

Import Modules

```
In [ ]: from luma.preprocessing.scaler import StandardScaler
from luma.reduction.linear import PCA
from luma.model_selection.split import TrainTestSplit
from luma.model_selection.search import GridSearchCV
from luma.classifier.logistic import SoftmaxRegressor
from luma.metric.classification import Accuracy
from luma.interface.util import Clone
from luma.visual.evaluation import (ConfusionMatrix,
                                   LearningCurve,
                                   ValidationCurve,
                                   ValidationHeatmap)

from sklearn.datasets import fetch_lfw_people
import matplotlib.pyplot as plt
import numpy as np
```

Load Dataset

```
In [ ]: lfw_people = fetch_lfw_people(min_faces_per_person=100, resize=0.4)
X = lfw_people.data
y = lfw_people.target

lfw_people.images.shape, X.shape, y.shape
```

```
Out[ ]: ((1140, 50, 37), (1140, 1850), (1140,))
```

Split Train and Test Sets

```
In [ ]: X_train, X_test, y_train, y_test = TrainTestSplit(X, y,
                                                         test_size=0.2,
                                                         random_state=42).get

X_train.shape, X_test.shape
```

```
Out[ ]: ((912, 1850), (228, 1850))
```

Scale Through StandardScaler

```
In [ ]: sc = StandardScaler()
X_train_std = sc.fit_transform(X_train)
X_test_std = sc.fit_transform(X_test)
```

Dimensionality Reduction Through PCA

```
In [ ]: pca = PCA(n_components=150)
pca.fit(X_train)

X_train_pca = pca.transform(X_train_std)
X_test_pca = pca.transform(X_test_std)
```

Tuning Hyperparameters of SoftmaxRegressor with GridSearchCV

```
In [ ]: param_grid = {'learning_rate': np.logspace(-3, 0, 8),
                     'alpha': np.logspace(-3, 0, 8)}

grid = GridSearchCV(estimator=SoftmaxRegressor(regularization='elastic-net'),
                    param_grid=param_grid,
                    cv=5,
                    refit=True,
                    random_state=42,
                    verbose=True)

grid.fit(X_train_pca, y_train)
```

Fitting 5 folds for 64 candidates, totalling 320 fits.

```
[CV] fold 1 - train-score: 0.955, test-score: 0.764
[CV] fold 2 - train-score: 0.944, test-score: 0.819
[CV] fold 3 - train-score: 0.651, test-score: 0.659
[CV] fold 4 - train-score: 0.851, test-score: 0.769
[CV] fold 5 - train-score: 0.824, test-score: 0.712
[GridSearchCV] candidate 1/64 {'learning_rate': 0.001, 'alpha': 0.001} - score: 0.745
[CV] fold 1 - train-score: 0.948, test-score: 0.824
[CV] fold 2 - train-score: 0.941, test-score: 0.841
[CV] fold 3 - train-score: 0.823, test-score: 0.692
[CV] fold 4 - train-score: 0.893, test-score: 0.769
[CV] fold 5 - train-score: 0.714, test-score: 0.630
[GridSearchCV] candidate 2/64 {'learning_rate': 0.001, 'alpha': 0.0026826957952797246} - score: 0.751
[CV] fold 1 - train-score: 0.940, test-score: 0.819
[CV] fold 2 - train-score: 0.945, test-score: 0.802
[CV] fold 3 - train-score: 0.960, test-score: 0.841
[CV] fold 4 - train-score: 0.953, test-score: 0.830
[CV] fold 5 - train-score: 0.799, test-score: 0.690
[GridSearchCV] candidate 3/64 {'learning_rate': 0.001, 'alpha': 0.0071968567300115215} - score: 0.796
[CV] fold 1 - train-score: 0.901, test-score: 0.780
[CV] fold 2 - train-score: 0.819, test-score: 0.714
[CV] fold 3 - train-score: 0.964, test-score: 0.786
[CV] fold 4 - train-score: 0.948, test-score: 0.846
[CV] fold 5 - train-score: 0.941, test-score: 0.859
[GridSearchCV] candidate 4/64 {'learning_rate': 0.001, 'alpha': 0.019306977288832496} - score: 0.797
[CV] fold 1 - train-score: 0.700, test-score: 0.747
[CV] fold 2 - train-score: 0.640, test-score: 0.593
[CV] fold 3 - train-score: 0.763, test-score: 0.643
```

