ECGR 5105 Homework 3

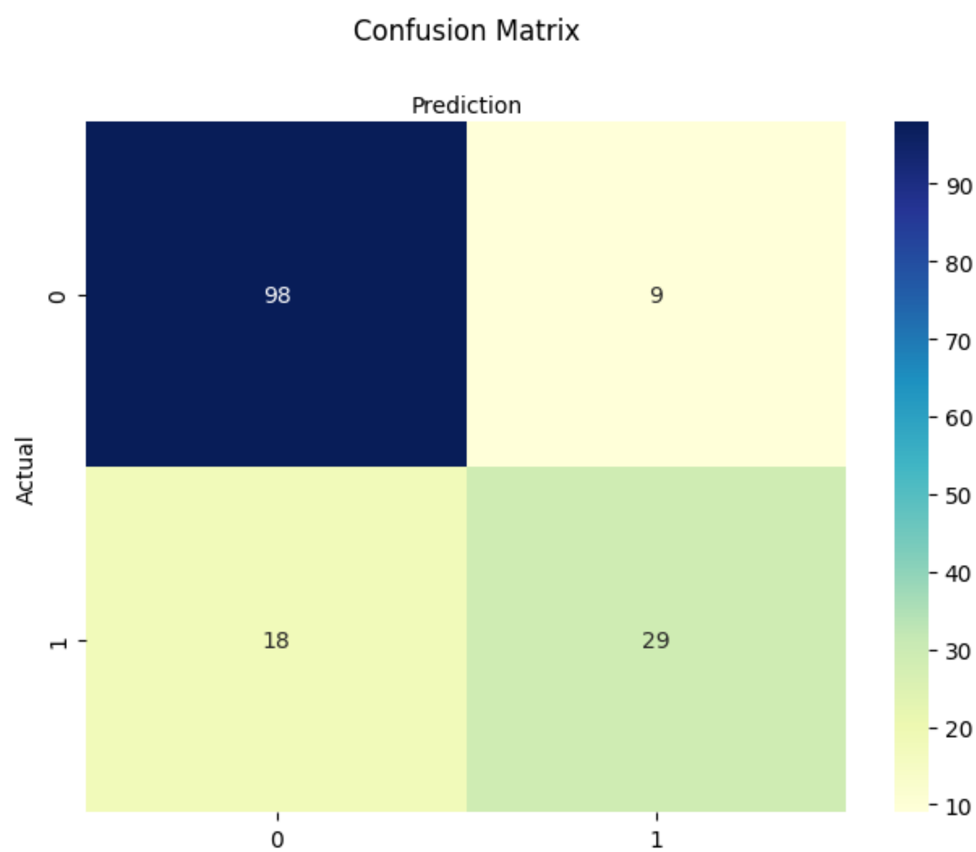
Owen Bailey-Waltz (801488178)
GitHub Link: https://github.com/obaileyw-uncc/ecgr5105/hw03_classification_binary

Note: in all models, g refers to the sigmoid/logistic function.

Problem 1: Diabetes dataset

Gradient descent and input feature standardization for the logistic regression in this problem was performed using the SciKit Learn Python library. The library does not allow the probing of training and validation loss values, and these values therefore could not be obtained. Results from training this model automatically using SciKit Learn are provided below.

$$h(\boldsymbol{x}) = g(-0.788 + 0.310x_1 + 1.060x_2 - 0.261x_3 + 0.069x_4 - 0.158x_5 + 0.684x_6 + 0.294x_7 + 0.240x_8)$$



TRAINING ACCURACY 0.7622
VALIDATION ACCURACY 0.8247

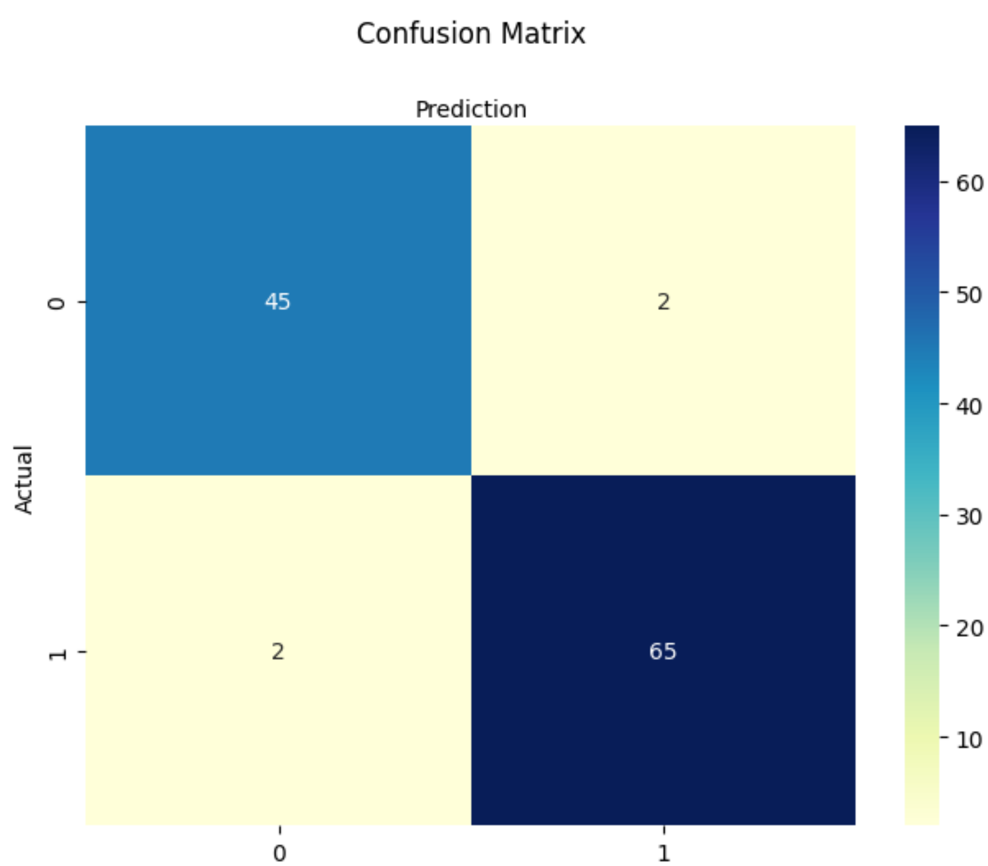
PRECISION 0.7632
RECALL 0.6170
F1 SCORE 0.6824

