CAP6673 Justin Johnson
Dr Taghi Khoshgoftaar Z23136514

Assignment 4
Module Order Modelling (MOM)

Introduction)

The Module Order Modelling (MOM) methodology is applied to fit and test data sets which describe software modules. The labelled data sets contain 9 attributes, where the 9^{th} attribute (label) provides the total number of faults for the given module. The training data contains 188 instances and a total of 427 faults among all instances. The test data contains 94 instances and a total of 241 faults. The goal of MOM is to order a set of software modules based on their predicted number of faults. Development managers are then able to select a percentage of the most fault prone modules for additional review. This is more flexible than traditional classification of modules as fault prone vs non-fault prone, as the team can then focus on the most fault prone modules, and select a cutoff point based on their available resources.

The labelled data sets are used to evaluate the performance of MOM. First, software modules are ordered by their actual number of faults, creating a perfect ranking. Cutoff points are selected between 5% and 50%, where a cutoff of 5% selects the top 5% of software modules which have the greatest number of faults. For each cutoff point, the number of actual faults captured in the given range is compared to the total number of faults contained in the data set, resulting in a percentage of total faults captured.

Next, modules are ordered by their predicted rank, where each module's predicted number of faults is calculated using linear regression models that were learned during Assignment 1. The predicted number of faults are only used to order the modules and create the predicted ranking. The same cutoff points are then applied to the predicted ranking. For each cutoff point, the number of actual faults captured in the given range is compared to the total number of faults in the data set.

The MOM model is evaluated by comparing the total faults captured with the predicted ranking to the total faults captured with the perfect ranking at various cutoff points. Tables and diagrams are provided to compare the perfect ranking results to the predicted ranking results.

The two linear regression models which will be used to predict module faults and determine predicted ranking are:

M5 Model: Linear Regression Model with M5 Method of Attribute Selection
*FAULTS = - 0.0516 * NUMUORS + 0.0341 * NUMUANDS - 0.0027 * TOTOTORS - 0.0372 * VG + 0.2119 * NLOGIC + 0.0018 * LOC + 0.005 * ELOC - 0.3091*

Greedy Model: Linear Regression Model with Greedy Method of Attribute Selection
*FAULTS = - 0.0482 * NUMUORS + 0.0336 * NUMUANDS - 0.0021 * TOTOTORS - 0.0337 * VG + 0.2088 * NLOGIC + 0.0019 * LOC - 0.3255*

These linear regression models will be referred to as the M5 Model and Greedy Model moving forward.

I) Determining Perfect Rank

The perfect rank for each data set is created by ordering the module instances by each module's number of faults, from greatest to least. An excel spreadsheet is used to load the data sets and order the data. Cutoff points ranging from top 5% to top 50% are selected, and for each cutoff the number of actual faults within the cutoff is compared to the total number of faults in the data set.

II) Determining Predicted Rank with M5 Model

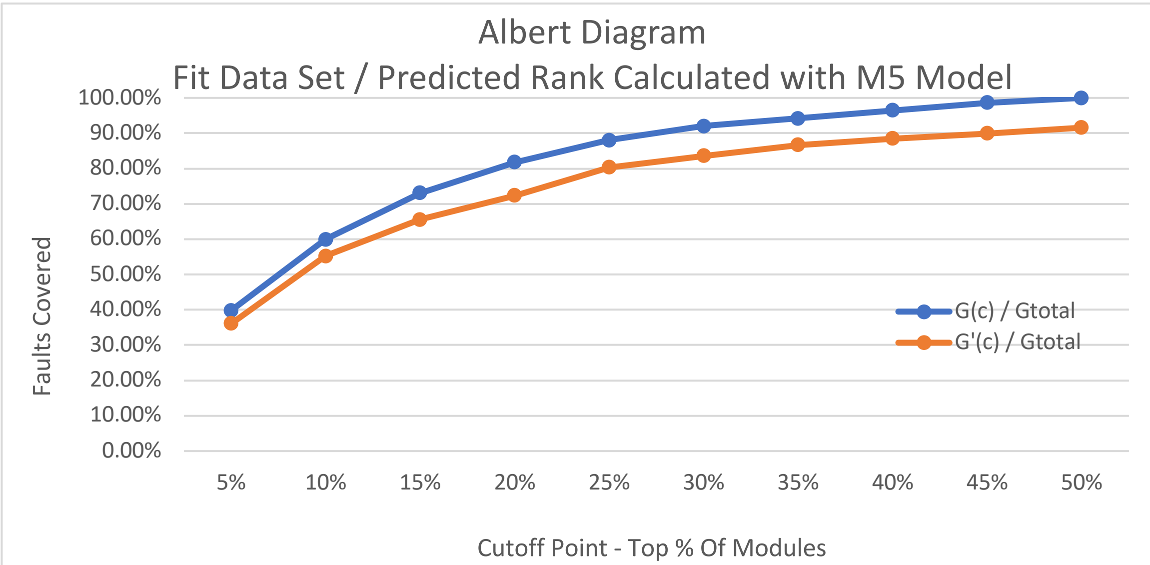
The predicted number of faults is calculated for fit and test data sets using the M5 Model. This calculation is performed using an excel spreadsheet. The predicted number of faults are then used to order the software modules from greatest predicted faults to least predicted faults. The predicted number of faults are only used for ordering the data. Cutoff points from top 5% to top 50% are selected, and the range's actual number of faults is compared to the data sets total number of faults.

