Matthew Cline
Machine Learning
Assignment 2: Logistic Regression
11 October 2017

Abstract
The objective of Assignment 2 was to implement a logistic regression model to be used as a classifier on the same Flu dataset that was used in assignment 1. The logistic regression algorithm was tested using various feature sets, regularization techniques, and feature scaling. The following sections detail the implementation choices that were made to accommodate each scenario and the evaluation of the resulting models.

Implementation
My personal goal for Assignment 2 was to generalize my algorithms enough to work on all the different feature sets that would be evaluated throughout the assignment. I also wanted the functions to have built in options for a regularization term. The first step in implementing a set of general algorithms was ingesting the data and preparing it for use in the algorithm.

The data was read in from the Excel file as a Pandas Dataframe, taking only the columns that would be needed for all of the models being developed in the assignment. The columns ingested were HndWshQual, Risk, KnowlTrans, Flu, and Gender. The data was then shuffled and the indices were reset. For each model, the data would be split into a matrix containing the appropriate feature set with a column filled with ones for the bias term. The Flu column was split off into a single vector to be used for the labels. Finally a vector for the thetas was initialized for each model.

The power of logistic regression as a classifier comes from passing the linear hypothisis function documented in Equation 1 through the Sigmoid function documented in Equation 2.

$$h_0(x) = \theta_0 + \theta_1 x \ldots + \theta_m x_m$$
Equation 1: Linear Hypothesis Function

$$f(x) = \frac{1}{1+e^{-x}}$$
Equation 2: Sigmoid Function

By passing the linear hypothesis function through the Sigmoid function, the range of the function is limited to (0,1), and the uncertainty around the decision boundary is minimized. The use of matrices and the dot product kept my functions scalable to datasets with many different feature spaces.

The cost function had to be modified slightly to take into account the logistic nature of the regression, but it is still built on the same principals as the linear regression cost function. The logistic cost function is documented in Equation 3.

$$J(\theta) = -1/m \sum_{i=1}^{m} y^{(i)} log(h(x^{(i)})) + (1 - y^{(i)}) log(1 - h(x^{(i)})$$

Equation 3: Logistic Cost Function

The gradient descent function was used to optimize the values of theta similarly to the linear regression assignment. For logistic regression the same partial derivative technique is used with respect to each of the theta values combined with the learning rate alpha. Due to the small size of the data set being trained on and the efficiency of modern computers alpha was not optimized, and a value 0.01 was used for the training of all the models. A higher alpha value would minimize the number of iterations needed in training a larger data set. A cost change threshold of 0.000001 was also used in the training of all the models.

The actual implementation of all of the algorithms discussed above can be seen in the supplementary file main.py, contained within the zip file with this document. The following sections will detail the results of each of the models that were tested. All of the models were trained on 60% of the total clean data. The threshold for the classification was optimized for each model on a validation data set that was composed of 20% of the total clean data. Finally each model was evaluated against a test data set that represented the final 20% of the total clean data. The same training, validation, and test data was used in the evaluation of each model.

Single Variable Logistic Regression

The first model that was evaluated was a single variable regression. The model was trained to classify a student as getting the Flu (‘Flu’ == 1) or not based on the student’s risk. Figure 1 shows a graph of the training iterations during gradient descent versus the cost associated with each set of theta values.

