??? Author(s) non exists. . . ???

AUTOML FEATURE SELECTION METHOD FOR CLASSIFICATION

Abstract *Feature selection is a process aiming for reducing the number of variables when building a prediction model or performing a machine learning procedure. In this paper, we suggest an automated machine learning mechanism for the task of feature selection, which relies on the comparison between two methods: random forest and XGBoost classifier. We present both backward and forward approaches for the feature selection process, and test our suggested algorithm on 4 different datasets. In all cases, the results show that the number of features for building the model can be significantly reduced, while model accuracy is maintained high. Our auto feature selection method presents an effective and efficient strategy for users to adopt in order to choose accurate algorithms and features that significantly influence the predicted variable.*

Keywords Feature selection, AutoML, Random forest, XGBoost

1. Introduction

Feature selection is one of the most important tasks and a core concept in machine learning, specifically in predictive models. Using irrelevant features when training a model may affect the performance of the model, reduce accuracy and cause overfitting. By choosing wisely the best and most significant features from the data when building the model, one can avoid overfitting, improve prediction accuracy and reduce the training time. Feature selection has been studied widely in the literature, see e.g. [3], [11], [19], [17], [30], [33], and many references therein. Feature selection is applied to many fields, such as statistical pattern recognition [2], [25], [15]; face recognition [23]; data mining and machine learning [18], [27], [9], [32]; text categorization [29]; customer relationship management [4]; bioinformatics [28]; genomics [1], cross-project defect prediction [24], and more. Furthermore, in [14], the authors provide a comprehensive survey on online feature selection with streaming features, i.e., when features are generated dynamically.

Feature selection methods are mainly divided into filter methods, wrapper methods and embedded methods. Filter methods use variable ranking techniques, and some ranking criteria to decide whether a variable should be removed from the model or not. In wrapper methods, a subset of features is evaluated using a machine learning algorithm that employs a search strategy to look through the space of possible feature subsets. Each subset is evaluated based on the quality of the performance of a given algorithm. Embedded methods perform feature selection during the modeling algorithm's execution. For a review of these methods, see [6].

In this paper, we present an automated feature selection mechanism. After receiving the data, the mechanism first executes two feature selection methods, random forest [5] and XGBoost [7]. Then, according to each method, it determines the importance of each feature and, as a result, which features should be used in the model.

Automated Machine Learning (AutoML) is an artificial intelligence-based method whose purpose to automate the process machine learning by building efficient and high model quality machine learning algorithms. A recent comprehensive survey on AutoML can be found in [13] and references therein.

As mentioned, we focus in this paper on the random forest classifier and the XGBoost algorithm. In [22], the authors state that a feature selection based on the random forest classifier has been found to provide multivariate feature importance scores which are relatively cheap to obtain, and which have been successfully applied to high dimensional data. Random forest performs an implicit feature selection, using a small subset of ”useful variables” for the classification only. This provides, eventually, an indicator of feature relevance. XGBoost is a scalable machine learning system that is commonly applied in tree boosting [7]. In [31], the authors state that the XGBoost algorithm provides a trained predictive model that automatically provides the trained feature importance estimates. The XGBoost algorithm improves the performance of the model by alleviating the effects of redundant features and noise. Moreover, the algorithm prevents overfitting through feature subsampling or column subsampling.

Naturally, one of the most interesting issues when performing variable selection is accuracy, see [12]. That is, we are interested in whether the accuracy achieved from using all features in the machine learning model is significantly greater than the accuracy of the model with only the selected (most important) variables; Or, whether it is sufficient to use a small (but how small) number of features, and nevertheless achieve almost the same accuracy.

Our proposed automated mechanism performs the random forest and XGBoost algorithms iteratively. In each iteration, we keep the most important features according to their rank in the random forest and XGBoost classifier, and use only them when solving some given classification problem. We then calculate the accuracy of this model and compare it with the accuracy of the full model, i.e. a random forest or an XGBoost classifier with all features. In the following iteration, we add another feature to the model (according to the ranks of the features), and calculate its accuracy. This procedure stops when there is only a negligible difference between the accuracy of the full model (with all features) and the partial model (with only the selected features). The rest of the paper is organized as follows: In Section 2 we describe our algorithm, while in Section 3 we present the implementation steps. Results and comparisons between the random forest classifier and XGBoost algorithm are given in Section 4. Section 5 concludes the paper.

