Finding Relevant Information in Big Datasets with ML

Uchechukwu F. Njoku
Universitat Politècnica de Catalunya
Barcelona, Spain
Université Libre de Bruxelles
Brussels, Belgium
uchechukwu.fortune.njoku@upc.edu

Alberto Abelló
Universitat Politècnica de Catalunya
Barcelona, Spain
alberto.abello@upc.edu

Besim Bilalli
Universitat Politècnica de Catalunya
Barcelona, Spain
besim.bilalli@upc.edu

Gianluca Bontempi
Université Libre de Bruxelles
Brussels, Belgium
gianluca.bontempi@ulb.be

ABSTRACT

Due to the abundance of data, noisy, irrelevant, or redundant features often need to be identified and discarded. Feature selection is a collection of methods used to ensure that only relevant data are used for a data analysis task. Extracting and using only useful data for analysis promotes model understanding and performance and reduces the model training time and variance, i.e., overfitting.

There is an abundance of methods for feature selection, and they can be categorised by various perspectives and are applicable to differing use cases. In this tutorial, we introduce the feature selection problem and present it from three perspectives of categorisation: *search strategy, model reliance, and relevance definition*. Furthermore, we propose a guideline for the use of the various methods. Lastly, we discuss current challenges and opportunities for research on feature selection.

1 INTRODUCTION

The increased interconnectivity and storage capabilities enable access to unprecedented amounts of data. Such data are the backbone of the recent advances in artificial intelligence and data science, such as smart cities [2] and data-driven decision-making. When building Machine Learning (ML) models, the bulk of the work lies in making the raw data suitable for use, known as data preprocessing, which consumes about 80% of the analysis time [20]. The first phase of data processing consists of data cleaning [7], dealing with missing and outlier values and data normalisation, among other issues. The outcome is a dataset that can already be used to build ML models. However, there could still be irrelevant or redundant features within this dataset [13], and one must be able to identify and select only the relevant features to the ML task at hand.

Feature Selection (FS) is a collection of techniques used to identify the subset of features that accurately describe the problem and incur minimum performance degradation. FS is important because redundant or irrelevant features within datasets often deteriorate data and model understanding, as well as lead to long training times and model overfitting due to high variance. Thus, FS has been shown to limit storage requirements, speed up the running time of learning algorithms, and improve data quality, model performance, data understanding, as well as model explainability [13]. It is, therefore, essential to understand the various FS techniques and how and when to apply them in order to make the best of available data when building ML models.

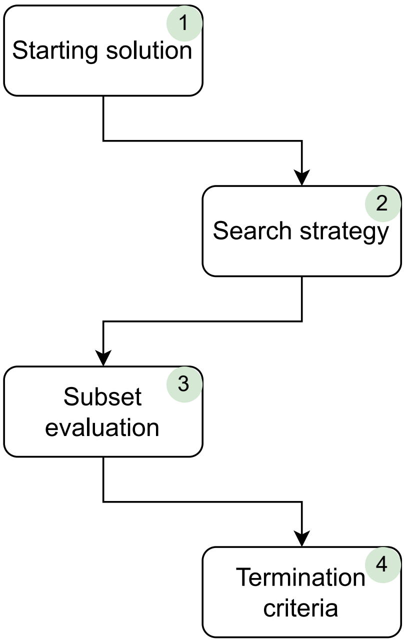


Figure 1: Components of the FS optimisation problem.

FS is an optimisation problem where, for a given dataset with m features, there are $2^m - 2$ possible subsets of features (excluding the empty and complete set) in the solution space, and the goal is to find one suitable subset for the ML task at hand. To begin with, a starting solution must be defined as shown in Figure 1. This is followed by choosing a suitable search strategy, subset evaluation approach, and termination criteria [21]. FS techniques can be categorised differently based on the search strategy and evaluation approach used. In this tutorial, we delve into this topic, examining the following three categorisations of FS methods: *selection strategy*, *model reliance*, and *relevance definition*.

2 TUTORIAL OUTLINE

The duration for the tutorial is 1.5 hours divided into three parts as detailed below.

