Breast Cancer Prediction using Logistic Regression

Bhakti Jadhav
Department of Computer Science
University at Buffalo
Buffalo NY, 14214
bhaktiha@buffalo.edu

Abstract

Machine learning has developed classification models that can be used to predict outcomes in individual cancer patients. The project performs classification for two class problem and categorize tumors into malignant or benign using features from digitized image of a fine needle aspirate (FNA) of a breast mass. For the implementation of the ML algorithms, the dataset was partitioned in the following way: 80% for training phase, 10% for the testing phase and 10% for validation phase. The hyper-parameters used for all the classifiers were manually assigned. Outcome show that Logistic Regression performed on data provided, performed well on classification task with a test accuracy of ≈90%.

1 Introduction

Breast cancer is one of most common cancer and is a topic of research with great value. The implementation of machine learning approaches in medical fields proves to be productive as such approaches may be considered of great assistance in the decision-making process of medical practitioners.
Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam, Online transactions Fraud or not Fraud. Logistic regression is a predictive analysis algorithm and transforms its output using the logistic sigmoid function to return a probability value.
Using Logistic Regression we need to classify suspected FNA cells to Benign (class 0) or Malignant (class 1).

Types of Logistic Regression

1. Binary (e.g. Tumor Malignant or Benign)
2. Multi-linear functions failsClass (e.g. Cats, dogs or Sheep's)

Logistic Regression uses a complex cost function defined as the '**Sigmoid function**' or also known as the 'logistic function'. The hypothesis of logistic regression tends it to limit the cost function between 0 and 1.

$$0 \leq h_\theta(x) \leq 1$$

Logistic Regression Hypothesis Expectation

2 Dataset

Wisconsin Diagnostic Breast Cancer (WDBC) dataset will be used for training, validation and testing. Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. The dataset features are as follows: (1) radius, (2) texture, (3) perimeter, (4) area, (5) smoothness, (6) compactness, (7) concavity, (8) concave points, (9) symmetry, and (10) fractal dimension. With each feature having three information:

1. mean,
2. standard error, and
3. "worst" or largest (mean of the three largest values) computed.

Thus, having a total of 30 dataset features.

- Datasets are linearly separable using all 30 input features
- Number of Instances: 569
- Class Distribution: 357 Benign, 212 Malignant

3 PreProcessing

We standardized the dataset before implementation of Logistic Regression in following ways:

1. Visualization of data is an imperative aspect of Machine Learning; it helps us understand the data. Libraries like pandas, Matplotlib helps us to get information about dataset.

   We identified that out of 569 cases, 357 are labeled as B(benign) and 212 as M(malignant).

```
1
B    357
M    212
dtype: int64
```

2. Process the original CSV dataset into a Pandas dataframe and extract features values and images from the data. Enumerate the diagnosis column such that M=1 and B=0. Also, set the ID column as index of the dataset as ID column will not be used for logistic regression.

dataset - DataFrame

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13
182	873701	1	15.7	20.31	101.2	766.6	0.09597	0.08799	0.06593	0.05189	0.1618	0.05549	0.3699	1.15
565	926682	1	20.13	28.25	131.2	1261	0.0978	0.1034	0.144	0.09791	0.1752	0.05533	0.7655	2.463
127	866203	1	19	18.91	123.4	1138	0.08217	0.08028	0.09271	0.05627	0.1946	0.05044	0.6896	1.342
474	911391	0	10.88	15.62	70.41	358.9	0.1007	0.1069	0.05115	0.01571	0.1861	0.06837	0.1482	0.538
449	911157302	1	21.1	20.52	138.1	1384	0.09684	0.1175	0.1572	0.1155	0.1554	0.05661	0.6643	1.361
212	8810703	1	28.11	18.47	188.5	2499	0.1142	0.1516	0.3201	0.1595	0.1648	0.05525	2.873	1.476
437	909220	0	14.04	15.98	89.78	611.2	0.08458	0.05895	0.03534	0.02944	0.1714	0.05898	0.3892	1.046
520	917092	0	9.295	13.9	59.96	257.8	0.1371	0.1225	0.03332	0.02421	0.2197	0.07696	0.3538	1.13
50	857343	0	11.76	21.6	74.72	427.9	0.08637	0.04966	0.01657	0.01115	0.1495	0.05888	0.4062	1.21
12	846226	1	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	0.2397	0.078	0.9555	3.568
193	875263	1	12.34	26.86	81.15	477.4	0.1034	0.1353	0.1085	0.04562	0.1943	0.06937	0.4053	1.809
144	869254	0	10.75	14.97	68.26	355.3	0.07793	0.05139	0.02251	0.007875	0.1399	0.05688	0.2525	1.239
113	864292	0	10.51	20.19	68.64	334.2	0.1122	0.1303	0.06476	0.03068	0.1922	0.07782	0.3336	1.86
309	893548	0	13.05	13.84	82.71	530.6	0.08352	0.03735	0.004559	0.008829	0.1453	0.05518	0.3975	0.8285
88	861597	0	12.36	21.8	79.78	466.1	0.08772	0.09445	0.06015	0.03745	0.193	0.06404	0.2978	1.502
340	89813	0	14.42	16.54	94.15	641.2	0.09751	0.1139	0.08007	0.04223	0.1912	0.06412	0.3491	0.7706
292	891670	0	12.95	16.02	83.14	513.7	0.1005	0.07943	0.06155	0.0337	0.173	0.0647	0.2094	0.7636
19	8510426	0	13.54	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.04781	0.1885	0.05766	0.2699	0.7886
353	9010018	1	15.08	25.74	98	716.6	0.1024	0.09769	0.1235	0.06553	0.1647	0.06464	0.6534	1.506
293	891703	0	11.85	17.46	75.54	432.7	0.08372	0.05642	0.02688	0.0228	0.1875	0.05715	0.207	1.238
163	8712064	0	12.34	22.22	79.85	464.5	0.1012	0.1015	0.0537	0.02822	0.1551	0.06761	0.2949	1.656
71	859711	0	8.888	14.64	58.79	244	0.09783	0.1531	0.08606	0.02872	0.1902	0.0898	0.5262	0.8522
566	926954	1	16.6	28.08	108.3	858.1	0.08455	0.1023	0.09251	0.05302	0.159	0.05648	0.4564	1.075
413	905557	0	14.99	22.11	97.53	693.7	0.08515	0.1025	0.06859	0.03876	0.1944	0.05913	0.3186	1.336

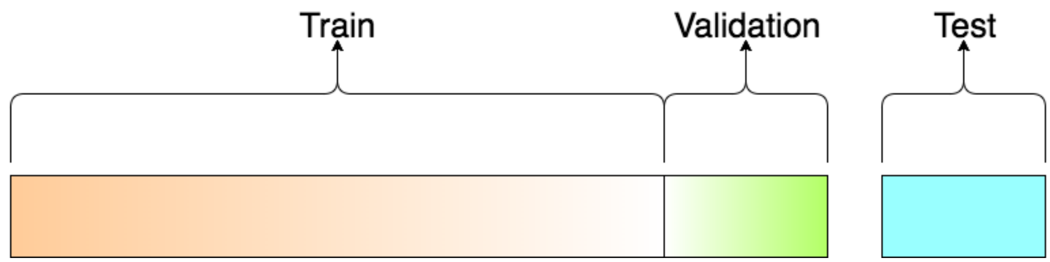
3. Normalization to change the values of columns in dataset so that the data is in common scale without losing integrity.

4. Slice the data in training, testing and validation sets.

   - Training Dataset (80%): The data used to train the model.

   - Testing Dataset(10%) : The sample of data used to provide evaluation of final model fit on training dataset.

   - Validation Dataset(10%) : This dataset is used while tuning hyperparameters and update the hyperparameters accordingly to get best fit model.



4 Architecture

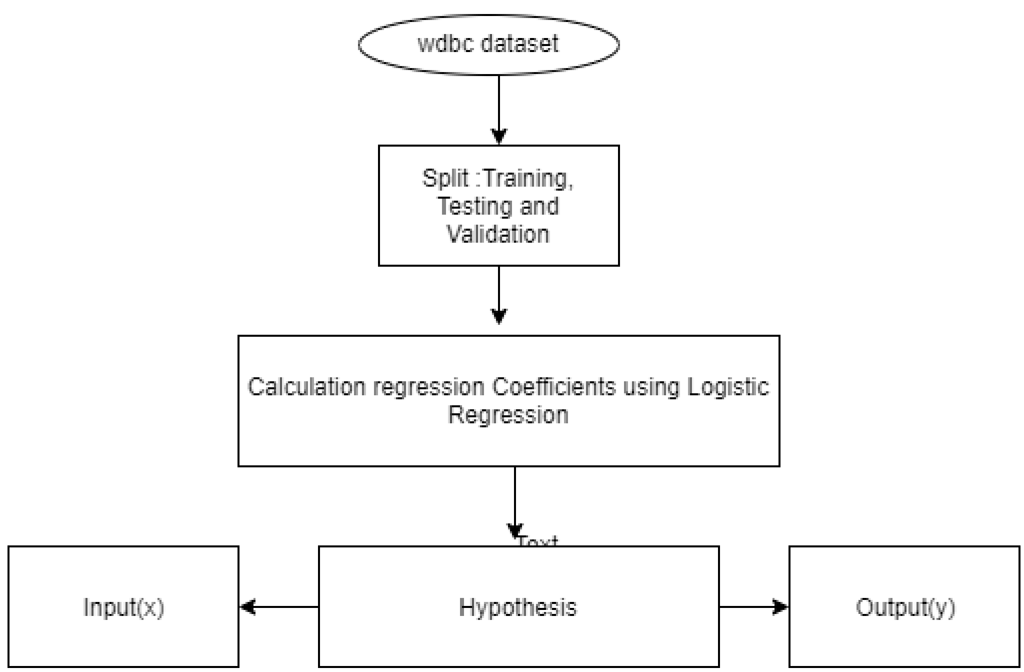


Figure: Configuration Graph

Logistic Regression is used to train the data. We used sigmoid function in order to predict values and map real value between 0 and 1.

$$h\theta(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

The Hypothesis of logistic regression

Inputs are passed through prediction function and we expect our classifier to give set of outputs or classes which is in out case Benign(0) and Malign(1).

After training the dataset, validate the regression performance of the model on the validation set. Tune hyper-parameters such that the cost function is minimum to give better performance on the validation set.

6 times the number of iteration and learning rate was altered to find the best case.
Below are the results for the same :

Tuning Hyper-Parameters

<u>epochs = 25000 & learningrate = 0.05</u>

Training data set

[[264 5]
[41 145]]

Training_Accuracy = 0.8989010989010989

Training_Precision = 0.9814126394052045

Training_Recall = 0.8655737704918033

Validation dataset

[[43 0]
[2 12]]

Validation_Accuracy = 0.964912280701754

Validation_Precision = 1.0

Validation_Recall = 0.9555555555555556

<u>epochs = 25000 & learningrate = 0.1</u>

Training data set

[[262 7]
[39 147]]

Training_Accuracy = 0.8989010989010989

Training_Precision = 0.9739776951672863

Training_Recall = 0.8704318936877077

Validation dataset

[[41 2]
[2 12]]

Validation_Accuracy = 0.929824561403508

Validation_Precision = 0.953488372093023

Validation_Recall = 0.9534883720930233

epochs = 30000 & learningrate = 0.05

Training data set	**Validation dataset**
[[264 5] [40 146]]	[[43 0] [2 12]]
Training_Accuracy = 0.9010989010989011	Validation_Accuracy = 0.964912280701754
Training_Precision = 0.9814126394052045	Validation_Precision = 1.0
Training_Recall = 0.868421052631579	Validation_Recall = 0.9555555555555556

epochs = 30000 & learningrate = 0.1

Training data set	**Validation dataset**
[[262 7] [35 151]]	[[40 3] [2 12]]
Training_Accuracy = 0.9076923076923077	Validation_Accuracy = 0.912280701754385
Training_Precision = 0.9739776951672863	Validation_Precision = 0.930232558139534
Training_Recall = 0.8821548821548821	Validation_Recall = 0.9523809523809523

epochs = 40000 & learningrate = 0.05

Training data set	**Validation dataset**
[[262 7] [39 147]]	[[42 1] [2 12]]
Training_Accuracy = 0.8989010989010989	Validation_Accuracy=0.947368421052631
Training_Precision = 0.9739776951672863	Validation_Precision = 0.976744186046511
Training_Recall = 0.8704318936877077	Validation_Recall = 0.9545454545454546

epochs = 40000 & learningrate = 0.1

Training data set

[[261   8]
 [34 152]]

Training_Accuracy = 0.9076923076923077

Training_Precision = 0.9702602230483272

Training_Recall = 0.8847457627118644

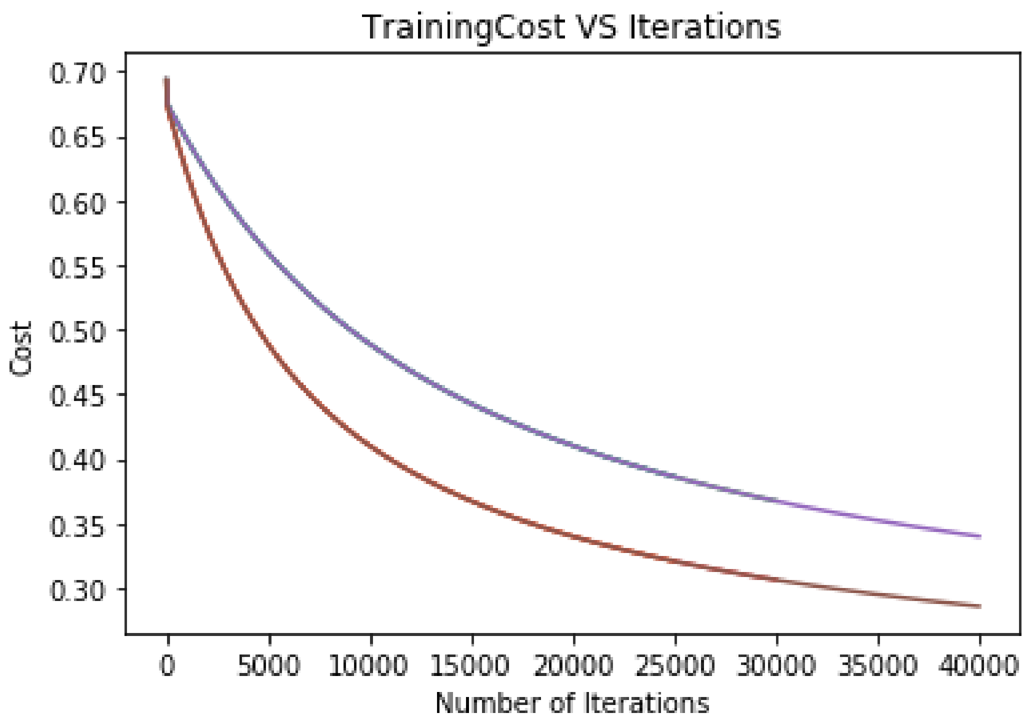
Validation dataset

[[40 3]
 [2 12]]

Validation_Accuracy = 0.912280701754385

Validation_Precision = 0.930232558139534

Validation_Recall = 0.9523809523809523

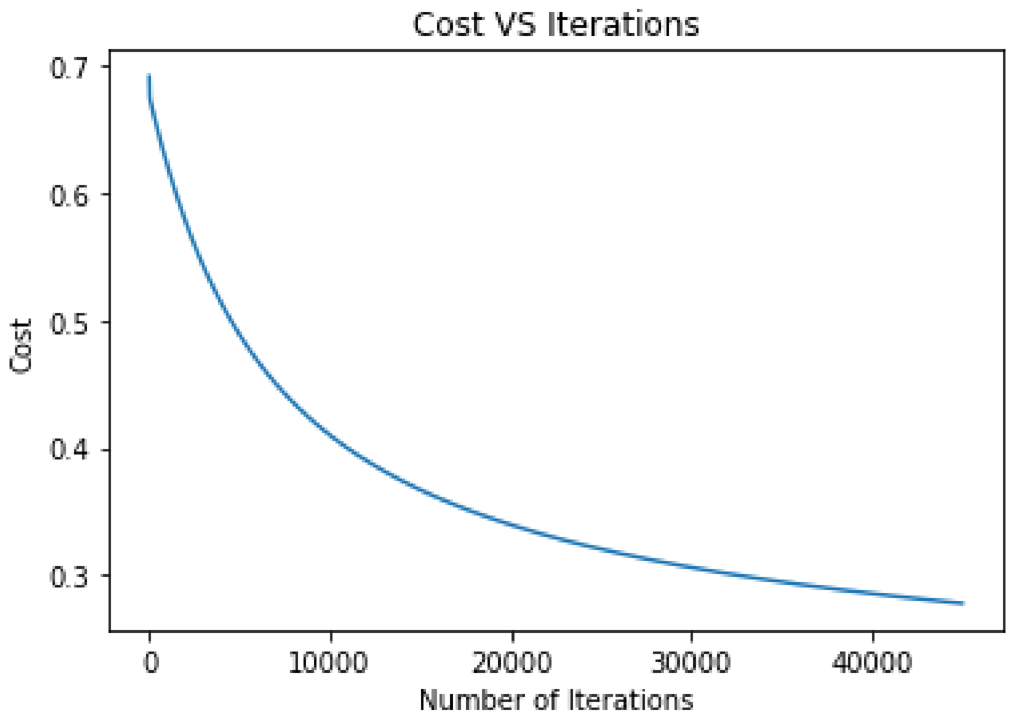


5 Result

By tuning hyperparameters, it is observed that the cost function gradually approaches minimum value.

The best case is with Number of iterations=40000 and learning rate =0.1

Training_Accuracy = 0.9120879120879121
Testing_Accuracy = 0.8947368421052632
Validation_Accuracy = 0.9122807017543859



6 Conclusion

The performance of Logistic Regression is high given the symptoms for breast cancer should exchibit certain clear patterns. We have achieved test accuracy of ≈90% on held out test data.

References

[1] https://medium.com/datadriveninvestor/breast-cancer-detection-using-machine-learning-475d3b63e18e

[2] https://www.kaggle.com/junkal/breast-cancer-prediction-using-machine-learning

[3] https://www.coursera.org/lecture/data-analysis-with-python/data-normalization-in-python-pqNBS

[4] https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148