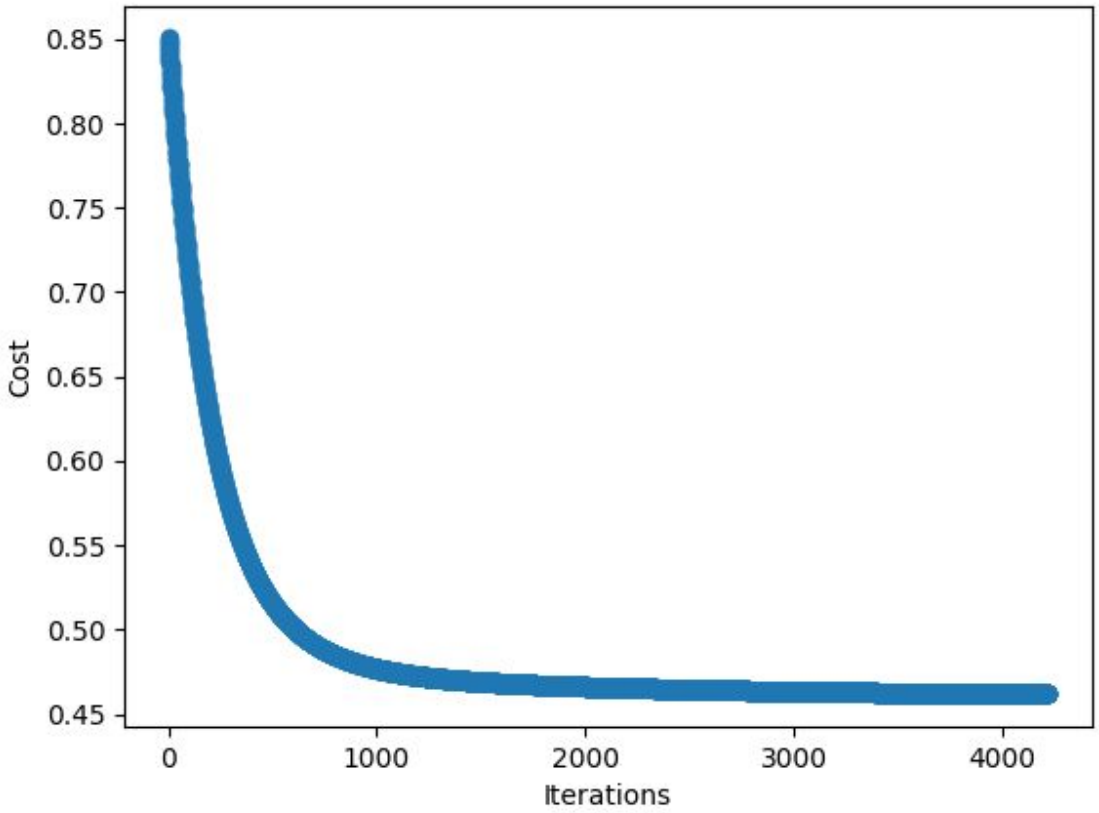

Figure 1: Single Feature Regression

The gradient descent function completed 4,221 iterations due to the low value of alpha, and the final cost associated with the learned thetas was 0.4617. The final theta values can be seen with the results of the evaluation in Table 1. After training was completed, the classification threshold was optimized on the validation data based on the F1 score of each iteration. The initial threshold was set to 0.1 and incremented by 0.001 each iteration through a value of 0.99. Obviously in most situations thresholds on either end of the interval will be useless, but the methodology provided a quick way to optimize the threshold systematically. The final threshold selected was 0.269. A confusion matrix containing the results of the evaluation of the model against the test data set is a subset of the results shown in Table 1.

Theta Values	Precision = 0.58	Recall = 0.54	F1 Score = 0.56
Bias = -1.29561	TP = 7	FP = 5	
Risk = 1.72486	FN = 6	TN = 53	

Table 1: Single Variable Results

Analysis of the results can be found in the discussion section after the results of each model have been presented.

Logistic Regression with 2 Variables

The second model that was evaluated added another variable to the feature space. In this model both Risk and Hand Wash Quality were mapped to the likelihood of a student contracting the Flu. The graph showing the number of iterations performed during gradient descent is shown in Figure 2.

