Project 1: Logistic Regression

Snehashish Mandal
Department of Computer Science
University at Buffalo
Buffalo, NY 14214
snehashi@buffalo.edu

Abstract

The goal of this project is to perform classification (logistic regression) using machine learning. It is for a two-class problem. The features used for classification are pre-computed from images of a fine needle aspirate (FNA) of a breast mass. The goal of the project is to classify suspected FNA cells to Benign (class 0) or Malignant (class 1) using logistic regression as the classifier. Wisconsin Diagnostic Breast Cancer (WDBC) dataset is being used for training, validation and testing. 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. The training dataset is used to train the logistic regression model. The weights are calculated using Gradient Descent method by minimizing the Cross-Entropy loss function until it converges. The validation dataset is then used to tune the hyper-parameter, learning rate (η). Then we use the trained model to predict the output of the training dataset.

1 Introduction

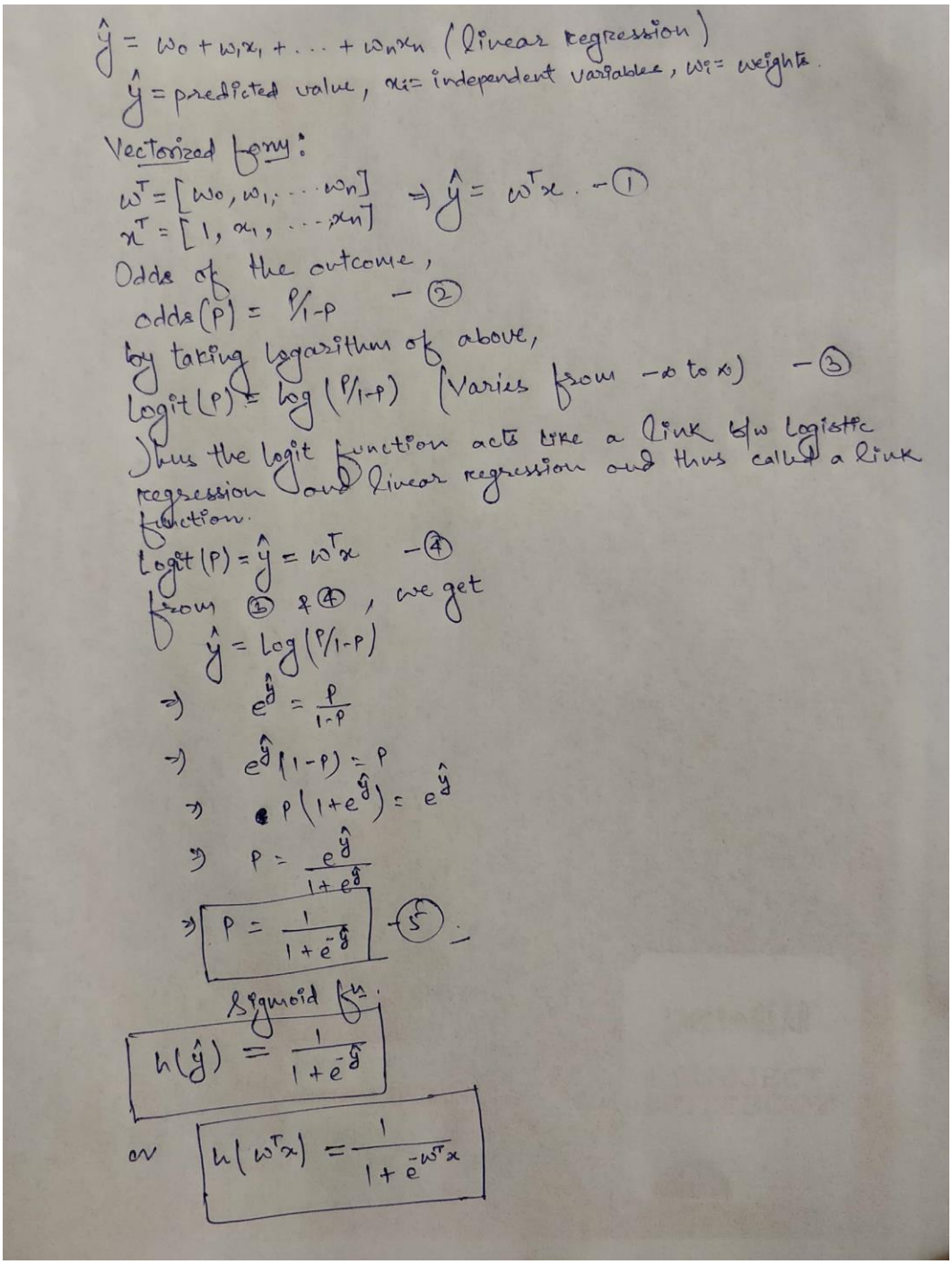
"A computer program is said to learn from experience E, with respect to some task T, and some performance measure P, if its performance on T, as measured by P, improves with experience E."

