Florida Atlantic University (FAU)

Assignment 5

Project objective: **MLP in Weka**

Written By: Kevin Tudor

Table of Contents

Requirements

Part 1/2 – Cost Sensitive Logistic Regression

- This part involves building and evaluating fault prediction models using Cost Sensitive Logistic Regression, implemented in Weka. Your task is to build models to classify modules as fault prone (fp) or not fault prone (nfP) based on the other attributes of programs in the dataset. Each model is to be built and evaluated using 10-fold cross validation on the fit data set, and then validated using the test data set.
- Use the fit dataset to build models based on 10-fold cross validation. When you build the model, you will get several statistical indicators, the measures of the quality of fit (in the case of fit data) and the predictive quality (for the test data), at the end of each run.
    - Confusion matrix
    - Type 1(false positive) and Type 2 (false negative) Error rates
- The "c" value should be varied as discussed in class and in the *tips*. Each of your "c" values and its respective performance measures for both fit and test data.
- Indicate which "c" value you chose and explain why.
- Include your analysis of the results.

Part 2/2 – Multi Layer Perceptron

- This part you will build and evaluate a MLP in Weka with the following criteria:
    - 3 nodes in one hidden layer ("hiddenLayers" setting)
    - 10% validation set size ("validationSetSize" setting)
    - Other MLP parameter settings are Weka default
        - FYI: Turning on and off the "GUI" parameter will provide a nice visual representation of the model you are building and it would be beneficial to you to try various parameters for the MLP and visualizing any
    - Include in your report the results for both fit and test and provide an analysis of the results.
    - Compare the results from Logistic Regression with MLP results. Provide analysis about the comparison in your report.

Part 1: Cost Sensitive Logistic Regression

10-fold -> Ratio	Type I %	Type II %
10-fold -> Ratio (1:2)	7.3%	18.2%
10-fold -> Ratio (1:3)	8.8%	16.4%
10-fold -> Ratio (1:4)	**10.2%**	**14.5%**
10-fold -> Ratio (2:1)	4.40%	36.40%
10-fold -> Ratio (1:1)	6.60%	23.60%
10-fold -> Ratio (2:3)	7.30%	21.80%
10-fold -> Ratio (3:1)	2.20%	41.80%
10-fold -> Ratio (3:2)	5.80%	30.90%
10-fold -> Ratio (1:3.5)	**9.50%**	**14.50%**

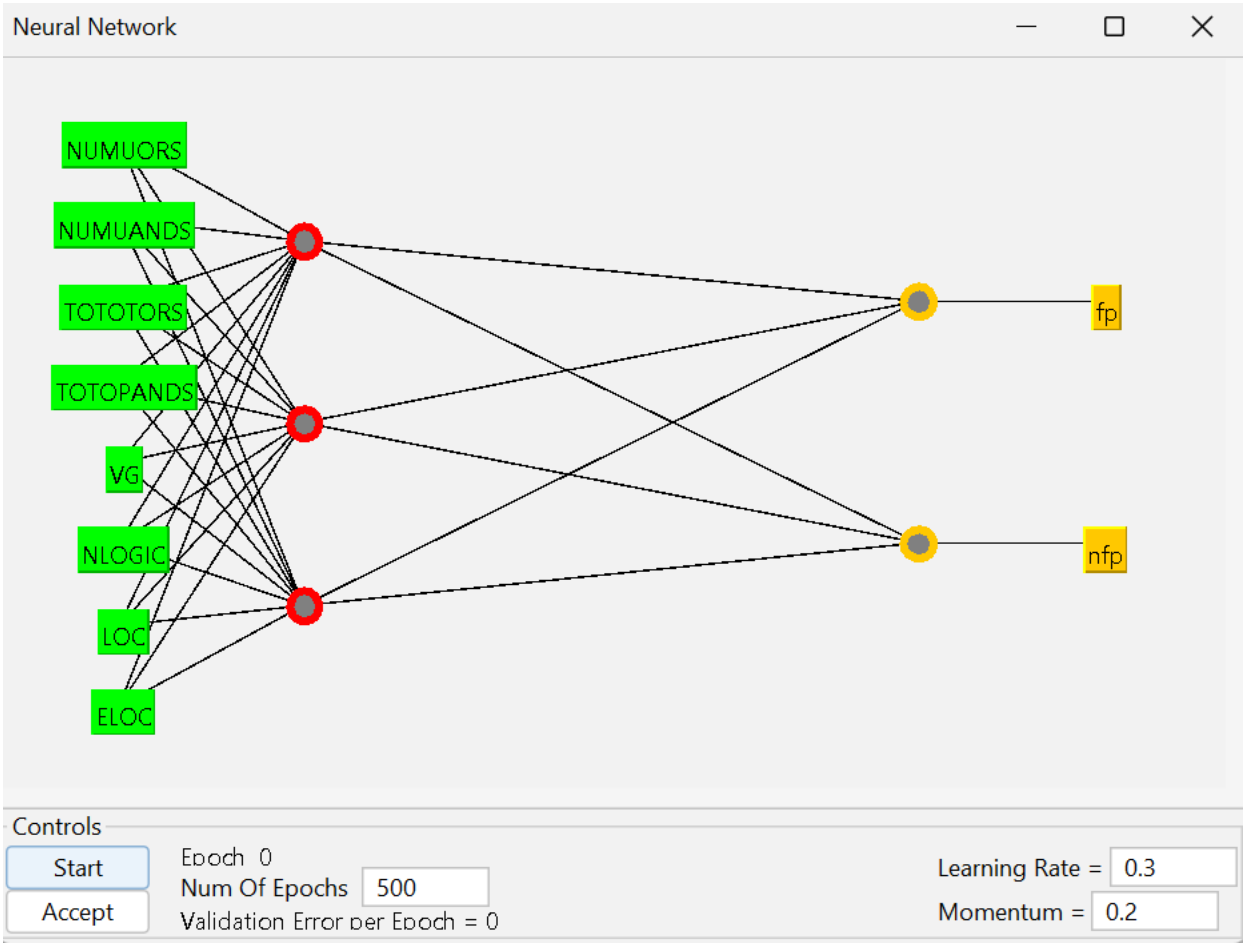
Test Set -> Ratio	Type I %	Type II %
Test Set -> Ratio (1:4)	**14.7%**	**17.9%**
Test Set -> Ratio (1:3.5)	**14.7%**	**17.9%**

