
Cancer Classification using Logistic Regression

Abhishek S. Dalvi
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
adalvi@buffalo.edu

Abstract

The purpose of this project is to predict a tumor as Benign or Malignant using features provided for the tumor. The classification will be done by a machine learning model using logistic regression which will be trained on a set of observations. The model will be tested on unseen data to calculate the accuracy metrics of the model.

1 Introduction

Logistic regression is the appropriate regression analysis to conduct when the variable to be predicted is binary (in this case B or M). Similar to other regression models, like linear regression, it is a prediction for the dependent binary variable; by trying to find a relationship between the dependent binary variable and other variables which are independent or dependent of each other. The main task is to find the relationship which is done by creating an iterative training model which introduces weights which can be associated as the relationship between the target binary variables and the independent variables.

2 Dataset

Wisconsin Diagnostic Breast Cancer (WDBC) dataset is used for training, validation and testing the model. The dataset contains 569 instances with 32 attributes (ID, diagnosis (B/M), 30 real-valued input features). Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass.

Computed features describes the following characteristics of the cell nuclei present in the image:

Table 1: Dataset Features

1	Radius (mean of distances from center to points on the perimeter)
2	Texture (standard deviation of gray-scale values)
3	Perimeter
4	Area
5	Smoothness (local variation in radius lengths)
6	Compactness ($\text{perimeter}^2/\text{area} - 1.0$)
7	Concavity (severity of concave portions of the contour)
8	Concave points (number of concave portions of the contour)
9	Symmetry
10	Fractal dimension ("coastline approximation" - 1)

The mean, standard error, and “worst” or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. The dataset was divided into 3 parts; training data, validation and testing data where each contained 80%, 10% and 10% of the total dataset respectively.

3 Pre-Processing

3.1 Replacing Binary Terms

The dataset contains the ids of the data which should be removed before calculation. The diagnosis values “Malignant” and “Benign”, which are represented by “M” and “B” will be replaced by another set of binary values; “1” and “0” respectively for ease of calculation.

3.1 Splitting Data into Input and Target Variables

The dataset is split into Input(X) and Targets(Y) which will help in further calculations for finding the relationship between them.

3.3 Normalization

The data is then normalized using min-max normalization which uses the minimum and maximum values of the matrix.

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}$$

3.3 Data Partitioning

The whole dataset is partitioned into three parts, Training, Validation and Testing randomly in the ratio of 80%, 10% and 10% respectively.

4 Architecture

The outcome in binomial logistic regression can be a 0 or a 1. The idea is then to estimate the probability of an outcome being a 1 or a 0. Given that the probability of the outcome being a 1 is given by p then the probability of it not occurring is given by 1-p. This can be seen as a special case of Binomial distribution called the Bernoulli distribution.

The idea in logistic regression is to cast the problem in form of generalized linear regression model[.

$$Y^{\wedge} = \beta_0 + \beta_1 X_1 + \dots + \beta_n X_n$$

where y^{\wedge} =predicted value, x = independent variables and the β are coefficients to be learnt. This can be compactly expressed in vector form:

$$W^T = [\beta_0, \beta_1, \dots, \beta_n]$$

$$X^T = [1, x^1, \dots, x^n]$$

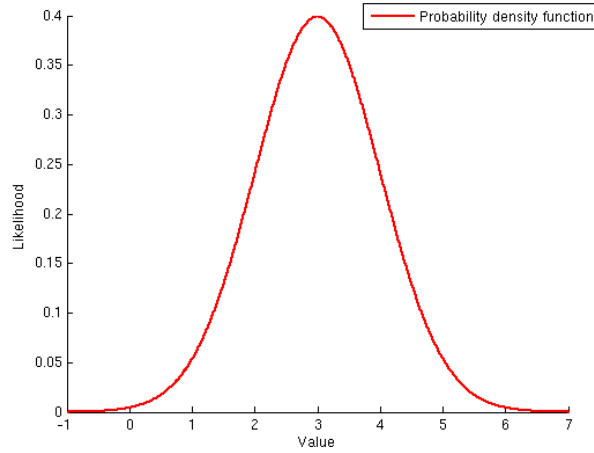
or

$$Y^{\wedge} = W^T X$$

Now taking the gaussian probability function

$$f(t) \triangleq \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(t-\mu)^2}{2\sigma^2}}$$

