Journal of Intelligent Systems and Internet of Things (JISIoT)
Vol. 11, No. 01, PP. 44-54, 2024
DOI: https://doi.org/10.54216/JISIoT.110105

Forward feature selection: empirical analysis

Firuz Kamalov[1,*], **Said Elnaffar**[2], **Aswani Cherukuri**[3], **Annapurna Jonnalagadda**[4]
[1]Department of Electrical Engineering, Canadian University Dubai, Dubai, UAE
[2] Department of Computer Science, Canadian University Dubai, Dubai, UAE
[3] School of Information Systems, Vellore Institute of Technology, India
[3] School of Computer Science and Engineering, Vellore Institute of Technology, India
Emails: firuz@cud.ac.ae; said.elnaffar@cud.ac.ae; cherukuri@acm.org; jannapurna@gmail.com
[*] Corresponding author: firuz@cud.ac.ae.

Abstract

Feature selection is an important preprocessing step in many data science and machine learning applications. Although there exist several sophisticated feature selection algorithms, their benefits are sometimes overshadowed by their complexity and slow execution. Therefore, in many cases, a more simple algorithm may be better suited. In this paper, we demonstrate that a rudimentary forward selection algorithm can achieve optimal performance with a low time complexity. Our study is based on an extensive empirical evaluation of the forward feature selection algorithm in the context of linear regression. Concretely, we compare the forward selection algorithm against the gold standard exhaustive search algorithm based on several datasets. The results show that the forward selection algorithm achieves high performance with relatively fast execution. Given the simplicity, accuracy, and speed of the forward feature selection algorithm, we recommend it as a primary feature selection method for most regression applications. Our results are particularly pertinent in the case of big data and real-time analysis.

Received: March 19, 2023 Revised: July 26, 2023 Accepted: November 27, 2023

Keywords: data transformation; data mining; standardization

1 Introduction

Feature selection is an important preprocessing step in many machine learning applications particularly in the case of high dimensional data. To learn effectively from high dimensional data requires a large amount of data which is often unavailable. For instance, a dataset with only 10 binary attributes has 1024 possible feature combinations. Assuming that each feature combination requires at least 10 samples to get a significant result, we need a total of approximately 10,000 samples in the dataset. A dataset containing categorical features with more than 2 classes or continuous features would require exponentially more data. Furthermore, feature selection leads to a more simple model that is more interpretable. Therefore, identifying and selecting the most relevant features in data plays an important role in data analysis.

Given the importance of feature selection, there exists a vast amount of literature devoted to the subject. The majority of the existing algorithms involve multi-step heuristics that are both computationally expensive and challenging to implement for a nonexpert. Although several proposed algorithms have shown robust results, their benefits should be weighted against the issues related to implementation and time complexity. As a result, in many cases, it may be advantageous to utilize a simple selection algorithm over a more complicated alternative.

In this paper, we investigate the performance a simple forward feature selection algorithm against the benchmark exhaustive search selection algorithm in the context of linear regression. The forward selection algorithm

iteratively adds a new feature to the optimal subset. As a result, the forward selection algorithm operates at $\mathcal{O}(K^2)$ time complexity. On the other hand, the exhaustive search method considers all the possible subsets of size k which is prohibitively expensive.

To measure the performance of the forward selection method, we conduct a range of numerical experiments based on several datasets. For the sake of efficiency, the experiments are done in the context of linear regression though its outcomes can be reasonably extended to other statistical and machine learning models. The results show that a basic forward selection algorithm is capable of achieving high performance at a comparatively little time. In addition, the simplicity and ubiquity of the algorithm allows it to be quickly implemented in any context. Our study indicates that the forward selection algorithm should be considered as a primary feature selection method in regression tasks.

The paper is structured as follows. Section 2 presents a short overview of the existing literature on the topic of feature selection. Section 3 provides the description of the feature selection algorithms considered in the paper. Section 4 describes the methodology. In Section 5, we present and analyze the results of the numerical experiments. And Section 6 concludes the paper with closing remarks. Identifying the relevant features in data is a key part of machine learning tasks. As such, it has attracted a significant amount of attention from research community [1]. Feature selection algorithms can be divided into three major categories: 1) filter methods, 2) wrapper methods, and 3) embedded methods. Filter methods employ a uniform metric to evaluate and rank individual features [2,3]. Wrapper methods such as recursive feature elimination utilize a base model to compute feature coefficients which are used to evaluate feature importance [4, 5]. Embedded methods such as Lasso regression perform automatic feature selection during the model training stage [6, 7].

Feature selection algorithms have steadily evolved in their complexity. Algorithm complexity has progressed in two major directions: 1) hybrid methods and 2) novel optimization techniques. Hybrid methods attempt to combine several techniques to achieve better performance. By leveraging the strengths of individual techniques, hybrid approaches attempt to improve the overall performance. In hybrid approaches, different methods are combined either through a bagging procedure where the results of several individual methods are combined into the final optimal subset via a voting system or a multi-stage procedure where several methods are applied sequentially to produce the final subset. As an example of a bagging procedure, the authors in [8] combined genetic algorithm and ReliefF to carry out feature selection for wheat yield prediction. Similarly, information gain and Fisher score were used together to reduce the initial set of features [9]. In [10], the authors ensembled 5 different filter methods in a bagging procedure to improve selection stability.

