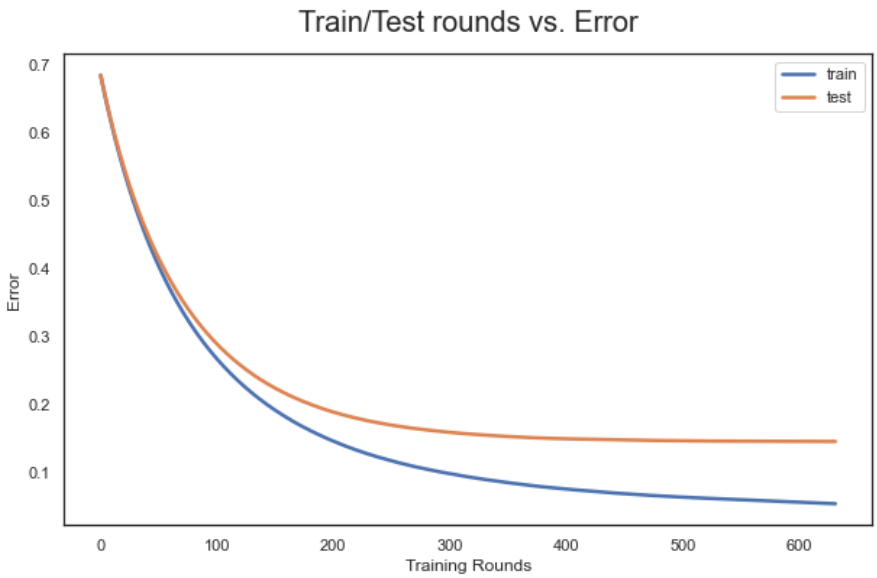


Figure 6: Results of XGBoost Log Loss Error for Train & Test Data

The model's overall accuracy was 0.94 which puts it on par wit the Random Forest model.

Metric	Value
Accuracy	0.94
Recall	0.11
Precision	0.51

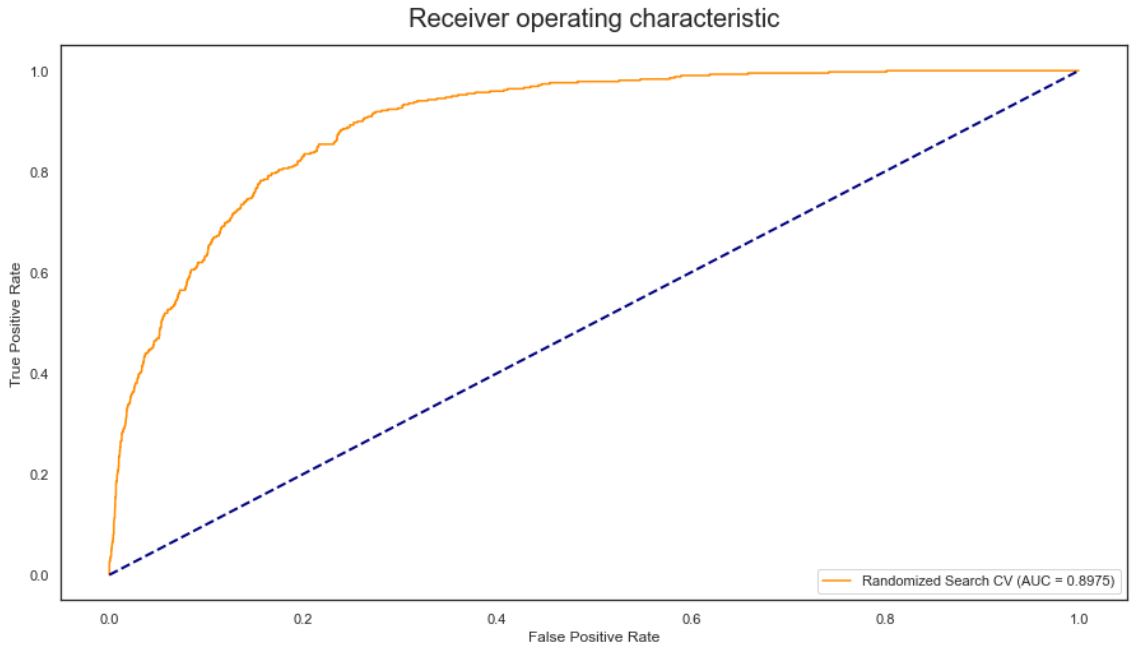


Figure 7: XGBoost ROC Curve

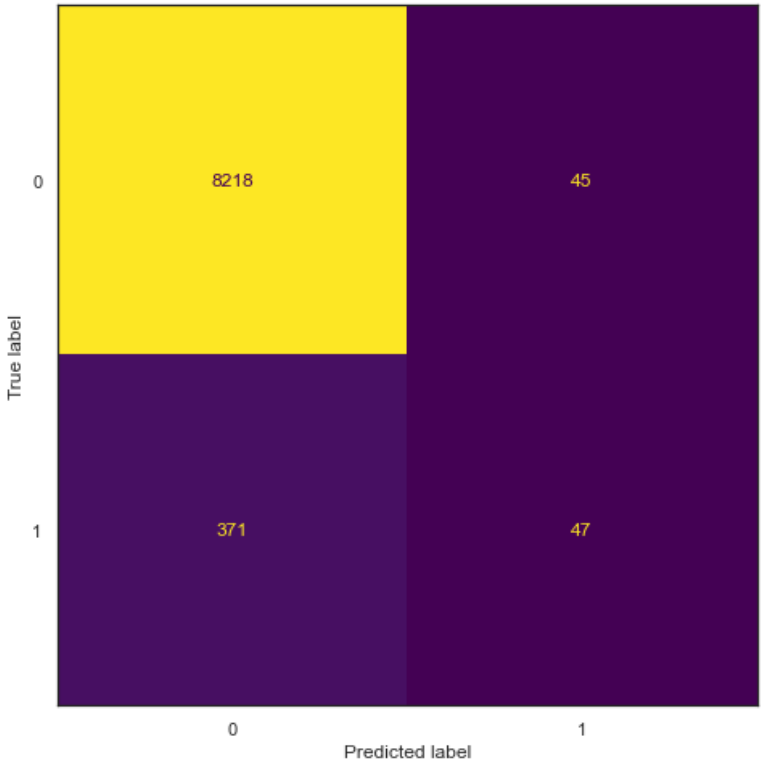


Figure 8: XGBoost Confusion Matrix

4 Conclusion

While both models provide strong accuracy results, the XGBoost model did provide the greatest results. This is due in no small part to the way that the algorithm learns from its mistakes to minimize loss. One challenge for XGBoost in this scenario is its more "black box" nature than the Random Forest model. Depending on the amounts of money invested in each of these companies, it is possible that management would feel more comfortable with a model that offered slightly more interpretability, especially if this is a new idea for them. Either way, both models offer strong performance as well as pros and cons for consideration.

Appendix

Code

Code begins on the following page.

Case Study 4

Description

Use Random Forest and XGBoost to accurately predict bankruptcy. Tune your models for maximum accuracy, but include precision and recall as summary metrics.

data

Email 1

From: Finance Department

To: Data Science Department

Subject: Financial Delinquency Project

We've collected our data. And we've noticed that you know what one of the biggest losses to our company is when companies go bankrupt and for our various investment strategies. So what we'd like to do is take a look at our historical data and see if there's any way that we can predict in the future that a company might go bankrupt, so that we can divest ourselves ahead of time.

Email 2

From: Finance Department

To: Data Science Department

Subject: RE: Financial Delinquency Project

Oh, and just to clarify, this dataset is collected over five years, but we don't care the exact year that a company will go bankrupt, just whether or not they will go bankrupt at all, based on the data. Thanks!

Setup

Load Libraries

In [1]:
```
# Import standard libraries
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

# Import sklearn libraries
from sklearn.ensemble import RandomForestClassifier
from sklearn.impute import SimpleImputer
from sklearn.metrics import auc
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.metrics import plot_confusion_matrix
from sklearn.metrics import roc_curve
from sklearn.metrics import accuracy_score
from sklearn.metrics import recall_score
from sklearn.metrics import precision_score
from sklearn.metrics import f1_score
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import RobustScaler

# Import other libraries
from scipy.io import arff
from statsmodels.stats.outliers_influence import variance_inflation_factor
import missingno
import os
import warnings
import xgboost
from xgboost import XGBClassifier

# Set options
pd.set_option("display.max_columns", None)
pd.options.display.max_rows = 10000
pd.options.display.max_columns = 10000

warnings.filterwarnings("ignore")
get_ipython().run_line_magic("matplotlib", "inline")
```

Load Data

In [2]:
```
# Iterate through files provided and make data frame
files = os.listdir('./data')
files.sort()
df = pd.DataFrame()
for i, filename in enumerate(files):
    file = "./data/"+filename
    data = arff.loadarff(file)
    df_temp = pd.DataFrame(data[0])
    df = pd.concat([df,df_temp])
```

In [3]:
```
df.head()
```

Out[3]:

	Attr1	Attr2	Attr3	Attr4	Attr5	Attr6	Attr7	Attr8	Attr9	Attr
0	0.200550	0.37951	0.39641	2.0472	32.3510	0.38825	0.249760	1.33050	1.1389	0.504
1	0.209120	0.49988	0.47225	1.9447	14.7860	0.00000	0.258340	0.99601	1.6996	0.497
2	0.248660	0.69592	0.26713	1.5548	-1.1523	0.00000	0.309060	0.43695	1.3090	0.304
3	0.081483	0.30734	0.45879	2.4928	51.9520	0.14988	0.092704	1.86610	1.0571	0.573
4	0.187320	0.61323	0.22960	1.4063	-7.3128	0.18732	0.187320	0.63070	1.1559	0.386

In [4]:
```
df.describe()
```

Out[4]:

	Attr1	Attr2	Attr3	Attr4	Attr5	A
count	43397.000000	43397.000000	43397.000000	43271.000000	4.331600e+04	43397.000
mean	0.035160	0.590212	0.114431	6.314702	-3.853466e+02	-0.05
std	2.994109	5.842748	5.439429	295.434425	6.124303e+04	7.20
min	-463.890000	-430.870000	-479.960000	-0.403110	-1.190300e+07	-508.41
25%	0.003429	0.268980	0.021521	1.049500	-4.908000e+01	0.000
50%	0.049660	0.471900	0.196610	1.569800	-1.034500e+00	0.000
75%	0.129580	0.688320	0.403390	2.787450	5.063425e+01	0.089
max	94.280000	480.960000	28.336000	53433.000000	1.250100e+06	543.250

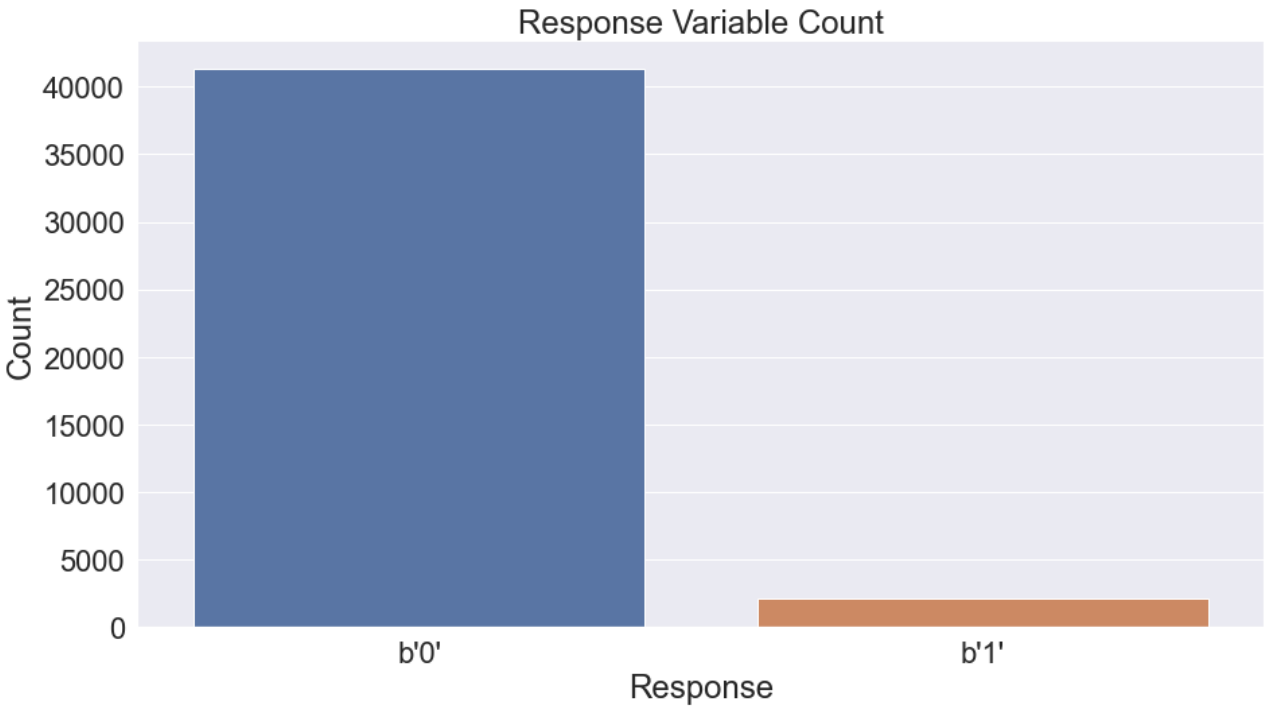
In [5]:
```
df.shape
```

Out[5]:
```
(43405, 65)
```

Examine Response Value

In [6]:
```
sns.set_style('darkgrid')
sns.set(rc = {'figure.figsize':(15,8)})
sns.set(font_scale = 2)
p = sns.countplot(x = 'class', data = df)
p.set_xlabel('Response')
p.set_ylabel('Count')
p.set_title('Response Variable Count')
```