Figure 1: Gaussian Normal Distribution



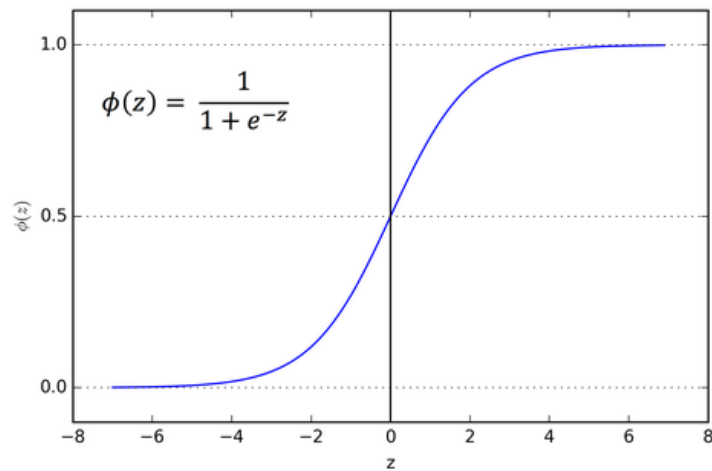
and applying on the bayes theorem on gaussian distribution

$$P(C_1|K) = \frac{P(X|C_1).P(C_1)}{P(X|C_1).P(C_1) + P(X|C_2).P(C_2)}$$

On simplification, the sigmoid activation function is formed

$$y = \frac{1}{1 + e^{-x}}.$$

Figure 2: Sigmoid activation function



The main reason why we use sigmoid function is because it exists between **(0 to 1)**. Therefore, it is especially used for models where we have to **predict the probability** as an output. Since probability of anything exists only between the range of **0 and 1**, sigmoid is the right choice.

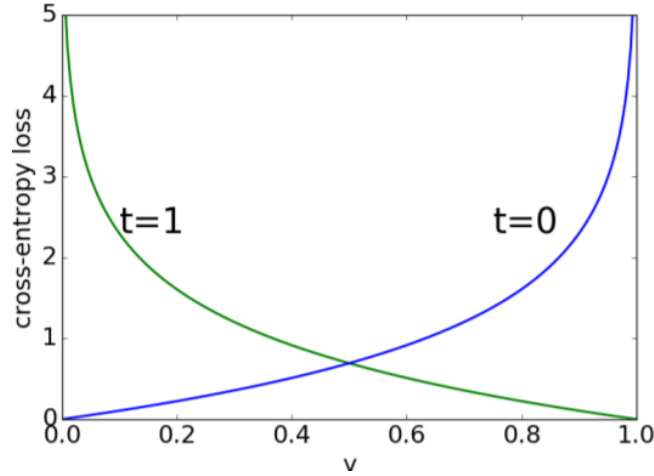
The function is **differentiable**, that means, we can find the slope of the sigmoid curve at any two points. Therefore, sigmoid activation is an apt choice for logistic regression.

The cross-entropy loss function is used to calculate the loss or cost of the prediction from the target values.

$$L(p,y)=-y\log(p)-(1-y)\log(1-p)$$

where p is the prediction and y is the target.

Figure 3: Cross Entropy Graph



From this graph we observe that the cross-entropy loss penalizes more for wrong predictions than right ones.

The last and the most important function is the gradient function which is to update weight

$$W_{new} = W_{old} + \alpha \frac{\partial L}{\partial W_n}$$

where α represents the learning rate.