2. The Method

In this paper, we define an AutoML method which performs the procedure of automated feature selection and reduction. The underlying process is as follows:

1. Run a selected algorithm on a full dataset D, i.e. with all features (in this paper we apply both the random forest classifier and the XGBoost algorithm).
2. Let $AC(D) =$ the accuracy of step 1.
3. Use a well-defined features importance method $f(D)$ (in this paper we use random forest classifier as well as the XGBoost algorithm).
4. Sort the $f(D)$ features list by importance. Let $X_1(D)$ denote the first feature in the ordered features' list (i.e., the most ”important” feature), and let $X_n(D)$ denote the last feature in the ordered features' list (i.e., the most ”un-important” feature).
5. **Option A:** Backward approach, i.e., remove variables until accuracy between a full model and a partial model exceeds some pre-determined error, denoted by E. The main steps in this approach are:
   (a) let $n =$ number of features in the dataset D.
   (b) Omit $X_n(D)$ from dataset D, and create $D_{new} = D[-X_n(D)]$.
   (c) Run the selected algorithm from step 1 on D_{new}.
   (d) $AC(D_{ne}w) =$ the accuracy of step 5.A.c.
   (e) While $[AC(D) - AC(D_{new}) \leq E$ and $n > 0]$ do
      i. $n = n - 1$
      ii. $D_{new} = D_{new}[-X_n(D)]$
      iii. Run the selected algorithm on D_{new}
      iv. $AC(D_{new}) =$ the accuracy of step (e)iii.

   Option B: Forward approach, i.e., start with a model consisting only the predicted (dependent) variable, and add (independent) features to the model, as long as the difference between the accuracy of the full model and the partial model is greater than some error E. Once the difference is less than E, we stop and use the model with only the selected features. The main steps in this approach are:
   (a) let $n =$ number of features in the dataset D and let $b = 1$.
   (b) Create a new empty dataset D_{new} (which contains only the (single) dependent variable).
   (c) Add $X_1(D)$ to D_{new}.
   (d) Run the selected algorithm from step 1 on D_{new}.
   (e) Let $AC(D_{new}) =$ the accuracy of step 5.B.d
   (f) While $[AC(D) - AC(D_new) > E$ and $b < n]$ do
      i. $b = b + 1$

ii. $D_{new} = D_{new}[+X_b(D)]$
iii. Run the selected algorithm from step 1 on D_{new}
iv. $AC(D_{new})$ = the accuracy of step (f)iii.

Note that the parameter E determines a threshold level for error accuracy. It should be modified according to various factors and considerations, such as:

- The research domain (for example, in health care domain the prediction must be very high).
- Quality of the data (sample size, missing values, outliers, etc.).
- Use case analysis.
- Other statistical measures and factors (dependencies, multi-collinearity, bias, etc.).
- Model flexibility.

We present both the backward and the forward approaches, since one approach might be more suitable than the other, depending on the research domain. For example, if we assume that accuracy of 80% is sufficient, we can apply the forward approach, i.e. add features gradually to the model until this level of accuracy is achieved. On the other hand, if we are interested in reducing the number of features but maintain some minimum deviation from the accuracy of the full model, we will prefer the backward approach.

3. Implementation

To illustrate our suggested mechanism, we perform the following implementations procedures:

1. We test our mechanism on 4 different datasets, which are presented and detailed in the sequel. In each dataset, we solve some classification problem.
2. We use the Random Forest and XGBoost algorithms in two manners: (i) we use it for feature selection, and (ii) we use it as the prediction model for the classification problem, and calculate its accuracy. For that purpose, we utilize the libraries sklearn.ensemble.RandomForestClassifier , see [26] and xgboost import XGBClassifier.
3. We use pandas (see [21]) for handling with our datasets and derive statistical results and measures.
4. We test our suggested procedure on the following datasets:

    (a) Dataset 1: Wine Quality, see [8]. This dataset consists of 4898 records, 11 features, and a categorical target variable (with 11 different classes).
    (b) Dataset 2: The Cleveland Heart Disease Dataset, see [16]. We used the processed.cleveland.data dataset which contain 303 records with total of 14 features including the classification target (with 5 classes).
    (c) Dataset 3: breast-cancer-Wisconsin, see e.g. [20]. This dataset consists of 699 records, 10 features, and a categorical target variable (with 2 classes).
    (d) Dataset 4: Internet Firewall (see e.g. [10]). This dataset consists of 65532 records, 12 features including the classification categorical target variable (with 4 different classes).

