**TO RUN THE PROGRAM:**

To complile the java program: javac LGD.java

To run the java program: java LGD

Standard libraries:

import java.util.Arrays;

import java.util.ArrayList;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.List;

import java.util.Scanner;

import java.util.Random;

jupyterNotebooks: seaborn and panda\_profiling

**Running Batch mode**

**Interactive Mode? (y or n)**

n

It will run all combinations of the following parameters:

learningRate\_inputs = {"0.1", "0.01", "0.001"};

numIterations\_inputs = {"100", "1000"};

allowableError\_inputs = {"0.1", "0.01"};

randomnessOfSGD\_inputs = {"0.1", "0.01"};

You should see the output:

LR: 0.100000 E: 0.1000 randomness var: 0.1000 Num Iterations: 100 \*\*\* Total Correct: 57 Total Incorrect: 2 Percent correct 96.61016949152543

LR: 0.100000 E: 0.1000 randomness var: 0.1000 Num Iterations: 1000 \*\*\* Total Correct: 57 Total Incorrect: 2 Percent correct 96.61016949152543

LR: 0.100000 E: 0.1000 randomness var: 0.0100 Num Iterations: 100 \*\*\* Total Correct: 55 Total Incorrect: 4 Percent correct 93.22033898305084

LR: 0.100000 E: 0.1000 randomness var: 0.0100 Num Iterations: 1000 \*\*\* Total Correct: 57 Total Incorrect: 2 Percent correct 96.61016949152543

LR: 0.100000 E: 0.0100 randomness var: 0.1000 Num Iterations: 100 \*\*\* Total Correct: 57 Total Incorrect: 2 Percent correct 96.61016949152543

LR: 0.100000 E: 0.0100 randomness var: 0.1000 Num Iterations: 1000 \*\*\* Total Correct: 57 Total Incorrect: 2 Percent correct 96.61016949152543

LR: 0.100000 E: 0.0100 randomness var: 0.0100 Num Iterations: 100 \*\*\* Total Correct: 55 Total Incorrect: 4 Percent correct 93.22033898305084

LR: 0.100000 E: 0.0100 randomness var: 0.0100 Num Iterations: 1000 \*\*\* Total Correct: 57 Total Incorrect: 2 Percent correct 96.61016949152543

LR: 0.010000 E: 0.1000 randomness var: 0.1000 Num Iterations: 100 \*\*\* Total Correct: 56 Total Incorrect: 3 Percent correct 94.91525423728814

LR: 0.010000 E: 0.1000 randomness var: 0.1000 Num Iterations: 1000 \*\*\* Total Correct: 57 Total Incorrect: 2 Percent correct 96.61016949152543

LR: 0.010000 E: 0.1000 randomness var: 0.0100 Num Iterations: 100 \*\*\* Total Correct: 51 Total Incorrect: 8 Percent correct 86.4406779661017

LR: 0.010000 E: 0.1000 randomness var: 0.0100 Num Iterations: 1000 \*\*\* Total Correct: 56 Total Incorrect: 3 Percent correct 94.91525423728814

LR: 0.010000 E: 0.0100 randomness var: 0.1000 Num Iterations: 100 \*\*\* Total Correct: 57 Total Incorrect: 2 Percent correct 96.61016949152543

LR: 0.010000 E: 0.0100 randomness var: 0.1000 Num Iterations: 1000 \*\*\* Total Correct: 55 Total Incorrect: 4 Percent correct 93.22033898305084

LR: 0.010000 E: 0.0100 randomness var: 0.0100 Num Iterations: 100 \*\*\* Total Correct: 48 Total Incorrect: 11 Percent correct 81.35593220338984

LR: 0.010000 E: 0.0100 randomness var: 0.0100 Num Iterations: 1000 \*\*\* Total Correct: 56 Total Incorrect: 3 Percent correct 94.91525423728814

LR: 0.001000 E: 0.1000 randomness var: 0.1000 Num Iterations: 100 \*\*\* Total Correct: 40 Total Incorrect: 19 Percent correct 67.79661016949152

LR: 0.001000 E: 0.1000 randomness var: 0.1000 Num Iterations: 1000 \*\*\* Total Correct: 42 Total Incorrect: 17 Percent correct 71.1864406779661

LR: 0.001000 E: 0.1000 randomness var: 0.0100 Num Iterations: 100 \*\*\* Total Correct: 51 Total Incorrect: 8 Percent correct 86.4406779661017

LR: 0.001000 E: 0.1000 randomness var: 0.0100 Num Iterations: 1000 \*\*\* Total Correct: 55 Total Incorrect: 4 Percent correct 93.22033898305084

LR: 0.001000 E: 0.0100 randomness var: 0.1000 Num Iterations: 100 \*\*\* Total Correct: 56 Total Incorrect: 3 Percent correct 94.91525423728814