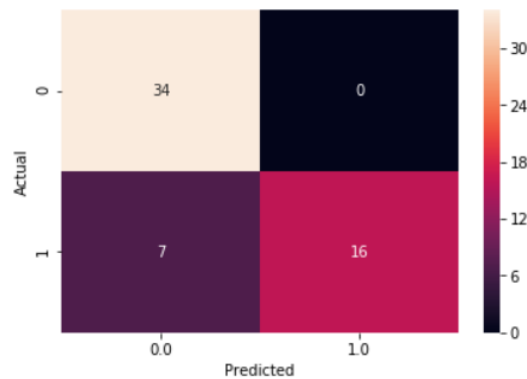
This updating of weights is done 'n' times where each loop is called an **epoch**. Thus the hyper-parameters here are the learning rate and number of epochs

5 Results

The model was trained using several hyper parameters of which some of the results are shown below. All the models were tested on the number of test data created while partitioning. The results were assessed on based on several metrics which are derived from the confusion matrix.

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions.

Figure 4: Sample Confusion matrix learning rate=0.8, number of epochs=50



Terms in confusion matrix:-

- **Positive (P)**: Observation is positive
- **Negative (N)**: Observation is not positive
- **True Positive (TP)**: Observation is positive, and is predicted to be positive.
- **False Negative (FN)**: Observation is positive, but is predicted negative.
- **True Negative (TN)**: Observation is negative, and is predicted to be negative.
- **False Positive (FP)**: Observation is negative, but is predicted positive.

The formulas for accuracy, precision, recall and F-1 score are given below

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

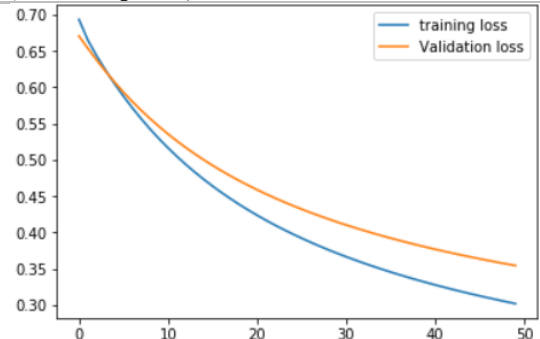
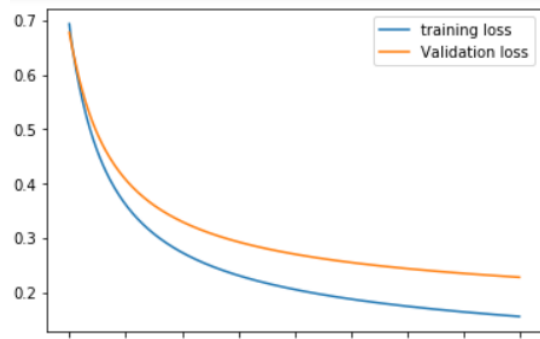
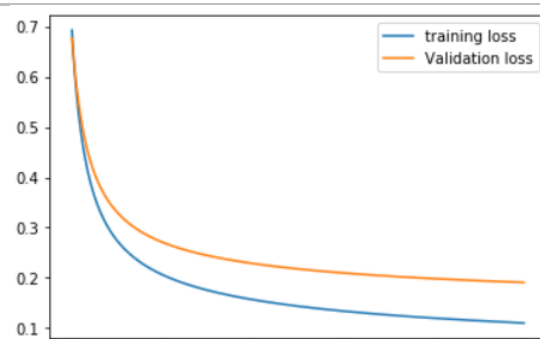
$$\text{Precision} = \frac{TP}{TP + FP}$$

