CAP6673 – Data Mining and Machine Learning

Module Order Modeling

Written by:

Christopher Foley
Z15092976

Academic Year: 2016-2017

Table of Contents

1. Introduction

Software quality control is customarily a reactive business, modules are inspected after flaws are found. The purpose of the assignments in this class are to explore models to predict which software modules are the most likely to contain flaws. Furthermore, in this exercise we continue the work of Khoshgoftar and Allen [1,2] and compare predicted module outcomes to actual outcomes while ordering for maximum impact.

Although the goal of error free software is laudable, large complicated systems, will contain errors. Software managers need a tool to determine which modules may be fixed making the largest impact on the final product and which modules may be fixed making the least impact. The data analysis is presented as an Alberg/Praeto chart.

The data was prepared and analyzed using Libre Office Calc, a spreadsheet program. The data, originally from a CCCS system analyzed in [1] was imported and column headers added. Then data columns were added to permit module orders to be recorded for each iteration. When module order could not be determined (predictions indicate the same number of faults) order was determined by the initial original module number. Within each model the order, sum of faults, percent of faults, Average Absolute Error and Average Relative Error were computed. Model reliability was determined by comparing predicted sum of faults to actual sum of faults.

Each model was applied to the Fit and Test data sets provided and the results mapped.

The following data fields were provided and used:

1 Khosgoftar, T. M. and Allen, E B. (1999) , *A Comparative Study of Ordering and Classification,*Empirical Software Engineering (1999) 4: 159. doi:10.1023/A:1009876418873 (text provided by author)

2 Khosgoftar, T. M. and Allen, E. B. (2003), Ordering fault-prone software modules," Software Quality Journal, vol. 11, no. 1, pp. 19–37, Mar. 2003, I

Attribute	Description
NUMORS	Number of Unique operators
NUMANDS	Number of Unique operands
TOTOTORS	Total number of operators
TOTOPANDS	Total number of operands
VG	McCabe Complexity Complex
NLOGIC	Number of logical ooperators
LOC	Number of lines of code
ELOC	Executable lines of code
FAULTS	Known number of faults

2. Models Used

When analyzing the data, following the examples of Khoshgotaar and Allen, The following calculations were used:

(a) The sum of the actual number of faults above the cutoff point: $G(c) = \sum_{i:\,above\,cutoff} F_i$

(b) The sum of the predicted faults above the cutoff point: $\hat{G}(c) = \sum_{i:\,above\,cutoff} \hat{F}_i$

The cutoff point *c* represents a percentage of the modules ordered based upon the decending number of faults found, with the original module number as a tie factor when modules had identical fault counts. In the analysis 1-c is used in the tables to indicate how many faults lie below the threshold module percentage count.

The predictions were then validated by calculating the average absolute error, AAE, and average relative error, ARE. Where AAE and ARE are calculated as follows:

$$AAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

$$AAE = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{y_i - \hat{y}_i}{y_i + 1}\right|$$

where y_i and $\hat{y}_i$ are the actual and predicted rankings respectively and n is the number of data samples. As expected lower average errors are best.

The actual data analysis tested the Fit and Test data sets using Libre Office Calc using a Linear Regression model with an M5 method of attribute selection and a Greedy method of attribute selection.

1 Linear Regression with M5 method of attribute selection

A M5 models combines a decision tree with the possibility of linear regression at each node. In the first assignment, where we were to predict the number of faults, the following model was created:

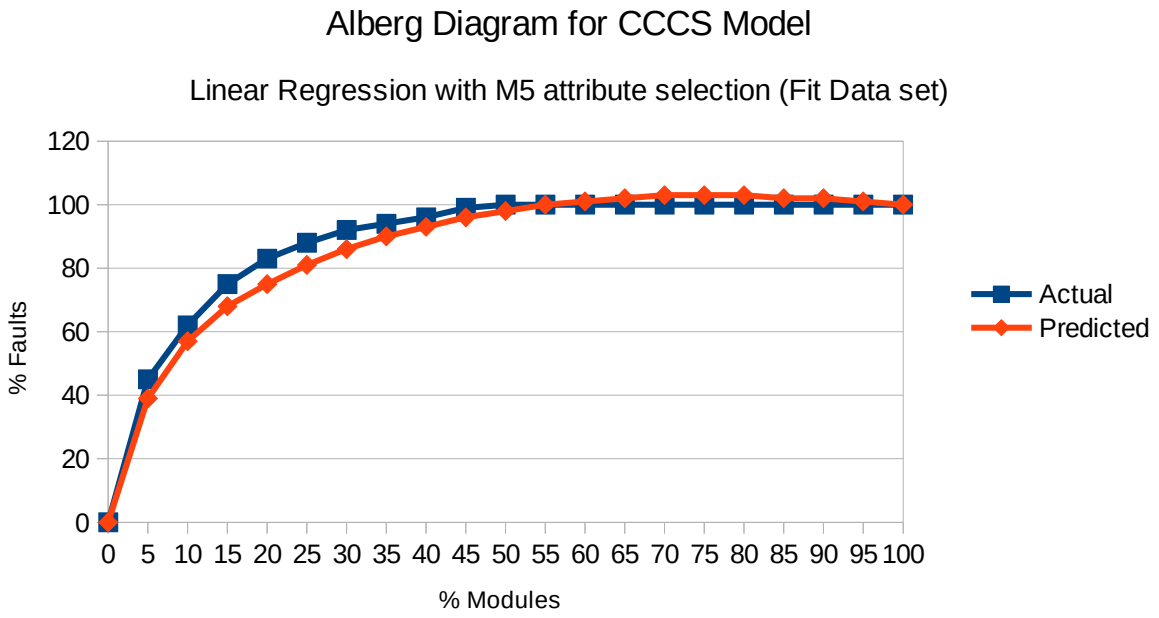
```
FAULTS =
      -0.0516 * NUMUORS +
       0.0341 * NUMUANDS +
      -0.0027 * TOTOTORS +
      -0.0372 * VG +
       0.2119 * NLOGIC +
       0.0018 * LOC +
       0.005  * ELOC +
      -0.3091
```