As an example of a multi-stage procedure, the authors in [11] first combined information gain and random forest to reduce the feature subset search space, and then utilized recursive feature elimination (RFE) to further narrow down the optimal subset. Similarly in [5], the authors first employed a random forest-based filter to reduce the initial feature set, then applied RFE to gradually eliminate the irrelevant features. In [12], the authors initially ranked features using XGBoost, then utilized a genetic algorithm to search through a reduced feature space.

Novel optimization techniques have been frequently employed to search through the space of feature subsets to find the optimal subset [13]. Various optimization techniques such Gorilla Troops Optimizer [14], Dynamic Butterfly Optimizer [15, 16], and others have been utilized in feature selection. Recently, feature selection based on deep learning has gained popularity. The layered structure of neural networks facilitates the learning of features during the training process. It enables both feature extraction and feature selection given the appropriate activation functions [17–20]. Despite the effectiveness of deep learning-based feature selection algorithms they are computationally intensive and require significant execution times.

2 Feature selection algorithms

In this section, we describe the feature selection methods considered in our study - forward selection, backward selection, and the exhaustive search. We provide the details of the algorithms and discuss their time complexity.

The basic forward feature selection algorithm is based on a greedy iterative feature selection process. Given a subset S_k of k selected features, the next feature $f_{k+1} \notin S_k$ is chosen so that the set $S_{k+1} = S_k \cup f_{k+1}$ achieves the best performance. In the present context, the subset performance is measures by the accuracy of the linear regression fit based on the data from the selected features. At each iteration, only $K-k$ features are considered, where K in the total number of features. It follows that the number of subsets required to evaluate in order to obtain the optimal subset of size k is

$$K + (K-1) + \cdots + (K-k+1) = \frac{K(K+1) - (K-k)(K-k+1)}{2} = \frac{k(2K-k+1)}{2}. \tag{1}$$

It can be seen from the above equation that the algorithm complexity is quadratic in the number of selected features k. However, note that the number of features in each subset is small which requires less computational capacity. For instance, in the initial iteration only subsets of size 1 are evaluated. In the second iteration, only the subsets of size 2 are considered. Calculating the linear regression model on 2 features is relatively fast.

In the exhaustive search algorithm - as the name suggests - all the subsets of a fixed size are considered. The number of subsets of size k searched in the exhaustive algorithm is $\binom{K}{k}$. Since the exhaustive algorithm considers *all* the possible subsets it provides the gold standard. On the other hand, it is prohibitively expensive. The number of subsets considered grows very quickly as k increases.

The backward feature selection iteratively removes features from the original set to obtain the optimal subset. Given a subset S_k of k selected features, the next feature $f_{k+1} \in S_k$ is chosen so that the set $S_{k+1} = S_k - f_{k+1}$ achieves the best performance. Thus, at each iteration, k features are considered. The number of subsets that is required to evaluate to obtain the solution is similar to the forward selection. However, the number of features in each subset is generally greater in backward selection. Since it is computationally less expensive to build regression models with fewer features, forward selection is ultimately more efficient than backward selection.

It is important to note that both the forward and backward selection methods can potentially miss the best overall subset. Since forward selection operates in a greedy fashion, the optimal feature combination may be omitted. Nevertheless, in practice, the best subset produced by the forward selection is often very similar to the one produced by the exhaustive search. Thus, given the speed of the forward selection, it provides a practical tool for feature selection.

3 Methodology

Our evaluation of the forward selection algorithm is based on its performance against the benchmark selection algorithms - exhaustive search and backward selection. The empirical evaluation is conducted using several datasets. For each dataset, we perform a two-part experiment. In the first part, the entire dataset is utilized for feature selection and the corresponding performance metrics are calculated. In the second part, we use bootstrap cross-validation to calculate the test MSE for each selection algorithm at various subset sizes. The two approaches combined provide a comprehensive analysis of the performances of feature selection methods.

The first part of the experiment involves employing the entire dataset to perform feature selection. For a given subset size k, we determine the optimal subset using the feature selection algorithms considered in the study. The subsets are evaluated based on the fit of the corresponding linear regression model. Once the optimal subset of size k is identified by the selection algorithm we calculate three key metrics related to the performance of the optimal subset.

The second part of the experiment is based on bootstrap cross-validation. In particular, we sample a training set of size n from the original dataset (with replacement). Then, a feature selection method is used to identify the optimal subset of size k based on the training set. Finally, the test MSE is calculated as the average of the squared difference between the actual and predicted values of the unselected samples. The bootstrap simulation is carried out 50 times for each dataset. The results are presented in the form of a boxplot.

3.1 Data

The details of the datasets employed in the experiments are provided in Table 1. The fundraising dataset contains information from a direct-mail fundraising campaign by a non-profit organization. The two wine quality datasets contain physicochemical features of red and white wines. The student datasets contain information related to student characteristics and performance in mathematics and Portuguese language courses. The superconductor dataset contains features of superconductors and the critical temperature. All the dataset have potentially irrelevant variables. So feature selection is advisable to improve performance.

