Financial Analytics Project

Part 2 Report

Extracting the best features from a model based clustered malware dataset.

Ezeanaka Kingsley Uchechukwu

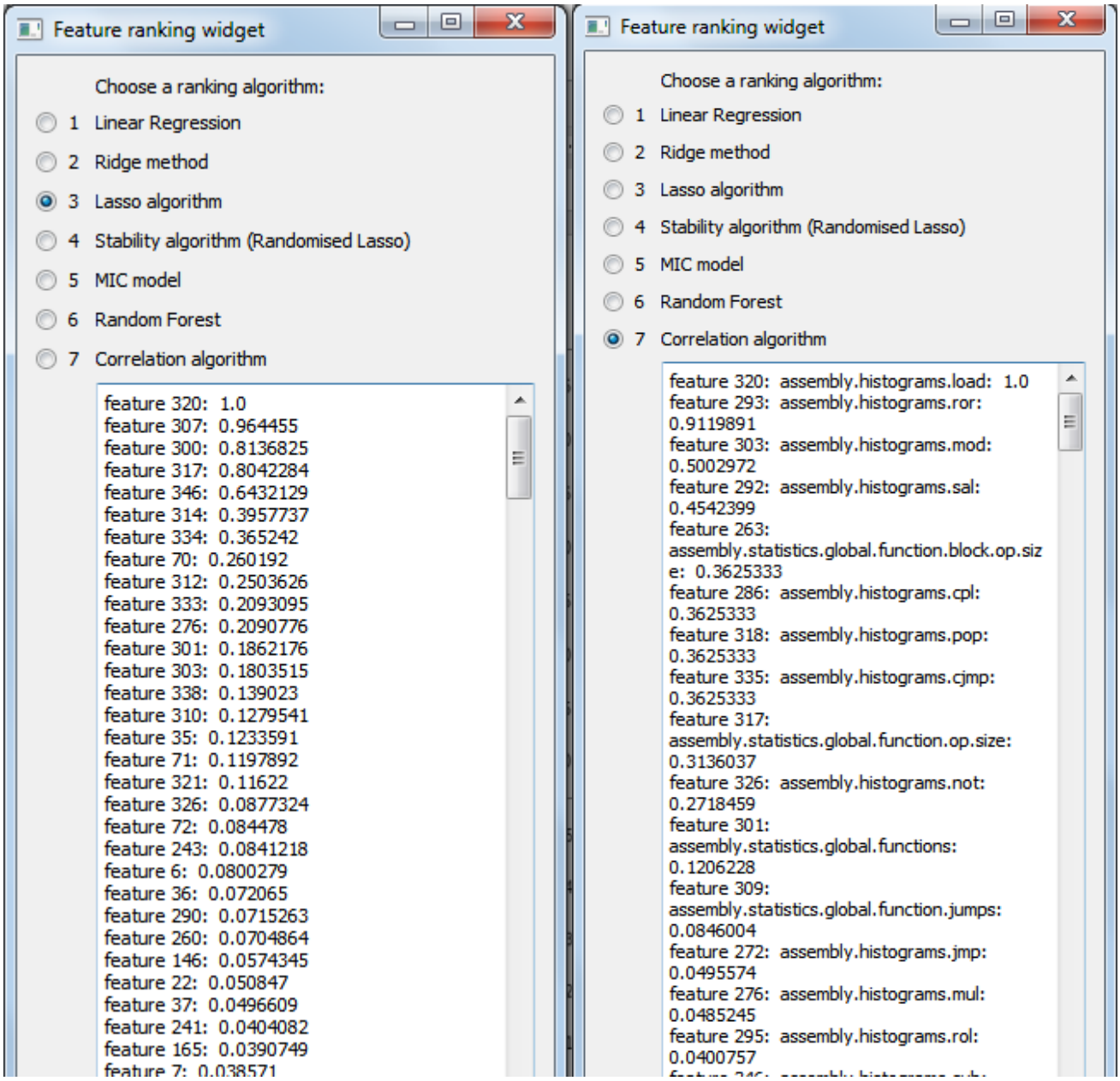
Introduction

The big question in this area of machine learning is “Which features should you use to create a predictive model?”**.** This is arguably a difficult question that may require a deep knowledge of the problem domain.

