CSE 574: INTRODUCTION TO MACHINE LEARNING
(FALL 2019)

Prof. S. N. Srihari
University at Buffalo

PROJECT REPORT

Submitted by –

Nitesh Shantha Kumar (UB id: 50321964)

Project 1: Logistic Regression

1. Abstract

Logistic Regression expands on linear regression as a classification algorithm which has only two possible outcomes, which are known as classes. It is a classification model that uses input variables to predict a categorical outcome variable that can take on one of a limited set of class values. Logistic regression uses the sigmoid function to weighted input values to generate a prediction of the data class. Logistic regression is used when the output has to be divided into 2 classes and the decision boundary can be used to split the data into binomial responses easily

2. Dataset

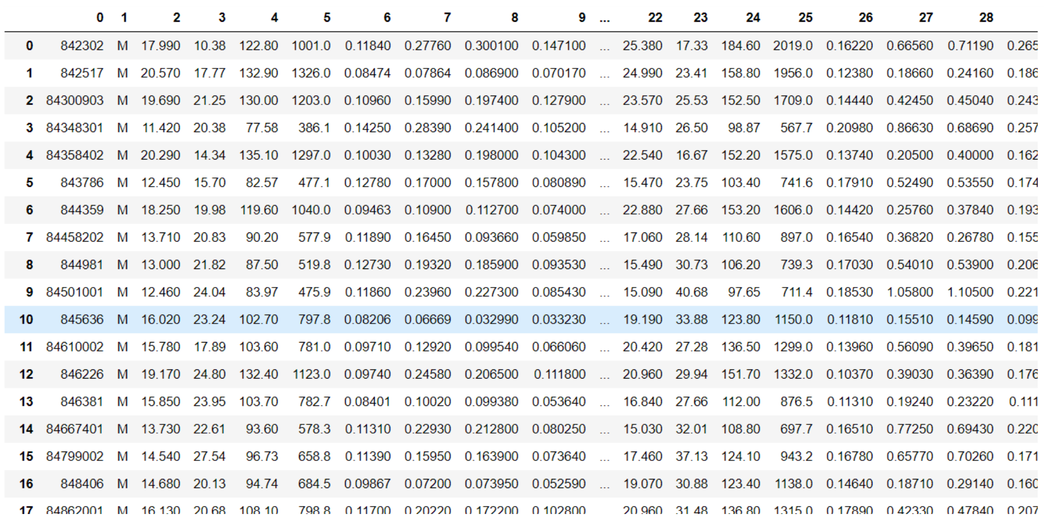
We are given a problem to predict whether a tumor is malignant or benign based on the dataset provided by WDBC. The Dataset consists of 569 rows of data specific to the characteristics of tumor like its radius, area, smoothness, etc., each having 32 attributes. Out of 32 features, 31 are independent variable and the column (say, result) with values ∈ {'M', 'B'}, is a dependent variable, whose value is predicted by the rest 30 features. 1 feature i.e. ID is redundant as it does not affect the value of result.

3. Preprocessing

Logistic Regression is implemented using following processes:

3.1. Input dataset is read through pandas:

The 'pandas' library is being used to read the dataset file into python code. The header is set as none, since the dataset has not been provided with headers. Following is the extracted dataset using pandas:



	0	1	2	3	4	5	6	7	8	9	...	22	23	24	25	26	27	28	
0	842302	M	17.990	10.38	122.80	1001.0	0.11840	0.27760	0.300100	0.147100	...	25.380	17.33	184.60	2019.0	0.16220	0.66560	0.71190	0.265
1	842517	M	20.570	17.77	132.90	1326.0	0.08474	0.07864	0.086900	0.070170	...	24.990	23.41	158.80	1956.0	0.12380	0.18660	0.24160	0.186
2	84300903	M	19.690	21.25	130.00	1203.0	0.10960	0.15990	0.197400	0.127900	...	23.570	25.53	152.50	1709.0	0.14440	0.42450	0.45040	0.243
3	84348301	M	11.420	20.38	77.58	386.1	0.14250	0.28390	0.241400	0.105200	...	14.910	26.50	98.87	567.7	0.20980	0.86630	0.68690	0.257
4	84358402	M	20.290	14.34	135.10	1297.0	0.10030	0.13280	0.198000	0.104300	...	22.540	16.67	152.20	1575.0	0.13740	0.20500	0.40000	0.162
5	843786	M	12.450	15.70	82.57	477.1	0.12780	0.17000	0.157800	0.080890	...	15.470	23.75	103.40	741.6	0.17910	0.52490	0.53550	0.174
6	844359	M	18.250	19.98	119.60	1040.0	0.09463	0.10900	0.112700	0.074000	...	22.880	27.66	153.20	1606.0	0.14420	0.25760	0.37840	0.193
7	84458202	M	13.710	20.83	90.20	577.9	0.11890	0.16450	0.093660	0.059850	...	17.060	28.14	110.60	897.0	0.16540	0.36820	0.26780	0.155
8	844981	M	13.000	21.82	87.50	519.8	0.12730	0.19320	0.185900	0.093530	...	15.490	30.73	106.20	739.3	0.17030	0.54010	0.53900	0.206
9	84501001	M	12.460	24.04	83.97	475.9	0.11860	0.23960	0.227300	0.085430	...	15.090	40.68	97.65	711.4	0.18530	1.05800	1.10500	0.221
10	845636	M	16.020	23.24	102.70	797.8	0.08206	0.06669	0.032990	0.033230	...	19.190	33.88	123.80	1150.0	0.11810	0.15510	0.14590	0.099
11	84610002	M	15.780	17.89	103.60	781.0	0.09710	0.12920	0.099540	0.066060	...	20.420	27.28	136.50	1299.0	0.13960	0.56090	0.39650	0.181
12	846226	M	19.170	24.80	132.40	1123.0	0.09740	0.24580	0.206500	0.111800	...	20.960	29.94	151.70	1332.0	0.10370	0.39030	0.36390	0.176
13	846381	M	15.850	23.95	103.70	782.7	0.08401	0.10020	0.099380	0.053640	...	16.840	27.66	112.00	876.5	0.11310	0.19240	0.23220	0.111
14	84667401	M	13.730	22.61	93.60	578.3	0.11310	0.22930	0.212800	0.080250	...	15.030	32.01	108.80	697.7	0.16510	0.77250	0.69430	0.220
15	84799002	M	14.540	27.54	96.73	658.8	0.11390	0.15950	0.163900	0.073640	...	17.460	37.13	124.10	943.2	0.16780	0.65770	0.70260	0.171
16	848406	M	14.680	20.13	94.74	684.5	0.09867	0.07200	0.073950	0.052590	...	19.070	30.88	123.40	1138.0	0.14640	0.18710	0.29140	0.160
17	84862001	M	16.130	20.68	108.10	798.8	0.11700	0.20220	0.172200	0.102800		20.960	31.48	136.80	1315.0	0.17890	0.42330	0.47840	0.207

3.2. Mapping Column 1:

The dataset has output tokens as {'M', 'B'}, where M indicates class 'Malignant' and B indicates class 'Benign' Cancer. It is important to change this indicator as the whole dataset would be comprised of float values.

3.3. Dataset classification:

The 1st column is considered redundant and is not used in Input nor Output. Data is classified into Training data (80% of the main dataset), Validation data (10% of the main dataset) and Test data (remaining 10%) using manual classification using numpy arrays. Here, the training data is used to train the logistic regression model. Weighed values are used in validation data to obtain optimum values of hyperparameters and test data is used to obtain the result of the trained model on real set of data.

3.4. Normalization:

30 features in the dataset representing a different characteristic of tumor has a chance that one of the features may have a higher range e.g. between 500-5000 and some may have very low range e.g. between 0.001-1. This ill-affects the regression as the low-range values may become redundant. To avoid redundancy, normalization of each data value is done.

3.5. Initializing w and b:

w is assigned with an array of random values and b is assigned with 0.

4. Algorithm

4.1. Gradient Descent:

For logistic regression, the hypothesis is sigmoid of 'w^TX+b'. We have to find values of w and b such that it minimizes the cost function. To overcome the issue of 'w^TX+b' ranging from 10 to 10000, Sigmoid function is used to create a threshold which basically gives the probability that w^TX+b is very big or small. Where, the condition will be either 0 or 1.