5. We implemented both backward and forward approaches (as described in Section 2) on each of the selected datasets detailed above. For each dataset, we provide the following results:
    (a) Feature importance sorted list derived from the random forest classifier and from the XGBoost algorithm.
    (b) A comparative accuracy graph per model of backward approach.
    (c) A comparative accuracy graph per model of forward approach.

At the end of the process, our procedure returns the best model with the optimal number of features selected for each dataset. The flow of the AutoML implementation steps is described in Figure 1. We start our implementation by splitting the data into a training set and a test set. Then, we run the random forest algorithm and generate the feature importance list. If the generated list is not empty, we drop one feature and rerun the algorithm for the new list. We then calculate the accuracy and save it in the algorithms feature accuracy list. We compile the random forest feature accuracy list if the importance is not greater than zero. Furthermore, we perform successive iterations for the procedure using the XGBoost algorithm, and compare the accuracy obtained by using the features from the two lists. The final output is the accuracy needed alongside the optimal number of features.

4. Results

In this section we present the results of our suggested mechanism, for each of the 4 datasets described in Section 2.

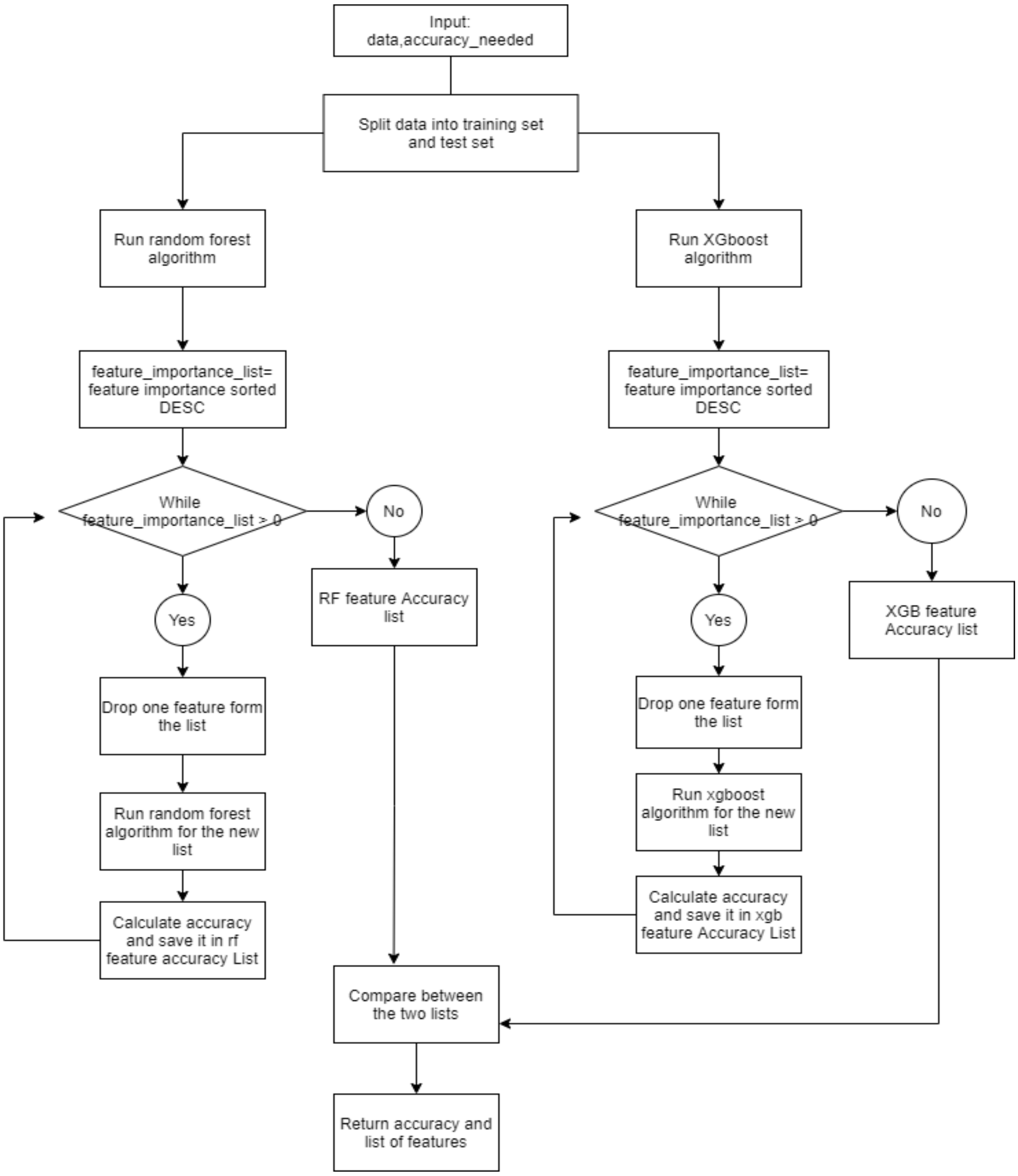
Input:
data,accuracy_needed
Split data into training set and test set
Run random forest algorithm
Run XGboost algorithm
feature_importance_list= feature importance sorted DESC
feature_importance_list= feature importance sorted DESC
While feature_importance_list > 0
No
While feature_importance_list > 0
No
Yes
RF feature Accuracy list
Yes
XGB feature Accuracy list
Drop one feature form the list
Drop one feature form the list
Run random forest algorithm for the new list
Run xgboost algorithm for the new list
Calculate accuracy and save it in rf feature accuracy List
Calculate accuracy and save it in xgb feature Accuracy List
Compare between the two lists
Return accuracy and list of features

