Name: Carlos Gross-Martinez

Z Number: 23226341

Course: CAP-6673 Data mining and Machine Learning

Assignment: 1B Prediction Modeling

## Assignment Report

This assignment consisted on implementing two prediction models in order to predict the number of faults on a program based on the attributes of the instance. The main models implemented in this assignment included the “Decision stump” and “Linear Regression” prediction models. Moreover, the linear regression model was built three times implementing different methods in each linear regression instance. By doing this, three different linear regression models were produced with different metrics. The methods used to build the linear regression models included the M5, greedy, and no selection methods. Once all models were trained, there was a total of four models. One of the them was based on the decision stump tree model while the last three were based on the linear regression model. All the aforementioned models were trained with a fit dataset containing eight input attributes, and one output attribute which is the number of faults in a program. In addition, the data table used for the fitting and training of the models consisted of 188 instances. Each model was built employing a ten-cross validation. This approach consisted in segmenting the fit data set in 10 equal parts. Of these 10 equal parts, 9 random segments of the data set were used to train the model while the last segment was used to validate it. This random segment selection for fitting, training, and validation is repeated ten times before the full model is built. It I important to note that if the number for cross validation was 5 folds instead of 10 folds, then the segmentation of the data set will be reduced from 10 to 5 and the number of iterations will also be reduced to 5. After the model was built with the fit dataset, the next step consisted in testing the trained models with a separate test data set. The test data set consisted of nine attributes just like the fit data set, and ninety-four instances. Each model in the assignment was

first trained with the fit data set using 10-fold cross validation and then tested the models with the test data set.

The first model built and tested was the decision stump tree model. The metrics for the training and testing results can be observed in table 1 below.

| | Time to Build Model | Correlation Coefficient | Mean Absolute Error | Root Mean Squared Error |
|---|---|---|---|---|
| *Training* | 0 sec | 0.5974 | 2.0918 | 3.5086 |
| *Testing* | 0 sec | 0.7386 | 1.9464 | 3.8155 |

Table 1: Decision Stump Model Metrics Results

From the results contained in table 1, it can be noted that the time taken to build and test the model was zero seconds. Moreover, it can also be seen that the correlation coefficient increased from the training to the testing of the model while also lightly reducing the mean absolute error. Additionally, it is necessary to mention that the root mean squared error increased slightly from the training to the to the testing of the model. Based on these numbers in can be concluded that the model performed better when it examined the test set, in comparison to when the model was being trained. This can not only be noted in the decrease of the mean absolute error between training and testing the model, but also in the significant increase of the correlation coefficient value which expresses a higher-level accuracy in the prediction of the number of faults of the program. The small increase in the root mean square error denotes that the distance from the predicted points and where the points actually lie in a graph widened slightly from the training of the model to the testing of the model. Finally, it is necessary to mention that once the training was completed for the model, the decision stump

model resulted in a classifier with conditional statement based on the NUMUANDS attribute of the fit data set.

The next model that was built and tested was linear regression with the M5 method. The metrics for the training and testing results can be observed in table 2 below.

| | Time to Build Model | Correlation Coefficient | Mean Absolute Error | Root Mean Squared Error |
|---|---|---|---|---|
| *Training* | 0 sec | 0.8068 | 1.6074 | 2.5852 |
| *Testing* | 0 sec | 0.8273 | 1.7717 | 3.4097 |

Table 2: M5 Method Linear Regression Model Metrics Results

From the results contained in table 2, it can be noted that the time taken to build and test the model was zero seconds. Moreover, it can also be seen that the correlation coefficient increased from the training to the testing of the model while also lightly increasing the mean absolute error. Additionally, it is necessary to mention that the root mean squared error increased slightly from the training to the to the testing of the model. Based on these numbers in can be concluded that the model performed minimally better when it examined the test set, in comparison to when the model was being trained. This can be noted in the miniscule increase of the correlation coefficient value which expresses a higher-level accuracy in the prediction of the number of faults of the program. Nevertheless, it is important to make note of the slight increase on the mean absolute error which translates into a slightly higher probability of error when conducting the fault prediction. Moreover, the significant increase in the root mean square error denotes that the distance from the predicted values and where the real values actually lie in a graph widened slightly from the training of the model to the testing of the model. Finally, it is necessary to mention that once the training was completed for the

model, the linear regression model resulted in an equation which has the fault attribute as the dependent variable, while the other eight attributes were implemented as linear independent variables multiplied by coefficients.

The subsequent model that was built and tested was linear regression with the greedy method. The metrics for the training and testing results can be observed in table 3 below.

| | Time to Build Model | Correlation Coefficient | Mean Absolute Error | Root Mean Squared Error |
|---|---|---|---|---|
| *Training* | 0 sec | 0.8081 | 1.6095 | 2.5801 |
| *Testing* | 0 sec | 0.8312 | 1.7797 | 3.3504 |

Table 3: Greedy Method Linear Regression Model Metrics Results

From the results contained in table 3, it can be noted that the time taken to build and test the model was zero seconds. Moreover, it can also be seen that the correlation coefficient increased from the training to the testing of the model while also lightly increasing the mean absolute error. Additionally, it is necessary to mention that the root mean squared error increased slightly from the training to the to the testing of the model. Based on these numbers in can be concluded that the model performed minimally better when it examined the test set, in comparison to when the model was being trained. This can be noted in the small increase of the correlation coefficient value which expresses a higher-level accuracy in the prediction of the number of faults of the program. Nevertheless, it is important to make note of the slight increase on the mean absolute error which translates into a slightly higher probability of error when conducting the fault prediction. Moreover, the significant increase in the root mean square error denotes that the distance from the predicted values and where the real values actually lie in a graph widened slightly from the training of the model to the testing of the

model. Finally, it is necessary to mention that once the training was completed for the model, the linear regression model resulted in an equation which has the fault attribute as the dependent variable. Moreover, only six out of the eight attributes were implemented as linear independent variables multiplied by coefficients.