Fit Data (188 instances, 427 Faults): Underlying Prediction Model = Linear Regression With M5 Feature Selection						
FAULTS = - 0.0516 * NUMUORS + 0.0341 * NUMUANDS - 0.0027 * TOTOTORS - 0.0372 * VG + 0.2119 * NLOGIC + 0.0018 * LOC + 0.005 * ELOC - 0.3091						
		Perfect Rank		Predicted Rank		Model Performance
Top % Of Modules	Total # Of Modules	G(c) = # Of Faults	Percentage Of Total Faults Accounted	G'(c) = # Of Faults	Percentage Of Total Faults Accounted	ϕ(c) = G'(c) / G(c)
5%	9	170	39.81%	154	36.07%	90.59%
10%	18	256	59.95%	236	55.27%	92.19%
15%	28	312	73.07%	280	65.57%	89.74%
20%	37	349	81.73%	309	72.37%	88.54%
25%	47	376	88.06%	343	80.33%	91.22%
30%	56	393	92.04%	357	83.61%	90.84%
35%	65	402	94.15%	370	86.65%	92.04%
40%	75	412	96.49%	378	88.52%	91.75%
45%	84	421	98.59%	384	89.93%	91.21%
50%	94	427	100.00%	391	91.57%	91.57%

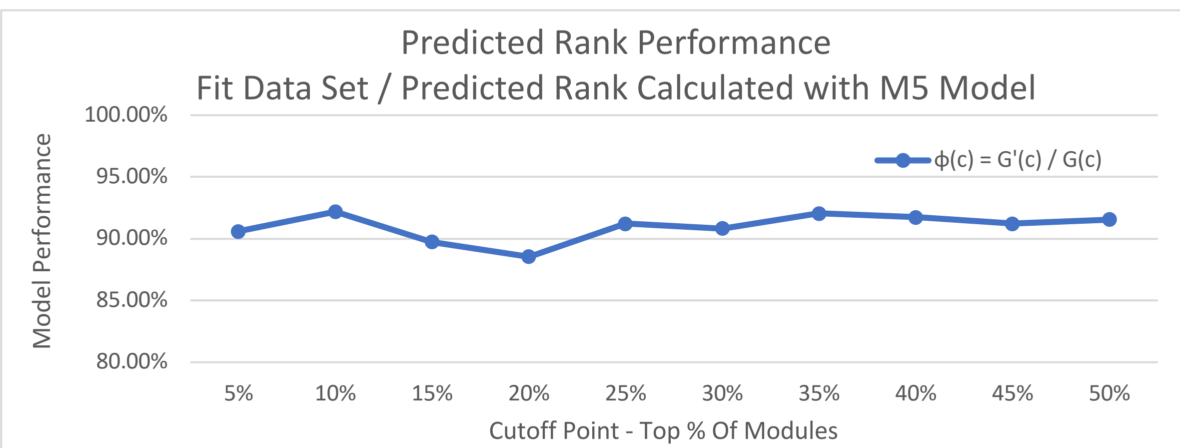
The above chart displays the results of MOM applied to the fit data set using the M5 Model to calculate predicted number of faults and create predicted ranking. Reviewing the perfect ranking provides significant insight into the fault prone software problem. In this data set, 81.73% of all faults in the data set are contained in just 20% of the modules, lining up very nicely with the Pareto Principle. The Pareto Principle is a regularly occurring phenomenon which states that for many events 80% of the effects arise from 20% of the causes. The table also shows that all faults from the fit data set are contained within the top 50% of modules when using the perfect rank.

