MNIST with SVM classifier

Implement \*\*multiclass MNIST digit recognition classifier\*\* with \*\*SVMs\*\*.

* Import basic libraries
* Import the libraries for performing classification
* Load MNIST dataset
* What are the shapes of training data testing data and corresponding labels?
* Plot some data sample images
* Flatten each input image into a vector of length 784 and normalize
* # Flatten each input image into a vector of length 784
* X\_train = X\_train.reshape(X\_train.shape[0], 28\*28)
* X\_test = X\_test.reshape(X\_test.shape[0], 28\*28)
* # Normalizing.
* X\_train = X\_train/255
* X\_test = X\_test/255
* What are the shapes of training and testing data after flattening?
* Let us consider the first 10,000 images in training dataset and first 2,000 images in testing dataset.
* Now train Linear SVM for MNIST multiclass classification

pipe\_1 = Pipeline([('scaler', MinMaxScaler()),

                         ("classifier", SVC(kernel = 'linear', C = 1))])

pipe\_1.fit(X\_train,y\_train.ravel())

# Evaluate the model using crossvalidation

acc = cross\_val\_score(pipe\_1, X\_train, y\_train.ravel(),cv=2)

print("Training Accuracy: {:.2f} %".format(acc.mean()\*100))

* What is the accuracy?
* visualize the confusion matrix using heatmap and classification report
* Now use the Nonlinear SVM for MNIST multiclass classification using pipeline
* pipe\_2 = Pipeline([('scaler', MinMaxScaler()),
* ("classifier", SVC(kernel = 'rbf', gamma = 0.1, C = 1))])
* pipe\_2.fit(X\_train,y\_train.ravel())
* # Evaluate the model using crossvalidation
* acc = cross\_val\_score(pipe\_2, X\_train, y\_train.ravel(),cv=2)
* print("Training Accuracy: {:.2f} %".format(acc.mean()\*100))
* What is the accuracy? Confusion matrix and classification report?
* Use a grid search cross-validation to explore combinations of parameters. Here we will adjust ``C`` (which controls the margin hardness) and ``gamma`` (which controls the size of the radial basis function kernel), and determine the best model. What are the best values of c and gamma?