It is actually possible to automatically select the very features in your data that prove to be the most useful or relevant for the given problem that you are working on. This is a process called feature selection. [1] One important attribute of feature selection is that it does not seek to modify the features in the original dataset but rather it selects the relevant features based on some scoring benchmark. The process of scoring the features and grouping them according to their scores is feature ranking[2].

In order to a make a simple, yet accurate predictive model, we need to select only the best features of that dataset. In part one of the project we already talked briefly about the different machine learning algorithms that are used to train a predictive model and generate set of scores for each of the features. In this part we will extract those best features, together with their names and their respective scores for our model construction.

Project solution description :



(a) (b)

Fig.1 *– Feature ranking scores from the highest ranked down to the lowest ranked.*

(a) In part one project phase without including the names of the features,

(b) In part two project phase with the corresponding feature names included

Figure 1 above displays the feature ranking interface at the two phases of our project. The first figure which comes from the first part of the project shows the list of the features numbered serially but sorted in descending order of their scores. In the second figure, we see an important modification in the display; In addition to the serially named features, we see the actual names of

the features that own those scores. And of course, we also see that while the first output is produced by Lasso training algorithm, the second feature ranking scores were generated by the Correlation algorithm.

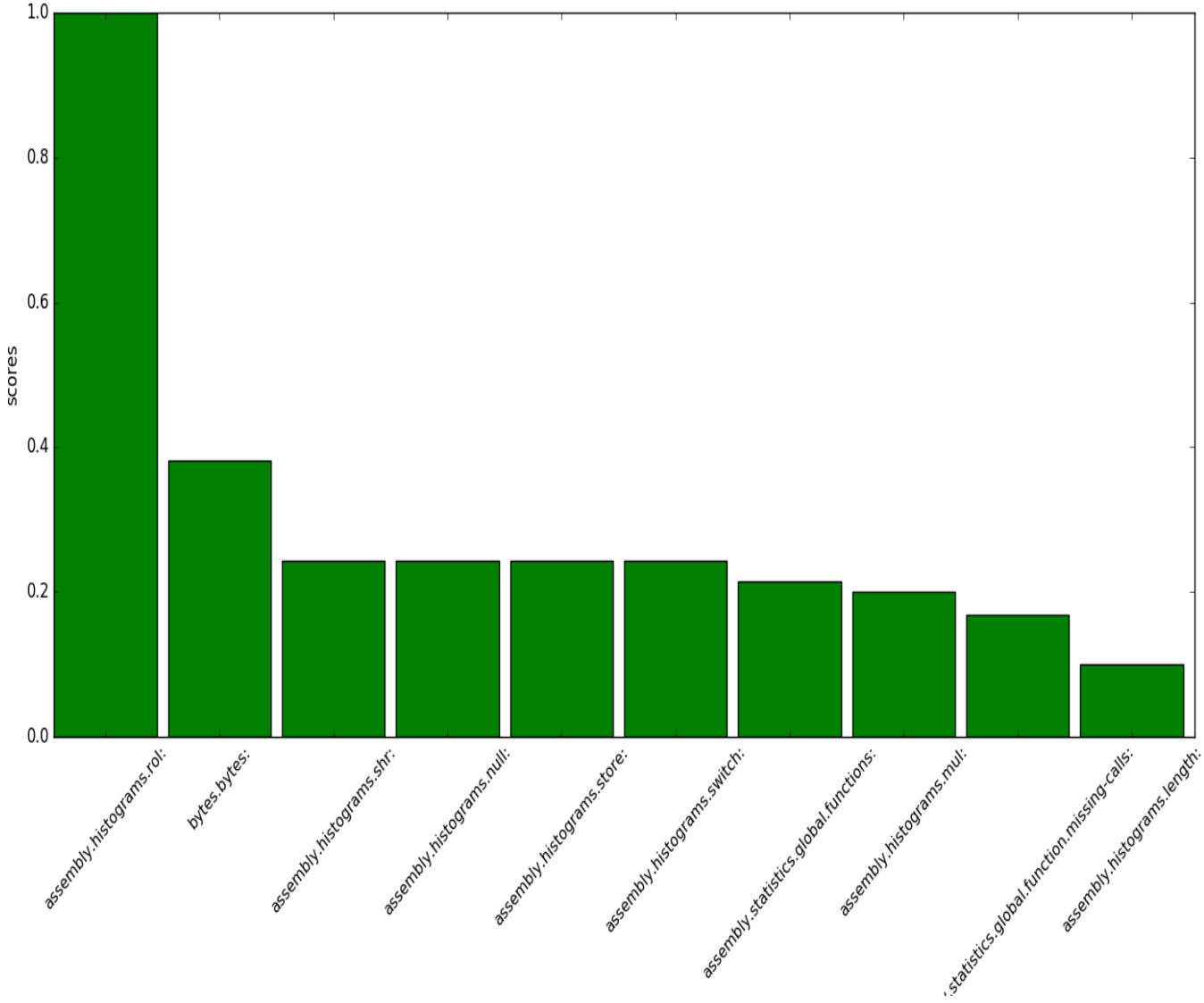


Fig.2 *– The top ten features for the 16-th cluster of our clustered malware dataset.*

Figure 2 above illustrates how the best features that got selected from one of the clusters compare against each other in terms of their relative importance scores. We see the highest ranked feature as “assembly-histograms-rol” and the least ranked ranked from our list of best features as “assembly-histograms-length”.

Related Work

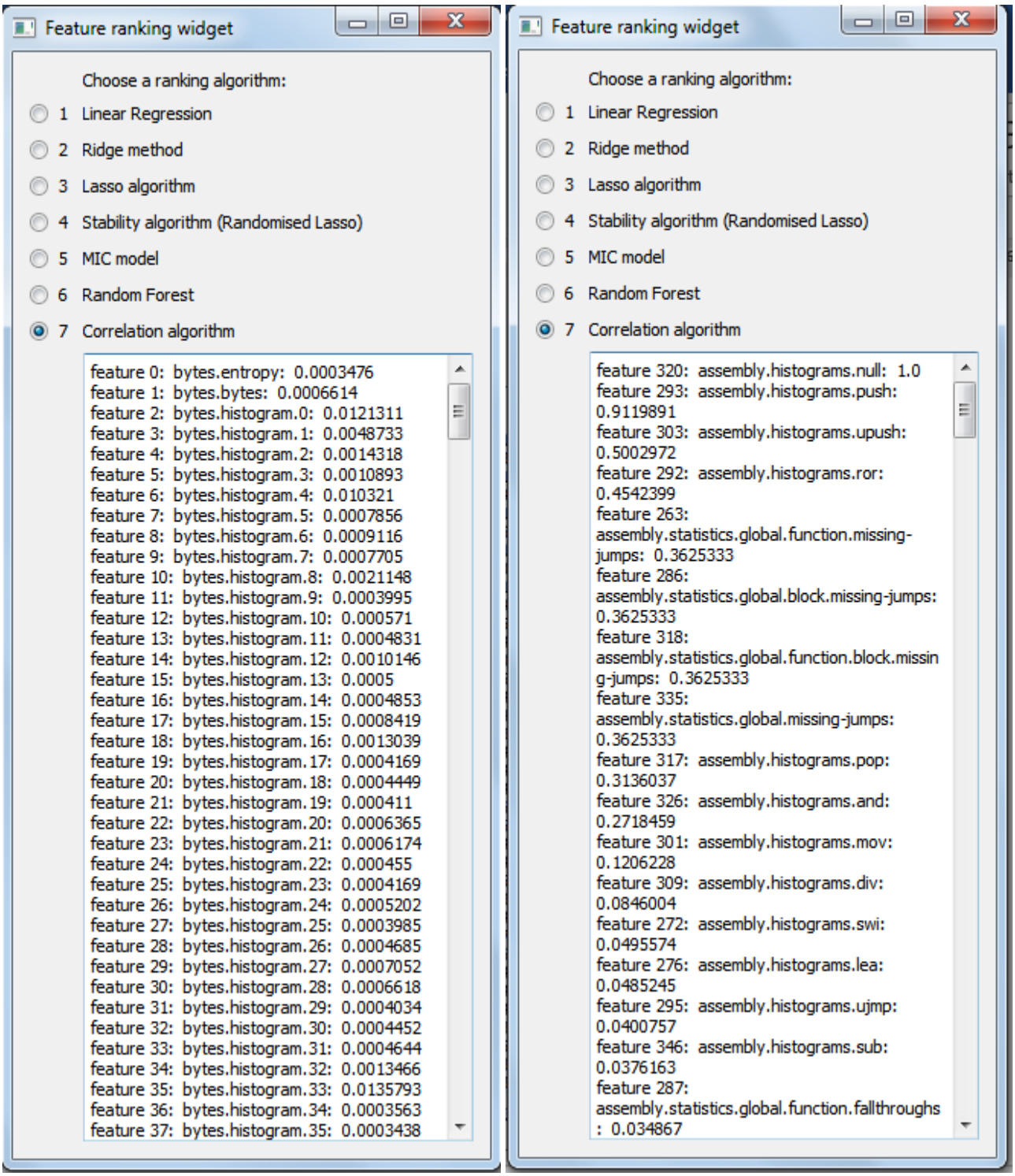
Feature ranking and selection is a fairly popular concept in the field of machine learning, and as such, there are a number of other algorithms in the industry that also compute the importance scores of dataset features. A number of them worthy of note include the recursive feature elimination algorithm