Figure 1. Flow chart of the implementation steps

4.1. Dataset 1 - wine quality dataset

Table 1 presents the sorted feature importance list, based on the outcomes of both the Random Forest algorithm, and the XGBoost algorithm. The results for dataset 1 show that the accuracy of a full random forest model (consisting all features) is 0.6020, while the accuracy of the full XGBoost model is 0.6562. Figure 2 presents the accuracy of the fitted random forest and XGBoost models under the backward approach. That is, we start with a full model with all 11 features and then remove features, according to their importance given in Table 1. In this case, it is evident that reducing the number of features to only 5 (out of 11) does not dramatically influence the accuracy of the model. However, it is shown in Figure 2 that accuracy obtained from the XGBoost method is better than the accuracy of random forest. Figure 3 depicts the accuracy for the forward approach. We start with a model consisting only the most important feature, which, for both random forest and XGBoost, results in low accuracy of about 0.51. We then add features according to their importance, until reaching the desired accuracy. Again, a good accuracy is reached with only 5 features. Both Figures 2 and 3 show that for dataset 1, the XGBoost model provides better accuracy than the random forest classifier.

Table 1

Feature importance for dataset 1 according to random forest and XGBoost

Feature name	Importance random forest	Feature name	Importance XGBoost
Alcohol	0.242851	Alcohol	0.201177
Sulphates	0.140236	Total sulfur dioxide	0.105005
Total sulfur dioxide	0.115642	sulphates	0.101907
Volatile acidity	0.111605	Volatile acidity	0.09821
Density	0.092982	Free sulfur dioxide	0.07577
Chlorides	0.057417	Fixed acidity	0.075138
Citric acid	0.053522	PH	0.074227
Fixed acidity	0.052005	Residual sugar	0.072228
PH	0.045732	Citric acid	0.065855
Residual sugar	0.044457	Density	0.065293
Free sulfur dioxide	0.043558	Chlorides	0.06519

4.2. Dataset 2 - the Cleveland heart disease dataset

Table 2 presents the results of the feature importance process, executed on dataset 2, obtained by random forest and XGBoost. Note that the accuracy of the full model according to random forest is 0.5604, and 0.4945 via XGBoost. It is shown in Figure 4 that according to random forest classifier, eliminating variables from the model increases accuracy. This often occurs, since having many variables in the model may cause overfitting and increase the variance. It appears that a model with 2 features reaches the best accuracy when using