Table 1: Datasets used in the numerical experiments.

name	samples	features
fundraising [21]	3470	12
winequality-red [22]	1599	12
winequality-white [22]	4898	12
student-mat [23]	395	14
student-por [23]	649	14
superconductor [24]	10000	41

3.2 Performance metrics

The performance criteria for the optimal subset include the adjusted R^2, Bayesian information criterion (BIC), and the theoretical test MSE (C_p). The adjusted R^2 is a modification of the traditional R^2 with additional penalty for extra model parameters. It is given by the following equation

$$\text{Adjusted } R^2 = 1 - \frac{(n-1)SSR}{(n-k)SST}, \tag{2}$$

where SSR is the sum squared residuals, SST is the sum squared total, and n is the number of samples in the dataset. The BIC is another popular metric that is used to measure the predictive error and thereby the quality of a model. It uses information theory to evaluate the goodness of a model. The BIC is given by the following equation:

$$\text{BIC} = -2\ln(L(\theta|D)) + k\ln(n), \tag{3}$$

where $L(\theta|D)$ is the maximized likelihood function. Finally, C_p is the asymptotic test MSE of the model. It represents the performance of the model on unseen data. The C_p is given by the following equation

$$C_p = \frac{1}{n}(SSR + 2k\hat{\sigma}). \tag{4}$$

4 Results and analysis

In this section, we discuss the results of the numerical experiments to compare the forward selection algorithm against the exhaustive search and backward selection. The results show the forward selection algorithm performs just as well as the exhaustive and backward selection methods. The forward selection algorithm shows robust performance as measured by the adjusted R^2, BIC, and C_p. In addition, bootstrap cross-validation reveals that forward selection is robust against sample variations. The performance of forward selection is comparable to exhaustive search across the range of all datasets considered in this study.

4.1 Fundraising dataset

This dataset presents the problem of predicting dollar amount of donors' contributions from a direct mail campaign based on their demographics and history of past contributions. In Figure 1, we present the adjusted R^2, BIC, and C_p for the regression models that were fitted based on the selected subsets using the `fundraising` dataset. In particular, for each subset size k, the optimal subset was selected using 3 selection methods: backward, exhaustive, and forward. The subsets were selected based on all the samples corresponding to the subset features. As shown in Figure 1, the forward selection algorithm produces exactly the same performance as the exhaustive search method across all three evaluation criteria. Furthermore, the results are consistent across different number of features selected. In the adjusted R^2 plot, both forward selection and exhaustive search produce the same values and reaching the maximum performance at $k = 7$. Similarly, BIC and C_p graphs for forward selection and exhaustive search are identical with optimal values achieved at $k = 7$. Backward selection produces the same results except at $k = 4$ where it slightly underperforms. The results in Figure 1 show that the forward selection algorithm is capable of providing the same results as the gold standard exhaustive search method at a fraction of time complexity.

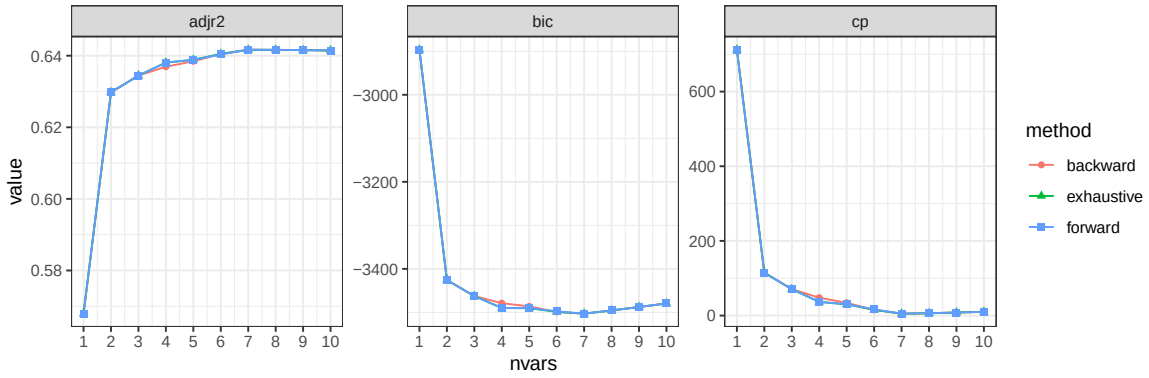

Figure 1: The results of feature selection using the entire `fundraising` dataset.

In Figure 2, we present the results of bootstrap cross-validation. In particular, we first select the optimal subset of features using the sampled train set and then calculate the corresponding regression MSE on the test set. The training and test subsets are selected via bootstrap method that is repeated 50 times. As shown in Figure 2, the MSE for forward selection is exactly the same as exhaustive search. The values are the same across different number of selected features with the lowest MSE achieved at $k = 7$ by both methods. It shows that over 50 repeated bootstrap samples forward selection and exhaustive search yield the same subsets. The results demonstrate the robustness of forward selection against different random samples.

