RESEARCH ARTICLE | AUGUST 08 2023

Boruta algorithm: An alternative feature selection method in credit scoring model FREE

Tri Handhika ✉; Murni; Rafi Mochamad Fahreza

AIP Conference Proceedings 2431, 080004 (2023)
https://doi.org/10.1063/5.0114178

Boruta Algorithm: An Alternative Feature Selection Method in Credit Scoring Model

Tri Handhika[1, a)], Murni[1, b)], Rafi Mochamad Fahreza[2, c)]

[1]*Centre for Computational Mathematics Studies, Gunadarma University, 100 Margonda*
Raya Street, Pondok Cina, Depok, West Java, 16424, Indonesia
[2]*Department of Informatics Engineering, Faculty of Industrial Technology, Gunadarma University, 100 Margonda*
Raya Street, Pondok Cina, Depok, West Java, 16424, Indonesia

[a)] Corresponding author: *trihandika@staff.gunadarma.ac.id*
[b)] *murnipskm@staff.gunadarma.ac.id*
[c)] *rafifahreza@student.gunadarma.ac.id*

Abstract. This paper analyzed the feature selection for reducing the number of input variables when developing a predictive model. Boruta Algorithm is using in this paper as a wrapper around a Random Forest classification algorithm. Boruta algorithm is one of the algorithms used to determine the significant variables (feature selection) in a classification model in the machine learning approach, as supervised learning. Our results show that on the German Credit Data from the UCI Machine Learning with 20 variables, feature selection using Boruta Algorithm with Python Programming obtains 4 significant features.

INTRODUCTION

Machine learning method consists of two approaches, i.e., supervised and unsupervised approach [1, 2]. In practice, supervised learning has two stages: feature selection and classification [3]. This study focuses on the feature selection issue for the supervised learning. Supervised learning is known as classification model.

In machine learning and statistics, feature selection, also known as variable selection or attribute selection, is the process of reducing the number of input variables when developing a predictive model [4, 5, 6, 7]. There are two approaches in performing the feature selection stage. The first is the statistical-based feature selection techniques and the second is artificial intelligence-based feature selection techniques [8].

For the first one, statistical-based feature selection methods involve evaluating the relationship between each input feature and the target feature using statistics. These methods selecting those input features that have the strongest relationship with the target feature. As we know, statistical-based feature selection techniques can be fast and effective, although the choice of statistical measures depends on the data type of both the input and output features. For this reason, many practitioners use statistical-based feature selection techniques to reducing the number of input features. It would be very helpful if the classification stages used in this classification model is also a statistical-based classifier. If not, we will work twice to solve the classification model problems, such as use statistical-based feature selection techniques for feature selection stages (i.e. Multivariate Adaptive Regression Spline (MARS) model) [9, 10] and use artificial intelligence-based classifier techniques to solve classification stage (i.e. Random Forest classifier) [11, 12, 13, 14].

Boruta algorithm uses a wrapper approach built around a Random Forest classifier which is homogeneous ensemble classification algorithm [5, 6]. Boruta algorithm offers problem solving of classification model problems, such as in feature selection and classification stage. The algorithm is an extension of the idea introduced by to determine relevance by comparing the relevance of the real features to that of the random probes [15]. In this study, we only use Boruta Algorithm to determine the significant features in a classification model, or feature selection stage.

3rd International Conference on Science and Innovated Engineering
AIP Conf. Proc. 2431, 080004-1–080004-6; https://doi.org/10.1063/5.0114178
Published by AIP Publishing. 978-0-7354-4481-2/$30.00

Feature selection is used in this study to determine the selection feature based on the 20 features in the German Credit Data which is taken from the UCI Machine Learning Repository. In some other study, Boruta algorithm implemented in the R package **randomForest** to solve feature selection [16]. We tried to use another programming language to solve this problem. Therefore, in this study the feature selection using Boruta Algorithm performed on python programming, as well as with BorutaPy Libary.