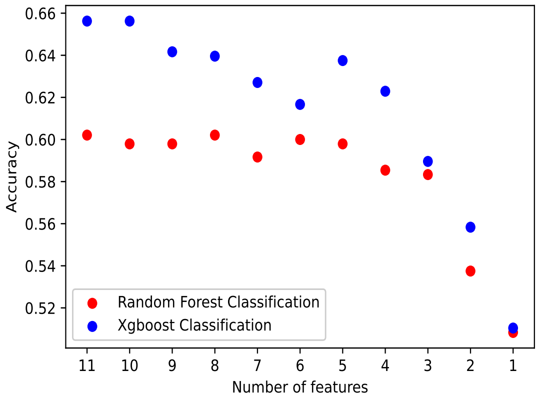

Figure 2. Model accuracy of the backward approach for dataset 1

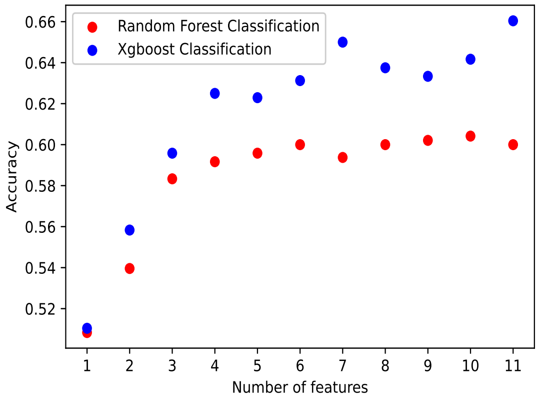

Figure 3. Model accuracy of the forward approach for dataset 1

random forest, and 4 features when using XGboost. This is also shown in Figure 5, where the accuracy is given for the forward approach, i.e. when adding features. A model with a single independent feature gives poor accuracy with random forest, but, surprisingly, using a single feature does not give the worst accuracy when using XGboost. Adding only a single extra feature to the model with random forest significantly improves accuracy, while in the XGboost model the accuracy rises in a more moderate manner. Overall, it is evident from Figures 4 and 5 that random forest results with better accuracy for dataset 2.

Table 2

Feature importance for dataset 2 according to random forest and XGBoost

Feature name	Importance random forest	Feature name	Importance XGBoost
feature 2	0.177796	feature 11	0.153479
feature 11	0.146085	feature 2	0.143723
feature 1	0.139925	feature 1	0.12426
feature 6	0.101466	feature 5	0.083033
feature 4	0.098772	feature 3	0.076648
feature 5	0.082157	feature 4	0.064499
feature 13	0.079420	feature 12	0.063392
feature 3	0.045154	feature 16	0.059724
feature 9	0.042083	feature 8	0.058084
feature 12	0.037607	feature 13	0.052149
feature 10	0.035161	feature 9	0.048032
feature 7	0.013243	feature 10	0.044038
feature 8	0.001131	feature 7	0.028939

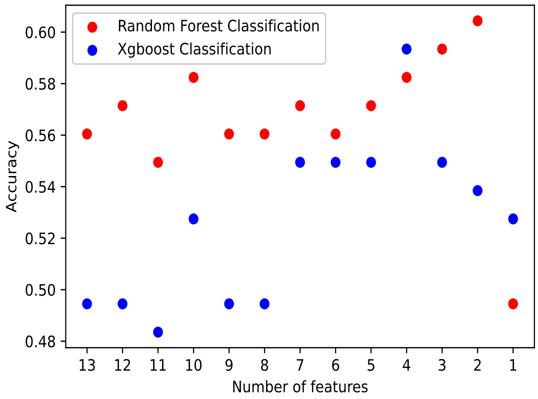

Figure 4. Model accuracy of the backward approach for dataset 2

4.3. Dataset 3 - breast-cancer-Wisconsin dataset

For the breast-cancer dataset, we present the order of feature importance in Table 3. The accuracy of a full Random Forest model and a full XGBoost model is about 0.9714 (both are very close). According to the backward approach, it is depicted in Figure 6 that a model with 3 features (out of 10),

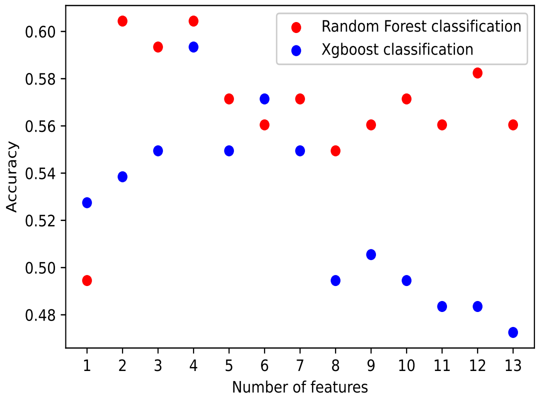

Figure 5. Model accuracy of the forward approach for dataset 2

reaches a very good accuracy for the random forest classifier (almost as good as for the full model), while 6 features provide good accuracy in the XGBoost model. This phenomenon is also presented in the lower part of Figure 7, in which the accuracy for the forward approach is shown.

