**SKLEARN Tests Summary**

**Test 1:**

clf\_image = MLPClassifier(activation = 'relu',

hidden\_layer\_sizes = (100, 100, 100), # 3 layers of 100 neurons each

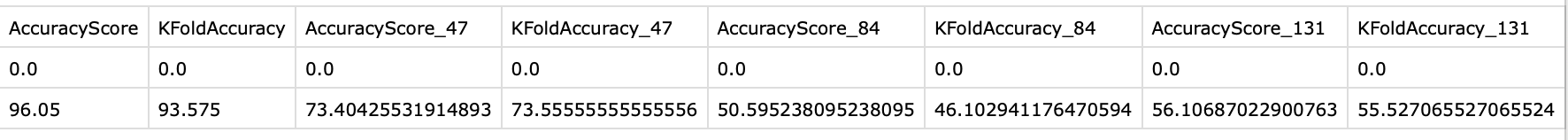
solver = 'adam',

verbose = True,

random\_state = 1,

max\_iter = 100,

early\_stopping=True) # batchsize = 200 default





**Test 2:**

clf\_image = MLPClassifier(activation = 'relu',

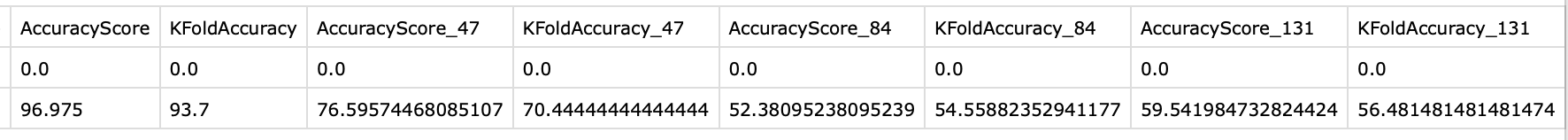
hidden\_layer\_sizes = (100, 100, 100), # 3 layers of 100 neurons each

solver = 'adam',

verbose = True,

max\_iter = 100,

early\_stopping=True) # batchsize = 200 default

****

****

**Test 3:**

clf\_image = MLPClassifier(activation = 'relu',

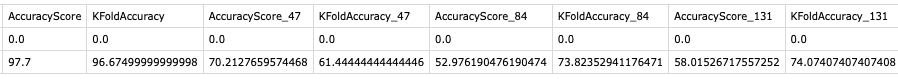
hidden\_layer\_sizes = (100, 100, 100), # 3 layers of 100 neurons each

solver = 'sgd',

verbose = True,

max\_iter = 100,

early\_stopping=True)

****

****

**Test 4:**

clf\_image = MLPClassifier(activation = 'relu',

hidden\_layer\_sizes = (100, 100, 100),

solver = 'sgd',

verbose = True,

max\_iter = 100,

early\_stopping=True) # batchsize = 200 default

test\_percent = 0.2

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, Y, shuffle=True, test\_size =test\_percent)

# Cross Validation

n\_splits = 10

random\_state = 100

kfold = model\_selection.KFold(n\_splits = n\_splits, random\_state = random\_state)

results = model\_selection.cross\_val\_score(clf\_image, x\_test, y\_test, cv = kfold)

KFoldAccuracy = (results.mean())\*100

KFoldAccuracy\_std = results.std()

****

**Test 5:**