- **Part 1 [30 minutes]**
  (1) Introduction to feature selection
  (2) Search perspective

Series ISSN: 2367-2005

10.48786/edbt.2024.85

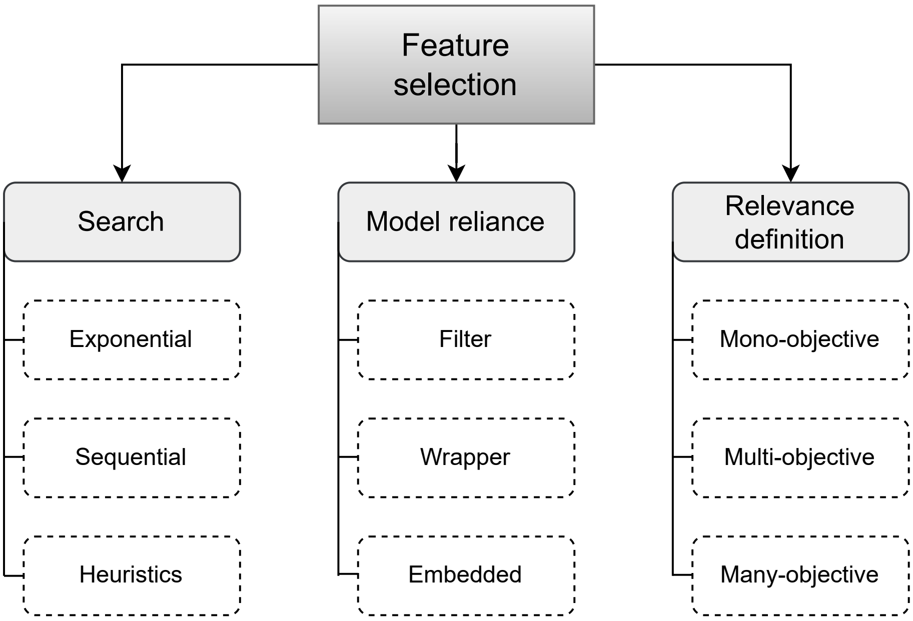


Figure 2: Categories of FS techniques.

  - Exponential
  - Sequential
  - Population-based
- **Part 2 [30 minutes]**
  (1) Evaluation perspective I (ML model reliance)
  - Filter
  - Wrapper
  - Embedded
- **Part 3 [30 minutes]**
  (1) Evaluation perspective II (Relevance definition)
  - Mono-objective
  - Multi-objective
  - Many-objective
  (2) Open challenges

2.1 Part 1

The tutorial begins with an introduction to FS, which is an optimisation problem with four components, as shown in Figure 1. The components can be instantiated differently, and in this tutorial, we present each one of them together with their possible configurations. The starting solution defines the origin of the search for a suitable subset of features. This could be an empty set, the full set of features, or a subset of features with any amount of features chosen randomly or through a predefined approach. After the starting solution is set, the search strategy indicates the course for navigating through the search space, which can be deterministic or not. For each potential solution visited during the search, the subset evaluation defines how it will be evaluated. This is the basis for comparing solutions in order to arrive at a final solution. Lastly, the termination criterion determines when to halt the exploration, which could be based on the elapsed search time, current solution quality, or other quality indicators.

Following this introduction, we look at the first classification of FS techniques, which is according to how the search space is explored to find a final solution. This can be *exponential*, *sequential*, or *population-based* [8]. With the *exponential* search strategy, all possible solutions in the search space are evaluated. This method guarantees that the overall best solution is found. However, it generates an exponential explosion, becoming unaffordable even for a moderate number of features; hence, it is not applicable in practice. The sequential strategy belongs to the deterministic and greedy algorithms family, which moves to the next best solution in each iteration of the search until the termination criterion is satisfied. It does not guarantee finding the optimum solution; however, it is more time-efficient than an exponential search. Lastly, the population-based search strategy follows metaheuristics that mimic natural evolution in finding a