-Tom Mitchell

Machine learning is the field of study where we program the computer to learn from its past experiences and predict the output based upon those experiences. Broadly, machine learning is classified into two types, viz, supervised learning and unsupervised learning. In supervised learning, we are given a dataset and already know what our correct output should look like, having the idea that there is a relationship between the input and the output. Whereas, unsupervised learning allows us to approach problems with little or no idea what our results should look like. We derive structures from data where we do not necessarily know the effect of the variables. [1]
Supervised learning is further divided into two subtypes, viz, regression problem or linear regression and classification problem or logistic regression. In linear regression, we try to predict the result within a continuous output, whereas in logistic regression we try to predict output in a discrete output. [1] In this project, two class output logistic regression model is used to predict the result of a wisconsin diagnostic breast cancer dataset as malignant or benign. The wdbc dataset is divided into training, validation and testing set. The genesis equation used in logistic regression is the sigmoid function. The cross-entropy loss of the model is then minimized using gradient descent method.

1.1 Sigmoid Function

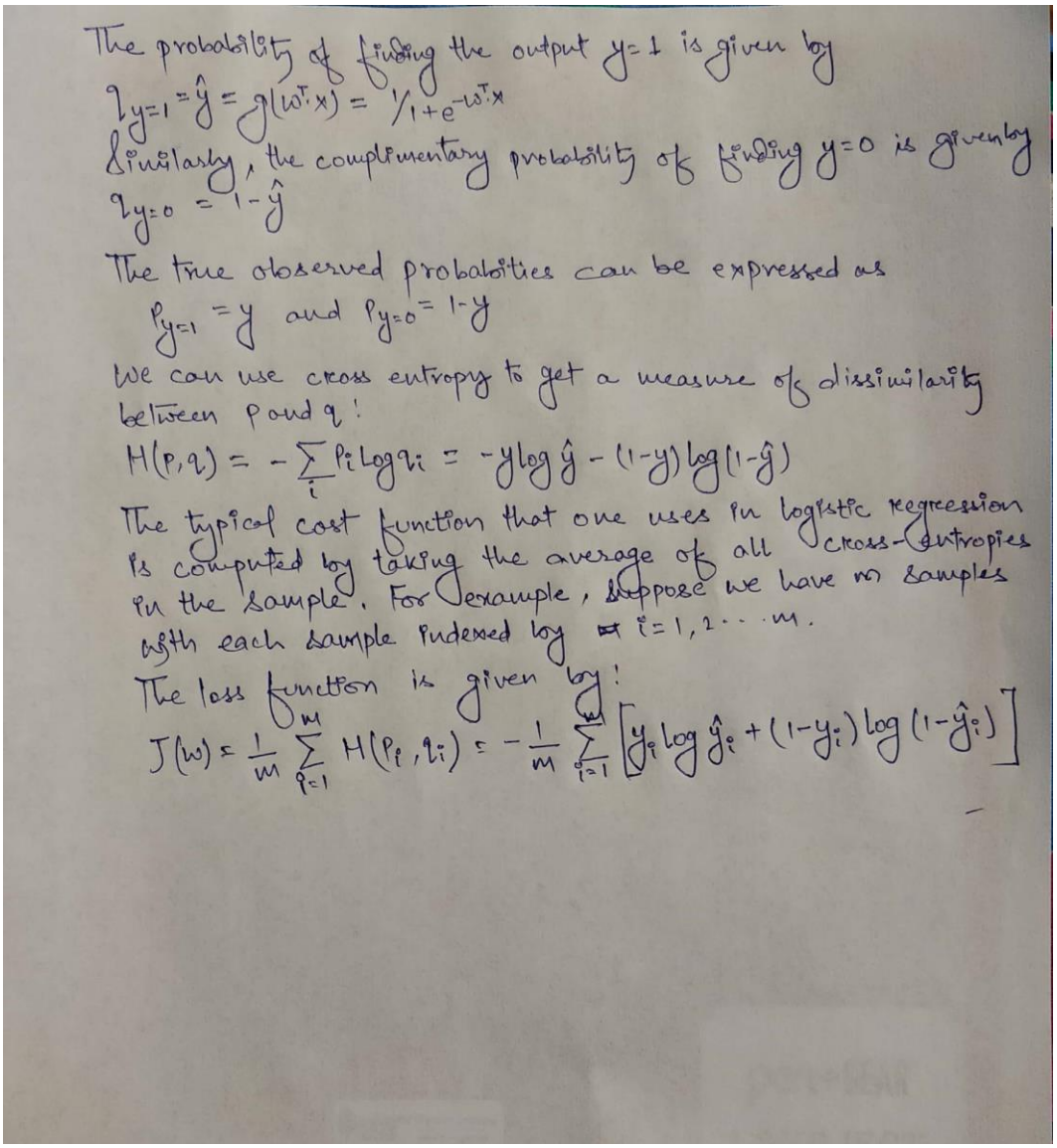
The outcome in binomial logistic regression can be a 0 or a 1. The idea is then to estimate the probability of an outcome being a 1 or a 0. Given that the probability of the outcome being a 1 is given by p then the probability of it not occurring is given by 1-p. This can be seen as a special case of Binomial distribution called the Bernoulli distribution. [2]