This model was applied to the data in the spreadsheet where each column was an attribute. It should be noted that the TOTOPANDS attribute was not used and the FAULTS was obviously ignored.

The Fit data, as analyzed with the M5 method of attribute selection, may be summarized in the following table and chart:

1-c	$G(c)/F_{tot}$	$\hat{G}(c)/F_{tot}$	$\phi(c)$
0.60	1.0000	1.0100	1.0100
0.55	1.0000	1.0046	1.0046
0.50	1.0000	0.9886	0.9886
0.45	0.9906	0.9673	0.9765
0.40	0.9672	0.9342	0.9659
0.35	0.9461	0.9005	0.9518
0.30	0.9251	0.8635	0.9334
0.25	0.8852	0.8144	0.9200
0.20	0.8314	0.7574	0.9110
0.15	0.7541	0.6837	0.9066

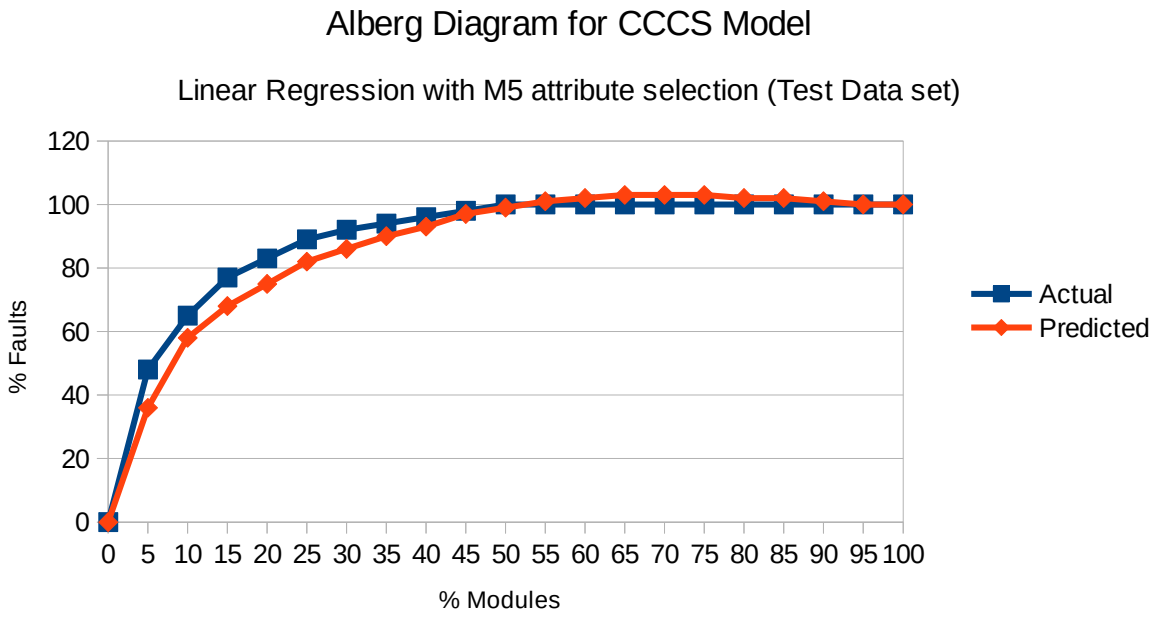
1-c	$G(c)/F_{tot}$	$\hat{G}(c)/F_{tot}$	$\phi(c)$
0.10	0.6300	0.5728	0.9092
0.05	0.4520	0.3931	0.8697



Of note is the observation that 50% of the modules contained 100% of the faults. The predicted rates were close up to the 50% then diverged slightly when the number of faults fluctuated on the interval (-2,2) when the actual number of faults dropped to 1. The Fit model using the M5 method of attribute selectiohad an AAE of 1.444 and an ARE of 0.6424.

