**Texas Tech University**

**Department of Computer Science**

**Course:** Introduction to Artificial Intelligence **Group:** 1

**Instructor:** Dr. Juan Carlos Rojas **Email:** [Juan-Carlos.Rojas@ttu.edu](mailto:Juan-Carlos.Rojas@ttu.edu)

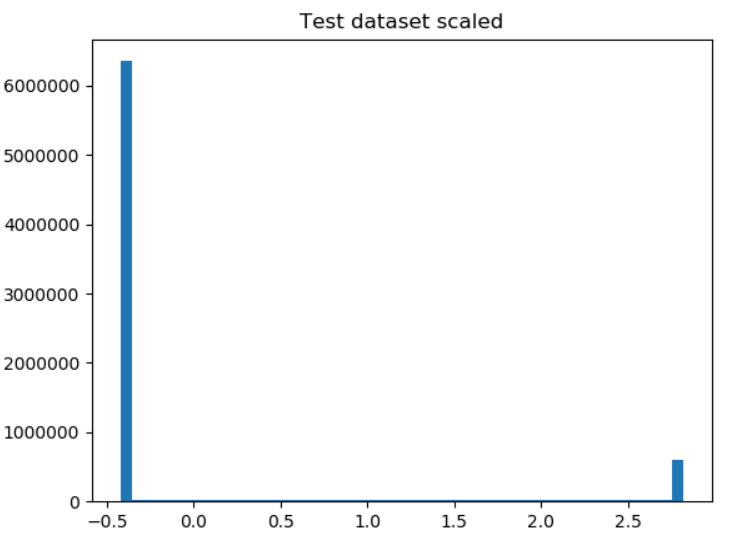
**Hours:** 8:00 – 12:00 (Saturdays) **Room:** 320

# Homework 4

Due Saturday, June 22 at 8:00am.

## Practice 1

* Load the MNIST dataset unscaled pickle file (mnist\_dataset\_unscaled.pickle)
* Apply standardization of the pixel values
  + Remove the mean, and divide by the standard deviation
  + Do it to both training and test data
* Plot a histogram of all the pixel values
  + Do it for the training data, and for the test data
* Plot the first 5 digits as images

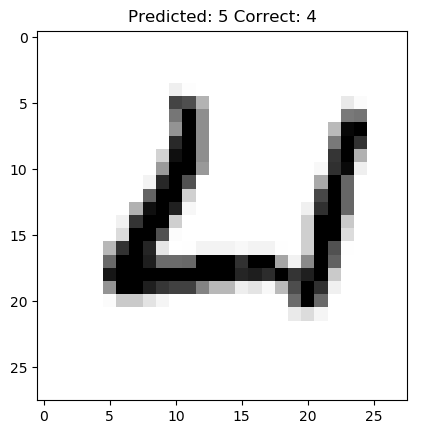
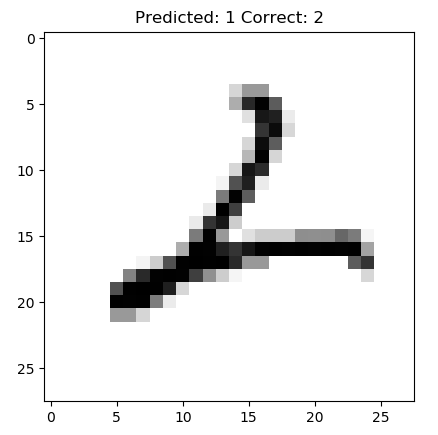
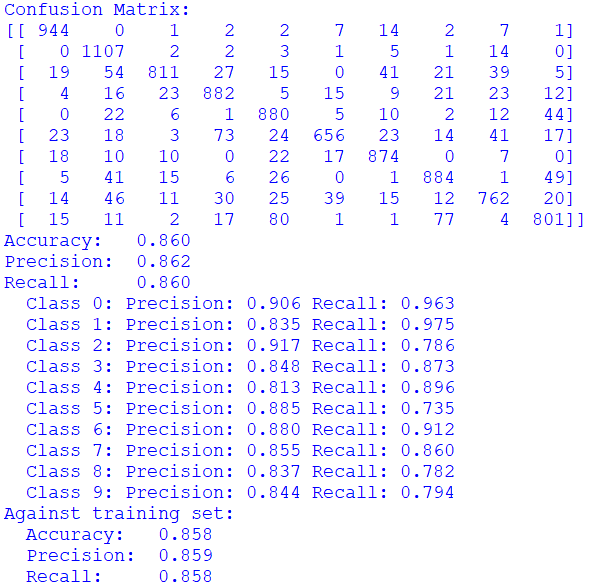


## Optional Problem 1b

* Repeat the Naive Bayes classifier taking into account class imbalance
  + Based on Mnist\_NaiveBayes\_3.py
* Compute the probability of every class *p(Ck)* 
  + From the counts of labels in the training set
* Use these factors to scale the probability images of every class
  + Did results improve?

## Practice 2

* Load the MNIST unscaled pickle file (mnist\_dataset\_unscaled.pickle)
* Scale the training and test data
* One-hot encode the labels
* Train a linear regression classifier
* Compute the predicted labels for the test data
* Compute and print the accuracy score
* Display 10 images of digits that were predicted incorrectly
  + Show the predicted and correct labels



## Practice 3

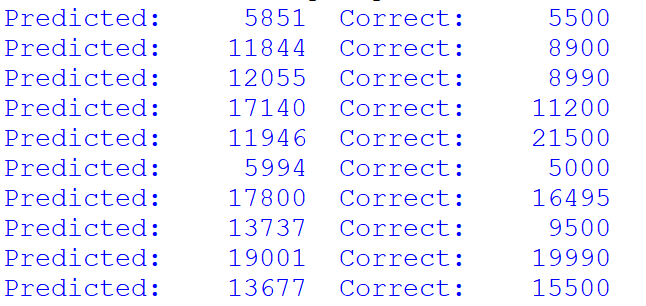
* Repeat the example in Mnist\_LogisticRegression\_1.py but use OvR classifications instead of multinomial optimization
  + Is it easier or harder to converge?
  + Are the results the same, better or worse?

## Optional Problem 3b

* Study the effects of image resolution on Softmax regression classification accuracy
  + Create scaled datasets at various resolutions
  + Run a softmax regression classifier on each of the sets
  + Note the prediction accuracy of each data set
* Compare the results with the behavior observed with random forest classifiers

## Practice 4

* Use Tensorflow to solve the Normal equations of a linear regression model of vehicle prices
  + See VehiclePrice\_LinearRegression\_tf\_1.py
* Add TensorFlow instructions to compute the prediction values against the test dataset
  + Print the first 10 prediction values
  + Compare with the first 10 correct label values



## Practice 5

* Use Tensorflow to to solve a linear regression model for vehicle prices using gradient descent
  + You can base it on VehiclePrice\_LinearRegression\_tf\_4.py
* Add Tensorflow instructions to compute the RMSE of the test predictions against the test labels



## Optional Problem 5b

* Adjust the learning rate in the gradient descent solution of the vehicle price linear regression
  + What happens if the learning rate is very large?
  + What happens if it is very small?