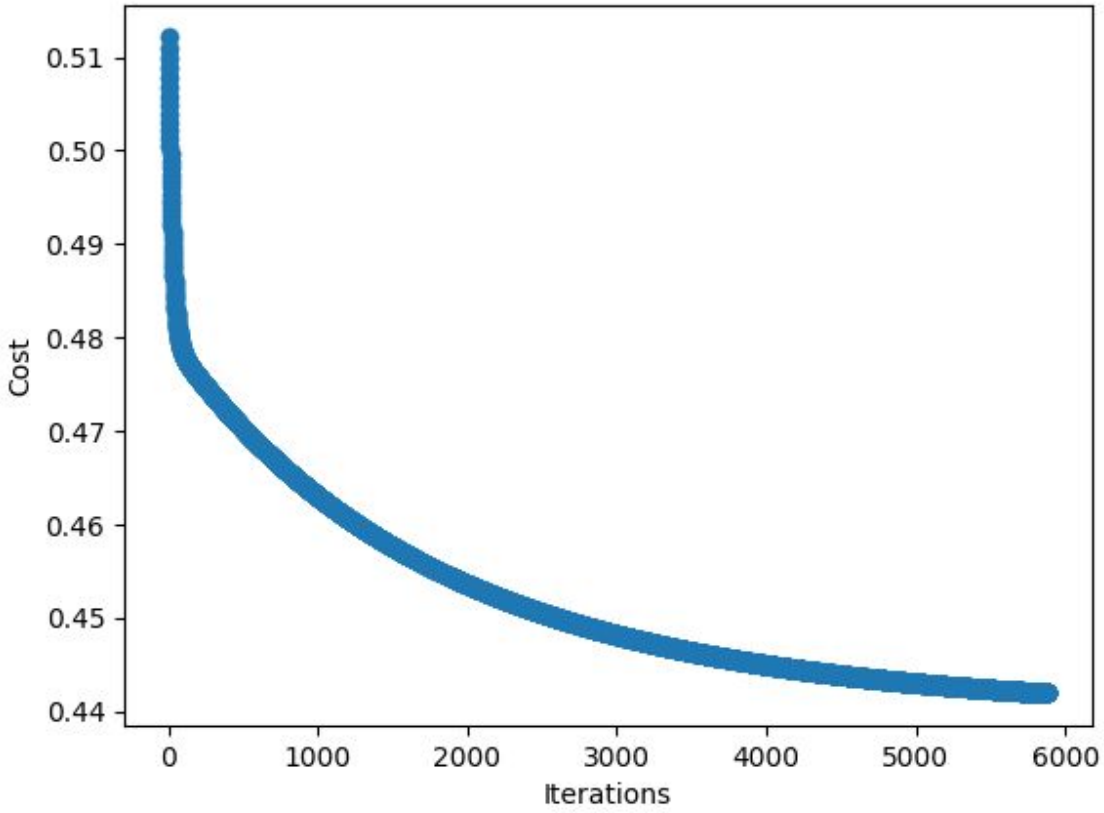

Figure 2: 2 Variable Regression

The gradient descent algorithm completed after 5,884 iterations.The convergence of the gradient descent function ended with a final cost of 0.44197. The classification threshold was again optimized on the validation data set and settled at a value of 0.115. The low value of the threshold can be attributed to the fact that the Hand Wash Quality data was not feature scaled to a similar representation as the Risk values. The results and final theta values are documented in Table 2.

Theta Values	Precision = 0.39	Recall = 1.0	F1 Score = 0.47
Bias = 0.699015	TP = 13	FP = 29	
Risk = 1.826021	FN = 0	TN = 29	
HndWsh = -0.5286			

Table 2: 2 Variable Regression Results

3 Variable Logistic Regression

The 3 varaible logistic regression model added the Knowledge of Transmission variable to the featurespace. The graph showing the convergence of the cost function during gradient descent is shown in Figure 3.

