CS1675 - Assignment 6

Zachary M. Mattis

February 28, 2019

I. Problem 1 - Support Vector Machines

a. Weight / Bias

Attribute	Weight
1	0.0445
2	0.0116
3	-0.0050
4	0.0003
5	-0.0002
6	0.0264
7	0.3524
8	0.0063

Table 1: Training Weights

Bias	-2.7182

Table 2: Training Bias

b. SVML

`apply_svml.m`

$$w^T x + b \geq 0$$

c. Misclassification, Confusion, Sensitivity, Specificity

Dataset	Misclassification Error	Sensitivity	Specificity
Training	0.2319	0.5750	0.8820
Testing	0.1965	0.6176	0.8820

Table 3: Misclassification Error, Sensitivity, Specificity

Predict / Target	1	1
1	115	40
0	85	299

Table 4: Training Confusion Matrix

Predict / Target	1	1
1	42	19
0	26	142

Table 5: Test Confusion Matrix

d. LR. vs. NB vs. SVML

Model	Misclassification	Sensitivity	Specificity
LR	0.2988	0.5900	0.8761
NB	0.4644	0.8400	0.5398
SVML	0.2319	0.5750	0.8820

Table 6: Training Model Comparison

Model	Misclassification	Sensitivity	Specificity
LR	0.2722	0.6765	0.8323
NB	0.3759	0.8676	0.5093
SVML	0.1965	0.6176	0.8820

Table 7: Testing Model Comparison

In all cases, the SVML model performed better than both Logistic Regression and Naïve Bayes from the assignment 5 implementation, with a lower sensitivity and a higher specificity.

II. Problem 2 - Deep Learning

a. Deep Learning Toolbox

`logistic_NN.m`

b. Multi-layer Neural Network

Model	Hidden Layers	Hidden Units	Misclassification
LR	0	-	0.2356
NN	1	2	0.2263
NN	1	3	0.2171
NN	1	5	0.2263
NN	1	10	0.2393

Table 8: Training NN Comparison

Model	Hidden Layers	Hidden Units	Misclassification
LR	0	-	0.2402
NN	1	2	0.2227
NN	1	3	0.2271
NN	1	5	0.2096
NN	1	10	0.1965

Table 9: Testing NN Comparison

Overall, the Neural Network model produced better Misclassification errors than the Logistic Regression model on both training and testing data, proving to be a superior model in this case.