The last model which was built and tested was linear regression with the no attribute method. The metrics for the training and testing results can be observed in table 4 below.

| | Time to Build Model | Correlation Coefficient | Mean Absolute Error | Root Mean Squared Error |
|---|---|---|---|---|
| *Training* | 0 sec | 0.8101 | 1.5864 | 2.5584 |
| *Testing* | 0 sec | 0.8273 | 1.7717 | 3.4097 |

Table 4: No Attribute Method Linear Regression Model Metrics Results

From the results contained in table 4, it can be noted that the time taken to build and test the model was zero seconds. Moreover, it can also be seen that the correlation coefficient increased from the training to the testing of the model while also lightly increasing the mean absolute error. Additionally, it is necessary to mention that the root mean squared error also increased from the training to the to the testing of the model. Based on these numbers in can be concluded that the model performed minimally better when it examined the test set, in comparison to when the model was being trained. This can be noted in the miniscule increase of the correlation coefficient value which expresses a higher-level accuracy in the prediction of the number of faults of the program. Nevertheless, it is important to make note of the slight increase on the mean absolute error which translates into a slightly higher probability of error when conducting the fault prediction. Moreover, the significant increase in the root mean square error denotes that the distance from the predicted values and where the real values

actually lie in a graph widened slightly from the training of the model to the testing of the model. Finally, it is necessary to mention that once the training was completed for the model, the linear regression model resulted in an equation which has the fault attribute as the dependent variable, while the other eight attributes were implemented as linear independent variables multiplied by coefficients.

Based on the data gathered from all the models, there were two similarities which remained apparent between the training and testing of all the models. The similarities included that all the models took 0 seconds to build. There was no difference in time cost when building any of the models implemented in the exercise. Moreover, all models also experienced an increase in the root square mean value. In addition, it is important emphasize that from all the models available, the decision stump model has the lowest magnitude of increase in the root square mean value.

With the similarities between the models covered, the next step consists in identifying the multiple differences that exists between all the models and methods. One of the main differences between the two models implemented in the assignment is how the resulting classifier is derived. From the decision stump tree, it can be seen that the resultant classifier can be simply defined as a conditional statement which utilizes one specific attribute and a specific threshold to conduct the fault prediction. On the other hand, on the linear regression methods, the classifier was a linear equation that took the attributes of the data as the independent variables of the equation while making the fault attribute the dependent output variable. Furthermore, it is critical to mention that when the decision stump tree method was the least accurate model when compared to the linear regression models since it had the lowest correlation coefficient value when compared to all of the other models. In addition, the decision stump model also had the highest absolute mean error value of all the models. From this information, it can be concluded that any of the linear regression methodologies used in this assignment will have decision a better performance in accurately predicting the number of faults in a program when compared against the decision stump model. Nevertheless, it is

imperative to mention that although the linear regression methods had better performance in the prediction of the faults, it was the decision stump tree which had the greater magnitude of improvement predicting the faults when the model was examined with the test data set. This model not only had a bigger magnitude in the increase of value of the correlation coefficient, but it was the only model that actually decrease its mean absolute error when confronted with the test data set.

With the comparison of the two main models employed in the assignment completed, the next phase of the report compares and contrast the performance of the three different methods used for the different linear regression models. Table 5 below compares the correlation coefficient value from all the models.

| | **Decision Stump Tree** | **M5 Linear Regression** | **Greedy Linear Regression** | **No Attribute Linear Regression** |
|---|---|---|---|---|
| *Correlation Coefficient Training* | 0.5974 | 0.8068 | 0.8081 | 0.8101 |
| *Correlation Coefficient Testing* | 0.7386 | 0.8273 | 0.8312 | 0.8273 |

Table 5: Training and Testing Correlation Coefficient Metrics Results

From table 5 above, it can be inferred that the linear regression model which used a greedy methodology obtained the highest correlation coefficient value from all the models. With that information it can be concluded that the greedy linear regression model will have the highest accuracy when predicting the fault number of a program when compared to all other methods and models.