[CV] fold 4 - train-score: 0.704, test-score: 0.659
[CV] fold 5 - train-score: 0.799, test-score: 0.707
[GridSearchCV] candidate 5/64 {'learning_rate': 0.001, 'alpha': 0.0517947467923121} - score: 0.670
[CV] fold 1 - train-score: 0.438, test-score: 0.571
[CV] fold 2 - train-score: 0.460, test-score: 0.484
[CV] fold 3 - train-score: 0.471, test-score: 0.440
[CV] fold 4 - train-score: 0.474, test-score: 0.429
[CV] fold 5 - train-score: 0.481, test-score: 0.402
[GridSearchCV] candidate 6/64 {'learning_rate': 0.001, 'alpha': 0.13894954943731375} - score: 0.465
[CV] fold 1 - train-score: 0.468, test-score: 0.451
[CV] fold 2 - train-score: 0.474, test-score: 0.429
[CV] fold 3 - train-score: 0.462, test-score: 0.478
[CV] fold 4 - train-score: 0.458, test-score: 0.495
[CV] fold 5 - train-score: 0.463, test-score: 0.473
[GridSearchCV] candidate 7/64 {'learning_rate': 0.001, 'alpha': 0.3727593720314938} - score: 0.465
[CV] fold 1 - train-score: 0.466, test-score: 0.462
[CV] fold 2 - train-score: 0.459, test-score: 0.489
[CV] fold 3 - train-score: 0.453, test-score: 0.511
[CV] fold 4 - train-score: 0.488, test-score: 0.374
[CV] fold 5 - train-score: 0.459, test-score: 0.489
[GridSearchCV] candidate 8/64 {'learning_rate': 0.001, 'alpha': 1.0} - score: 0.465
[CV] fold 1 - train-score: 0.890, test-score: 0.824
[CV] fold 2 - train-score: 0.944, test-score: 0.857
[CV] fold 3 - train-score: 0.930, test-score: 0.846
[CV] fold 4 - train-score: 0.778, test-score: 0.692
[CV] fold 5 - train-score: 0.786, test-score: 0.717
[GridSearchCV] candidate 9/64 {'learning_rate': 0.0026826957952797246, 'alpha': 0.001} - score: 0.787
[CV] fold 1 - train-score: 0.964, test-score: 0.791
[CV] fold 2 - train-score: 0.956, test-score: 0.791
[CV] fold 3 - train-score: 0.849, test-score: 0.775
[CV] fold 4 - train-score: 0.948, test-score: 0.808
[CV] fold 5 - train-score: 0.934, test-score: 0.853
[GridSearchCV] candidate 10/64 {'learning_rate': 0.0026826957952797246, 'alpha': 0.0026826957952797246} - score: 0.804
[CV] fold 1 - train-score: 0.612, test-score: 0.511
[CV] fold 2 - train-score: 0.941, test-score: 0.824
[CV] fold 3 - train-score: 0.756, test-score: 0.665
[CV] fold 4 - train-score: 0.968, test-score: 0.769
[CV] fold 5 - train-score: 0.944, test-score: 0.832
[GridSearchCV] candidate 11/64 {'learning_rate': 0.0026826957952797246, 'alpha': 0.0071968567300115215} - score: 0.720
[CV] fold 1 - train-score: 0.752, test-score: 0.632
[CV] fold 2 - train-score: 0.947, test-score: 0.824
[CV] fold 3 - train-score: 0.790, test-score: 0.747
[CV] fold 4 - train-score: 0.959, test-score: 0.830
[CV] fold 5 - train-score: 0.665, test-score: 0.582
[GridSearchCV] candidate 12/64 {'learning_rate': 0.0026826957952797246, 'alpha': 0.019306977288832496} - score: 0.723
[CV] fold 1 - train-score: 0.807, test-score: 0.714
[CV] fold 2 - train-score: 0.627, test-score: 0.560
[CV] fold 3 - train-score: 0.671, test-score: 0.654
[CV] fold 4 - train-score: 0.704, test-score: 0.593
[CV] fold 5 - train-score: 0.701, test-score: 0.674
[GridSearchCV] candidate 13/64 {'learning_rate': 0.0026826957952797246, 'alpha': 0.0517947467923121} - score: 0.639
[CV] fold 1 - train-score: 0.471, test-score: 0.440
[CV] fold 2 - train-score: 0.460, test-score: 0.484
[CV] fold 3 - train-score: 0.466, test-score: 0.462
[CV] fold 4 - train-score: 0.467, test-score: 0.456
[CV] fold 5 - train-score: 0.460, test-score: 0.484
[GridSearchCV] candidate 14/64 {'learning_rate': 0.0026826957952797246, 'alpha': 0.13894954943731375} - score: 0.465
[CV] fold 1 - train-score: 0.449, test-score: 0.527
[CV] fold 2 - train-score: 0.467, test-score: 0.456
[CV] fold 3 - train-score: 0.463, test-score: 0.473
[CV] fold 4 - train-score: 0.473, test-score: 0.434
[CV] fold 5 - train-score: 0.473, test-score: 0.435
[GridSearchCV] candidate 15/64 {'learning_rate': 0.0026826957952797246, 'alpha': 0.3727593720314938} - score: 0.465
[CV] fold 1 - train-score: 0.466, test-score: 0.462
[CV] fold 2 - train-score: 0.470, test-score: 0.445
[CV] fold 3 - train-score: 0.455, test-score: 0.505
[CV] fold 4 - train-score: 0.470, test-score: 0.445
[CV] fold 5 - train-score: 0.464, test-score: 0.467
[GridSearchCV] candidate 16/64 {'learning_rate': 0.0026826957952797246, 'alpha': 1.0} - score: 0.465
[CV] fold 1 - train-score: 0.912, test-score: 0.808
[CV] fold 2 - train-score: 0.773, test-score: 0.725
[CV] fold 3 - train-score: 0.926, test-score: 0.775
[CV] fold 4 - train-score: 0.953, test-score: 0.808
[CV] fold 5 - train-score: 0.941, test-score: 0.821
[GridSearchCV] candidate 17/64 {'learning_rate': 0.0071968567300115215, 'alpha': 0.001} - score: 0.787
[CV] fold 1 - train-score: 0.797, test-score: 0.731
[CV] fold 2 - train-score: 0.932, test-score: 0.868
[CV] fold 3 - train-score: 0.960, test-score: 0.797
[CV] fold 4 - train-score: 0.930, test-score: 0.791
[CV] fold 5 - train-score: 0.933, test-score: 0.783
[GridSearchCV] candidate 18/64 {'learning_rate': 0.0071968567300115215, 'alpha': 0.0026826957952797246} - score: 0.794
[CV] fold 1 - train-score: 0.841, test-score: 0.791
[CV] fold 2 - train-score: 0.671, test-score: 0.527
[CV] fold 3 - train-score: 0.914, test-score: 0.797
[CV] fold 4 - train-score: 0.923, test-score: 0.813
[CV] fold 5 - train-score: 0.839, test-score: 0.745
[GridSearchCV] candidate 19/64 {'learning_rate': 0.0071968567300115215, 'alpha': 0.0071968567300115215} - score: 0.735
[CV] fold 1 - train-score: 0.766, test-score: 0.665
[CV] fold 2 - train-score: 0.953, test-score: 0.830
[CV] fold 3 - train-score: 0.947, test-score: 0.841
[CV] fold 4 - train-score: 0.922, test-score: 0.769
[CV] fold 5 - train-score: 0.837, test-score: 0.745
[GridSearchCV] candidate 20/64 {'learning_rate': 0.0071968567300115215, 'alpha': 0.019306977288832496} - score: 0.770
[CV] fold 1 - train-score: 0.810, test-score: 0.720
[CV] fold 2 - train-score: 0.852, test-score: 0.753
[CV] fold 3 - train-score: 0.742, test-score: 0.648
[CV] fold 4 - train-score: 0.682, test-score: 0.692
[CV] fold 5 - train-score: 0.720, test-score: 0.598
[GridSearchCV] candidate 21/64 {'learning_rate': 0.0071968567300115215, 'alpha': 0.0517947467923121} - score: 0.682
[CV] fold 1 - train-score: 0.460, test-score: 0.484
[CV] fold 2 - train-score: 0.466, test-score: 0.462
[CV] fold 3 - train-score: 0.468, test-score: 0.451
[CV] fold 4 - train-score: 0.456, test-score: 0.500
[CV] fold 5 - train-score: 0.474, test-score: 0.429
[GridSearchCV] candidate 22/64 {'learning_rate': 0.0071968567300115215, 'alpha': 0.13894954943731375} - score: 0.465
[CV] fold 1 - train-score: 0.452, test-score: 0.516
[CV] fold 2 - train-score: 0.466, test-score: 0.462

