

Case Study 5: Cybersecurity & Firewall Traffic

Matt Farrow

July 3, 2022

1 Introduction

In this study, the goal is to take a [data set](#) of internet traffic records from Firat University in Turkey, and through the use of Support Vector Classifier (SVC) and Stochastic Gradient Descent (SGD) models, make decisions about whether to allow, deny, drop, and reset-both the traffic.

2 Methods

2.1 Data Examination

An initial examination of the data revealed 65,532 observations and 12 features, including 'Action', the multiclass response variable of interest (Table 1).

Feature
Source
Port
Destination Port
NAT Source Port
NAT Destination Port
Action
Bytes
Bytes Sent
Bytes Received
Packets
Elapsed Time (sec)
pkts_sent
pkts_received

Table 1: Feature List

In taking a closer look at the response variable, 'Action', I noticed that the data is heavily imbalanced, with `allow` being the primary action, and `reset-both` being almost non-existent in the data. As a result, that value was dropped from the data set in order to make a small correction to the imbalanced data.

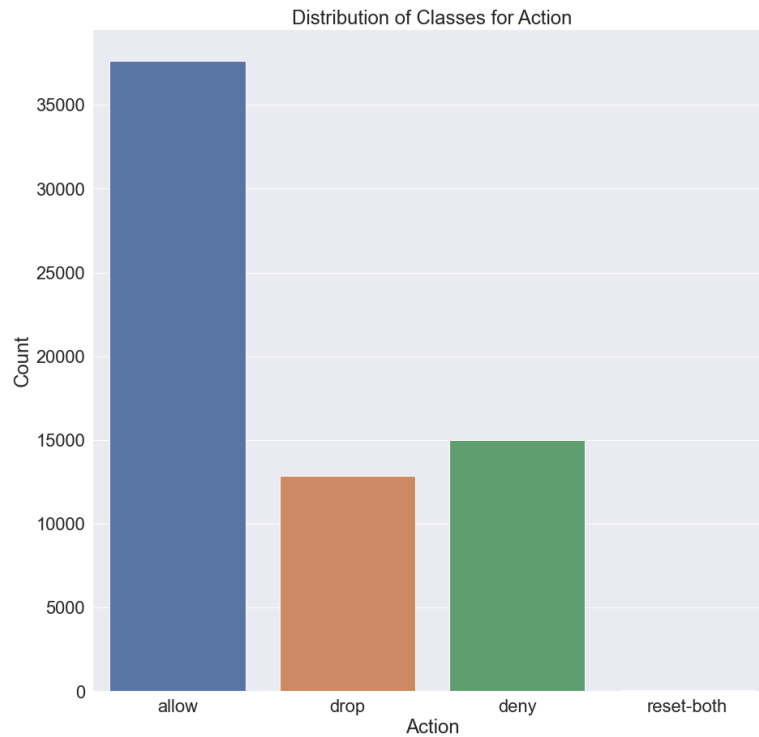


Figure 1: Distribution of Response Variable

The one piece of feature engineering that I did was to create an attribute to note, through either a 0 or 1, whether the source or destination port required network address translation (NAT). I then examined the pairwise correlations of the numeric features in the data. Unsurprisingly, bytes is closely related to packets and pkts_sent, but I ended up deciding to leave those features in the data.

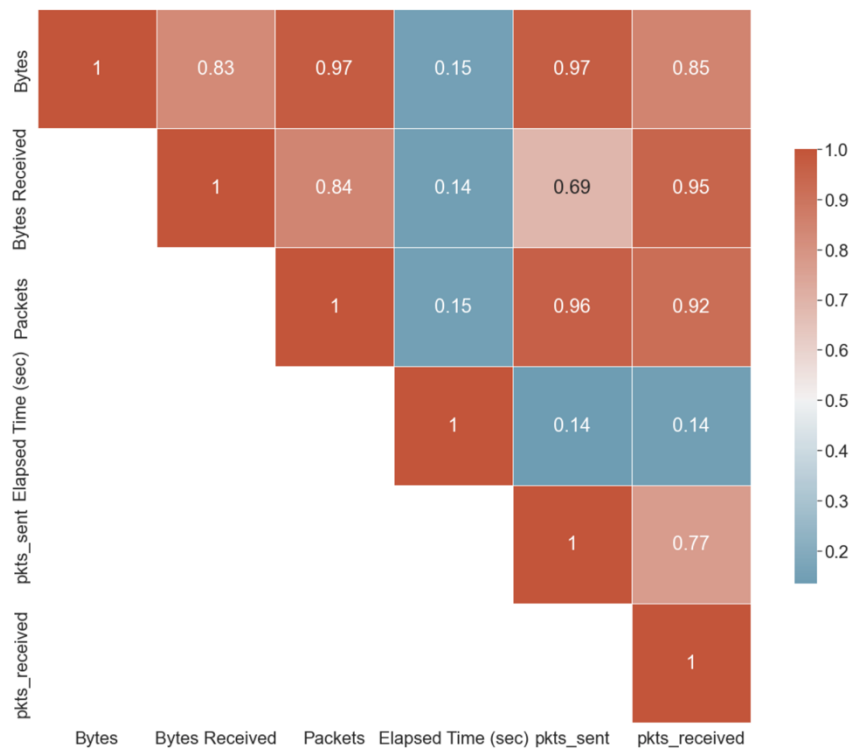


Figure 2: Pairwise Correlation of Numeric Variables

2.2 Model Preparation & Execution

The data was split into test and training data sets using an 80/20 split with a stratified shuffle and columns that needed to be one-hot encoded were identified (Table 2).

Feature
Source Port
Destination Port
NAT Source Port
NAT Destination Port
Source Need NAT
Destination Need NAT

Table 2: Features to One-Hot Encode

A pipeline was built to define the model and handle the one-hot encoding and scaling of the numeric features before running everything through a randomized cross-validation with a hyperparameter search for the SVC (Table 3) and SGD (Table 4) models.

Attribute	Values to Search
C	0.001, 0.01, 0.1, 1, 10, 100
Gamma	'scale', 'auto', 1, 0.1, 0.01, 0.001, 0.0001
Kernel	'linear', 'poly', 'rbf'

Table 3: SVC Hyperparameter Search

Attribute	Values to Search
Alpha	0.000001, 0.00001, 0.0001, 0.001, 0.01
Eta0 (learning rate 'optimal')	0
Eta0 (learning rate 'constant')	0.01, 0.1, 1.0, 5, 10
Class Weight	'balanced'

Table 4: SGD Hyperparameter Search

Additionally, in order to try and visualize the work that the SVC model was doing, the response variable was collapsed to only two potential actions – allow and deny. That allowed me to plot the results and see the impact of the hyperparameter choices, rather than simply visualize them through confusion matrices. In that model, the hyperparameter list was also simplified (Table 5).

Attribute	Values to Search
Kernel	'linear', 'rbf'
C	0.001, 0.01, 0.1, 1, 10, 100

Table 5: SVC Model (Allow/Deny) Hyperparameter Search

3 Results

3.1 Support Vector Classifier Model

The SVC model performed with an extremely high score across all measures (Table 6). Additionally, the confusion matrix (Figure 1) details the small number of errors made in the model with deny being the most frequent incorrect classification (as allow).

	Precision	Recall	F1-Score	Support
allow	0.999334	0.997078	0.998205	7528
deny	0.992027	0.995997	0.994008	2998
drop	0.997287	0.999222	0.998251	2570
accuracy			0.997251	13096
macro avg	0.996214	0.997432	0.996821	13096
weighted avg	0.997259	0.997251	0.997253	13096

Table 6: SVC Model Performance

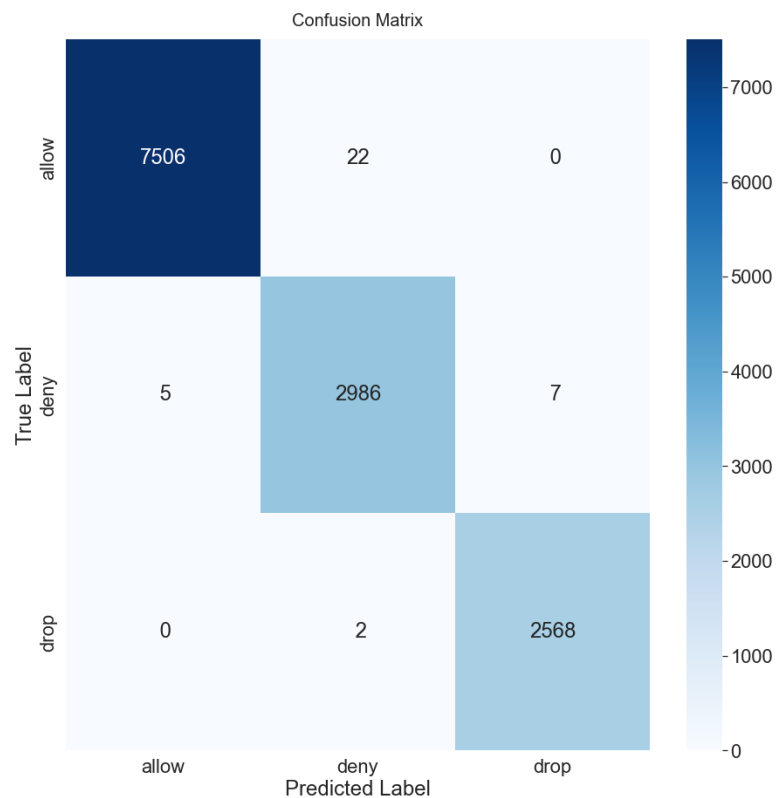


Figure 3: SVC Confusion Matrix

The linear kernel shows an extremely high accuracy score, followed by rbf, and finally poly has a significant drop off in accuracy (Figure 4). Examining the accuracy performance of the 'C' parameter, I note that the performance of all values below 1 is 57.5% accuracy, but at C = 1, the accuracy jumps to 91.9% (Figure 5).

Kernel	Accuracy Score
linear	0.9928
poly	0.5748
rbf	0.9187

Figure 4: SVC Kernel Accuracy

C	Accuracy Score
0.001	0.5748
0.01	0.5748
0.1	0.5748
1	0.9187
10	0.9193
100	0.9497

Figure 5: SVC 'C' Accuracy

3.2 Stochastic Gradient Descent Model

For all intents and purposes, the model performance of the SGD model is almost identical to that of the SVC model (Table 7).

	Precision	Recall	F1-Score	Support
allow	0.999334	0.996148	0.997738	7528
deny	0.990381	0.995997	0.993181	2998
drop	0.997284	1.000000	0.998640	2570
accuracy			0.996869	13096
macro avg	0.995666	0.997382	0.996520	13096
weighted avg	0.996882	0.996869	0.996872	13096

Table 7: SGD Model Performance

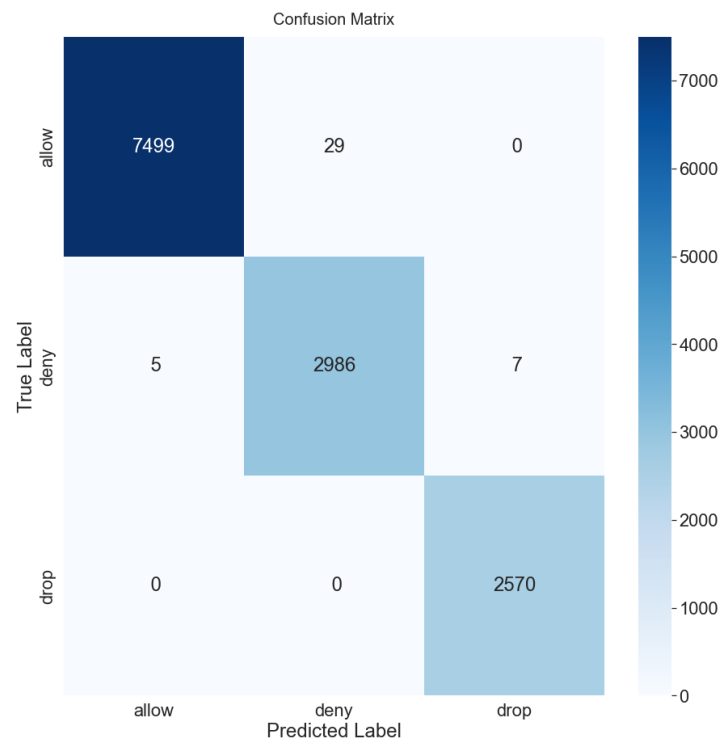


Figure 6: SGD Confusion Matrix

3.3 Two-Class SVC Model

For the two-class SVC model, the training and testing accuracy were both 99.8% and the confusion matrix shows the small number of misclassifications that the model made (Figure 7).

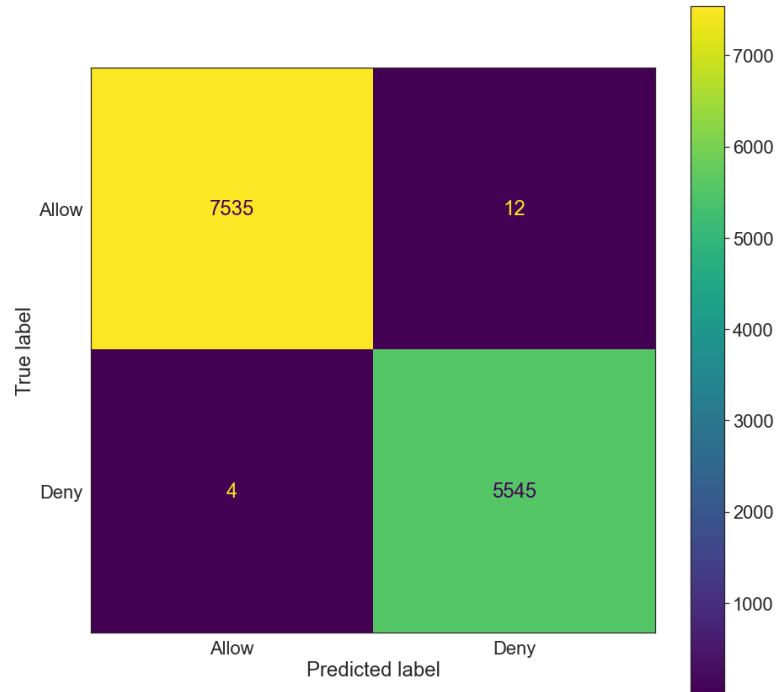


Figure 7: Two-Class SVC Model Confusion Matrix

In terms of kernel performance, the 'linear' kernel returned a 99.9% accuracy whereas the 'rbf' kernel was only able to return 77.9% accuracy. This is able to be visualized in the plots which show the distribution of points and the fitted hyperplane (Figure 8).

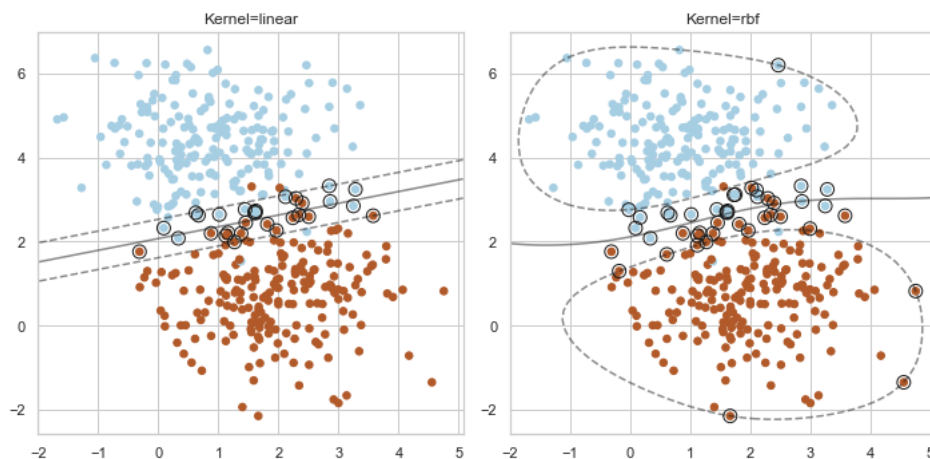


Figure 8: Kernel Hyperplane Performance and Margin

Visualizing the performance of the linear (Figure 9) and rbf (Figure 10) kernels, I was able to see how the different levels of 'C' affected the accuracy results in the original models. When $C < 1$, it is possible to see the points that the model did not correctly capture through the margin and hyperplane.

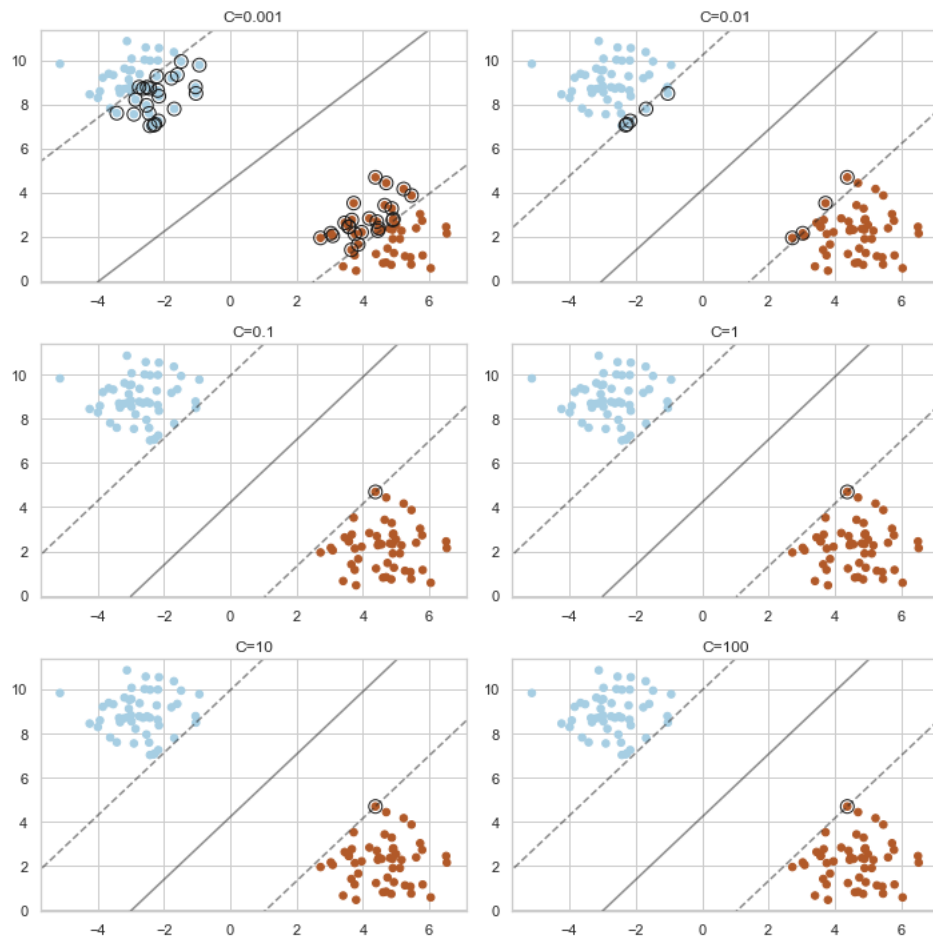


Figure 9: Linear Kernel Performance at Different Levels of 'C'

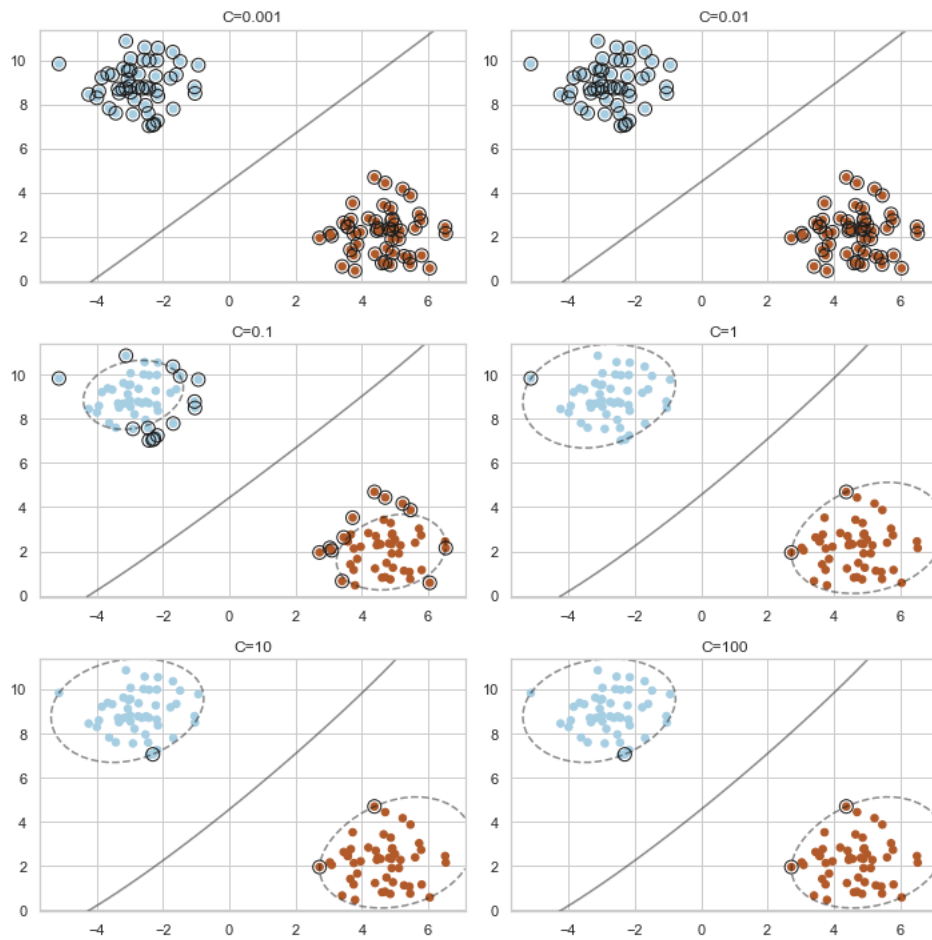


Figure 10: RBF Kernel Performance at Different Levels of 'C'

4 Conclusion

In conclusion, all three models were able to produce incredibly accurate accuracy scores which does give me pause to think that more work should have been done to take multicollinearity into account. In terms of model preference, I'm partial to the two-class SVC model due to the ability to visualize the hyperplane, margin, and model performance.