Even more interesting is the power of the predicted ranking, as its performance average for all cutoffs is 90.9%. The performance is calculated by dividing total number of faults in the predicted rank cutoff range by the total number of faults in the perfect rank cutoff. Even if the predicted number of faults calculated by the M5 model is not correct, it still provides sufficient knowledge to rank the modules effectively from most fault prone to least fault prone. If a cutoff of top 20% was selected using the predicted rank, 72.37% of all faults in the data set would be covered.

The Alberg Diagram below plots G(c) / G_{total} and G'(c) / G_{total}, where G(c) denotes the total number of faults covered in given cutoff using the perfect rank, and G'(c) denotes the total number of faults covered in a given cutoff using the predicted rank. G_{total} is the total number of faults in the given data set.



The M5 Model is effective in predicting module faults for ranking purposes, as the two trends (predicted vs perfect) are a close match. The closer the predicted rank trend line is to the perfect rank trend line, the better the underlying prediction model is at ordering the modules.

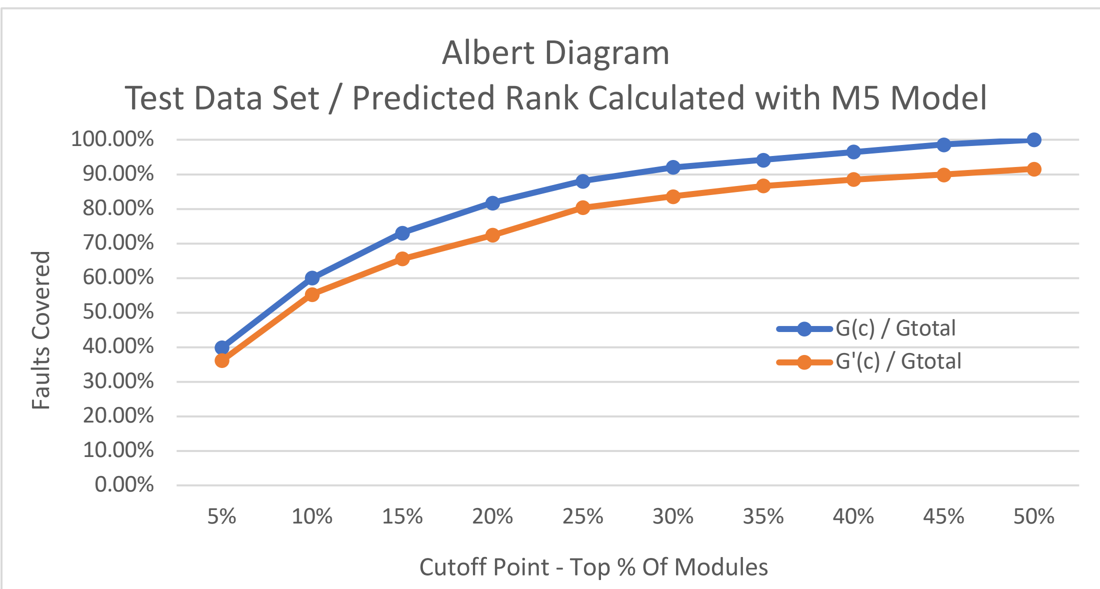


The performance diagram displays the MOM performance across all selected cutoff points. For all cutoff points, the performance is roughly 90%, with a low of 88.54% at 20% cutoff and a high of 92.19% at 10% cutoff. The consistent accuracy across various cutoff points implies that the underlying prediction model (M5) is robust.

