Anthony Tummillo

HW Report #6

Problem 1.

(a)

Code is written as the function normalize_data.m which was included in the code submission.

(b)

Done

(c)

weights:

w0 = -2.3183
w1 = 2.9235
w2 = 3.7750
w3 = -0.8629
w4 = -2.0081
w5 = -2.3304
w6 = 1.2046
w7 = 0.4079
w8 = 1.9199

(d)

Train error: 0.3061

Test error: 0.2707

Train matrix:

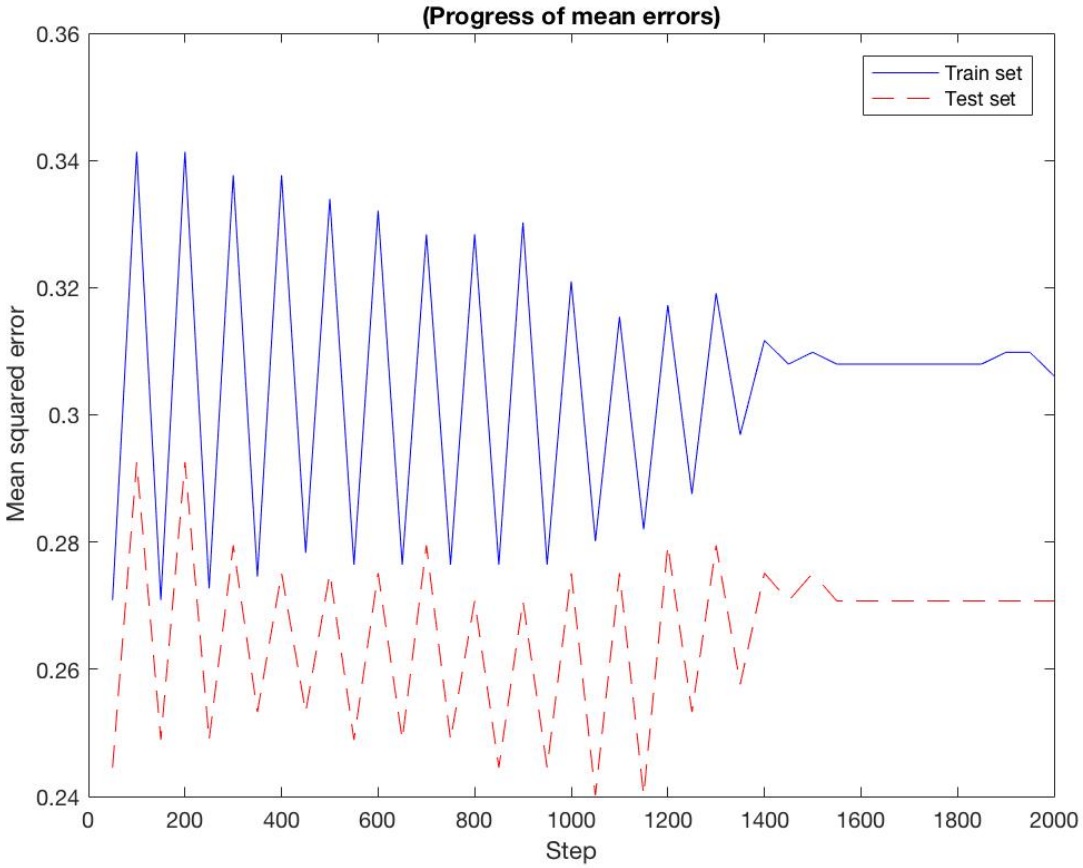
111	76
89	263

Test matrix:

46	40
22	121

Test sensitivity = 0.6765

Test specificity = 0.7516



This graph was generated for the model described in the instructions without any alteration.

(e)
With weights initialized to 0:
Train error = 0.2542
Test error = 0.2358

With learning rate of 0.05
Train error = 0.2913
Test error = 0.2969

With epoch of 100
Train error = 0.3061
Test error = 0.2707

The best result I got was a Train error of 0.2542 and a Test error of 0.2358. This was achieved by only changing the initialization of the weights from 1 to 0.

Problem 2.1

Part a.
Done

Part b.

Attribute 1: exponential
Attribute 2: normal
Attribute 3: normal
Attribute 4: normal
Attribute 5: exponential
Attribute 6: normal
Attribute 7: exponential
Attribute 8: exponential

Problem 2.2

Part a.
Function is written

Part b.

