```python
import numpy as np
import srcmatrix as src
import infomatrix as info

#from .Calculate_W import calculate_W_by_svd
from .Calculate_W import *

class mCLESS:
    def __init__(self):
        self.W = np.empty((1,1), dtype='double')

    def fit(self, X, y):
        A = info.Information_matrix(X)
        B = src.Source_matrix(y)
        #self.W = calculate_W_by_svd(A,B)
        self.W = calculate_W_by_normal(A,B)
        return self

    def predict(self,X):
        A = info.Information_matrix(X)
        B_pred = A @ self.W
        N = len(B_pred)
        y_pred = np.zeros(N)
        for i in range(N):
            c = np.argmax(B_pred[i])
            y_pred[i] = c
        return y_pred

    def score(self, X, y):
        y_pred = self.predict(X)
        confusion_matrix = np.zeros((len(np.unique(y)),len(np.unique(y))))
        for i in zip(y,y_pred):
            confusion_matrix[int(i[0]),int(i[1])]+=1
        accuracy = np.trace(confusion_matrix)/np.sum(confusion_matrix)
        return accuracy

```