Appendix

Code

Code begins on the following page.

Case Study 5

Description

Build a classifier to determine whether or not firewall requests should be automated.

[source](#)

Attributes

- Source Port
- Destination Port
- NAT Source Port
- NAT Destination Port
- Action
- Bytes
- Bytes Sent,
- Bytes Received
- Packets
- Elapsed Time (sec)
- pkts_sent
- pkts_received

Response (Action)

- allow
- action
- drop
- reset-both

Class Notes

- Try to do all four classes - but they are "horribly" imbalanced (one has very few samples)
- bytes, packets are ordinal (whole integers)
- Ports are in a sense categorical (needs one hot encoded)
 - Can try to group if desired
- Make sure to do random shuffle when splitting/cross validation
- Column E (allow, not allow, etc) is target variable
- Do test/train split instead of cross validation (takes too long)
- Can make a new features out of pkts_sent and pkts_received (maybe the delta?)

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.compose import ColumnTransformer
from sklearn.datasets import make_blobs
from sklearn.linear_model import SGDClassifier
from sklearn.metrics import accuracy_score
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 f1_score
from sklearn.metrics import plot_confusion_matrix
from sklearn.metrics import plot_roc_curve
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import roc_curve
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import train_test_split
from sklearn.multiclass import OneVsRestClassifier
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import OneHotEncoder
from sklearn.svm import LinearSVC
from sklearn.svm import SVC

# Import other libraries
import os
import warnings
import joblib
from mlxtend.plotting import plot_decision_regions

# 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]: df = pd.read_csv('log2.csv')
```

Exploratory Data Analysis

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

```
Out[3]:
```

	Source Port	Destination Port	NAT Source Port	NAT Destination Port	Action	Bytes	Bytes Sent	Bytes Received	Packets	Elapsed Time (s)
0	57222	53	54587	53	allow	177	94	83	2	
1	56258	3389	56258	3389	allow	4768	1600	3168	19	
2	6881	50321	43265	50321	allow	238	118	120	2	1
3	50553	3389	50553	3389	allow	3327	1438	1889	15	
4	50002	443	45848	443	allow	25358	6778	18580	31	

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

```
Out[4]:
```

	Source Port	Destination Port	NAT Source Port	NAT Destination Port	Action	Bytes	Bytes Sent	Bytes Received	Packet
65527	63691	80	13237	80	allow	314	192	122	
65528	50964	80	13485	80	allow	4680740	67312	4613428	467
65529	54871	445	0	0	drop	70	70	0	
65530	54870	445	0	0	drop	70	70	0	
65531	54867	445	0	0	drop	70	70	0	

```
In [5]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 65532 entries, 0 to 65531
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Source Port                          65532 non-null  int64
1   Destination Port                     65532 non-null  int64
2   NAT Source Port                      65532 non-null  int64
3   NAT Destination Port                65532 non-null  int64
4   Action                              65532 non-null  object
5   Bytes                               65532 non-null  int64
6   Bytes Sent                          65532 non-null  int64
7   Bytes Received                      65532 non-null  int64
8   Packets                             65532 non-null  int64
9   Elapsed Time (sec)                  65532 non-null  int64
10  pkts_sent                           65532 non-null  int64
11  pkts_received                       65532 non-null  int64
dtypes: int64(11), object(1)
memory usage: 6.0+ MB

```

In [6]: `df.shape`

Out[6]: (65532, 12)

In [7]: `df.describe()`

Out[7]:

	Source Port	Destination Port	NAT Source Port	NAT Destination Port	Bytes	Bytes
count	65532.000000	65532.000000	65532.000000	65532.000000	6.553200e+04	6.553200e+04
mean	49391.969343	10577.385812	19282.972761	2671.049930	9.712395e+04	2.238580e+04
std	15255.712537	18466.027039	21970.689669	9739.162278	5.618439e+06	3.828139e+06
min	0.000000	0.000000	0.000000	0.000000	6.000000e+01	6.000000e+01
25%	49183.000000	80.000000	0.000000	0.000000	6.600000e+01	6.600000e+01
50%	53776.500000	445.000000	8820.500000	53.000000	1.680000e+02	9.000000e+01
75%	58638.000000	15000.000000	38366.250000	443.000000	7.522500e+02	2.100000e+02
max	65534.000000	65535.000000	65535.000000	65535.000000	1.269359e+09	9.484772e+08

In [8]: `# Missing values?`
`df.isnull().sum()`

```
Out[8]: Source Port      0
        Destination Port 0
        NAT Source Port  0
        NAT Destination Port 0
        Action           0
        Bytes            0
        Bytes Sent       0
        Bytes Received    0
        Packets          0
        Elapsed Time (sec) 0
        pkts_sent        0
        pkts_received     0
        dtype: int64
```

```
In [9]: df.nunique()
```

```
Out[9]: Source Port      22724
        Destination Port  3273
        NAT Source Port   29152
        NAT Destination Port 2533
        Action            4
        Bytes            10724
        Bytes Sent        6683
        Bytes Received    8814
        Packets          1116
        Elapsed Time (sec) 915
        pkts_sent        749
        pkts_received     922
        dtype: int64
```

```
In [10]: df['Action'].value_counts()
```

```
Out[10]: allow      37640
        deny       14987
        drop       12851
        reset-both    54
        Name: Action, dtype: int64
```

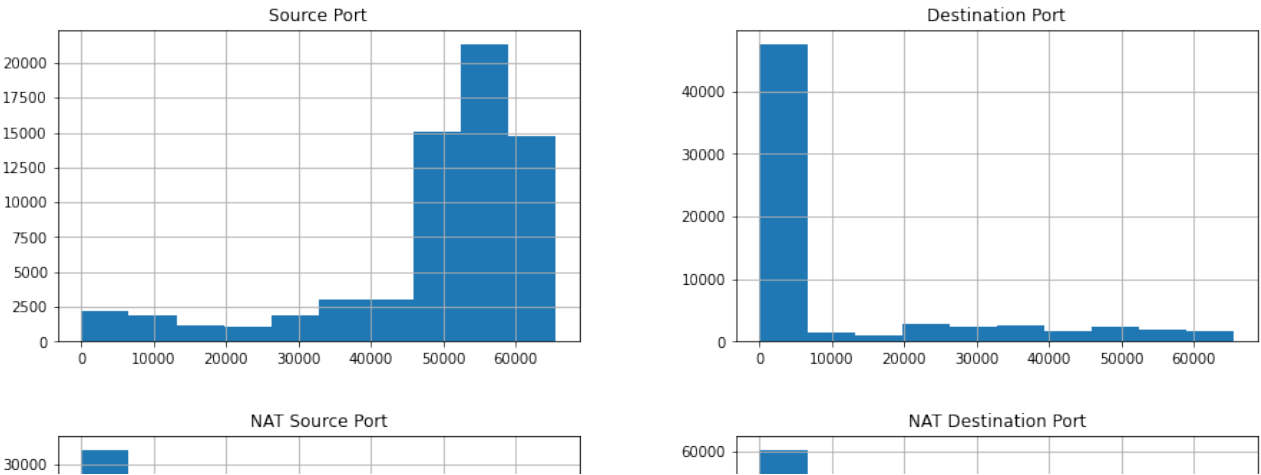
```
In [11]: # Pairwise Correlation
        df.corr()
```

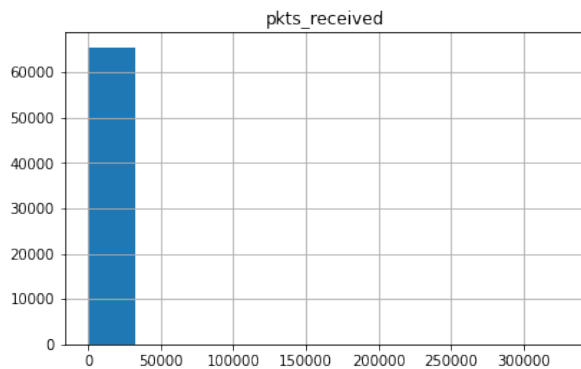
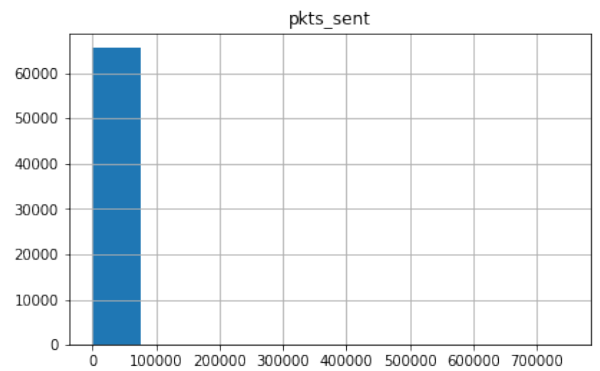
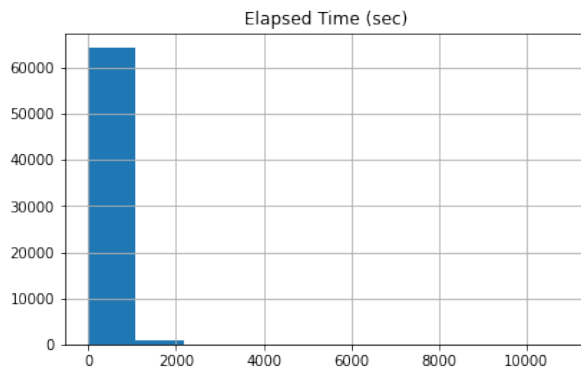
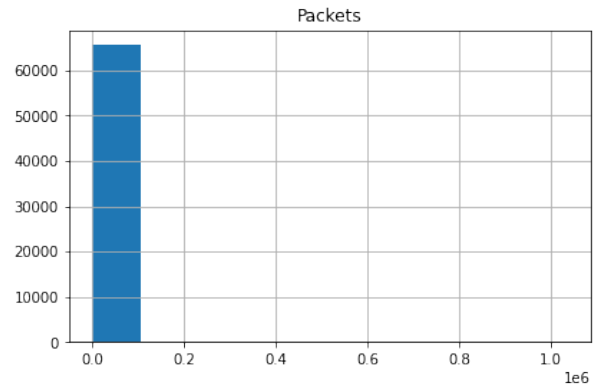
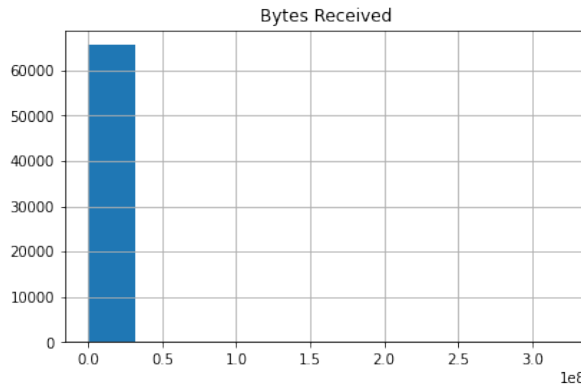
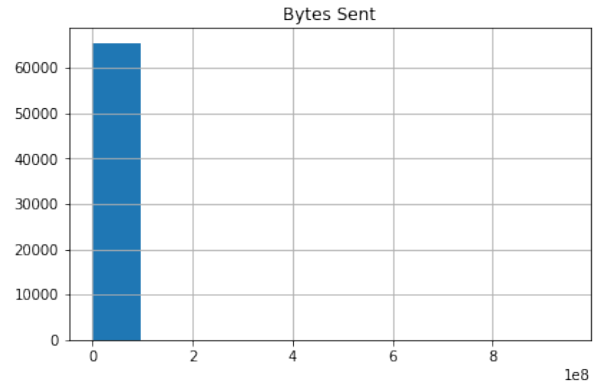
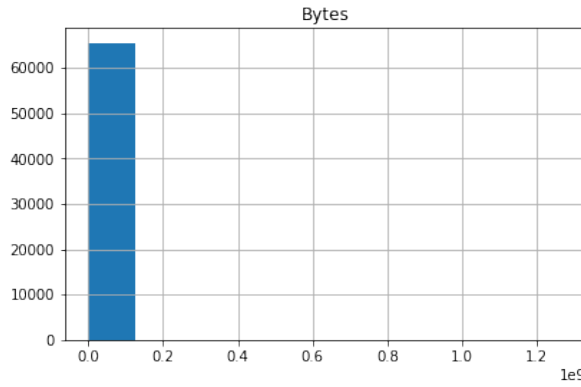
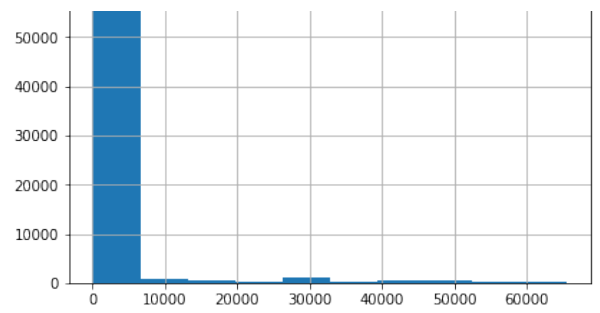
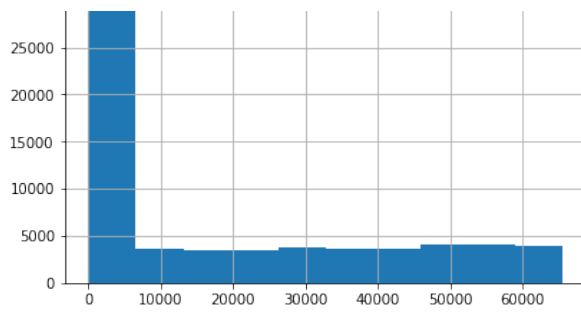
Out[11]:

	Source Port	Destination Port	NAT Source Port	NAT Destination Port	Bytes	Bytes Sent	Byt Receiv
Source Port	1.000000	-0.332246	0.145391	-0.024843	0.000221	-0.000931	0.0019
Destination Port	-0.332246	1.000000	-0.281676	0.410042	-0.005297	0.001675	-0.0146
NAT Source Port	0.145391	-0.281676	1.000000	0.178435	0.010659	0.002242	0.0208
NAT Destination Port	-0.024843	0.410042	0.178435	1.000000	0.003975	0.007904	-0.0032
Bytes	0.000221	-0.005297	0.010659	0.003975	1.000000	0.933462	0.8302
Bytes Sent	-0.000931	0.001675	0.002242	0.007904	0.933462	1.000000	0.5750
Bytes Received	0.001950	-0.014684	0.020827	-0.003216	0.830225	0.575047	1.0000
Packets	-0.001742	-0.006063	0.012633	0.004605	0.974379	0.887596	0.8430
Elapsed Time (sec)	-0.046515	0.023537	0.141485	0.219776	0.148834	0.126039	0.1436
pkts_sent	-0.001422	-0.002134	0.007180	0.006136	0.966548	0.973976	0.6909
pkts_received	-0.001962	-0.010909	0.018772	0.001747	0.850209	0.639098	0.9460

In [12]: df.hist(figsize = (15, 30), layout = (6, 2))

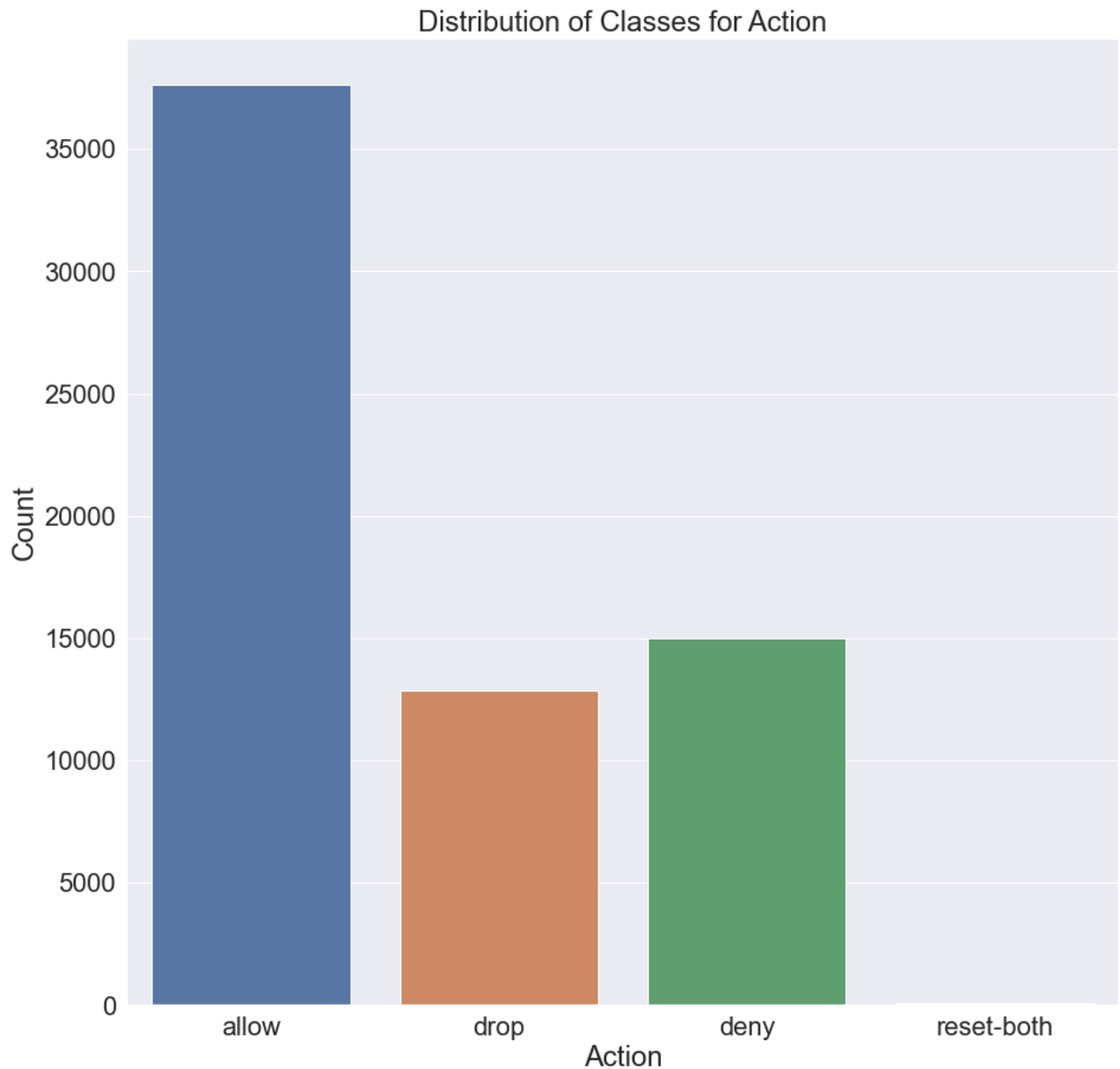
Out[12]: array([[<AxesSubplot:title={ 'center': 'Source Port' }>,
<AxesSubplot:title={ 'center': 'Destination Port' }>],
[<AxesSubplot:title={ 'center': 'NAT Source Port' }>,
<AxesSubplot:title={ 'center': 'NAT Destination Port' }>],
[<AxesSubplot:title={ 'center': 'Bytes' }>,
<AxesSubplot:title={ 'center': 'Bytes Sent' }>],
[<AxesSubplot:title={ 'center': 'Bytes Received' }>,
<AxesSubplot:title={ 'center': 'Packets' }>],
[<AxesSubplot:title={ 'center': 'Elapsed Time (sec)' }>,
<AxesSubplot:title={ 'center': 'pkts_sent' }>],
[<AxesSubplot:title={ 'center': 'pkts_received' }>, <AxesSubplot:>]],
dtype=object)