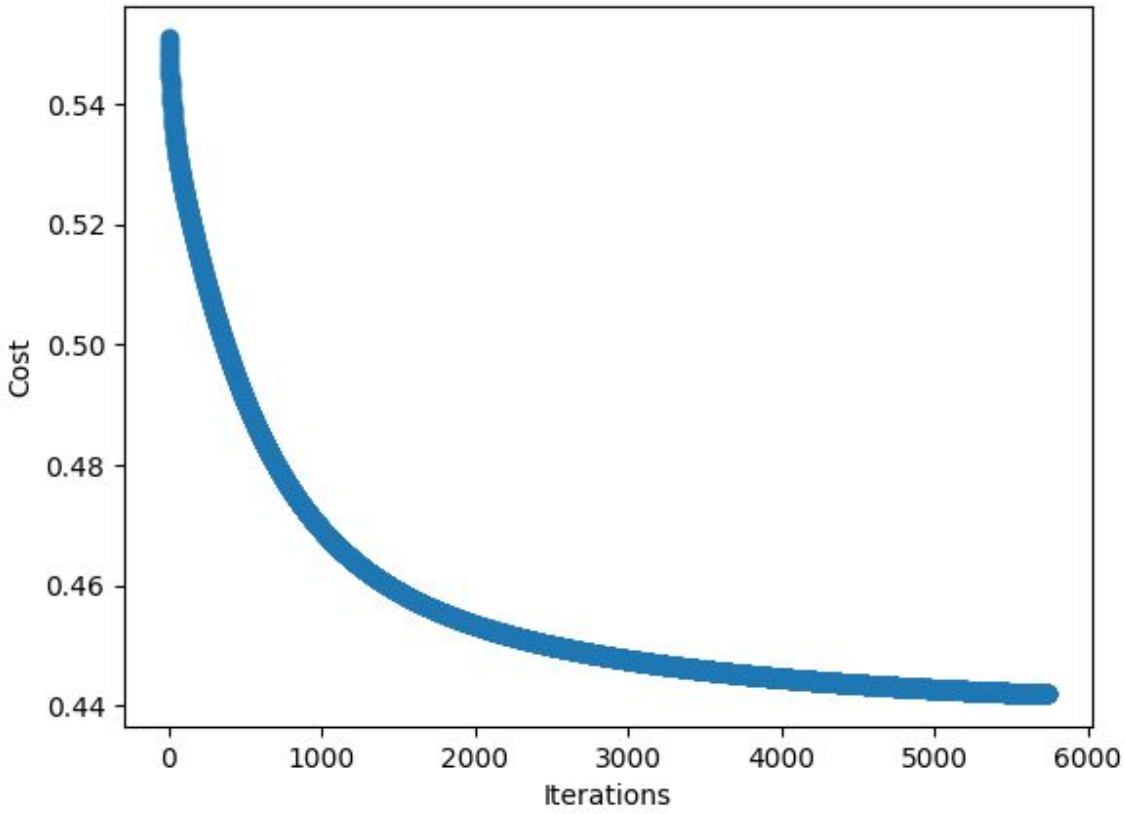

Figure 3: 3 Variable Gradient Descent

The gradient descent function converged after 5,743 iterations, with a final cost value of 0.441975 for the training data. Affter threshold optimization carried out on the validation data set, the final threshold was set at 0.129. The results including the final theta values, confusion matrix and F1 scrore are documented in Table 3.

Theta Values	Precision = 0.35	Recall = 1.0	F1 Score = 0.52
Bias = 0.715060	TP = 13	FP = 24	
Risk = 1.821654	FN = 0	TN = 34	
HndWsh = -0.531615			
KnowlTrans = -0.044			

Table 3: 3 Variable Regression Results

4 Variable Logistic Regression

The four variable regression model restored the feature space to its original state making use of Risk, Hand Wash Quality, Knowledge of Transmission, and Gender. Again no feature scaling has been used on any of the models up to this point. The Gender variable is the first binary variable that has been introduced to the model. The gradient descent convergence graph is shown in Figure 4.