Gradient

$$\boxed{z = w_1x_1 + w_2x_2 + b} \longrightarrow \boxed{\hat{y} = a = \sigma(z)} \longrightarrow \boxed{L(\hat{y}, y)}$$

$$\Leftrightarrow \boxed{a = \hat{y}.}$$

$$\boxed{w_1} \Rightarrow \frac{\partial(L)}{\partial w_1} = \frac{\partial L}{\partial a} \cdot \frac{\partial a}{\partial z} \cdot \frac{\partial(z)}{\partial w_1}$$

$$\frac{\partial L}{\partial a} = \frac{\partial}{\partial a}(-y \log a - (1-y)\log(1-a))$$

$$= -y\left(\frac{1}{a}\right) - (-1)\frac{(1-y)}{(1-a)}$$

$$\boxed{\frac{\partial L}{\partial a} = \left(\frac{-y}{a}\right) + \left(\frac{1-y}{1-a}\right)}$$

$$\boxed{\frac{\partial a}{\partial z} = a(1-a)}$$

$$\boxed{\frac{\partial z}{\partial w_1} = x_1}$$

$$\frac{\partial L}{\partial w_1} = \left(\left(\frac{-y}{a} + \frac{(1-y)}{1-a}\right) \cdot (a)(1-a)\right) \cdot x_1$$

$$= (a-y) \cdot x_1$$

Update for w_1,

$$\frac{\partial L}{\partial w_1} = (a-y) \cdot x_1$$

Here, $(a-y) = \frac{\partial L}{\partial z}$

$$\boxed{w_1 = w_1 - \alpha \frac{\partial L}{\partial w_1}}$$

Similarly, for all paramelers

$$\boxed{w_i = w_i - \alpha \frac{\partial L}{\partial w_i}}$$ $i = 1, 2, \dots, m$

m = no of paramaters

$$\boxed{b = b - \alpha \frac{\partial L}{\partial b}}$$

Where, $\frac{\partial L}{\partial b} = (a-y)$

The process of minimizing the gradient descent has 2 process:

4.2. Forward Propagation: where the above mentioned $w^T X+b$ is proceeded by sigmoid to calculate the input for the log values in cross entropy loss used to calculate cost.

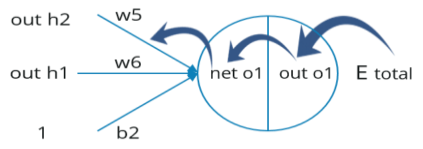
$$Cost(h_\Theta(x), y) = -y \log(h_\Theta(x)) - (1-y) \log (1- h_\Theta(x))$$

If y = 1, (1-y) term will become zero, therefore $-\log (h_\Theta(x))$ alone will be present

If y = 0, (y) term will become zero, therefore $-\log (1- h_\Theta(x))$ alone will be present

4.3. Back Propagation: Weights are approximately updated for efficient minimization of loss function. w and b are updated with its respective values for the next iteration which again carry forwards the same process to minimize the loss.

$$\frac{\delta Etotal}{\delta w5} = \frac{\delta Etotal}{\delta out\ o1} * \frac{\delta out\ o1}{\delta net\ o1} * \frac{\delta net\ o1}{\delta w5}$$



4.4. Predict function:

This function is basically used to predict the values of the vector by assigning 1 if that particular value is greater than 0.5 otherwise 0.

4.5. Calling the Logistical model:

The model is a combination of Gradient Descent and Predict function. After prediction vectors are obtained from the model, confusion matrix is used to calculate Accuracy, Precision and Recall.

