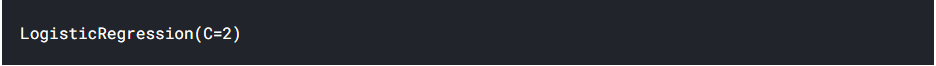
#Logistic Regression Algorithm

from sklearn.linear\_model import LogisticRegression

lor=LogisticRegression(C=2, max\_iter=100)

lor= lor.fit(X\_train , y\_train)

lor



#Accuracy

y\_pred1 = lor.predict(X\_test)

lr=lor.score(X\_test, y\_test)

print('Accuracy score= {:.2f}'.format(lor.score(X\_test, y\_test)))



#Confusion Matrix

from sklearn.metrics import classification\_report, confusion\_matrix

from mlxtend.plotting import plot\_confusion\_matrix

print('\n')

print("confusion matrix")

print('\n')

CR=confusion\_matrix(y\_test, y\_pred1)

print(CR)

print('\n')

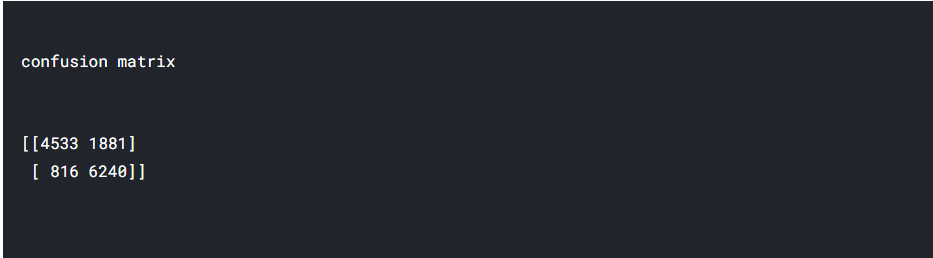
fig, ax = plot\_confusion\_matrix(conf\_mat=CR,figsize=(10, 10),

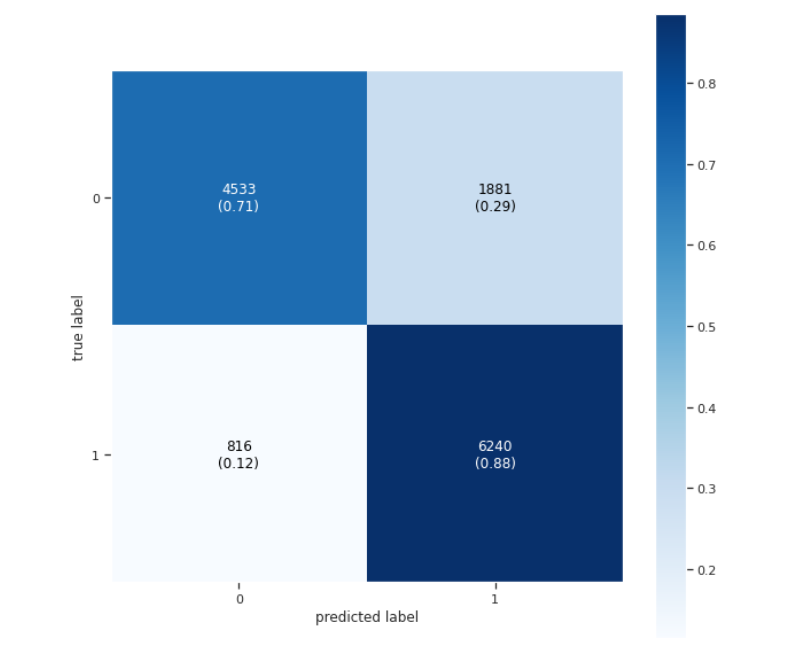
show\_absolute=True,

show\_normed=True,

colorbar=True)

plt.show()





Logistic Regression Logistic Regression is a parametric classifier which is easy to train and has low variance. The feature sampling probability is adapted according to the predictive performance and the weights of the logistic regression. The overall objective is to optimize the predictive performance of a classifier while favoring also sparse and stable models.