BORUTA ALGORITHM

Boruta algorithm is one of the algorithms used to determine the significant variables (feature selection) in a classification model in the machine learning approach. This feature selection stage is necessary because there is often a decrease in model accuracy when the number of variables in the model is far from the optimal, as we know parsimonious model. The following are the steps in running the Boruta Algorithm [4]:

a) Extend the information system by adding a copy of all the variables used (the information systems are always extended by at least 5 shadow attributes, although the number of original attributes is less than 5);
b) Randomize additional attributes to remove their correlation with the dependent variable;
c) Execute the classification model for the extended information system and collect the calculated Z-score;
d) Determine the maximum Z-score of the shadow attributes (MZSA), and assign each attribute that has a Z-score that is better than the MZSA;
e) For each attribute whose importance cannot be determined, perform a two-way equation test with MZSA;
f) Regard the attributes which are of significantly less importance than MZSA as "unimportant" and permanently removed from the information system;
g) Consider attributes that are of significantly more importance than MZSA as "important";
h) Remove all shadow attributes;
i) Repeat this procedure until the importance is established for all attributes, or the algorithm has reached the pre-defined limits on the classification model used.

The idea of Boruta algorithm and the basis for the foundation of the random forest classifier is same, that by adding randomness to the system and collecting results from the ensemble of randomized samples. Based on the idea, we can reduce the misleading effects of random fluctuation and correlation [4]. On Boruta algorithm, this extra randomness will provide clarity about which attributes are absolutely important [4].

Some advantages of Boruta algorithm such as works well for both classification and regression problem, considers multi-variable relationships, handling interactions between variables, and improve on random forest variable importance measure which is a very popular method for variable selection.

Figure 1 illustrates the flowchart for this research where dataset is divided into two groups, i.e. training and validation sets. Dataset in the training set is then processed to select significant features. In this study, we used Boruta algorithm which was discussed before in point (a) to (i) as a feature selector.

IMPLEMENTATION

In this study, we using Python programming to solve feature selection. One library in the Python programming language that can be used in performing feature selection is BorutaPy library. The following describes some of the functions and parameters contained in the BorutaPy library which are useful in feature selection to determine significant features in a classification model.

a) **n_estimators**
set the number of estimators needed in the ensemble method. As if set **auto** then the amount will be automatically adjusted to the size of the dataset used.
b) **alpha**
Determine the p-value in testing the significance of features in the model where the default value is 5%.
c) **verbose**
Adjust the verbosity level of the resulting output.
d) **random_state**
Determine random numbers in managing the Boruta algorithm execution.

e) **fit**
Make adjustments between the models used and the available datasets.

f) **support_**
Show significant variables in a model.

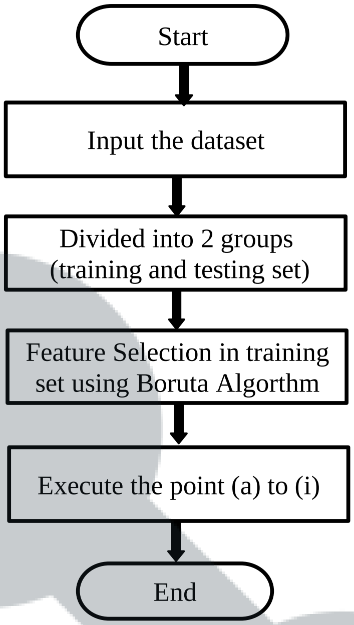
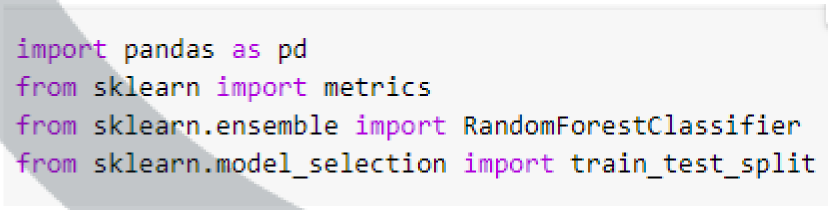

Start
Input the dataset
Divided into 2 groups (training and testing set)
Feature Selection in training set using Boruta Algorthm
Execute the point (a) to (i)
End

FIGURE 1. Flowchart

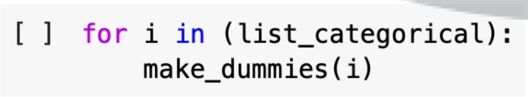
The data used in this paper is German Credit Data which is taken from the UCI Machine Learning Repository. This data consists of 20 features, such as status of existing checking account, duration in month, credit history, purpose, credit amount, savings account/bonds, present employment since, installment rate in percentage of disposable income, personal status and sex, other debtors/guarantors, present residence since, property, age, other installment plans, housing, number of existing credits at this bank, job, number of people being liable to provide maintenance for, telephone, and foreign worker.