[CV] fold 3 - train-score: 0.478, test-score: 0.412
[CV] fold 4 - train-score: 0.458, test-score: 0.495
[CV] fold 5 - train-score: 0.471, test-score: 0.440
[GridSearchCV] candidate 23/64 {'learning_rate': 0.0071968567300115215, 'alpha': 0.3727593720314938} - score: 0.465
[CV] fold 1 - train-score: 0.456, test-score: 0.500
[CV] fold 2 - train-score: 0.474, test-score: 0.429
[CV] fold 3 - train-score: 0.460, test-score: 0.484
[CV] fold 4 - train-score: 0.464, test-score: 0.467
[CV] fold 5 - train-score: 0.470, test-score: 0.446
[GridSearchCV] candidate 24/64 {'learning_rate': 0.0071968567300115215, 'alpha': 1.0} - score: 0.465
[CV] fold 1 - train-score: 0.671, test-score: 0.560
[CV] fold 2 - train-score: 0.892, test-score: 0.841
[CV] fold 3 - train-score: 0.875, test-score: 0.775
[CV] fold 4 - train-score: 0.968, test-score: 0.780
[CV] fold 5 - train-score: 0.762, test-score: 0.658
[GridSearchCV] candidate 25/64 {'learning_rate': 0.019306977288832496, 'alpha': 0.001} - score: 0.723
[CV] fold 1 - train-score: 0.811, test-score: 0.753
[CV] fold 2 - train-score: 0.760, test-score: 0.692
[CV] fold 3 - train-score: 0.927, test-score: 0.824
[CV] fold 4 - train-score: 0.814, test-score: 0.709
[CV] fold 5 - train-score: 0.926, test-score: 0.837
[GridSearchCV] candidate 26/64 {'learning_rate': 0.019306977288832496, 'alpha': 0.0026826957952797246} - score: 0.763
[CV] fold 1 - train-score: 0.715, test-score: 0.637
[CV] fold 2 - train-score: 0.938, test-score: 0.819
[CV] fold 3 - train-score: 0.953, test-score: 0.802
[CV] fold 4 - train-score: 0.803, test-score: 0.709
[CV] fold 5 - train-score: 0.913, test-score: 0.832
[GridSearchCV] candidate 27/64 {'learning_rate': 0.019306977288832496, 'alpha': 0.0071968567300115215} - score: 0.760
[CV] fold 1 - train-score: 0.895, test-score: 0.747
[CV] fold 2 - train-score: 0.792, test-score: 0.692
[CV] fold 3 - train-score: 0.937, test-score: 0.808
[CV] fold 4 - train-score: 0.952, test-score: 0.819
[CV] fold 5 - train-score: 0.929, test-score: 0.842
[GridSearchCV] candidate 28/64 {'learning_rate': 0.019306977288832496, 'alpha': 0.019306977288832496} - score: 0.782
[CV] fold 1 - train-score: 0.730, test-score: 0.714
[CV] fold 2 - train-score: 0.766, test-score: 0.665
[CV] fold 3 - train-score: 0.768, test-score: 0.725
[CV] fold 4 - train-score: 0.716, test-score: 0.648
[CV] fold 5 - train-score: 0.716, test-score: 0.685
[GridSearchCV] candidate 29/64 {'learning_rate': 0.019306977288832496, 'alpha': 0.0517947467923121} - score: 0.688
[CV] fold 1 - train-score: 0.468, test-score: 0.451
[CV] fold 2 - train-score: 0.479, test-score: 0.407
[CV] fold 3 - train-score: 0.458, test-score: 0.495
[CV] fold 4 - train-score: 0.466, test-score: 0.462
[CV] fold 5 - train-score: 0.453, test-score: 0.511
[GridSearchCV] candidate 30/64 {'learning_rate': 0.019306977288832496, 'alpha': 0.13894954943731375} - score: 0.465
[CV] fold 1 - train-score: 0.464, test-score: 0.467
[CV] fold 2 - train-score: 0.466, test-score: 0.462
[CV] fold 3 - train-score: 0.471, test-score: 0.440
[CV] fold 4 - train-score: 0.459, test-score: 0.489
[CV] fold 5 - train-score: 0.464, test-score: 0.467
[GridSearchCV] candidate 31/64 {'learning_rate': 0.019306977288832496, 'alpha': 0.3727593720314938} - score: 0.465
[CV] fold 1 - train-score: 0.468, test-score: 0.451
[CV] fold 2 - train-score: 0.466, test-score: 0.462
[CV] fold 3 - train-score: 0.462, test-score: 0.478
[CV] fold 4 - train-score: 0.470, test-score: 0.445
[CV] fold 5 - train-score: 0.459, test-score: 0.489
[GridSearchCV] candidate 32/64 {'learning_rate': 0.019306977288832496, 'alpha': 1.0} - score: 0.465
[CV] fold 1 - train-score: 0.860, test-score: 0.780
[CV] fold 2 - train-score: 0.938, test-score: 0.852
[CV] fold 3 - train-score: 0.926, test-score: 0.808
[CV] fold 4 - train-score: 0.955, test-score: 0.791
[CV] fold 5 - train-score: 0.856, test-score: 0.761
[GridSearchCV] candidate 33/64 {'learning_rate': 0.0517947467923121, 'alpha': 0.001} - score: 0.798
[CV] fold 1 - train-score: 0.708, test-score: 0.637
[CV] fold 2 - train-score: 0.956, test-score: 0.802
[CV] fold 3 - train-score: 0.948, test-score: 0.830
[CV] fold 4 - train-score: 0.859, test-score: 0.753
[CV] fold 5 - train-score: 0.853, test-score: 0.815
[GridSearchCV] candidate 34/64 {'learning_rate': 0.0517947467923121, 'alpha': 0.0026826957952797246} - score: 0.767
[CV] fold 1 - train-score: 0.945, test-score: 0.841
[CV] fold 2 - train-score: 0.923, test-score: 0.786
[CV] fold 3 - train-score: 0.927, test-score: 0.885
[CV] fold 4 - train-score: 0.810, test-score: 0.714
[CV] fold 5 - train-score: 0.890, test-score: 0.772
[GridSearchCV] candidate 35/64 {'learning_rate': 0.0517947467923121, 'alpha': 0.0071968567300115215} - score: 0.799
[CV] fold 1 - train-score: 0.782, test-score: 0.670
[CV] fold 2 - train-score: 0.944, test-score: 0.802
[CV] fold 3 - train-score: 0.692, test-score: 0.610
[CV] fold 4 - train-score: 0.918, test-score: 0.835
[CV] fold 5 - train-score: 0.942, test-score: 0.788
[GridSearchCV] candidate 36/64 {'learning_rate': 0.0517947467923121, 'alpha': 0.019306977288832496} - score: 0.741
[CV] fold 1 - train-score: 0.873, test-score: 0.802
[CV] fold 2 - train-score: 0.847, test-score: 0.725
[CV] fold 3 - train-score: 0.730, test-score: 0.676
[CV] fold 4 - train-score: 0.655, test-score: 0.615
[CV] fold 5 - train-score: 0.650, test-score: 0.549
[GridSearchCV] candidate 37/64 {'learning_rate': 0.0517947467923121, 'alpha': 0.0517947467923121} - score: 0.674
[CV] fold 1 - train-score: 0.452, test-score: 0.516
[CV] fold 2 - train-score: 0.484, test-score: 0.390
[CV] fold 3 - train-score: 0.460, test-score: 0.484
[CV] fold 4 - train-score: 0.456, test-score: 0.500
[CV] fold 5 - train-score: 0.473, test-score: 0.435
[GridSearchCV] candidate 38/64 {'learning_rate': 0.0517947467923121, 'alpha': 0.13894954943731375} - score: 0.465
[CV] fold 1 - train-score: 0.455, test-score: 0.505
[CV] fold 2 - train-score: 0.477, test-score: 0.418
[CV] fold 3 - train-score: 0.468, test-score: 0.451
[CV] fold 4 - train-score: 0.453, test-score: 0.511
[CV] fold 5 - train-score: 0.471, test-score: 0.440
[GridSearchCV] candidate 39/64 {'learning_rate': 0.0517947467923121, 'alpha': 0.3727593720314938} - score: 0.465
[CV] fold 1 - train-score: 0.475, test-score: 0.423
[CV] fold 2 - train-score: 0.453, test-score: 0.511
[CV] fold 3 - train-score: 0.471, test-score: 0.440
[CV] fold 4 - train-score: 0.473, test-score: 0.434
[CV] fold 5 - train-score: 0.452, test-score: 0.516
[GridSearchCV] candidate 40/64 {'learning_rate': 0.0517947467923121, 'alpha': 1.0} - score: 0.465
[CV] fold 1 - train-score: 0.819, test-score: 0.709