Table 3

Feature importance for dataset 3 according to random forest and XGBoost

Feature name	Importance random forest	Feature name	Importance XGBoost
feature 7	0.256161	feature 7	0.565556
feature 8	0.233745	feature 8	0.231707
feature 4	0.155182	feature 3	0.056456
feature 3	0.128431	feature 4	0.050602
feature 5	0.092941	feature 2	0.044598
feature 2	0.080215	feature 9	0.024274
feature 9	0.034542	feature 6	0.010903
feature 6	0.015189	feature 5	0.010789
feature 10	0.002378	feature 10	0.005116
feature 1	0.001223	feature 1	0

4.4. Dataset 4 - internet firewall dataset

In the last example we consider the firewall data set. Feature importance is given in Table 4. The accuracy of a full Random Forest model with all 11 features is 0.9984 and for XGBoost is 0.9986. However, our results in Figures

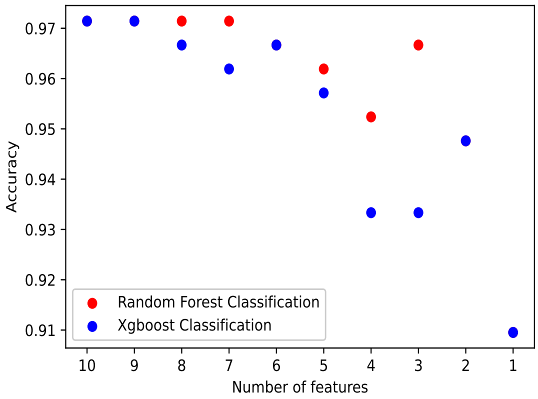

Figure 6. Model accuracy of the backward approach for dataset 3

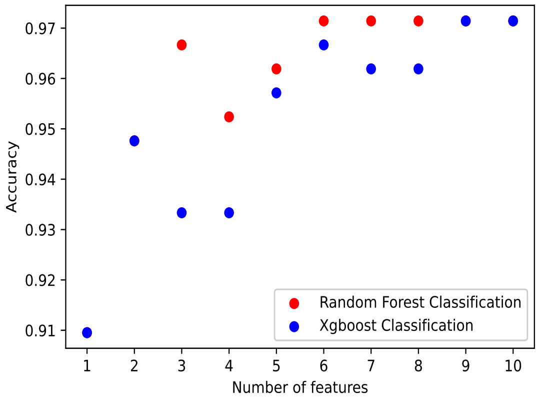

Figure 7. Model accuracy of the forward approach for dataset 3

8 and 9 show that even a model with only 2 features reaches almost the same accuracy, either by using random forest or XGBoost.

5. Concluding remarks

In this paper we presented an automated feature importance method based on random forest and XGBoost algorithms. For a given dataset, the proposed mechanism suggests which features should be used in the model and which

Table 4
Feature importance for dataset 4 according to random forest and XGBoost

Feature name	Importance random forest	Feature name	Importance XGBoost
Destination Port	0.225071	Elapsed Time	0.793872
Elapsed Time	0.192756	Destination Port	0.083326
NAT Source Port	0.144005	Bytes	0.077337
NAT Destination Port	0.120887	Packets	0.03966
Packets	0.074335	NAT Source Port	0.00164
Bytes	0.065065	Bytes Received	0.001225
pkts received	0.050558	Bytes Sent	0.001096
Bytes Sent	0.046179	NAT Destination Port	0.001009
Source Port	0.040731	Source Port	0.000374
Bytes Received	0.038632	pkts received	0.000265
pkts sent	0.001781	pkts sent	0.000197