Open Google Colab then call up some basic libraries needed in classification modeling, such as the Pandas and sklearn libraries, by typing the following code:



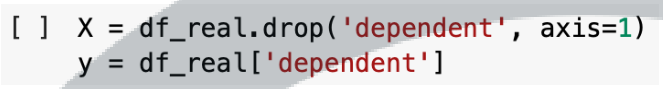
```
import pandas as pd
from sklearn import metrics
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
```

German Credit Data uploaded in Excel format. As for the columns classified as categoric variables, then they are stored in a list for later processing into dummy variables. After that, define some of the functions to measure categorical variables into dummy variables. Implement these functions on categorical variables that have previously been collected in a list through the following code:



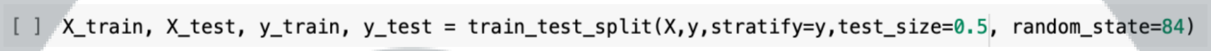
```
[ ]  for i in (list_categorical):
         make_dummies(i)
```

The dependent variable (Y) belongs to a categorical variable (binary) so that the problem is represented in a classification model. The other variables are considered as independent variables (X). The following is the code to separate the two types of variables so that they can be modeled further.



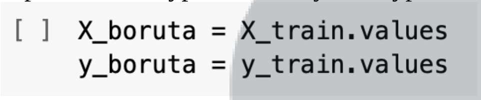
```
X = df_real.drop('dependent', axis=1)
y = df_real['dependent']
```

The next step is splitting the data into two groups known as training data and testing data. For this study, the separation was carried out proportionally between classes in each group using the Simple Random Sampling Technique with the **train_test_split** function where the comparison of the size of the training data and the testing data was determined to be the same, namely 50:50, as coded as follows:



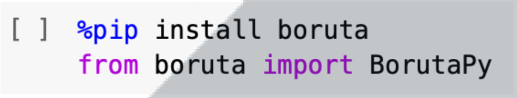
```
X_train, X_test, y_train, y_test = train_test_split(X,y,stratify=y,test_size=0.5, random_state=84)
```

The first step in performing feature selection using the Boruta Algorithm with the BorutaPy Library is to convert the separated data types into array data types according to the needs of using the library, as shown below:



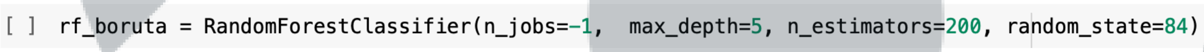
```
X_boruta = X_train.values
y_boruta = y_train.values
```

Furthermore, the execution of the installation and the call to the Boruta library on Google Colab as follows:



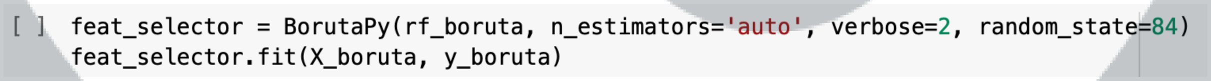
```
%pip install boruta
from boruta import BorutaPy
```

Define the classification model that will be used in which the Boruta Algorithm will be executed on that model. The classification model used is the Random Forest model available in the Sklearn library with the **RandomForestClassifier** function code where the value of each required parameter is determined subjectively as follows:



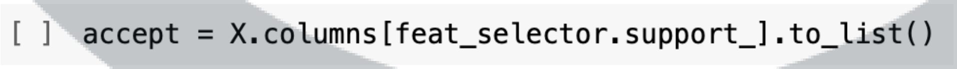
```
rf_boruta = RandomForestClassifier(n_jobs=-1,  max_depth=5, n_estimators=200, random_state=84)
```

Now the Boruta Algorithm is ready to be executed in the Random Forest model, as coded below using the BorutaPy function where the selected variable has a standard significance value of 5%, with a maximum iteration of 100 times:



```
feat_selector = BorutaPy(rf_boruta, n_estimators='auto', verbose=2, random_state=84)
feat_selector.fit(X_boruta, y_boruta)
```

The output of the code is to classify the features contained in the classification model into 3 (three) parts according to the size of the contribution of each feature in classifying each observation. To find out which features contribute strong and weak, it can be seen through the support function as follows:



```
accept = X.columns[feat_selector.support_].to_list()
```

Based on code ‘accept’, we find that the significant feature from German Credit Data are status of existing checking account, duration in month, credit amount, and age. The result of feature selection using Boruta algorithm is the same as if we use Multivariate Adaptive Regression Spline (MARS) model in other study [2]. In other words, it can be concluded that this Boruta algorithm is effective in performing feature selection. Furthermore, this Boruta algorithm can be used for the next stage, i.e. classification.