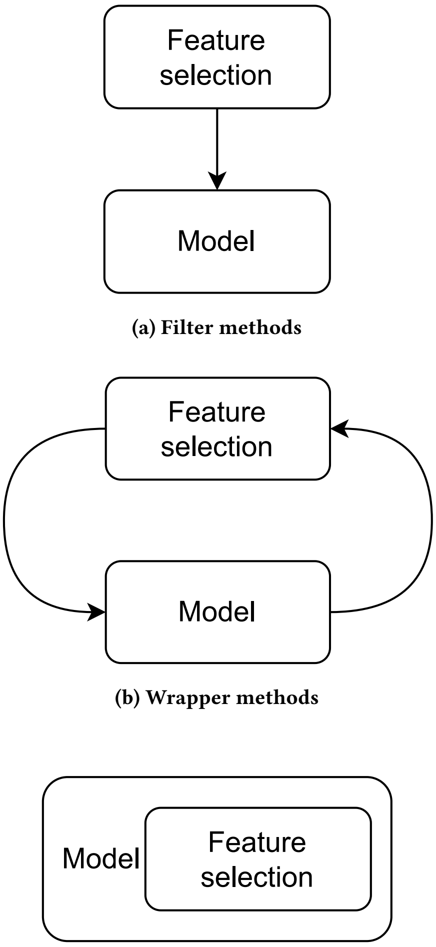
Table 1: Examples of FS methods and their categorisation.

FS Technique	Search strategy	Model reliance	Relevance definition
Branch and bound [16], Exhaustive [12]	Exponential	Filter/Wrapper	Mono-objective
Information Gain, Gini [14]	Sequential	Filter	Mono-objective
Forward selection, backward elimination [12]	Sequential	Wrapper	Mono-objective
Decision Tree [22], Ridge or Lasso regression [15]	Sequential	Embedded	Mono-objective
mRMR-PSO [4]	Population-based	Hybrid	Multi-objective
Genetic Algorithms [3]	Population-based	Filter/Wrapper	Mono-objective
NSGA-II [9]	Population-based	Filter/Wrapper	Multi-objective
NSGA-III [23]	Population-based	Filter/Wrapper	Many-objective

suitable subset of features. Although it does not guarantee finding the overall best solution, this approach is non-deterministic, and, by definition, it explores the solution space in a more diverse manner than the sequential one. Table 1 shows examples of the three classes of FS according to search strategy.

2.2 Part 2

In this part of the tutorial, we present the most common categorisation of feature selection techniques, that is, the one based on the subset evaluation, particularly on how the FS technique relies on an ML model to evaluate a possible solution. Based on this, the three classes of FS techniques are: *filter*, *wrapper* and *embedded* methods [5] shown in Figure 3. With filter methods, FS is completely disconnected from the ML model, as shown in Figure 3a. Indeed, these methods do not use the ML model to be trained to select relevant features. Instead, the relationships between the features are explored to evaluate a candidate solution. Their non-reliance on ML models makes them the fastest class of FS methods [18], and the solutions yielded are model agnostic and so can be applied to any ML model. The wrapper methods iteratively use ML models to measure subset relevance until the termination criterion is satisfied, as shown in Figure 3b. Due to their iterative nature, this FS technique class is the most computationally expensive. However, the solutions yielded are usually more performant [19] since they are tailored for the ML model of interest. Finally, embedded methods, shown in Figure 3c, are ML algorithms that perform FS intrinsically, such as those generating tree-based models and models with regularisation. Embedded methods yield good quality results of the chosen model and come at only the computational cost of building the model with no iteration required. More than one method of different classes can be combined to harness their advantages in what is called hybrid FS [10]. The hybrid approach combines the characteristics and functionalities of different classes of FS techniques to harness their strengths and mitigate their weaknesses. Examples of the classes of FS according to model reliance are shown in Table 1.



(a) Filter methods

(b) Wrapper methods

(c) Embedded methods

Figure 3: Categorisation of FS techniques based on model dependence.