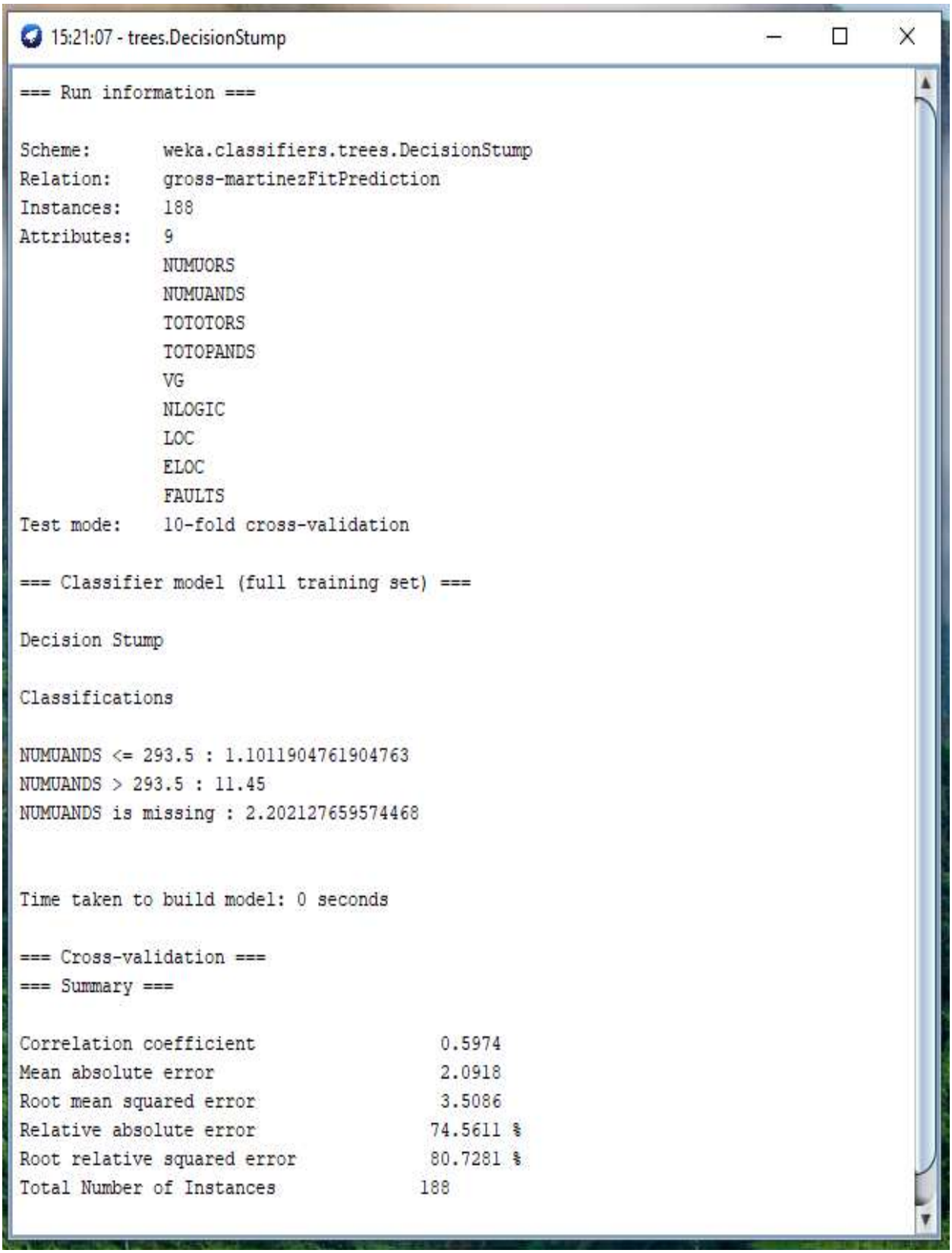
| | Decision Stump Tree | M5 Linear Regression | Greedy Linear Regression | No Attribute Linear Regression |
|---|---|---|---|---|
| *Mean Absolute Error Training* | 2.0918 | 1.6074 | 1.6095 | 1.5864 |
| *Mean Absolute Error Testing* | 1.9464 | 1.7717 | 1.7797 | 1.7717 |

Table 6: Training and Testing Mean Absolute Error Metrics Results

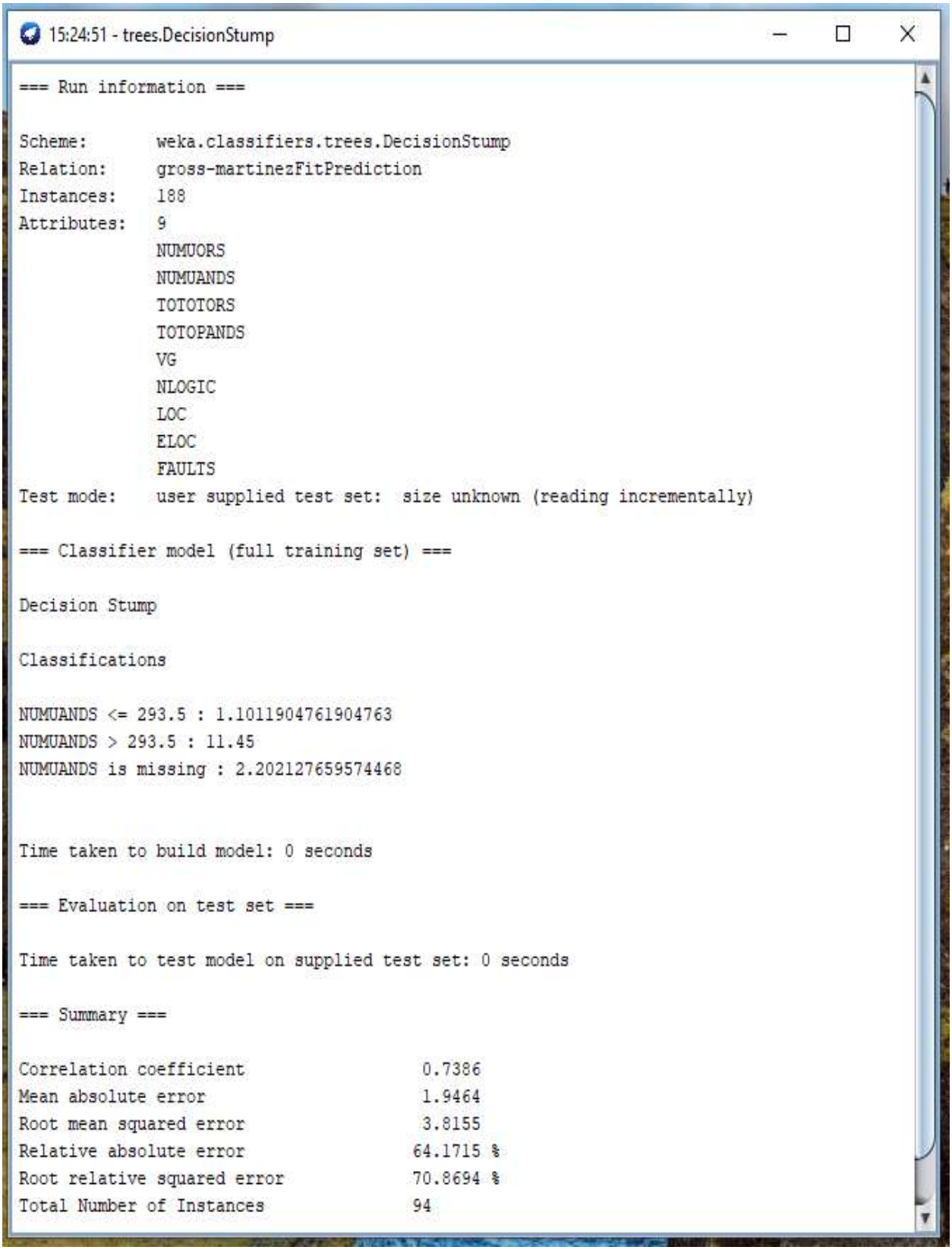
From table 6 above, it can be inferred that the linear regression model which used the M5 and No attribute methodologies obtained the lowest mean absolute error value from all the models. With that information it can be concluded that the no attribute and M5 methodologies of the linear regression model will have the least amount of error when predicting the number of faults in the program when compared to all other methods and models.