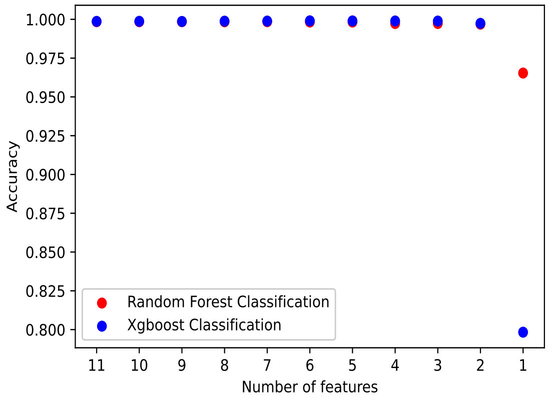

Figure 8. Model accuracy of the backward approach for dataset 4

should be omitted from it, while maintaining high accuracy. Reducing the number of features may reduce the complexity of the model, and, as shown in our examples, does not influence drastically on performance (i.e. model accuracy). Specifically, we tested our method on 4 different datasets, by solving some classification problem. For each dataset, we first performed the random forest and the XGBoost algorithm to derive feature importance. Then, according to the importance of features, we employed the backward approach (i.e., start with a full model and remove variables according to some accuracy criteria) and the forward approach (start with an empty model and add variables according to some pre-determined criteria). The measured accuracy is referred

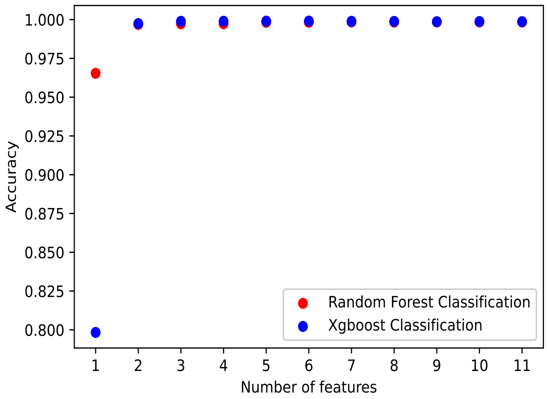

Figure 9. Model accuracy of the forward approach for dataset 4

to a classification model, which we conducted using random forest. We concluded, in all datasets, that the number of features used for building the model may be reduced by half (and even by more than that), while keeping model accuracy very close to the accuracy of a full model (with all features). The results also show that for some datasets the random forest classifier outperforms XGBoost (datasets 2 and 3), while for dataset 1 the XGBoost gives better accuracy. For dataset 4, both methods yield quite the same accuracy, except for the case when only a single feature is used. In this case, random forest is better. This auto-feature selection method presents an effective process of selecting the optimal number of features for predictive machine learning models, thus enhancing the accuracy of the fit. Implementing a machine learning model with the appropriate features increases the model's performance and reduces the computational costs. Overall, the method is efficient and states which features strongly influence the response variable.

References

[1] Alexe G., Alexe S., Hammer P.L., Vizvari B.: Pattern-based feature selection in genomics and proteomics. In: *Annals of Operations Research*, vol. 148(1), pp. 189–201, 2006.

[2] Ben-Bassat M.: Pattern recognition and reduction of dimensionality. In: *Handbook of Statistics*, vol. 2(1982), pp. 773–910, 1982.

[3] Blum A.L., Langley P.: Selection of relevant features and examples in machine learning. In: *Artificial Intelligence*, vol. 97(1-2), pp. 245–271, 1997.

[4] Bobrowski L.: Feature selection based on some homogeneity coefficient. In: *9th International Conference on Pattern Recognition*, pp. 544–545. IEEE Computer Society, 1988.

[5] Breiman L.: Random forests. In: *Machine Learning*, vol. 45(1), pp. 5–32, 2001.

[6] Chandrashekar G., Sahin F.: A survey on feature selection methods. In: *Computers & Electrical Engineering*, vol. 40(1), pp. 16–28, 2014.

[7] Chen T., Guestrin C.: Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. 2016.

[8] Cortez P., Cerdeira A., Almeida F., Matos T., Reis J.: Modeling wine preferences by data mining from physicochemical properties. In: *Decision Support Systems*, vol. 47(4), pp. 547–553, 2009.

[9] Dy J.G., Brodley C.E.: Feature selection for unsupervised learning. In: *Journal of Machine Learning Research*, vol. 5(Aug), pp. 845–889, 2004.

[10] Ertam F., Kaya M.: Classification of firewall log files with multiclass support vector machine. In: *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pp. 1–4. IEEE, 2018.