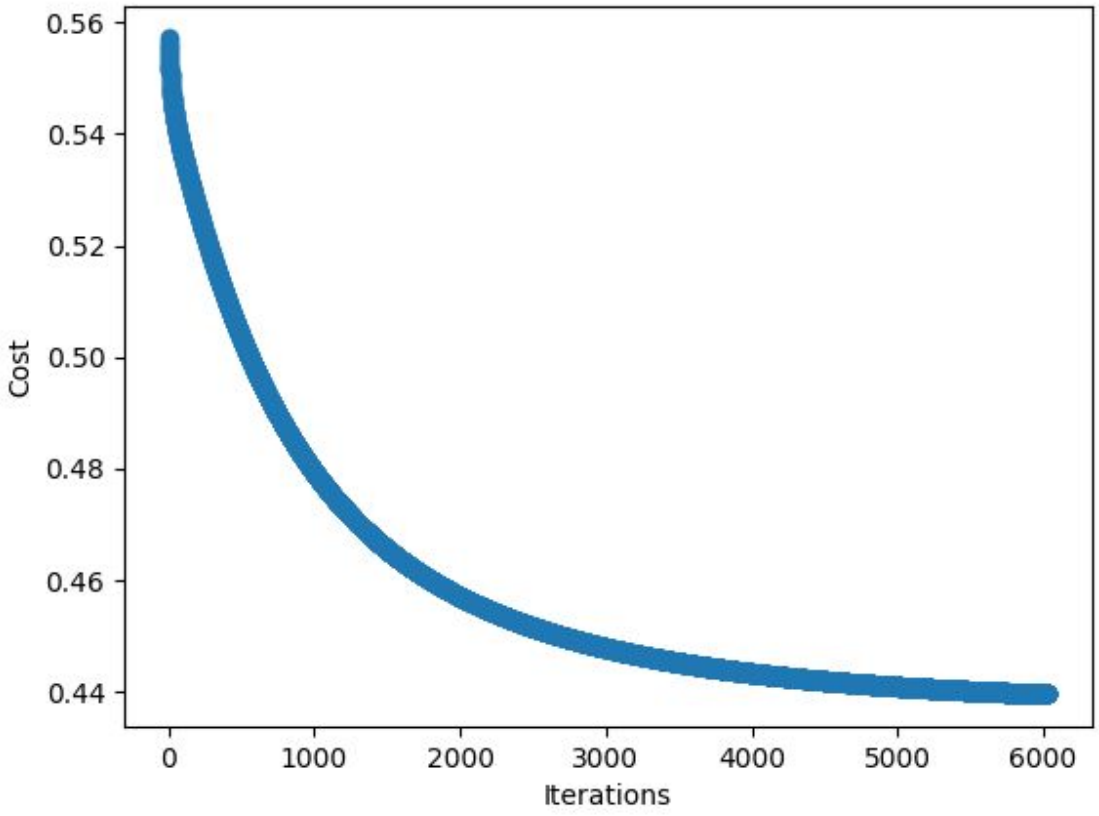

Figure 4: Four Variable Convergence Graph

The gradient descent function converged after 6,033 iterations, with a final cost of 0.43963. The classification threshold was optimized and finalized with a value of 0.142. The resulting theta values and evaluation results are shown in Table 4.

Theta Values	Precision = 0.35	Recall = 1.0	F1 Score = 0.52
Bias = 0.636119	TP = 13	FP = 24	
Risk = 1.895883	FN = 0	TN = 34	
HndWsh = -0.576705			
KnowTrans = -0.0169			
Gender = 0.438711			

Table 4: Four Variable Logistic Regression Results

Four Variable Logistic Regression with Regularization

The next model tested was the same four variable model from the previous section, but with a regularization term used to help prevent overfitting. The regularization term is added to the cost function to penalize certain features from impacting the classification when they really do not contribute. The regularization term λ is shown in the updated cost function in Equation 4.

$$J(\theta) = -1/m\,[\sum_{i=1}^{m} y^{(i)} log(h(x^{(i)})) + (1 - y^{(i)}) log(1 - h(x^{(i)}) + \lambda \sum_{j=1}^{n} \theta_j^2\,]$$

Equation 4: Cost Function with Regularization

The regularization term can be used to prevent overfitting in the model. The higher the regularization term, the less chance there will be for overfitting, because the features will be de-emphasized. Figures 5 shows the gradient descent convergence graphs for λ values of 5, 10, and 25 respectively.