Out[6]:
```
Text(0.5, 1.0, 'Response Variable Count')
```



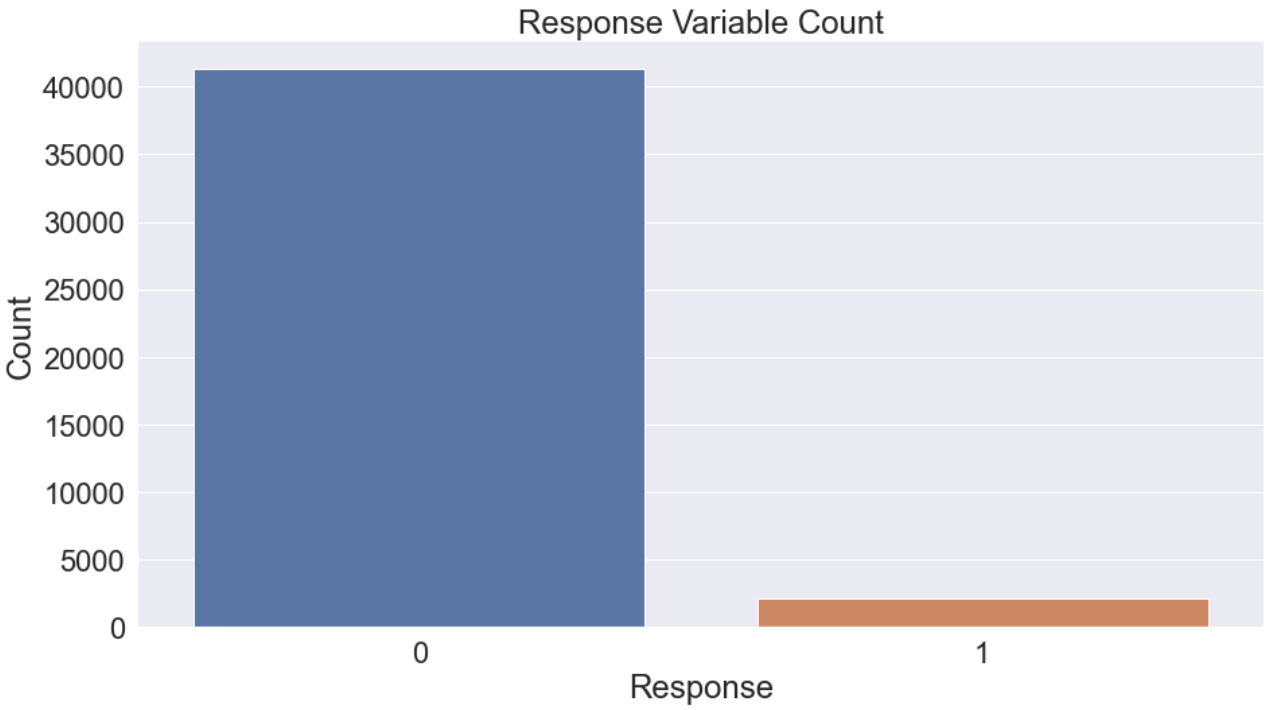
Rename each of the classes to simply be '0' and '1'.

In [7]:
```
# Clean up response
df['class'] = df['class'].replace([b'0', b'1'], [0, 1])
df['class'].value_counts()
```

Out[7]:
```
0    41314
1     2091
Name: class, dtype: int64
```

In [8]:
```
sns.set_style('darkgrid')
sns.set(rc = {'figure.figsize':(15,8)})
sns.set(font_scale = 2)
p = sns.countplot(x = 'class', data = df)
p.set_xlabel('Response')
p.set_ylabel('Count')
p.set_title('Response Variable Count')
```

Out[8]:
```
Text(0.5, 1.0, 'Response Variable Count')
```



Profile the Data

In [9]:
```
# from pandas_profiling import ProfileReport
# profile = ProfileReport(df, minimal = True)
# profile.to_file(output_file = "output.html")
```

Examine Missing Data

In [10]:
```
# missingno.matrix(df)
```

In [11]:
```
# missingno.matrix(df.loc[df['class']==1])
```

In [12]:
```
# missingno.matrix(df.loc[df['class']==0])
```

In [13]:
```
missing_cols = []
missing_vals = []

# Calculate the number of missing values
for column in df.columns:
    missing_cols.append(column)
    missing_vals.append(df[column].isnull().sum())

# Create a dataframe with the list of attributes and the count of missing va
missing = pd.DataFrame({'Attributes': missing_cols,
                        'Count': missing_vals})

# Define percentage
missing['Pct'] = (missing['Count'] / missing['Count'].sum()) * 100

# Sort 'Pct' from smallest to largest
missing = missing.sort_values('Pct', ascending = True)

# Format 'Pct' as a percentage
missing.style.format({
    'Pct': '{:,.2%}'.format
})
```

Out[13]:

	Attributes	Count	Pct
64	class	0	0.00%
54	Attr55	1	0.24%
58	Attr59	7	1.69%
56	Attr57	7	1.69%
50	Attr51	8	1.94%
37	Attr38	8	1.94%
35	Attr36	8	1.94%
34	Attr35	8	1.94%
28	Attr29	8	1.94%
24	Attr25	8	1.94%
21	Attr22	8	1.94%
17	Attr18	8	1.94%
13	Attr14	8	1.94%
0	Attr1	8	1.94%
5	Attr6	8	1.94%
6	Attr7	8	1.94%
2	Attr3	8	1.94%
1	Attr2	8	1.94%
9	Attr10	8	1.94%

47	Attr48	9	2.18%
8	Attr9	9	2.18%
14	Attr15	36	8.71%
10	Attr11	44	10.65%
57	Attr58	84	20.33%
4	Attr5	89	21.54%
7	Attr8	94	22.75%
49	Attr50	94	22.75%
33	Attr34	94	22.75%
16	Attr17	94	22.75%
25	Attr26	95	22.99%
15	Attr16	95	22.99%
60	Attr61	102	24.68%
43	Attr44	127	30.73%
42	Attr43	127	30.73%
41	Attr42	127	30.73%
12	Attr13	127	30.73%
38	Attr39	127	30.73%
22	Attr23	127	30.73%
55	Attr56	127	30.73%
48	Attr49	127	30.73%
29	Attr30	127	30.73%
19	Attr20	127	30.73%
61	Attr62	127	30.73%
30	Attr31	127	30.73%
18	Attr19	128	30.98%
62	Attr63	134	32.43%
3	Attr4	134	32.43%
39	Attr40	134	32.43%
32	Attr33	134	32.43%
11	Attr12	134	32.43%
45	Attr46	135	32.67%
46	Attr47	297	71.87%
51	Attr52	301	72.84%

31	Attr32	368	89.06%
40	Attr41	754	182.47%
52	Attr53	812	196.51%
53	Attr54	812	196.51%
63	Attr64	812	196.51%
27	Attr28	812	196.51%
23	Attr24	922	223.13%
44	Attr45	2147	519.58%
59	Attr60	2152	520.79%
26	Attr27	2764	668.89%
20	Attr21	5854	1,416.68%
36	Attr37	18984	4,594.16%

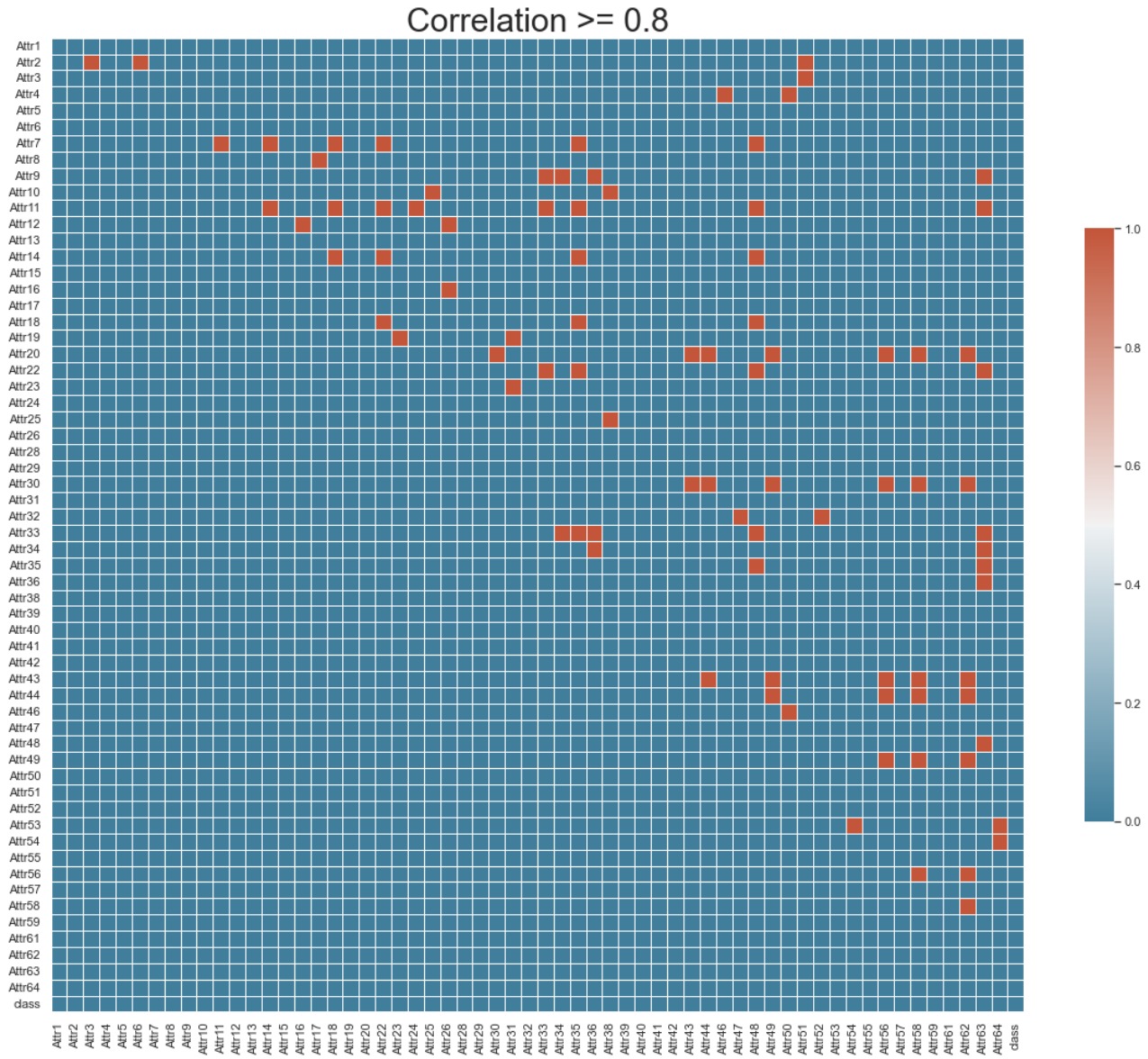
In [14]:
```
df = df.drop(['Attr45', 'Attr60', 'Attr27', 'Attr21', 'Attr37'], axis = 1)
df.shape
```

Out[14]:
```
(43405, 60)
```

Correlation

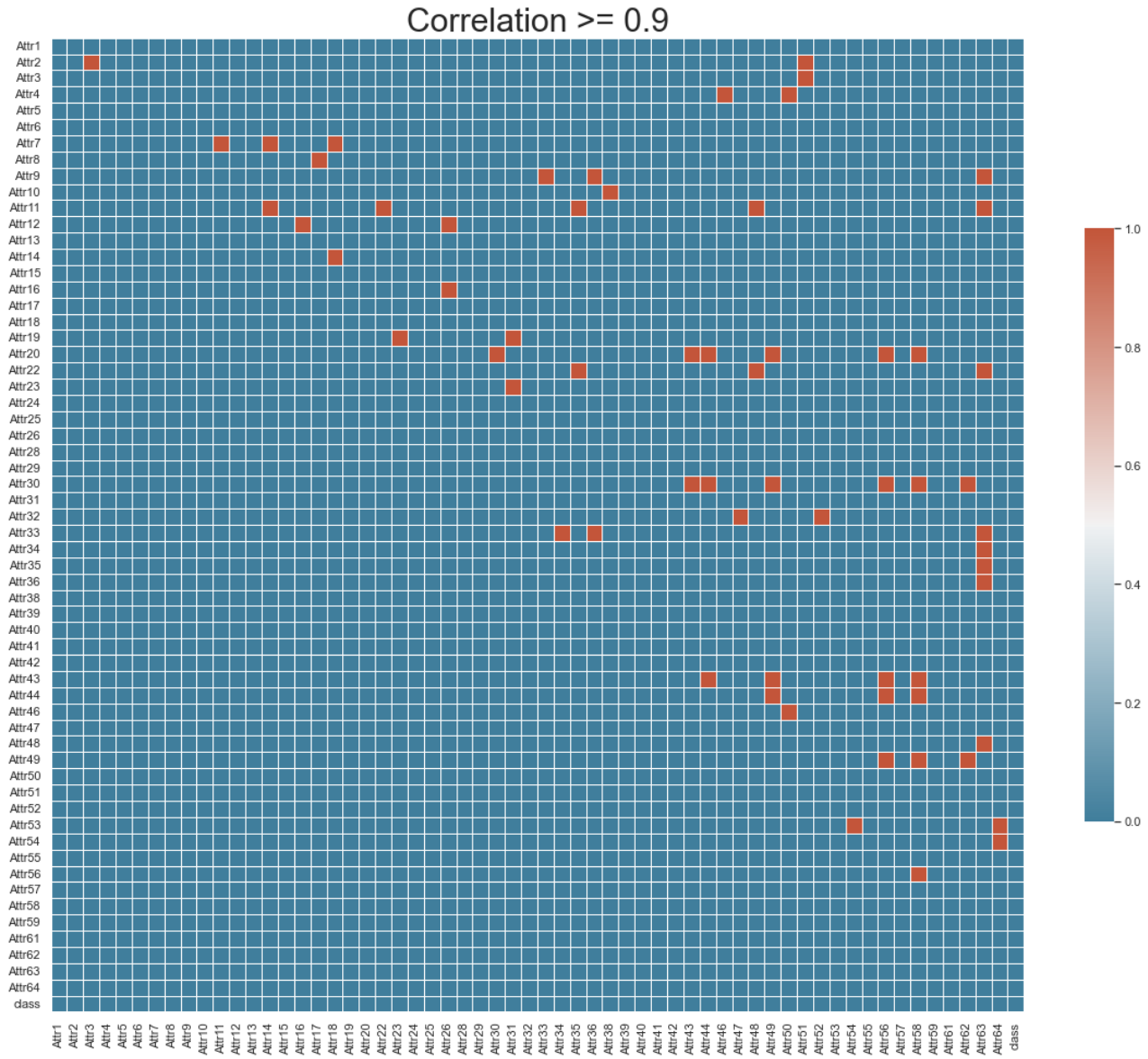
In [15]:
```
corr = df[df.select_dtypes(include = "number").columns].corr().abs()
upper_tri = corr.where(np.triu(np.ones(corr.shape), k = 1).astype(np.bool))

sns.set_theme(style = "white")
plt.subplots(figsize = (20,20))
sns.heatmap(upper_tri >= 0.8,
            cmap = sns.diverging_palette(230,
                                         20,
                                         as_cmap = True),
            vmax = 1,
            center = 0.5,
            annot = False,
            square = True,
            linewidths = .5,
            cbar_kws = {"shrink": .5}).set_title('Correlation >= 0.8', fonts
```