Program Pipeline

The program builds on what we have for the first part of the project. We recall that the feature ranking widget reads a dataset of multiple features, numbers the features serially and trains the dataset using one of several machine learning algorithms to generate importance scores for all the features in the given dataset, and produces an output containing the scores of the features all listed in descending order of their importance with the features numbered serially. In this final part of the project, the program extracts the names associated with the numbered features in the process of training the dataset and prints the names alongside that the numbering of the features.

In the second phase of the part 2 of the project, the feature ranking scores generated are each copied to files for subsequent use in this phase. Hence the scores generated for the features associated with the label01 cluster will be copied and stored in the scores01.txt file. The score files are then used as inputs for extracting the best features which we decide, will be the top ten features in the feature rankings.

These top ten features together with their names and respective scores are then used plot a bar graph that visually shows how their relative importance of the features compare with one another.



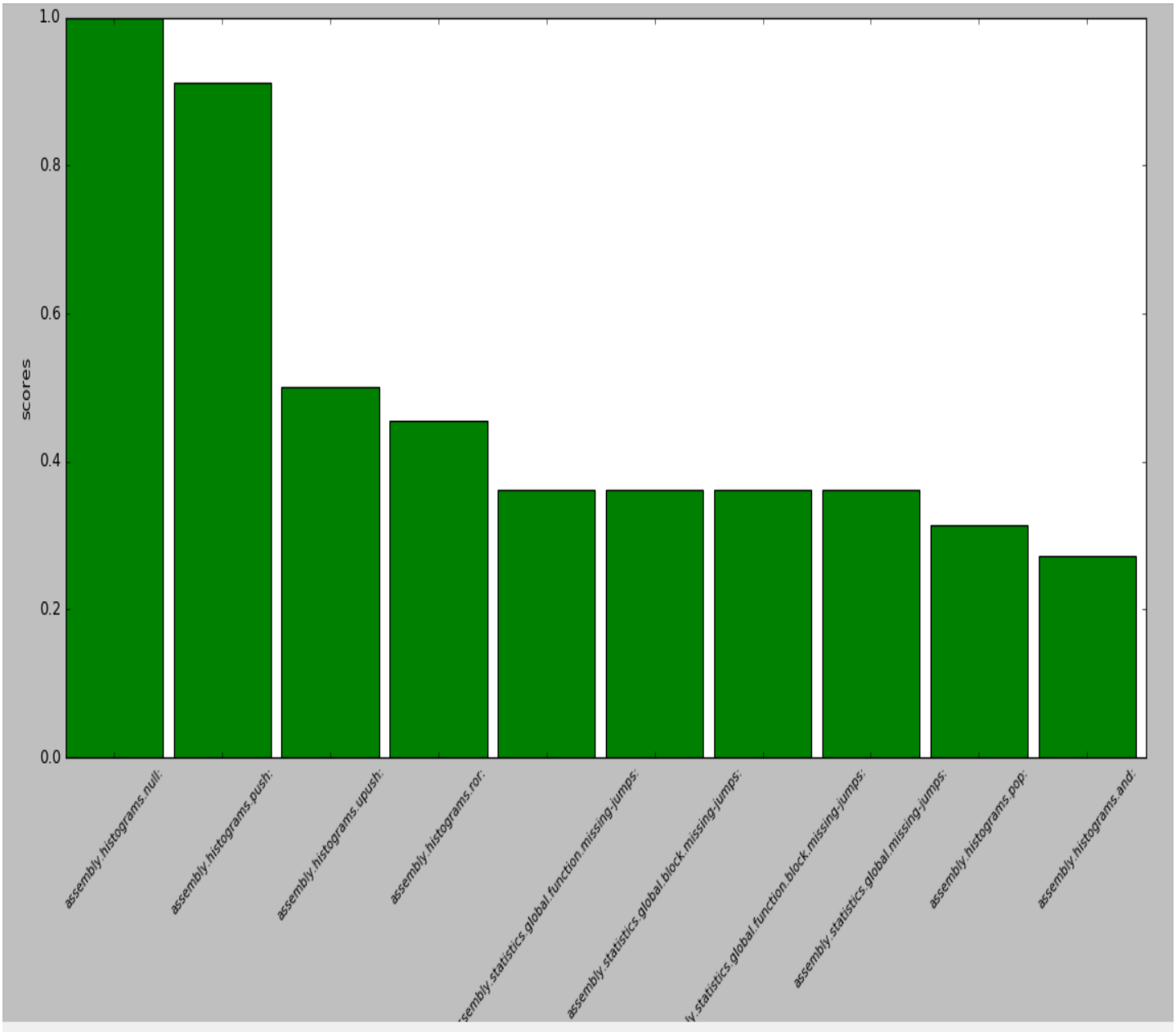
(a) (b)

Figure 2. Screenshot of the widget executing Correlation ranking algorithm

a) Before sorting the importance scores of the features
b) After sorting the scores.

Figure 2 above illustrates what we have been talking about in the previous page.