Based on all the data gathered in this assignment, it can be easily concluded that the decision stump tree was the worst model built for the fault prediction of the provided data sets. Based on the results of the training and testing of the model, the decision stump tree had the highest mean absolute error value with the lowest coefficient correlation value. On the other hand, the model which had the best performance for the fault prediction with the provided data set is the linear regression model implementing a greedy methodology. This model did not have the lowest mean absolute error value when compared to its counterparts M5 and No attribute methods. Nevertheless, the difference in the mean absolute error value between the greedy method and the other two linear regression methods was so small, that it allowed for the greedy method to remain the one scheme with the highest performance of all the methods and models. This is also because the greedy linear regression method also has the highest correlation coefficient value when compared to all the other available models and methods. All the results from the all the models and methods from the Weka tool can be seen in the subsequent pages below.

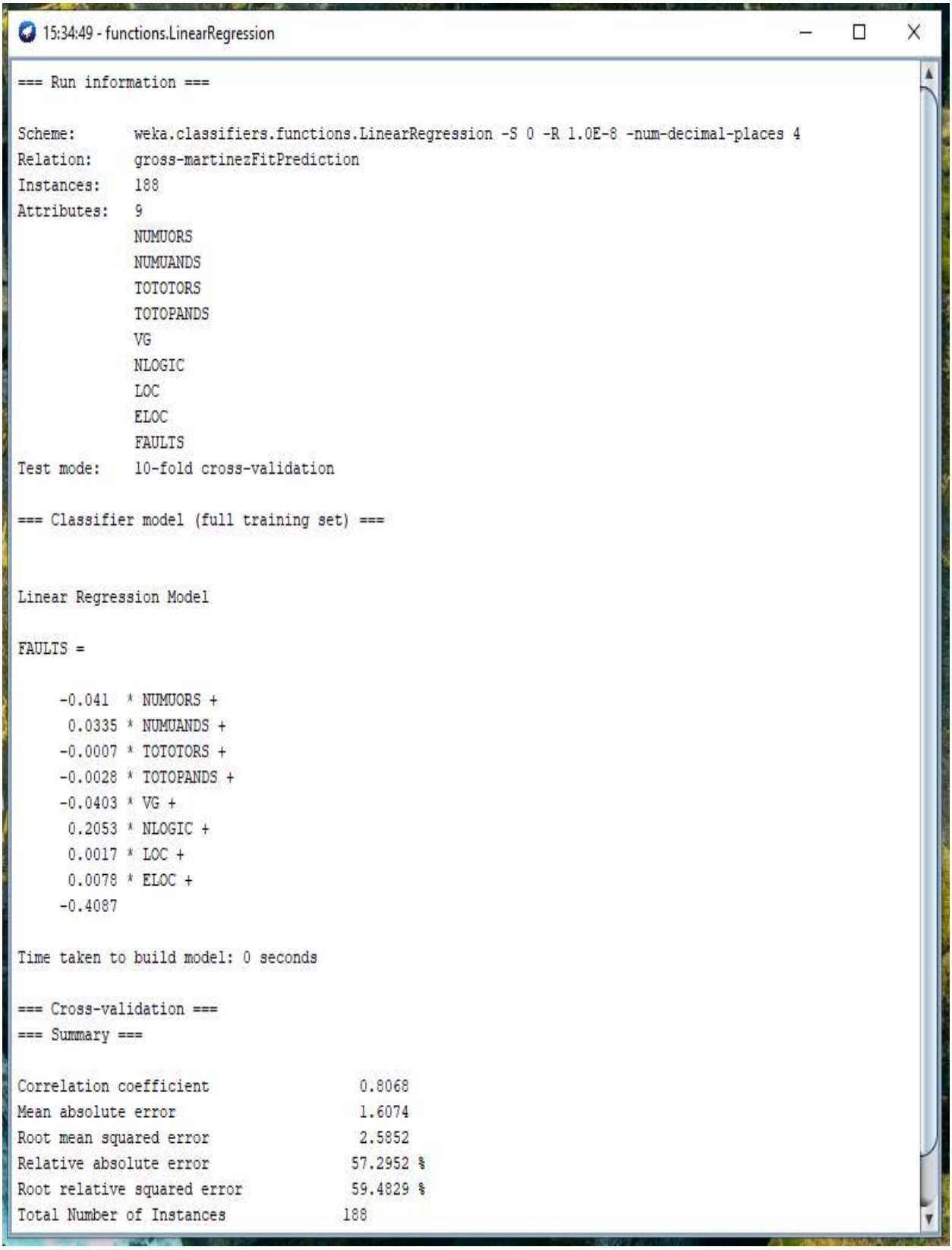