Out[15]:
```
Text(0.5, 1.0, 'Correlation >= 0.8')
```



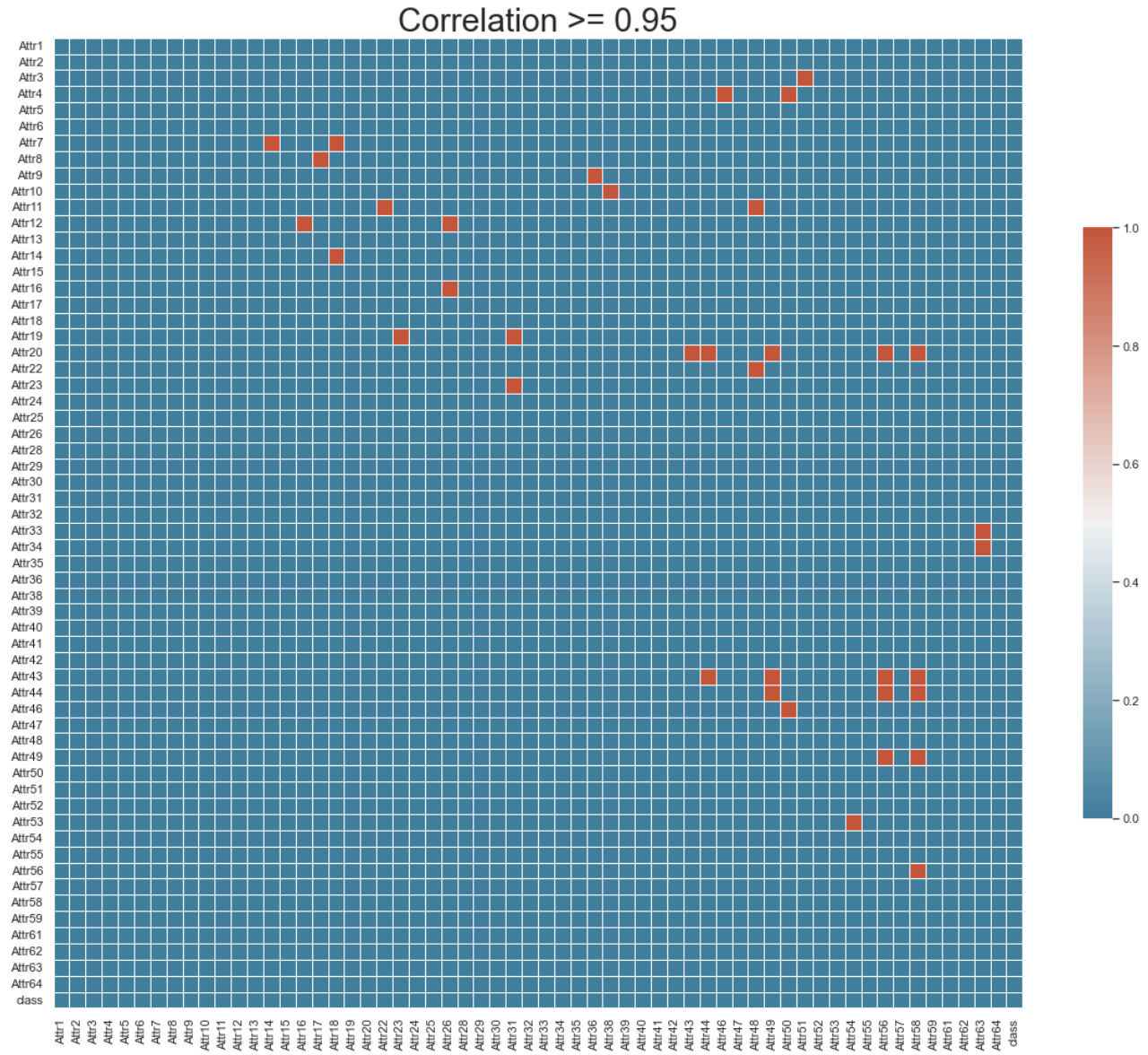
In [16]:
```
corr = df[df.select_dtypes(include = "number").columns].corr().abs()
upper_tri = corr.where(np.triu(np.ones(corr.shape),k = 1).astype(np.bool))
sns.set_theme(style = "dark")
plt.subplots(figsize = (20,20))
sns.heatmap(upper_tri >= 0.9,
            cmap = sns.diverging_palette(230,
                                         20,
                                         as_cmap = True),
            vmax = 1,
            center = 0.5,
            annot = False,
            square = True,
            linewidths = .5,
            cbar_kws = {"shrink": .5}).set_title('Correlation >= 0.9', fonts
```

Out[16]:
```
Text(0.5, 1.0, 'Correlation >= 0.9')
```



In [17]:
```
corr = df[df.select_dtypes(include = "number").columns].corr().abs()
upper_tri = corr.where(np.triu(np.ones(corr.shape),k = 1).astype(np.bool))
sns.set_theme(style = "dark")
plt.subplots(figsize = (20,20))
sns.heatmap(upper_tri >= 0.95,
            cmap = sns.diverging_palette(230,
                                         20,
                                         as_cmap = True),
            vmax = 1,
            center = 0.5,
            annot = False,
            square = True,
            linewidths = .5,
            cbar_kws = {"shrink": .5}).set_title('Correlation >= 0.95', font
```

Out[17]:
```
Text(0.5, 1.0, 'Correlation >= 0.95')
```



In [18]:
```
corr = df[df.select_dtypes(include = "number").columns].corr().abs()
upper_tri = corr.where(np.triu(np.ones(corr.shape),k = 1).astype(np.bool))
sns.set_theme(style = "dark")
sns.set(font_scale = 2)
plt.subplots(figsize = (20,20))
sns.heatmap(upper_tri >= 0.98,
            cmap = sns.diverging_palette(230,
                                         20,
                                         as_cmap = True),
            vmax = 1,
            center = 0.5,
            annot = False,
            square = True,
            linewidths = .5,
            cbar_kws = {"shrink": .5}).set_title('Correlation >= 0.98', font
```

Out[18]:
```
Text(0.5, 1.0, 'Correlation >= 0.98')
```

In [19]:
```
# Check for correlation greater than 0.98
[column for column in upper_tri.columns if any(upper_tri[column] >= 0.98)]
```

Out[19]:
```
['Attr14',
 'Attr16',
 'Attr17',
 'Attr18',
 'Attr22',
 'Attr23',
 'Attr26',
 'Attr31',
 'Attr36',
 'Attr38',
 'Attr43',
 'Attr44',
 'Attr46',
 'Attr50',
 'Attr51',
 'Attr54',
 'Attr56',
 'Attr58',
 'Attr63']
```

In [20]:
```
# Drop highly correlated variables
cols_to_drop = [column for column in upper_tri.columns if any(upper_tri[colu
df.drop(cols_to_drop, axis = 1, inplace = True)
```

In [21]:
```
df.shape
```

Out[21]:
```
(43405, 41)
```

Final Check

In [22]:
```
corr = df[df.select_dtypes(include = "number").columns].corr().abs()
upper_tri = corr.where(np.triu(np.ones(corr.shape),k=1).astype(np.bool))

sns.set_theme(style = "white")
plt.subplots(figsize = (20,20))
cmap = sns.diverging_palette(230, 20, as_cmap=True)
sns.heatmap(upper_tri,
            cmap = cmap,
            vmax = 1,
            center = 0.5,
            annot = False,
            square = True,
            linewidths = .5,
            cbar_kws = {"shrink": .5})
```

Out[22]:
```
<AxesSubplot:>
```

Helper Functions

In [23]:

```
def get_acc_score(model, x, y):
    return model.score(x, y)

def plot_roc_curve_custom(x_test, y_test, model):
    fpr, tpr, _ = roc_curve(y_test, model.predict_proba(x_test)[:,1])
    roc_auc = auc(fpr, tpr)
    plt.plot(fpr,
             tpr,
             color = 'darkorange',
             label = 'Randomized Search CV (AUC = ' + str(round(roc_auc, 5))
    plt.plot([0, 1],
             [0, 1],
             color = 'navy', lw = 2, linestyle='--')
    plt.legend(loc='lower right')
    plt.ylabel('True Positive Rate')
    plt.xlabel('False Positive Rate')
    plt.title('Receiver operating characteristic',fontdict={'fontsize':20},p
```

```
def get_confusion_matrix(y, yhat, title = "Confusion Matrix"):
    cm_n = confusion_matrix(y, yhat)
    ax = sns.heatmap(cm_n,
                     cmap = 'Blues',
                     nnot = True,
                     fmt = '2d',
                     xticklabels = ['Not Bankrupt','Bankrupt'],
                     yticklabels = ['Not Bankrupt','Bankrupt'])
    ax.set(xlabel='Predicted Label', ylabel='True Label')
    ax.set_title(title,fontdict = {'fontsize':20}, pad = 10)

def get_classification_report(x_train, y_train, x_test, y_test, pred, model)
    """
    This function is used to get comprehensive classification report:
    Training Accuracy, Test Accuracy, print classification_report
    plot confusion matrix, and plot roc curve.
    """
    print(f"Train accuracy: {get_acc_score(model, x_train, y_train):,.5f}")
    print("Test result:")
    print(classification_report(y_test,pred))
    fpr, tpr, thresholds = roc_curve(y_test, model.predict_proba(x_test)[:,1
    print(f"Test AUC: {auc(fpr, tpr):,.5f}")
    plot_roc_curve_custom(x_test, y_test, model)
    plot_confusion_matrix(model, x_test, y_test,colorbar=False)
    plt.grid(False)

def cv_common(df, columns):
    cv_result_summary = df[columns]
    cv_result_summary.index  = np.arange(1,len(cv_result_summary)+1)
    cv_result_summary = cv_result_summary.reset_index()
    return cv_result_summary


def cv_summary(estimator, columns):
    df = pd.DataFrame(estimator.cv_results_)
    cv_result_summary = cv_common(df, columns)
    cv_result_summary = cv_result_summary.rename(columns = {
        "index": "param_combination",
        "mean_test_score": "mean_validation_score",
        "rank_test_score": "rank_validation_score"
    })
    return cv_result_summary
```

Model Prep

In [25]:
```
# Prepare test and train data
X = df.loc[:, df.columns != 'class'].values
y = df['class'].values

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size = 0.2, random_state = 1, shuffle = True, stratify = y)

# Impute
impute_mean = SimpleImputer(missing_values = np.nan, strategy = 'mean')
impute_mean.fit(X_train)
X_train = impute_mean.transform(X_train)
X_test = impute_mean.transform(X_test)

# Normalize the data
transformer = RobustScaler().fit(X_train)
transformer = RobustScaler().fit(X_test)
X_train = transformer.transform(X_train)
X_test = transformer.transform(X_test)
```

Model Building

Random Forest Grid Search

In [26]:
```
params_rf = {
    "model__n_estimators": [250, 500, 750, 1000],
    "model__criterion": ['gini','entropy'],
    "model__max_depth":[5, 10, 15, 20],
    "model__min_samples_split":[4, 6, 8, 10],
    "model__n_jobs": [-1],
    "model__class_weight":["balanced"]
}

cv = StratifiedKFold(n_splits = 10,
                     shuffle = True,
                     random_state = 1)
```

In [27]:
```
%%time

# Define preprocessing for columns
rfc = RandomForestClassifier(random_state = 1)

rf_pipeline = Pipeline(steps=[('preprocessing', preprocessing),('model', rfc

# Fit
clf_rfc = RandomizedSearchCV(rf_pipeline,
                             params_rf,
                             cv = cv,
                             scoring = 'roc_auc',
                             n_jobs = -1,
                             verbose = 15,
                             return_train_score = True)

search_rfc = clf_rfc.fit(X_train,y_train)
```