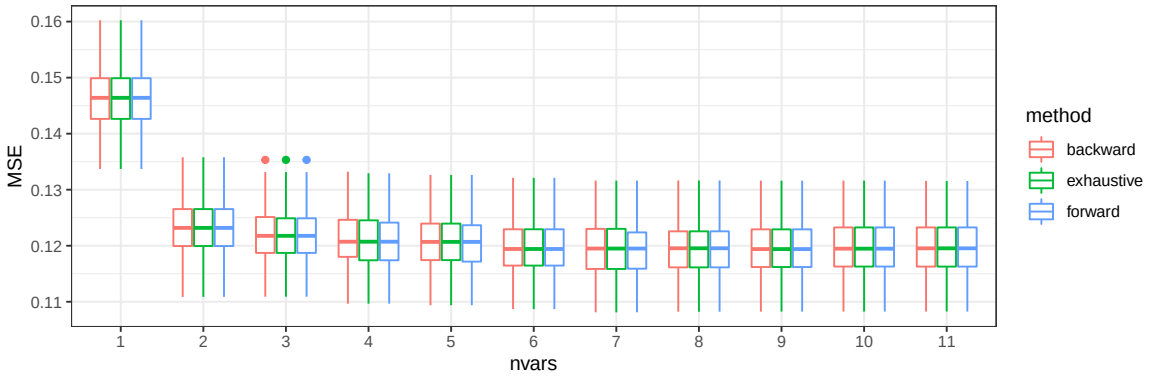

Figure 2: The results of feature selection using bootstrap cross-validation (`fundraising`).

4.2 Wine quality datasets

We consider two datasets related to the Portuguese "Vinho Verde" wine. The datasets contain physicochemical input and sensory output variables. In Figure 3, we present the results of the experiment using the `winequality-red` dataset. The results show that forward selection and exhaustive search produce identical values for all the evaluation criteria.

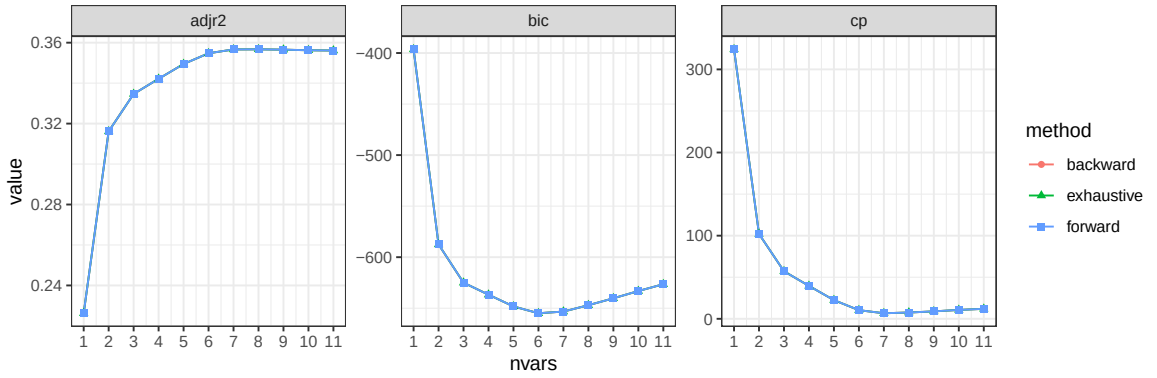

Figure 3: The results of feature selection using the entire `winequality-red` dataset.

In Figure 4, we present the outcome of bootstrap cross-validation. The optimal subset of features are selected based on the sampled train set and the MSE is calculated on the test set. As before, the bootstrap cross-validation is repeated 50 times. The results show that forward selection achieves the same or better performance than exhaustive search and backward selection. For instance, for $k = 8$ and $k = 9$ forward selection achieves lower MSE than the benchmark methods. Given that exhaustive search is the gold standard of feature selection, the performance of forward selection is remarkable.

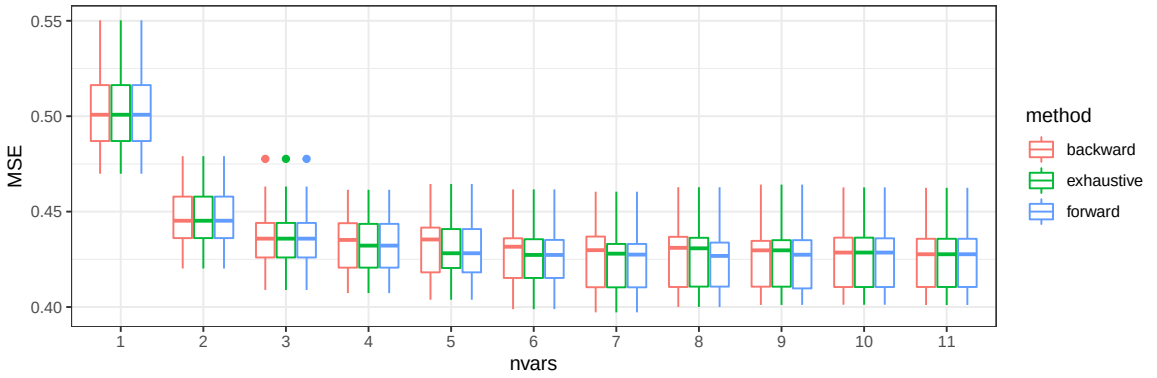

Figure 4: The results of feature selection using bootstrap cross-validation (`winequality-red`).