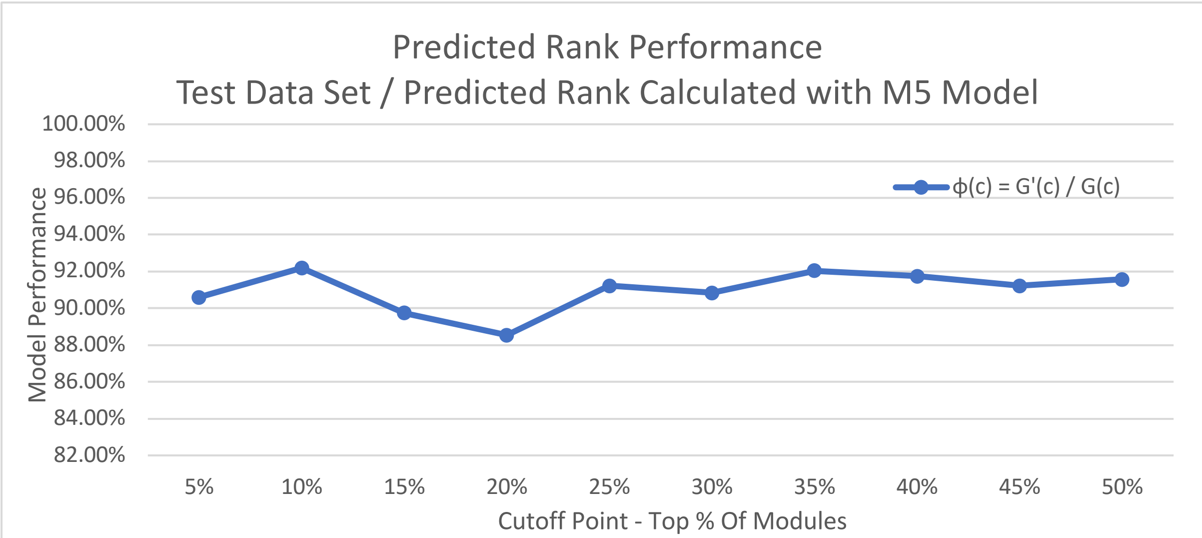
This process is then repeated using the test data set in combination with the M5 model. As this is a different data set, it makes sense to analyze the perfect rank's results to gain a better understanding of the software module ordering problem as a whole. Once again when using perfect ranking, the top 20% of the modules contains 82.16% of the total number of faults in the entire data set. Also, all faults in the entire data set are contained in the top 50% of the modules when ordered using perfect ranking. This enforces the pattern that was observed with the fit data set. Next, the results obtained using the M5 Model to determine the predicted ranking will be analyzed.

Test Data (94 instances, 241 faults): Underlying Prediction Model = Linear Regression With M5 Feature Selection						
FAULTS = - 0.0516 * NUMUORS + 0.0341 * NUMUANDS - 0.0027 * TOTOTORS - 0.0372 * VG + 0.2119 * NLOGIC + 0.0018 * LOC + 0.005 * ELOC - 0.3091						
		Perfect Rank		Predicted Rank		Model Performance
Top % Of Modules	Total # Of Modules	G(c) = # Of Faults	G(c) / Gtotal	G'(c) = # Of Faults	G'(c) / Gtotal	ϕ(c) = G'(c) / G(c)
5%	4	101	41.91%	87	36.10%	86.14%
10%	9	152	63.07%	126	52.28%	82.89%
15%	14	181	75.10%	155	64.32%	85.64%
20%	18	198	82.16%	187	77.59%	94.44%
25%	23	213	88.38%	196	81.33%	92.02%
30%	28	223	92.53%	197	81.74%	88.34%
35%	32	227	94.19%	206	85.48%	90.75%
40%	37	232	96.27%	212	87.97%	91.38%
45%	42	237	98.34%	219	90.87%	92.41%
50%	47	241	100.00%	221	91.70%	91.70%

The predicted rank performance average across all cutoffs is 89.57%, very close to the performance results observed against the fit data set. Using predicted ranking, the top 20% of the modules contains 77.59% of the total number of faults in the test data, which is more than the 72.37% covered with 20% cutoff on the fit data.



The Alberg Diagram displays results similar to those observed with the fit data. The close correlation between the predicted ranking coverage (red) and the perfect ranking coverage (blue) suggests that the M5 model is effective in predicting module faults for ordering purposes. This does not imply that there is low error in predicting the number of faults, but it does imply that it is effective in predicting the number of faults in a manner that allows for accurate ordering of modules.



The performance plot for test data set displays a performance trend consistent with that achieved when evaluating with the fit data test. A minimum performance of 82.89% is observed when a cutoff of 10% is selected, and a maximum performance of 94.44% if observed at the 20% cutoff. We are not so much interested in a minimum and maximum as we are in a high accuracy that is consistent across all cutoff points, as this is the result of an underlying prediction model that effectively ranks modules.