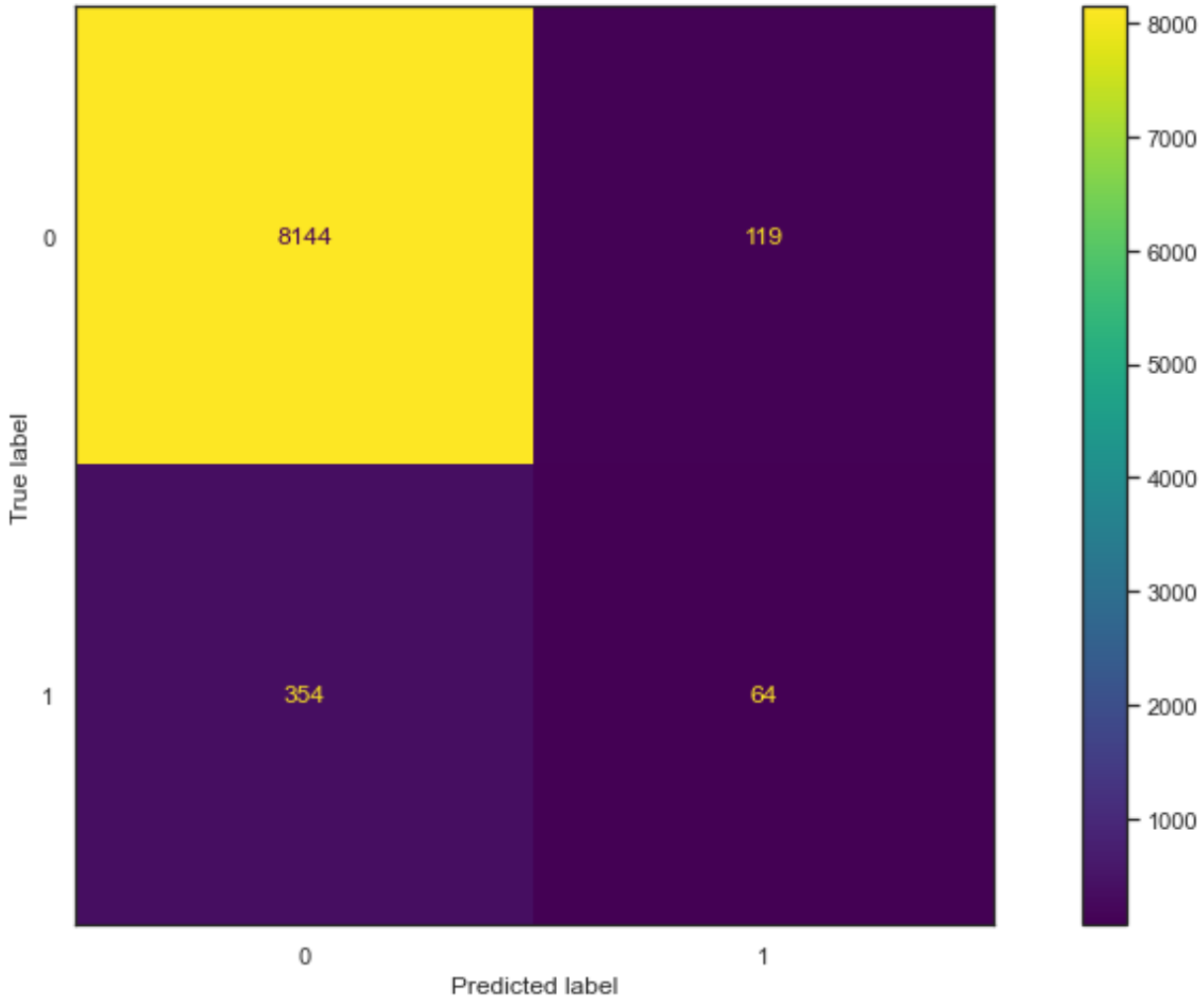
```
Fitting 10 folds for each of 10 candidates, totalling 100 fits
CPU times: user 1min 50s, sys: 725 ms, total: 1min 51s
Wall time: 8min 47s
```

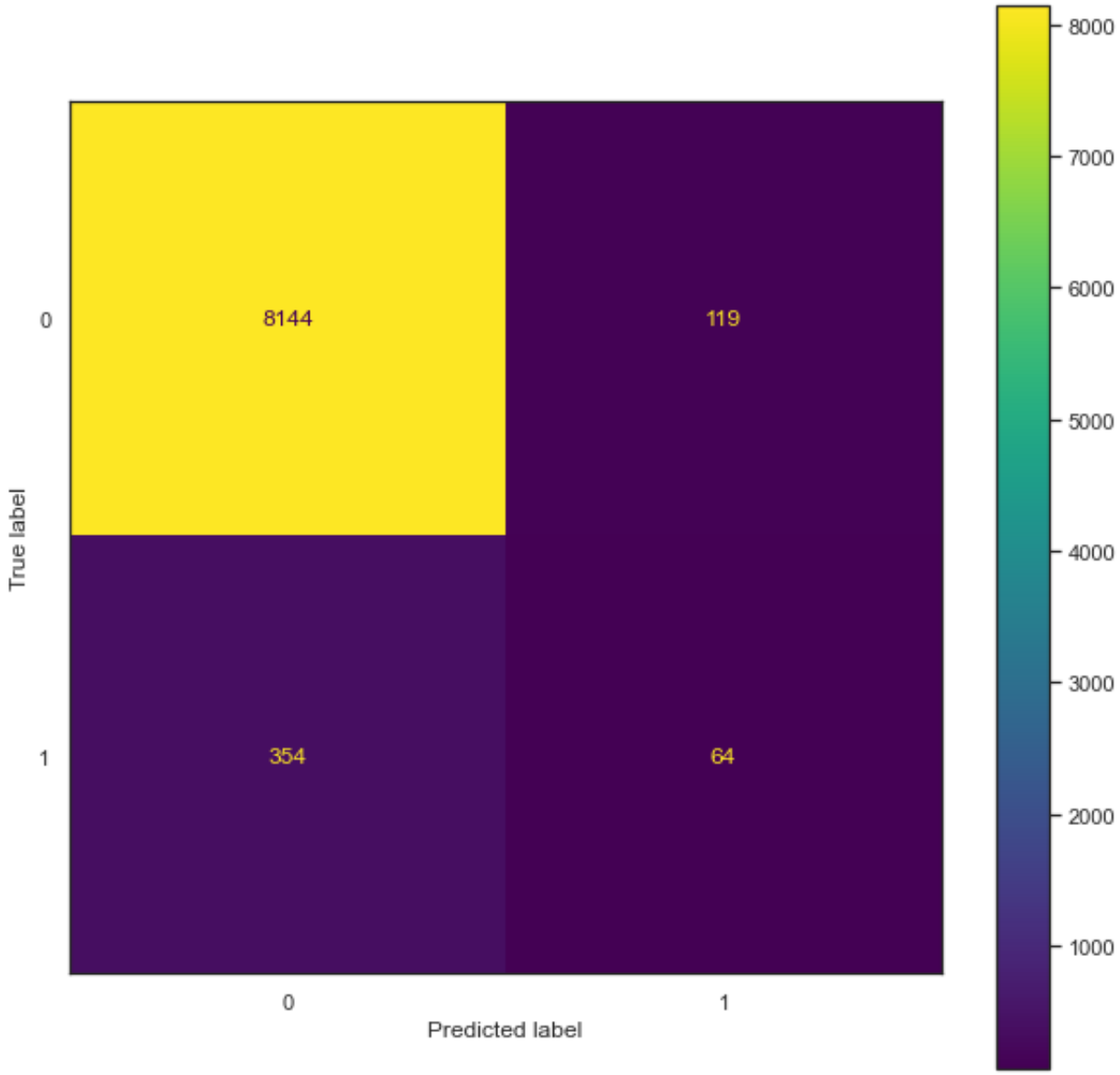
In [28]:
```
# Evaluate model
y_hat_rf_test = search_rfc.predict(X_test)
accuracy_score(y_hat_rf_test, y_test)
```

Out[28]:
```
0.9455131897246861
```

In [29]:
```
# Confusion matrix
confusion_matrix(y_test, y_hat_rf_test)
disp = ConfusionMatrixDisplay.from_estimator(search_rfc, X_test, y_test)
fig, ax = plt.subplots(figsize=(10,10))
disp.plot(ax = ax)
```

Out[29]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x13f103af
0>
```


8144
119
354
64
0
1
True label
0
1
Predicted label
8000
7000
6000
5000
4000
3000
2000
1000

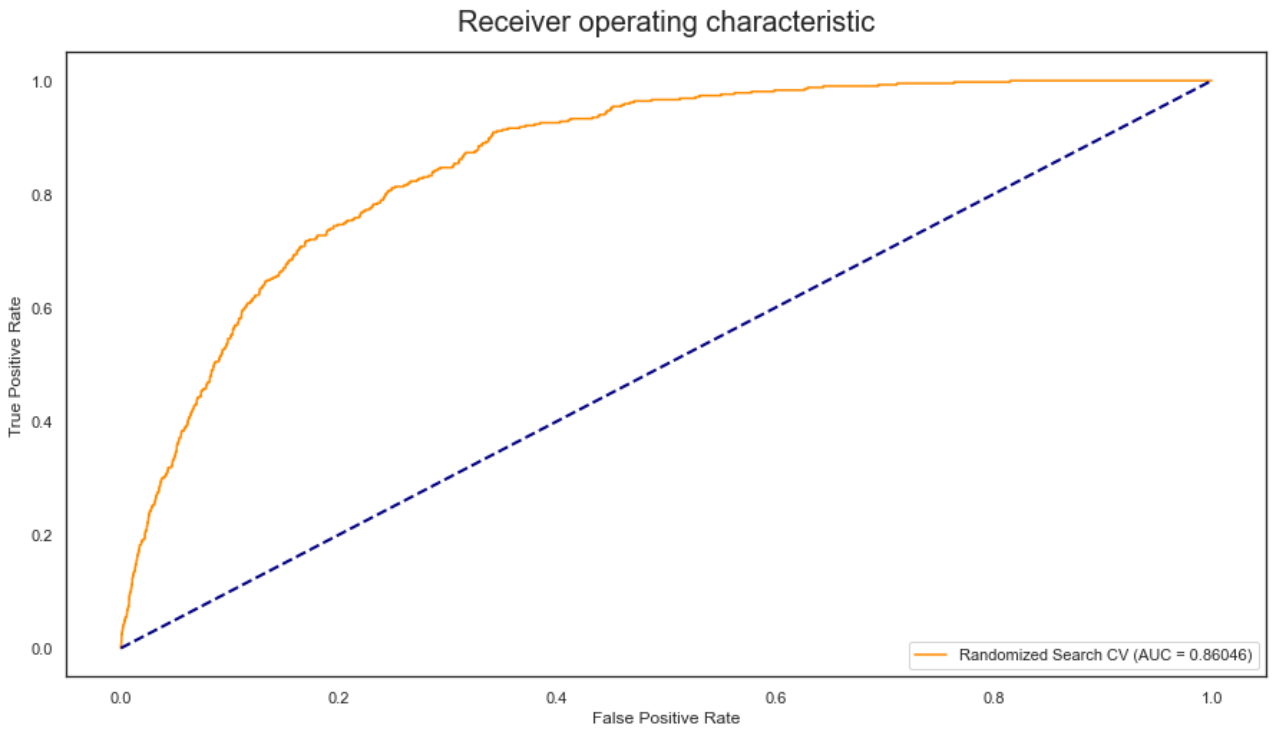


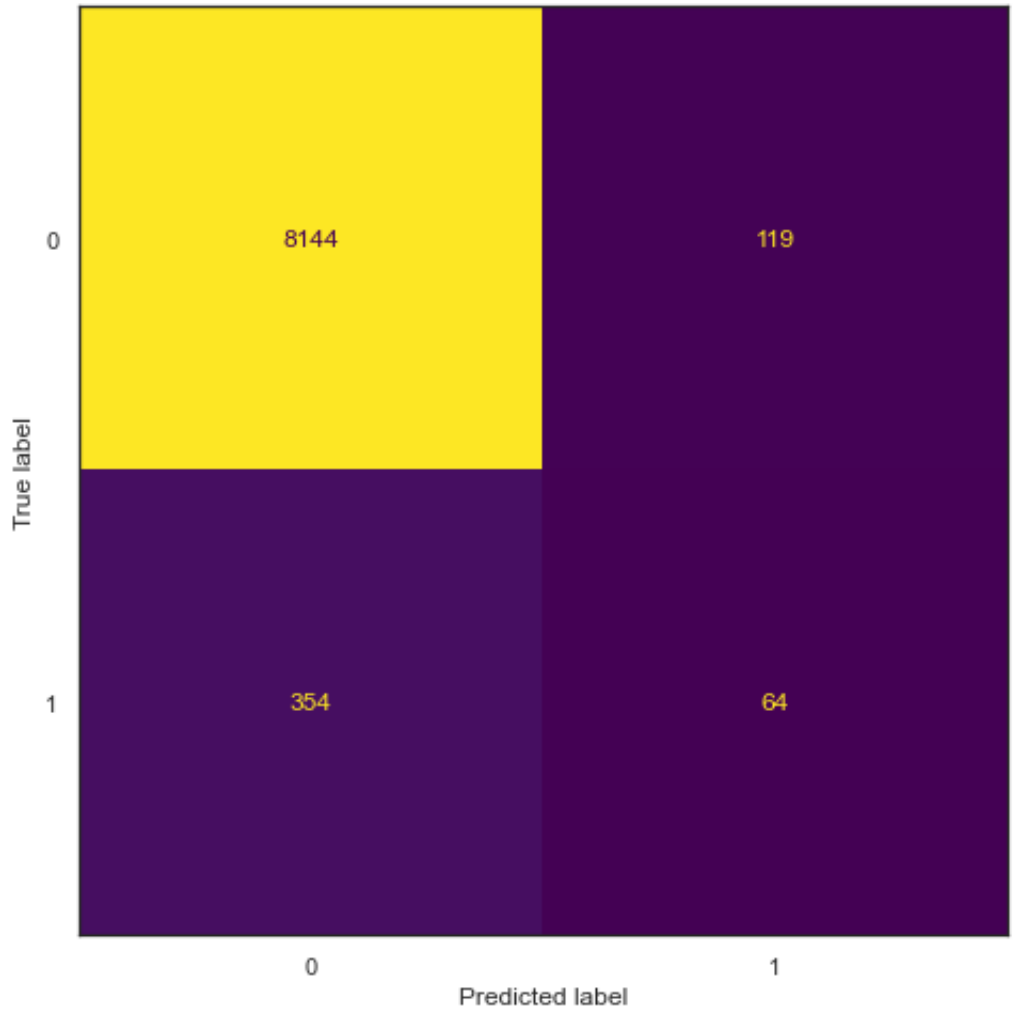
In [30]:
```
# Precision and recall
print("Recall:", recall_score(y_test, y_hat_rf_test, pos_label = 1, average
print("Precision:", precision_score(y_test, y_hat_rf_test, pos_label = 1, av
```

```
Recall: 0.15311004784688995
Precision: 0.34972677595628415
```

In [31]:
```
get_classification_report(X_train, y_train, X_test, y_test, y_hat_rf_test, s
```

```
Train accuracy: 0.99895
Test result:
              precision    recall  f1-score   support

           0       0.96      0.99      0.97      8263
           1       0.35      0.15      0.21       418

    accuracy                           0.95      8681
   macro avg       0.65      0.57      0.59      8681
weighted avg       0.93      0.95      0.94      8681