In Figure 5, we present the outcome of the experiment using the `winequality-white` dataset. The results show that forward selection produces the same results as exhaustive search except at $k = 5$ and $k = 6$. The optimal number of selected features is $k = 8$. The performance metrics at $k = 8$ are the same for all three selection methods. The results indicate that while forward selection slightly underperforms exhaustive search, it achieves overall similar results as the benchmark methods.

In Figure 6, we present the outcome of bootstrap cross-validation using the `winequality-white` dataset. The results show that in most cases forward selection performs identically to exhaustive search with exception of $k = 5$ and $k = 7$. At $k = 5$, forward selection underperforms exhaustive search, while at $k = 7$ it outperforms exhaustive search. We also note, as shown in Figure 6, that forward selection outperforms backward selection in several instances.

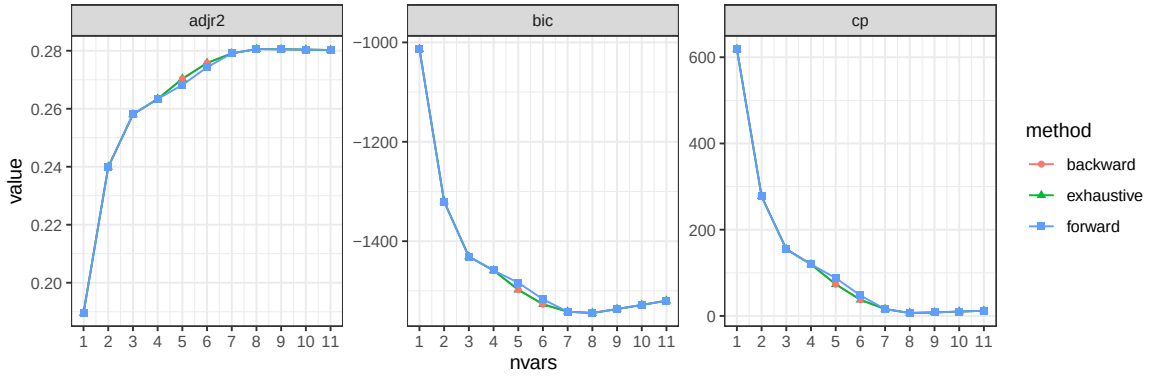

Figure 5: The results of feature selection using the entire `winequality-white` dataset.

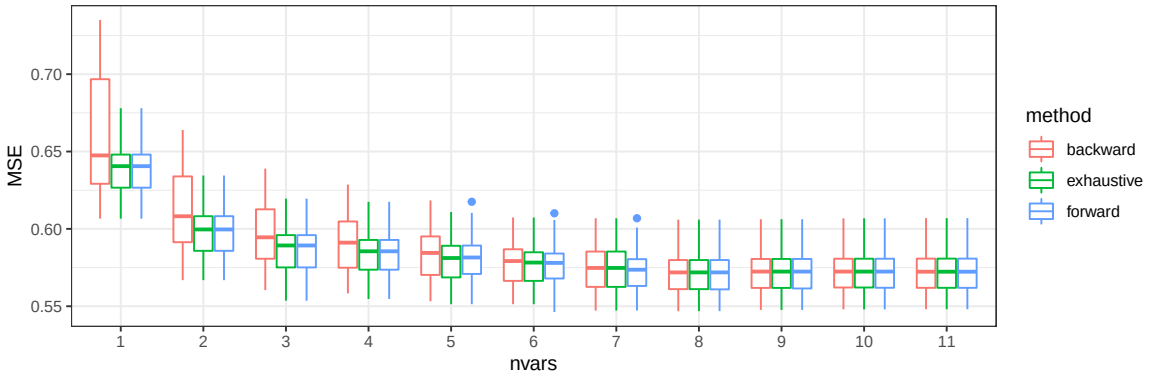

Figure 6: The results of feature selection using bootstrap cross-validation (`winequality-white`).

4.3 Mathematics dataset

The dataset is related to student performance in high school mathematics. The data attributes include student grades, demographic, social and school related features. In Figure 7, we present the outcome of the experiment using the `student-mat` dataset. The results show that forward selection achieves the same results as exhaustive search with only a few exceptions. We note that the optimal result is achieved at different values of k for different performance metrics. However, the corresponding metric values are the same for forward selection and exhaustive search. Similar to above, the experiments on the `student-mat` dataset demonstrate that while forward selection slightly underperforms exhaustive search, it achieves overall the same results.

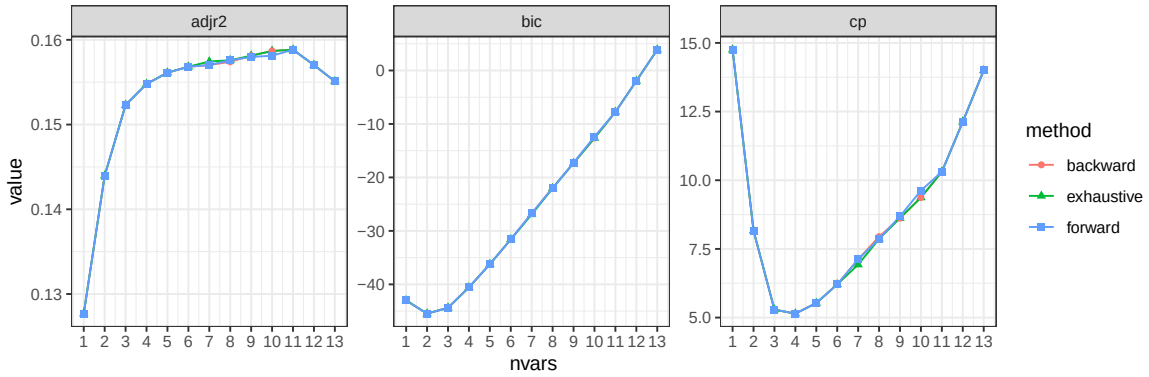

Figure 7: The results of feature selection using the entire `student-mat` dataset.