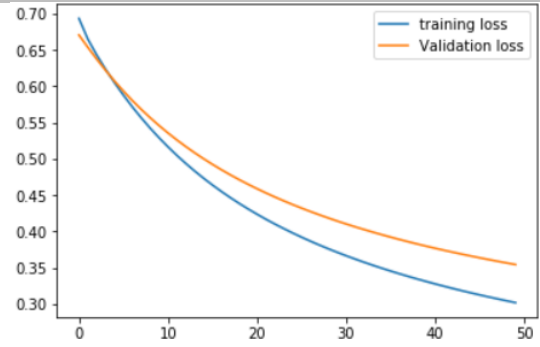
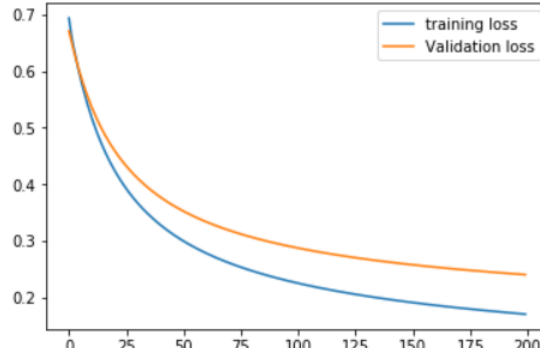
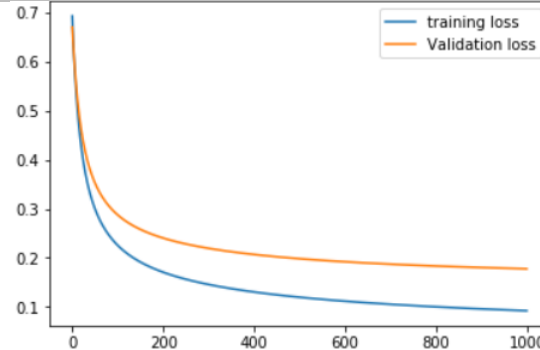
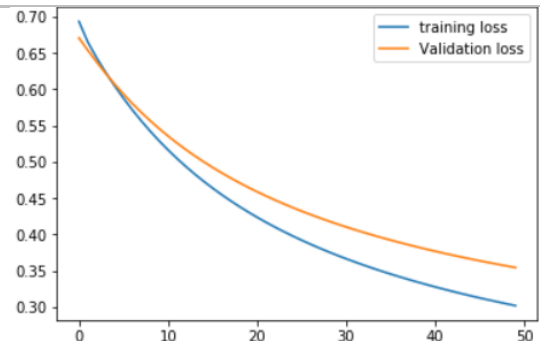
$$F - \text{measure} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