Test AUC: 0.86046
```


Receiver operating characteristic
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Randomized Search CV (AUC = 0.86046)


8144
119
354
64
True label
Predicted label
0
1

XGBoost

In [32]:
```
X_train_xg = preprocessing.fit_transform(X_train)
X_test_xg = preprocessing.transform(X_test)

dtrain = xgboost.DMatrix(X_train_xg, label=y_train)
dtest = xgboost.DMatrix(X_test_xg, label=y_test)

evallist = [(dtrain,'train'), (dtest,'eval')]
```

In [33]:
```
num_round = 1000
param = {
    'max_depth':10,
    'objective':'binary:logistic',
    'eval_metric':'logloss',
    'eta': 0.01
}
```

In [34]:
```
%%time
xg_model = xgboost.cv(params = param,
                      dtrain = dtrain,
                      num_boost_round = num_round,
                      nfold = 5,
                      verbose_eval = False,
                      stratified = True,
                      early_stopping_rounds = 5,
                      seed = 1)
```

```
CPU times: user 42min 49s, sys: 28.4 s, total: 43min 18s
Wall time: 2min 46s
```

In [35]:
```
xg_model.head()
```

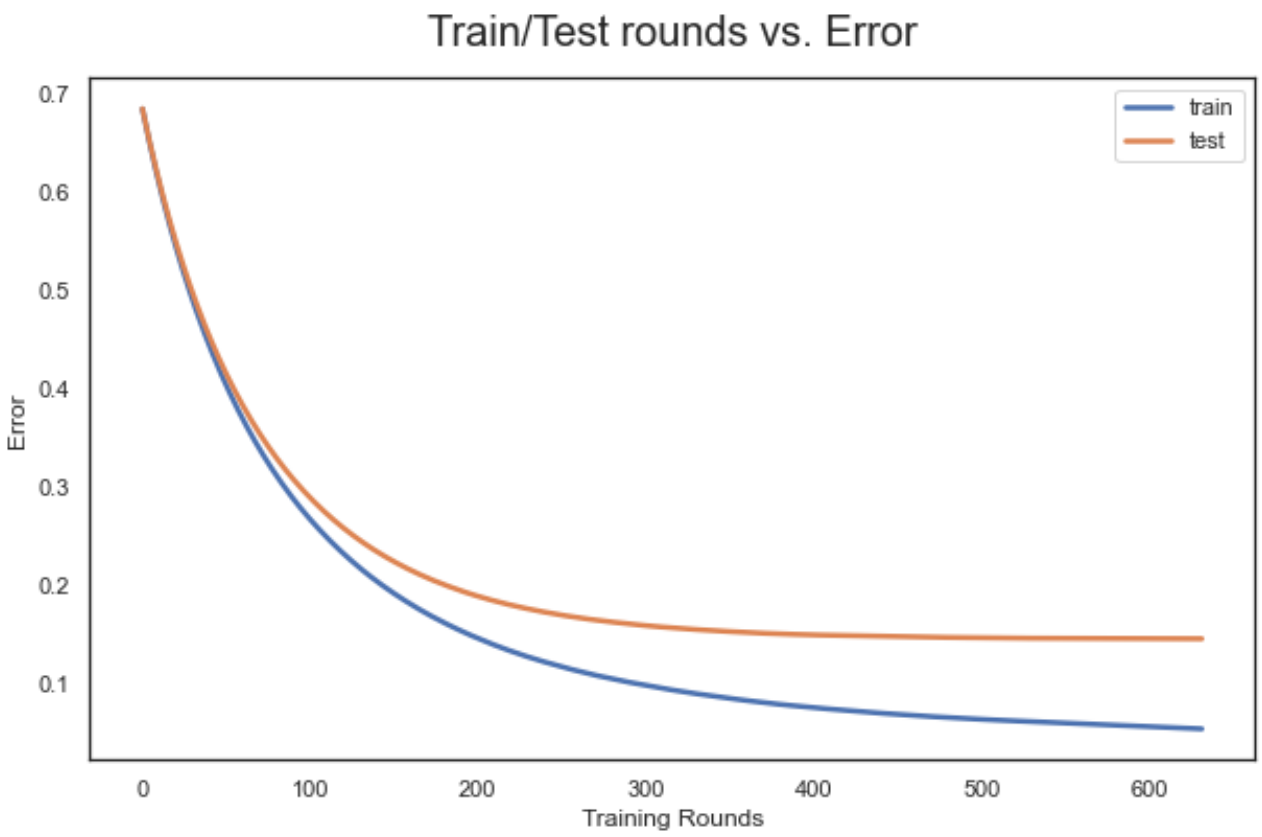
Out[35]:

	train-logloss-mean	train-logloss-std	test-logloss-mean	test-logloss-std
0	0.684635	0.000032	0.684912	0.000020
1	0.676306	0.000068	0.676835	0.000029
2	0.668136	0.000098	0.668908	0.000027
3	0.660119	0.000137	0.661150	0.000032
4	0.652246	0.000162	0.653531	0.000041

In [36]:
```
xg_model.shape
```

Out[36]:
```
(632, 4)
```

In [37]:
```
def plot_early_stop_rounds():
    plt.subplots(figsize=(10, 6))
    plt.plot(xg_model['train-logloss-mean'],lw = 2.5, label = 'train')
    plt.plot(xg_model['test-logloss-mean'],lw = 2.5, label = 'test')
    plt.title('Train/Test rounds vs. Error',fontdict = {'fontsize':20}, pad
    plt.ylabel('Error')
    plt.xlabel('Training Rounds')
    plt.legend()
    plt.show()
plot_early_stop_rounds()
```



Search for best parameters using sklearn wrapper

In [38]:
```
X_train_xg = preprocessing.fit_transform(X_train)
X_test_xg = preprocessing.transform(X_test)
```

In [39]:
```
search_space = [
    {
        "model__learning_rate"    : [0.01, 0.1] ,
        "model__max_depth"        : [7, 8, 9],
        "model__subsample"        : [0.7, 0.9, 1],
        "model__gamma"            : [ 0.1, 0.4, 0.6],
     }
]
```

In [44]:
```
## %%time
model_xgb = xgboost.XGBClassifier(
    n_estimators = 1000,
    use_label_encoder = False,
    objective = "binary:logistic",
    random_state = 1,
)

fit_params={"model__early_stopping_rounds": 100,
            "model__eval_metric" : "auc",
            "model__verbose":False,
            "model__eval_set" : [[X_test_xg, y_test]]}

xgb_pipeline = Pipeline(steps=[('preprocessing',preprocessing),
                               ('model',model_xgb)])

clf_xgb = RandomizedSearchCV(xgb_pipeline,
                             search_space,
                             cv = cv,
                             scoring = 'roc_auc',
                             n_iter = 10,
                             n_jobs = -1,
                             verbose = 15,
                             return_train_score = True,
                             random_state = 1)

search_clf_xgb = clf_xgb.fit(X_train, y_train, **fit_params)
```

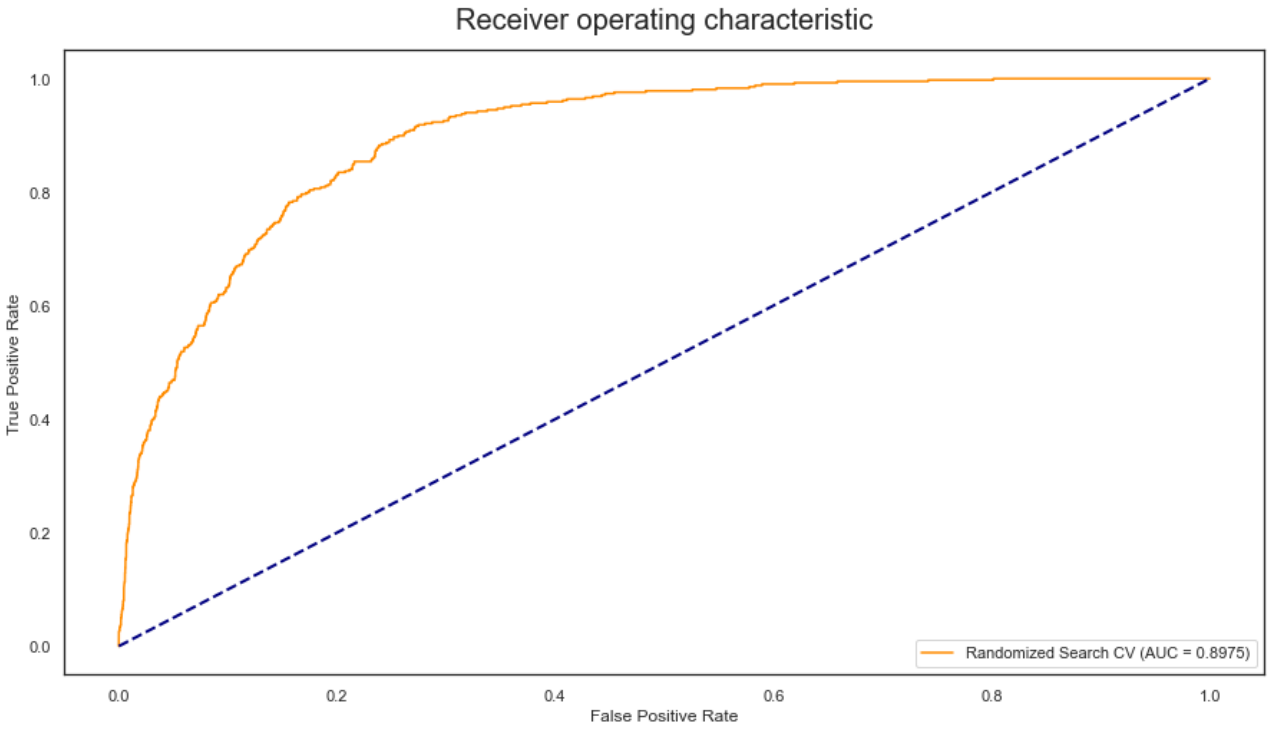
```
Fitting 10 folds for each of 10 candidates, totalling 100 fits
/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packa
ges/xgboost/sklearn.py:793: UserWarning: `eval_metric` in `fit` method is de
precated for better compatibility with scikit-learn, use `eval_metric` in co
nstructor or`set_params` instead.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packa
ges/xgboost/sklearn.py:793: UserWarning: `early_stopping_rounds` in `fit` me
thod is deprecated for better compatibility with scikit-learn, use `early_st
opping_rounds` in constructor or`set_params` instead.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packa
ges/xgboost/sklearn.py:793: UserWarning: `eval_metric` in `fit` method is de
precated for better compatibility with scikit-learn, use `eval_metric` in co
nstructor or`set_params` instead.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packa
ges/xgboost/sklearn.py:793: UserWarning: `early_stopping_rounds` in `fit` me
thod is deprecated for better compatibility with scikit-learn, use `early_st
opping_rounds` in constructor or`set_params` instead.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packa
ges/xgboost/sklearn.py:793: UserWarning: `eval_metric` in `fit` method is de
precated for better compatibility with scikit-learn, use `eval_metric` in co
nstructor or`set_params` instead.
  warnings.warn(
/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packa
ges/xgboost/sklearn.py:793: UserWarning: `eval_metric` in `fit` method is de
```

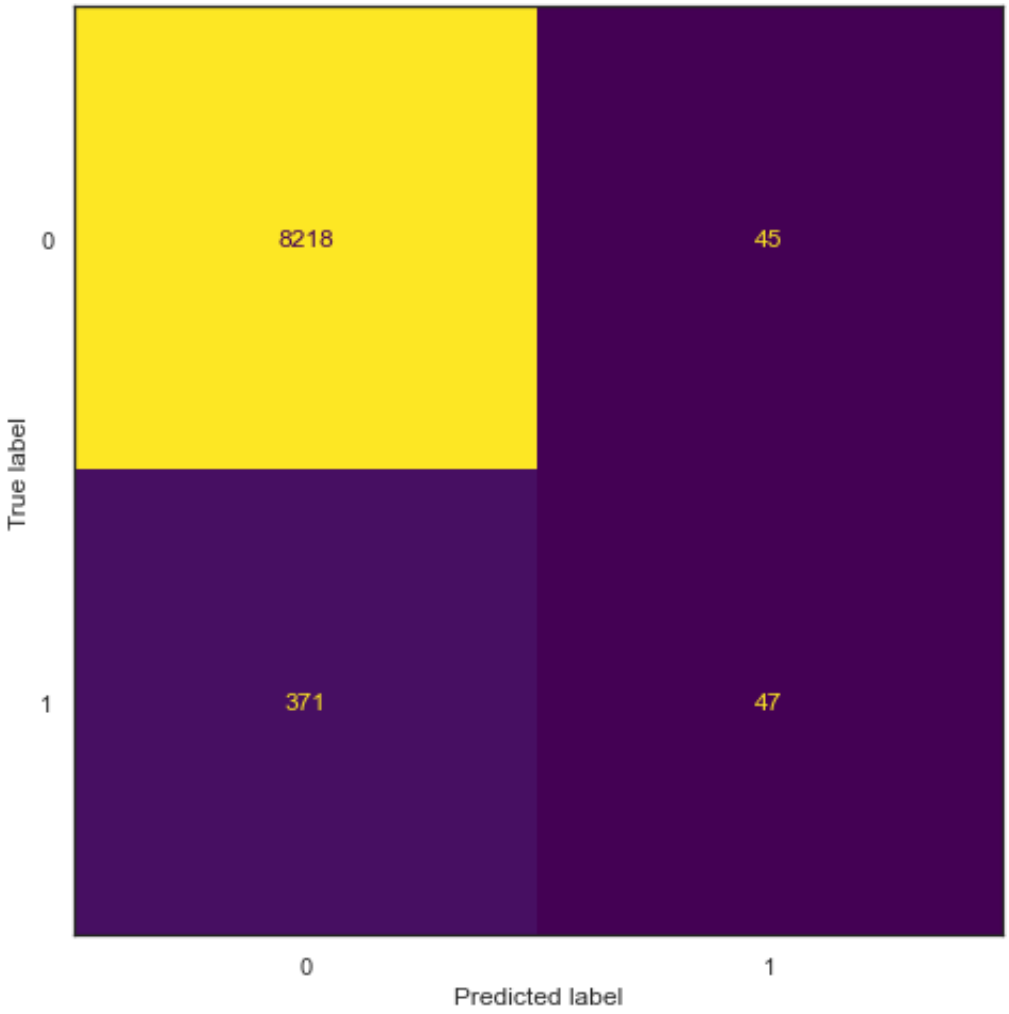
Out[42]:

	param_combination	param_model__learning_rate	param_model__max_depth	param_mod
0	1	0.1	8	
1	2	0.01	7	
2	3	0.01	9	
3	4	0.1	8	
4	5	0.1	8	
5	6	0.1	7	
6	7	0.1	9	
7	8	0.01	8	
8	9	0.1	7	
9	10	0.01	7	

In [43]:
```
y_hat_xgb = search_clf_xgb.predict(X_test)
get_classification_report(X_train, y_train, X_test, y_test, y_hat_xgb, searc
```