[CV] fold 2 - train-score: 0.786, test-score: 0.720
[CV] fold 3 - train-score: 0.933, test-score: 0.846
[CV] fold 4 - train-score: 0.947, test-score: 0.819
[CV] fold 5 - train-score: 0.918, test-score: 0.832
[GridSearchCV] candidate 41/64 {'learning_rate': 0.13894954943731375, 'alpha': 0.001} - score: 0.785
[CV] fold 1 - train-score: 0.949, test-score: 0.802
[CV] fold 2 - train-score: 0.945, test-score: 0.764
[CV] fold 3 - train-score: 0.723, test-score: 0.703
[CV] fold 4 - train-score: 0.864, test-score: 0.753
[CV] fold 5 - train-score: 0.904, test-score: 0.848
[GridSearchCV] candidate 42/64 {'learning_rate': 0.13894954943731375, 'alpha': 0.0026826957952797246} - score: 0.774
[CV] fold 1 - train-score: 0.958, test-score: 0.813
[CV] fold 2 - train-score: 0.964, test-score: 0.846
[CV] fold 3 - train-score: 0.808, test-score: 0.659
[CV] fold 4 - train-score: 0.726, test-score: 0.593
[CV] fold 5 - train-score: 0.804, test-score: 0.712
[GridSearchCV] candidate 43/64 {'learning_rate': 0.13894954943731375, 'alpha': 0.0071968567300115215} - score: 0.725
[CV] fold 1 - train-score: 0.799, test-score: 0.676
[CV] fold 2 - train-score: 0.833, test-score: 0.764
[CV] fold 3 - train-score: 0.956, test-score: 0.813
[CV] fold 4 - train-score: 0.863, test-score: 0.764
[CV] fold 5 - train-score: 0.948, test-score: 0.864
[GridSearchCV] candidate 44/64 {'learning_rate': 0.13894954943731375, 'alpha': 0.019306977288832496} - score: 0.776
[CV] fold 1 - train-score: 0.819, test-score: 0.720
[CV] fold 2 - train-score: 0.868, test-score: 0.775
[CV] fold 3 - train-score: 0.866, test-score: 0.775
[CV] fold 4 - train-score: 0.870, test-score: 0.780
[CV] fold 5 - train-score: 0.734, test-score: 0.679
[GridSearchCV] candidate 45/64 {'learning_rate': 0.13894954943731375, 'alpha': 0.0517947467923121} - score: 0.746
[CV] fold 1 - train-score: 0.477, test-score: 0.418
[CV] fold 2 - train-score: 0.458, test-score: 0.495
[CV] fold 3 - train-score: 0.463, test-score: 0.473
[CV] fold 4 - train-score: 0.474, test-score: 0.429
[CV] fold 5 - train-score: 0.453, test-score: 0.511
[GridSearchCV] candidate 46/64 {'learning_rate': 0.13894954943731375, 'alpha': 0.13894954943731375} - score: 0.465
[CV] fold 1 - train-score: 0.455, test-score: 0.505
[CV] fold 2 - train-score: 0.471, test-score: 0.440
[CV] fold 3 - train-score: 0.459, test-score: 0.489
[CV] fold 4 - train-score: 0.463, test-score: 0.473
[CV] fold 5 - train-score: 0.477, test-score: 0.418
[GridSearchCV] candidate 47/64 {'learning_rate': 0.13894954943731375, 'alpha': 0.3727593720314938} - score: 0.465
[CV] fold 1 - train-score: 0.470, test-score: 0.445
[CV] fold 2 - train-score: 0.468, test-score: 0.451
[CV] fold 3 - train-score: 0.462, test-score: 0.478
[CV] fold 4 - train-score: 0.473, test-score: 0.434
[CV] fold 5 - train-score: 0.452, test-score: 0.516
[GridSearchCV] candidate 48/64 {'learning_rate': 0.13894954943731375, 'alpha': 1.0} - score: 0.465
[CV] fold 1 - train-score: 0.753, test-score: 0.648
[CV] fold 2 - train-score: 0.700, test-score: 0.621
[CV] fold 3 - train-score: 0.977, test-score: 0.786
[CV] fold 4 - train-score: 0.870, test-score: 0.791
[CV] fold 5 - train-score: 0.677, test-score: 0.603
[GridSearchCV] candidate 49/64 {'learning_rate': 0.3727593720314938, 'alpha': 0.001} - score: 0.690
[CV] fold 1 - train-score: 0.927, test-score: 0.813
[CV] fold 2 - train-score: 0.947, test-score: 0.846
[CV] fold 3 - train-score: 0.930, test-score: 0.802
[CV] fold 4 - train-score: 0.949, test-score: 0.830
[CV] fold 5 - train-score: 0.926, test-score: 0.848
[GridSearchCV] candidate 50/64 {'learning_rate': 0.3727593720314938, 'alpha': 0.0026826957952797246} - score: 0.828
[CV] fold 1 - train-score: 0.804, test-score: 0.758
[CV] fold 2 - train-score: 0.908, test-score: 0.819
[CV] fold 3 - train-score: 0.956, test-score: 0.780
[CV] fold 4 - train-score: 0.692, test-score: 0.566
[CV] fold 5 - train-score: 0.938, test-score: 0.810
[GridSearchCV] candidate 51/64 {'learning_rate': 0.3727593720314938, 'alpha': 0.0071968567300115215} - score: 0.747
[CV] fold 1 - train-score: 0.782, test-score: 0.687
[CV] fold 2 - train-score: 0.777, test-score: 0.714
[CV] fold 3 - train-score: 0.944, test-score: 0.879
[CV] fold 4 - train-score: 0.655, test-score: 0.621
[CV] fold 5 - train-score: 0.959, test-score: 0.832
[GridSearchCV] candidate 52/64 {'learning_rate': 0.3727593720314938, 'alpha': 0.019306977288832496} - score: 0.747
[CV] fold 1 - train-score: 0.858, test-score: 0.813