The test data for the same had the following results:

1-c	$G(c)/F_{tot}$	$\hat{G}(c)/F_{tot}$	$\phi(c)$
0.60	1.0000	1.0100	1.0100
0.55	1.0000	1.0100	1.0100
0.50	1.0000	0.9970	0.9970
0.45	0.9875	0.9874	0.9989
0.40	0.9668	0.9430	0.9754
0.35	0.9461	0.9034	0.9549
0.30	0.9294	0.8688	0.9348
0.25	0.8921	0.8210	0.9203
0.20	0.8381	0.7554	0.9013
0.15	0.7718	0.6857	0.8884
0.10	0.6597	0.5476	0.8301
0.05	0.4689	0.3965	0.8450



Again a minority of modules contained all the errors and clearly, less than 10% of the modules contained 50% of the errors. The Test data had an AAE of 1.8392 and an ARE of 0.5939. The AAE and ARE would be smaller if errors were rounded and predictions less than zero were marked as 0.

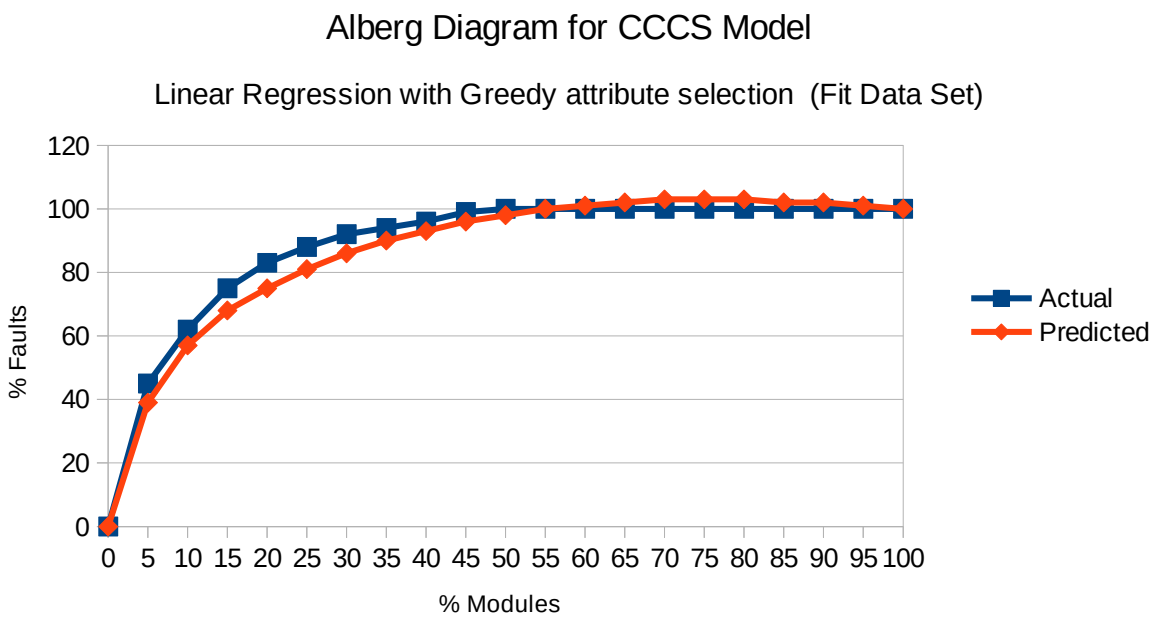
2 Linear Regression with Greedy Method of attribute selection

A greedy model will create a model and remove a rule from the rule tree. If the resulting model results in a better model, the rule is removed and the next rule is reviewed. When all rules have been examined, duplicates are removed and the resulting model used. The FIT model produced the folllowing model:

```
FAULTS =
      -0.0482 * NUMUORS +
       0.0336 * NUMUANDS +
      -0.0021 * TOTOTORS +
      -0.0337 * VG +
       0.2088 * NLOGIC +
       0.0019 * LOC +
      -0.3255
```

The Fit data may be summarized in the following table and chart:

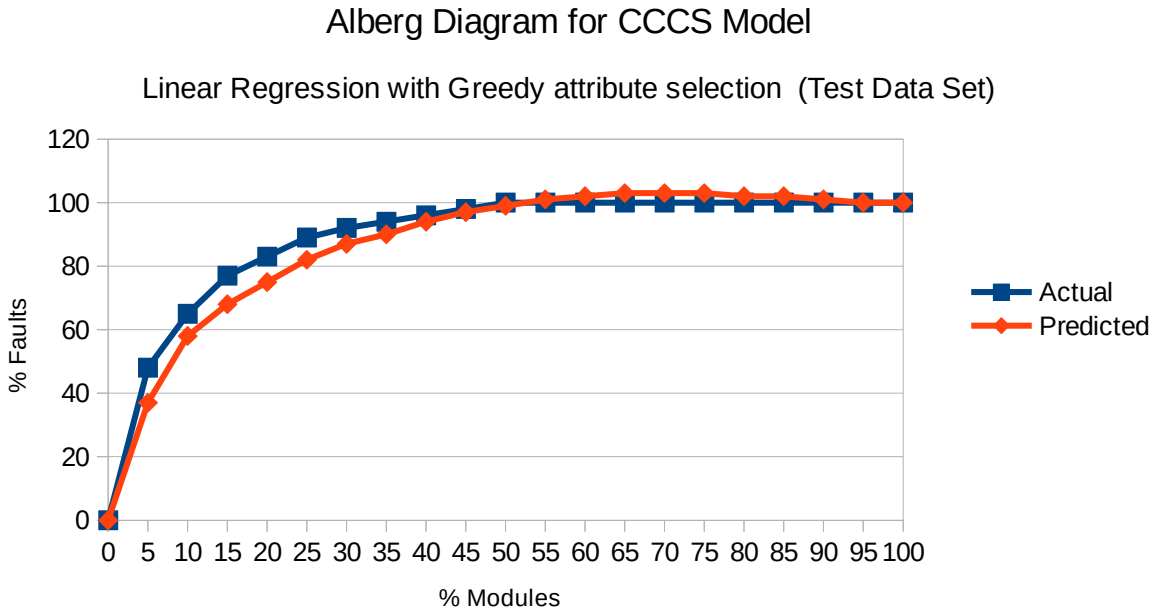
1-c	$G(c)/F_{tot}$	$\hat{G}(c)/F_{tot}$	$\phi(c)$
0.60	1.0000	1.0100	1.0100
0.55	1.0000	1.0100	1.0100
0.50	1.0000	0.9970	0.9970
0.45	0.9906	0.9874	0.9989
0.40	0.9672	0.9430	0.9754
0.35	0.9461	0.9034	0.9549
0.30	0.9251	0.8688	0.9348
0.25	0.8852	0.8210	0.9203
0.20	0.8314	0.7554	0.9013
0.15	0.7541	0.6857	0.8884
0.10	0.6300	0.5476	0.8301
0.05	0.4520	0.3965	0.8455



Again, of note is the observation that 50% of the modules contained 100% of the faults. The predicted rates were close up to the 50% then diverged slightly when the number of faults fluctuated on the interval (-2,2) when the actual number of faults dropped to 1. The Fit model had an AAE of 1.4470 and an ARE of 0.6396.

The Test looked like this:

1-c	$G(c)/F_{tot}$	$\hat{G}(c)/F_{tot}$	$\phi(c)$
0.60	1.0000	1.0100	1.0100
0.55	1.0000	1.0100	1.0100
0.50	1.0000	1.0000	1.0000
0.45	0.9875	0.9842	0.9967
0.40	0.9668	0.9534	0.9861
0.35	0.9461	0.9141	0.9662
0.30	0.9294	0.8718	0.9380
0.25	0.8921	0.8241	0.9237
0.20	0.8381	0.7583	0.9047
0.15	0.7718	0.6878	0.8911
0.10	0.6597	0.5870	0.8898
0.05	0.4689	0.3707	0.7906



The Test data sorted with the Greedy Attribute selection had an AAE of 1.8393 and a ARE of 0.5971.

3. Conclusions

In the Software Development Example as shown above, Module Order Modeling (MOM) is an excellent tool for managers to predict which modules are likely to contain errors. In Exercise 1 where the models were built, we determined how to determine which modules were the most complex, this tool allows us to determine which complex modules are most likely to contain errors.

Care should be taken when using the models because saying that a module is statistically likely to contain errors is not saying that it contains errors.

4. Appendix A

The following is a sample of the Fit data set used as input to the Calc program:

```
22,85,203,174,9,0,362,40,0,nfp
21,87,186,165,5,0,379,32,0,nfp
30,107,405,306,25,0,756,99,0,nfp
6,5,19,6,2,0,160,9,0,nfp
21,47,168,148,7,0,352,29,0,nfp
28,38,161,114,10,3,375,40,0,nfp
27,218,1522,1328,114,0,1026,310,0,nfp
21,78,156,135,5,0,300,27,0,nfp
6,13,55,38,1,0,291,21,0,nfp
7,6,19,8,2,0,135,9,0,nfp
22,83,168,145,6,0,317,30,0,nfp
```