!pip install **-**q pycaret

|████████████████████████████████| 256kB 7.1MB/s

Installing build dependencies ... done

Getting requirements to build wheel ... done

Installing backend dependencies ... done

Preparing wheel metadata ... done

|████████████████████████████████| 81kB 5.9MB/s

|████████████████████████████████| 6.8MB 13.2MB/s

|████████████████████████████████| 61kB 6.8MB/s

|████████████████████████████████| 174kB 50.3MB/s

|████████████████████████████████| 2.0MB 41.7MB/s

|████████████████████████████████| 14.2MB 273kB/s

|████████████████████████████████| 112kB 39.6MB/s

|████████████████████████████████| 61kB 7.4MB/s

|████████████████████████████████| 10.1MB 34.0MB/s

|████████████████████████████████| 1.1MB 35.7MB/s

|████████████████████████████████| 645kB 35.8MB/s

|████████████████████████████████| 112kB 52.8MB/s

|████████████████████████████████| 3.1MB 32.9MB/s

|████████████████████████████████| 81kB 8.3MB/s

|████████████████████████████████| 1.2MB 29.3MB/s

|████████████████████████████████| 174kB 41.1MB/s

|████████████████████████████████| 153kB 48.0MB/s

|████████████████████████████████| 81kB 9.4MB/s

|████████████████████████████████| 61kB 7.5MB/s

|████████████████████████████████| 1.1MB 25.0MB/s

|████████████████████████████████| 296kB 42.9MB/s

|████████████████████████████████| 71kB 8.1MB/s

|████████████████████████████████| 71kB 8.1MB/s

|████████████████████████████████| 81kB 8.9MB/s

Building wheel for pyLDAvis (PEP 517) ... done

Building wheel for umap-learn (setup.py) ... done

Building wheel for pyod (setup.py) ... done

Building wheel for phik (setup.py) ... done

Building wheel for htmlmin (setup.py) ... done

Building wheel for pynndescent (setup.py) ... done

Building wheel for databricks-cli (setup.py) ... done

Building wheel for prometheus-flask-exporter (setup.py) ... done

Building wheel for alembic (setup.py) ... done

ERROR: google-colab 1.0.0 has requirement requests~=2.23.0, but you'll have requests 2.25.1 which is incompatible.

ERROR: datascience 0.10.6 has requirement folium==0.2.1, but you'll have folium 0.8.3 which is incompatible.

ERROR: phik 0.11.2 has requirement scipy>=1.5.2, but you'll have scipy 1.4.1 which is incompatible.

ERROR: pyldavis 3.3.1 has requirement numpy>=1.20.0, but you'll have numpy 1.19.5 which is incompatible.

ERROR: pyldavis 3.3.1 has requirement pandas>=1.2.0, but you'll have pandas 1.1.5 which is incompatible.

In [1]:

**import** pandas **as** pd

In [2]:

data **=** pd**.**read\_csv('/content/forestfires.csv')

In [4]:

data**.**head()

Out[4]:

|  | **X** | **Y** | **month** | **day** | **FFMC** | **DMC** | **DC** | **ISI** | **temp** | **RH** | **wind** | **rain** | **area** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 7 | 5 | mar | fri | 86.2 | 26.2 | 94.3 | 5.1 | 8.2 | 51 | 6.7 | 0.0 | 0.0 |
| **1** | 7 | 4 | oct | tue | 90.6 | 35.4 | 669.1 | 6.7 | 18.0 | 33 | 0.9 | 0.0 | 0.0 |
| **2** | 7 | 4 | oct | sat | 90.6 | 43.7 | 686.9 | 6.7 | 14.6 | 33 | 1.3 | 0.0 | 0.0 |
| **3** | 8 | 6 | mar | fri | 91.7 | 33.3 | 77.5 | 9.0 | 8.3 | 97 | 4.0 | 0.2 | 0.0 |
| **4** | 8 | 6 | mar | sun | 89.3 | 51.3 | 102.2 | 9.6 | 11.4 | 99 | 1.8 | 0.0 | 0.0 |