```
Train accuracy: 0.99983
Test result:
              precision    recall  f1-score   support

           0       0.96      0.99      0.98      8263
           1       0.51      0.11      0.18       418

    accuracy                           0.95      8681
   macro avg       0.73      0.55      0.58      8681
weighted avg       0.94      0.95      0.94      8681

Test AUC: 0.89750
```






```
[CV 1/10; 1/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 1/10; 1/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.990, test=0.838) total time=  42.9s
[CV 7/10; 2/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 7/10; 2/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.997, test=0.834) total time=  43.3s
[CV 3/10; 4/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 3/10; 4/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.995, test=0.826) total time= 1.7min
[CV 9/10; 5/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=8, model__n_estimators=250, mod
el__n_jobs=-1
[CV 9/10; 5/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=8, model__n_estimators=250, model
__n_jobs=-1;, score=(train=0.998, test=0.867) total time=  47.0s
[CV 1/10; 7/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, m
```

```
odel__n_jobs=-1
[CV 1/10; 7/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, mod
el__n_jobs=-1;, score=(train=0.828, test=0.779) total time=  27.1s
[CV 8/10; 7/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, m
odel__n_jobs=-1
[CV 8/10; 7/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, mod
el__n_jobs=-1;, score=(train=0.825, test=0.794) total time=  20.6s
[CV 3/10; 9/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=10, model__min_samples_split=8, model__n_estimators=250,
model__n_jobs=-1
[CV 3/10; 9/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=10, model__min_samples_split=8, model__n_estimators=250, mo
del__n_jobs=-1;, score=(train=0.955, test=0.814) total time=  45.7s
[CV 9/10; 9/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=10, model__min_samples_split=8, model__n_estimators=250,
model__n_jobs=-1
[CV 9/10; 9/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=10, model__min_samples_split=8, model__n_estimators=250, mo
del__n_jobs=-1;, score=(train=0.953, test=0.834) total time=  43.9s
[CV 5/10; 10/10] START model__class_weight=balanced, model__criterion=gini,
model__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, m
odel__n_jobs=-1
[CV 5/10; 10/10] END model__class_weight=balanced, model__criterion=gini, mo
del__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, mod
el__n_jobs=-1;, score=(train=0.992, test=0.856) total time= 2.1min
[CV 2/10; 1/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.7
[CV 2/10; 1/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.7;, score=(train=0.911, test=0.844) total time=
16.6s
[CV 10/10; 2/10] START model__gamma=0.1, model__learning_rate=0.01, model__m
ax_depth=7, model__subsample=1
[CV 10/10; 2/10] END model__gamma=0.1, model__learning_rate=0.01, model__max
_depth=7, model__subsample=1;, score=(train=0.807, test=0.783) total time=
9.8s
[CV 4/10; 3/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=9, model__subsample=1
[CV 4/10; 3/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=9, model__subsample=1;, score=(train=0.834, test=0.762) total time=  1
5.6s
[CV 8/10; 3/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=9, model__subsample=1
[CV 8/10; 3/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=9, model__subsample=1;, score=(train=0.848, test=0.782) total time=  2
2.7s
[CV 8/10; 4/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=1
[CV 8/10; 4/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=1;, score=(train=0.907, test=0.826) total time=  18
.2s
[CV 5/10; 5/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.9
[CV 5/10; 5/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.9;, score=(train=0.993, test=0.896) total time=
```

```
50.2s
[CV 8/10; 6/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=1
[CV 8/10; 6/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=1;, score=(train=0.981, test=0.863) total time=  43
.5s
[CV 8/10; 7/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=9, model__subsample=0.9
[CV 8/10; 7/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=9, model__subsample=0.9;, score=(train=0.998, test=0.873) total time=
58.4s
[CV 4/10; 9/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=0.9
[CV 4/10; 9/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=0.9;, score=(train=0.991, test=0.880) total time=
55.7s
[CV 7/10; 10/10] START model__gamma=0.4, model__learning_rate=0.01, model__m
ax_depth=7, model__subsample=0.9
[CV 7/10; 10/10] END model__gamma=0.4, model__learning_rate=0.01, model__max
_depth=7, model__subsample=0.9;, score=(train=0.797, test=0.722) total time=
9.9s
[CV 9/10; 1/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 9/10; 1/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.991, test=0.855) total time=  45.3s
[CV 6/10; 3/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=20, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 6/10; 3/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=20, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.999, test=0.849) total time= 1.8min
[CV 3/10; 5/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=8, model__n_estimators=250, mod
el__n_jobs=-1
[CV 3/10; 5/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=8, model__n_estimators=250, model
__n_jobs=-1;, score=(train=0.997, test=0.831) total time=  46.0s
[CV 4/10; 6/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mo
del__n_jobs=-1
[CV 4/10; 6/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mode
l__n_jobs=-1;, score=(train=0.954, test=0.829) total time= 1.2min
[CV 4/10; 8/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=750,
model__n_jobs=-1
[CV 4/10; 8/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=750, mo
del__n_jobs=-1;, score=(train=0.995, test=0.849) total time= 2.4min
[CV 9/10; 10/10] START model__class_weight=balanced, model__criterion=gini,
model__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, m
odel__n_jobs=-1
[CV 9/10; 10/10] END model__class_weight=balanced, model__criterion=gini, mo
del__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, mod
el__n_jobs=-1;, score=(train=0.992, test=0.849) total time= 1.5min
```

```
[CV 5/10; 1/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.7
[CV 5/10; 1/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.7;, score=(train=0.993, test=0.905) total time=
46.8s
[CV 10/10; 3/10] START model__gamma=0.6, model__learning_rate=0.01, model__m
ax_depth=9, model__subsample=1
[CV 10/10; 3/10] END model__gamma=0.6, model__learning_rate=0.01, model__max
_depth=9, model__subsample=1;, score=(train=0.820, test=0.782) total time=
12.7s
[CV 7/10; 4/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=1
[CV 7/10; 4/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=1;, score=(train=0.990, test=0.855) total time=  43
.2s
[CV 10/10; 5/10] START model__gamma=0.4, model__learning_rate=0.1, model__ma
x_depth=8, model__subsample=0.9
[CV 10/10; 5/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_
depth=8, model__subsample=0.9;, score=(train=0.922, test=0.834) total time=
18.8s
[CV 5/10; 6/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=1
[CV 5/10; 6/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=1;, score=(train=0.995, test=0.895) total time= 1.2
min
[CV 1/10; 8/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=8, model__subsample=0.9
[CV 1/10; 8/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=8, model__subsample=0.9;, score=(train=0.861, test=0.789) total time=
19.2s
[CV 6/10; 8/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=8, model__subsample=0.9
[CV 6/10; 8/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=8, model__subsample=0.9;, score=(train=0.862, test=0.774) total time=
17.6s
[CV 1/10; 9/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=0.9
[CV 1/10; 9/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=0.9;, score=(train=0.995, test=0.886) total time= 1
.2min
[CV 5/10; 2/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 5/10; 2/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.997, test=0.866) total time=  46.6s
[CV 9/10; 3/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=20, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 9/10; 3/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=20, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.999, test=0.864) total time= 1.8min
[CV 8/10; 5/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=8, model__n_estimators=250, mod
el__n_jobs=-1
[CV 8/10; 5/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=8, model__n_estimators=250, model
```

```
__n_jobs=-1;, score=(train=0.997, test=0.843) total time=  46.8s
[CV 10/10; 6/10] START model__class_weight=balanced, model__criterion=gini,
model__max_depth=10, model__min_samples_split=10, model__n_estimators=500, m
odel__n_jobs=-1
[CV 10/10; 6/10] END model__class_weight=balanced, model__criterion=gini, mo
del__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mod
el__n_jobs=-1;, score=(train=0.953, test=0.843) total time= 1.2min
[CV 10/10; 8/10] START model__class_weight=balanced, model__criterion=entrop
y, model__max_depth=15, model__min_samples_split=8, model__n_estimators=750,
model__n_jobs=-1
[CV 10/10; 8/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=750, mo
del__n_jobs=-1;, score=(train=0.995, test=0.854) total time= 2.4min
[CV 4/10; 1/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.7
[CV 4/10; 1/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.7;, score=(train=0.838, test=0.765) total time=
10.8s
[CV 9/10; 2/10] START model__gamma=0.1, model__learning_rate=0.01, model__ma
x_depth=7, model__subsample=1
[CV 9/10; 2/10] END model__gamma=0.1, model__learning_rate=0.01, model__max_
depth=7, model__subsample=1;, score=(train=0.876, test=0.828) total time=  5
8.5s
[CV 1/10; 5/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.9
[CV 1/10; 5/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.9;, score=(train=0.994, test=0.884) total time=
49.5s
[CV 3/10; 6/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=1
[CV 3/10; 6/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=1;, score=(train=0.996, test=0.880) total time= 1.3
min
[CV 2/10; 8/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=8, model__subsample=0.9
[CV 2/10; 8/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=8, model__subsample=0.9;, score=(train=0.850, test=0.823) total time=
16.8s
[CV 7/10; 8/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=8, model__subsample=0.9
[CV 7/10; 8/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=8, model__subsample=0.9;, score=(train=0.805, test=0.721) total time=
11.5s
[CV 10/10; 8/10] START model__gamma=0.6, model__learning_rate=0.01, model__m
ax_depth=8, model__subsample=0.9
[CV 10/10; 8/10] END model__gamma=0.6, model__learning_rate=0.01, model__max
_depth=8, model__subsample=0.9;, score=(train=0.843, test=0.812) total time=
14.5s
[CV 6/10; 9/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=0.9
[CV 6/10; 9/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=0.9;, score=(train=0.993, test=0.872) total time= 1
.0min
[CV 10/10; 1/10] START model__class_weight=balanced, model__criterion=gini,
model__max_depth=15, model__min_samples_split=10, model__n_estimators=250, m
odel__n_jobs=-1
[CV 10/10; 1/10] END model__class_weight=balanced, model__criterion=gini, mo
```

```
del__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mod
el__n_jobs=-1;, score=(train=0.991, test=0.858) total time=  45.1s
[CV 4/10; 3/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=20, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 4/10; 3/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=20, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.999, test=0.858) total time= 1.8min
[CV 10/10; 4/10] START model__class_weight=balanced, model__criterion=entrop
y, model__max_depth=15, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 10/10; 4/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.994, test=0.854) total time= 1.6min
[CV 6/10; 7/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, m
odel__n_jobs=-1
[CV 6/10; 7/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, mod
el__n_jobs=-1;, score=(train=0.828, test=0.785) total time=  25.4s
[CV 2/10; 9/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=10, model__min_samples_split=8, model__n_estimators=250,
model__n_jobs=-1
[CV 2/10; 9/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=10, model__min_samples_split=8, model__n_estimators=250, mo
del__n_jobs=-1;, score=(train=0.958, test=0.852) total time=  42.8s
[CV 8/10; 9/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=10, model__min_samples_split=8, model__n_estimators=250,
model__n_jobs=-1
[CV 8/10; 9/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=10, model__min_samples_split=8, model__n_estimators=250, mo
del__n_jobs=-1;, score=(train=0.953, test=0.828) total time=  44.8s
[CV 4/10; 10/10] START model__class_weight=balanced, model__criterion=gini,
model__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, m
odel__n_jobs=-1
[CV 4/10; 10/10] END model__class_weight=balanced, model__criterion=gini, mo
del__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, mod
el__n_jobs=-1;, score=(train=0.992, test=0.846) total time= 2.2min
[CV 8/10; 1/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.7
[CV 8/10; 1/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.7;, score=(train=0.996, test=0.882) total time=
52.3s
[CV 4/10; 4/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=1
[CV 4/10; 4/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=1;, score=(train=0.995, test=0.884) total time= 1.0
min
[CV 2/10; 6/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=1
[CV 2/10; 6/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=1;, score=(train=0.813, test=0.811) total time=  10
.9s
[CV 6/10; 6/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=1
[CV 6/10; 6/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=1;, score=(train=0.960, test=0.848) total time=  29
```

```
.4s
[CV 2/10; 7/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=9, model__subsample=0.9
[CV 2/10; 7/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=9, model__subsample=0.9;, score=(train=0.847, test=0.817) total time=
13.6s
[CV 6/10; 7/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=9, model__subsample=0.9
[CV 6/10; 7/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=9, model__subsample=0.9;, score=(train=0.997, test=0.869) total time=
52.3s
[CV 9/10; 8/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=8, model__subsample=0.9
[CV 9/10; 8/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=8, model__subsample=0.9;, score=(train=0.825, test=0.781) total time=
14.2s
[CV 5/10; 9/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=0.9
[CV 5/10; 9/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=0.9;, score=(train=0.990, test=0.895) total time=
52.9s
[CV 6/10; 10/10] START model__gamma=0.4, model__learning_rate=0.01, model__m
ax_depth=7, model__subsample=0.9
[CV 6/10; 10/10] END model__gamma=0.4, model__learning_rate=0.01, model__max
_depth=7, model__subsample=0.9;, score=(train=0.853, test=0.777) total time=
15.9s
[CV 2/10; 2/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 2/10; 2/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.997, test=0.865) total time=  46.8s
[CV 2/10; 4/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 2/10; 4/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.995, test=0.865) total time= 1.8min
[CV 2/10; 5/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=8, model__n_estimators=250, mod
el__n_jobs=-1
[CV 2/10; 5/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=8, model__n_estimators=250, model
__n_jobs=-1;, score=(train=0.997, test=0.867) total time=  46.7s
[CV 5/10; 6/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mo
del__n_jobs=-1
[CV 5/10; 6/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mode
l__n_jobs=-1;, score=(train=0.955, test=0.852) total time= 1.2min
[CV 3/10; 8/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=750,
model__n_jobs=-1
[CV 3/10; 8/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=750, mo
del__n_jobs=-1;, score=(train=0.995, test=0.827) total time= 2.4min
[CV 8/10; 10/10] START model__class_weight=balanced, model__criterion=gini,
```