2.3 Part 3

We present a third classification of FS methods in this section based on the number of objectives considered in defining the relevance of a feature subset. Each potential solution encountered in the solution space must be evaluated; this is how solutions are compared in order to find the final solution. To determine the performance of the candidate solution, various objectives could be considered, such as the subset size, model performance (e.g. accuracy, F1-score, AUC), fairness measures (e.g., statistical parity, equalised odds), redundancy measures, and other task custom objectives. When only one objective is used to evaluate the performance of a candidate solution, this is referred to as *mono-objective* FS; with two or three objectives, it is called *multi-objective* FS [1], and for four or more objectives we refer to it as *many-objective* FS [17]. More often, mono- or multi-objective FS is used as they are less complex to implement than the many-objective FS. The right choice depends on the ML task at hand. Some examples of mono-, multi-, and many-objective algorithms for FS are presented in Table 1.

To conclude, we discuss some open challenges in FS [6, 11], including:

(1) Efficiency of FS methods for
  (a) High-dimensional data
  (b) Large number of instances
(2) Scalability of FS methods considering
  (a) Parallelism
  (b) Distribution
(3) Adequacy of FS methods in dealing with unbalanced classification
(4) Real-time FS methods with streaming data
(5) Distributed FS methods
(6) Inclusion of feature cost in FS
(7) Visualization and Interpretability of FS results
  (a) Feature-model relationship
  (b) Multiple solutions
(8) FS methods for multi-label classification
(9) Transfer learning for FS
(10) Simultaneous optimisation for FS and instance selection
(11) Many-objective FS

The earlier listed challenges of scalability and dealing with unbalanced, streaming and distributed data are also present in everyday ML and thus have received more attention, although less for multi- and many-objective FS. Many-objective FS remains the least researched type of FS; however, it more accurately tackles the feature selection problem [17], which requires that multiple objectives be considered in the selection process.

3 GOALS AND OBJECTIVES

This tutorial aims to provide the audience with an in-depth introduction to feature selection. After the tutorial, the audience should be able to understand why FS is needed, the pros and cons of the various techniques and be able to choose an appropriate method for their ML tasks. Additionally, attendees will discover open problems and hot topics for research in feature selection.

4 INTENDED AUDIENCE

The intended audience for the tutorial is individuals who analyse data by creating ML models in academia and industry. The tutorial is also helpful for data scientists to improve their data analysis in terms of data understanding and model execution time, performance, and explainability.

5 BIOGRAPHY

Uchechukwu F. Njoku received the Erasmus Mundus Joint Masters degree in big data management and analytics from the Universitat Politècnica de Catalunya (UPC), Université Libre de Bruxelles (ULB), and Technische Universiteit Eindhoven (TU/e). She is currently pursuing the DEDS joint Ph.D. degree with UPC and ULB under the Horizon 2020 Marie Skłodowska-Curie Innovative Training Networks. Her research is focused on optimising feature selection for Big Data.

Alberto Abelló is a Full Professor in the Department of Services and Information Systems Engineering at UPC and an expert in Big Data and Business Intelligence. He carried out research stays at universities in Spain, Germany, France, Uruguay, and Scotland. He has participated in 24 national and international research projects or networks of excellence and has signed R&D agreements with companies such as Hewlett Packard or SAP. At the request of the UPC Center for Development Cooperation, he accepted the challenge of collaborating with the WHO Department of Neglected Tropical Diseases to create an information system for the monitoring of those diseases.

Besim Bilalli is an Assistant Professor in the Department of Services and Information Systems Engineering at the Universitat Politècnica de Catalunya (UPC). He obtained his doctorate as a joint degree between UPC and Poznan University of Technology (PUT) under the IT4BI-DC doctoral programme in 2018. He was awarded with a Juan de la Cierva research fellowship from the Spanish Ministry of Science and Innovation in 2021. His research interests include big data management and processing for data science.

Gianluca Bontempi (Senior Member, IEEE) is currently a Full Professor with the Computer Science Department at ULB and the Co-Head of the ULB Machine Learning Group. He is the author of more than 250 scientific publications. He is the co-author of several open-source software packages for bioinformatics, data mining, and prediction. His current research interests include big data mining, machine learning, bioinformatics, causal inference, predictive modelling, and their application to complex tasks in engineering (time series forecasting and fraud detection) and life science (network inference and gene signature extraction).

ACKNOWLEDGMENTS

The project leading to this publication has received funding from the European Commission under the European Union's Horizon 2020 research and innovation programme (grant agreement No 955895).

