CSE 474/574: Introduction to Machine Learning

LOGISTIC REGRESSION ALGORITHM TO PREDICT BREAST CANCER

By,

SHUBHANGI MANE

#50295705

ABSTRACT:

Breast cancer is one the most prominent cancers among women, irrespective of the age group. In this project, the diagnosis of breast cancer is complemented by using logistic regression and a model is created to estimate the breast cancer risk. These results can be used to make a proper judgment as to estimate the presence of breast cancer.

INTRODUCTION:

In this paper Breast cancer dataset is collected from, the Wisconsin Diagnostic Breast and has 569 instances with 31 attributes. The features used for classification are pre-computed from images of a fine needle aspirate of a breast mass. Data set is pre-processed first and fed Simple Logistic-regression method. The results obtained are evaluated on various parameters like Accuracy, Precision and Recall. The target values are scalars that can take two values {1: Malignant, 0: Benign}. Although the training target values are discrete we use logistic regression to obtain real values which is more useful for classification.

The classification table from 569 samples shows the occurrence from prediction and observation samples, producing percentage of correct classification for results is 98.24%. The accuracy is compared with validated samples which are 57 samples and the percentage of correct classification is 100.00%.

DATASET:

Wisconsin Diagnostic Breast Cancer (WDBC) dataset has 569 instances with 32 attributes. Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. We have used logistic regression to train the model. This analysis aims to observe which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in model selection and hyper parameter selection. The goal is to classify whether the breast cancer is benign or malignant.

Attribute Information:

ID number 2) Diagnosis (M = malignant, B = benign)

Ten real-valued features are computed for each cell nucleus:

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 (perimeter² / area — 1.0)

7.concavity (severity of concave portions of the contour)

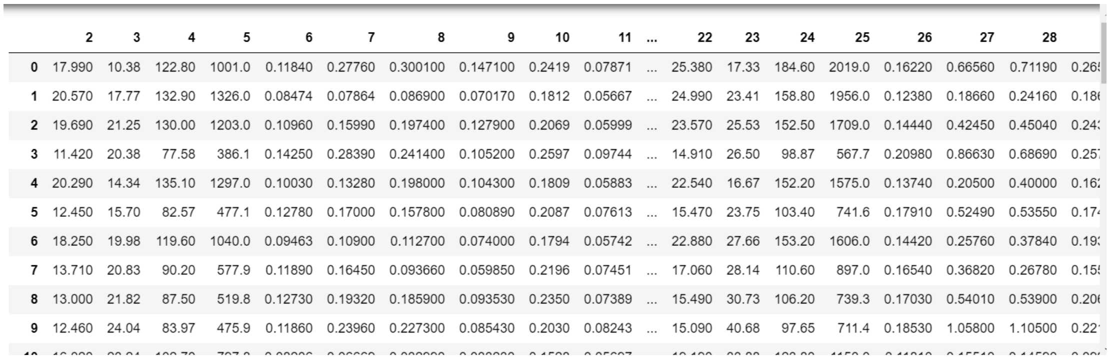
8.concave points (number of concave portions of the contour)

9.symmetry

10.fractal dimension (“coastline approximation” — 1)

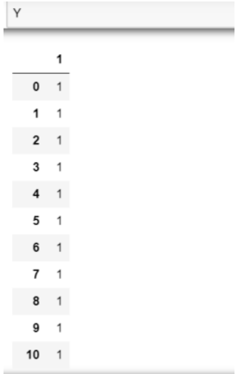
The mean, standard error and “worst” or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, and field 23 is Worst Radius.

The data set looks as follows:



	2	3	4	5	6	7	8	9	10	11	...	22	23	24	25	26	27	28	
0	17.990	10.38	122.80	1001.0	0.11840	0.27760	0.300100	0.147100	0.2419	0.07871	...	25.380	17.33	184.60	2019.0	0.16220	0.66560	0.71190	0.265
1	20.570	17.77	132.90	1326.0	0.08474	0.07864	0.086900	0.070170	0.1812	0.05667	...	24.990	23.41	158.80	1956.0	0.12380	0.18660	0.24160	0.186
2	19.690	21.25	130.00	1203.0	0.10960	0.15990	0.197400	0.127900	0.2069	0.05999	...	23.570	25.53	152.50	1709.0	0.14440	0.42450	0.45040	0.243
3	11.420	20.38	77.58	386.1	0.14250	0.28390	0.241400	0.105200	0.2597	0.09744	...	14.910	26.50	98.87	567.7	0.20980	0.86630	0.68690	0.257
4	20.290	14.34	135.10	1297.0	0.10030	0.13280	0.198000	0.104300	0.1809	0.05883	...	22.540	16.67	152.20	1575.0	0.13740	0.20500	0.40000	0.162
5	12.450	15.70	82.57	477.1	0.12780	0.17000	0.157800	0.080890	0.2087	0.07613	...	15.470	23.75	103.40	741.6	0.17910	0.52490	0.53550	0.174
6	18.250	19.98	119.60	1040.0	0.09463	0.10900	0.112700	0.074000	0.1794	0.05742	...	22.880	27.66	153.20	1606.0	0.14420	0.25760	0.37840	0.193
7	13.710	20.83	90.20	577.9	0.11890	0.16450	0.093660	0.059850	0.2196	0.07451	...	17.060	28.14	110.60	897.0	0.16540	0.36820	0.26780	0.155
8	13.000	21.82	87.50	519.8	0.12730	0.19320	0.185900	0.093530	0.2350	0.07389	...	15.490	30.73	106.20	739.3	0.17030	0.54010	0.53900	0.206
9	12.460	24.04	83.97	475.9	0.11860	0.23960	0.227300	0.085430	0.2030	0.08243	...	15.090	40.68	97.65	711.4	0.18530	1.05800	1.10500	0.221

‘Diagnosis’ is the column which is used to predict, which says if the cancer is M = malignant or B = benign. 1 means the cancer is malignant and 0 means benign. We can identify that out of the 569 persons, 357 are labeled as B (benign) and 212 as M (malignant). We have transformed the value of M as 1 and B as 0 as shown below



Y

	1
0	1
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1

SPLITING THE DATASET:

We have divided the data set into three parts namely Training set, Test set and Validate set. Here the training set comprises of 80% of the dataset, whereas Test set and Validate set comprises 10% of dataset each. Training input set contains 455 instances with 30 attributes and Training output set contains 455 instances with 1 attribute which predicts whether it is malignant or benign. Test input set and Validate input set each contain 57 instances with 30 attributes, where output contains 57 instances.

The model is initially fit on a Training dataset. The model is trained on the training dataset using a supervised learning method. In practice, the training dataset often consist of pairs of an input vector and the corresponding output vector which is commonly denoted as the target, either predicts whether it is malignant or benign.

INPUT DATASET:

DATASET	**INSTANCES**	**ATTRIBUTES**
ACTUAL	569	30
TRAINING (80%)	455	30
VALIDATION (10%)	57	30
TEST (10%)	57	30

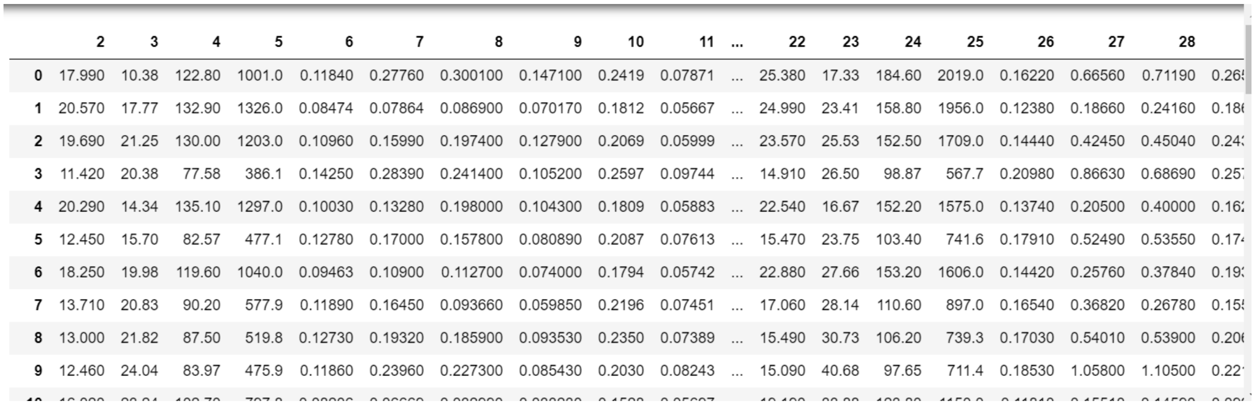
OUTPUT DATASET:

DATASET	**INSTANCES**	**ATTRIBUTES**
ACTUAL	569	1
TRAINING (80%)	455	1
VALIDATION (10%)	57	1
TEST (10%)	57	1

PREPROCESSING:

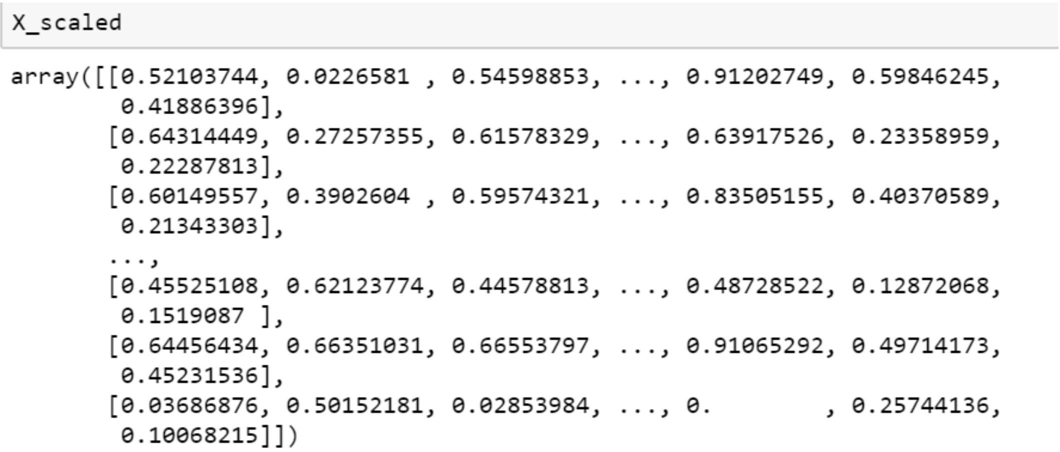
Data preprocessing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn Most of the times, your dataset will contain features highly varying in magnitudes, units and range. We need to bring all features to the same level of magnitudes. This can be achieved by scaling. I have used MinMax Scaler. This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one. Normalization is applied for both the inputs and output datasets, so the range of values are between 0 and 1. The column named ID is removed from the given dataset as, the ID in the given dataset corresponds to the patients ID and it has no significance in prediction of malignance or benign.

DATA BEFORE PREPROCESSING:



	2	3	4	5	6	7	8	9	10	11	...	22	23	24	25	26	27	28	
0	17.990	10.38	122.80	1001.0	0.11840	0.27760	0.300100	0.147100	0.2419	0.07871	...	25.380	17.33	184.60	2019.0	0.16220	0.66560	0.71190	0.265
1	20.570	17.77	132.90	1326.0	0.08474	0.07864	0.086900	0.070170	0.1812	0.05667	...	24.990	23.41	158.80	1956.0	0.12380	0.18660	0.24160	0.186
2	19.690	21.25	130.00	1203.0	0.10960	0.15990	0.197400	0.127900	0.2069	0.05999	...	23.570	25.53	152.50	1709.0	0.14440	0.42450	0.45040	0.243
3	11.420	20.38	77.58	386.1	0.14250	0.28390	0.241400	0.105200	0.2597	0.09744	...	14.910	26.50	98.87	567.7	0.20980	0.86630	0.68690	0.257
4	20.290	14.34	135.10	1297.0	0.10030	0.13280	0.198000	0.104300	0.1809	0.05883	...	22.540	16.67	152.20	1575.0	0.13740	0.20500	0.40000	0.162
5	12.450	15.70	82.57	477.1	0.12780	0.17000	0.157800	0.080890	0.2087	0.07613	...	15.470	23.75	103.40	741.6	0.17910	0.52490	0.53550	0.174
6	18.250	19.98	119.60	1040.0	0.09463	0.10900	0.112700	0.074000	0.1794	0.05742	...	22.880	27.66	153.20	1606.0	0.14420	0.25760	0.37840	0.193
7	13.710	20.83	90.20	577.9	0.11890	0.16450	0.093660	0.059850	0.2196	0.07451	...	17.060	28.14	110.60	897.0	0.16540	0.36820	0.26780	0.155
8	13.000	21.82	87.50	519.8	0.12730	0.19320	0.185900	0.093530	0.2350	0.07389	...	15.490	30.73	106.20	739.3	0.17030	0.54010	0.53900	0.206
9	12.460	24.04	83.97	475.9	0.11860	0.23960	0.227300	0.085430	0.2030	0.08243	...	15.090	40.68	97.65	711.4	0.18530	1.05800	1.10500	0.221

DATA AFTER PREPROCESSING:



```
X_scaled
```

```
array([[0.52103744, 0.0226581 , 0.54598853, ..., 0.91202749, 0.59846245,
        0.41886396],
       [0.64314449, 0.27257355, 0.61578329, ..., 0.63917526, 0.23358959,
        0.22287813],
       [0.60149557, 0.3902604 , 0.59574321, ..., 0.83505155, 0.40370589,
        0.21343303],
       ...,
       [0.45525108, 0.62123774, 0.44578813, ..., 0.48728522, 0.12872068,
        0.1519087 ],
       [0.64456434, 0.66351031, 0.66553797, ..., 0.91065292, 0.49714173,
        0.45231536],
       [0.03686876, 0.50152181, 0.02853984, ..., 0.        , 0.25744136,
        0.10068215]])
```

ARCHITECTURE:

Logistic regression produces a logistic curve, which is limited to values between 0 and 1. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 or 0. Logistic regression uses sigmoid function**.**

LOGISTIC REGRESSION EQUATIONS:

MODEL:

Input = Dataset with 569 instances for 30 attributes.

Output = 0 or 1

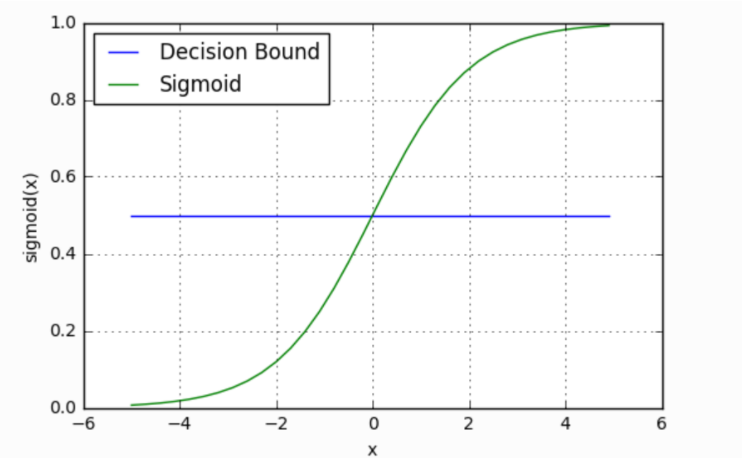
Hypothesis = Z = W0+W1X+W2X2+….+WnXn + Bias

hΘ(x) = sigmoid (Z)

SIGMOID Function:

g(z) = 1/(1 + e ^(−z))

$$S(z) = \frac{1}{1+e^{-z}}$$



COST Function:

Loss(hΘ(x),y)= -ylog(hΘ(x))-(1-y)log(1- hΘ(x))

Gradients:

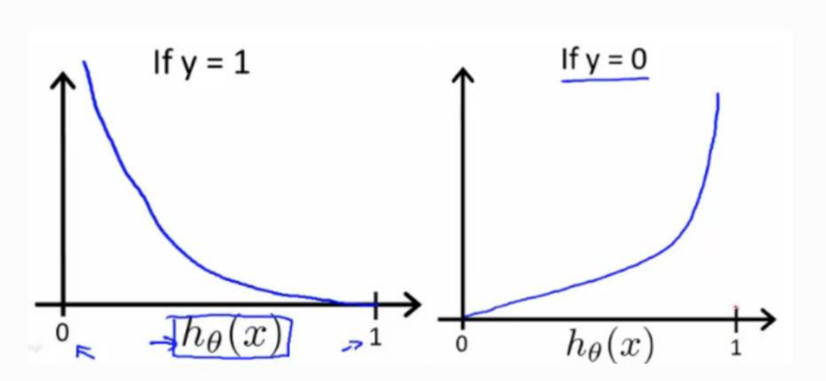
dw = (1 / m) * (x* (Sigmoid(W.T* X)-T))

db = (1 / m) * sum((Sigmoid(W.T* X)-Y)

COST= loss(hΘ(x),y)/ m(number of features)

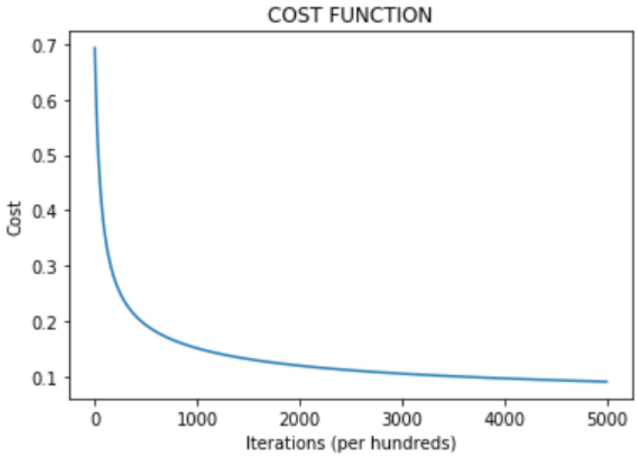
$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$$

OUTPUT:



RESULTS AND ANALYSIS FOR TRAINING SET:

The below graph depicts Cost versus Number of iterations at **learning rate=0.3**

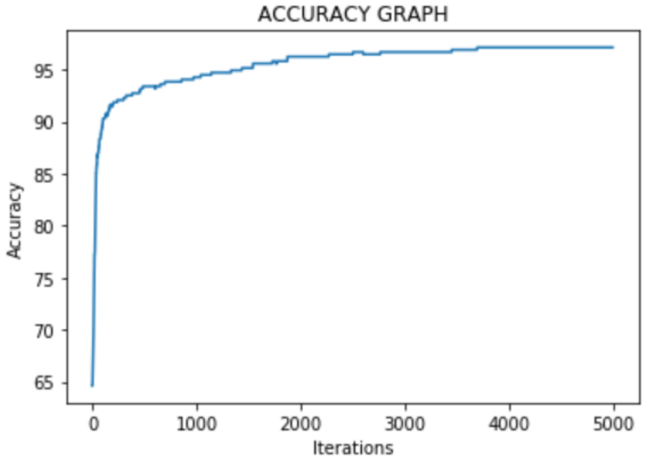


Accuracy for training set is 98.46%

Precision for Training set is 98.80%

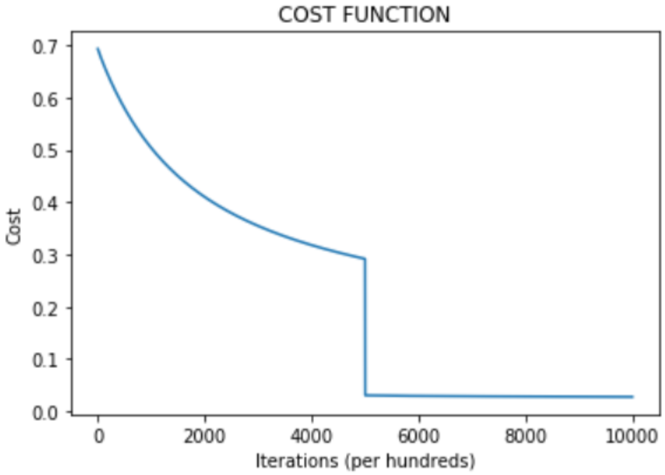
Recall for Training set is 97.40%

Accuracy graph for Training set:



RESULTS AND ANALYSIS FOR VALIDATION SET:

The below graph depicts Cost versus Number of iterations at **learning rate: 0.01**

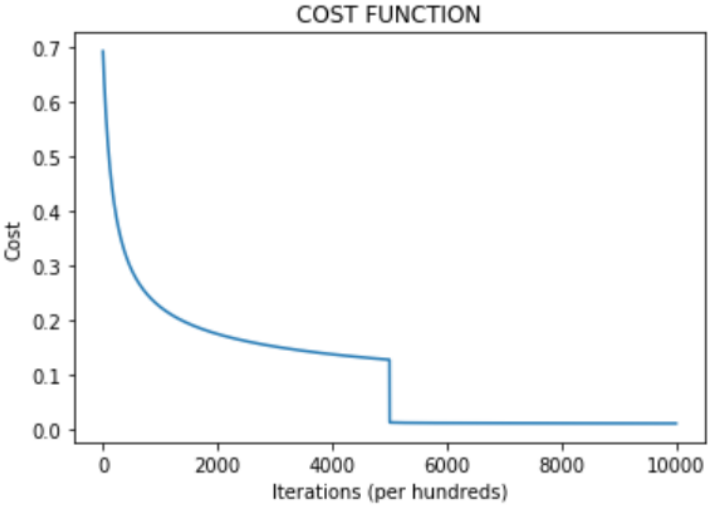


Accuracy for Validation set is 96.49%

Precision for Validation set is 100.00%

Recall for Validation set is 87.50%

The below graph depicts Cost versus Number of iterations at **learning rate: 0.1**

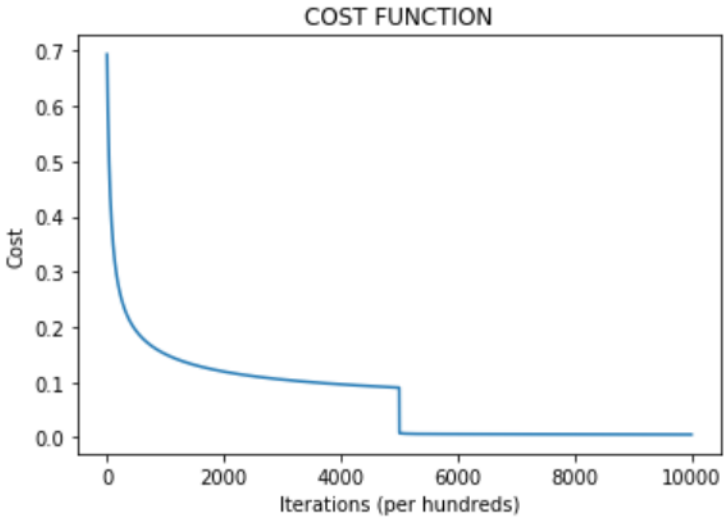


Accuracy for Validation set is 98.24%

Precision for Validation set is 100.00%

Recall for Validation set is 93.75%

The below graph depicts Cost versus Number of iterations at **learning rate: 0.3**



Accuracy for Validation set is 100.00%

Precision for Validation set is 100.00%

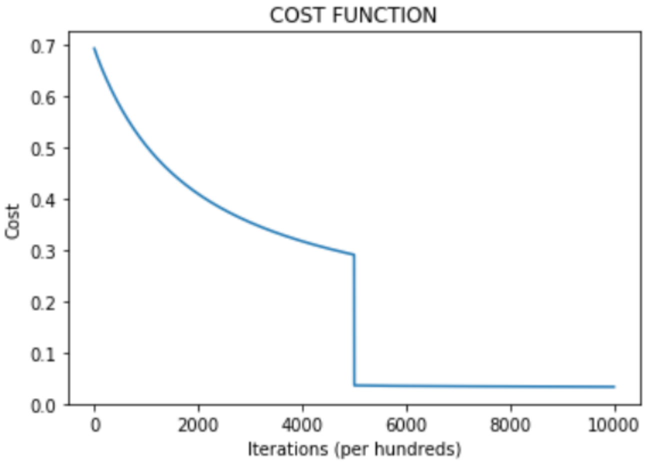
Recall for Validation set is 100.00%

CONCLUSION:

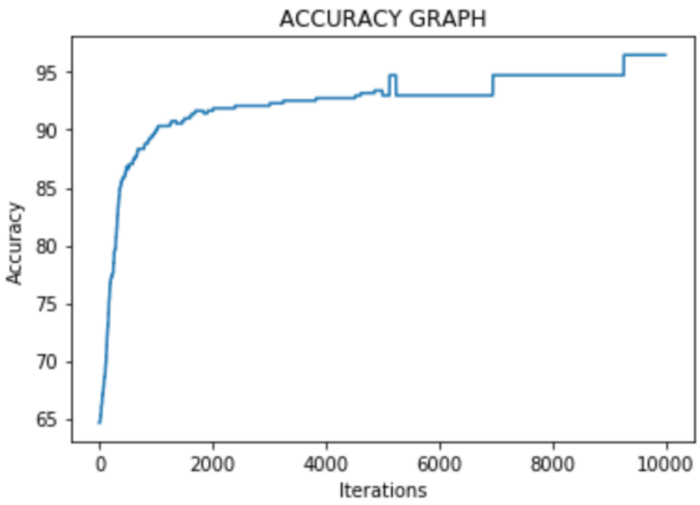
From the above observations it can be concluded that the best **learning rate is at 0.3**, where accuracy is 96%

Following are the observations on Test dataset:

The below graph depicts Cost versus Number of iterations at **learning rate: 0.3**



Accuracy graph for Test set:



Accuracy for Test set is 96.49%

Precision for Test set is 96.30%

Recall for Test set is 96.30%

REFERENCES:

1. https://www.researchgate.net/publication/331233978_Predicting_Breast_Cancer_using_Logistic_Regression_and_Multi-Class_Classifiers
2. https://www.semanticscholar.org/paper/BREAST-CANCER-ANALYSIS-USING-LOGISTIC-REGRESSION-Yusuff-Mohamad/c36ca47eae1da05d61a71ee0517c256d9232577a
3. https://www.google.com/search?client=firefox-b-d&q=breast+cancer+classification+using+logistic+regression+reports
4. https://www.arpapress.com/Volumes/Vol10Issue1/IJRRAS_10_1_02.pdf
5. https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8
6. https://towardsdatascience.com/logistic-regression-for-dummies-a-detailed-explanation-9597f76edf46
7. https://medium.com/data-science-group-iitr/logistic-regression-simplified-9b4efe801389
8. https://www.coursehero.com/file/38950423/logistic-regression-beamerpdf/
9. https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html