Problem 2: Breast cancer dataset

(a) No weight penalty

Gradient descent and input feature standardization for the logistic regression in this problem was performed using the SciKit Learn Python library. The library does not allow the probing of training and validation loss values, and these values therefore could not be obtained. Results from training this model automatically using SciKit Learn are provided below. All thirty features were used as inputs to the model.

$$\begin{aligned} h(\boldsymbol{x}) = g(0.248 &- 0.351x_1 - 0.489x_2 - 0.341x_3 - 0.408x_4 - 0.192x_5 + 0.448x_6 - 0.669x_7 \\ &+ 0.845x_8 - 0.338x_9 + 0.213x_{10} - 1.391x_{11} + 0.039x_{12} - 0.857x_{13} \\ &- 0.971x_{14} + 0.251x_{15} + 0.667x_{16} + 0.121x_{17} - 0.222x_{18} + 0.120x_{19} \\ &+ 0.865x_{20} - 0.932x_{21} - 1.041x_{22} - 0.767x_{23} - 0.890x_{24} - 0.536x_{25} \\ &- 0.020x_{26} - 0.870x_{27} - 0.975x_{28} - 0.515x_{29} - 0.611x_{30}) \end{aligned}$$



TRAINING ACCURACY 0.9890
VALIDATION ACCURACY 0.9649
PRECISION 0.9701
RECALL 0.9701
F1 SCORE 0.9701

(b) With weight penalty

The following examples repeated part (a) using L1 weight penalties with λ values of 10, 1, 0.1, 0.01 and 0.001.

λ = 10

$$\begin{aligned} h(\boldsymbol{x}) = g(&-1.223 - 0.029x_2 - 0.870x_5 + 4.451x_6 - 3.293x_7 - 1.951x_8 - 0.621x_9 \\ &- 4.328x_{11} + 0.451x_{12} - 4.090x_{14} + 0.400x_{15} + 1.895x_{17} + 2.786x_{18} \\ &- 0.262x_{19} + 6.410x_{20} - 0.575x_{21} - 2.913x_{22} - 6.795x_{24} + 0.250x_{26} \\ &- 1.991x_{27} - 0.216x_{28} - 0.586x_{29} - 5.503x_{30}) \end{aligned}$$

Confusion matrix

Predicted → Actual ↓	0	1
0	44	3
1	3	64

Parameters

TRAINING ACCURACY 0.9890
VALIDATION ACCURACY 0.9474
PRECISION 0.9552
RECALL 0.9552
F1 SCORE 0.9552

λ = 1

$$\begin{aligned} h(\boldsymbol{x}) = g(&-0.075x_2 - 0.833x_8 - 0.186x_9 - 2.936x_{11} + 0.646x_{16} + 0.623x_{20} - 1.383x_{21} \\ &- 1.399x_{22} - 0.106x_{23} - 2.637x_{24} - 0.279x_{25} - 0.917x_{27} - 1.683x_{28} \\ &- 0.497x_{29} - 0.175x_{30}) \end{aligned}$$

Confusion matrix

Predicted → Actual ↓	0	1

0	44	3
1	2	65

Parameters

TRAINING ACCURACY 0.9890

VALIDATION ACCURACY 0.9561

PRECISION 0.9559

RECALL 0.9701

F1 SCORE 0.9630

λ = 0.1

$$h(\boldsymbol{x}) = g(0.343 - 0.472x_8 - 0.555x_{11} - 1.998x_{21} - 0.533x_{22} - 0.130x_{25} - 1.021x_{28} - 0.179x_{29})$$

Confusion matrix

Predicted → Actual ↓	0	1
0	45	2
1	1	66

Parameters

TRAINING ACCURACY 0.9758

VALIDATION ACCURACY 0.9737

PRECISION 0.9706

RECALL 0.9851

F1 SCORE 0.9778

λ = 0.01

$$h(\boldsymbol{x}) = g(-0.123x_{21} - 0.193x_{23} - 0.468x_{28})$$

Confusion matrix

Predicted → Actual ↓	0	1
0	45	2
1	7	60

Parameters

TRAINING ACCURACY 0.9275

VALIDATION ACCURACY 0.9211

PRECISION 0.9677

RECALL 0.8955

F1 SCORE 0.9302

λ = 0.001

$$h(\boldsymbol{x}) = g(0)$$

Confusion matrix

Predicted → Actual ↓	0	1
0	47	0
1	67	0

Parameters

TRAINING ACCURACY 0.3626

VALIDATION ACCURACY 0.4123

PRECISION 0

RECALL 0

F1 SCORE 0

Additional commentary

Reducing the λ factor in the parameter penalty case to 0.001 penalizes large parameters to such a degree where the parameter values go to zero, causing the model to become completely insensitive to any stimulus. The maximum sensitivity and maximum recall values are both achieved with a λ of 0.1, with values above and below this penalty amount resulting in a lower score in both quality categories. Until all parameters begin to vanish due to floating point constraints, the lower values of λ resulted in an observed increased model sensitivity with more total true predictions present in the test class.