1.2 Cross Entropy Error Function

Cross entropy can be used to define a loss function in machine learning and optimization. The true probability p_i is the true label, and the given distribution q_i is the predicted value of

the current model. More specifically, consider logistic regression, which (in its most basic form) deals with classifying a given set of data points into two possible classes generically labelled 0 and 1. The logistic regression model thus predicts an output **Y** Ɛ {0, 1}, given an input vector **X**. The probability is modeled using the sigmoid function $g(z) = \frac{1}{1+e^{-z}}$. [3]



1.3 Gradient Descent

After getting the cross-entropy loss function, we need to find the weights of the logistic model such that the cross-entropy loss is minimum. The method used to minimize the loss and calculate the weight is called as gradient descent. In this method we keep minimizing the loss function and update the weight until it converges. The equation and proof of gradient descent is given below.

Gradient Descent

$$w_i = w_i - \eta \frac{\partial J(w)}{\partial w_i} \quad - (1)$$

$$J(w) = -\frac{1}{m} \sum_{i=1}^{m} \left[y_i \log \hat{y}_i + (1-y_i) \log (1-\hat{y}_i) \right] \quad - (2)$$

$$\frac{\partial J(w)}{\partial w_i} = -\frac{1}{m} \sum_{i=1}^{m} \left[y_i \cdot \frac{1}{\hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial w_i} + (1-y_i) \cdot \frac{1}{(1-\hat{y}_i)} \cdot (-1) \frac{\partial \hat{y}_i}{\partial w_i} \right]$$

$$= -\frac{1}{m} \sum_{i=1}^{m} \frac{\partial \hat{y}_i}{\partial w_i} \left[\frac{y_i}{\hat{y}_i} - \frac{1-y_i}{(1-\hat{y}_i)} \right] \quad - (3)$$

$$\hat{y}_i = \frac{1}{1+e^{-w_i x}} \quad - (4)$$

$$\frac{\partial \hat{y}_i}{\partial w_i} = \frac{\partial \left(1+e^{-w_i x}\right)^{-1}}{\partial w_i}$$