As we can see, after completing the sorting of the importance scores, the highest ranked feature is “assembly.histogram.null”. This feature, together with the next 9 highly ranked features are used in making the bar chart seen in Figure 3 below.



Limitations and challenges

Some of the limitations and challenges encountered in phase 1 remained in this phase, albeit, with lesser intensity. My knowledge of the python programming language has gradually progressed over the course of the project development but some difficulties associated with beginners of any programming languages continue to persist.

Some of the algorithms for training the dataset are more effective than others in distinguishing features in terms of the ranking scores. The random forest algorithm, for example, was able to distinctly give scores above 0 for

only about a dozen features out of the vast number of features in the input dataset for cluster 66.

The graph also struggles to show the complete names of some of the features with unusually long names. Hence if you have a look at the feature ranking charts of some of the generated features, you would see that some of the feature names were not completely displayed.

Future plan

The plan the rest of this part of the project would be automating the process of reading the cluster files rather than having them done manually one file at a time.

References

[1] Bermingham, Mairead L.; Pong-Wong, Ricardo; Spiliopoulou, Athina; Hayward, Caroline; Rudan, Igor; Campbell, Harry; Wright, Alan F.; Wilson, James F.; Agakov, Felix; Navarro, Pau; Haley, Chris S. (2015). *"Application of high-dimensional feature selection: evaluation for genomic prediction in man"*. *Sci. Rep.* .

[2] Gareth James; Daniela Witten; Trevor Hastie; Robert Tibshirani (2013). *An Introduction to Statistical Learning*. Springer. p. 204.