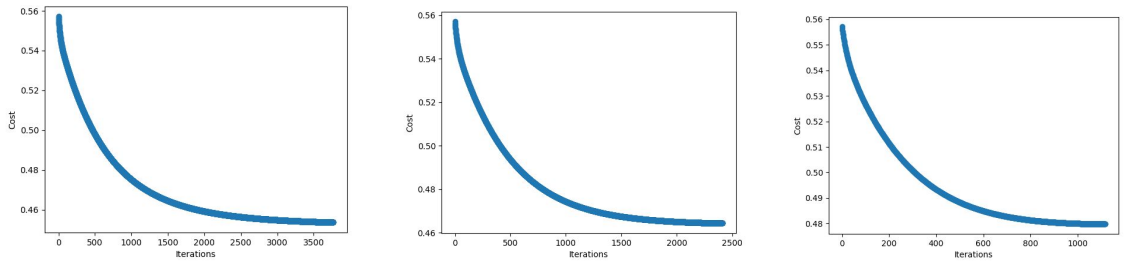

Figure 5: Convergence Graphs with Regularization Terms of 5, 10, and 25

The higher the regularization term value, the quicker the gradient descent function converged. Higher regularization values also resulted in lower theta values causing less emphasis to be put

on the features during predictions. The theta values and the results from each model are shown in Tables 5, 6, and 7.

Theta Values	Precision = 0.325	Recall = 1.0	F1 Score = 0.49
Bias = 0.288752	TP = 13	FP = 27	
Risk = 1.079899	FN = 0	TN = 31	
HndWsh = -0.447409			
KnowTrans = -0.0535			
Gender = 0.282806			λ = 5

Table 5: Results with Regularization Term of 5

Theta Values	Precision = 0.317	Recall = 1.0	F1 Score = 0.48
Bias = 0.231035	TP = 13	FP = 28	
Risk = 0.768623	FN = 0	TN = 30	
HndWsh = -0.425369			
KnowTrans = -0.0361			
Gender = 0.264248			λ = 10

Table 6: Results with Regularization Term of 10

Theta Values	Precision = 0.28	Recall = 0.62	F1 Score = 0.38
Bias = 0.193502	TP = 8	FP = 21	
Risk = 0.447041	FN = 5	TN = 37	
HndWsh = -0.409757			
KnowTrans = 0.0343			
Gender = 0.266244			λ = 25

Table 7: Results with Regularization Term of 25

As shown in the results table, the larger the regularization term the worse the model performed. These results imply that the model was not suffering from overfitting, and applying more regularization caused the model to underfit the data.

Four Variable Logistic Regression with Regularization and Feature Scaling

The final model that was evaluated took the model from the previous section using a regularization factor of 5 and applied feature scaling to all of the features, except the binary gender feature, before training the model. The convergence graph of the gradient descent function is shown in Figure 6.