In [6]:

data['area']**.**count

Out[6]:

In [21]:

data**=**data**.**drop(['X','Y','month','day'],axis**=**1)

In [22]:

data**.**head()

Out[22]:

|  | **FFMC** | **DMC** | **DC** | **ISI** | **temp** | **RH** | **wind** | **rain** | **area** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 86.2 | 26.2 | 94.3 | 5.1 | 8.2 | 51 | 6.7 | 0.0 | 0.0 |
| **1** | 90.6 | 35.4 | 669.1 | 6.7 | 18.0 | 33 | 0.9 | 0.0 | 0.0 |
| **2** | 90.6 | 43.7 | 686.9 | 6.7 | 14.6 | 33 | 1.3 | 0.0 | 0.0 |
| **3** | 91.7 | 33.3 | 77.5 | 9.0 | 8.3 | 97 | 4.0 | 0.2 | 0.0 |
| **4** | 89.3 | 51.3 | 102.2 | 9.6 | 11.4 | 99 | 1.8 | 0.0 | 0.0 |

**Regression**

In [23]:

**from** pycaret.regression **import** **\***

reg1 **=** setup(data **=** data, target **=** 'area')

|  | **Description** | **Value** |
| --- | --- | --- |
| **0** | session\_id | 767 |
| **1** | Target | area |
| **2** | Original Data | (517, 9) |
| **3** | Missing Values | False |
| **4** | Numeric Features | 8 |
| **5** | Categorical Features | 0 |
| **6** | Ordinal Features | False |
| **7** | High Cardinality Features | False |
| **8** | High Cardinality Method | None |
| **9** | Transformed Train Set | (361, 8) |
| **10** | Transformed Test Set | (156, 8) |
| **11** | Shuffle Train-Test | True |
| **12** | Stratify Train-Test | False |
| **13** | Fold Generator | KFold |
| **14** | Fold Number | 10 |
| **15** | CPU Jobs | -1 |
| **16** | Use GPU | False |
| **17** | Log Experiment | False |
| **18** | Experiment Name | reg-default-name |
| **19** | USI | aa7e |
| **20** | Imputation Type | simple |
| **21** | Iterative Imputation Iteration | None |
| **22** | Numeric Imputer | mean |
| **23** | Iterative Imputation Numeric Model | None |
| **24** | Categorical Imputer | constant |
| **25** | Iterative Imputation Categorical Model | None |
| **26** | Unknown Categoricals Handling | least\_frequent |
| **27** | Normalize | False |
| **28** | Normalize Method | None |
| **29** | Transformation | False |
| **30** | Transformation Method | None |
| **31** | PCA | False |
| **32** | PCA Method | None |
| **33** | PCA Components | None |
| **34** | Ignore Low Variance | False |
| **35** | Combine Rare Levels | False |
| **36** | Rare Level Threshold | None |
| **37** | Numeric Binning | False |
| **38** | Remove Outliers | False |
| **39** | Outliers Threshold | None |
| **40** | Remove Multicollinearity | False |
| **41** | Multicollinearity Threshold | None |
| **42** | Clustering | False |
| **43** | Clustering Iteration | None |
| **44** | Polynomial Features | False |
| **45** | Polynomial Degree | None |
| **46** | Trignometry Features | False |
| **47** | Polynomial Threshold | None |
| **48** | Group Features | False |
| **49** | Feature Selection | False |
| **50** | Feature Selection Method | classic |
| **51** | Features Selection Threshold | None |
| **52** | Feature Interaction | False |
| **53** | Feature Ratio | False |
| **54** | Interaction Threshold | None |
| **55** | Transform Target | False |
| **56** | Transform Target Method | box-cox |