```
model__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, m
odel__n_jobs=-1
[CV 8/10; 10/10] END model__class_weight=balanced, model__criterion=gini, mo
del__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, mod
el__n_jobs=-1;, score=(train=0.992, test=0.836) total time= 1.5min
[CV 3/10; 1/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.7
[CV 3/10; 1/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.7;, score=(train=1.000, test=0.876) total time= 1
.5min
[CV 7/10; 5/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.9
[CV 7/10; 5/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.9;, score=(train=0.994, test=0.859) total time=
49.8s
[CV 10/10; 6/10] START model__gamma=0.6, model__learning_rate=0.1, model__ma
x_depth=7, model__subsample=1
[CV 10/10; 6/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_
depth=7, model__subsample=1;, score=(train=0.806, test=0.784) total time=  1
0.4s
[CV 1/10; 7/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=9, model__subsample=0.9
[CV 1/10; 7/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=9, model__subsample=0.9;, score=(train=0.997, test=0.883) total time=
56.5s
[CV 4/10; 8/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=8, model__subsample=0.9
[CV 4/10; 8/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=8, model__subsample=0.9;, score=(train=0.884, test=0.789) total time=
39.9s
[CV 8/10; 9/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=0.9
[CV 8/10; 9/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=0.9;, score=(train=0.983, test=0.873) total time=
45.3s
[CV 9/10; 10/10] START model__gamma=0.4, model__learning_rate=0.01, model__m
ax_depth=7, model__subsample=0.9
[CV 9/10; 10/10] END model__gamma=0.4, model__learning_rate=0.01, model__max
_depth=7, model__subsample=0.9;, score=(train=0.815, test=0.779) total time=
11.5s
[CV 4/10; 2/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 4/10; 2/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.998, test=0.855) total time=  46.6s
[CV 8/10; 3/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=20, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 8/10; 3/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=20, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.999, test=0.850) total time= 1.8min
[CV 6/10; 5/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=8, model__n_estimators=250, mod
el__n_jobs=-1
[CV 6/10; 5/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=8, model__n_estimators=250, model
```

```
__n_jobs=-1;, score=(train=0.997, test=0.839) total time=  47.1s
[CV 8/10; 6/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mo
del__n_jobs=-1
[CV 8/10; 6/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mode
l__n_jobs=-1;, score=(train=0.956, test=0.831) total time= 1.2min
[CV 8/10; 8/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=750,
model__n_jobs=-1
[CV 8/10; 8/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=750, mo
del__n_jobs=-1;, score=(train=0.995, test=0.836) total time= 2.4min
[CV 9/10; 1/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.7
[CV 9/10; 1/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.7;, score=(train=0.997, test=0.877) total time=
58.0s
[CV 5/10; 4/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=1
[CV 5/10; 4/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=1;, score=(train=0.872, test=0.841) total time=  14
.5s
[CV 3/10; 5/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.9
[CV 3/10; 5/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.9;, score=(train=1.000, test=0.880) total time= 1
.5min
[CV 5/10; 7/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=9, model__subsample=0.9
[CV 5/10; 7/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=9, model__subsample=0.9;, score=(train=1.000, test=0.903) total time= 1
.4min
[CV 9/10; 9/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=0.9
[CV 9/10; 9/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=0.9;, score=(train=0.985, test=0.876) total time=
46.0s
[CV 10/10; 10/10] START model__gamma=0.4, model__learning_rate=0.01, model__
max_depth=7, model__subsample=0.9
[CV 10/10; 10/10] END model__gamma=0.4, model__learning_rate=0.01, model__ma
x_depth=7, model__subsample=0.9;, score=(train=0.831, test=0.806) total time
=  10.5s
[CV 3/10; 1/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 3/10; 1/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.990, test=0.829) total time=  43.7s
[CV 8/10; 2/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 8/10; 2/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.997, test=0.840) total time=  44.4s
[CV 5/10; 4/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=500,
```

```
model__n_jobs=-1
[CV 5/10; 4/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.995, test=0.857) total time= 1.7min
[CV 10/10; 5/10] START model__class_weight=balanced, model__criterion=gini,
model__max_depth=20, model__min_samples_split=8, model__n_estimators=250, mo
del__n_jobs=-1
[CV 10/10; 5/10] END model__class_weight=balanced, model__criterion=gini, mo
del__max_depth=20, model__min_samples_split=8, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.997, test=0.864) total time=  46.5s
[CV 2/10; 7/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, m
odel__n_jobs=-1
[CV 2/10; 7/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, mod
el__n_jobs=-1;, score=(train=0.828, test=0.815) total time=  28.1s
[CV 9/10; 7/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, m
odel__n_jobs=-1
[CV 9/10; 7/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, mod
el__n_jobs=-1;, score=(train=0.824, test=0.822) total time=  20.7s
[CV 5/10; 9/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=10, model__min_samples_split=8, model__n_estimators=250,
model__n_jobs=-1
[CV 5/10; 9/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=10, model__min_samples_split=8, model__n_estimators=250, mo
del__n_jobs=-1;, score=(train=0.954, test=0.849) total time=  45.9s
[CV 1/10; 10/10] START model__class_weight=balanced, model__criterion=gini,
model__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, m
odel__n_jobs=-1
[CV 1/10; 10/10] END model__class_weight=balanced, model__criterion=gini, mo
del__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, mod
el__n_jobs=-1;, score=(train=0.992, test=0.833) total time= 2.4min
[CV 6/10; 1/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.7
[CV 6/10; 1/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.7;, score=(train=0.968, test=0.855) total time=
26.5s
[CV 5/10; 3/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=9, model__subsample=1
[CV 5/10; 3/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=9, model__subsample=1;, score=(train=0.890, test=0.853) total time=  4
5.1s
[CV 2/10; 5/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.9
[CV 2/10; 5/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.9;, score=(train=0.970, test=0.877) total time=
28.6s
[CV 9/10; 5/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.9
[CV 9/10; 5/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.9;, score=(train=0.999, test=0.886) total time= 1
.3min
[CV 9/10; 7/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=9, model__subsample=0.9
[CV 9/10; 7/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
```

```
epth=9, model__subsample=0.9;, score=(train=1.000, test=0.879) total time= 1
.9min
[CV 8/10; 10/10] START model__gamma=0.4, model__learning_rate=0.01, model__m
ax_depth=7, model__subsample=0.9
[CV 8/10; 10/10] END model__gamma=0.4, model__learning_rate=0.01, model__max
_depth=7, model__subsample=0.9;, score=(train=0.856, test=0.799) total time=
19.8s
[CV 3/10; 2/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 3/10; 2/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.997, test=0.833) total time=  46.6s
[CV 10/10; 3/10] START model__class_weight=balanced, model__criterion=entrop
y, model__max_depth=20, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 10/10; 3/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=20, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.999, test=0.867) total time= 1.8min
[CV 7/10; 5/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=8, model__n_estimators=250, mod
el__n_jobs=-1
[CV 7/10; 5/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=8, model__n_estimators=250, model
__n_jobs=-1;, score=(train=0.997, test=0.830) total time=  46.9s
[CV 9/10; 6/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mo
del__n_jobs=-1
[CV 9/10; 6/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mode
l__n_jobs=-1;, score=(train=0.954, test=0.842) total time= 1.2min
[CV 1/10; 9/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=10, model__min_samples_split=8, model__n_estimators=250,
model__n_jobs=-1
[CV 1/10; 9/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=10, model__min_samples_split=8, model__n_estimators=250, mo
del__n_jobs=-1;, score=(train=0.952, test=0.816) total time=  40.2s
[CV 7/10; 9/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=10, model__min_samples_split=8, model__n_estimators=250,
model__n_jobs=-1
[CV 7/10; 9/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=10, model__min_samples_split=8, model__n_estimators=250, mo
del__n_jobs=-1;, score=(train=0.954, test=0.809) total time=  44.5s
[CV 3/10; 10/10] START model__class_weight=balanced, model__criterion=gini,
model__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, m
odel__n_jobs=-1
[CV 3/10; 10/10] END model__class_weight=balanced, model__criterion=gini, mo
del__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, mod
el__n_jobs=-1;, score=(train=0.992, test=0.823) total time= 2.2min
[CV 10/10; 1/10] START model__gamma=0.4, model__learning_rate=0.1, model__ma
x_depth=8, model__subsample=0.7
[CV 10/10; 1/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_
depth=8, model__subsample=0.7;, score=(train=0.828, test=0.794) total time=
10.7s
[CV 8/10; 2/10] START model__gamma=0.1, model__learning_rate=0.01, model__ma
x_depth=7, model__subsample=1
[CV 8/10; 2/10] END model__gamma=0.1, model__learning_rate=0.01, model__max_
```

```
depth=7, model__subsample=1;, score=(train=0.854, test=0.797) total time=  3
2.3s
[CV 9/10; 3/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=9, model__subsample=1
[CV 9/10; 3/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=9, model__subsample=1;, score=(train=0.819, test=0.759) total time=  1
4.8s
[CV 6/10; 4/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=1
[CV 6/10; 4/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=1;, score=(train=0.983, test=0.859) total time=  36
.8s
[CV 8/10; 5/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.9
[CV 8/10; 5/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.9;, score=(train=0.997, test=0.879) total time= 1
.0min
[CV 3/10; 7/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=9, model__subsample=0.9
[CV 3/10; 7/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=9, model__subsample=0.9;, score=(train=1.000, test=0.877) total time= 1
.3min
[CV 2/10; 9/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=0.9
[CV 2/10; 9/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=0.9;, score=(train=0.989, test=0.894) total time=
51.7s
[CV 4/10; 10/10] START model__gamma=0.4, model__learning_rate=0.01, model__m
ax_depth=7, model__subsample=0.9
[CV 4/10; 10/10] END model__gamma=0.4, model__learning_rate=0.01, model__max
_depth=7, model__subsample=0.9;, score=(train=0.869, test=0.787) total time=
28.9s
[CV 1/10; 2/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 1/10; 2/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.997, test=0.844) total time=  46.4s
[CV 7/10; 3/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=20, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 7/10; 3/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=20, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.999, test=0.832) total time= 1.8min
[CV 4/10; 5/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=8, model__n_estimators=250, mod
el__n_jobs=-1
[CV 4/10; 5/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=8, model__n_estimators=250, model
__n_jobs=-1;, score=(train=0.998, test=0.850) total time=  46.7s
[CV 6/10; 6/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mo
del__n_jobs=-1
[CV 6/10; 6/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mode
l__n_jobs=-1;, score=(train=0.954, test=0.824) total time= 1.2min
[CV 5/10; 8/10] START model__class_weight=balanced, model__criterion=entropy
```

```
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=750,
model__n_jobs=-1
[CV 5/10; 8/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=750, mo
del__n_jobs=-1;, score=(train=0.995, test=0.856) total time= 2.4min
[CV 1/10; 1/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.7
[CV 1/10; 1/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.7;, score=(train=1.000, test=0.889) total time= 1
.4min
[CV 6/10; 5/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.9
[CV 6/10; 5/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.9;, score=(train=0.995, test=0.878) total time=
55.2s
[CV 9/10; 6/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=1
[CV 9/10; 6/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=1;, score=(train=1.000, test=0.885) total time= 2.2
min
[CV 2/10; 10/10] START model__gamma=0.4, model__learning_rate=0.01, model__m
ax_depth=7, model__subsample=0.9
[CV 2/10; 10/10] END model__gamma=0.4, model__learning_rate=0.01, model__max
_depth=7, model__subsample=0.9;, score=(train=0.828, test=0.816) total time=
13.8s
[CV 5/10; 10/10] START model__gamma=0.4, model__learning_rate=0.01, model__m
ax_depth=7, model__subsample=0.9
[CV 5/10; 10/10] END model__gamma=0.4, model__learning_rate=0.01, model__max
_depth=7, model__subsample=0.9;, score=(train=0.870, test=0.856) total time=
31.6s
[CV 5/10; 1/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 5/10; 1/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.991, test=0.860) total time=  44.2s
[CV 10/10; 2/10] START model__class_weight=balanced, model__criterion=gini,
model__max_depth=20, model__min_samples_split=10, model__n_estimators=250, m
odel__n_jobs=-1
[CV 10/10; 2/10] END model__class_weight=balanced, model__criterion=gini, mo
del__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mod
el__n_jobs=-1;, score=(train=0.997, test=0.865) total time=  44.7s
[CV 6/10; 4/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 6/10; 4/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.994, test=0.834) total time= 1.7min
[CV 2/10; 6/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mo
del__n_jobs=-1
[CV 2/10; 6/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mode
l__n_jobs=-1;, score=(train=0.957, test=0.853) total time= 1.2min
[CV 10/10; 7/10] START model__class_weight=balanced, model__criterion=entrop
y, model__max_depth=5, model__min_samples_split=4, model__n_estimators=250,
model__n_jobs=-1
```

```
[CV 10/10; 7/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, mod
el__n_jobs=-1;, score=(train=0.825, test=0.794) total time=  20.8s
[CV 6/10; 9/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=10, model__min_samples_split=8, model__n_estimators=250,
model__n_jobs=-1
[CV 6/10; 9/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=10, model__min_samples_split=8, model__n_estimators=250, mo
del__n_jobs=-1;, score=(train=0.953, test=0.822) total time=  46.0s
[CV 2/10; 10/10] START model__class_weight=balanced, model__criterion=gini,
model__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, m
odel__n_jobs=-1
[CV 2/10; 10/10] END model__class_weight=balanced, model__criterion=gini, mo
del__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, mod
el__n_jobs=-1;, score=(train=0.993, test=0.859) total time= 2.4min
[CV 2/10; 2/10] START model__gamma=0.1, model__learning_rate=0.01, model__ma
x_depth=7, model__subsample=1
[CV 2/10; 2/10] END model__gamma=0.1, model__learning_rate=0.01, model__max_
depth=7, model__subsample=1;, score=(train=0.778, test=0.774) total time=  1
0.5s
[CV 7/10; 2/10] START model__gamma=0.1, model__learning_rate=0.01, model__ma
x_depth=7, model__subsample=1
[CV 7/10; 2/10] END model__gamma=0.1, model__learning_rate=0.01, model__max_
depth=7, model__subsample=1;, score=(train=0.799, test=0.719) total time=  1
0.1s
[CV 2/10; 3/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=9, model__subsample=1
[CV 2/10; 3/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=9, model__subsample=1;, score=(train=0.793, test=0.779) total time=  1
3.7s
[CV 6/10; 3/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=9, model__subsample=1
[CV 6/10; 3/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=9, model__subsample=1;, score=(train=0.917, test=0.802) total time= 1.
2min
[CV 1/10; 6/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=1
[CV 1/10; 6/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=1;, score=(train=0.996, test=0.890) total time= 1.3
min
[CV 10/10; 7/10] START model__gamma=0.4, model__learning_rate=0.1, model__ma
x_depth=9, model__subsample=0.9
[CV 10/10; 7/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_
depth=9, model__subsample=0.9;, score=(train=1.000, test=0.904) total time=
2.2min
[CV 4/10; 1/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 4/10; 1/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.991, test=0.844) total time=  44.7s
[CV 2/10; 3/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=20, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 2/10; 3/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=20, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.999, test=0.873) total time= 1.8min
```