[CV] fold 2 - train-score: 0.855, test-score: 0.736
[CV] fold 3 - train-score: 0.697, test-score: 0.560
[CV] fold 4 - train-score: 0.688, test-score: 0.665
[CV] fold 5 - train-score: 0.865, test-score: 0.745
[GridSearchCV] candidate 53/64 {'learning_rate': 0.3727593720314938, 'alpha': 0.0517947467923121} - score: 0.704
[CV] fold 1 - train-score: 0.458, test-score: 0.495
[CV] fold 2 - train-score: 0.464, test-score: 0.467
[CV] fold 3 - train-score: 0.449, test-score: 0.527
[CV] fold 4 - train-score: 0.474, test-score: 0.429
[CV] fold 5 - train-score: 0.479, test-score: 0.408
[GridSearchCV] candidate 54/64 {'learning_rate': 0.3727593720314938, 'alpha': 0.13894954943731375} - score: 0.465
[CV] fold 1 - train-score: 0.468, test-score: 0.451
[CV] fold 2 - train-score: 0.470, test-score: 0.445
[CV] fold 3 - train-score: 0.463, test-score: 0.473
[CV] fold 4 - train-score: 0.466, test-score: 0.462
[CV] fold 5 - train-score: 0.457, test-score: 0.495
[GridSearchCV] candidate 55/64 {'learning_rate': 0.3727593720314938, 'alpha': 0.3727593720314938} - score: 0.465
[CV] fold 1 - train-score: 0.470, test-score: 0.445
[CV] fold 2 - train-score: 0.453, test-score: 0.511
[CV] fold 3 - train-score: 0.456, test-score: 0.500
[CV] fold 4 - train-score: 0.478, test-score: 0.412
[CV] fold 5 - train-score: 0.467, test-score: 0.457
[GridSearchCV] candidate 56/64 {'learning_rate': 0.3727593720314938, 'alpha': 1.0} - score: 0.465
[CV] fold 1 - train-score: 0.949, test-score: 0.819
[CV] fold 2 - train-score: 0.959, test-score: 0.780
[CV] fold 3 - train-score: 0.934, test-score: 0.824
[CV] fold 4 - train-score: 0.681, test-score: 0.654
[CV] fold 5 - train-score: 0.620, test-score: 0.603
[GridSearchCV] candidate 57/64 {'learning_rate': 1.0, 'alpha': 0.001} - score: 0.736
[CV] fold 1 - train-score: 0.958, test-score: 0.780
[CV] fold 2 - train-score: 0.671, test-score: 0.626
[CV] fold 3 - train-score: 0.940, test-score: 0.841
[CV] fold 4 - train-score: 0.826, test-score: 0.709
[CV] fold 5 - train-score: 0.938, test-score: 0.821
[GridSearchCV] candidate 58/64 {'learning_rate': 1.0, 'alpha': 0.0026826957952797246} - score: 0.755

```
[CV] fold 1 - train-score: 0.822, test-score: 0.698
[CV] fold 2 - train-score: 0.929, test-score: 0.764
[CV] fold 3 - train-score: 0.911, test-score: 0.846
[CV] fold 4 - train-score: 0.949, test-score: 0.857
[CV] fold 5 - train-score: 0.798, test-score: 0.723
[GridSearchCV] candidate 59/64 {'learning_rate': 1.0, 'alpha': 0.0071968567300115215} - score: 0.778
[CV] fold 1 - train-score: 0.959, test-score: 0.835
[CV] fold 2 - train-score: 0.941, test-score: 0.819
[CV] fold 3 - train-score: 0.952, test-score: 0.786
[CV] fold 4 - train-score: 0.941, test-score: 0.819
[CV] fold 5 - train-score: 0.680, test-score: 0.620
[GridSearchCV] candidate 60/64 {'learning_rate': 1.0, 'alpha': 0.019306977288832496} - score: 0.776
[CV] fold 1 - train-score: 0.867, test-score: 0.791
[CV] fold 2 - train-score: 0.690, test-score: 0.687
[CV] fold 3 - train-score: 0.755, test-score: 0.703
[CV] fold 4 - train-score: 0.690, test-score: 0.681
[CV] fold 5 - train-score: 0.885, test-score: 0.739
[GridSearchCV] candidate 61/64 {'learning_rate': 1.0, 'alpha': 0.0517947467923121} - score: 0.720
[CV] fold 1 - train-score: 0.453, test-score: 0.511
[CV] fold 2 - train-score: 0.470, test-score: 0.445
[CV] fold 3 - train-score: 0.468, test-score: 0.451
[CV] fold 4 - train-score: 0.463, test-score: 0.473
[CV] fold 5 - train-score: 0.470, test-score: 0.446
[GridSearchCV] candidate 62/64 {'learning_rate': 1.0, 'alpha': 0.13894954943731375} - score: 0.465
[CV] fold 1 - train-score: 0.464, test-score: 0.467
[CV] fold 2 - train-score: 0.468, test-score: 0.451
[CV] fold 3 - train-score: 0.468, test-score: 0.451
[CV] fold 4 - train-score: 0.456, test-score: 0.500
[CV] fold 5 - train-score: 0.467, test-score: 0.457
[GridSearchCV] candidate 63/64 {'learning_rate': 1.0, 'alpha': 0.3727593720314938} - score: 0.465
[CV] fold 1 - train-score: 0.458, test-score: 0.495
[CV] fold 2 - train-score: 0.471, test-score: 0.440
[CV] fold 3 - train-score: 0.455, test-score: 0.505
[CV] fold 4 - train-score: 0.470, test-score: 0.445
[CV] fold 5 - train-score: 0.471, test-score: 0.440
[GridSearchCV] candidate 64/64 {'learning_rate': 1.0, 'alpha': 1.0} - score: 0.465

[GridSearchCV] Best params: {'learning_rate': 0.3727593720314938, 'alpha': 0.0026826957952797246}
[GridSearchCV] Best score: 0.8278069756330627
<luma.classifier.logistic.SoftmaxRegressor at 0x1053dba60>
```

Out []:

Extract Best Model

```
In [ ]: model = grid.best_model
        grid.best_params, grid.best_score
```

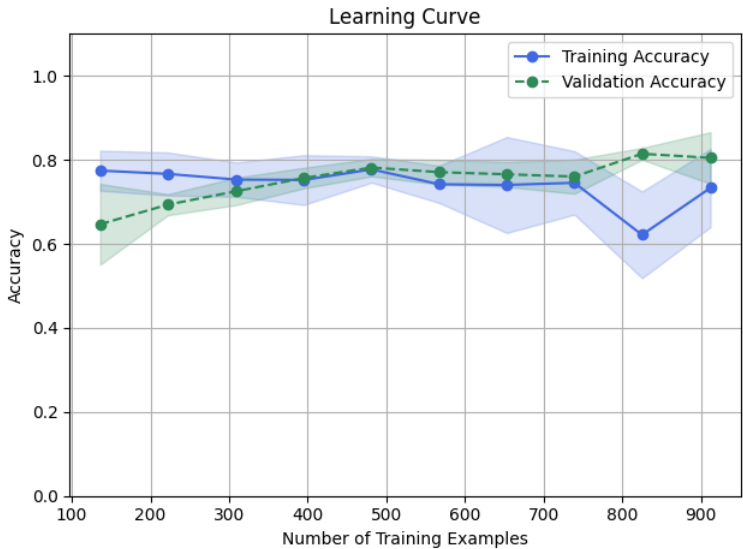
```
Out [ ]: ({'learning_rate': 0.3727593720314938, 'alpha': 0.0026826957952797246},
         0.8278069756330627)
```