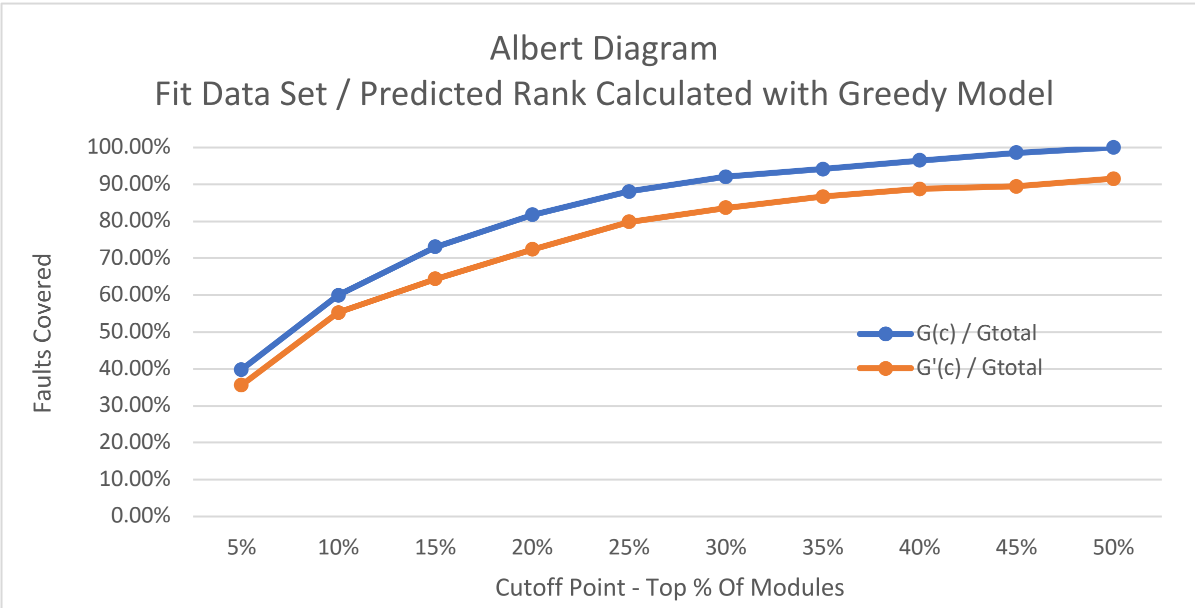
Although it can be considered insignificant, there is slightly more variance in the performance plot of MOM with the test data set than was previously seen using the fit data set. It is important to remember that the linear regression model being used, the M5 model, was trained using the fit data set. Having already seen the data, it is not surprising that the model favors the fit data set slightly.

III) Determining Predicted Rank with Greedy Model

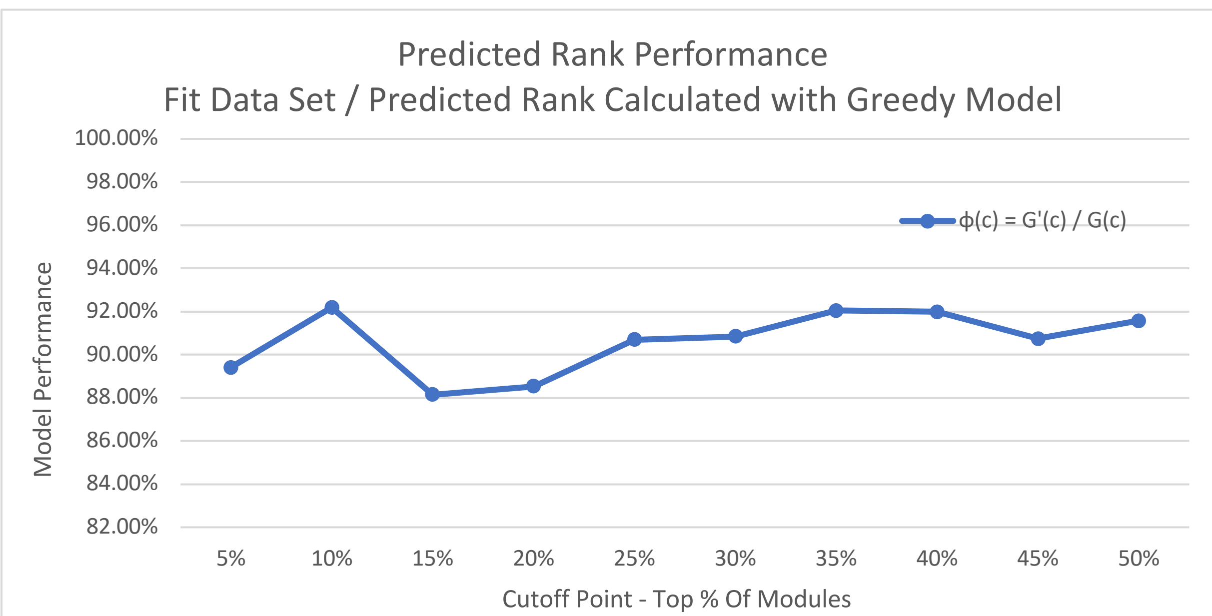
The procedure outlined in Part II is repeated with a new underlying prediction model, the linear regression model with greedy feature selection that was calculated in Assignment 1. Results will be presented and compared to those of Part II.