```
[CV 5/10; 5/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=8, model__n_estimators=250, mod
el__n_jobs=-1
[CV 5/10; 5/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=8, model__n_estimators=250, model
__n_jobs=-1;, score=(train=0.997, test=0.871) total time=  46.5s
[CV 7/10; 6/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mo
del__n_jobs=-1
[CV 7/10; 6/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mode
l__n_jobs=-1;, score=(train=0.956, test=0.811) total time= 1.2min
[CV 6/10; 8/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=750,
model__n_jobs=-1
[CV 6/10; 8/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=750, mo
del__n_jobs=-1;, score=(train=0.994, test=0.835) total time= 2.4min
[CV 6/10; 2/10] START model__gamma=0.1, model__learning_rate=0.01, model__ma
x_depth=7, model__subsample=1
[CV 6/10; 2/10] END model__gamma=0.1, model__learning_rate=0.01, model__max_
depth=7, model__subsample=1;, score=(train=0.842, test=0.768) total time=  1
8.0s
[CV 1/10; 3/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=9, model__subsample=1
[CV 1/10; 3/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=9, model__subsample=1;, score=(train=0.852, test=0.782) total time=  1
8.9s
[CV 7/10; 3/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=9, model__subsample=1
[CV 7/10; 3/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=9, model__subsample=1;, score=(train=0.820, test=0.730) total time=  1
4.9s
[CV 3/10; 4/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=1
[CV 3/10; 4/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=1;, score=(train=1.000, test=0.881) total time= 2.6
min
[CV 5/10; 8/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=8, model__subsample=0.9
[CV 5/10; 8/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=8, model__subsample=0.9;, score=(train=0.859, test=0.850) total time=
25.4s
[CV 3/10; 9/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=0.9
[CV 3/10; 9/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=0.9;, score=(train=1.000, test=0.883) total time= 1
.6min
[CV 6/10; 2/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 6/10; 2/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.997, test=0.839) total time=  46.6s
[CV 1/10; 4/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
```

```
[CV 1/10; 4/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.994, test=0.834) total time= 1.7min
[CV 7/10; 4/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 7/10; 4/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.995, test=0.826) total time= 1.6min
[CV 3/10; 7/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, m
odel__n_jobs=-1
[CV 3/10; 7/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, mod
el__n_jobs=-1;, score=(train=0.826, test=0.785) total time=  26.3s
[CV 1/10; 8/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=750,
model__n_jobs=-1
[CV 1/10; 8/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=750, mo
del__n_jobs=-1;, score=(train=0.994, test=0.836) total time= 2.4min
[CV 7/10; 10/10] START model__class_weight=balanced, model__criterion=gini,
model__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, m
odel__n_jobs=-1
[CV 7/10; 10/10] END model__class_weight=balanced, model__criterion=gini, mo
del__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, mod
el__n_jobs=-1;, score=(train=0.992, test=0.824) total time= 1.6min
[CV 7/10; 1/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.7
[CV 7/10; 1/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.7;, score=(train=0.995, test=0.859) total time=
48.7s
[CV 2/10; 4/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=1
[CV 2/10; 4/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=1;, score=(train=0.916, test=0.853) total time=  20
.0s
[CV 10/10; 4/10] START model__gamma=0.4, model__learning_rate=0.1, model__ma
x_depth=8, model__subsample=1
[CV 10/10; 4/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_
depth=8, model__subsample=1;, score=(train=0.817, test=0.782) total time=  1
1.8s
[CV 4/10; 5/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=0.9
[CV 4/10; 5/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=0.9;, score=(train=0.993, test=0.874) total time=
50.2s
[CV 7/10; 6/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=1
[CV 7/10; 6/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=1;, score=(train=0.980, test=0.846) total time=  40
.2s
[CV 7/10; 7/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=9, model__subsample=0.9
[CV 7/10; 7/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=9, model__subsample=0.9;, score=(train=0.999, test=0.866) total time= 1
.2min
```

```
[CV 7/10; 9/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=0.9
[CV 7/10; 9/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=0.9;, score=(train=0.797, test=0.722) total time=
9.9s
[CV 10/10; 9/10] START model__gamma=0.6, model__learning_rate=0.1, model__ma
x_depth=7, model__subsample=0.9
[CV 10/10; 9/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_
depth=7, model__subsample=0.9;, score=(train=1.000, test=0.904) total time=
1.6min
[CV 2/10; 1/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 2/10; 1/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.992, test=0.861) total time=  44.0s
[CV 9/10; 2/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 9/10; 2/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.997, test=0.867) total time=  44.4s
[CV 4/10; 4/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 4/10; 4/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.995, test=0.849) total time= 1.7min
[CV 1/10; 6/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mo
del__n_jobs=-1
[CV 1/10; 6/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mode
l__n_jobs=-1;, score=(train=0.953, test=0.822) total time= 1.2min
[CV 7/10; 7/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, m
odel__n_jobs=-1
[CV 7/10; 7/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, mod
el__n_jobs=-1;, score=(train=0.828, test=0.772) total time=  20.6s
[CV 4/10; 9/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=10, model__min_samples_split=8, model__n_estimators=250,
model__n_jobs=-1
[CV 4/10; 9/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=10, model__min_samples_split=8, model__n_estimators=250, mo
del__n_jobs=-1;, score=(train=0.954, test=0.829) total time=  46.1s
[CV 10/10; 9/10] START model__class_weight=balanced, model__criterion=entrop
y, model__max_depth=10, model__min_samples_split=8, model__n_estimators=250,
model__n_jobs=-1
[CV 10/10; 9/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=10, model__min_samples_split=8, model__n_estimators=250, mo
del__n_jobs=-1;, score=(train=0.951, test=0.842) total time=  44.2s
[CV 6/10; 10/10] START model__class_weight=balanced, model__criterion=gini,
model__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, m
odel__n_jobs=-1
[CV 6/10; 10/10] END model__class_weight=balanced, model__criterion=gini, mo
del__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, mod
```

```
el__n_jobs=-1;, score=(train=0.991, test=0.829) total time= 2.1min
[CV 3/10; 2/10] START model__gamma=0.1, model__learning_rate=0.01, model__ma
x_depth=7, model__subsample=1
[CV 3/10; 2/10] END model__gamma=0.1, model__learning_rate=0.01, model__max_
depth=7, model__subsample=1;, score=(train=0.987, test=0.874) total time= 6.
3min
[CV 6/10; 1/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 6/10; 1/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.990, test=0.838) total time=  44.7s
[CV 1/10; 3/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=20, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 1/10; 3/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=20, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.999, test=0.855) total time= 1.7min
[CV 8/10; 4/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 8/10; 4/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.995, test=0.835) total time= 1.6min
[CV 4/10; 7/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, m
odel__n_jobs=-1
[CV 4/10; 7/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, mod
el__n_jobs=-1;, score=(train=0.827, test=0.776) total time=  26.3s
[CV 7/10; 8/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=750,
model__n_jobs=-1
[CV 7/10; 8/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=750, mo
del__n_jobs=-1;, score=(train=0.995, test=0.827) total time= 2.4min
[CV 10/10; 10/10] START model__class_weight=balanced, model__criterion=gini,
model__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, m
odel__n_jobs=-1
[CV 10/10; 10/10] END model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=15, model__min_samples_split=4, model__n_estimators=1000, mo
del__n_jobs=-1;, score=(train=0.992, test=0.854) total time= 1.5min
[CV 5/10; 2/10] START model__gamma=0.1, model__learning_rate=0.01, model__ma
x_depth=7, model__subsample=1
[CV 5/10; 2/10] END model__gamma=0.1, model__learning_rate=0.01, model__max_
depth=7, model__subsample=1;, score=(train=0.843, test=0.839) total time=  2
4.2s
[CV 3/10; 3/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=9, model__subsample=1
[CV 3/10; 3/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=9, model__subsample=1;, score=(train=0.998, test=0.875) total time= 6.
8min
[CV 8/10; 1/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 8/10; 1/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mode
```

```
l__n_jobs=-1;, score=(train=0.991, test=0.837) total time=  45.1s
[CV 3/10; 3/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=20, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 3/10; 3/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=20, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.999, test=0.842) total time= 1.8min
[CV 9/10; 4/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 9/10; 4/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.995, test=0.848) total time= 1.6min
[CV 5/10; 7/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, m
odel__n_jobs=-1
[CV 5/10; 7/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=5, model__min_samples_split=4, model__n_estimators=250, mod
el__n_jobs=-1;, score=(train=0.822, test=0.832) total time=  25.9s
[CV 9/10; 8/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=750,
model__n_jobs=-1
[CV 9/10; 8/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=750, mo
del__n_jobs=-1;, score=(train=0.995, test=0.848) total time= 2.5min
[CV 1/10; 2/10] START model__gamma=0.1, model__learning_rate=0.01, model__ma
x_depth=7, model__subsample=1
[CV 1/10; 2/10] END model__gamma=0.1, model__learning_rate=0.01, model__max_
depth=7, model__subsample=1;, score=(train=0.867, test=0.796) total time=  4
8.4s
[CV 1/10; 4/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=1
[CV 1/10; 4/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=1;, score=(train=0.997, test=0.877) total time= 1.2
min
[CV 4/10; 6/10] START model__gamma=0.6, model__learning_rate=0.1, model__max
_depth=7, model__subsample=1
[CV 4/10; 6/10] END model__gamma=0.6, model__learning_rate=0.1, model__max_d
epth=7, model__subsample=1;, score=(train=0.997, test=0.890) total time= 1.4
min
[CV 3/10; 8/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=8, model__subsample=0.9
[CV 3/10; 8/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=8, model__subsample=0.9;, score=(train=0.995, test=0.876) total time=
4.3min
[CV 7/10; 1/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mo
del__n_jobs=-1
[CV 7/10; 1/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=15, model__min_samples_split=10, model__n_estimators=250, mode
l__n_jobs=-1;, score=(train=0.991, test=0.826) total time=  45.2s
[CV 5/10; 3/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=20, model__min_samples_split=8, model__n_estimators=500,
model__n_jobs=-1
[CV 5/10; 3/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=20, model__min_samples_split=8, model__n_estimators=500, mo
del__n_jobs=-1;, score=(train=0.999, test=0.870) total time= 1.8min
```

```
[CV 1/10; 5/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=20, model__min_samples_split=8, model__n_estimators=250, mod
el__n_jobs=-1
[CV 1/10; 5/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=20, model__min_samples_split=8, model__n_estimators=250, model
__n_jobs=-1;, score=(train=0.997, test=0.844) total time=  45.8s
[CV 3/10; 6/10] START model__class_weight=balanced, model__criterion=gini, m
odel__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mo
del__n_jobs=-1
[CV 3/10; 6/10] END model__class_weight=balanced, model__criterion=gini, mod
el__max_depth=10, model__min_samples_split=10, model__n_estimators=500, mode
l__n_jobs=-1;, score=(train=0.954, test=0.816) total time= 1.2min
[CV 2/10; 8/10] START model__class_weight=balanced, model__criterion=entropy
, model__max_depth=15, model__min_samples_split=8, model__n_estimators=750,
model__n_jobs=-1
[CV 2/10; 8/10] END model__class_weight=balanced, model__criterion=entropy,
model__max_depth=15, model__min_samples_split=8, model__n_estimators=750, mo
del__n_jobs=-1;, score=(train=0.995, test=0.865) total time= 2.5min
[CV 4/10; 2/10] START model__gamma=0.1, model__learning_rate=0.01, model__ma
x_depth=7, model__subsample=1
[CV 4/10; 2/10] END model__gamma=0.1, model__learning_rate=0.01, model__max_
depth=7, model__subsample=1;, score=(train=0.887, test=0.803) total time= 1.
1min
[CV 9/10; 4/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=8, model__subsample=1
[CV 9/10; 4/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=8, model__subsample=1;, score=(train=1.000, test=0.883) total time= 1.6
min
[CV 4/10; 7/10] START model__gamma=0.4, model__learning_rate=0.1, model__max
_depth=9, model__subsample=0.9
[CV 4/10; 7/10] END model__gamma=0.4, model__learning_rate=0.1, model__max_d
epth=9, model__subsample=0.9;, score=(train=0.997, test=0.874) total time=
54.4s
[CV 8/10; 8/10] START model__gamma=0.6, model__learning_rate=0.01, model__ma
x_depth=8, model__subsample=0.9
[CV 8/10; 8/10] END model__gamma=0.6, model__learning_rate=0.01, model__max_
depth=8, model__subsample=0.9;, score=(train=0.894, test=0.815) total time=
51.2s
[CV 1/10; 10/10] START model__gamma=0.4, model__learning_rate=0.01, model__m
ax_depth=7, model__subsample=0.9
[CV 1/10; 10/10] END model__gamma=0.4, model__learning_rate=0.01, model__max
_depth=7, model__subsample=0.9;, score=(train=0.843, test=0.778) total time=
17.2s
[CV 3/10; 10/10] START model__gamma=0.4, model__learning_rate=0.01, model__m
ax_depth=7, model__subsample=0.9
[CV 3/10; 10/10] END model__gamma=0.4, model__learning_rate=0.01, model__max
_depth=7, model__subsample=0.9;, score=(train=0.991, test=0.877) total time=
3.5min
```