To implement the Logistic Regression algorithm to solve the Binary classification problem.

PROJECT DESCRIPTION

- Programming language used : Python
- Data Structure : Lists, matrix
- File Name : LogisticRegression.py
- Inputs: classification.txt
- Output:
    - Logistic Regression – Weight matrix such that the non-linear transformation's $\Theta(s) = e^s/(1+e^s)$ predicted probabilities for the input lie between 0 and 1.

PART 1 - IMPLEMENTATION

- Modules created :

  A. Logistic Regression:

  1. ReadFile(): to save the input file into List of points for the given dimensions in the global variable Data_List and output_coordinate_list for the classification label.
  2. sigmoid(z): calculates $\Theta(s) = e^s/(1+e^s)$ for each data point.
  3. CountUnclassified(theta, X, y): Calculates the misclassified points in each iteration based on the probability rule, if >0.5 and classification variable =1 , then classified.
  4. Logistic(): For each of the 7000 iterations, checks the probability of each point and finds the error. The objective is to minimize the error as we iterate forward. The weights are recalculated accordingly.
  5. Weightupdate(m): to store the weights after every iteration in a global variable weightMatrix.

- Termination Condition :
    - LogisticRegression: Terminates when 7000 iterations are complete or all the points in the input are correctly classified.
- Result Interpretation:

  Logistic regression output presents the weight vector which satisfies the binary classification of points into positive and negative classes.