```
In [13]: sns.set(rc = {'figure.figsize':(15,15)})
sns.set(font_scale = 2)
p = sns.countplot(df['Action'])
p.set_title('Distribution of Classes for Action')
p.set_ylabel('Count')
```

```
Out[13]: Text(0, 0.5, 'Count')
```



```
In [14]: # Drop 'reset-both' since it doesn't even register on the bar graph
df = df.loc[df['Action'] != 'reset-both']
```

```
In [15]: print(f"Source Port Unique Levels: {len(df['Source Port'].unique())}")
print(f"Destination Port Unique Levels: {len(df['Destination Port'].unique())}")
print(f"NAT Source Port Unique Levels: {len(df['NAT Source Port'].unique())}")
print(f"NAT Destination Port Unique Levels: {len(df['NAT Destination Port'].unique())}")
```

```
Source Port Unique Levels: 22692
Destination Port Unique Levels: 3264
NAT Source Port Unique Levels: 29143
NAT Destination Port Unique Levels: 2533
```

```
In [16]: # Does the source or destination port need NAT?
def nat_needed(num: int) -> int:
    if num == 0:
        return num
    else:
        return 1

df['Source Need NAT'] = df['NAT Source Port'].apply(lambda x: nat_needed(x))
df['Destination Need NAT'] = df['NAT Destination Port'].apply(lambda x: nat_

df.head()
```

```
Out[16]:
```

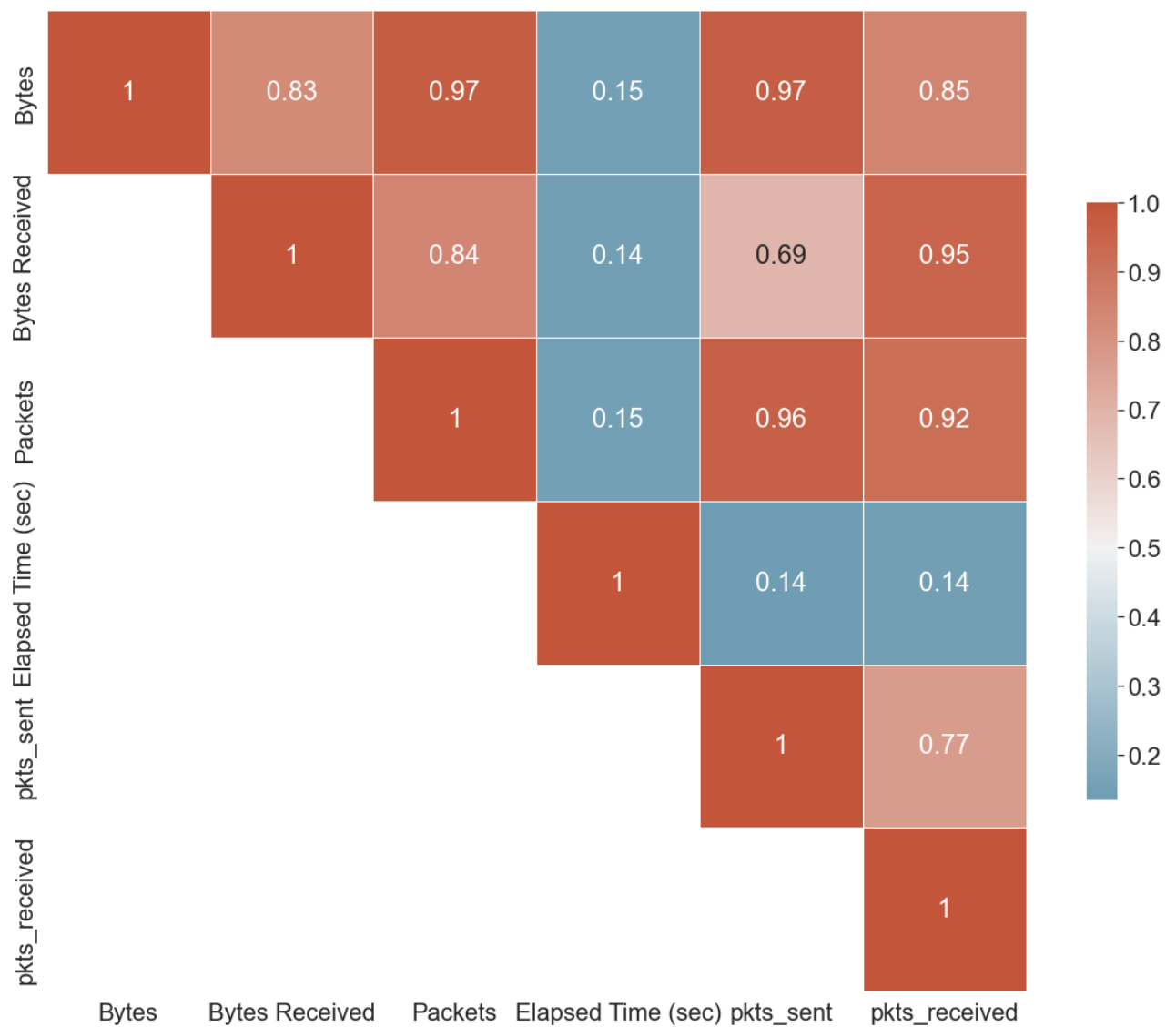
	Source Port	Destination Port	NAT Source Port	NAT Destination Port	Action	Bytes	Bytes Sent	Bytes Received	Packets	Elapsed Time (s)
0	57222	53	54587	53	allow	177	94	83	2	
1	56258	3389	56258	3389	allow	4768	1600	3168	19	
2	6881	50321	43265	50321	allow	238	118	120	2	1
3	50553	3389	50553	3389	allow	3327	1438	1889	15	
4	50002	443	45848	443	allow	25358	6778	18580	31	

```
In [17]: # Define numeric variables
df_numeric = df[['Bytes', 'Bytes Received', 'Packets', 'Elapsed Time (sec)', 'pk
```

```
In [18]: # Check correlation of numeric variables
corr = df_numeric.corr().abs()
upper_tri = corr.where(np.triu(np.ones(corr.shape), k = 0).astype(np.bool))

sns.set(font_scale = 2)
sns.set_style(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 = True,
            square = True,
            linewidths = .5,
            cbar_kws = {"shrink": .5})
```

```
Out[18]: <AxesSubplot:>
```



Test/Train Splits

```
In [19]: x, y = df.drop(['Action'], inplace = False, axis = 1), df['Action']
          x_train, x_test, y_train, y_test = train_test_split(x, y,
                                                             test_size = 0.20,
                                                             shuffle = True,
                                                             random_state = 1,
                                                             stratify = y
                                                             )
```

```
In [20]: # Define columns to one-hot encode
          df_ohe = ['Source Port',
                    'Destination Port',
                    'NAT Source Port',
                    'NAT Destination Port',
                    'Source Need NAT',
                    'Destination Need NAT']

          # Define remaining numeric columns
          df_numeric = [names for names in X.columns if names not in df_ohe]
```

Helper Functions

```
In [21]: # Define cross-validation
cv = StratifiedKFold(n_splits = 10, shuffle = True, random_state = 1)

# Define cross-validation summary
def cv_summary(estimator, columns):
    df = pd.DataFrame(estimator.cv_results_)
    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()
    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

# Define get accuracy score
def get_acc_score(model, x, y):
    """
    Calculates score from estimator (gridsearch) inherit scorer
    """
    return model.score(x, y)

# Define get confusion matrix
def get_confusion_matrix(y, yhat, mat_title="Confusion Matrix"):
    x_axis_labels = ['allow', 'deny', 'drop']
    y_axis_labels = ['allow', 'deny', 'drop']
    cm_n = confusion_matrix(y, yhat)
    ax = sns.heatmap(cm_n, cmap='Blues', annot=True, fmt='2d', xticklabels=x_axis_labels, yticklabels=y_axis_labels)
    ax.set(xlabel='Predicted Label', ylabel='True Label')
    ax.set_title(mat_title, fontdict={'fontsize': 20}, pad=15)

# Define get classification report
def get_classification_report(x_train, y_train, x_test, y_test, pred, model):
    """
    This function is used to get comprehensive classification report for bin
    Training Accuracy, Test Accuracy, print classification_report, plot conf
    """
    print(f"Training Score: {get_acc_score(model, x_train, y_train):.6f}")
    print("Test result:")
    print(classification_report(y_test, pred, digits=6))
    get_confusion_matrix(y_test, pred)
```

Support Vector Classifier Model

```
In [22]: params_svc = {
    "model__C": [0.001, 0.01, 0.1, 1, 10, 100],
    "model__gamma": ["scale", "auto", 1, 0.1, 0.01, 0.001, 0.0001],
    "model__kernel": ["linear", "poly", "rbf"],
    "model__class_weight": ["balanced"]
}
```

```
In [23]: %time

# Define preprocessing for columns
t = [('cat', OneHotEncoder(handle_unknown='ignore'), df_ohe),
      ('scale', MinMaxScaler(), df_numeric)]
col_transform = ColumnTransformer(t)

# Define model
model_svc = SVC()

# Define pipeline
pipeline_svc = Pipeline(steps=[('preprocessing', col_transform), ('model', model_svc)])

# Define model
clf_svc = RandomizedSearchCV(pipeline_svc,
                             params_svc,
                             n_iter = 10,
                             scoring = 'f1_weighted',
                             random_state = 1,
                             verbose = 15,
                             cv = cv,
                             n_jobs = -1,
                             return_train_score = True)
```

CPU times: user 2 μ s, sys: 1e+03 ns, total: 3 μ s
 Wall time: 5.01 μ s

```
In [24]: %time

# Run model
search_svc = clf_svc.fit(X_train, y_train)
```

CPU times: user 1 μ s, sys: 1e+03 ns, total: 2 μ s
 Wall time: 4.77 μ s
 Fitting 10 folds for each of 10 candidates, totalling 100 fits

```
In [25]: df_search_svc_cv_results_columns = ['param_model__kernel',
                                             'param_model__gamma',
                                             'param_model__class_weight',
                                             'param_model__C',
                                             'mean_train_score',
                                             'mean_test_score',
                                             'rank_test_score']

df_search_svc_cv_results = cv_summary(search_svc, df_search_svc_cv_results_columns)
df_search_svc_cv_results
```

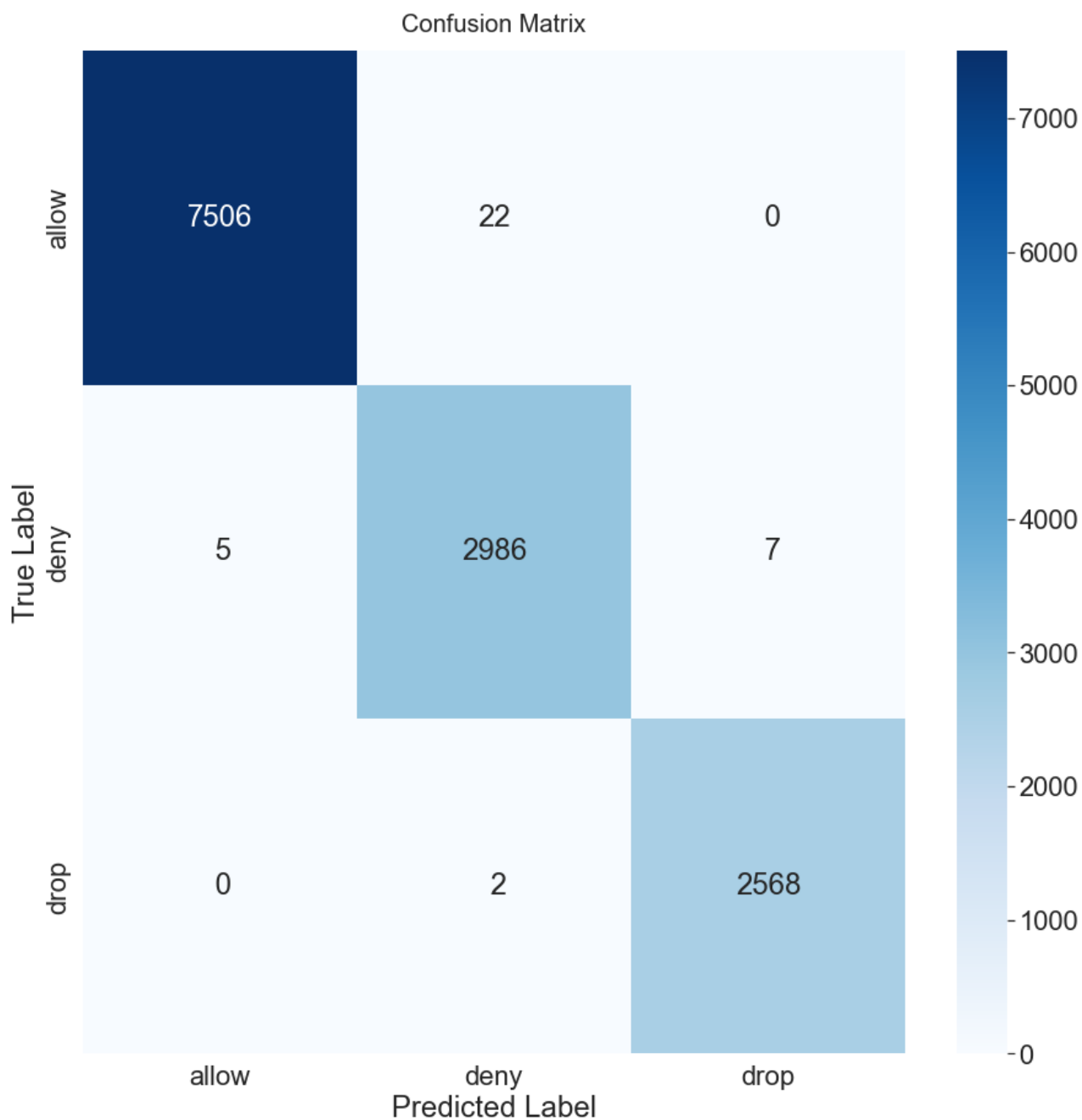
Out [25]:	param_combination	param_model__kernel	param_model__gamma	param_model__class_v
0	1	linear	0.001	ba
1	2	poly	auto	ba
2	3	poly	0.1	ba
3	4	linear	1	ba
4	5	linear	0.0001	ba
5	6	rbf	0.1	ba
6	7	rbf	0.01	ba
7	8	poly	0.0001	ba
8	9	rbf	0.1	ba
9	10	linear	scale	ba