# Decision Stump Model Training Output



# Decision Stump Model Test Output



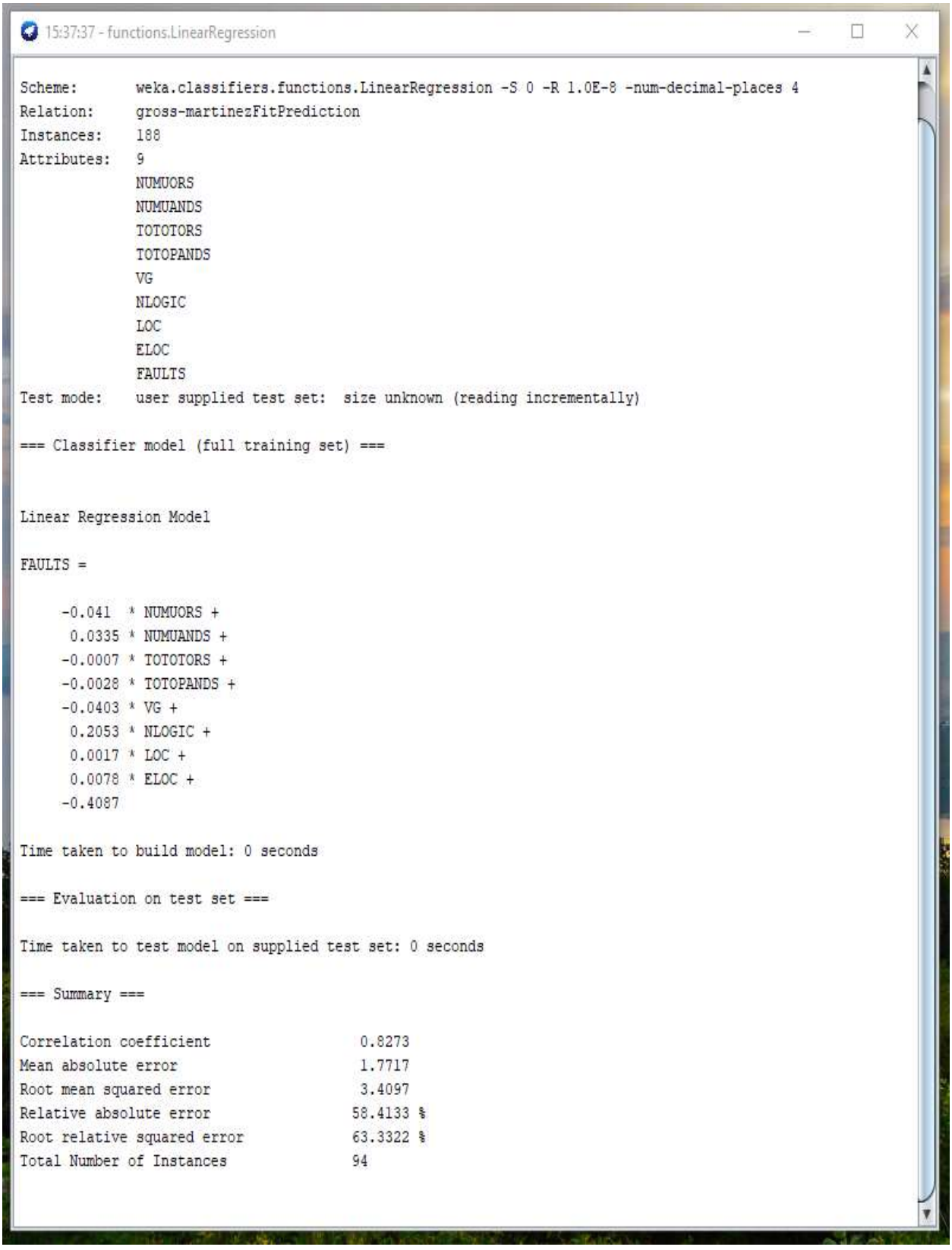
# M5-Method Linear Regression Model Training Output



15:34:49 - functions.LinearRegression

```
=== Run information ===

Scheme:       weka.classifiers.functions.LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4
Relation:     gross-martinezFitPrediction
Instances:    188
Attributes:   9
              NUMUORS
              NUMUANDS
              TOTOTORS
              TOTOPANDS
              VG
              NLOGIC
              LOC
              ELOC
              FAULTS
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===


Linear Regression Model

FAULTS =

     -0.041  * NUMUORS +
      0.0335 * NUMUANDS +
     -0.0007 * TOTOTORS +
     -0.0028 * TOTOPANDS +
     -0.0403 * VG +
      0.2053 * NLOGIC +
      0.0017 * LOC +
      0.0078 * ELOC +
     -0.4087

Time taken to build model: 0 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.8068
Mean absolute error                      1.6074
Root mean squared error                  2.5852
Relative absolute error                 57.2952 %
Root relative squared error             59.4829 %
Total Number of Instances              188
```