In Figure 8, we present the outcome of bootstrap cross-validation using the `student-mat` dataset. The

results show that forward selection slightly outperforms exhaustive search. In particular, for values $k = 4, 6, 7, 8, 10,$ and 11, the MSE for forward selection is lower than that of exhaustive search. It demonstrates that, over repeated sampling of the dataset, forward selection is capable of achieving higher accuracy than the benchmark methods.

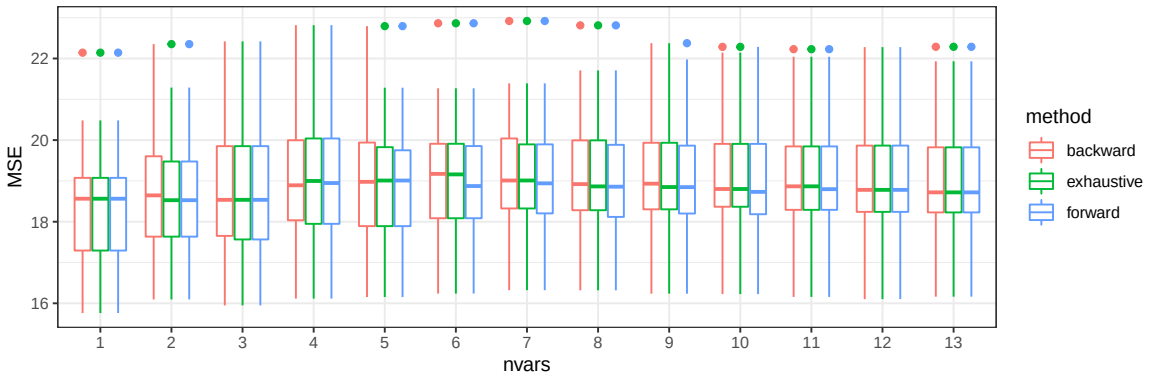

Figure 8: The results of feature selection using bootstrap cross-validation (`student-mat`).

4.4 Portugese dataset

This dataset is related to student performance in Portuguese language course. The data attributes are the same as in the mathematics dataset above. In Figure 9, we present the outcome of the experiment using the `student-por` dataset. The results show that forward selection achieves identical results as the benchmark methods.

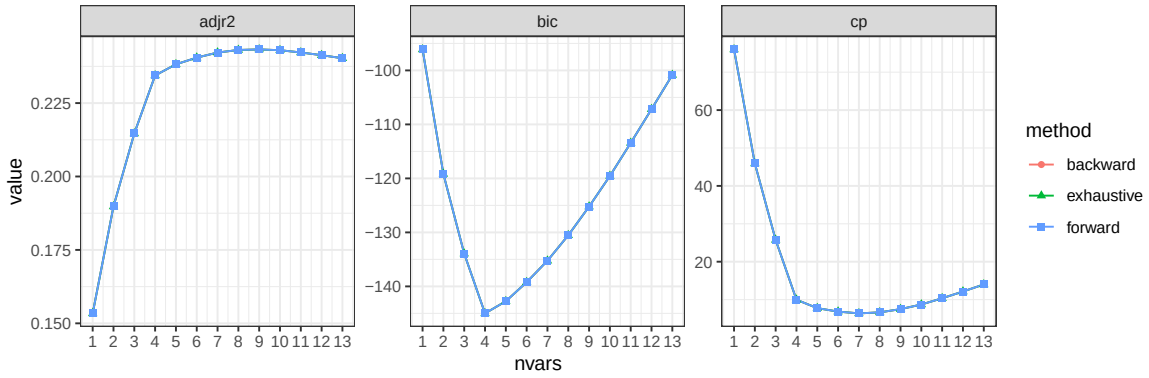

Figure 9: The results of feature selection using the entire `student-por` dataset.

In Figure 10, we present the outcome of bootstrap cross-validation using the `student-por` dataset. The results show that forward selection slightly outperforms the benchmark methods. In particular, at $k = 6, 7, 8,$ and 9, forward selection achieves the lowest MSE. However, we note that at $k = 4$ forward selection produces slightly higher MSE than the benchmarks. The experiment shows that on average forward selection achieves comparable, if not better, results than benchmark methods.