```
In [26]: predict = search_svc.predict(X_test)
get_classification_report(X_train, y_train, X_test, y_test, predict, search_
```

Training Score: 0.998417

Test result:

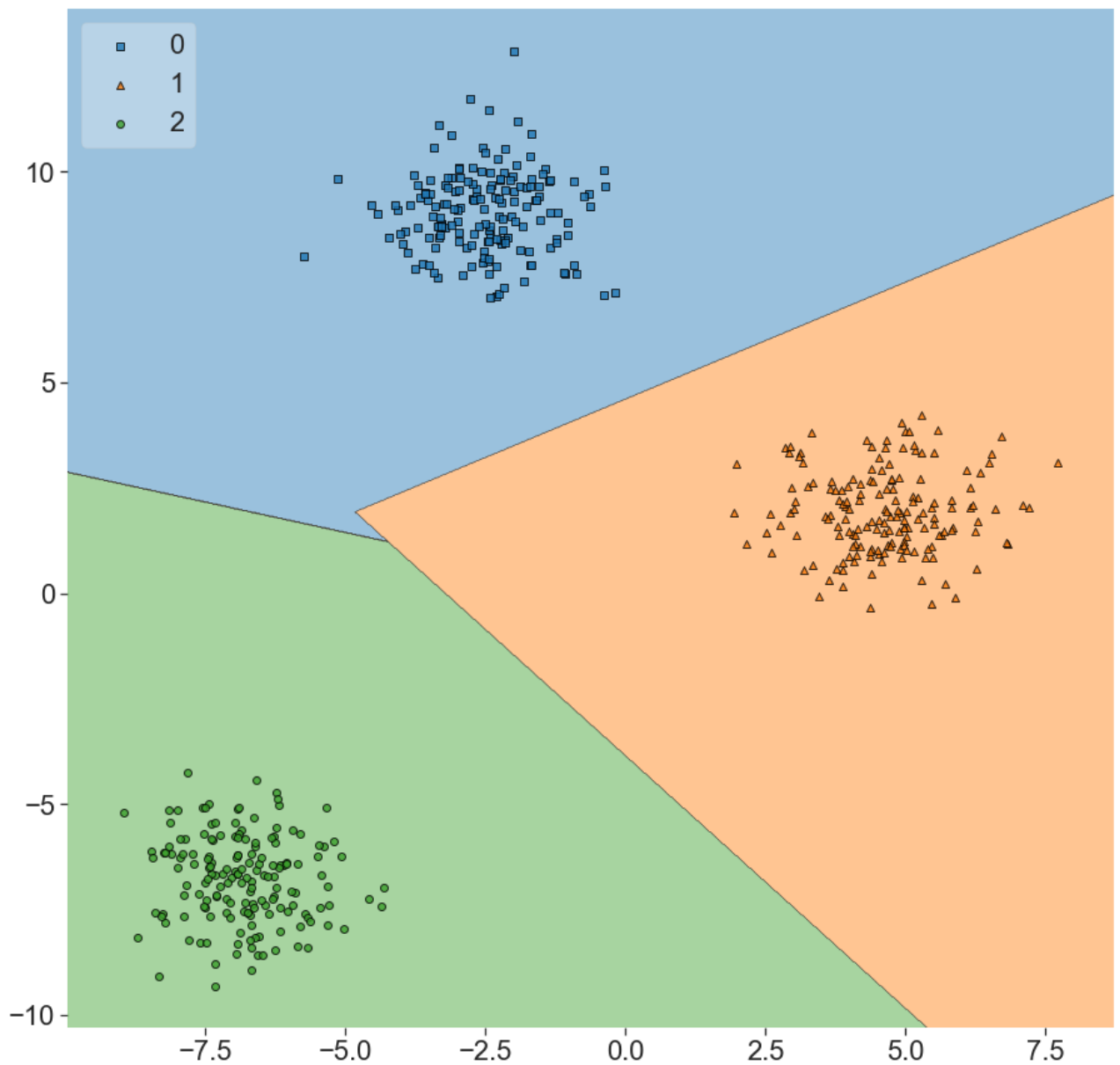
	precision	recall	f1-score	support
allow	0.999334	0.997078	0.998205	7528
deny	0.992027	0.995997	0.994008	2998
drop	0.997282	0.999222	0.998251	2570
accuracy			0.997251	13096
macro avg	0.996214	0.997432	0.996821	13096
weighted avg	0.997259	0.997251	0.997253	13096



```
In [27]: X, y = make_blobs(n_samples = 500,
                           centers = 3,
                           random_state = 42)

# Training a classifier
svm = SVC(C=0.5, kernel='linear')
svm.fit(X, y)

# Plotting decision regions
plot_decision_regions(X, y, clf=svm, legend=2)
plt.show()
```



Evaluate Kernels

In [28]: `p = ['linear', 'poly', 'rbf']`

```
for i in p:
    model = SVC(kernel = i)
    model.fit(X_train, y_train)
    y_hat_rf_test = model.predict(X_test)
    print(i, ":", accuracy_score(y_hat_rf_test, y_test))
```

```
[CV 3/10; 2/10] START model__C=0.1, model__class_weight=balanced, model__gamma=auto, model__kernel=poly
[CV 3/10; 2/10] END model__C=0.1, model__class_weight=balanced, model__gamma=auto, model__kernel=poly;; score=(train=0.064, test=0.065) total time=23.7min
[CV 7/10; 6/10] START model__C=0.1, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf
[CV 7/10; 6/10] END model__C=0.1, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf;; score=(train=0.996, test=0.997) total time= 1.7min
```



```
[CV 7/10; 7/10] START model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf
[CV 7/10; 7/10] END model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf;; score=(train=0.998, test=0.998) total time= 3.1min
[CV 3/10; 9/10] START model__C=100, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf
[CV 3/10; 9/10] END model__C=100, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf;; score=(train=0.998, test=0.997) total time= 3.1min
[CV 7/10; 9/10] START model__C=100, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf
[CV 7/10; 9/10] END model__C=100, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf;; score=(train=0.998, test=0.998) total time= 3.2min
[CV 8/10; 10/10] START model__C=0.1, model__class_weight=balanced, model__gamma=scale, model__kernel=linear
[CV 8/10; 10/10] END model__C=0.1, model__class_weight=balanced, model__gamma=scale, model__kernel=linear;; score=(train=0.996, test=0.996) total time= 1.4min
[CV 1/10; 1/10] START model__C=100, model__class_weight=balanced, model__gamma=0.001, model__kernel=linear
[CV 1/10; 1/10] END model__C=100, model__class_weight=balanced, model__gamma=0.001, model__kernel=linear;; score=(train=0.998, test=0.998) total time= 2.6min
[CV 6/10; 3/10] START model__C=1, model__class_weight=balanced, model__gamma=0.1, model__kernel=poly
[CV 6/10; 3/10] END model__C=1, model__class_weight=balanced, model__gamma=0.1, model__kernel=poly;; score=(train=0.996, test=0.996) total time= 3.4min
[CV 2/10; 4/10] START model__C=10, model__class_weight=balanced, model__gamma=1, model__kernel=linear
[CV 2/10; 4/10] END model__C=10, model__class_weight=balanced, model__gamma=1, model__kernel=linear;; score=(train=0.998, test=0.997) total time= 1.4min
[CV 4/10; 4/10] START model__C=10, model__class_weight=balanced, model__gamma=1, model__kernel=linear
[CV 4/10; 4/10] END model__C=10, model__class_weight=balanced, model__gamma=1, model__kernel=linear;; score=(train=0.998, test=0.998) total time= 1.6min
[CV 9/10; 4/10] START model__C=10, model__class_weight=balanced, model__gamma=1, model__kernel=linear
[CV 9/10; 4/10] END model__C=10, model__class_weight=balanced, model__gamma=1, model__kernel=linear;; score=(train=0.998, test=0.996) total time= 7.0min
[CV 5/10; 5/10] START model__C=10, model__class_weight=balanced, model__gamma=0.0001, model__kernel=linear
[CV 5/10; 5/10] END model__C=10, model__class_weight=balanced, model__gamma=0.0001, model__kernel=linear;; score=(train=0.998, test=0.997) total time= 2.3min
[CV 1/10; 6/10] START model__C=0.1, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf
[CV 1/10; 6/10] END model__C=0.1, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf;; score=(train=0.996, test=0.997) total time= 2.2min
[CV 8/10; 7/10] START model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf
[CV 8/10; 7/10] END model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf;; score=(train=0.998, test=0.997) total time= 3.2min
[CV 6/10; 9/10] START model__C=100, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf
[CV 6/10; 9/10] END model__C=100, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf;; score=(train=0.999, test=0.997) total time= 3.6min
```

```
[CV 2/10; 10/10] START model__C=0.1, model__class_weight=balanced, model__gamma=scale, model__kernel=linear
[CV 2/10; 10/10] END model__C=0.1, model__class_weight=balanced, model__gamma=scale, model__kernel=linear;; score=(train=0.996, test=0.996) total time=1.1min
[CV 3/10; 10/10] START model__C=0.1, model__class_weight=balanced, model__gamma=scale, model__kernel=linear
[CV 3/10; 10/10] END model__C=0.1, model__class_weight=balanced, model__gamma=scale, model__kernel=linear;; score=(train=0.996, test=0.995) total time=1.4min
[CV 5/10; 10/10] START model__C=0.1, model__class_weight=balanced, model__gamma=scale, model__kernel=linear
[CV 5/10; 10/10] END model__C=0.1, model__class_weight=balanced, model__gamma=scale, model__kernel=linear;; score=(train=0.996, test=0.996) total time=1.9min
[CV 3/10; 1/10] START model__C=100, model__class_weight=balanced, model__gamma=0.001, model__kernel=linear
[CV 3/10; 1/10] END model__C=100, model__class_weight=balanced, model__gamma=0.001, model__kernel=linear;; score=(train=0.998, test=0.997) total time=2.3min
[CV 8/10; 2/10] START model__C=0.1, model__class_weight=balanced, model__gamma=auto, model__kernel=poly
[CV 8/10; 2/10] END model__C=0.1, model__class_weight=balanced, model__gamma=auto, model__kernel=poly;; score=(train=0.064, test=0.064) total time=23.6min
[CV 6/10; 7/10] START model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf
[CV 6/10; 7/10] END model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf;; score=(train=0.998, test=0.997) total time=3.2min
[CV 4/10; 9/10] START model__C=100, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf
[CV 4/10; 9/10] END model__C=100, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf;; score=(train=0.998, test=0.998) total time=3.3min
[CV 9/10; 9/10] START model__C=100, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf
[CV 9/10; 9/10] END model__C=100, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf;; score=(train=0.998, test=0.996) total time=3.0min
[CV 7/10; 10/10] START model__C=0.1, model__class_weight=balanced, model__gamma=scale, model__kernel=linear
[CV 7/10; 10/10] END model__C=0.1, model__class_weight=balanced, model__gamma=scale, model__kernel=linear;; score=(train=0.996, test=0.997) total time=1.6min
[CV 2/10; 2/10] START model__C=0.1, model__class_weight=balanced, model__gamma=auto, model__kernel=poly
[CV 2/10; 2/10] END model__C=0.1, model__class_weight=balanced, model__gamma=auto, model__kernel=poly;; score=(train=0.085, test=0.085) total time=23.6min
[CV 4/10; 6/10] START model__C=0.1, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf
[CV 4/10; 6/10] END model__C=0.1, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf;; score=(train=0.996, test=0.997) total time=1.7min
[CV 9/10; 7/10] START model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf
[CV 9/10; 7/10] END model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf;; score=(train=0.998, test=0.996) total time=3.1min
```

```

[CV 5/10; 9/10] START model__C=100, model__class_weight=balanced, model__gamma
ma=0.1, model__kernel=rbf
[CV 5/10; 9/10] END model__C=100, model__class_weight=balanced, model__gamma
=0.1, model__kernel=rbf;; score=(train=0.998, test=0.997) total time= 3.6min
[CV 1/10; 10/10] START model__C=0.1, model__class_weight=balanced, model__ga
mma=scale, model__kernel=linear
[CV 1/10; 10/10] END model__C=0.1, model__class_weight=balanced, model__gamm
a=scale, model__kernel=linear;; score=(train=0.996, test=0.997) total time=
1.2min
[CV 4/10; 10/10] START model__C=0.1, model__class_weight=balanced, model__ga
mma=scale, model__kernel=linear
[CV 4/10; 10/10] END model__C=0.1, model__class_weight=balanced, model__gamm
a=scale, model__kernel=linear;; score=(train=0.996, test=0.997) total time=
1.5min
[CV 6/10; 10/10] START model__C=0.1, model__class_weight=balanced, model__ga
mma=scale, model__kernel=linear
[CV 6/10; 10/10] END model__C=0.1, model__class_weight=balanced, model__gamm
a=scale, model__kernel=linear;; score=(train=0.996, test=0.996) total time=
2.3min
linear : 0.9927458766035431
[CV 6/10; 2/10] START model__C=0.1, model__class_weight=balanced, model__gam
ma=auto, model__kernel=poly
[CV 6/10; 2/10] END model__C=0.1, model__class_weight=balanced, model__gamma
=auto, model__kernel=poly;; score=(train=0.064, test=0.064) total time=23.6m
in
[CV 3/10; 6/10] START model__C=0.1, model__class_weight=balanced, model__gam
ma=0.1, model__kernel=rbf
[CV 3/10; 6/10] END model__C=0.1, model__class_weight=balanced, model__gamma
=0.1, model__kernel=rbf;; score=(train=0.996, test=0.995) total time= 1.6min
[CV 5/10; 7/10] START model__C=100, model__class_weight=balanced, model__gam
ma=0.01, model__kernel=rbf
[CV 5/10; 7/10] END model__C=100, model__class_weight=balanced, model__gamma
=0.01, model__kernel=rbf;; score=(train=0.998, test=0.997) total time= 3.1mi
n
[CV 1/10; 9/10] START model__C=100, model__class_weight=balanced, model__gam
ma=0.1, model__kernel=rbf
[CV 1/10; 9/10] END model__C=100, model__class_weight=balanced, model__gamma
=0.1, model__kernel=rbf;; score=(train=0.998, test=0.997) total time= 3.5min
[CV 8/10; 9/10] START model__C=100, model__class_weight=balanced, model__gam
ma=0.1, model__kernel=rbf
[CV 8/10; 9/10] END model__C=100, model__class_weight=balanced, model__gamma
=0.1, model__kernel=rbf;; score=(train=0.998, test=0.997) total time= 3.9min
[CV 9/10; 10/10] START model__C=0.1, model__class_weight=balanced, model__ga
mma=scale, model__kernel=linear
[CV 9/10; 10/10] END model__C=0.1, model__class_weight=balanced, model__gamm
a=scale, model__kernel=linear;; score=(train=0.996, test=0.995) total time=
3.1min
[CV 4/10; 1/10] START model__C=100, model__class_weight=balanced, model__gam
ma=0.001, model__kernel=linear
[CV 4/10; 1/10] END model__C=100, model__class_weight=balanced, model__gamma
=0.001, model__kernel=linear;; score=(train=0.998, test=0.998) total time= 2
.4min
[CV 10/10; 2/10] START model__C=0.1, model__class_weight=balanced, model__ga
mma=auto, model__kernel=poly
[CV 10/10; 2/10] END model__C=0.1, model__class_weight=balanced, model__gamm
a=auto, model__kernel=poly;; score=(train=0.064, test=0.064) total time=23.6
min

```

```

[CV 10/10; 7/10] START model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf
[CV 10/10; 7/10] END model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf;; score=(train=0.998, test=0.997) total time= 2.9min
[CV 2/10; 9/10] START model__C=100, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf
[CV 2/10; 9/10] END model__C=100, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf;; score=(train=0.998, test=0.997) total time= 3.7min
[CV 10/10; 9/10] START model__C=100, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf
[CV 10/10; 9/10] END model__C=100, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf;; score=(train=0.998, test=0.998) total time= 3.7min
[CV 10/10; 10/10] START model__C=0.1, model__class_weight=balanced, model__gamma=scale, model__kernel=linear
[CV 10/10; 10/10] END model__C=0.1, model__class_weight=balanced, model__gamma=scale, model__kernel=linear;; score=(train=0.996, test=0.996) total time= 3.0min
poly : 0.5748320097739767
[CV 2/10; 1/10] START model__C=100, model__class_weight=balanced, model__gamma=0.001, model__kernel=linear
[CV 2/10; 1/10] END model__C=100, model__class_weight=balanced, model__gamma=0.001, model__kernel=linear;; score=(train=0.998, test=0.997) total time= 2.3min
[CV 9/10; 2/10] START model__C=0.1, model__class_weight=balanced, model__gamma=auto, model__kernel=poly
[CV 9/10; 2/10] END model__C=0.1, model__class_weight=balanced, model__gamma=auto, model__kernel=poly;; score=(train=0.064, test=0.064) total time=23.7min
[CV 1/10; 8/10] START model__C=100, model__class_weight=balanced, model__gamma=0.0001, model__kernel=poly
[CV 1/10; 8/10] END model__C=100, model__class_weight=balanced, model__gamma=0.0001, model__kernel=poly;; score=(train=0.064, test=0.064) total time=13.1min
[CV 5/10; 2/10] START model__C=0.1, model__class_weight=balanced, model__gamma=auto, model__kernel=poly
[CV 5/10; 2/10] END model__C=0.1, model__class_weight=balanced, model__gamma=auto, model__kernel=poly;; score=(train=0.064, test=0.064) total time=23.6min
[CV 5/10; 6/10] START model__C=0.1, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf
[CV 5/10; 6/10] END model__C=0.1, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf;; score=(train=0.996, test=0.996) total time= 2.0min
[CV 2/10; 8/10] START model__C=100, model__class_weight=balanced, model__gamma=0.0001, model__kernel=poly
[CV 2/10; 8/10] END model__C=100, model__class_weight=balanced, model__gamma=0.0001, model__kernel=poly;; score=(train=0.064, test=0.064) total time=13.1min
[CV 7/10; 1/10] START model__C=100, model__class_weight=balanced, model__gamma=0.001, model__kernel=linear
[CV 7/10; 1/10] END model__C=100, model__class_weight=balanced, model__gamma=0.001, model__kernel=linear;; score=(train=0.998, test=0.998) total time= 2.3min
[CV 1/10; 3/10] START model__C=1, model__class_weight=balanced, model__gamma=0.1, model__kernel=poly
[CV 1/10; 3/10] END model__C=1, model__class_weight=balanced, model__gamma=0

```

```
.1, model__kernel=poly;; score=(train=0.996, test=0.997) total time= 2.9min
[CV 7/10; 3/10] START model__C=1, model__class_weight=balanced, model__gamma
=0.1, model__kernel=poly
[CV 7/10; 3/10] END model__C=1, model__class_weight=balanced, model__gamma=0
.1, model__kernel=poly;; score=(train=0.996, test=0.997) total time= 2.7min
[CV 5/10; 4/10] START model__C=10, model__class_weight=balanced, model__gamma
a=1, model__kernel=linear
[CV 5/10; 4/10] END model__C=10, model__class_weight=balanced, model__gamma=
1, model__kernel=linear;; score=(train=0.998, test=0.997) total time=12.6min
[CV 1/10; 5/10] START model__C=10, model__class_weight=balanced, model__gamma
a=0.0001, model__kernel=linear
[CV 1/10; 5/10] END model__C=10, model__class_weight=balanced, model__gamma=
0.0001, model__kernel=linear;; score=(train=0.998, test=0.998) total time= 1
.9min
[CV 7/10; 5/10] START model__C=10, model__class_weight=balanced, model__gamma
a=0.0001, model__kernel=linear
[CV 7/10; 5/10] END model__C=10, model__class_weight=balanced, model__gamma=
0.0001, model__kernel=linear;; score=(train=0.998, test=0.998) total time= 1
.3min
[CV 9/10; 6/10] START model__C=0.1, model__class_weight=balanced, model__gamma
=0.1, model__kernel=rbf
[CV 9/10; 6/10] END model__C=0.1, model__class_weight=balanced, model__gamma
=0.1, model__kernel=rbf;; score=(train=0.996, test=0.995) total time= 1.7min
[CV 3/10; 8/10] START model__C=100, model__class_weight=balanced, model__gamma
=0.0001, model__kernel=poly
[CV 3/10; 8/10] END model__C=100, model__class_weight=balanced, model__gamma
=0.0001, model__kernel=poly;; score=(train=0.064, test=0.065) total time=13.
2min
[CV 1/10; 2/10] START model__C=0.1, model__class_weight=balanced, model__gamma
=auto, model__kernel=poly
[CV 1/10; 2/10] END model__C=0.1, model__class_weight=balanced, model__gamma
=auto, model__kernel=poly;; score=(train=0.085, test=0.085) total time=23.8m
in
[CV 8/10; 6/10] START model__C=0.1, model__class_weight=balanced, model__gamma
=0.1, model__kernel=rbf
[CV 8/10; 6/10] END model__C=0.1, model__class_weight=balanced, model__gamma
=0.1, model__kernel=rbf;; score=(train=0.996, test=0.996) total time= 2.1min
[CV 4/10; 8/10] START model__C=100, model__class_weight=balanced, model__gamma
=0.0001, model__kernel=poly
[CV 4/10; 8/10] END model__C=100, model__class_weight=balanced, model__gamma
=0.0001, model__kernel=poly;; score=(train=0.420, test=0.420) total time=13.
2min
[CV 4/10; 2/10] START model__C=0.1, model__class_weight=balanced, model__gamma
=auto, model__kernel=poly
[CV 4/10; 2/10] END model__C=0.1, model__class_weight=balanced, model__gamma
=auto, model__kernel=poly;; score=(train=0.064, test=0.064) total time=23.6m
in
[CV 6/10; 6/10] START model__C=0.1, model__class_weight=balanced, model__gamma
=0.1, model__kernel=rbf
[CV 6/10; 6/10] END model__C=0.1, model__class_weight=balanced, model__gamma
=0.1, model__kernel=rbf;; score=(train=0.996, test=0.996) total time= 2.1min
[CV 5/10; 8/10] START model__C=100, model__class_weight=balanced, model__gamma
=0.0001, model__kernel=poly
[CV 5/10; 8/10] END model__C=100, model__class_weight=balanced, model__gamma
=0.0001, model__kernel=poly;; score=(train=0.420, test=0.420) total time=13.
2min
[CV 5/10; 1/10] START model__C=100, model__class_weight=balanced, model__gamma
```

