```
DATA = load("data.mat");

Data = DATA.('imageTrain');
labelData = DATA.('labelTrain');

testData = DATA.('imageTestNew');
labelTest = DATA.('labelTestNew');

FEATURE_SIZE = 784;
% Question 1:
sampleTrain = imread('sampletrain.png');
sampleTest = imread('sampletest.png');

reshapedSampleTrain = reshape(sampleTrain, [1, FEATURE_SIZE]);
reshapedSampleTest = reshape(sampleTest, [1, FEATURE_SIZE]);
cov_xy = (double(reshapedSampleTrain) * double(reshapedSampleTest).');
var_x = double(reshapedSampleTrain) * double(reshapedSampleTrain).';
a = cov_xy / var_x;
y = sampleTest / a;
% a = 0.6796
imshow(y);

% Question 2
% Stack the column vectors
x_train_stacked = reshape(Data, FEATURE_SIZE, length(labelData));
x_test_stacked = reshape(testData, FEATURE_SIZE, length(labelTest));
% Normalize the data
Classifier2 = NNClassifier();
Classifier2 = create(Classifier2, x_train_stacked, labelData);
y_pred2 = predictNorm(Classifier2, x_test_stacked, labelTest);
error2 = getError(Classifier2, y_pred2, labelTest);


% Question 2 error rate
error2(:,3) = error2(:,3) * 100;
plot([0:9], error2(:, 3));
xlim([0, 9]);
ytickformat('percentage');
title("Error Rates for different Labels");
xlabel("Labels");
ylabel("Percentages");

% Error for Class 0: [6.66666666666667]%
% Error for Class 1: 11.8421052631579%
% Error for Class 2: 6.66666666666667%
% Error for Class 3: 9.52380952380952%
% Error for Class 4: 18.3333333333333%
% Error for Class 5: 10%
% Error for Class 6: 6.97674418604651%
% Error for Class 7: 7.69230769230769%
% Error for Class 8: 11.7647058823529%
% Error for Class 9: 15.0943396226415%



% Question 2 Total Error
totalError2 = sum(error2(:,1)) / sum(error2(:,2)); % Total count / total error
% Total Error = 10.80%

% Initialize class
Classifier3 = NNClassifier();
Classifier3 = create(Classifier3, x_train_stacked, labelData); % instantiate members
y_pred3 = predict(Classifier3, x_test_stacked, labelTest); % predict test data
error3 = getError(Classifier3, y_pred3, labelTest); % Get error for each label, total↙
count of numbers, and error rate for each


% Question 3 error rate
error3(:,3) = error3(:,3) * 100;
plot([0:9], error3(:, 3));
xlim([0, 9]);
ytickformat('percentage');
title("Error Rates for different Labels");
xlabel("Labels");
ylabel("Percentages");

% Error for Class 0: 11.6279069767442%
% Error for Class 1: 49.6240601503759%
% Error for Class 2: 2.63157894736842%
% Error for Class 3: 11.7647058823529%
% Error for Class 4: 13.4615384615385%
% Error for Class 5: 13.9534883720930%
% Error for Class 6: 5.88235294117647%
% Error for Class 7: 8.16326530612245%
% Error for Class 8: 17.8571428571429%
% Error for Class 9: 13.0434782608696%



% Question 3 Total Error
totalError3 = sum(error3(:,1)) / sum(error3(:,2)); % Total count / total error
% Total Error = 21.20


%% After using ML and finding the best fitting parameters using cov(x,y) / var(x) to↙
scale it down, we can lower the total error from  21.2% to 10.8%
%% The variance measures how evenly distributed it is within the mean and the covarance↙
informs us of the relation between the training and test.
%% This will scale down the data relative to x (our training to best approximate).
```