Fit Data (188 instances, 427 faults): Underlying Prediction Model = Linear Regression With Greedy Feature Selection						
FAULTS = - 0.0482 * NUMUORS + 0.0336 * NUMUANDS - 0.0021 * TOTOTORS - 0.0337 * VG + 0.2088 * NLOGIC + 0.0019 * LOC - 0.3255						
		Perfect Rank		**Predicted Rank**		**Model Performance**
Top % Of Modules	Total # Of Modules	G(c) = # Of Faults	G(c) / Gtotal	G'(c) = # Of Faults	G'(c) / Gtotal	ϕ(c) = G'(c) / G(c)
5%	9	170	39.81%	152	35.60%	89.41%
10%	18	256	59.95%	236	55.27%	92.19%
15%	28	312	73.07%	275	64.40%	88.14%
20%	37	349	81.73%	309	72.37%	88.54%
25%	47	376	88.06%	341	79.86%	90.69%
30%	56	393	92.04%	357	83.61%	90.84%
35%	65	402	94.15%	370	86.65%	92.04%
40%	75	412	96.49%	379	88.76%	91.99%
45%	84	421	98.59%	382	89.46%	90.74%
50%	94	427	100.00%	391	91.57%	91.57%

The fit and test data sets are the same as in Part II, all remarks made regarding the perfect rank still apply. We are interested in examining the results of the predicted rank that is achieved using the Greedy Model. Using the predicted rank and a cutoff of 20%, 72.37% of all faults are covered with an accuracy of 88.54%.



The Albert Diagram again displays a close relation between the predicted rank coverage trend and the perfect rank coverage trend. Using the Greedy Model as the underlying predictor, we are able to capture 79.86% of all faults by selecting the top 25% of the modules when ranked by predicted number of faults.