```
ma=0.001, model__kernel=linear
[CV 5/10; 1/10] END model__C=100, model__class_weight=balanced, model__gamma
=0.001, model__kernel=linear;; score=(train=0.998, test=0.997) total time= 2
.6min
[CV 5/10; 3/10] START model__C=1, model__class_weight=balanced, model__gamma
=0.1, model__kernel=poly
[CV 5/10; 3/10] END model__C=1, model__class_weight=balanced, model__gamma=0
.1, model__kernel=poly;; score=(train=0.996, test=0.996) total time= 3.2min
[CV 1/10; 4/10] START model__C=10, model__class_weight=balanced, model__gamma
a=1, model__kernel=linear
[CV 1/10; 4/10] END model__C=10, model__class_weight=balanced, model__gamma=
1, model__kernel=linear;; score=(train=0.998, test=0.998) total time= 1.5min
[CV 3/10; 4/10] START model__C=10, model__class_weight=balanced, model__gamma
a=1, model__kernel=linear
[CV 3/10; 4/10] END model__C=10, model__class_weight=balanced, model__gamma=
1, model__kernel=linear;; score=(train=0.998, test=0.997) total time= 1.6min
[CV 6/10; 4/10] START model__C=10, model__class_weight=balanced, model__gamma
a=1, model__kernel=linear
[CV 6/10; 4/10] END model__C=10, model__class_weight=balanced, model__gamma=
1, model__kernel=linear;; score=(train=0.999, test=0.997) total time=12.4min
[CV 2/10; 5/10] START model__C=10, model__class_weight=balanced, model__gamma
a=0.0001, model__kernel=linear
[CV 2/10; 5/10] END model__C=10, model__class_weight=balanced, model__gamma=
0.0001, model__kernel=linear;; score=(train=0.998, test=0.997) total time= 2
.0min
[CV 8/10; 5/10] START model__C=10, model__class_weight=balanced, model__gamma
a=0.0001, model__kernel=linear
[CV 8/10; 5/10] END model__C=10, model__class_weight=balanced, model__gamma=
0.0001, model__kernel=linear;; score=(train=0.998, test=0.997) total time= 1
.6min
[CV 10/10; 6/10] START model__C=0.1, model__class_weight=balanced, model__gamma
a=0.1, model__kernel=rbf
[CV 10/10; 6/10] END model__C=0.1, model__class_weight=balanced, model__gamma
a=0.1, model__kernel=rbf;; score=(train=0.996, test=0.996) total time= 1.7mi
n
[CV 6/10; 8/10] START model__C=100, model__class_weight=balanced, model__gamma
a=0.0001, model__kernel=poly
[CV 6/10; 8/10] END model__C=100, model__class_weight=balanced, model__gamma
=0.0001, model__kernel=poly;; score=(train=0.420, test=0.420) total time=13.
7min
[CV 9/10; 1/10] START model__C=100, model__class_weight=balanced, model__gamma
a=0.001, model__kernel=linear
[CV 9/10; 1/10] END model__C=100, model__class_weight=balanced, model__gamma
=0.001, model__kernel=linear;; score=(train=0.998, test=0.996) total time= 2
.5min
[CV 4/10; 3/10] START model__C=1, model__class_weight=balanced, model__gamma
=0.1, model__kernel=poly
[CV 4/10; 3/10] END model__C=1, model__class_weight=balanced, model__gamma=0
.1, model__kernel=poly;; score=(train=0.996, test=0.997) total time= 3.0min
[CV 10/10; 3/10] START model__C=1, model__class_weight=balanced, model__gamma
a=0.1, model__kernel=poly
[CV 10/10; 3/10] END model__C=1, model__class_weight=balanced, model__gamma=
0.1, model__kernel=poly;; score=(train=0.996, test=0.996) total time= 3.3min
[CV 8/10; 4/10] START model__C=10, model__class_weight=balanced, model__gamma
a=1, model__kernel=linear
[CV 8/10; 4/10] END model__C=10, model__class_weight=balanced, model__gamma=
1, model__kernel=linear;; score=(train=0.998, test=0.997) total time= 7.0min
```

```
[CV 6/10; 5/10] START model__C=10, model__class_weight=balanced, model__gamma=0.0001, model__kernel=linear
[CV 6/10; 5/10] END model__C=10, model__class_weight=balanced, model__gamma=0.0001, model__kernel=linear;; score=(train=0.999, test=0.997) total time= 2.1min
[CV 10/10; 5/10] START model__C=10, model__class_weight=balanced, model__gamma=0.0001, model__kernel=linear
[CV 10/10; 5/10] END model__C=10, model__class_weight=balanced, model__gamma=0.0001, model__kernel=linear;; score=(train=0.998, test=0.998) total time= 1.5min
[CV 1/10; 7/10] START model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf
[CV 1/10; 7/10] END model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf;; score=(train=0.998, test=0.998) total time= 2.0min
[CV 7/10; 8/10] START model__C=100, model__class_weight=balanced, model__gamma=0.0001, model__kernel=poly
[CV 7/10; 8/10] END model__C=100, model__class_weight=balanced, model__gamma=0.0001, model__kernel=poly;; score=(train=0.420, test=0.420) total time=14.8min
[CV 6/10; 1/10] START model__C=100, model__class_weight=balanced, model__gamma=0.001, model__kernel=linear
[CV 6/10; 1/10] END model__C=100, model__class_weight=balanced, model__gamma=0.001, model__kernel=linear;; score=(train=0.999, test=0.997) total time= 2.4min
[CV 2/10; 3/10] START model__C=1, model__class_weight=balanced, model__gamma=0.1, model__kernel=poly
[CV 2/10; 3/10] END model__C=1, model__class_weight=balanced, model__gamma=0.1, model__kernel=poly;; score=(train=0.996, test=0.996) total time= 2.9min
[CV 8/10; 3/10] START model__C=1, model__class_weight=balanced, model__gamma=0.1, model__kernel=poly
[CV 8/10; 3/10] END model__C=1, model__class_weight=balanced, model__gamma=0.1, model__kernel=poly;; score=(train=0.996, test=0.996) total time= 3.3min
[CV 7/10; 4/10] START model__C=10, model__class_weight=balanced, model__gamma=1, model__kernel=linear
[CV 7/10; 4/10] END model__C=10, model__class_weight=balanced, model__gamma=1, model__kernel=linear;; score=(train=0.998, test=0.998) total time=12.4min
[CV 3/10; 5/10] START model__C=10, model__class_weight=balanced, model__gamma=0.0001, model__kernel=linear
[CV 3/10; 5/10] END model__C=10, model__class_weight=balanced, model__gamma=0.0001, model__kernel=linear;; score=(train=0.998, test=0.997) total time= 2.1min
[CV 9/10; 5/10] START model__C=10, model__class_weight=balanced, model__gamma=0.0001, model__kernel=linear
[CV 9/10; 5/10] END model__C=10, model__class_weight=balanced, model__gamma=0.0001, model__kernel=linear;; score=(train=0.998, test=0.996) total time= 1.7min
[CV 2/10; 7/10] START model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf
[CV 2/10; 7/10] END model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf;; score=(train=0.998, test=0.997) total time= 2.0min
[CV 8/10; 8/10] START model__C=100, model__class_weight=balanced, model__gamma=0.0001, model__kernel=poly
[CV 8/10; 8/10] END model__C=100, model__class_weight=balanced, model__gamma=0.0001, model__kernel=poly;; score=(train=0.420, test=0.420) total time=14.7min
```

```
[CV 10/10; 1/10] START model__C=100, model__class_weight=balanced, model__gamma=0.001, model__kernel=linear
[CV 10/10; 1/10] END model__C=100, model__class_weight=balanced, model__gamma=0.001, model__kernel=linear;; score=(train=0.998, test=0.998) total time=2.1min
[CV 7/10; 2/10] START model__C=0.1, model__class_weight=balanced, model__gamma=auto, model__kernel=poly
[CV 7/10; 2/10] END model__C=0.1, model__class_weight=balanced, model__gamma=auto, model__kernel=poly;; score=(train=0.064, test=0.064) total time=23.3min
[CV 4/10; 7/10] START model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf
[CV 4/10; 7/10] END model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf;; score=(train=0.998, test=0.998) total time=2.5min
[CV 9/10; 8/10] START model__C=100, model__class_weight=balanced, model__gamma=0.0001, model__kernel=poly
[CV 9/10; 8/10] END model__C=100, model__class_weight=balanced, model__gamma=0.0001, model__kernel=poly;; score=(train=0.420, test=0.420) total time=14.0min
[CV 8/10; 1/10] START model__C=100, model__class_weight=balanced, model__gamma=0.001, model__kernel=linear
[CV 8/10; 1/10] END model__C=100, model__class_weight=balanced, model__gamma=0.001, model__kernel=linear;; score=(train=0.998, test=0.997) total time=2.5min
[CV 3/10; 3/10] START model__C=1, model__class_weight=balanced, model__gamma=0.1, model__kernel=poly
[CV 3/10; 3/10] END model__C=1, model__class_weight=balanced, model__gamma=0.1, model__kernel=poly;; score=(train=0.996, test=0.995) total time=3.0min
[CV 9/10; 3/10] START model__C=1, model__class_weight=balanced, model__gamma=0.1, model__kernel=poly
[CV 9/10; 3/10] END model__C=1, model__class_weight=balanced, model__gamma=0.1, model__kernel=poly;; score=(train=0.996, test=0.995) total time=3.4min
[CV 10/10; 4/10] START model__C=10, model__class_weight=balanced, model__gamma=1, model__kernel=linear
[CV 10/10; 4/10] END model__C=10, model__class_weight=balanced, model__gamma=1, model__kernel=linear;; score=(train=0.998, test=0.998) total time=6.7min
[CV 4/10; 5/10] START model__C=10, model__class_weight=balanced, model__gamma=0.0001, model__kernel=linear
[CV 4/10; 5/10] END model__C=10, model__class_weight=balanced, model__gamma=0.0001, model__kernel=linear;; score=(train=0.998, test=0.998) total time=2.3min
[CV 2/10; 6/10] START model__C=0.1, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf
[CV 2/10; 6/10] END model__C=0.1, model__class_weight=balanced, model__gamma=0.1, model__kernel=rbf;; score=(train=0.996, test=0.996) total time=1.8min
[CV 3/10; 7/10] START model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf
[CV 3/10; 7/10] END model__C=100, model__class_weight=balanced, model__gamma=0.01, model__kernel=rbf;; score=(train=0.998, test=0.997) total time=2.6min
[CV 10/10; 8/10] START model__C=100, model__class_weight=balanced, model__gamma=0.0001, model__kernel=poly
[CV 10/10; 8/10] END model__C=100, model__class_weight=balanced, model__gamma=0.0001, model__kernel=poly;; score=(train=0.420, test=0.420) total time=13.9min
```



```
rbf : 0.9186774587660355
```

```
In [29]: C = [0.001, 0.01, 0.1, 1, 10, 100]

for i in C:
    model = SVC(C = i)
    model.fit(X_train, y_train)
    y_hat_rf_test = model.predict(X_test)
    print(str(i), ":", accuracy_score(y_hat_rf_test, y_test))

0.001 : 0.5748320097739767
0.01  : 0.5748320097739767
0.1   : 0.5748320097739767
1     : 0.9186774587660355
10    : 0.9192883323152108
100   : 0.9496792913866829
```

SGD Classifier

```
In [30]: def plot_svc_decision_function(model, ax=None, plot_support=True):
    """Plot the decision function for a 2D SVC"""
    if ax is None:
        ax = plt.gca()
    xlim = ax.get_xlim()
    ylim = ax.get_ylim()

    # create grid to evaluate model
    x = np.linspace(xlim[0], xlim[1], 30)
    y = np.linspace(ylim[0], ylim[1], 30)
    Y, X = np.meshgrid(y, x)
    xy = np.vstack([X.ravel(), Y.ravel()]).T
    P = model.decision_function(xy).reshape(X.shape)

    # plot decision boundary and margins
    ax.contour(X, Y, P, colors='k',
               levels=[-1, 0, 1], alpha=0.5,
               linestyles=['--', '-', '--'])

    # plot support vectors
    if plot_support:
        ax.scatter(model.support_vectors_[:, 0],
                   model.support_vectors_[:, 1],
                   s=300, linewidth=1, facecolors='none');
    ax.set_xlim(xlim)
    ax.set_ylim(ylim)
```

```
In [31]: params_sgd = [
    # optimal learning rate
    {
        "model__learning_rate": ["optimal"],
        "model__alpha": [0.000001, 0.00001, 0.0001, 0.001, 0.01],
        "model__eta0": [0],
        "model__class_weight":["balanced"]
    },
    # constant learning rate (eta0 cannot be 0 for this)
    {
        "model__learning_rate": ["constant"],
        "model__alpha": [0.000001, 0.00001, 0.0001, 0.001, 0.01],
        "model__eta0": [0.01, 0.1, 1.0, 5, 10],
        "model__class_weight":["balanced"]
    }
]
```

```
In [32]: %%time

# define preprocessing for columns
t = [('cat', OneHotEncoder(handle_unknown='ignore'), df_ohe),
      ('scale', MinMaxScaler(), df_numeric)]

col_transform = ColumnTransformer(t)

model_sgd = SGDClassifier(loss = 'log_loss')

pipeline_sgd = Pipeline(steps=[('preprocessing', col_transform), ('model', model_sgd)])

#Fit
clf_sgd = GridSearchCV(pipeline_sgd,
                       params_sgd,
                       scoring = 'f1_weighted',
                       verbose = 15,
                       cv = cv,
                       n_jobs = -1,
                       return_train_score = True)

search_sgd = clf_sgd.fit(X_train, y_train)
```

Fitting 10 folds for each of 30 candidates, totalling 300 fits
 CPU times: user 3.13 s, sys: 1.49 s, total: 4.62 s
 Wall time: 26.4 s

```
In [33]: df_search_sgd_cv_results_columns = ['param_model__alpha',
                                             'param_model__eta0',
                                             'param_model__learning_rate',
                                             'mean_train_score',
                                             'mean_test_score',
                                             'rank_test_score']

df_search_sgd_cv_results = cv_summary(search_sgd, df_search_sgd_cv_results_columns)
df_search_sgd_cv_results
```

Out[33]:

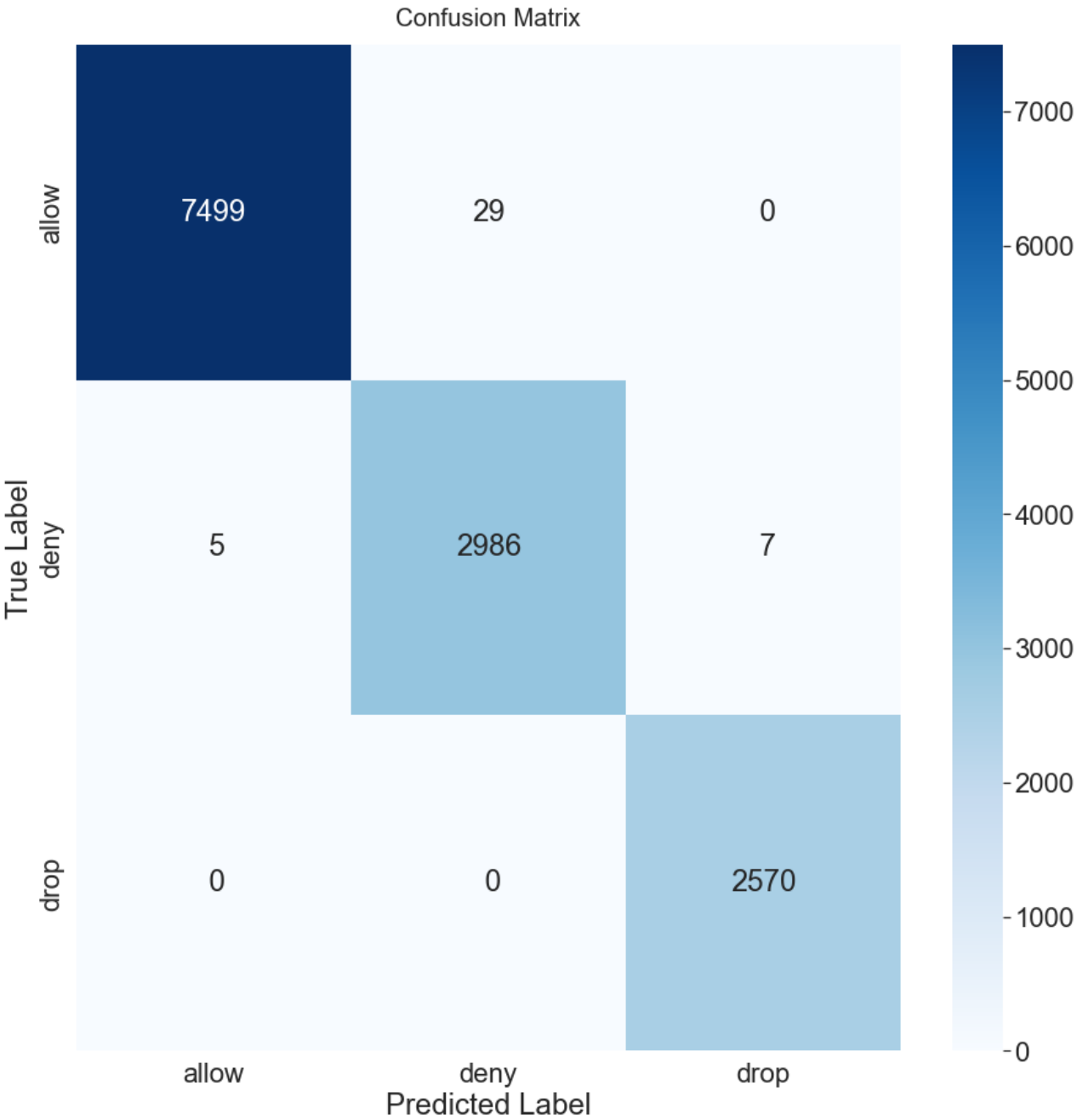
	param_combination	param_model__alpha	param_model__eta0	param_model__learning_
0	1	0.000001	0	op
1	2	0.00001	0	op
2	3	0.0001	0	op
3	4	0.001	0	op
4	5	0.01	0	op
5	6	0.000001	0.01	cons
6	7	0.000001	0.1	cons
7	8	0.000001	1.0	cons
8	9	0.000001	5	cons
9	10	0.000001	10	cons
10	11	0.00001	0.01	cons
11	12	0.00001	0.1	cons
12	13	0.00001	1.0	cons
13	14	0.00001	5	cons
14	15	0.00001	10	cons
15	16	0.0001	0.01	cons
16	17	0.0001	0.1	cons
17	18	0.0001	1.0	cons
18	19	0.0001	5	cons
19	20	0.0001	10	cons
20	21	0.001	0.01	cons
21	22	0.001	0.1	cons
22	23	0.001	1.0	cons
23	24	0.001	5	cons
24	25	0.001	10	cons
25	26	0.01	0.01	cons
26	27	0.01	0.1	cons
27	28	0.01	1.0	cons
28	29	0.01	5	cons
29	30	0.01	10	cons