LR: 0.001000 E: 0.0100 randomness var: 0.1000 Num Iterations: 1000 \*\*\* Total Correct: 57 Total Incorrect: 2 Percent correct 96.61016949152543

LR: 0.001000 E: 0.0100 randomness var: 0.0100 Num Iterations: 100 \*\*\* Total Correct: 53 Total Incorrect: 6 Percent correct 89.83050847457628

LR: 0.001000 E: 0.0100 randomness var: 0.0100 Num Iterations: 1000 \*\*\* Total Correct: 53 Total Incorrect: 6 Percent correct 89.83050847457628

**Running Interactively**

If you run it interactively, you can specify the parameters (learning rate, allowable error, randomness numbers, Stochastic Gradient Descent or Batch Gradient Descent, number of iterations)

**Interactive Mode? (y or n)**

y

**Enter learning rates (float -> e.g 0.01) (space separated)**

0.1 0.01 0.001

**Enter allowable error (float -> e.g. 0.1) (space separated)**

90.00 99.00

**Enter randomness numbers (0.0 - 1.0) for SGD or (2.0) for Batch Gradient Descent (float -> e.g. 0.01) (space separated)**

2.0 0.1 0.01

**Enter max number of iterations (int -> e.g. 1000) (space separated)**

1000

**Explanation of Parameters**

You can run Batch or Stochastic Gradient Descent by selecting a real value for randomness (I wasn't sure how random Stochastic Descent needed to be - e.g. how often to update weight vectors in a given run)

So if you select

1/testSetSize - it will randomly update the weight vector based on one feature vector each iteration

0.1 - it will update the weight vector 10 percent of the time (mini batch)

0.001 - it will update the weight vector 1% of the time. (mini batch)

2.0 - it will take the worst vector (steepest gradient) and update once per iteration based on that

You can specify how close the probability needs to be to satisfy a match to the 0 or 1 diagnosis:

0.10 - means allow for 0.10 error in prediction to still be correct

0.01 - means allow for 0.01 error in prediction to still be correct

So for example if you say 0.50 then you will probably get everything correct because you only need to be 50% accurate to count the response as correct. (Correct means that it matches the test data diagnosis)

Sample Output:

Interactive Mode? (y or n)

y

Enter learning rates (float -> e.g. 0.01) (space separated)

0.01

Enter allowable error (float -> e.g. 0.1) (space separated)

0.1 0.01 0.001

Enter randomness numbers (0.0 - 1.0) for SGD or (2.0) for Batch Gradient Descent (float -> e.g. 0.01) (space separated)

0.1 0.01

Enter max number of iterations (int -> e.g. 1000) (space separated)

1000

Running Logistic Regression SGD on dataset: Arrhythmia\_TrainingData.csv training set size 236 test set size 59

LR: 0.010000 E: 0.1000 randomness var: 0.1000 Num Iterations: 1000 \*\*\* Total Correct: 57 Total Incorrect: 2 Percent correct 96.61016949152543

LR: 0.010000 E: 0.1000 randomness var: 0.0100 Num Iterations: 1000 \*\*\* Total Correct: 56 Total Incorrect: 3 Percent correct 94.91525423728814

LR: 0.010000 E: 0.0100 randomness var: 0.1000 Num Iterations: 1000 \*\*\* Total Correct: 55 Total Incorrect: 4 Percent correct 93.22033898305084

LR: 0.010000 E: 0.0100 randomness var: 0.0100 Num Iterations: 1000 \*\*\* Total Correct: 56 Total Incorrect: 3 Percent correct 94.91525423728814

LR: 0.010000 E: 0.0010 randomness var: 0.1000 Num Iterations: 1000 \*\*\* Total Correct: 57 Total Incorrect: 2 Percent correct 96.61016949152543

LR: 0.010000 E: 0.0010 randomness var: 0.0100 Num Iterations: 1000 \*\*\* Total Correct: 55 Total Incorrect: 4 Percent correct 93.22033898305084

**Changing Data Sets**

You can also change datasets, see more details, etc, but to do that you have to open the main java file (LGD.java) and edit it – changing the name of the data set to one of the commented out ones.

String trainingFilename = "Arrhythmia\_TrainingData.csv";

String testingFilename = "Arrhythmia\_TestingData.csv";

//trainingFilename = "IrisTrainingData.csv";

//testingFilename = "IrisTestingData.csv";

//trainingFilename = "HeartTrainingData.csv";

//testingFilename = "HeartTestingData.csv";

//trainingFilename = "Arrhythmia\_First\_TrainingData.csv";

//testingFilename = "Arrhythmia\_First\_TestingData.csv";

//trainingFilename = "Arrhythmia\_Last\_TrainingData.csv";

//testingFilename = "Arrhythmia\_Last\_TestingData.csv";

**Showing more details about the algorithm**

And setting any of the following to true:

boolean details = false;

boolean summary = true;

boolean showEinEout = false;

or at the top of the LGD class:

private boolean seeVectorInfo = false;

But be warned, these will give very detailed info!