5. Results

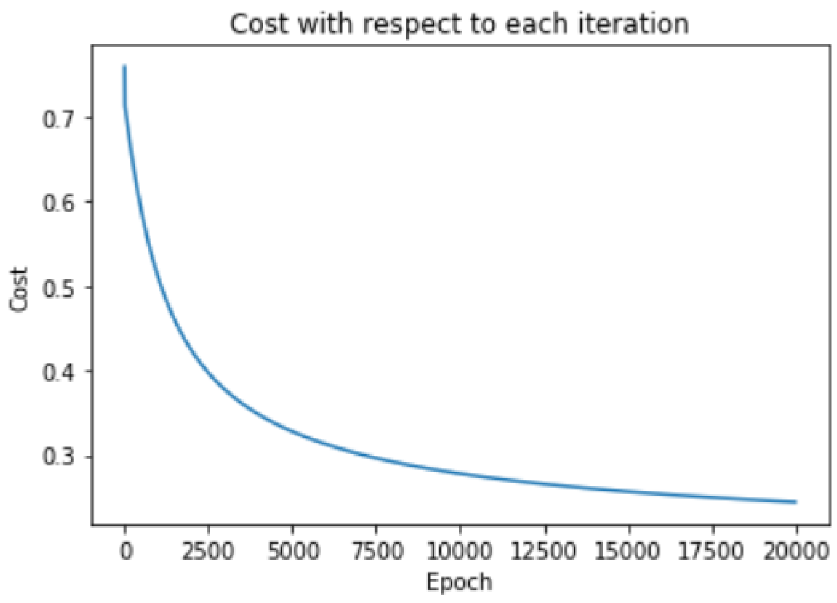
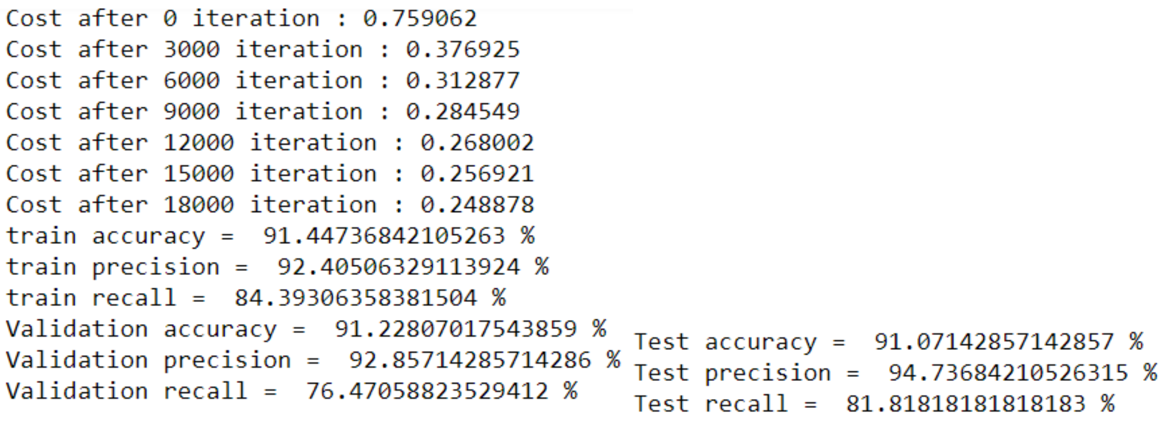


Figure 1 Iteration number: 20000, learning rate: 0.5

In Figure 1,

The cost function value for each epoch while training with 0.5 learning rate is mapped. We can observe that; the cost function gradually reduces to a minimum value.



```
Cost after 0 iteration : 0.759062
Cost after 3000 iteration : 0.376925
Cost after 6000 iteration : 0.312877
Cost after 9000 iteration : 0.284549
Cost after 12000 iteration : 0.268002
Cost after 15000 iteration : 0.256921
Cost after 18000 iteration : 0.248878
train accuracy =  91.44736842105263 %
train precision =  92.40506329113924 %
train recall =  84.39306358381504 %
Validation accuracy =  91.22807017543859 %
Validation precision =  92.85714285714286 %
Validation recall =  76.47058823529412 %
```

```
Test accuracy =  91.07142857142857 %
Test precision =  94.73684210526315 %
Test recall =  81.81818181818183 %
```

Figure 2 Cost and output of Accuracy, Precision and Recall of Train, Validation and Test data

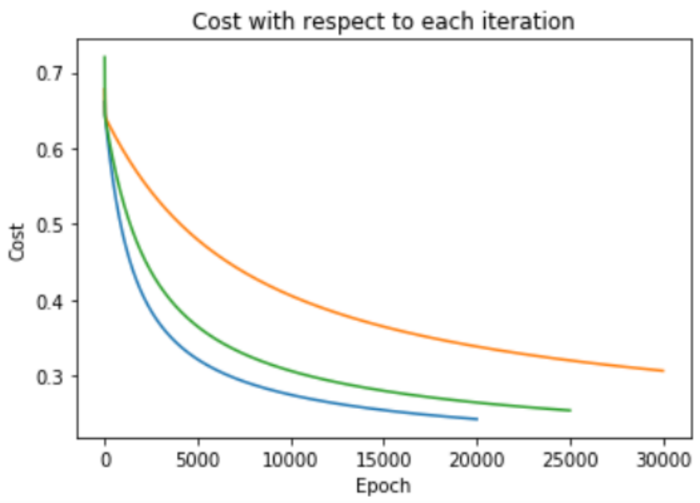


Figure 2 cost v/s epochs

In Figure 2,

The cost function with respect to multiple learning rate is mapped. Learning rates used are 0.3 for 25000 iterations, 0.1 for 30000 and 0.5 for 20000 iterations. We can observe that; the cost function gradually reduces to a minimum value.

6. Conclusion

Thus, Logistic Regression is implemented by having training data train with an accuracy of 91%, modified the hyperparameters to achieve 91% is validation data and predicted the output for test data with an accuracy of 91%.

7. References:

- https://en.wikipedia.org/wiki/Logistic_regression
- https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc