Chiranjeev
P21CS007
chiranjeev.1@iitj.ac.in

Assignment - 1
Half-space Classifier and Logistic Regression

01-09-2021

I. Dataset Description:

a. **Name of Dataset :** Breast Cancer Wisconsin (Diagnostic) Dataset (link - Dataset link)

b. **About Dataset :** Features in the data are computed from a digitalized image of a fine needle aspirate (FNA) of breast mass that describe characteristics of the cell nuclei present in the image in the 3-dimensional space.

c. **Dataset features :**

```
id
diagnosis
radius_mean
texture_mean
perimeter_mean
area_mean
smoothness_mean
compactness_mean
concavity_mean
concave points_mean
symmetry_mean
fractal_dimension_mean
radius_se
texture_se
perimeter_se
area_se
smoothness_se
compactness_se
concavity_se
concave points_se
symmetry_se
fractal_dimension_se
radius_worst
texture_worst
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave points_worst
symmetry_worst
fractal_dimension_worst
```

d. **Features Dropped :**
Feature ‘id’ is dropped as for classification task id is not an attribute of breast.
And also found that there are not any Not Available (NA) values in the dataset.

e. **Features Selected :**
All features mentioned above are chosen except ‘id’ for training and testing as all features describes the attributes of breast for cancer detection.

II. Approach for Splitting the Dataset:

Used train_test_split of sklearn to split the dataset into train and test.

Tried 3 different types of splits:

1. **70 % - train set, 30% test set**
2. **80% - train set, 20% test set**
3. **90 % - train set, 10% test set**

III. Techniques Used:

A. Half space classifier using LP solver :

Pre-processing Steps :

Steps to make Non-linearly separable dataset as Linearly separable dataset:

1. Created train test split of train:test on whole dataset(samples)
2. On training set centroid is calculated for 2 classes Malignant and Benign and Euclidean distance is performed on each data point in train set thus finally those data points are selected which are near to the centroid for each class.
3. Now, data is made linearly separable (checked by plotting PCA graph and using linear programming solver).
4. Now training set for Linearly Separable Dataset is X_sep, and its ground truth labels (Y_sep) are:
   -1(for Malignant), 1(for Benign)
5. And test set for Linear Separable Dataset remains unchanged so that it can tell how the model to show some data points in test set can't be classified correctly. (i.e., X_test,Y_test)

API-Lp Solver used: Scipy (scipy.optimize.linprog)

1. **70 % - train set for Lp solver, 30% test set :**

    1. **Training Accuracy :** 100.0% (259 correctly classified and 0 incorrectly classified)
    2. **Testing Accuracy :** 92.9824% (159 correctly classified and 12 incorrectly classified)
    3. **Confusion Matrix for Test set:**

```
confusion_matrix:
 [[ 54   8]
 [  4 105]]
```

2. **80 % - train set for Lp solver, 20% test set :**

    1. **Training Accuracy :** 100.0% (288 correctly classified and 0 incorrectly classified)

    2. **Testing Accuracy :** 92.1052% (105 correctly classified and 9 incorrectly classified)
    3. **Confusion Matrix for Test set:**

```
confusion_matrix:
 [[39  4]
 [ 5 66]]
```

3. **90 % - train set for Lp solver, 10% test set :**

    1. **Training Accuracy :** 100.0% (326 correctly classified and 0 incorrectly classified)

    2. **Testing Accuracy :** 94.7368% (54 correctly classified and 3 incorrectly classified)

    3. **Confusion Matrix for Test set:**

```
confusion_matrix:
 [[16  1]
 [ 2 38]]
```

B. <u>Half space classifier using Perceptron Algorithm</u> :

Pre-processing Steps :

<u>Steps to make Non-linearly separable dataset as Linearly separable dataset:</u>

1. Created train test split of train:test on whole dataset(samples)
2. On training set centroid is calculated for 2 classes Malignant and Benign and Euclidean distance is performed on each data point in train set thus finally those data points are selected which are near to the centroid for each class.
3. Now, data is made linearly separable (checked by plotting PCA graph and using linear programming solver).
4. Now training set for Linearly Separable Dataset is
   X_sep, and its ground truth labels (Y_sep) are:
   -1(for Malignant), 1(for Benign)
5. And test set for Linear Separable Dataset remains unchanged so that it can tell how the model to show some data points in test set can't be classified correctly. (i.e., X_test,Y_test)

1. 70 % - train set, 30% test set :

1. **Training Accuracy :** 100% (259 correctly classified and 0 incorrectly classified)
2. **Testing Accuracy :** 92.9824% (159 correctly classified and 12 incorrectly classified)
3. **Number Of Iterations :** 44
4. **Confusion Matrix for Test set:**

```
confusion_matrix:
 [[ 54   8]
 [  4 105]]
```

2. 80 % - train set, 20% test set :

1. **Training Accuracy :** 100% (288 correctly classified and 0 incorrectly classified)

2. **Testing Accuracy :** 94.7368% (108 correctly classified and 6 incorrectly classified)

3. **Number Of Iterations :** 34

4. **Confusion Matrix for Test set:**

```
confusion_matrix:
 [[40  3]
 [ 3 68]]
```

3. 90 % - train set, 10% test set :

1. **Training Accuracy :** 100% (326 correctly classified and 0 incorrectly classified)

2. **Testing Accuracy :** 94.7368% (54 correctly classified and 3 incorrectly classified)

3. **Number Of Iterations :** 29

4. **Confusion Matrix for Test set:**

```
confusion_matrix:
 [[17  0]
 [ 3 37]]
```

C. <u>Logistic Regression Classifier</u> :

Note : Here used whole Breast Cancer Dataset for train and test

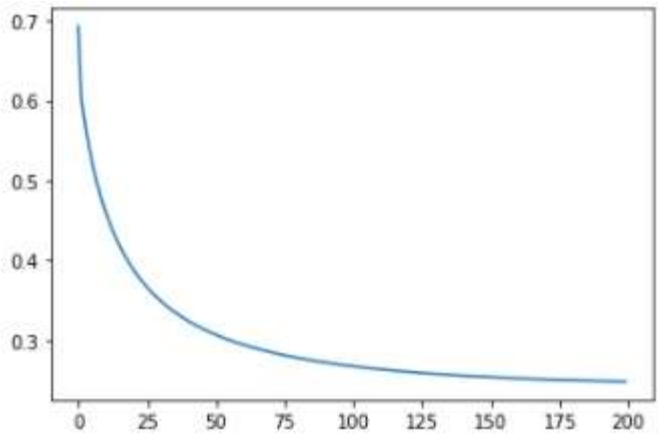
(i) <u>Model build from Scratch</u> :

1. 70 % - train set, 30% test set :

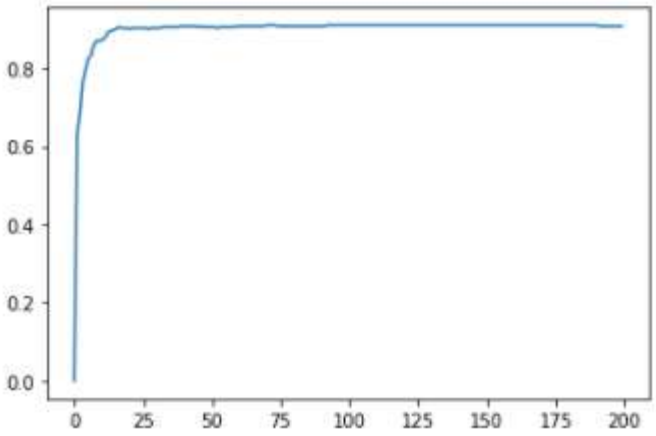
1. **Training Accuracy :** 90.7035% (361 correctly classified and 37 incorrectly classified)
2. **Testing Accuracy :** 94.7368% (162 correctly classified and 9 incorrectly classified)
3. **Confusion Matrix for Test set :**

```
confusion_matrix:
 [[ 57   6]
 [  3 105]]
```

4. **Learning rate :** 0.001
5. **Number Of Epochs :** 200
6. **Graph Plot : No. of epochs (on X-axis) Vs Loss (on Y-axis):**



7. **Graph Plot : No. of epochs (on X-axis) Vs Accuracy (on Y-axis):**



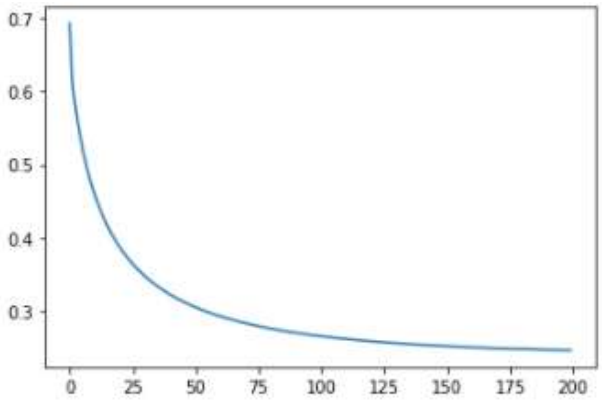
2. 80 % - train set, 20% test set :

1. **Training Accuracy :** 91.4285% (416 correctly classified and 39 incorrectly classified)
2. **Testing Accuracy :** 93.8596% (107 correctly classified and 7 incorrectly classified)
3. **Confusion Matrix for Test set :**

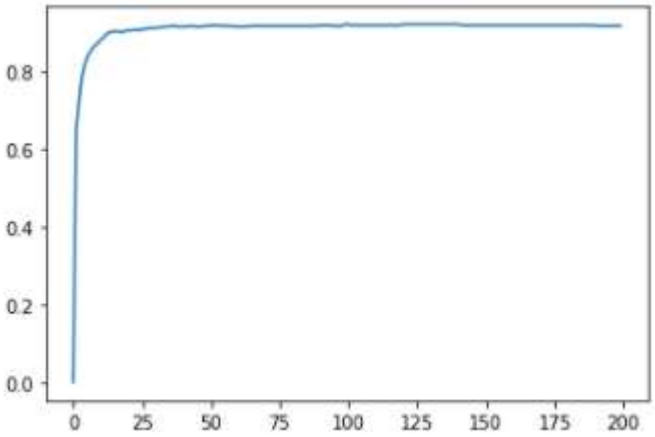
```
confusion_matrix:
 [[142  27]
 [ 12 274]]
```

4. **Learning rate :** 0.001
5. **Number Of Epochs :** 200

6. **Graph Plot : No. of epochs (on X-axis) Vs Loss (on Y-axis):**



8. **Graph Plot : No. of epochs (on X-axis) Vs Accuracy (on Y-axis):**



3. 90 % - train set, 10% test set :

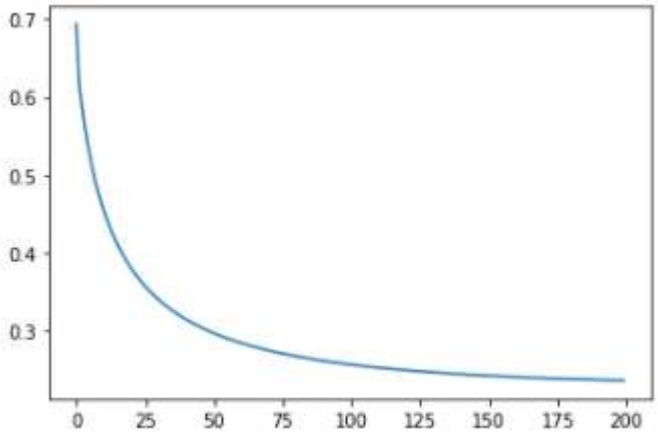
1. **Training Accuracy :** 91.6015% (469 correctly classified and 43 incorrectly classified)
2. **Testing Accuracy :** 92.9824% (53 correctly classified and 4 incorrectly classified)
3. **Confusion Matrix for Test set :**

```
confusion_matrix:
 [[15  2]
 [ 2 38]]
```

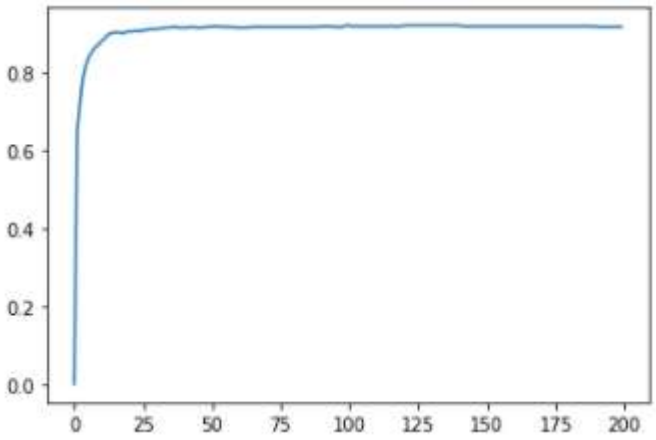
4. **Learning rate :** 0.001

5. **Number Of Epochs :** 200

6. **Graph Plot : No. of epochs (on X-axis) Vs Loss (on Y-axis):**



7. **Graph Plot : No. of epochs (on X-axis) Vs Accuracy (on Y-axis):**



(ii) <u>Model build with Sklearn</u> :

1. **70 % - train set, 30% test set :**

    1. **Training Accuracy :** 93.9698% (374 correctly classified and 24 incorrectly classified)

    2. **Testing Accuracy :** 97.0760% (166 correctly classified and 5 incorrectly classified)

    3. **Confusion Matrix for Test set :**

```
confusion_matrix:
 [[ 59   4]
 [  1 107]]
```

2. **80 % - train set, 20% test set :**

    1. **Training Accuracy :** 94.5054% (430 correctly classified and 25 incorrectly classified)

    2. **Testing Accuracy :** 96.49122% (110 correctly classified and 4 incorrectly classified)

    3. **Confusion Matrix for Test set :**

```
confusion_matrix:
 [[40  3]
 [ 1 70]]
```

3. **90 % - train set, 10% test set :**

    1. **Training Accuracy :** 95.5078% (489 correctly classified and 23 incorrectly classified)

    2. **Testing Accuracy :** 98.2456% (56 correctly classified and 1 incorrectly classified)

    3. **Confusion Matrix for Test set :**

```
confusion_matrix:
 [[16  1]
 [ 0 40]]
```