In [24]:

best\_model**=**compare\_models(n\_select**=**3)

|  | **Model** | **MAE** | **MSE** | **RMSE** | **R2** | **RMSLE** | **MAPE** | **TT (Sec)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **huber** | Huber Regressor | 14.9330 | 5775.5513 | 51.5948 | -0.0967 | 1.4137 | 1.0282 | 0.043 |
| **br** | Bayesian Ridge | 22.1262 | 5640.2299 | 52.3419 | -3.9629 | 2.1710 | 6.6082 | 0.018 |
| **llar** | Lasso Least Angle Regression | 22.1024 | 5641.0953 | 52.3341 | -3.9956 | 2.1711 | 6.5625 | 0.013 |
| **omp** | Orthogonal Matching Pursuit | 22.9264 | 5606.0882 | 53.0576 | -4.6008 | 2.1387 | 8.2239 | 0.013 |
| **par** | Passive Aggressive Regressor | 33.2102 | 6687.2869 | 62.5216 | -5.4028 | 2.2684 | 10.6615 | 0.017 |
| **en** | Elastic Net | 23.5175 | 5660.0218 | 53.9036 | -5.5115 | 2.1633 | 7.6922 | 0.014 |
| **lasso** | Lasso Regression | 23.5442 | 5662.3945 | 53.9399 | -5.6200 | 2.1648 | 7.7291 | 0.013 |
| **ridge** | Ridge Regression | 23.9268 | 5681.4428 | 54.3661 | -6.0723 | 2.1776 | 7.9900 | 0.012 |
| **lar** | Least Angle Regression | 24.0083 | 5696.1420 | 54.5872 | -6.0939 | 2.1794 | 8.0042 | 0.016 |
| **lr** | Linear Regression | 24.0083 | 5696.1419 | 54.5872 | -6.0939 | 2.1794 | 8.0042 | 0.308 |
| **gbr** | Gradient Boosting Regressor | 21.4574 | 6048.6374 | 58.6851 | -13.5416 | 1.8782 | 6.3936 | 0.075 |
| **rf** | Random Forest Regressor | 24.7736 | 6426.5958 | 64.2608 | -13.7077 | 2.0469 | 6.7849 | 0.474 |
| **knn** | K Neighbors Regressor | 26.4318 | 6896.6046 | 68.3579 | -15.0193 | 2.0704 | 6.0150 | 0.066 |
| **lightgbm** | Light Gradient Boosting Machine | 26.9711 | 6089.6048 | 60.8658 | -17.8824 | 2.1933 | 14.3354 | 0.087 |
| **et** | Extra Trees Regressor | 26.1449 | 6360.8497 | 63.6861 | -25.6073 | 2.1237 | 7.7784 | 0.414 |
| **ada** | AdaBoost Regressor | 31.1298 | 5944.5425 | 61.8251 | -34.5055 | 2.3929 | 10.9239 | 0.080 |
| **dt** | Decision Tree Regressor | 26.5822 | 8645.1321 | 77.4821 | -55.7468 | 2.0094 | 6.3430 | 0.022 |

In [25]:

print(best\_model)

[HuberRegressor(alpha=0.0001, epsilon=1.35, fit\_intercept=True, max\_iter=100,

tol=1e-05, warm\_start=False), BayesianRidge(alpha\_1=1e-06, alpha\_2=1e-06, alpha\_init=None,

compute\_score=False, copy\_X=True, fit\_intercept=True,

lambda\_1=1e-06, lambda\_2=1e-06, lambda\_init=None, n\_iter=300,

normalize=False, tol=0.001, verbose=False), LassoLars(alpha=1.0, copy\_X=True, eps=2.220446049250313e-16, fit\_intercept=True,

fit\_path=True, jitter=None, max\_iter=500, normalize=True,

positive=False, precompute='auto', random\_state=767, verbose=False)]

In [26]:

huber\_model**=**create\_model('huber', fold **=** 20)

|  | **MAE** | **MSE** | **RMSE** | **R2** | **RMSLE** | **MAPE** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | 14.2551 | 2126.9363 | 46.1187 | -0.0792 | 1.4566 | 0.6568 |
| **1** | 7.5434 | 163.9068 | 12.8026 | -0.2469 | 1.3378 | 0.9563 |
| **2** | 9.6375 | 659.9386 | 25.6893 | -0.1059 | 1.3749 | 0.5885 |
| **3** | 4.7094 | 68.7168 | 8.2896 | -0.1016 | 1.1770 | 0.9225 |
| **4** | 13.0405 | 1061.1431 | 32.5752 | -0.1197 | 1.4974 | 0.6530 |
| **5** | 5.3678 | 115.1296 | 10.7298 | -0.1271 | 1.2248 | 0.7314 |
| **6** | 76.3897 | 68412.9342 | 261.5587 | -0.0881 | 2.0335 | 0.6082 |
| **7** | 8.0693 | 196.5693 | 14.0203 | -0.2549 | 1.3494 | 1.4799 |
| **8** | 3.0700 | 37.5692 | 6.1294 | -0.0557 | 0.9315 | 1.3182 |
| **9** | 6.8765 | 226.5815 | 15.0526 | -0.0897 | 1.3252 | 0.7812 |
| **10** | 1.8018 | 3.8855 | 1.9712 | -0.2331 | 0.8988 | 0.7167 |
| **11** | 2.1758 | 7.3173 | 2.7050 | 0.0733 | 0.8909 | 3.2580 |
| **12** | 17.8831 | 4267.2166 | 65.3239 | -0.0593 | 1.5239 | 0.8832 |
| **13** | 28.9122 | 4287.5634 | 65.4795 | -0.1984 | 1.9755 | 0.8494 |
| **14** | 11.0069 | 392.9248 | 19.8223 | -0.3293 | 1.5372 | 1.3310 |
| **15** | 6.8951 | 152.1759 | 12.3360 | -0.2216 | 1.3403 | 0.7714 |
| **16** | 49.4399 | 31122.0250 | 176.4144 | -0.0771 | 1.8249 | 0.7355 |
| **17** | 2.7138 | 12.1047 | 3.4792 | -0.0655 | 1.0134 | 0.7675 |
| **18** | 8.4447 | 268.3440 | 16.3812 | -0.2193 | 1.3735 | 0.7455 |
| **19** | 20.2560 | 1849.5179 | 43.0060 | -0.2327 | 1.7957 | 0.7509 |
| **Mean** | 14.9244 | 5771.6250 | 41.9942 | -0.1416 | 1.3941 | 0.9753 |
| **SD** | 17.8075 | 15857.4294 | 63.3096 | 0.0935 | 0.3234 | 0.5760 |

In [28]:

plot\_model(huber\_model)

In [29]:

plot\_model(huber\_model,'vc')

In [30]:

plot\_model(huber\_model,'error')

**Classification**

In [43]:

**def** preprocessing(df):

df**=**df**.**copy()

df['area']**=**df['area']**.**apply(**lambda** x: 1 **if** x**>**0 **else** 0)

**return** df

In [44]:

df**=**preprocessing(data)

In [45]:

df

Out[45]:

|  | **FFMC** | **DMC** | **DC** | **ISI** | **temp** | **RH** | **wind** | **rain** | **area** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 86.2 | 26.2 | 94.3 | 5.1 | 8.2 | 51 | 6.7 | 0.0 | 0 |
| **1** | 90.6 | 35.4 | 669.1 | 6.7 | 18.0 | 33 | 0.9 | 0.0 | 0 |
| **2** | 90.6 | 43.7 | 686.9 | 6.7 | 14.6 | 33 | 1.3 | 0.0 | 0 |
| **3** | 91.7 | 33.3 | 77.5 | 9.0 | 8.3 | 97 | 4.0 | 0.2 | 0 |
| **4** | 89.3 | 51.3 | 102.2 | 9.6 | 11.4 | 99 | 1.8 | 0.0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **512** | 81.6 | 56.7 | 665.6 | 1.9 | 27.8 | 32 | 2.7 | 0.0 | 1 |
| **513** | 81.6 | 56.7 | 665.6 | 1.9 | 21.9 | 71 | 5.8 | 0.0 | 1 |
| **514** | 81.6 | 56.7 | 665.6 | 1.9 | 21.2 | 70 | 6.7 | 0.0 | 1 |
| **515** | 94.4 | 146.0 | 614.7 | 11.3 | 25.6 | 42 | 4.0 | 0.0 | 0 |
| **516** | 79.5 | 3.0 | 106.7 | 1.1 | 11.8 | 31 | 4.5 | 0.0 | 0 |

517 rows × 9 columns

In [50]:

**from** pycaret.classification **import** **\***

reg2 **=** setup(data **=** df, target **=** 'area')

|  | **Description** | **Value** |
| --- | --- | --- |
| **0** | session\_id | 4804 |
| **1** | Target | area |
| **2** | Target Type | Binary |
| **3** | Label Encoded | 0: 0, 1: 1 |
| **4** | Original Data | (517, 9) |
| **5** | Missing Values | False |
| **6** | Numeric Features | 8 |
| **7** | Categorical Features | 0 |
| **8** | Ordinal Features | False |
| **9** | High Cardinality Features | False |
| **10** | High Cardinality Method | None |
| **11** | Transformed Train Set | (361, 8) |
| **12** | Transformed Test Set | (156, 8) |
| **13** | Shuffle Train-Test | True |
| **14** | Stratify Train-Test | False |
| **15** | Fold Generator | StratifiedKFold |
| **16** | Fold Number | 10 |
| **17** | CPU Jobs | -1 |
| **18** | Use GPU | False |
| **19** | Log Experiment | False |
| **20** | Experiment Name | clf-default-name |
| **21** | USI | f09f |
| **22** | Imputation Type | simple |
| **23** | Iterative Imputation Iteration | None |
| **24** | Numeric Imputer | mean |
| **25** | Iterative Imputation Numeric Model | None |
| **26** | Categorical Imputer | constant |
| **27** | Iterative Imputation Categorical Model | None |
| **28** | Unknown Categoricals Handling | least\_frequent |
| **29** | Normalize | False |
| **30** | Normalize Method | None |
| **31** | Transformation | False |
| **32** | Transformation Method | None |
| **33** | PCA | False |
| **34** | PCA Method | None |
| **35** | PCA Components | None |
| **36** | Ignore Low Variance | False |
| **37** | Combine Rare Levels | False |
| **38** | Rare Level Threshold | None |
| **39** | Numeric Binning | False |
| **40** | Remove Outliers | False |
| **41** | Outliers Threshold | None |
| **42** | Remove Multicollinearity | False |
| **43** | Multicollinearity Threshold | None |
| **44** | Clustering | False |
| **45** | Clustering Iteration | None |
| **46** | Polynomial Features | False |
| **47** | Polynomial Degree | None |
| **48** | Trignometry Features | False |
| **49** | Polynomial Threshold | None |
| **50** | Group Features | False |
| **51** | Feature Selection | False |
| **52** | Feature Selection Method | classic |
| **53** | Features Selection Threshold | None |
| **54** | Feature Interaction | False |
| **55** | Feature Ratio | False |
| **56** | Interaction Threshold | None |
| **57** | Fix Imbalance | False |
| **58** | Fix Imbalance Method | SMOTE |

In [52]:

best\_model**=**compare\_models()

|  | **Model** | **Accuracy** | **AUC** | **Recall** | **Prec.** | **F1** | **Kappa** | **MCC** | **TT (Sec)** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ada** | Ada Boost Classifier | 0.5763 | 0.5924 | 0.5703 | 0.5718 | 0.5689 | 0.1532 | 0.1539 | 0.111 |
| **knn** | K Neighbors Classifier | 0.5679 | 0.5727 | 0.5307 | 0.5713 | 0.5480 | 0.1357 | 0.1370 | 0.116 |
| **et** | Extra Trees Classifier | 0.5596 | 0.5737 | 0.5464 | 0.5572 | 0.5490 | 0.1186 | 0.1198 | 0.468 |
| **rf** | Random Forest Classifier | 0.5456 | 0.5646 | 0.5467 | 0.5392 | 0.5381 | 0.0908 | 0.0921 | 0.514 |
| **dt** | Decision Tree Classifier | 0.5349 | 0.5340 | 0.5363 | 0.5247 | 0.5254 | 0.0702 | 0.0716 | 0.020 |
| **lr** | Logistic Regression | 0.5295 | 0.5324 | 0.5810 | 0.5269 | 0.5483 | 0.0600 | 0.0596 | 0.060 |
| **gbc** | Gradient Boosting Classifier | 0.5264 | 0.5509 | 0.5029 | 0.5221 | 0.5100 | 0.0523 | 0.0532 | 0.098 |
| **lightgbm** | Light Gradient Boosting Machine | 0.5236 | 0.5377 | 0.5252 | 0.5187 | 0.5185 | 0.0477 | 0.0484 | 0.036 |
| **nb** | Naive Bayes | 0.5152 | 0.5871 | 0.9552 | 0.5057 | 0.6610 | 0.0369 | 0.0924 | 0.019 |
| **ridge** | Ridge Classifier | 0.5074 | 0.0000 | 0.5699 | 0.5008 | 0.5302 | 0.0156 | 0.0141 | 0.013 |
| **lda** | Linear Discriminant Analysis | 0.5074 | 0.5449 | 0.5810 | 0.5032 | 0.5369 | 0.0159 | 0.0143 | 0.015 |
| **qda** | Quadratic Discriminant Analysis | 0.4959 | 0.0000 | 1.0000 | 0.4959 | 0.6629 | 0.0000 | 0.0000 | 0.021 |
| **svm** | SVM - Linear Kernel | 0.4794 | 0.0000 | 0.6147 | 0.3390 | 0.4349 | -0.0384 | -0.0899 | 0.016 |

In [53]:

print(best\_model)

AdaBoostClassifier(algorithm='SAMME.R', base\_estimator=None, learning\_rate=1.0,

n\_estimators=50, random\_state=4804)

In [68]:

adaboost\_model**=**create\_model('ada', fold **=** 20)

|  | **Accuracy** | **AUC** | **Recall** | **Prec.** | **F1** | **Kappa** | **MCC** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 0.5263 | 0.4556 | 0.4444 | 0.5000 | 0.4706 | 0.0447 | 0.0449 |
| **1** | 0.5556 | 0.5432 | 0.4444 | 0.5714 | 0.5000 | 0.1111 | 0.1140 |
| **2** | 0.5000 | 0.5432 | 0.4444 | 0.5000 | 0.4706 | 0.0000 | 0.0000 |
| **3** | 0.6111 | 0.7160 | 0.6667 | 0.6000 | 0.6316 | 0.2222 | 0.2236 |
| **4** | 0.6111 | 0.6914 | 0.6667 | 0.6000 | 0.6316 | 0.2222 | 0.2236 |
| **5** | 0.5556 | 0.7407 | 0.7778 | 0.5385 | 0.6364 | 0.1111 | 0.1240 |
| **6** | 0.5556 | 0.4815 | 0.5556 | 0.5556 | 0.5556 | 0.1111 | 0.1111 |
| **7** | 0.5000 | 0.4506 | 0.5556 | 0.5000 | 0.5263 | 0.0000 | 0.0000 |
| **8** | 0.3333 | 0.4136 | 0.2222 | 0.2857 | 0.2500 | -0.3333 | -0.3419 |
| **9** | 0.5000 | 0.5802 | 0.5556 | 0.5000 | 0.5263 | 0.0000 | 0.0000 |
| **10** | 0.4444 | 0.5000 | 0.2222 | 0.4000 | 0.2857 | -0.1111 | -0.1240 |
| **11** | 0.6111 | 0.5309 | 0.6667 | 0.6000 | 0.6316 | 0.2222 | 0.2236 |
| **12** | 0.6667 | 0.6728 | 0.8889 | 0.6154 | 0.7273 | 0.3333 | 0.3721 |
| **13** | 0.5000 | 0.5741 | 0.3333 | 0.5000 | 0.4000 | 0.0000 | 0.0000 |
| **14** | 0.5000 | 0.5309 | 0.6667 | 0.5000 | 0.5714 | 0.0000 | 0.0000 |
| **15** | 0.5556 | 0.5370 | 0.4444 | 0.5714 | 0.5000 | 0.1111 | 0.1140 |
| **16** | 0.5556 | 0.5679 | 0.4444 | 0.5714 | 0.5000 | 0.1111 | 0.1140 |
| **17** | 0.6111 | 0.6543 | 0.7778 | 0.5833 | 0.6667 | 0.2222 | 0.2357 |
| **18** | 0.4444 | 0.5864 | 0.4444 | 0.4444 | 0.4444 | -0.1111 | -0.1111 |
| **19** | 0.4444 | 0.4750 | 0.3750 | 0.3750 | 0.3750 | -0.1250 | -0.1250 |
| **Mean** | 0.5291 | 0.5623 | 0.5299 | 0.5156 | 0.5150 | 0.0571 | 0.0599 |
| **SD** | 0.0753 | 0.0898 | 0.1767 | 0.0836 | 0.1212 | 0.1514 | 0.1583 |