```
In [34]: predict_sgd = search_sgd.predict(X_test)
get_classification_report(X_train, y_train, X_test, y_test, predict_sgd, sea
```

Training Score: 0.997826

Test result:

	precision	recall	f1-score	support
allow	0.999334	0.996148	0.997738	7528
deny	0.990381	0.995997	0.993181	2998
drop	0.997284	1.000000	0.998640	2570
accuracy			0.996869	13096
macro avg	0.995666	0.997382	0.996520	13096
weighted avg	0.996882	0.996869	0.996872	13096



Collapsing to Two Classes

```
In [35]: df['target'] = df['Action'].map({'allow': "Allow",
                                         'deny': "Deny",
                                         'drop': "Deny",
                                         'reset-both': "Deny"})
df["target"].value_counts(normalize = False)
```

```
Out[35]: Allow      37640
Deny        27838
Name: target, dtype: int64
```

```
In [36]: df.drop(columns = "Action", inplace = True)
df.head()
```

```
Out[36]:
```

	Source Port	Destination Port	NAT Source Port	NAT Destination Port	Bytes	Bytes Sent	Bytes Received	Packets	Elapsed Time (sec)	pkt
0	57222	53	54587	53	177	94	83	2	30	
1	56258	3389	56258	3389	4768	1600	3168	19	17	
2	6881	50321	43265	50321	238	118	120	2	1199	
3	50553	3389	50553	3389	3327	1438	1889	15	17	
4	50002	443	45848	443	25358	6778	18580	31	16	

```
In [37]: X = df.loc[:, df.columns != 'target'].values
y = df['target'].values

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state = 42)
```

```
In [38]: param_grid= {'kernel': ('linear', 'rbf'),
                      'C': [0.001, 0.01, 0.1, 1, 10, 100]}
```

```
base_estimator = SVC(gamma = 'scale')
```

```
grid_search = GridSearchCV(base_estimator,
                           param_grid = param_grid,
                           n_jobs = -1)
```

```
grid_search.fit(X_train, y_train)
```

```
[CV 2/10; 1/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
```

```
[CV 2/10; 1/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.998, test=0.997) total time= 1.3s
```

```
[CV 7/10; 3/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
```

```
[CV 7/10; 3/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.996, test=0.997) total time= 0.7s
```

```
[CV 3/10; 5/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
```

```
[CV 3/10; 5/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.991, test=0.990) total
```

```
time= 0.7s
[CV 10/10; 6/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 10/10; 6/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 1.0s
[CV 4/10; 8/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 4/10; 8/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.998, test=0.998)
total time= 0.8s
[CV 2/10; 10/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 2/10; 10/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.997, test=0.997)
total time= 1.1s
[CV 10/10; 11/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 10/10; 11/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 1.0s
[CV 8/10; 13/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 8/10; 13/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.996, test=0.997)
total time= 0.7s
[CV 2/10; 15/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 2/10; 15/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.995, test=0.995)
total time= 0.8s
[CV 8/10; 16/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 8/10; 16/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 0.9s
[CV 5/10; 18/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 5/10; 18/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.992, test=0.990)
total time= 0.7s
[CV 1/10; 20/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 1/10; 20/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.996, test=0.997)
total time= 0.9s
[CV 7/10; 21/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 7/10; 21/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.8s
[CV 3/10; 23/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 3/10; 23/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.991, test=0.990)
total time= 0.7s
[CV 9/10; 24/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
```

```

el__eta0=5, model__learning_rate=constant
[CV 9/10; 24/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.991, test=0.987) total time= 1.0s
[CV 7/10; 26/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 7/10; 26/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.990, test=0.992) total time= 0.8s
[CV 2/10; 28/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 2/10; 28/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.990, test=0.992) total time= 0.8s
[CV 7/10; 29/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 7/10; 29/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.728, test=0.730) total time= 1.0s
[CV 10/10; 30/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 10/10; 30/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.728, test=0.728) total time= 0.6s
[CV 2/10; 2/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 2/10; 2/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.997, test=0.997) total time= 1.0s
[CV 2/10; 3/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 2/10; 3/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.996, test=0.996) total time= 0.7s
[CV 6/10; 4/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 6/10; 4/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.991, test=0.990) total time= 0.7s
[CV 3/10; 6/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 3/10; 6/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.995) total time= 0.9s
[CV 9/10; 7/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 9/10; 7/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.995) total time= 0.7s
[CV 6/10; 9/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 6/10; 9/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.998, test=0.996) total time= 1.1s
[CV 2/10; 11/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 2/10; 11/30] END model__alpha=1e-05, model__class_weight=balanced, model

```

```
__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 0.9s
[CV 8/10; 12/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 8/10; 12/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.996)
total time= 0.7s
[CV 4/10; 14/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 4/10; 14/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.997, test=0.998)
total time= 0.7s
[CV 10/10; 15/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 10/10; 15/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.992, test=0.992)
total time= 0.7s
[CV 6/10; 17/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 6/10; 17/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 0.5s
[CV 2/10; 19/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 2/10; 19/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.995, test=0.995)
total time= 0.5s
[CV 8/10; 20/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 8/10; 20/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.996, test=0.994)
total time= 0.9s
[CV 3/10; 22/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 3/10; 22/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.991, test=0.990)
total time= 0.6s
[CV 10/10; 23/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 10/10; 23/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.991, test=0.990)
total time= 0.6s
[CV 6/10; 25/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 6/10; 25/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.728, test=0.728)
total time= 1.0s
[CV 2/10; 27/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 2/10; 27/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.7s
[CV 8/10; 28/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 8/10; 28/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.7s
```



```
[CV 4/10; 30/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 4/10; 30/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.420, test=0.420) total time= 1.2s
[CV 4/10; 2/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 4/10; 2/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.997, test=0.998) total time= 0.9s
[CV 7/10; 2/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 7/10; 2/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.997, test=0.998) total time= 0.7s
[CV 4/10; 4/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 4/10; 4/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.991, test=0.991) total time= 0.6s
[CV 9/10; 5/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 9/10; 5/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.991, test=0.987) total time= 0.6s
[CV 6/10; 7/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 6/10; 7/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.996) total time= 0.5s
[CV 1/10; 9/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 1/10; 9/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.998, test=0.997) total time= 0.8s
[CV 7/10; 10/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 7/10; 10/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.997, test=0.997) total time= 0.9s
[CV 3/10; 12/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 3/10; 12/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.995) total time= 0.7s
[CV 9/10; 13/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 9/10; 13/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.997, test=0.997) total time= 0.7s
[CV 5/10; 15/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 5/10; 15/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.8s
[CV 1/10; 17/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
```

```

[CV 1/10; 17/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.8s
[CV 3/10; 18/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 3/10; 18/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.996, test=0.995) total time= 0.7s
[CV 9/10; 19/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 9/10; 19/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.996, test=0.995) total time= 0.7s
[CV 6/10; 21/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 6/10; 21/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.991, test=0.990) total time= 0.8s
[CV 2/10; 23/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 2/10; 23/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.990, test=0.992) total time= 0.6s
[CV 6/10; 24/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 6/10; 24/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.991, test=0.990) total time= 0.8s
[CV 9/10; 25/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 9/10; 25/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.991, test=0.987) total time= 0.8s
[CV 5/10; 27/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 5/10; 27/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.991, test=0.989) total time= 0.8s
[CV 1/10; 29/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 1/10; 29/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.740, test=0.742) total time= 0.9s
[CV 7/10; 30/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 7/10; 30/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.693, test=0.694) total time= 0.8s
[CV 1/10; 2/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 1/10; 2/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.998, test=0.997) total time= 1.0s
[CV 9/10; 2/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 9/10; 2/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.997, test=0.995) total time= 1.0s

```

```
l time= 0.7s
[CV 7/10; 4/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 7/10; 4/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.990, test=0.992) total time= 0.7s
[CV 4/10; 6/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 4/10; 6/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.9s
[CV 10/10; 7/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 10/10; 7/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.996) total time= 0.7s
[CV 5/10; 9/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 5/10; 9/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.998, test=0.998) total time= 0.9s
[CV 1/10; 11/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 1/10; 11/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.8s
[CV 7/10; 12/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 7/10; 12/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.6s
[CV 3/10; 14/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 3/10; 14/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.992, test=0.991) total time= 0.6s
[CV 9/10; 15/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 9/10; 15/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.997, test=0.995) total time= 0.6s
[CV 2/10; 17/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 2/10; 17/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 0.7s
[CV 8/10; 18/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 8/10; 18/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 0.7s
[CV 5/10; 20/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 5/10; 20/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.992, test=0.990) total time= 0.9s
[CV 1/10; 22/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
```

```

el__eta0=0.1, model__learning_rate=constant
[CV 1/10; 22/30] END model__alpha=0.001, model__class_weight=balanced, model
__eta0=0.1, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.7s
[CV 6/10; 23/30] START model__alpha=0.001, model__class_weight=balanced, mod
el__eta0=1.0, model__learning_rate=constant
[CV 6/10; 23/30] END model__alpha=0.001, model__class_weight=balanced, model
__eta0=1.0, model__learning_rate=constant;; score=(train=0.991, test=0.990)
total time= 0.7s
[CV 3/10; 25/30] START model__alpha=0.001, model__class_weight=balanced, mod
el__eta0=10, model__learning_rate=constant
[CV 3/10; 25/30] END model__alpha=0.001, model__class_weight=balanced, model
__eta0=10, model__learning_rate=constant;; score=(train=0.991, test=0.990) t
otal time= 0.9s
[CV 8/10; 26/30] START model__alpha=0.01, model__class_weight=balanced, mode
l__eta0=0.01, model__learning_rate=constant
[CV 8/10; 26/30] END model__alpha=0.01, model__class_weight=balanced, model
__eta0=0.01, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.8s
[CV 3/10; 28/30] START model__alpha=0.01, model__class_weight=balanced, mode
l__eta0=1.0, model__learning_rate=constant
[CV 3/10; 28/30] END model__alpha=0.01, model__class_weight=balanced, model
__eta0=1.0, model__learning_rate=constant;; score=(train=0.991, test=0.990) t
otal time= 0.8s
[CV 9/10; 29/30] START model__alpha=0.01, model__class_weight=balanced, mode
l__eta0=5, model__learning_rate=constant
[CV 9/10; 29/30] END model__alpha=0.01, model__class_weight=balanced, model
__eta0=5, model__learning_rate=constant;; score=(train=0.693, test=0.691) tot
al time= 1.0s
[CV 10/10; 1/30] START model__alpha=1e-06, model__class_weight=balanced, mod
el__eta0=0, model__learning_rate=optimal
[CV 10/10; 1/30] END model__alpha=1e-06, model__class_weight=balanced, model
__eta0=0, model__learning_rate=optimal;; score=(train=0.998, test=0.997) tot
al time= 1.3s
[CV 6/10; 3/30] START model__alpha=0.0001, model__class_weight=balanced, mod
el__eta0=0, model__learning_rate=optimal
[CV 6/10; 3/30] END model__alpha=0.0001, model__class_weight=balanced, model
__eta0=0, model__learning_rate=optimal;; score=(train=0.996, test=0.996) tot
al time= 0.7s
[CV 1/10; 5/30] START model__alpha=0.01, model__class_weight=balanced, model
__eta0=0, model__learning_rate=optimal
[CV 1/10; 5/30] END model__alpha=0.01, model__class_weight=balanced, model
__eta0=0, model__learning_rate=optimal;; score=(train=0.990, test=0.992) total
time= 0.7s
[CV 7/10; 6/30] START model__alpha=1e-06, model__class_weight=balanced, mode
l__eta0=0.01, model__learning_rate=constant
[CV 7/10; 6/30] END model__alpha=1e-06, model__class_weight=balanced, model
__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.997)
total time= 1.0s
[CV 8/10; 8/30] START model__alpha=1e-06, model__class_weight=balanced, mode
l__eta0=1.0, model__learning_rate=constant
[CV 8/10; 8/30] END model__alpha=1e-06, model__class_weight=balanced, model
__eta0=1.0, model__learning_rate=constant;; score=(train=0.998, test=0.997) t
otal time= 0.8s
[CV 1/10; 10/30] START model__alpha=1e-06, model__class_weight=balanced, mod
el__eta0=10, model__learning_rate=constant
[CV 1/10; 10/30] END model__alpha=1e-06, model__class_weight=balanced, model

```

```

__eta0=10, model__learning_rate=constant;; score=(train=0.997, test=0.998) t
otal time= 1.2s
[CV 2/10; 12/30] START model__alpha=1e-05, model__class_weight=balanced, mod
el__eta0=0.1, model__learning_rate=constant
[CV 2/10; 12/30] END model__alpha=1e-05, model__class_weight=balanced, model
__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.996)
total time= 0.8s
[CV 5/10; 13/30] START model__alpha=1e-05, model__class_weight=balanced, mod
el__eta0=1.0, model__learning_rate=constant
[CV 5/10; 13/30] END model__alpha=1e-05, model__class_weight=balanced, model
__eta0=1.0, model__learning_rate=constant;; score=(train=0.997, test=0.997)
total time= 0.8s
[CV 4/10; 15/30] START model__alpha=1e-05, model__class_weight=balanced, mod
el__eta0=10, model__learning_rate=constant
[CV 4/10; 15/30] END model__alpha=1e-05, model__class_weight=balanced, model
__eta0=10, model__learning_rate=constant;; score=(train=0.997, test=0.997) t
otal time= 0.8s
[CV 9/10; 16/30] START model__alpha=0.0001, model__class_weight=balanced, mo
del__eta0=0.01, model__learning_rate=constant
[CV 9/10; 16/30] END model__alpha=0.0001, model__class_weight=balanced, mode
l__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.995
) total time= 0.9s
[CV 6/10; 18/30] START model__alpha=0.0001, model__class_weight=balanced, mo
del__eta0=1.0, model__learning_rate=constant
[CV 6/10; 18/30] END model__alpha=0.0001, model__class_weight=balanced, mode
l__eta0=1.0, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 0.7s
[CV 3/10; 20/30] START model__alpha=0.0001, model__class_weight=balanced, mo
del__eta0=10, model__learning_rate=constant
[CV 3/10; 20/30] END model__alpha=0.0001, model__class_weight=balanced, mode
l__eta0=10, model__learning_rate=constant;; score=(train=0.995, test=0.995)
total time= 1.2s
[CV 4/10; 22/30] START model__alpha=0.001, model__class_weight=balanced, mod
el__eta0=0.1, model__learning_rate=constant
[CV 4/10; 22/30] END model__alpha=0.001, model__class_weight=balanced, model
__eta0=0.1, model__learning_rate=constant;; score=(train=0.991, test=0.991)
total time= 0.6s
[CV 9/10; 23/30] START model__alpha=0.001, model__class_weight=balanced, mod
el__eta0=1.0, model__learning_rate=constant
[CV 9/10; 23/30] END model__alpha=0.001, model__class_weight=balanced, model
__eta0=1.0, model__learning_rate=constant;; score=(train=0.991, test=0.987)
total time= 0.6s
[CV 5/10; 25/30] START model__alpha=0.001, model__class_weight=balanced, mod
el__eta0=10, model__learning_rate=constant
[CV 5/10; 25/30] END model__alpha=0.001, model__class_weight=balanced, model
__eta0=10, model__learning_rate=constant;; score=(train=0.990, test=0.988) t
otal time= 1.0s
[CV 1/10; 27/30] START model__alpha=0.01, model__class_weight=balanced, mode
l__eta0=0.1, model__learning_rate=constant
[CV 1/10; 27/30] END model__alpha=0.01, model__class_weight=balanced, model
__eta0=0.1, model__learning_rate=constant;; score=(train=0.990, test=0.992) t
otal time= 0.8s
[CV 7/10; 28/30] START model__alpha=0.01, model__class_weight=balanced, mode
l__eta0=1.0, model__learning_rate=constant
[CV 7/10; 28/30] END model__alpha=0.01, model__class_weight=balanced, model
__eta0=1.0, model__learning_rate=constant;; score=(train=0.990, test=0.992) t
otal time= 0.8s

```

```
[CV 3/10; 30/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 3/10; 30/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.728, test=0.728) total time= 1.2s
[CV 6/10; 1/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 6/10; 1/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.998, test=0.997) total time= 1.3s
[CV 3/10; 3/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 3/10; 3/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.996, test=0.995) total time= 0.7s
[CV 9/10; 4/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 9/10; 4/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.991, test=0.987) total time= 0.7s
[CV 5/10; 6/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 5/10; 6/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 1.0s
[CV 2/10; 8/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 2/10; 8/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.998, test=0.998) total time= 0.8s
[CV 7/10; 9/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 7/10; 9/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.997, test=0.997) total time= 1.0s
[CV 3/10; 11/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 3/10; 11/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.995) total time= 0.9s
[CV 9/10; 12/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 9/10; 12/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.995) total time= 0.7s
[CV 5/10; 14/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 5/10; 14/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 0.7s
[CV 1/10; 16/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 1/10; 16/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.8s
[CV 7/10; 17/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
```

```
[CV 7/10; 17/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.6s
[CV 3/10; 19/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 3/10; 19/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.6s
[CV 9/10; 20/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 9/10; 20/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.990, test=0.987) total time= 0.7s
[CV 5/10; 22/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 5/10; 22/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.991, test=0.989) total time= 0.5s
[CV 1/10; 24/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 1/10; 24/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.990, test=0.991) total time= 0.8s
[CV 7/10; 25/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 7/10; 25/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.994, test=0.996) total time= 0.7s
[CV 3/10; 27/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 3/10; 27/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.991, test=0.990) total time= 0.6s
[CV 9/10; 28/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 9/10; 28/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.991, test=0.987) total time= 0.6s
[CV 5/10; 30/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 5/10; 30/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.693, test=0.691) total time= 1.1s
[CV 7/10; 1/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 7/10; 1/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.998, test=0.998) total time= 1.4s
[CV 9/10; 3/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 9/10; 3/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.996, test=0.995) total time= 0.7s
[CV 6/10; 5/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 6/10; 5/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.991, test=0.990) total
```

```
time= 0.7s
[CV 2/10; 7/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 2/10; 7/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.997) total time= 0.9s
[CV 3/10; 8/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 3/10; 8/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.998, test=0.998) total time= 0.9s
[CV 10/10; 9/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 10/10; 9/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.998, test=0.997) total time= 1.0s
[CV 5/10; 11/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 5/10; 11/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 0.9s
[CV 1/10; 13/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 1/10; 13/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.997, test=0.997) total time= 0.7s
[CV 7/10; 14/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 7/10; 14/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 0.8s
[CV 4/10; 16/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 4/10; 16/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.9s
[CV 10/10; 17/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 10/10; 17/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 0.7s
[CV 7/10; 19/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 7/10; 19/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.720, test=0.720) total time= 0.8s
[CV 3/10; 21/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 3/10; 21/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.991, test=0.990) total time= 0.8s
[CV 9/10; 22/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 9/10; 22/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.991, test=0.987) total time= 0.7s
[CV 7/10; 24/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
```



```

el__eta0=5, model__learning_rate=constant
[CV 7/10; 24/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.990, test=0.992) total time= 0.8s
[CV 1/10; 26/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 1/10; 26/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.990, test=0.992) total time= 0.7s
[CV 4/10; 27/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 4/10; 27/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.991, test=0.991) total time= 0.7s
[CV 10/10; 28/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 10/10; 28/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.991, test=0.990) total time= 0.6s
[CV 6/10; 30/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 6/10; 30/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.676, test=0.675) total time= 1.0s
[CV 5/10; 2/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 5/10; 2/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.997, test=0.997) total time= 1.0s
[CV 10/10; 2/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 10/10; 2/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.997, test=0.997) total time= 0.8s
[CV 8/10; 4/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 8/10; 4/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.990, test=0.992) total time= 0.6s
[CV 2/10; 6/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 2/10; 6/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 0.9s
[CV 8/10; 7/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 8/10; 7/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.996) total time= 0.7s
[CV 4/10; 9/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 4/10; 9/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.998, test=0.997) total time= 0.9s
[CV 9/10; 10/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 9/10; 10/30] END model__alpha=1e-06, model__class_weight=balanced, model

```

```
__eta0=10, model__learning_rate=constant;; score=(train=0.998, test=0.997) t
otal time= 0.8s
[CV 6/10; 12/30] START model__alpha=1e-05, model__class_weight=balanced, mod
el__eta0=0.1, model__learning_rate=constant
[CV 6/10; 12/30] END model__alpha=1e-05, model__class_weight=balanced, model
__eta0=0.1, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 0.6s
[CV 2/10; 14/30] START model__alpha=1e-05, model__class_weight=balanced, mod
el__eta0=5, model__learning_rate=constant
[CV 2/10; 14/30] END model__alpha=1e-05, model__class_weight=balanced, model
__eta0=5, model__learning_rate=constant;; score=(train=0.997, test=0.997) to
tal time= 0.6s
[CV 7/10; 15/30] START model__alpha=1e-05, model__class_weight=balanced, mod
el__eta0=10, model__learning_rate=constant
[CV 7/10; 15/30] END model__alpha=1e-05, model__class_weight=balanced, model
__eta0=10, model__learning_rate=constant;; score=(train=0.995, test=0.996) t
otal time= 0.9s
[CV 5/10; 17/30] START model__alpha=0.0001, model__class_weight=balanced, mo
del__eta0=0.1, model__learning_rate=constant
[CV 5/10; 17/30] END model__alpha=0.0001, model__class_weight=balanced, mode
l__eta0=0.1, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 0.6s
[CV 1/10; 19/30] START model__alpha=0.0001, model__class_weight=balanced, mo
del__eta0=5, model__learning_rate=constant
[CV 1/10; 19/30] END model__alpha=0.0001, model__class_weight=balanced, mode
l__eta0=5, model__learning_rate=constant;; score=(train=0.995, test=0.995) t
otal time= 0.7s
[CV 7/10; 20/30] START model__alpha=0.0001, model__class_weight=balanced, mo
del__eta0=10, model__learning_rate=constant
[CV 7/10; 20/30] END model__alpha=0.0001, model__class_weight=balanced, mode
l__eta0=10, model__learning_rate=constant;; score=(train=0.996, test=0.997)
total time= 0.8s
[CV 2/10; 22/30] START model__alpha=0.001, model__class_weight=balanced, mod
el__eta0=0.1, model__learning_rate=constant
[CV 2/10; 22/30] END model__alpha=0.001, model__class_weight=balanced, model
__eta0=0.1, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.7s
[CV 7/10; 23/30] START model__alpha=0.001, model__class_weight=balanced, mod
el__eta0=1.0, model__learning_rate=constant
[CV 7/10; 23/30] END model__alpha=0.001, model__class_weight=balanced, model
__eta0=1.0, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.7s
[CV 2/10; 25/30] START model__alpha=0.001, model__class_weight=balanced, mod
el__eta0=10, model__learning_rate=constant
[CV 2/10; 25/30] END model__alpha=0.001, model__class_weight=balanced, model
__eta0=10, model__learning_rate=constant;; score=(train=0.990, test=0.991) t
otal time= 1.1s
[CV 9/10; 26/30] START model__alpha=0.01, model__class_weight=balanced, mode
l__eta0=0.01, model__learning_rate=constant
[CV 9/10; 26/30] END model__alpha=0.01, model__class_weight=balanced, model
__eta0=0.01, model__learning_rate=constant;; score=(train=0.991, test=0.987)
total time= 0.7s
[CV 5/10; 28/30] START model__alpha=0.01, model__class_weight=balanced, mode
l__eta0=1.0, model__learning_rate=constant
[CV 5/10; 28/30] END model__alpha=0.01, model__class_weight=balanced, model
__eta0=1.0, model__learning_rate=constant;; score=(train=0.991, test=0.989) t
otal time= 0.8s
```

```

[CV 2/10; 30/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 2/10; 30/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.990, test=0.992) total time= 1.2s
[CV 3/10; 2/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 3/10; 2/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.997, test=0.996) total time= 1.0s
[CV 8/10; 2/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 8/10; 2/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.997, test=0.997) total time= 0.7s
[CV 5/10; 4/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 5/10; 4/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.991, test=0.989) total time= 0.6s
[CV 10/10; 5/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 10/10; 5/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.991, test=0.990) total time= 0.6s
[CV 5/10; 7/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 5/10; 7/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.997) total time= 0.5s
[CV 2/10; 9/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 2/10; 9/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.998, test=0.997) total time= 1.0s
[CV 8/10; 10/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 8/10; 10/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.997, test=0.997) total time= 0.9s
[CV 4/10; 12/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 4/10; 12/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.997) total time= 0.7s
[CV 10/10; 13/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 10/10; 13/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.997, test=0.996) total time= 0.7s
[CV 6/10; 15/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 6/10; 15/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.995, test=0.995) total time= 0.8s
[CV 3/10; 17/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant

```

```

[CV 3/10; 17/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.996, test=0.995)
total time= 0.7s
[CV 9/10; 18/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 9/10; 18/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.996, test=0.995)
total time= 0.7s
[CV 4/10; 20/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 4/10; 20/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.990, test=0.991)
total time= 0.9s
[CV 10/10; 21/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 10/10; 21/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.991, test=0.990)
total time= 0.8s
[CV 8/10; 23/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 8/10; 23/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.7s
[CV 4/10; 25/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 4/10; 25/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.729, test=0.729)
total time= 0.9s
[CV 6/10; 26/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 6/10; 26/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.991, test=0.990)
total time= 0.8s
[CV 1/10; 28/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 1/10; 28/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.8s
[CV 5/10; 29/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 5/10; 29/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.728, test=0.727)
total time= 1.2s
[CV 4/10; 1/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 4/10; 1/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.998, test=0.998)
total time= 1.3s
[CV 4/10; 3/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 4/10; 3/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.996, test=0.997)
total time= 0.7s
[CV 2/10; 5/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 2/10; 5/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.990, test=0.992)
total time= 0.7s

```

```
time= 0.7s
[CV 6/10; 6/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 6/10; 6/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 1.0s
[CV 1/10; 8/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 1/10; 8/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.998, test=0.997)
total time= 0.8s
[CV 9/10; 9/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 9/10; 9/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.998, test=0.997)
total time= 1.1s
[CV 8/10; 11/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 8/10; 11/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 1.0s
[CV 7/10; 13/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 7/10; 13/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.997, test=0.997)
total time= 0.7s
[CV 3/10; 15/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 3/10; 15/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.996, test=0.995)
total time= 0.9s
[CV 10/10; 16/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 10/10; 16/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 0.9s
[CV 7/10; 18/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 7/10; 18/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.995, test=0.996)
total time= 0.7s
[CV 2/10; 20/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 2/10; 20/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.996, test=0.997)
total time= 0.9s
[CV 8/10; 21/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 8/10; 21/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.8s
[CV 4/10; 23/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 4/10; 23/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.991, test=0.991)
total time= 0.7s
[CV 10/10; 24/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
```

```

del__eta0=5, model__learning_rate=constant
[CV 10/10; 24/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.991, test=0.990) total time= 0.9s
[CV 5/10; 26/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 5/10; 26/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.991, test=0.989) total time= 0.8s
[CV 10/10; 27/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 10/10; 27/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.991, test=0.990) total time= 0.8s
[CV 8/10; 29/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 8/10; 29/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.693, test=0.694) total time= 1.2s
[CV 3/10; 1/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 3/10; 1/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.998, test=0.997) total time= 1.4s
[CV 2/10; 4/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 2/10; 4/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.990, test=0.992) total time= 0.7s
[CV 8/10; 5/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 8/10; 5/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.990, test=0.992) total time= 0.6s
[CV 3/10; 7/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 3/10; 7/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.998) total time= 0.9s
[CV 6/10; 8/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 6/10; 8/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.999, test=0.997) total time= 0.9s
[CV 4/10; 10/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 4/10; 10/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.997, test=0.998) total time= 1.0s
[CV 7/10; 11/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 7/10; 11/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.9s
[CV 4/10; 13/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 4/10; 13/30] END model__alpha=1e-05, model__class_weight=balanced, model

```

```

__eta0=1.0, model__learning_rate=constant;; score=(train=0.997, test=0.998)
total time= 0.7s
[CV 9/10; 14/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 9/10; 14/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.997, test=0.995) total time= 0.8s
[CV 5/10; 16/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 5/10; 16/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 0.9s
[CV 1/10; 18/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 1/10; 18/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.7s
[CV 6/10; 19/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 6/10; 19/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.995, test=0.994) total time= 0.8s
[CV 4/10; 21/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 4/10; 21/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.991, test=0.991) total time= 0.8s
[CV 10/10; 22/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 10/10; 22/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.991, test=0.990) total time= 0.7s
[CV 5/10; 24/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 5/10; 24/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.991, test=0.989) total time= 0.9s
[CV 3/10; 26/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 3/10; 26/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.991, test=0.990) total time= 0.7s
[CV 7/10; 27/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 7/10; 27/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.990, test=0.992) total time= 0.8s
[CV 4/10; 29/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 4/10; 29/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.712, test=0.715) total time= 1.1s
[CV 1/10; 1/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 1/10; 1/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.999, test=0.997) total time= 1.4s

```

```
[CV 1/10; 4/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 1/10; 4/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.990, test=0.992) total time= 0.7s
[CV 5/10; 5/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 5/10; 5/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.991, test=0.989) total time= 0.7s
[CV 1/10; 7/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 1/10; 7/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.997) total time= 0.9s
[CV 10/10; 8/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 10/10; 8/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.998, test=0.998) total time= 0.9s
[CV 6/10; 10/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 6/10; 10/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.998, test=0.997) total time= 1.0s
[CV 1/10; 12/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 1/10; 12/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.8s
[CV 2/10; 13/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 2/10; 13/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.997, test=0.997) total time= 0.8s
[CV 10/10; 14/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 10/10; 14/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 0.8s
[CV 7/10; 16/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 7/10; 16/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.9s
[CV 4/10; 18/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 4/10; 18/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.991, test=0.991) total time= 0.7s
[CV 10/10; 19/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 10/10; 19/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 0.8s
[CV 5/10; 21/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
```


[CV 5/10; 21/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.991, test=0.989) total time= 0.8s

[CV 1/10; 23/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant

[CV 1/10; 23/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.990, test=0.992) total time= 0.6s

[CV 4/10; 24/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant

[CV 4/10; 24/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.991, test=0.991) total time= 1.0s

[CV 4/10; 26/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant

[CV 4/10; 26/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.991, test=0.991) total time= 0.8s

[CV 9/10; 27/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant

[CV 9/10; 27/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.991, test=0.987) total time= 0.8s

[CV 6/10; 29/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant

[CV 6/10; 29/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.728, test=0.727) total time= 1.0s

[CV 8/10; 1/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal

[CV 8/10; 1/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.998, test=0.997) total time= 1.4s

[CV 8/10; 3/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal

[CV 8/10; 3/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.996, test=0.996) total time= 0.7s

[CV 4/10; 5/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal

[CV 4/10; 5/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.991, test=0.991) total time= 0.7s

[CV 9/10; 6/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant

[CV 9/10; 6/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.995) total time= 1.0s

[CV 5/10; 8/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant

[CV 5/10; 8/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.998, test=0.997) total time= 0.8s

[CV 8/10; 9/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant

[CV 8/10; 9/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.998, test=0.997) total time= 1.0s

```
al time= 1.0s
[CV 4/10; 11/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 4/10; 11/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.997)
total time= 1.0s
[CV 10/10; 12/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 10/10; 12/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.996)
total time= 0.7s
[CV 6/10; 14/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 6/10; 14/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 0.8s
[CV 3/10; 16/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 3/10; 16/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.995)
total time= 0.9s
[CV 9/10; 17/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 9/10; 17/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.996, test=0.995)
total time= 0.7s
[CV 5/10; 19/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 5/10; 19/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.994, test=0.994)
total time= 0.8s
[CV 2/10; 21/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 2/10; 21/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.8s
[CV 8/10; 22/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 8/10; 22/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.7s
[CV 3/10; 24/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 3/10; 24/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.990, test=0.989)
total time= 0.9s
[CV 2/10; 26/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 2/10; 26/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.7s
[CV 6/10; 27/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 6/10; 27/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.991, test=0.990)
total time= 0.8s
[CV 2/10; 29/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
```

```

l__eta0=5, model__learning_rate=constant
[CV 2/10; 29/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.693, test=0.693) total time= 1.0s
[CV 8/10; 30/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 8/10; 30/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.719, test=0.719) total time= 0.8s
[CV 9/10; 1/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 9/10; 1/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.998, test=0.996) total time= 1.4s
[CV 10/10; 3/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 10/10; 3/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.996, test=0.996) total time= 0.7s
[CV 7/10; 5/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 7/10; 5/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.990, test=0.992) total time= 0.6s
[CV 4/10; 7/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 4/10; 7/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.998) total time= 0.8s
[CV 7/10; 8/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 7/10; 8/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.998, test=0.998) total time= 0.8s
[CV 5/10; 10/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 5/10; 10/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.991, test=0.988) total time= 1.0s
[CV 9/10; 11/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 9/10; 11/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.995) total time= 1.0s
[CV 6/10; 13/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 6/10; 13/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.997, test=0.997) total time= 0.8s
[CV 1/10; 15/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 1/10; 15/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.8s
[CV 6/10; 16/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 6/10; 16/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.997, test=0.997) total time= 0.8s

```

```

l__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996
) total time= 0.9s
[CV 2/10; 18/30] START model__alpha=0.0001, model__class_weight=balanced, mo
del__eta0=1.0, model__learning_rate=constant
[CV 2/10; 18/30] END model__alpha=0.0001, model__class_weight=balanced, mode
l__eta0=1.0, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 0.7s
[CV 8/10; 19/30] START model__alpha=0.0001, model__class_weight=balanced, mo
del__eta0=5, model__learning_rate=constant
[CV 8/10; 19/30] END model__alpha=0.0001, model__class_weight=balanced, mode
l__eta0=5, model__learning_rate=constant;; score=(train=0.996, test=0.996) t
otal time= 0.7s
[CV 1/10; 21/30] START model__alpha=0.001, model__class_weight=balanced, mod
el__eta0=0.01, model__learning_rate=constant
[CV 1/10; 21/30] END model__alpha=0.001, model__class_weight=balanced, model
__eta0=0.01, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.7s
[CV 7/10; 22/30] START model__alpha=0.001, model__class_weight=balanced, mod
el__eta0=0.1, model__learning_rate=constant
[CV 7/10; 22/30] END model__alpha=0.001, model__class_weight=balanced, model
__eta0=0.1, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.7s
[CV 8/10; 24/30] START model__alpha=0.001, model__class_weight=balanced, mod
el__eta0=5, model__learning_rate=constant
[CV 8/10; 24/30] END model__alpha=0.001, model__class_weight=balanced, model
__eta0=5, model__learning_rate=constant;; score=(train=0.728, test=0.729) to
tal time= 0.7s
[CV 10/10; 25/30] START model__alpha=0.001, model__class_weight=balanced, mo
del__eta0=10, model__learning_rate=constant
[CV 10/10; 25/30] END model__alpha=0.001, model__class_weight=balanced, mode
l__eta0=10, model__learning_rate=constant;; score=(train=0.991, test=0.990)
total time= 0.9s
[CV 8/10; 27/30] START model__alpha=0.01, model__class_weight=balanced, mode
l__eta0=0.1, model__learning_rate=constant
[CV 8/10; 27/30] END model__alpha=0.01, model__class_weight=balanced, model
__eta0=0.1, model__learning_rate=constant;; score=(train=0.990, test=0.992) t
otal time= 0.8s
[CV 3/10; 29/30] START model__alpha=0.01, model__class_weight=balanced, mode
l__eta0=5, model__learning_rate=constant
[CV 3/10; 29/30] END model__alpha=0.01, model__class_weight=balanced, model
__eta0=5, model__learning_rate=constant;; score=(train=0.989, test=0.990) tot
al time= 1.0s
[CV 9/10; 30/30] START model__alpha=0.01, model__class_weight=balanced, mode
l__eta0=10, model__learning_rate=constant
[CV 9/10; 30/30] END model__alpha=0.01, model__class_weight=balanced, model
__eta0=10, model__learning_rate=constant;; score=(train=0.729, test=0.726) to
tal time= 0.6s
[CV 5/10; 1/30] START model__alpha=1e-06, model__class_weight=balanced, mode
l__eta0=0, model__learning_rate=optimal
[CV 5/10; 1/30] END model__alpha=1e-06, model__class_weight=balanced, model
__eta0=0, model__learning_rate=optimal;; score=(train=0.999, test=0.998) tota
l time= 1.3s
[CV 5/10; 3/30] START model__alpha=0.0001, model__class_weight=balanced, mod
el__eta0=0, model__learning_rate=optimal
[CV 5/10; 3/30] END model__alpha=0.0001, model__class_weight=balanced, model
__eta0=0, model__learning_rate=optimal;; score=(train=0.996, test=0.996) tot
al time= 0.7s

```

```
[CV 10/10; 4/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal
[CV 10/10; 4/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.991, test=0.990) total time= 0.7s
[CV 8/10; 6/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 8/10; 6/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 1.0s
[CV 9/10; 8/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 9/10; 8/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.998, test=0.995) total time= 0.8s
[CV 3/10; 10/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 3/10; 10/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.998, test=0.998) total time= 0.9s
[CV 6/10; 11/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 6/10; 11/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 1.0s
[CV 3/10; 13/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 3/10; 13/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.997, test=0.996) total time= 0.7s
[CV 8/10; 14/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 8/10; 14/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 0.7s
[CV 2/10; 16/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 2/10; 16/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 0.9s
[CV 8/10; 17/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 8/10; 17/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.996, test=0.996) total time= 0.7s
[CV 4/10; 19/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 4/10; 19/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.991, test=0.991) total time= 0.7s
[CV 10/10; 20/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 10/10; 20/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.729, test=0.728) total time= 0.8s
[CV 6/10; 22/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
```

[CV 6/10; 22/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.991, test=0.990) total time= 0.6s

[CV 2/10; 24/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant

[CV 2/10; 24/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.990, test=0.992) total time= 0.9s

[CV 8/10; 25/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant

[CV 8/10; 25/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.990, test=0.991) total time= 1.1s

[CV 4/10; 28/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant

[CV 4/10; 28/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.991, test=0.991) total time= 0.8s

[CV 10/10; 29/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant

[CV 10/10; 29/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.693, test=0.692) total time= 1.0s

[CV 6/10; 2/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal

[CV 6/10; 2/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.997, test=0.996) total time= 1.0s

[CV 1/10; 3/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal

[CV 1/10; 3/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.996, test=0.997) total time= 0.7s

[CV 3/10; 4/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal

[CV 3/10; 4/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0, model__learning_rate=optimal;; score=(train=0.991, test=0.990) total time= 0.7s

[CV 1/10; 6/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant

[CV 1/10; 6/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.996, test=0.997) total time= 0.9s

[CV 7/10; 7/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant

[CV 7/10; 7/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.997, test=0.997) total time= 0.7s

[CV 3/10; 9/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant

[CV 3/10; 9/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.997, test=0.996) total time= 0.9s

[CV 10/10; 10/30] START model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant

[CV 10/10; 10/30] END model__alpha=1e-06, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.998, test=0.997)