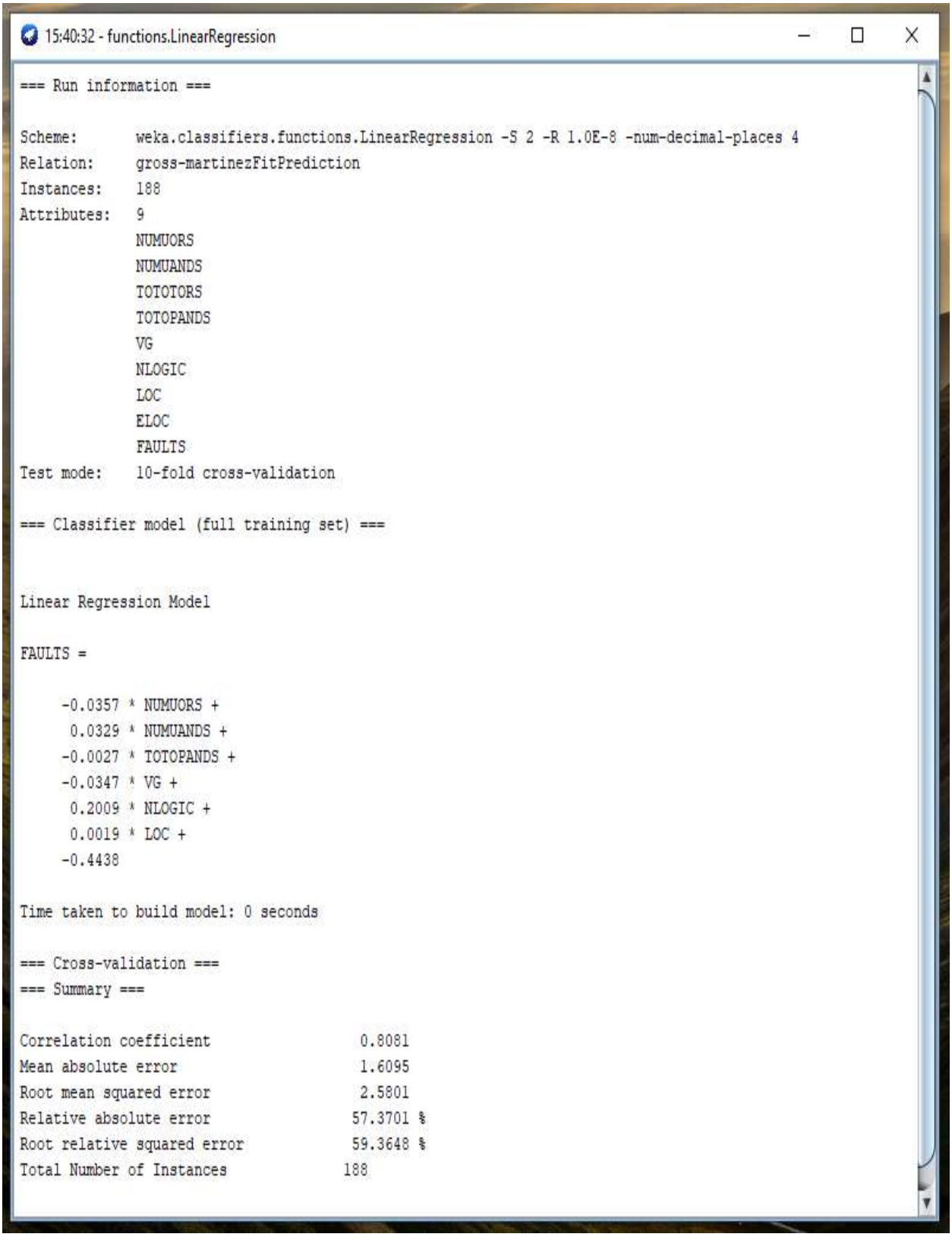
# M5-Method Linear Regression Model Test Output



15:37:37 - functions.LinearRegression

```
Scheme:       weka.classifiers.functions.LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4
Relation:     gross-martinezFitPrediction
Instances:    188
Attributes:   9
              NUMUORS
              NUMUANDS
              TOTOTORS
              TOTOPANDS
              VG
              NLOGIC
              LOC
              ELOC
              FAULTS
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===


Linear Regression Model

FAULTS =

     -0.041  * NUMUORS +
      0.0335 * NUMUANDS +
     -0.0007 * TOTOTORS +
     -0.0028 * TOTOPANDS +
     -0.0403 * VG +
      0.2053 * NLOGIC +
      0.0017 * LOC +
      0.0078 * ELOC +
     -0.4087

Time taken to build model: 0 seconds

=== Evaluation on test set ===

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

=== Summary ===

Correlation coefficient                  0.8273
Mean absolute error                      1.7717
Root mean squared error                  3.4097
Relative absolute error                 58.4133 %
Root relative squared error             63.3322 %
Total Number of Instances               94
```

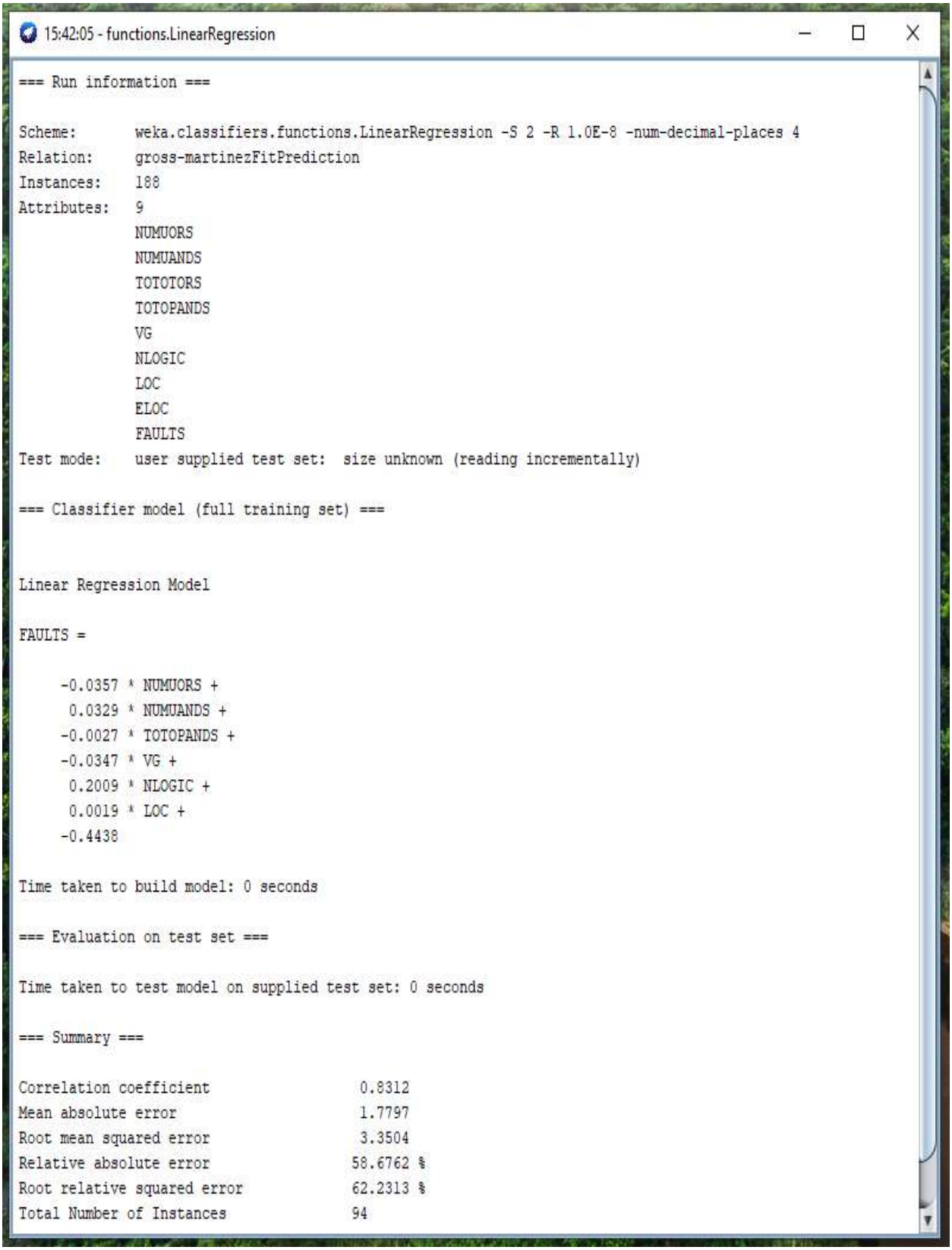
# Greedy-Method Linear Regression Model Training Output