4.5 Superconductor dataset

This dataset contains features extracted from superconductors together with the critical temperature. The results of the experiments based on the `superconductor` dataset are presented in Figure 11. The results

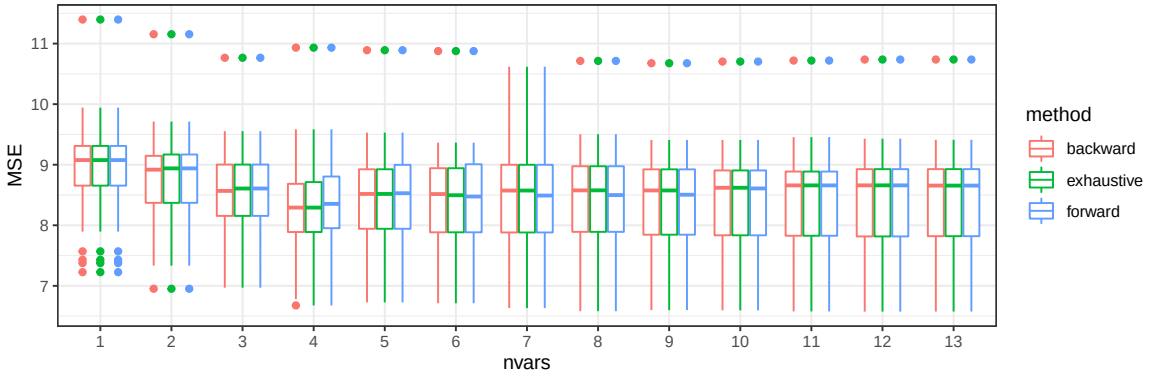

Figure 10: The results of feature selection using bootstrap cross-validation (`student-por`).

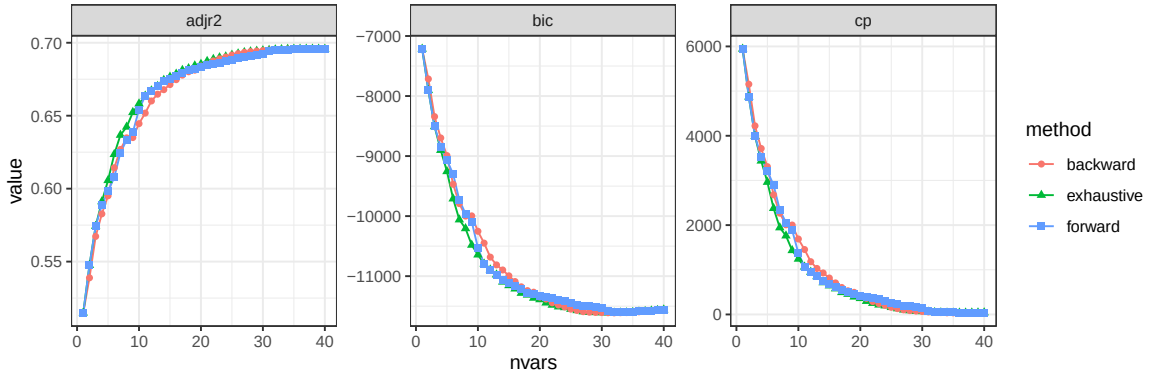

Figure 11: The results of feature selection using the entire `superconductor` dataset.

are mixed. The performance of forward selection compared to exhaustive search varies depending the subset size (nvars). However, the overall performance of all three feature selection algorithms is very similar.

In Figure 12, we present the outcome of bootstrap cross-validation using the `superconductor` dataset. The results show that while the exhaustive search produces the lowest MSE for values $k = 6$ to $k = 10$, forward selection produces the lowest MSE for values $k = 11$ to $k = 17$. For larger values of k the MSE is similar for all three methods.

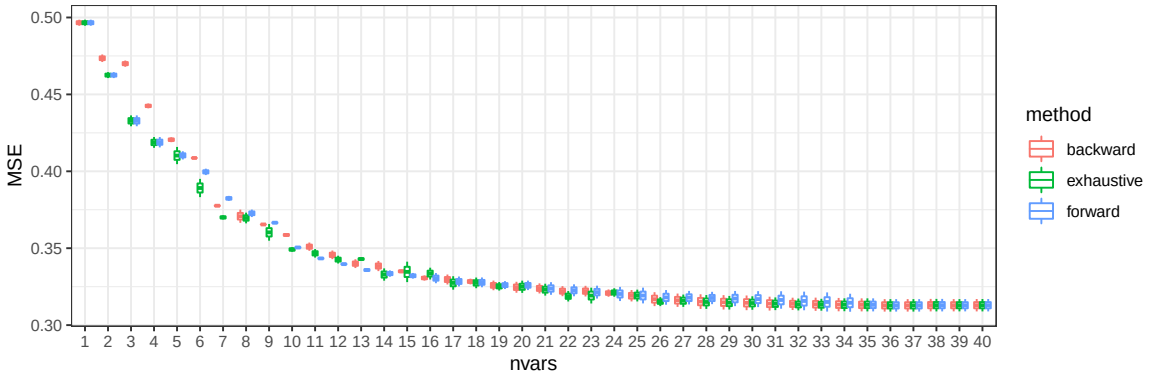

Figure 12: The results of feature selection using bootstrap cross-validation (`superconductor`).

5 Conclusion

Many existing feature selection methods are based on complex algorithms that require expert knowledge and significant computing power. In this study, we find that simple forward selection algorithm can achieve robust results at a fraction of the time required by more complex algorithms. We compare forward selection to exhaustive search and backward selection. The results show that the features identified by the forward selection algorithm produce the same accuracy as the exhaustive search.