$$= -\left(1+e^{-w_i x}\right)^{-2} \cdot e^{-w_i x} \cdot (-x)$$

$$= \frac{e^{-w_i x}}{1+e^{-w_i x}} \cdot \frac{1}{1+e^{-w_i x}} \cdot x$$

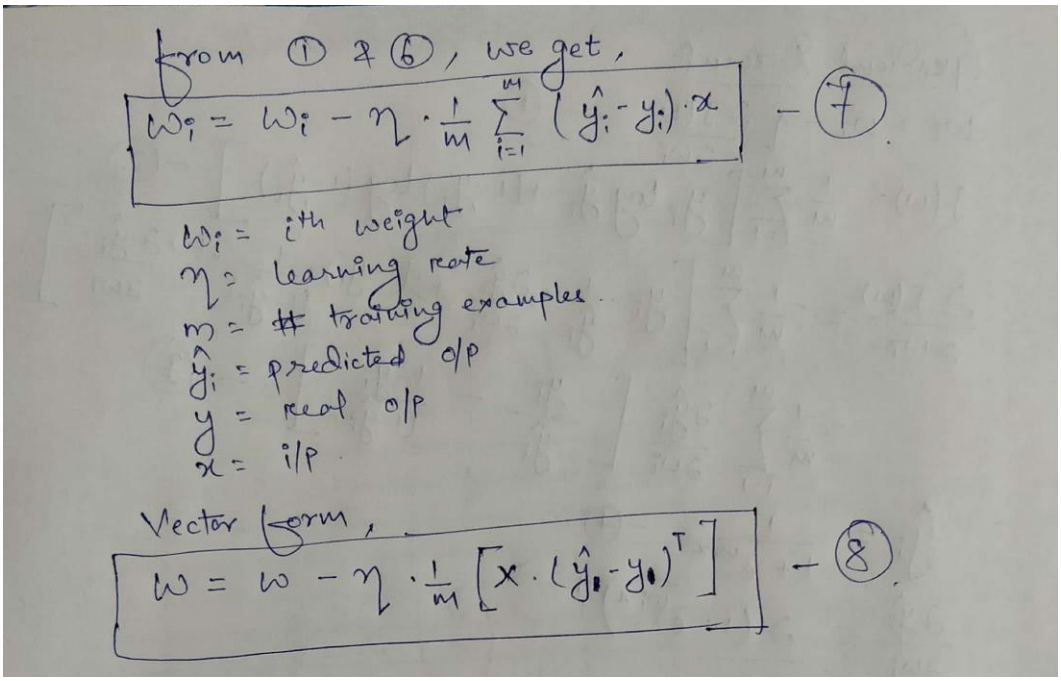
$$= (1-\hat{y}_i) \cdot \hat{y}_i \cdot x \quad - (5)$$

$$\frac{\partial J(w)}{\partial w_i} = -\frac{1}{m} \sum_{i=1}^{m} \frac{\partial \hat{y}_i}{\partial w_i} \left[\frac{y_i - y_i \hat{y}_i - \hat{y}_i + y_i \hat{y}_i}{\hat{y}_i (1-\hat{y}_i)} \right]$$

From (5).

$$= -\frac{1}{m} \sum_{i=1}^{m} (1-\hat{y}_i) \cdot \hat{y}_i \cdot x \left[\frac{y_i - \hat{y}_i}{\hat{y}_i (1-\hat{y}_i)} \right]$$

$$\boxed{\frac{\partial J(w)}{\partial w_i} = -\frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i) \cdot x} \quad - (6)$$



2 Dataset

Wisconsin Diagnostic Breast Cancer (WDBC) dataset is being used for training (80%), validation (10%) and testing (10%). 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 _ne needle aspirate (FNA) of a breast mass. Computed features describe the following characteristics of the cell nuclei present in the image:

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 (perimeter^2/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. [4]

3 Preprocessing

The preprocessing step includes data partitioning, training of the model using gradient descent, tuning the hyper-parameters and finally testing the model using the test dataset.

3.1 Data Splitting

Wisconsin Diagnostic Breast Cancer (WDBC) dataset is divided into training set (80%),

validation set (10%) and testing set (10%) randomly. While splitting, the ID column has been dropped and M & B values are mapped to 1 & 0 respectively. The 30 features are loaded to X, and the output, B (0) or M (1) are loaded to Y.

3.2 Tuning the hyper parameters

Hit and trial method has been used to tune the hyper parameter learning rate (η). Initial values of η were taken as [.00001, .001, .003, .01, .03, .1, .3, 1, 3, 10, 100, 1000]. Using the validation dataset loss is calculated for each of the learning rates and graphs between loss and epochs were plotted for each learning rates. Plots having learning rates very small have very slight slope and plot with learning rates very high have steep slope and sometimes the cost increases as well. By looking at the graph, an appropriate learning rate has been selected, whose graph is properly decreasing during the whole epoch. In this case, η = 0.1 has been selected to train the model.

3.3 Testing the model

After the logistic model is trained using the training set and validated using the validation set, the performance of the model is tested by passing the testing test and calculating the Accuracy, Precision and Recall with respect to the original output available.

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

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

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

Where, TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

For this model, the accuracy is 98.25%, precision is 95.24% and recall is 100.00%

4 Results

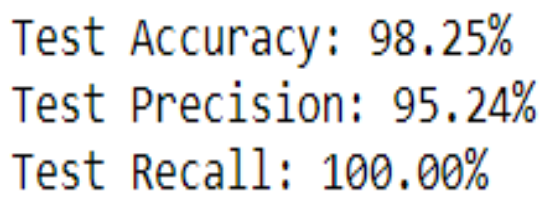


Figure 1: Results after testing the model with testing dataset

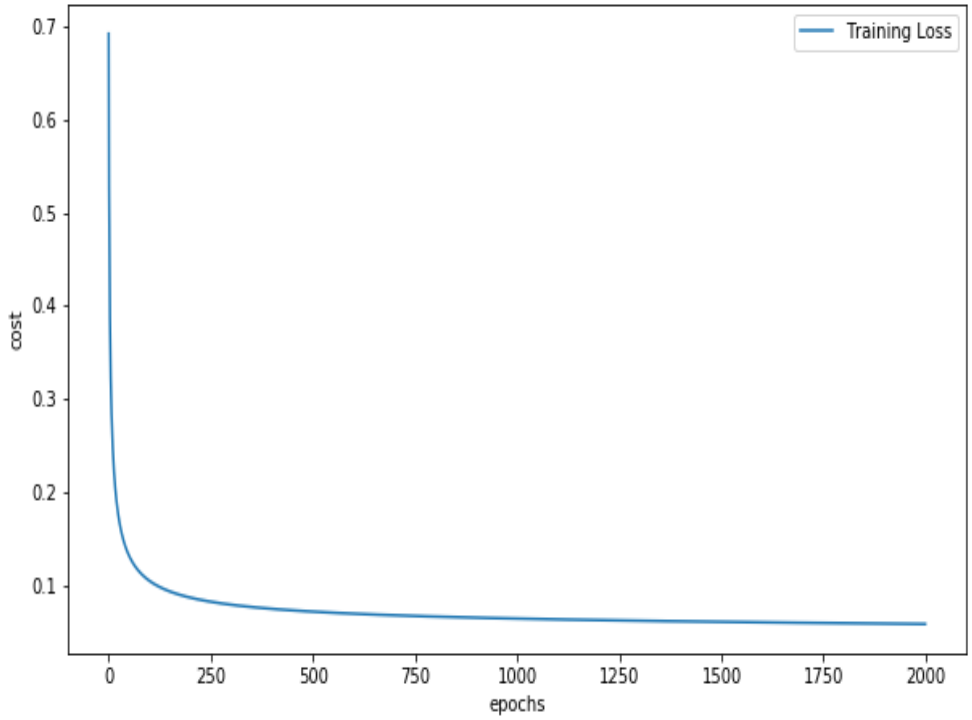