Evaluate Model

1. Learning Curve

```
In [ ]: lr_curve = LearningCurve(estimator=Clone(model, False).get,
                                X=X_train_pca,
                                y=y_train,
                                train_sizes=np.linspace(0.15, 1.0, 10),
                                test_size=0.3,
                                cv=5,
                                metric=Accuracy,
                                random_state=42)

lr_curve.plot(show=True)
```



```
Out [ ]: <Axes: title={'center': 'Learning Curve'}, xlabel='Number of Training Examples', ylabel='Accuracy'>
```

2. Validation Curves for learning_rate and alpha

```
In [ ]: fig = plt.figure(figsize=(10, 5))
        ax1 = fig.add_subplot(1, 2, 1)
        ax2 = fig.add_subplot(1, 2, 2)

        val_curve_C = ValidationCurve(estimator=Clone(model, False).get,
```

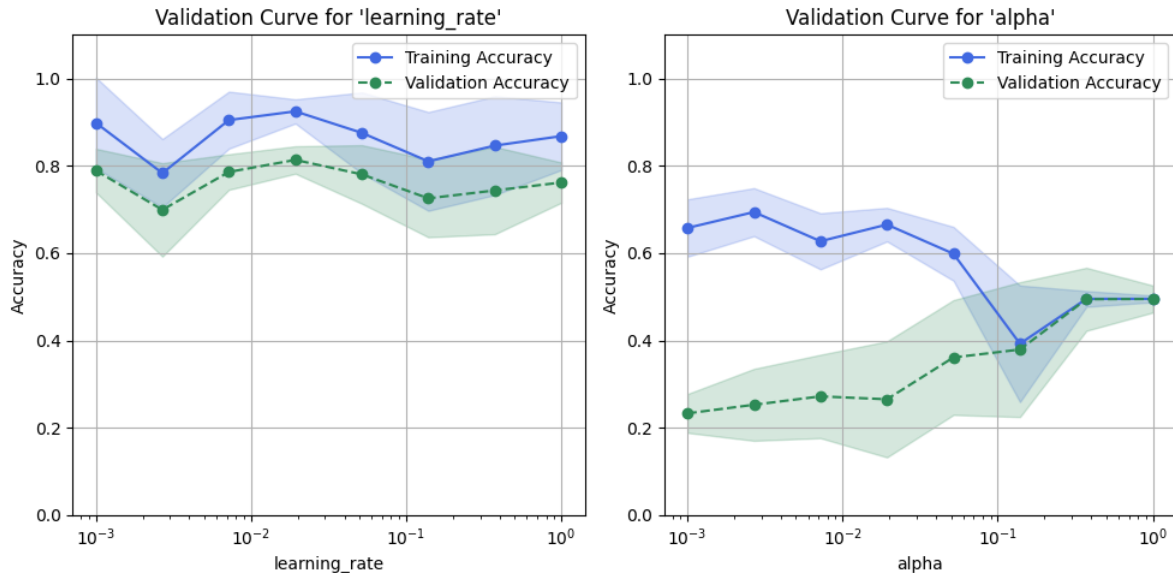
```

X=X_train_pca,
y=y_train,
param_name='learning_rate',
param_range=param_grid['learning_rate'],
cv=5,
metric=Accuracy,
random_state=42)

val_curve_gamma = ValidationCurve(estimator=Clone(model, False).get,
X=X_test_pca,
y=y_train,
param_name='alpha',
param_range=param_grid['alpha'],
cv=5,
metric=Accuracy,
random_state=42)

val_curve_C.plot(ax1, xscale='log')
val_curve_gamma.plot(ax2, xscale='log', show=True)

```



```
Out[ ]: <Axes: title={'center': "Validation Curve for 'alpha'"}, xlabel='alpha', ylabel='Accuracy'>
```

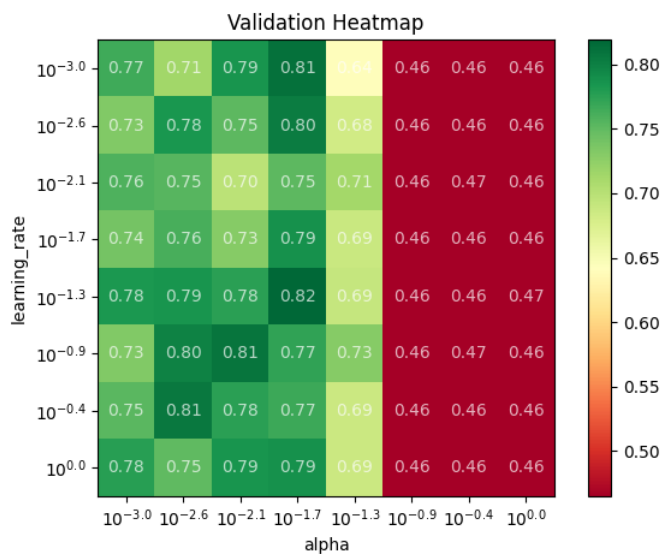
3. Validation Heatmap between learning_rate and alpha

```

In [ ]: val_hmap = ValidationHeatmap(estimator=Clone(model, False).get,
X=X_train_pca,
y=y_train,
param_dict=param_grid,
cv=5,
metric=Accuracy,
random_state=42)

val_hmap.plot(show=True)