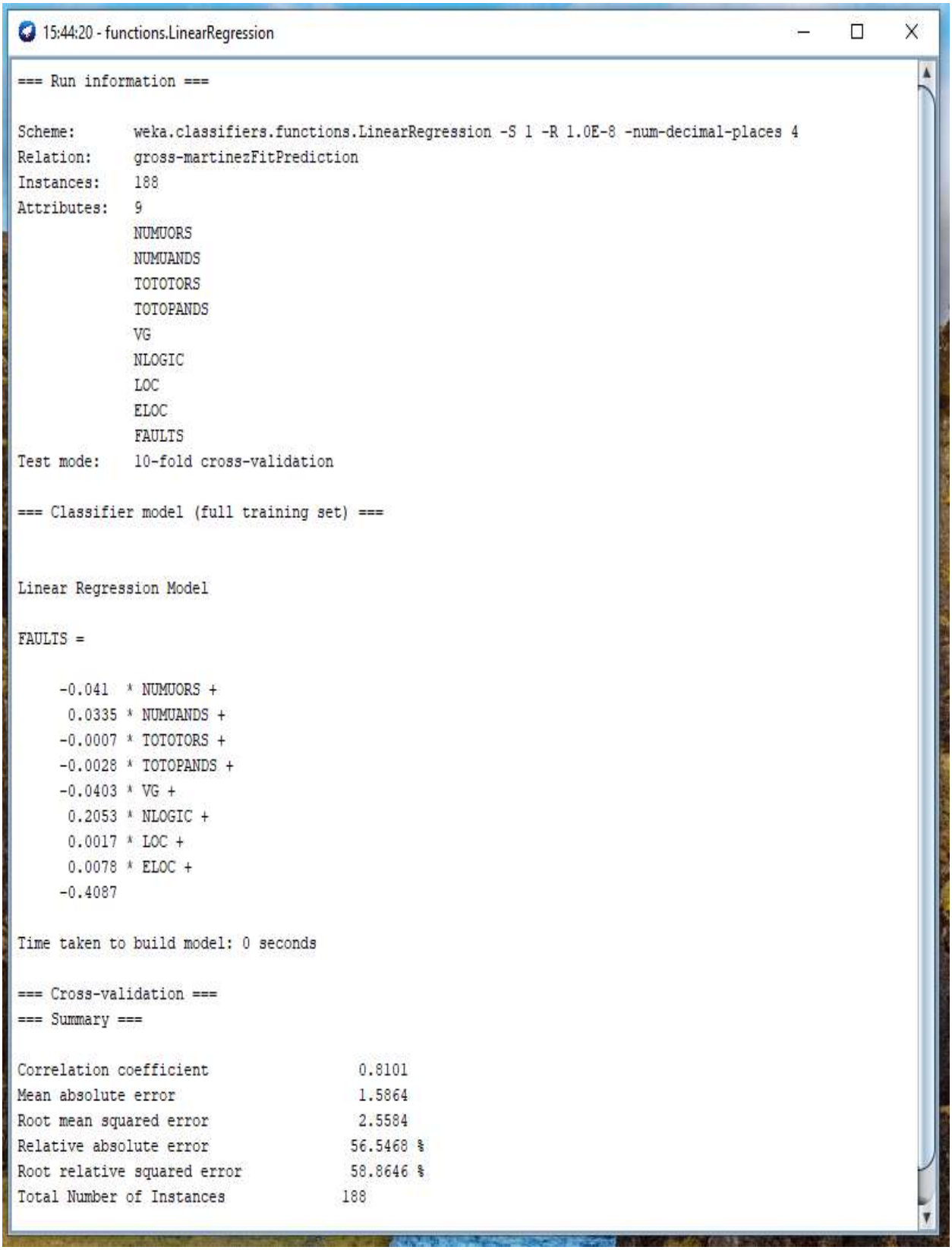
15:40:32 - functions.LinearRegression

```
=== Run information ===

Scheme:       weka.classifiers.functions.LinearRegression -S 2 -R 1.0E-8 -num-decimal-places 4
Relation:     gross-martinezFitPrediction
Instances:    188
Attributes:   9
              NUMUORS
              NUMUANDS
              TOTOTORS
              TOTOPANDS
              VG
              NLOGIC
              LOC
              ELOC
              FAULTS
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===


Linear Regression Model

FAULTS =

     -0.0357 * NUMUORS +
      0.0329 * NUMUANDS +
     -0.0027 * TOTOPANDS +
     -0.0347 * VG +
      0.2009 * NLOGIC +
      0.0019 * LOC +
     -0.4438

Time taken to build model: 0 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.8081
Mean absolute error                      1.6095
Root mean squared error                  2.5801
Relative absolute error                 57.3701 %
Root relative squared error             59.3648 %
Total Number of Instances              188
```