Class 0 prior: 0.6289
Class 1 prior: 0.3711

exp_0_1_muhat: 3.2419	exp_1_1_muhat: 4.71
exp_0_5_muhat: 67.7168	exp_1_5_muhat: 103.72
exp_0_7_muhat: 0.4164	exp_1_7_muhat: 0.5491
exp_0_8_muhat: 31.1032	exp_1_8_muhat: 37.12

norm_0_2_mu: 109.6254	norm_1_2_mu: 141.395
norm_0_3_mu: 67.5339	norm_1_3_mu: 70.19
norm_0_4_mu: 19.7316	norm_1_4_mu: 22.935
norm_0_6_mu: 30.3059	norm_1_6_mu: 35.258

norm_0_2_sigma: 26.2304	norm_1_2_sigma: 33.6655
norm_0_3_sigma: 18.6683	norm_1_3_sigma: 21.6213
norm_0_4_sigma: 14.5828	norm_1_4_sigma: 17.8275
norm_0_6_sigma: 7.7258	norm_1_6_sigma: 7.3286

I have included the data that proves to be useful to our Naïve Bayes classifier even though some more was returned from each call of expfit and normfit.

Problem 2.3

Part (a)
Done

Part(b)
Train error = 0.2393

Test error = 0.2271

Train Confusion Matrix

121	50
79	289

Test Confusion Matrix

39	23
29	138

Sensitivity = 0.5735 (for test)
Specificity = 0.8571 (for test)

Part (c)
The Naïve Bayes model has lower test and train error, and has both higher specificity and lower sensitivity than the Logistic Regression model does. As of now the Naïve Bayes model appears to do a better job of predicting the class of the data than the logistic regression does (without altering anything in logistic regression).

Problem 3.

I used a cost of 1

Train error = 0.2319

Test error = 0.1965

Train Confusion Matrix

115	40
85	299

Test Confusion Matrix

42	19

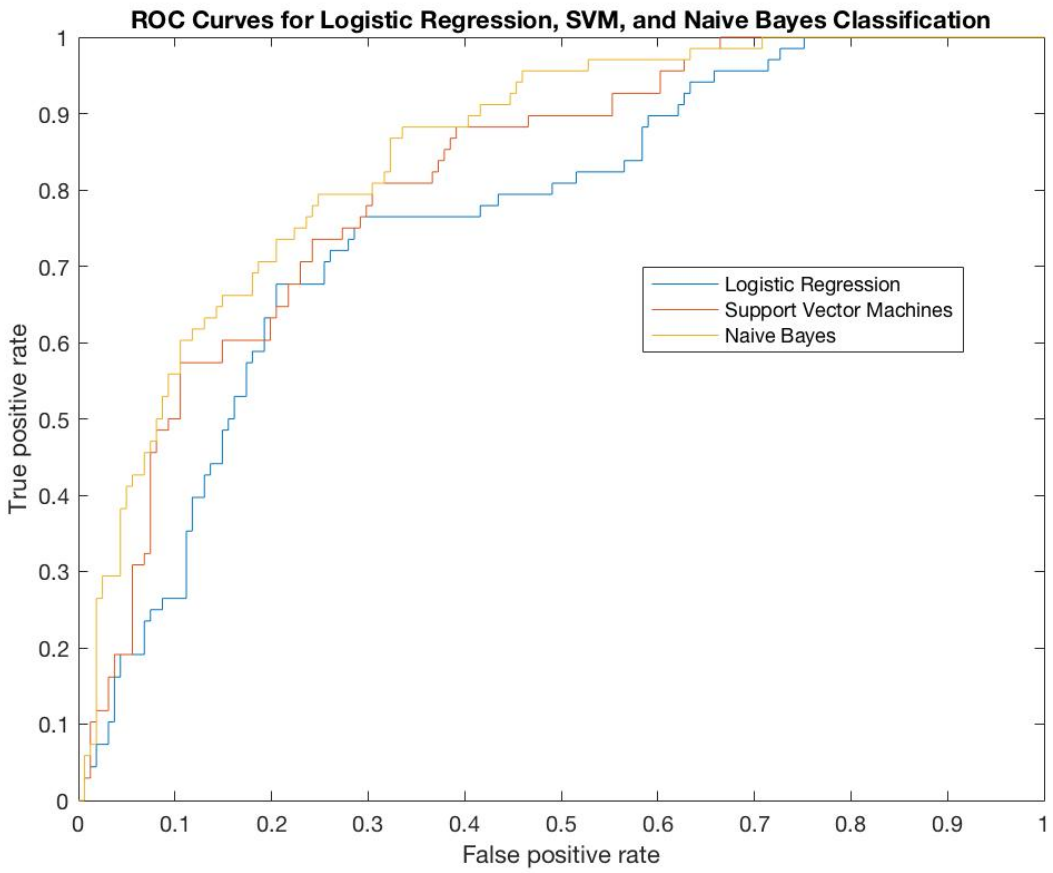
This model has a lower train error and test error than both the Logistic Regression and Naïve Bayes models. So far it appears that the SVM model has done the best job of predicting the class for input data.

Problem 4.

Part a.
Done.

Part b.
Done.

Part c.



AUClog: 0.7633

AUCnb: 0.8150

AUCsvm: 0.8497

Based on these AUC values (higher AUC meaning that the model does a better job of predicting the class for input data) it appears that my previous conclusions are correct. The SVM model does the best job, Naïve Bayes does the second-best job, and Logistic Regression (using the model described by the assignment) does the third-best job of predicting the class based on input data.