```



```
Out[ ]: <Axes: title={'center': 'Validation Heatmap'}, xlabel='alpha', ylabel='learning_rate'>
```

4. Confusion Matrix

```

In [ ]: fig = plt.figure(figsize=(13, 5))
ax1 = fig.add_subplot(1, 2, 1)
ax2 = fig.add_subplot(1, 2, 2)

last_names = [name.rsplit(' ')[-1] for name in lfw_people.target_names]

conf_train = ConfusionMatrix(y_true=y_train,

```

```

y_pred=model.predict(X_train_pca),
labels=last_names)

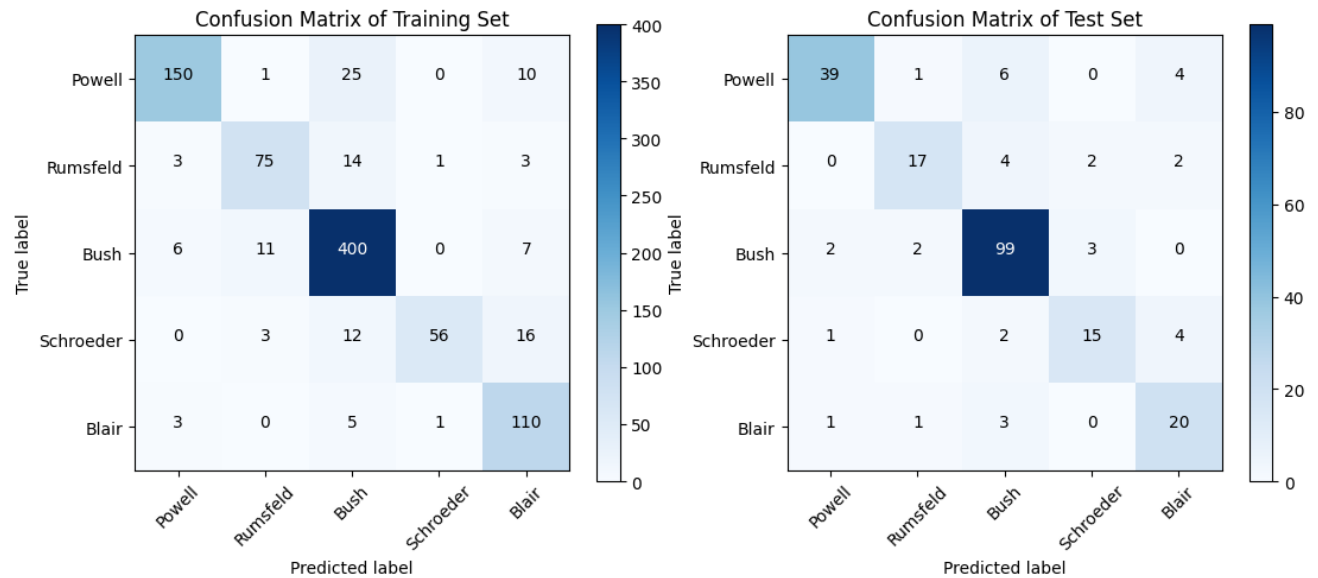
conf_test = ConfusionMatrix(y_true=y_test,
                             y_pred=model.predict(X_test_pca),
                             labels=last_names)

conf_train.plot(ax1)
conf_test.plot(ax2)

ax1.set_title('Confusion Matrix of Training Set')
ax2.set_title('Confusion Matrix of Test Set')

```

Out[]: Text(0.5, 1.0, 'Confusion Matrix of Test Set')



5. Actual Prediction Comparison

```

In [ ]: def plot_gallery(images, titles, h, w, n_row, n_col):
plt.figure(figsize=(1.5 * n_col, 2 * n_row))
plt.subplots_adjust(bottom=0, left=0.01, right=0.99, top=0.90, hspace=0.35)

for i in range(n_row * n_col):
    plt.subplot(n_row, n_col, i + 1)
    plt.imshow(images[i].reshape((h, w)), cmap=plt.cm.gray)
    plt.title(titles[i], size=12)
    plt.xticks(())
    plt.yticks(())

plt.tight_layout()
plt.show()

```

```

In [ ]: def get_title(pred, true, names, i):
pred_name = names[pred[i]].rsplit(' ', 1)[-1]
true_name = names[true[i]].rsplit(' ', 1)[-1]
return 'pred: %s\ntrue: %s' % (pred_name, true_name)
























```

```

In [ ]: test_pred = model.predict(X_test_pca)
titles = [
    get_title(test_pred, y_test, lfw_people.target_names, i)
    for i in range(y_test.shape[0])
]

plot_gallery(X_test, titles, *lfw_people.images.shape[1:], 4, 6)

```

<div>pred: Bush true: Powell</div> 	<div>pred: Powell true: Powell</div> 	<div>pred: Bush true: Bush</div> 	<div>pred: Schroeder true: Schroeder</div> 	<div>pred: Bush true: Bush</div> 	<div>pred: Bush true: Bush</div> 
<div>pred: Powell true: Powell</div> 	<div>pred: Blair true: Blair</div> 	<div>pred: Bush true: Bush</div> 	<div>pred: Bush true: Bush</div> 	<div>pred: Blair true: Blair</div> 	<div>pred: Bush true: Bush</div> 
<div>pred: Rumsfeld true: Rumsfeld</div> 	<div>pred: Powell true: Powell</div> 	<div>pred: Bush true: Bush</div> 	<div>pred: Powell true: Powell</div> 	<div>pred: Blair true: Blair</div> 	<div>pred: Blair true: Blair</div> 
<div>pred: Bush true: Bush</div> 	<div>pred: Powell true: Powell</div> 	<div>pred: Powell true: Powell</div> 	<div>pred: Bush true: Bush</div> 	<div>pred: Rumsfeld true: Rumsfeld</div> 	<div>pred: Blair true: Blair</div> 