Artefact: An R Implementation of the AutoSpearman Function

Jirayus Jiarpakdee, Chakkrit Tantithamthavorn, Christoph Treude
School of Computer Science, The University of Adelaide, Australia.
firstname.lastname@adelaide.edu.au

***Abstract*—This artifact is the implementation of `AutoSpearman`, an automated metric selection approach based on correlation analyses. The goal of `AutoSpearman` is to automatically mitigate correlated metrics prior to constructing analytical models. This artifact is implemented as an R package and is available in the GitHub repository. We provide descriptions and R code snippets for the installation of `AutoSpearman` and usage examples.**

I. Introduction

The interpretation of defect models heavily relies on the software metrics that are used to construct them. However, software metrics often have strong correlation among themseleves and such correlated metrics may have a negative impact on the interpretation of defect models [3]. Thus, to automatically mitigate correlated metrics prior to constructing analytical models, we propose `AutoSpearman`, an automated metric selection approach based on correlation analyses, in our recent work which is published at ICSME 2018 [4]. We implement `AutoSpearman` as an R package which consists of three functions, i.e., `loadDefectDataset`, `plotVarClus`, and `AutoSpearman`. Below, we provide descriptions and R code snippets for the installation of our R package and usage examples for the package's functions.

II. The Artifact

In this section, we discuss how to setup and install the artifact, and the three functions of the artifact.

Setup and Installing the Artifact. To setup and install this artifact, we use the `install_github` function as provided by the `devtools` R package. The function installs our R package from the GitHub repository [1]. Below, we provide an R code snippet to setup and install the artifact.

```
install.packages('devtools')
library('devtools')
install_github('software-analytics/Rnalytica')
```

The loadDefectDataset function loads a collection of publicly-available defect datasets. The detailed explanation for each defect dataset can be found in the online repository. Below, we provide an R code snippet that loads the `Eclipse Platform 2` dataset as provided by Zimmermann *et al.* [6].

```
library('Rnalytica')
Data = loadDefectDataset('eclipse-2.0')
```

The plotVarClus function measures pair-wise correlations among input metrics and presents a visualization of the hierarchical cluster analysis on these correlations. Correlated metrics (i.e., metrics that have their correlation coefficient above the threshold) are highlighted in red, while non-correlated metrics are highlighted in green. The default setting of the correlation calculation is `spearman` (i.e., the Spearman rank correlation test) with a correlation threshold of 0.7, as suggested by Kraemer *et al.* [5] (i.e., a Spearman correlation coefficient of above *0.7 is considered as a strong correlation*). Below, we provide an R code snippet that applies the `plotVarClus` function on the `Eclipse Platform 2` dataset.

```
library('Rnalytica')
Data = loadDefectDataset('eclipse-2.0')
plotVarClus(dataset = Data$data, metrics = Data$indep,
    ↪ correlation = 'spearman',
    ↪ correlation.threshold = 0.7)
```

The AutoSpearman function identifies and mitigates correlated metrics based on the Spearman rank correlation test and the Variance Inflation Factor (VIF) analysis. The output of this function is the subset of metrics that do not have strong correlation among themselves. Similar to the `plotVarClus` function, the default setting of the Spearman correlation threshold is 0.7. Furthermore, the default setting of the VIF threshold is 5 as suggested by Fox [2]. Below, we provide an R code snippet that applies the `AutoSpearman` function on the `Eclipse Platform 2` dataset.

```
library('Rnalytica')
Data = loadDefectDataset('eclipse-2.0')
AutoSpearman(dataset = Data$data, metrics = Data$indep,
    ↪ spearman.threshold = 0.7, vif.threshold = 5)
```

References

[1] "Rnalytica: An R package of the Miscellaneous Functions for Data Analytics Research," *https://github.com/software-analytics/Rnalytica.*

[2] J. Fox, *Applied regression analysis and generalized linear models.* Sage Publications, 2015.

[3] J. Jiarpakdee, C. Tantithamthavorn, and A. E. Hassan, "The Impact of Correlated Metrics on Defect Models," *arXiv preprint arXiv:1801.10271*, p. To Appear, 2018.

[4] J. Jiarpakdee, C. Tantithamthavorn, and C. Treude, "AutoSpearman: Automatically Mitigating Correlated Software Metrics for Interpreting Defect Models," in *Proceedings of the International Conference on Software Maintenance and Evolution (ICSME)*, 2018, p. To appear.

[5] H. C. Kraemer, G. A. Morgan, N. L. Leech, J. A. Gliner, J. J. Vaske, and R. J. Harmon, "Measures of Clinical Significance," *Journal of the American Academy of Child & Adolescent Psychiatry (JAACAP)*, vol. 42, no. 12, pp. 1524–1529, 2003.

[6] T. Zimmermann, R. Premraj, and A. Zeller, "Predicting Defects for Eclipse," in *Proceedings of the International Workshop on Predictor Models in Software Engineering (PROMISE)*, 2007, pp. 9–19.