CONCLUSION

In this paper, we have been discussed feature selection for reducing the number of input variables when developing a predictive model. Feature selection in this paper using Boruta Algorithm, as a wrapper around a Random Forest classification algorithm. They are implemented to the German Credit Data with python programming, as BorutaPy Library. The results show that there are 4 features that are significant from 20 features German Credit Data, i.e. status of existing checking account, duration in month, credit amount, and age.

As future work, the same experiment can be implemented to the other data sets. Furthermore, Boruta Algorithm can be used to classification. Therefore, the Boruta algorithm can effectively in solving supervised learning problems, both for the feature selection and classification stages.

ACKNOWLEDGMENTS

This research was funded by PTUPT Research Grant 2020, Direktorat Riset dan Pengabdian Masyarakat, Direktorat Jenderal Penguatan Riset dan Pengembangan Kementerian Riset, Teknologi, dan Pendidikan Tinggi (DRPM KEMENRISTEKDIKTI) Indonesia to Gunadarma University No. 225/SP2H/AMD/LT/DRPM/2020.

REFERENCES

1. Murni, T. Handhika, A. Fahrurozi, I. Sari, and D.P. Lestari, “ Hybrid Method for Sentiment Analysis Using Homogenous Ensemble Classifier,” In Proceedings of the 2019 2nd International Conference of Computerand Informatics Engineering (IC2IE), pp 232-236 (2019).
2. F.Y. Osisanwo, J.E.T. Akinsola, O. Awodele, J.O. Hinmikaiye, Olakanmi, and J. Akinjobi, “Supervised Machine Learning Algorithms: Classification and Comparison,” International Journal of Computer Trends and Technology, 48 (3), pp 128-138 (2017).
3. T. Handhika, A. Fahrurozi, R.I.M. Zen, D.P. Lestari, I. Sari, and Murni, “Modified Average of the Base-Level Models in the Hill-Climbing Bagged Ensemble Selection Algorithm for Credit Scoring,” Procedia Computer Science, 157, pp 229-237 (2019).
4. I. Guyon and A. Elisseeff, “An Introduction to Variable and Feature Selection,” Journal of Machine Learning Research, 3, pp 1157–1182 (2003).
5. M.B. Kursa and W.R. Rudnicki, “Feature Selection with the Boruta Package,” Journal of Statistical Software, 36, pp 1-13 (2010).
6. X. Xu, H. Gu, Y. Wang, J. Wang and P. Qin, “ Autoencoder Based Feature Selection Method for Classification of Anticancer Drug Response,” Front Genet, 10, pp 1-10 (2019).
7. M. Aryuni and E.D. Madyatmadja, “Feature Selection in Credit Scoring Model for Credit Card Applicants in XYZ Bank: A Comparative Study,” International Journal of Multimedia and Ubiquitous Engineering, 10 (5), pp 17-24 (2015).
8. X.L. Li and Y. Zhong, “An Overview of Personal Credit Scoring: Techniques and Future Work,” International Journal of Intelligence Science, 2, pp 181-189 (2012).
9. T.S. Lee and I.F. Chen, “A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines,” Expert Systems with Applications, 28(4), pp 743–752 (2005).
10. J.H. Friedman, “ Multivariate adaptive regression splines,” The Annals of Statistics, 19(1), pp 1–141 (1991).
11. I. Brown and C. Mues, “An experimental comparison of classification algorithms for imbalanced credit scoring data sets,” Expert Systems withApplications, 39(3), pp 3446–3453 (2012).
12. S. Lessmann, B. Baesens, H.V. Seow and L.C. Thomas, “Benchmarking State-of-The-Art Classification Algorithms For Credit Scoring: An Update of Research,” European Journal of Operational Research, 244, pp. 124-136 (2015).
13. F. Louzada, A. Ara, and G.B. Fernandes, “Classification methods applied to credit scoring: Systematic review and overall comparison,” Surveys in Operations Research and Management Science, 21, 117–134 (2016).
14. L Breiman L, “Random Forests,” Machine Learning, 45, pp 5–32A (2001).
15. H. Stoppiglia, G. Dreyfus, R. Dubois, and Y. Oussar, “Ranking a Random Feature for Variable and Feature Selection,” Journal of Machine Learning Research, 3, pp 1399–1414 (2003).

16. Liaw and M. Wiener, “Classification and Regression by randomForest,” R News, 2(3), pp 18–22 (2002). Available: http://CRAN.R-project.org/doc/Rnews/.