```
total time= 0.7s
[CV 5/10; 12/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 5/10; 12/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 0.6s
[CV 1/10; 14/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant
[CV 1/10; 14/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=5, model__learning_rate=constant;; score=(train=0.996, test=0.997)
total time= 0.7s
[CV 8/10; 15/30] START model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 8/10; 15/30] END model__alpha=1e-05, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 0.8s
[CV 4/10; 17/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant
[CV 4/10; 17/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=0.1, model__learning_rate=constant;; score=(train=0.996, test=0.997)
total time= 0.7s
[CV 10/10; 18/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 10/10; 18/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.996, test=0.996)
total time= 0.7s
[CV 6/10; 20/30] START model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 6/10; 20/30] END model__alpha=0.0001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.991, test=0.990)
total time= 0.8s
[CV 9/10; 21/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 9/10; 21/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.991, test=0.987)
total time= 0.8s
[CV 5/10; 23/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 5/10; 23/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.991, test=0.989)
total time= 0.7s
[CV 1/10; 25/30] START model__alpha=0.001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant
[CV 1/10; 25/30] END model__alpha=0.001, model__class_weight=balanced, model__eta0=10, model__learning_rate=constant;; score=(train=0.990, test=0.992)
total time= 1.1s
[CV 10/10; 26/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant
[CV 10/10; 26/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=0.01, model__learning_rate=constant;; score=(train=0.991, test=0.990)
total time= 0.7s
[CV 6/10; 28/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
[CV 6/10; 28/30] END model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant;; score=(train=0.991, test=0.990)
total time= 0.8s
[CV 1/10; 30/30] START model__alpha=0.01, model__class_weight=balanced, model__eta0=1.0, model__learning_rate=constant
```

```
l__eta0=10, model__learning_rate=constant  
[CV 1/10; 30/30] END model__alpha=0.01, model__class_weight=balanced, model_  
_eta0=10, model__learning_rate=constant;; score=(train=0.728, test=0.729) to  
tal time= 1.1s
```

Out[38]:

```
► GridSearchCV  
  ► estimator: SVC  
    ► SVC
```

In [39]:

```
# Confirm number of support vectors  
len(grid_search.best_estimator_.n_support_)
```

Out[39]:

2

In [40]:

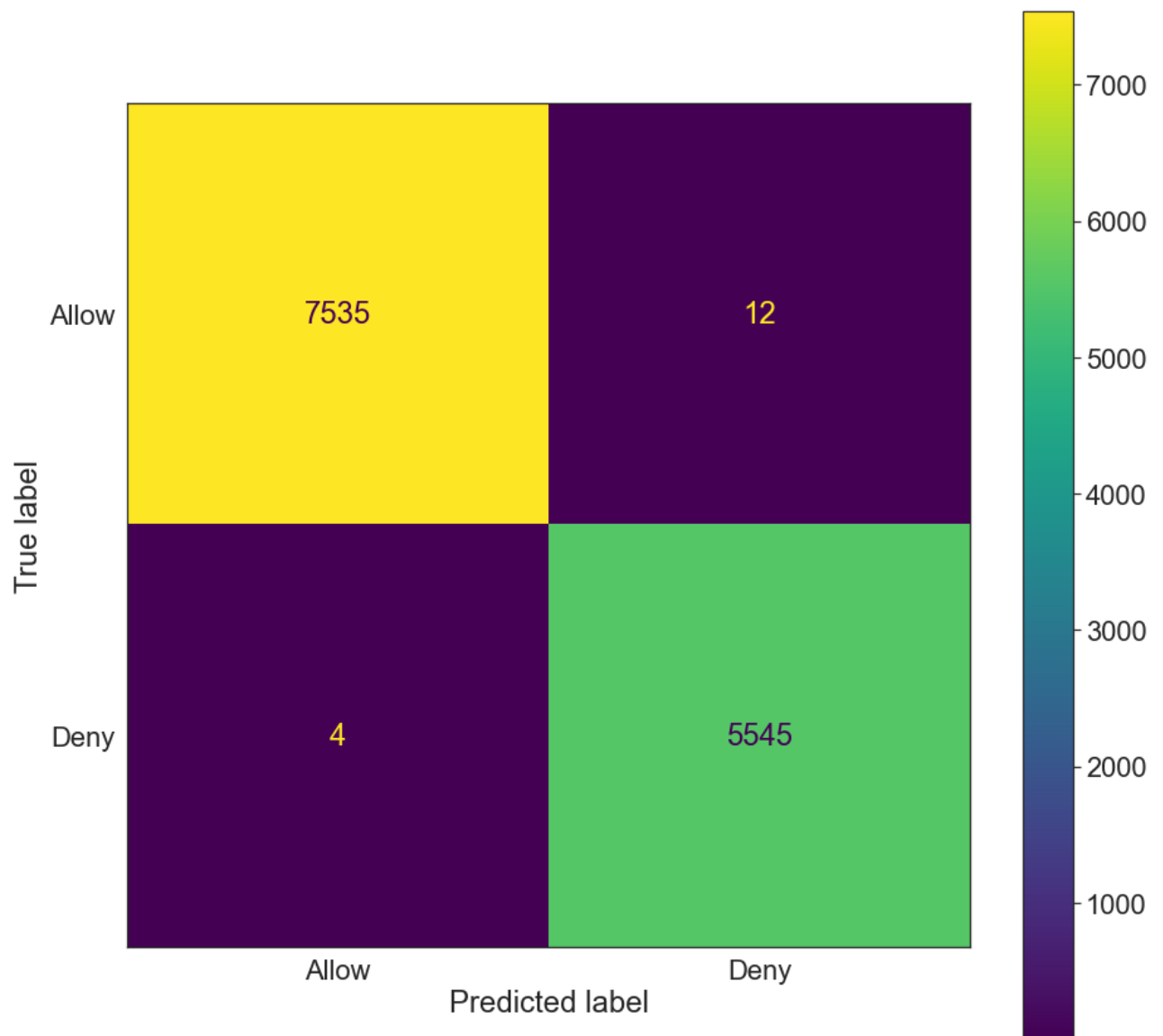
```
# Evaluate the training accuracy of the model  
y_hat_rf_train = grid_search.predict(X_train)  
accuracy_score(y_hat_rf_train, y_train)
```

Out[40]:

0.9989118399450193

In [41]:

```
# Build confusion matrix  
confusion_matrix(y_train, y_hat_rf_train)  
disp = ConfusionMatrixDisplay.from_estimator(grid_search, X_test, y_test)
```

```
In [42]: # Evaluate the test accuracy of the model
y_hat_rf_test = grid_search.predict(X_test)
accuracy_score(y_hat_rf_test, y_test)
```

```
Out[42]: 0.9987782529016493
```

```
In [43]: # Check precision and recall scores
print("Recall:", recall_score(y_test,
                              y_hat_rf_test,
                              pos_label = "Allow",
                              average = 'binary'))

print("Precision:", precision_score(y_test,
                                    y_hat_rf_test,
                                    pos_label = "Deny",
                                    average = 'binary'))
```

```
Recall: 0.998409964224195
Precision: 0.997840561454022
```

Evaluate Hyperparameters

```
In [44]: p = ['linear', 'rbf']

for i in p:
    model = SVC(kernel = i)
    model.fit(X_train, y_train)
    y_hat_rf_test = model.predict(X_test)
    print(i, ":", accuracy_score(y_hat_rf_test, y_test))

linear : 0.9987782529016493
rbf : 0.778787416004887
```

```

In [45]: sns.set(font_scale = 1)
sns.set_style(style = "whitegrid")

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

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size = 0.3, random_state = 42)

X, y = make_blobs(n_samples=400, centers=2, random_state=0)
plt.figure(figsize=(10, 5))

for i, K in enumerate(['linear', 'rbf']):
    clf = SVC(C = 1, kernel= K , random_state = 42).fit(X, y)
    decision_function = clf.decision_function(X)
    support_vector_indices = np.where(
        np.abs(decision_function) <= 1 + 1e-15)[0]
    support_vectors = X[support_vector_indices]

    plt.subplot(1, 2, i + 1)
    plt.scatter(X[:, 0],
                X[:, 1],
                c = y,
                s = 30,
                cmap = plt.cm.Paired)

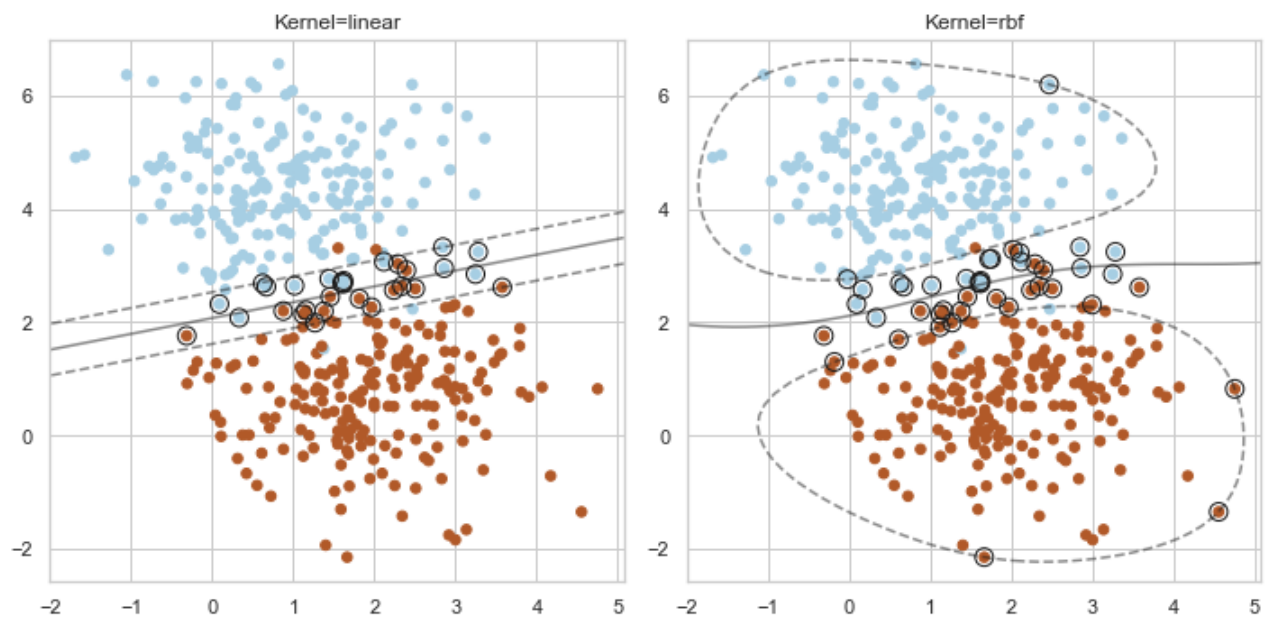
    ax = plt.gca()
    xlim = ax.get_xlim()
    ylim = ax.get_ylim()
    xx, yy = np.meshgrid(np.linspace(xlim[0], xlim[1], 50),
                          np.linspace(ylim[0], ylim[1], 50))
    Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)
    plt.contour(xx,
                yy,
                Z,
                colors = 'k',
                levels = [-1, 0, 1],
                alpha = 0.5,
                linestyles = ['--', '-', '--'])

    plt.scatter(support_vectors[:, 0],
                support_vectors[:, 1],
                s = 100,
                linewidth = 1,
                facecolors = 'none',
                edgecolors = 'k')

    plt.title("Kernel=" + str(K))

plt.tight_layout()
plt.show()

```



```
In [46]: C = [0.001, 0.01, 0.1, 1, 10, 100]
for i in C:
    model = SVC(C = i)
    model.fit(X_train, y_train)
    y_hat_rf_test = model.predict(X_test)
    print(str(i), ":", accuracy_score(y_hat_rf_test, y_test))
```

```
0.001 : 0.574628385257585
0.01  : 0.574628385257585
0.1   : 0.574628385257585
1     : 0.8639788230502953
10    : 0.9350437792710242
100   : 0.9725615964162085
```

Linear Kernel

```

In [47]: sns.set(font_scale = 1)
sns.set_style(style = "whitegrid")

X, y = make_blobs(n_samples = 100,
                  centers = 2,
                  random_state = 42)

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

for i, C in enumerate([0.001, 0.01, 0.1, 1, 10, 100]):
    clf = SVC(C = C,
              kernel = 'linear',
              random_state = 42).fit(X, y)
    decision_function = clf.decision_function(X)
    support_vector_indices = np.where(
        np.abs(decision_function) <= 1 + 1e-15)[0]
    support_vectors = X[support_vector_indices]

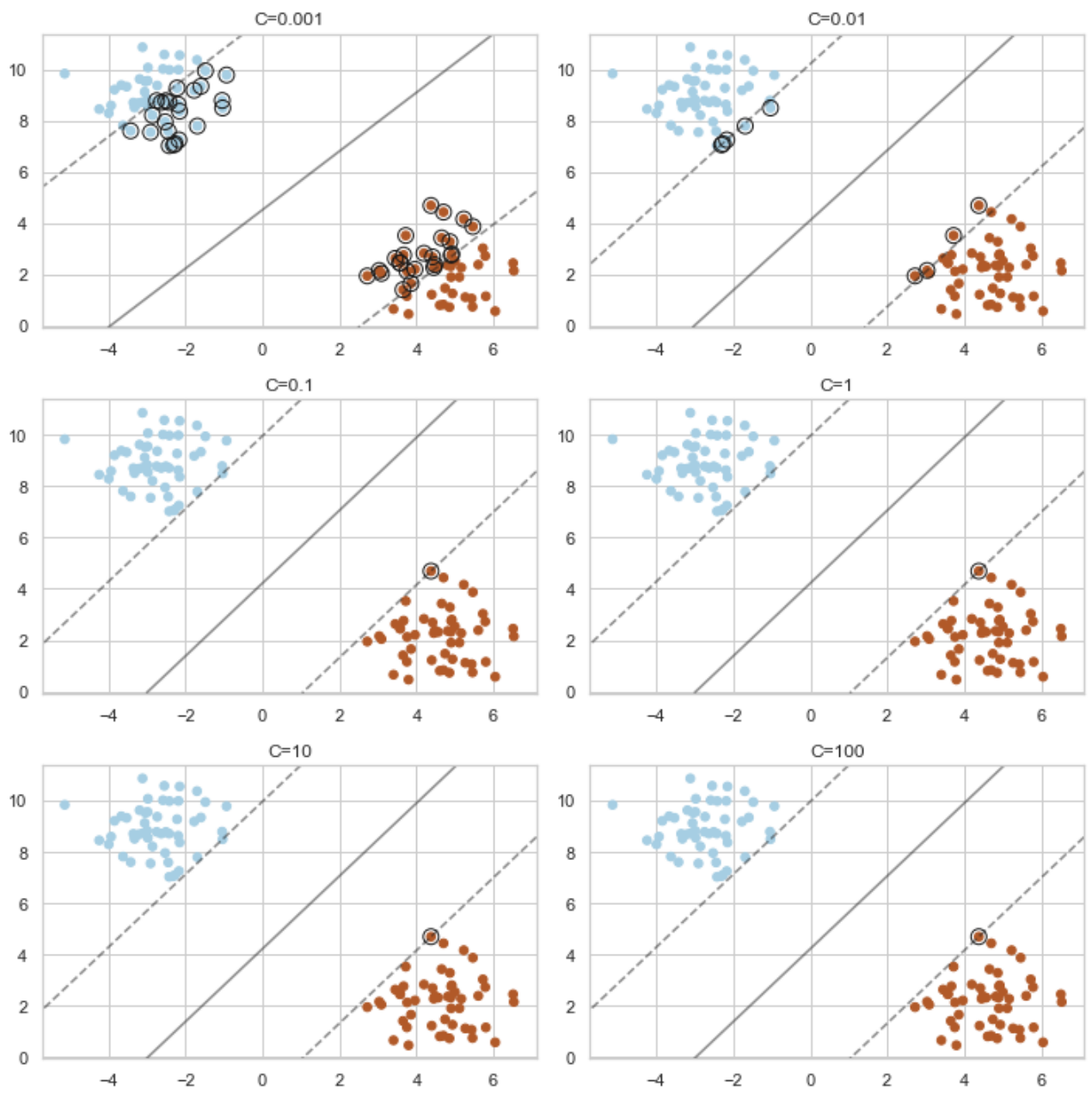
    plt.subplot(3, 2, i + 1)
    plt.scatter(X[:, 0],
                X[:, 1],
                c = y,
                s = 30,
                cmap = plt.cm.Paired)

    ax = plt.gca()
    xlim = ax.get_xlim()
    ylim = ax.get_ylim()
    xx, yy = np.meshgrid(np.linspace(xlim[0], xlim[1], 50),
                          np.linspace(ylim[0], ylim[1], 50))
    Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)
    plt.contour(xx,
                yy,
                Z,
                colors = 'k',
                levels = [-1, 0, 1],
                alpha = 0.5,
                linestyles=['--', '-', '--'])
    plt.scatter(support_vectors[:, 0],
                support_vectors[:, 1],
                s = 100,
                linewidth = 1,
                facecolors = 'none',
                edgecolors = 'k')

    plt.title("C=" + str(C))

plt.tight_layout()
plt.show()

```



RBF Kernel

```

In [48]: sns.set(font_scale = 1)
sns.set_style(style = "whitegrid")

X, y = make_blobs(n_samples = 100,
                  centers = 2,
                  random_state = 42)

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

for i, C in enumerate([0.001, 0.01, 0.1, 1, 10, 100]):
    clf = SVC(C = C,
              kernel = 'rbf',
              random_state = 42).fit(X, y)
    decision_function = clf.decision_function(X)
    support_vector_indices = np.where(
        np.abs(decision_function) <= 1 + 1e-15)[0]
    support_vectors = X[support_vector_indices]

    plt.subplot(3, 2, i + 1)
    plt.scatter(X[:, 0],
                X[:, 1],
                c = y,
                s = 30,
                cmap = plt.cm.Paired)

    ax = plt.gca()
    xlim = ax.get_xlim()
    ylim = ax.get_ylim()
    xx, yy = np.meshgrid(np.linspace(xlim[0], xlim[1], 50),
                          np.linspace(ylim[0], ylim[1], 50))
    Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)
    plt.contour(xx,
                yy,
                Z,
                colors = 'k',
                levels = [-1, 0, 1],
                alpha = 0.5,
                linestyles=['--', '-', '--'])
    plt.scatter(support_vectors[:, 0],
                support_vectors[:, 1],
                s = 100,
                linewidth = 1,
                facecolors = 'none',
                edgecolors = 'k')

    plt.title("C=" + str(C))

plt.tight_layout()
plt.show()

```