Forward selection is simple to implement and requires minimal computing resources. Given its performance against the gold standard exhaustive search algorithm, it should be considered as a viable tool in feature selection. Our findings are particularly relevant in the case of big data and real-time analysis, where execution time plays a significant role.

References

[1] Dokeroglu, T., Deniz, A., & Kiziloz, H. E. (2022). A comprehensive survey on recent metaheuristics for feature selection. Neurocomputing, 494, 269-296.

[2] Pudjihartono, N., Fadason, T., Kempa-Liehr, A. W., & O'Sullivan, J. M. (2022). A review of feature selection methods for machine learning-based disease risk prediction. Frontiers in Bioinformatics, 2, 927312.

[3] Kamalov, F., Thabtah, F., & Leung, H. H. (2023). Feature selection in imbalanced data. Annals of Data Science, 10(6), 1527-1541.

[4] Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. Machine learning, 46, 389-422.

[5] Kamalov, F., Sulieman, H., Moussa, S., Reyes, J. A., & Safaraliev, M. (2023). Nested ensemble selection: An effective hybrid feature selection method. Heliyon, 9(9).

[6] Chen, H., Xu, K., Chen, L., & Jiang, Q. (2021). Self-Expressive Kernel Subspace Clustering Algorithm for Categorical Data with Embedded Feature Selection. Mathematics, 9(14), 1680.

[7] Gurrib, I., Kamalov, F., Starkova, O., Elshareif, E. E., & Contu, D. (2023). Drivers of the next-minute Bitcoin price using sparse regre

ssions. Studies in Economics and Finance.

[8] Kour, H., Pandith, V., Manhas, J., & Sharma, V. (2023). Machine Learning-Based Hybrid Model for Wheat Yield Prediction. Machine Intelligence, Big Data Analytics, and IoT in Image Processing: Practical Applications, 151-176.

[9] Li, M., Wang, H., Yang, L., Liang, Y., Shang, Z., & Wan, H. (2020). Fast hybrid dimensionality reduction method for classification based on feature selection and grouped feature extraction. Expert Systems with Applications, 150, 113277.

[10] Alelyani, S. (2021). Stable bagging feature selection on medical data. Journal of Big Data, 8(1), 1-18.

[11] Yin, Y., Jang-Jaccard, J., Xu, W., Singh, A., Zhu, J., Sabrina, F., & Kwak, J. (2023). IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset. Journal of Big Data, 10(1), 1-26.

[12] Deng, X., Li, M., Deng, S., & Wang, L. (2022). Hybrid gene selection approach using XGBoost and multi-objective genetic algorithm for cancer classification. Medical & Biological Engineering & Computing, 60(3), 663-681.

[13] Abu Khurma, R., Aljarah, I., Sharieh, A., Abd Elaziz, M., Damaševičius, R., & Krilavičius, T. (2022). A review of the modification strategies of the nature inspired algorithms for feature selection problem. Mathematics, 10(3), 464.

[14] Kareem, S. S., Mostafa, R. R., Hashim, F. A., & El-Bakry, H. M. (2022). An effective feature selection model using hybrid metaheuristic algorithms for iot intrusion detection. Sensors, 22(4), 1396.

[15] Tiwari, A., & Chaturvedi, A. (2022). A hybrid feature selection approach based on information theory and dynamic butterfly optimization algorithm for data classification. Expert Systems with Applications, 196, 116621.

[16] EL-Hasnony, I. M., Elhoseny, M., & Tarek, Z. (2022). A hybrid feature selection model based on butterfly optimization algorithm: COVID-19 as a case study. Expert Systems, 39(3), e12786.

[17] Afza, F., Sharif, M., Khan, M. A., Tariq, U., Yong, H. S., & Cha, J. (2022). Multiclass skin lesion classification using hybrid deep features selection and extreme learning machine. Sensors, 22(3), 799.

[18] Mahendran, N., & PM, D. R. V. (2022). A deep learning framework with an embedded-based feature selection approach for the early detection of the Alzheimer's disease. Computers in Biology and Medicine, 141, 105056.

[19] Satrya, G. B., Ramatryana, I. N. A., & Shin, S. Y. (2023). Compressive Sensing of Medical Images Based on HSV Color Space. Sensors, 23(5), 2616.

[20] Mohamed, T., Ibrahim, A., Faiz, T., Alhasan, W., Atta, A., Mago, V., ... & Munir, S. (2022, October). Intelligent Hand Gesture Recognition System Empowered With CNN. In 2022 International Conference on Cyber Resilience (ICCR) (pp. 1-8). IEEE.

[21] Flores, E. (n.d.). Direct-Mail Fundraising. RPubs. Retrieved June 7, 2022, from https://rpubs.com/elizabethfl/646805

[22] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. Decision support systems, 47(4), 547-553.

[23] Cortez, P., & Silva, A. M. G. (2008). Using data mining to predict secondary school student performance.

[24] Hamidieh, K. (2018). A data-driven statistical model for predicting the critical temperature of a superconductor. Computational Materials Science, 154, 346-354.