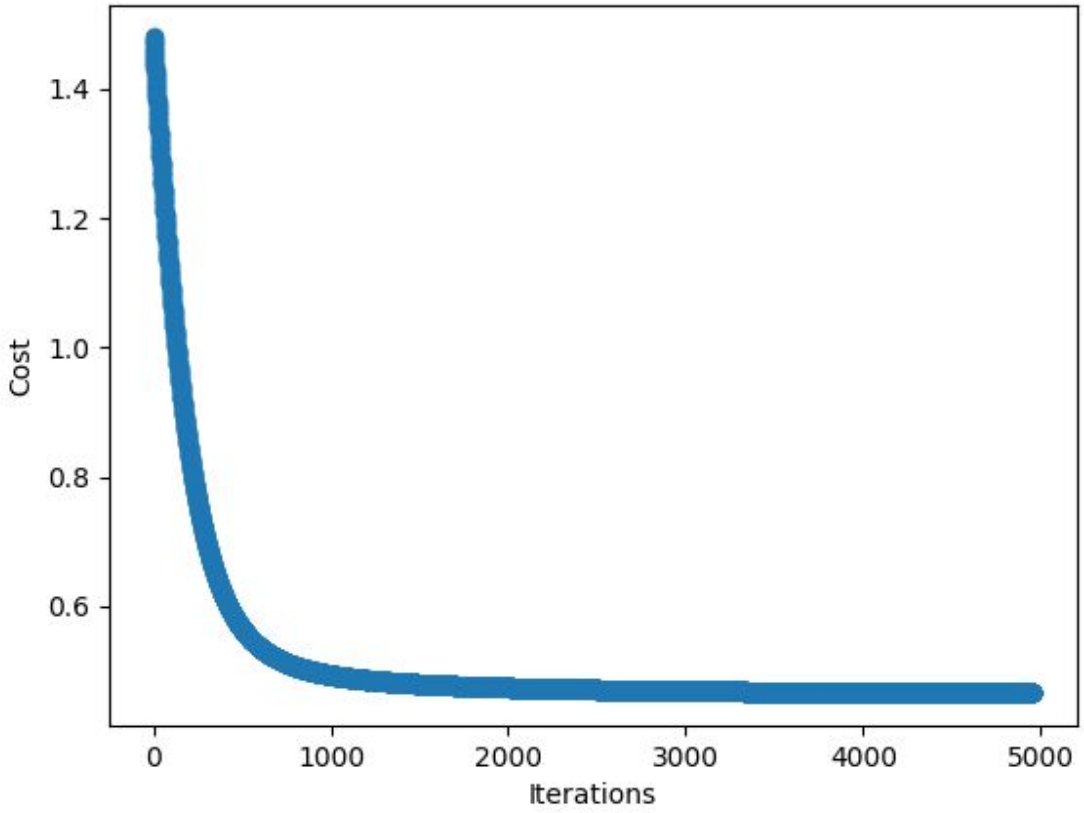

Figure 6: Featured Scaled Convergence Graph

The gradient descent function completed 4,958 iterations before converging with a final cost of 0.46484. After optimizing the classification threshold using the validation data set, the final threshold was set to 0.181. The results of the evaluation on the test data are shown in Table 8.

Theta Values	Precision = 0.317	Recall = 1.0	F1 Score = 0.48
Bias = -1.281078	TP = 13	FP = 28	

Risk = 0.912287	FN = 0	TN = 30	
HndWsh = -0.361630			
KnowTrans = -0.0645			
Gender = -0.095551			

Table 8: Results with Feature Scaling

Analysis

The best performing model by far was the most simple model mapping only Risk to the target variable. However none of the models were particularly effective at predicting the appropriate outcome. Because many of the students in the data set had not contracted the flu, the models were rewarded more for identifying the positive identifications than negative identifications when optimizing the classification threshold based on F1 scores. This led to low threshold values for the models and eliminated most of the false negatives.

The models with more features each took more iterations to train due to the increased complexity of learning more theta values. The extra iterations were offset by the regularization term in the later models however. Regularization was not beneficial in this case though because the model was not overfitting to start with. With the higher regularization term values, the models began to significantly underfit.

If repeating the experiments in Assignment 2, I would explore other ways of optimizing the classification threshold due to the small size of my validation data set. Optimizing for such a small subset of the data probably hurt the general performance of the models in the test set, creating a type of overfitting that could not be addressed by the regularization term.