EAS 595: Fundamentals of Artificial Intelligence (Spring 2020)

Dr. David Doerman, Mihir Chauhan
University at Buffalo, The State University of New York
Buffalo, New York 14260
Contact: mihirhem@buffalo.edu

February 17, 2020

1 Task

The task of this project is to perform classification using machine learning. It is a two class problem. The features used for classification are pre-computed from images of a fine needle aspirate (FNA) of a breast mass. Your task is to classify suspected FNA cells to Benign (class 0) or Malignant (class 1) using logistic regression as the classifier. You are required to document the results using the three machine learning tools:

- WEKA: Graphical User Inferface
- Scikit learn: Python machine learning library
- Python from scratch

The dataset in use is the Wisconsin Diagnostic Breast Cancer (wdbc.dataset). Deadline to submit the code and the report on timberlake server is February 26, 2020.

2 Dataset

Wisconsin Diagnostic Breast Cancer (WDBC) dataset will be used for training, validation and testing. The dataset contains 569 instances with 32 attributes (ID, diagnosis (B/M), 30 real-valued input features). Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. Computed features describes the following characteristics of the cell nuclei present in the image:

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 ($perimeter^2/area - 1.0$)
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.

3 Plan of Work

1. **Load wdbc dataset to WEKA:** Using WEKA explorer load and pre-process the dataset.
2. **Perform classification on WEKA:** Using WEKA's classify tool perform logistic regression on processed data.
3. **Analyze classifier results on WEKA:** Using WEKA's Classifier output tool describe the output of the classifier in your report.
4. **Extract features values and Image Ids from the data:** Process the original CSV data files into a Numpy matrix or Pandas Dataframe. Apply feature scaling technique to make sure that all the features are in the same level of magnitude.
5. **Data Partitioning:** Partition your data into training and testing data. Randomly choose 80% of the data for training and the rest for testing.
6. **Train using Scikit learn library:** Use Scikit learn library logistic regression function to train on the dataset.
7. **Print results of Scikit learn library:** Your code should print Accuracy, Precision, Recall and Confusion matrix resulting from scikit learn logistic regression model. Your report should describe the results.
8. **Train using Logistic Regression using Gradient Descent:** Implement Gradient Descent algorithm for logistic regression to train on wdbc dataset.
9. **Print results** Your code should print Accuracy, Precision, Recall and Confusion matrix resulting from logistic regression model coded from scratch. Your report should describe the results.

4 Evaluation

1. **Task 1:** 35 points for describing the classifier results of WEKA.
2. **Task 2:** 35 points for writing code to use Scikit learn library for performing logistic regression on dataset. Also, describe Accuracy, Precision, Recall and confusion matrix resulting from sklearn classifier in your report.
3. **Task 3:** 30 points for implement gradient descent algorithm for logistic regression using Python. Also, describe Accuracy, Precision, Recall and confusion matrix resulting from the classifier in your report.

5 Definitions

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

6 Deliverables

There are two deliverables: report and code. After finishing the project, you may be asked to demonstrate it to the TAs, particularly if your results and reasoning in your report are not clear enough.

1. Report

   The report should describe the results from each of the three tasks. Submit your report named `proj1.pdf` on Ublearns as well as Autolab.

2. Code

   The code for task 2 and 3 should be in Python only. You can submit multiple files, but the name of the entrance file should be main.ipynb. Please provide necessary comments in the code. Python code and data files should be packed in a ZIP file named `proj1code.zip`. Submit the Python code on ublearns as well as Autolab.