[11] Guyon I., Elisseeff A.: An introduction to variable and feature selection. In: *Journal of Machine Learning Research*, vol. 3(Mar), pp. 1157–1182, 2003.

[12] Haury A.C., Gestraud P., Vert J.P.: The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. In: *PLOS ONE*, vol. 6(12), p. e28210, 2011.

[13] He X., Zhao K., Chu X.: AutoML: A Survey of the State-of-the-Art. In: *Knowledge-Based Systems*, vol. 212, p. 106622, 2021.

[14] Hu X., Zhou P., Li P., Wang J., Wu X.: A survey on online feature selection with streaming features. In: *Frontiers of Computer Science*, vol. 12(3), pp. 479–493, 2018.

[15] Jain A., Zongker D.: Feature selection: Evaluation, application, and small sample performance. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19(2), pp. 153–158, 1997.

[16] Janosi A., Steinbrunn W., Pfisterer M., Detrano R.: UCI machine learning repository-heart disease data set. In: *School Inf. Comput. Sci., Univ.*

California, Irvine, CA, USA, 1988.

[17] Liu H., Motoda H.: *Feature selection for knowledge discovery and data mining*, vol. 454. Springer Science & Business Media, 2012.

[18] Liu H., Setiono R.: Feature Selection and Classification-A Probabilistic Wrapper Approach. In: *Proceedings of 9th International Conference on Industrial and Engineering Applications of AI and ES*, pp. 419–424. 1997.

[19] Liu H., Yu L.: Toward integrating feature selection algorithms for classification and clustering. In: *IEEE Transactions on Knowledge and Data Engineering*, vol. 17(4), pp. 491–502, 2005.

[20] Mangasarian O.L., Wolberg W.H.: Cancer diagnosis via linear programming. Tech. rep., University of Wisconsin-Madison Department of Computer Sciences, 1990.

[21] McKinney W., et al.: Data structures for statistical computing in python. In: *Proceedings of the 9th Python in Science Conference*, vol. 445, pp. 51–56. Austin, TX, 2010.

[22] Menze B.H., Kelm B.M., Masuch R., Himmelreich U., Bachert P., Petrich W., Hamprecht F.A.: A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. In: *BMC Bioinformatics*, vol. 10(1), p. 213, 2009.

[23] Moujahid A., Dornaika F.: Feature selection for spatially enhanced LBP: application to face recognition. In: *International Journal of Data Science and Analytics*, vol. 5(1), pp. 11–18, 2018.

[24] Ozturk M.M.: complexFuzzy: A novel clustering method for selecting training instances of cross-project defect prediction. In: *Computer Science*, vol. 22(1), 2021.

[25] Parsons L., Haque E., Liu H.: Subspace clustering for high dimensional data: a review. In: *Acm Sigkdd Explorations Newsletter*, vol. 6(1), pp. 90–105, 2004.

[26] Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., et al.: Scikit-learn: Machine learning in Python. In: *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[27] Reinartz T.: A unifying view on instance selection. In: *Data Mining and Knowledge Discovery*, vol. 6(2), pp. 191–210, 2002.

[28] Saeys Y., Inza I., Larrañaga P.: A review of feature selection techniques in bioinformatics. In: *Bioinformatics*, vol. 23(19), pp. 2507–2517, 2007.

[29] Swets D.L., Weng J.J.: Efficient content-based image retrieval using automatic feature selection. In: *Proceedings of International Symposium on Computer Vision-ISCV*, pp. 85–90. IEEE, 1995.

[30] Tang J., Alelyani S., Liu H.: Feature selection for classification: A review. In: *Data Classification: Algorithms and Applications*, p. 37, 2014.

[31] Wang Y., Ni X.S.: A XGBoost risk model via feature selection and Bayesian hyper-parameter optimization. In: *International Journal of Database Management Systems*, vol. 11(1), pp. 1–17, 2019.

[32] Xiang S., Nie F., Meng G., Pan C., Zhang C.: Discriminative least squares regression for multiclass classification and feature selection. In: *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23(11), pp. 1738–1754, 2012.

[33] Zhang R., Nie F., Li X., Wei X.: Feature selection with multi-view data: A survey. In: *Information Fusion*, vol. 50, pp. 158–167, 2019.

Affiliations

No Authors... no affiliations... ???

Received: ???
Revised: ???
Accepted: ???