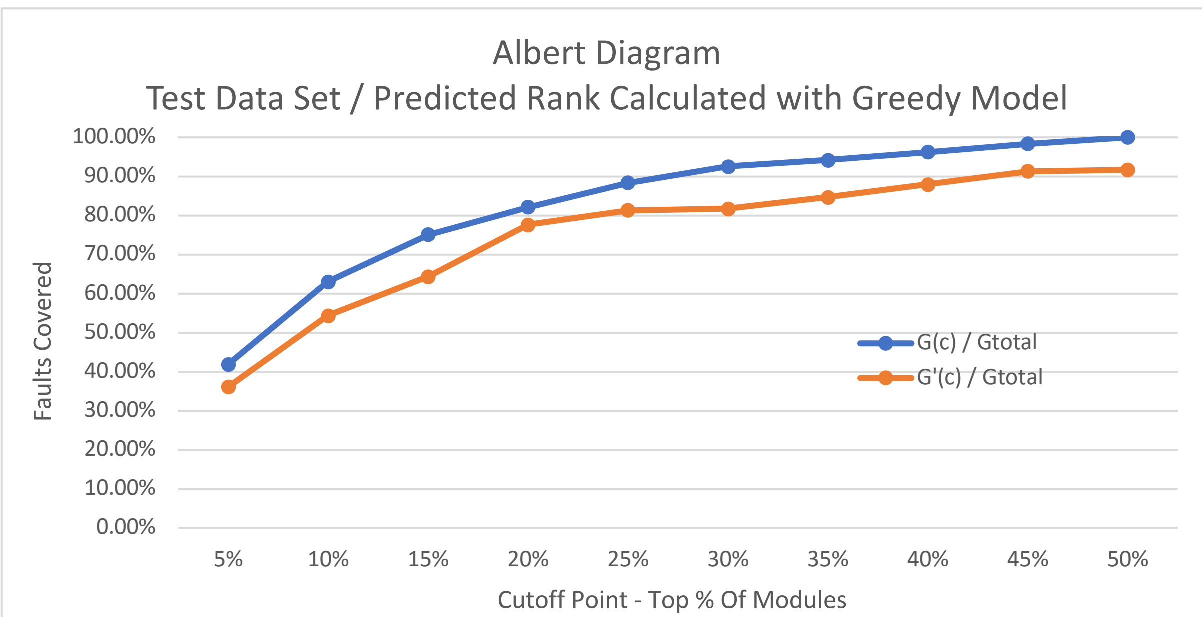


The performance plot for the M5 Model predictor and the fit data set displays a consistent accuracy over all cutoff points, with an average of 90.61%. There is a minimum accuracy 88.14% at a cutoff 15% and a maximum accuracy of 92.19% at a cutoff of 10%.

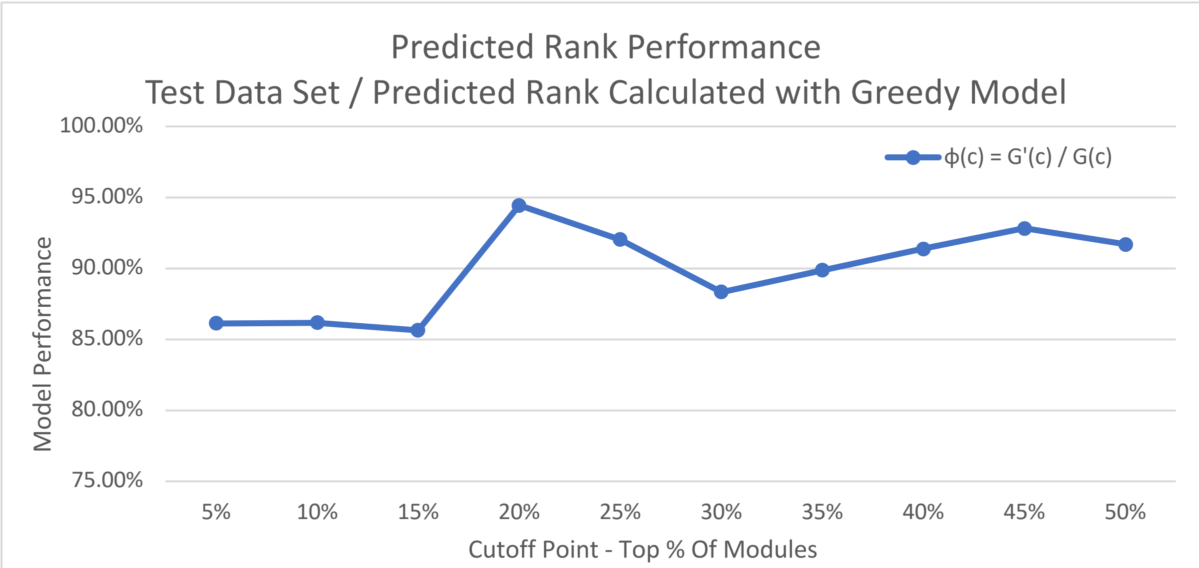
Finally, we apply MOM to the test data set using the Greedy Model as the underlying predictor.

Test Data (94 instances, 241 faults): Underlying Prediction Model = Linear Regression With Greedy Feature Selection						
FAULTS = - 0.0482 * NUMUORS + 0.0336 * NUMUANDS - 0.0021 * TOTOTORS - 0.0337 * VG + 0.2088 * NLOGIC + 0.0019 * LOC - 0.3255						
		Perfect Rank		Predicted Rank		Model Performance
Top % Of Modules	Total # Of Modules	G(c) = # Of Faults	G(c) / Gtotal	G'(c) = # Of Faults	G'(c) / Gtotal	ϕ(c) = G'(c) / G(c)
5%	4	101	41.91%	87	36.10%	86.14%
10%	9	152	63.07%	131	54.36%	86.18%
15%	14	181	75.10%	155	64.32%	85.64%
20%	18	198	82.16%	187	77.59%	94.44%
25%	23	213	88.38%	196	81.33%	92.02%
30%	28	223	92.53%	197	81.74%	88.34%
35%	32	227	94.19%	204	84.65%	89.87%
40%	37	232	96.27%	212	87.97%	91.38%
45%	42	237	98.34%	220	91.29%	92.83%
50%	47	241	100.00%	221	91.70%	91.70%