In [69]:

plot\_model(adaboost\_model)

In [70]:

plot\_model(adaboost\_model,'learning')

In [71]:

tuned\_adaboost**=**tune\_model(adaboost\_model,optimize**=**'Accuracy')

|  | **Accuracy** | **AUC** | **Recall** | **Prec.** | **F1** | **Kappa** | **MCC** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 0.4865 | 0.4678 | 0.6111 | 0.4783 | 0.5366 | -0.0203 | -0.0211 |
| **1** | 0.5556 | 0.6173 | 0.5556 | 0.5556 | 0.5556 | 0.1111 | 0.1111 |
| **2** | 0.6111 | 0.6883 | 0.5000 | 0.6429 | 0.5625 | 0.2222 | 0.2279 |
| **3** | 0.5556 | 0.5185 | 0.5000 | 0.5625 | 0.5294 | 0.1111 | 0.1118 |
| **4** | 0.4444 | 0.5154 | 0.4444 | 0.4444 | 0.4444 | -0.1111 | -0.1111 |
| **5** | 0.4722 | 0.4660 | 0.1667 | 0.4286 | 0.2400 | -0.0556 | -0.0702 |
| **6** | 0.6389 | 0.7037 | 0.6111 | 0.6471 | 0.6286 | 0.2778 | 0.2782 |
| **7** | 0.6667 | 0.6497 | 0.6667 | 0.6667 | 0.6667 | 0.3333 | 0.3333 |
| **8** | 0.6667 | 0.6698 | 0.6667 | 0.6667 | 0.6667 | 0.3333 | 0.3333 |
| **9** | 0.5556 | 0.6300 | 0.7647 | 0.5200 | 0.6190 | 0.1299 | 0.1443 |
| **Mean** | 0.5653 | 0.5927 | 0.5487 | 0.5613 | 0.5449 | 0.1332 | 0.1338 |
| **SD** | 0.0760 | 0.0870 | 0.1562 | 0.0872 | 0.1208 | 0.1513 | 0.1536 |

In [73]:

plot\_model(tuned\_adaboost,'learning')

In [74]:

save\_model(tuned\_adaboost,'forestfiremodel')

Transformation Pipeline and Model Succesfully Saved

Out[74]:

(Pipeline(memory=None,

steps=[('dtypes',

DataTypes\_Auto\_infer(categorical\_features=[],

display\_types=True, features\_todrop=[],

id\_columns=[],

ml\_usecase='classification',

numerical\_features=[], target='area',

time\_features=[])),

('imputer',

Simple\_Imputer(categorical\_strategy='not\_available',

fill\_value\_categorical=None,

fill\_value\_numerical=None,

numeric\_strateg...

('dummy', Dummify(target='area')),

('fix\_perfect', Remove\_100(target='area')),

('clean\_names', Clean\_Colum\_Names()),

('feature\_select', 'passthrough'), ('fix\_multi', 'passthrough'),

('dfs', 'passthrough'), ('pca', 'passthrough'),

['trained\_model',

AdaBoostClassifier(algorithm='SAMME.R', base\_estimator=None,

learning\_rate=0.2, n\_estimators=30,

random\_state=4804)]],

verbose=False), 'forestfiremodel.pkl')