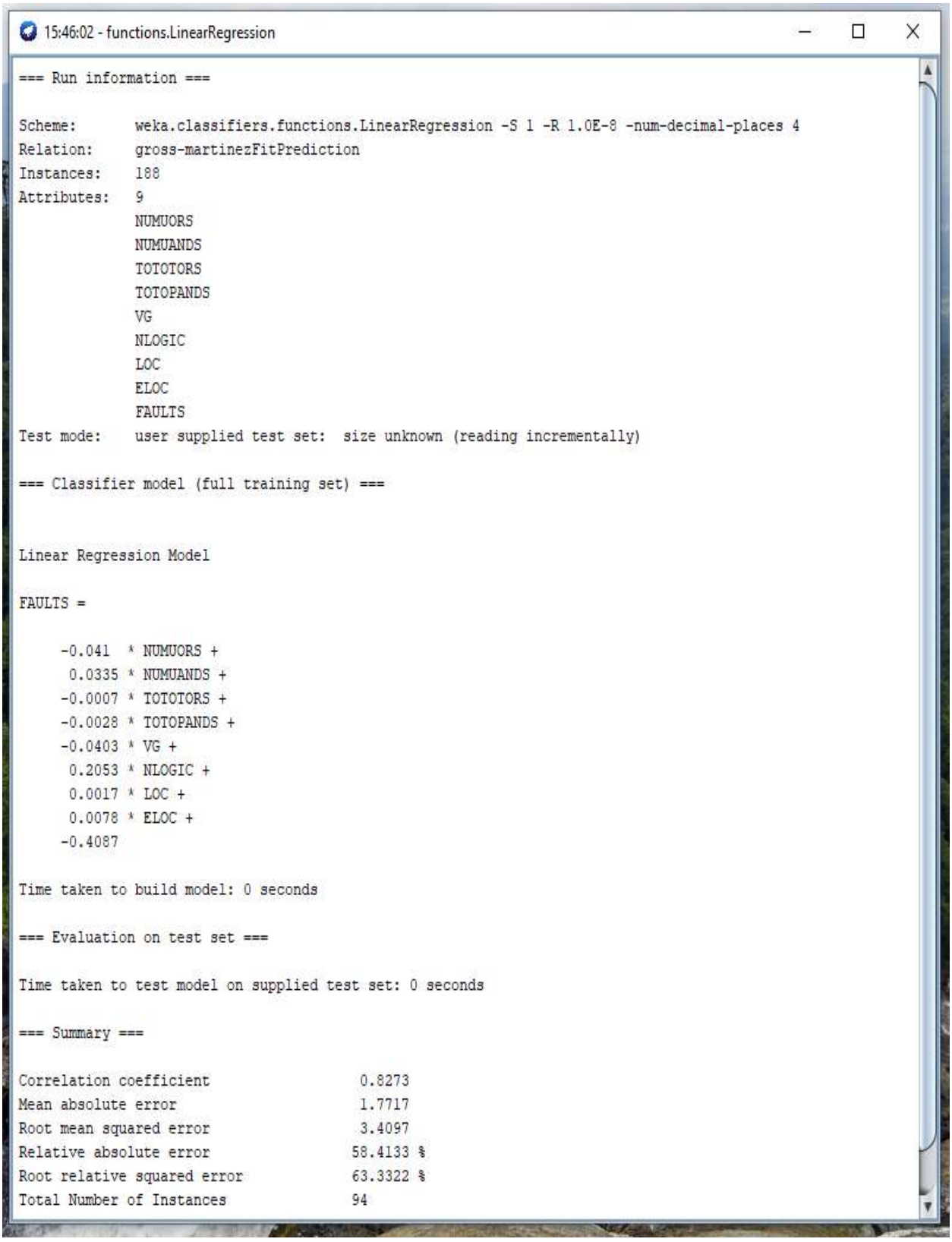
# Greedy-Method Linear Regression Model Test Output



# No Attribute Method Linear Regression Model Training Output



# No Attribute Method Linear Regression Model Test Output



15:46:02 - functions.LinearRegression

```
=== Run information ===

Scheme:       weka.classifiers.functions.LinearRegression -S 1 -R 1.0E-8 -num-decimal-places 4
Relation:     gross-martinezFitPrediction
Instances:    188
Attributes:   9
              NUMUORS
              NUMUANDS
              TOTOTORS
              TOTOPANDS
              VG
              NLOGIC
              LOC
              ELOC
              FAULTS
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===


Linear Regression Model

FAULTS =

     -0.041  * NUMUORS +
      0.0335 * NUMUANDS +
     -0.0007 * TOTOTORS +
     -0.0028 * TOTOPANDS +
     -0.0403 * VG +
      0.2053 * NLOGIC +
      0.0017 * LOC +
      0.0078 * ELOC +
     -0.4087

Time taken to build model: 0 seconds

=== Evaluation on test set ===

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

=== Summary ===

Correlation coefficient                  0.8273
Mean absolute error                      1.7717
Root mean squared error                  3.4097
Relative absolute error                 58.4133 %
Root relative squared error             63.3322 %
Total Number of Instances               94
```