Analysis

A “c” value of ~(1:4) will result in the optimal Type I/II error rates. I chose this “c” value because for all the other error rates it is the overall lowest and balanced. When choosing a ratio where the rightmost value is higher, the results are better whereas a larger number on the left results in less optimal results. Note: the optimal error rates can possibly be improved by incrementing the ratio by 0.1 (ex: 3.0, 3.1 ,3.2 ,3.3 , …, 4.0).

Part 2: Multi-Layer Perceptron

MLP GUI = True



MLP Error rates

Ratio	Type I %	Type II %
10-Fold	**8.0%**	**36.4%**
Test Set	**11.8%**	**17.9%**

Comparisons

Logistic Regression

Ratio	Type I %	Type II %
10-Fold -> Ratio (1:4)	**10.2%**	**14.5%**
10-Fold -> Ratio (1:3.5)	**9.5%**	**14.5%**
Test Set -> Ratio (1:4)	**14.7%**	**17.9%**
Test Set -> Ratio (1:3.5)	**14.7%**	**17.9%**

MLP

Ratio	Type I %	Type II %
10-Fold	**8.0%**	**36.4%**
Test Set	**11.8%**	**17.9%**

For Logistic Regression with 10-fold validation, a ratio of 1:3.5 has the lower overall error rates whereas the ratio of 1:4 has low error rates that are more balanced. Of the two ratios when using the test data set the error rate are the same. When comparing logistic regression to MLP the error rates for Logistic Regression are most optimal as MLP has higher rates that are less balanced.

Raw Data

MLP Cross validation

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         161               83.8542 %
Incorrectly Classified Instances        31               16.1458 %
Kappa statistic                          0.5847
Mean absolute error                      0.2105
Root mean squared error                  0.3141
Relative absolute error                 51.3374 %
Root relative squared error             69.4472 %
Total Number of Instances              192

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.636    0.080    0.761      0.636    0.693      0.589    0.930     0.860     fp
                 0.920    0.364    0.863      0.920    0.890      0.589    0.930     0.971     nfp
Weighted Avg.    0.839    0.282    0.834      0.839    0.834      0.589    0.930     0.939

=== Confusion Matrix ===

   a   b   <-- classified as
  35  20 |   a = fp
  11 126 |   b = nfp
```

MLP Test Set

```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances          83               86.4583 %
Incorrectly Classified Instances        13               13.5417 %
Kappa statistic                          0.6823
Mean absolute error                      0.1981
Root mean squared error                  0.301
Relative absolute error                 48.0882 %
Root relative squared error             66.2218 %
Total Number of Instances               96

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.821    0.118    0.742      0.821    0.780      0.684    0.928     0.888     fp
                 0.882    0.179    0.923      0.882    0.902      0.684    0.928     0.957     nfp
Weighted Avg.    0.865    0.161    0.870      0.865    0.866      0.684    0.928     0.937

=== Confusion Matrix ===

  a  b   <-- classified as
 23  5 |  a = fp
  8 60 |  b = nfp
```