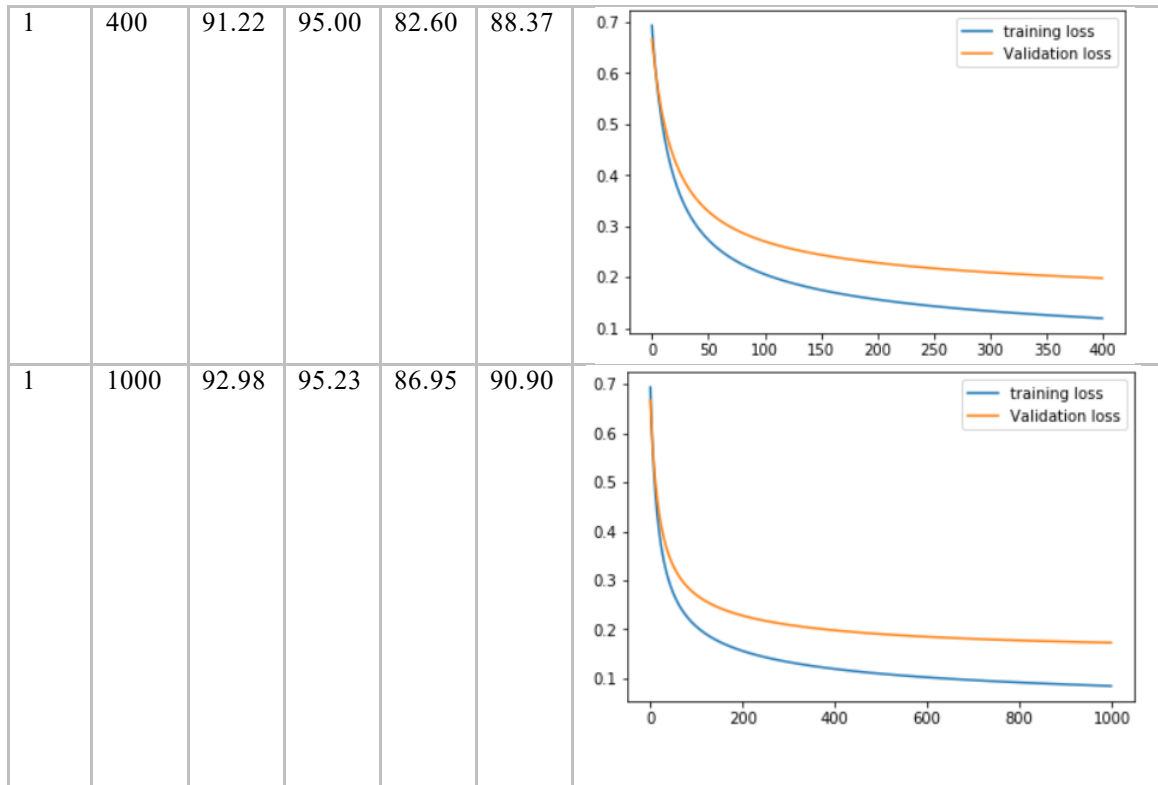
In the results table mentioned below, the symbols/alphabets represent the following parameters. The values are taken up to 2 decimal places hence might be same for some observations.

c→ Learning rate; **E**→ Number of Epochs; **A**→ Accuracy; **P**→ Precision; **R**→ Recall; **F1**→ F-1 Score

Table 2: Results Table

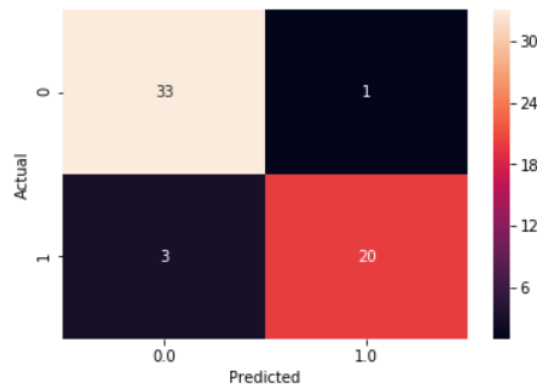
c	E	A	P	R	F-1	Train loss and Validation loss per Epoch (Loss vs epochs)
0.5	50	84.21	100	69.53	82.05	
0.5	400	92.98	100	82.60	90.47	
0.5	1000	92.98	95.23	86.95	90.90	

0.8	50	87.77	99.82	69.75	82.15	
0.8	200	92.98	100	82.60	90.47	
0.8	1000	92.98	95.23	86.95	90.90	
1	50	87.77	100	69.53	82.05	



It can be seen from the observations that a **learning rate of 0.8 with 1000 epochs is the best approach** which has a good accuracy and a good balance of precision and recall (which can be seen from the f-1 score). A larger learning rate could be considered but it could skip the global maxima. The number of epochs if increased would prove to be futile as the validation graph flattens out, therefore there would not be any increase in accuracy. If the number of epochs are decreased then it might lead to slight under fitting.

Figure 5: Confusion matrix for **learning rate=0.8, number of epochs=1000**



6 Conclusion

It can be concluded that a larger learning rate could skip the global maxima. The number of epochs if increased can prove to be futile as the validation graph flattens out, therefore there would not be any increase in accuracy. If the number of epochs are decreased then it might lead to slight under fitting. The model seems to produce accurate result with efficiency and will possibly produce accurate output on real time data. The future scope of this project can be to use adaptive methods to decide learning rates such as adagrad, rmsprop and adam optimizers.