With a cutoff of 20%, the predicted rank provides 77.59% coverage of all faults contained in the data set, producing an accuracy of 94.44%. This is identical to the coverage and accuracy which was achieved using the M5 Model as the underlying predictor of the test data in Part II. The Greedy Model coverage and accuracy is very similar to the M5 Model coverage and accuracy from Part II, implying that the predicted rank generated from both models is very close to the same.



The Albert Diagram compares the predicted rank's fault coverage to the perfect rank's fault coverage. The two trends are closest at 20%, indicating that this model will have the highest accuracy at a cutoff of 20%. Similar to previous Albert Diagrams, the two trends are very closely related. This suggests that both of the underlying linear regression prediction models are effective in predicting the number of faults for ranking purposes.



The performance plot for MOM using the Greedy Model on the test set indicates a maximum accuracy of 94.44% at a cutoff of 20%, a minimum accuracy of 85.64% at 15% cutoff, and an average accuracy of 89.85%. These results are consistent with previous models, all of which had average accuracies around 90%. Similar to Part II, the accuracy of MOM against the test data set is slightly less than the accuracy of MOM against the fit data set. The difference is negligible, decreasing from 90.61% down to just 89.85%.

IV) Conclusions

Module Order Modelling (MOM) was applied to fit and test data sets using two different underlying linear regression prediction models. The first regression model was trained using M5 feature selection, and the second was trained using Greedy feature selection. The linear regression models are used to predict the number of faults for each software module. The modules are then sorted from greatest predicted faults to least predicted faults. Various cutoff points ranging from 5% to 50% are selected. For each cutoff, the percentage of faults covered by the predicted ranking is calculated and compared to the percentage of faults covered by the perfect ranking. The performance of MOM is calculated by dividing the total number of faults covered using predicted rank by the total number of faults covered using perfect rank. The following table summarizes the accuracy of both models with both the fit and test data sets.

MOM Performance Comparison With Linear Regression Predictor

	M5 Feature Selection	M5 Feature Selection	Greedy Feature Selection	Greedy Feature Selection
	Fit Data	Test Data	Fit Data	Test Data
Cutoff	$\phi(c) = G'(c) / G(c)$	$\phi(c) = G'(c) / G(c)$	$\phi(c) = G'(c) / G(c)$	$\phi(c) = G'(c) / G(c)$
5%	90.59%	86.14%	89.41%	86.14%
10%	92.19%	82.89%	92.19%	86.18%
15%	89.74%	85.64%	88.14%	85.64%
20%	88.54%	94.44%	88.54%	94.44%
25%	91.22%	92.02%	90.69%	92.02%
30%	90.84%	88.34%	90.84%	88.34%
35%	92.04%	90.75%	92.04%	89.87%
40%	91.75%	91.38%	91.99%	91.38%
45%	91.21%	92.41%	90.74%	92.83%
50%	91.57%	91.70%	91.57%	91.70%
Average	**90.97%**	**89.57%**	**90.61%**	**89.85%**

Both linear regression models led to module ordering with high accuracy near 90%, indicating that the underlying prediction models effectively predicted the number of faults for ordering purposes. This does not imply that the models accurately predicted the number of faults for each module, rather that they predicted the number of faults in such a way that allowed the predicted rank to closely match the perfect rank.

Compared to traditional classification of software modules, as fault prone vs non-fault prone, MOM is favorable for practical application. It is flexible in the sense that development teams can select a cutoff point that matches their available resources. If a team only has resources to apply software improvement to 10% of their modules, MOM can be applied to rank modules by predicted number of faults and then the top 10% can be selected for review. Since modules are effectively ordered from greatest predicted faults to least predicted faults, development teams can maximize productivity by focusing on the most fault prone modules first. This also eliminates the need to define a fault prone threshold (total number of faults which defines a module 'fault prone'). Another key takeaway is the presence of the Pareto Principle, where roughly 80% of the faults come from just 20% of the software modules.