Figure 2: Training loss vs Epoch

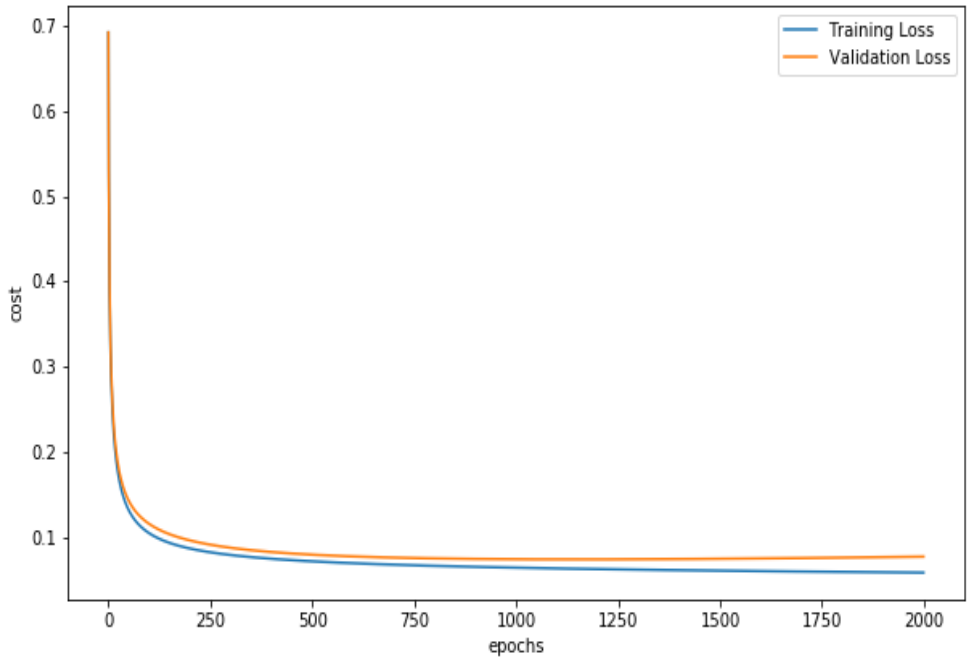


Figure 3: Training loss vs Validation Loss

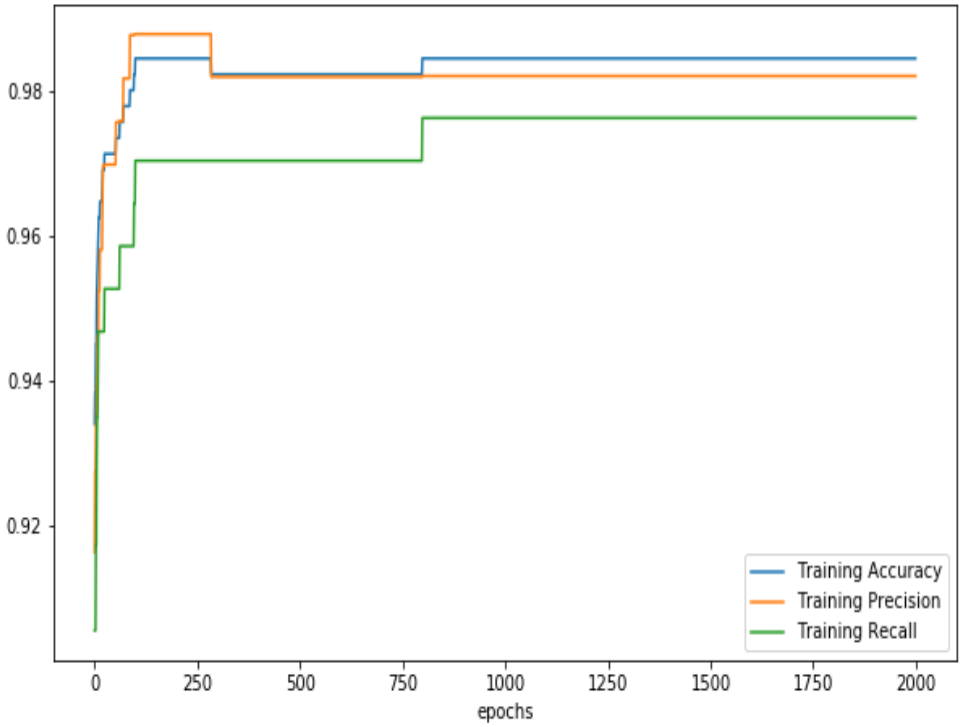


Figure 4: Training Accuracy, precision, recall vs epoch

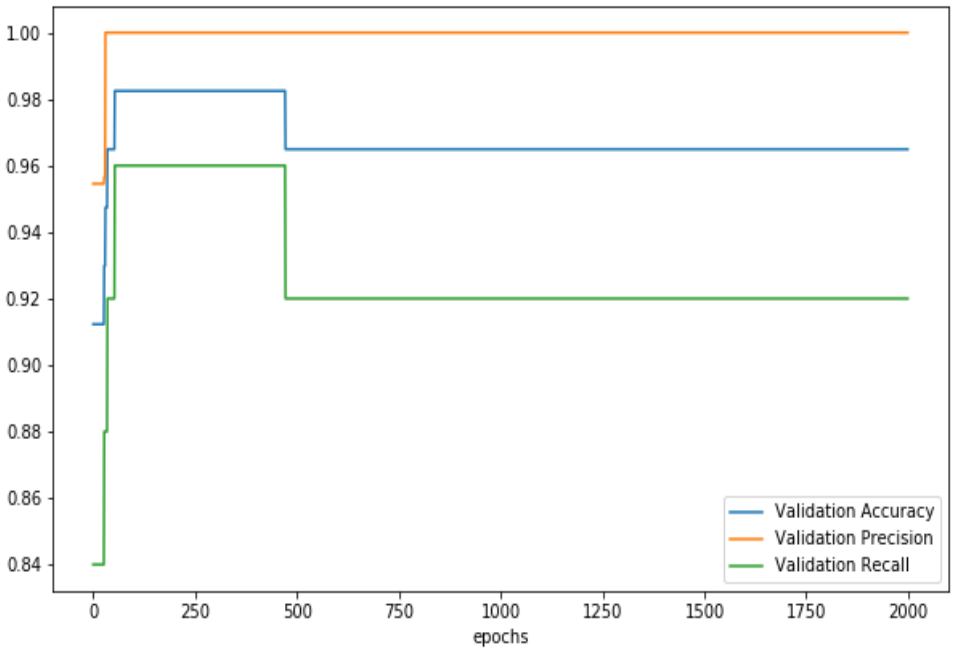


Figure 5: Validation Accuracy, precision, recall vs epoch

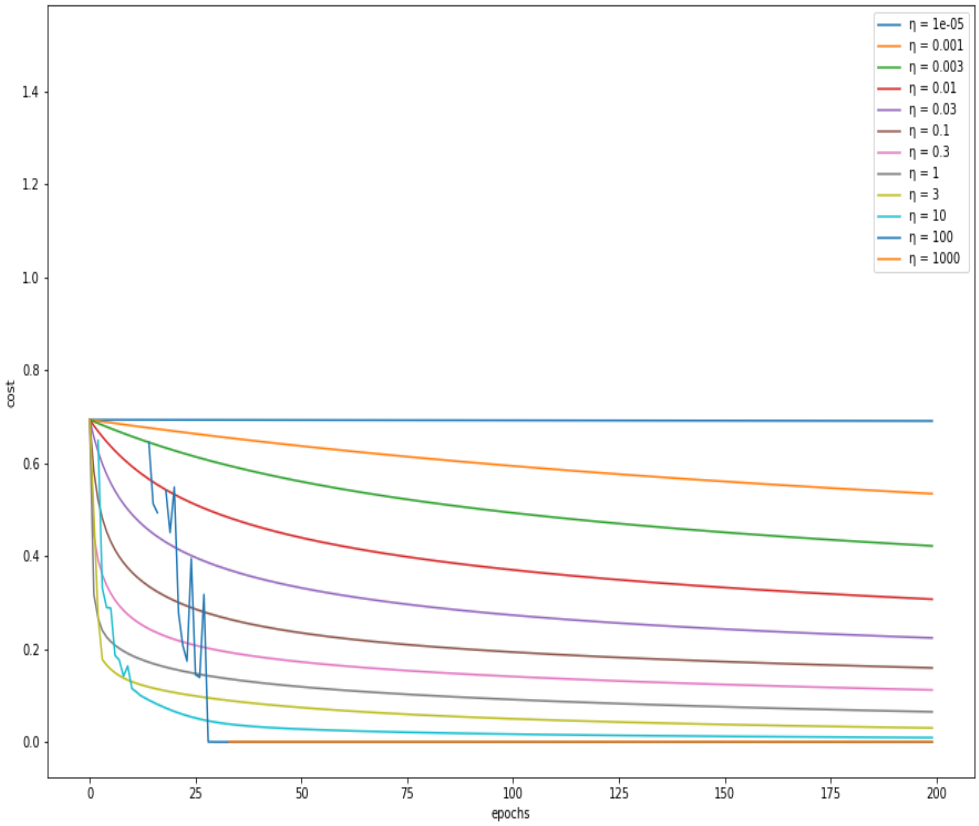


Figure 6: cost vs epochs for different η

5 Conclusion

By looking at the accuracy (98.25%), precision (95.24%) and recall (100%), it can be said that the logistic model is trained properly as per our requirement and is giving expected outputs.

References

[1] Ng, Andrew Machine Learning Course. https://www.coursera.org/learn/machine-learning

[2] Madsen, Henrik; Thyregod, Poul (2011). *Introduction to General and Generalized Linear Models*. Chapman & Hall/CRCC. ISBN 978-1-4200-9155-7.

[3] Murphy, Kevin (2012). Machine Learning: *A Probabilistic Perspective.* MIT. ISBN 978-0262018029.

[4] Project description PDF.