REFERENCES

[1] Qasem Al-Tashi, Said Jadid Abdulkadir, Helmi Md Rais, Seyedali Mirjalili, and Hitham Alhussian. 2020. Approaches to multi-objective feature selection: A systematic literature review. *IEEE Access* 8 (2020), 125076–125096.

[2] Zaheer Allam and Zaynah A Dhunny. 2019. On big data, artificial intelligence and smart cities. *Cities* 89 (2019), 80–91.

[3] Oluleye H Babatunde, Leisa Armstrong, Jinsong Leng, and Dean Diepeveen. 2014. A genetic algorithm-based feature selection. (2014).

[4] Sandhya Rani Bansal, Savita Wadhawan, and Rajeev Goel. 2022. mrmr-pso: A hybrid feature selection technique with a multiobjective approach for sign language recognition. *Arabian Journal for Science and Engineering* 47, 8 (2022), 10365–10380.

[5] Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. 2013. A review of feature selection methods on synthetic data. *Knowledge and information systems* 34 (2013), 483–519.

[6] Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. 2015. Recent advances and emerging challenges of feature selection in the context of big data. *Knowledge-based systems* 86 (2015), 33–45.

[7] Xu Chu, Ihab F Ilyas, Sanjay Krishnan, and Jiannan Wang. 2016. Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 international conference on management of data*. 2201–2206.

[8] Naoual El Aboudi and Laila Benhlima. 2016. Review on wrapper feature selection approaches. In *2016 International Conference on Engineering & MIS (ICEMIS)*. IEEE, 1–5.

[9] Tarek M Hamdani, Jin-Myung Won, Adel M Alimi, and Fakhri Karray. 2007. Multi-objective feature selection with NSGA II. In *Adaptive and Natural Computing Algorithms: 8th International Conference, ICANNGA 2007, Warsaw, Poland, April 11-14, 2007, Proceedings, Part I 8*. Springer, 240–247.

[10] Hui-Huang Hsu, Cheng-Wei Hsieh, and Ming-Da Lu. 2011. Hybrid feature selection by combining filters and wrappers. *Expert Systems with Applications* 38, 7 (2011), 8144–8150.

[11] Ruwang Jiao, Bach Hoai Nguyen, Bing Xue, and Mengjie Zhang. 2023. A survey on evolutionary multiobjective feature selection in classification: approaches, applications, and challenges. *IEEE Transactions on Evolutionary Computation* (2023).

[12] Alan Jović, Karla Brkić, and Nikola Bogunović. 2015. A review of feature selection methods with applications. In *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*. Ieee, 1200–1205.

[13] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. 2014. A survey of feature selection and feature extraction techniques in machine learning. In *2014 science and information conference*. IEEE, 372–378.

[14] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. 2017. Feature selection: A data perspective. *ACM computing surveys (CSUR)* 50, 6 (2017), 1–45.

[15] Ramakrishnan Muthukrishnan and R Rohini. 2016. LASSO: A feature selection technique in predictive modeling for machine learning. In *2016 IEEE international conference on advances in computer applications (ICACA)*. IEEE, 18–20.

[16] Narendra and Fukunaga. 1977. A branch and bound algorithm for feature subset selection. *IEEE Transactions on computers* 100, 9 (1977), 917–922.

[17] Uchechukwu F Njoku, Alberto Abelló, Besim Bilalli, and Gianluca Bontempi. 2023. A data-science pipeline to enable the Interpretability of Many-Objective Feature Selection. *arXiv preprint arXiv:2311.18746* (2023).

[18] Uchechukwu Fortune Njoku, Alberto Abelló Gamazo, Besim Bilalli, and Gianluca Bontempi. 2022. Impact of filter feature selection on classification: an empirical study. In *Proceedings of the 24rd International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP): co-located with the 24th International Conference on Extending Database Technology and the 24th International Conference on Database Theory (EDBT/ICDT 2022): Regne Unit, March 29, 2022*. CEUR-WS. org, 71–80.

[19] Uchechukwu Fortune Njoku, Alberto Abelló Gamazo, Besim Bilalli, and Gianluca Bontempi. 2023. Wrapper methods for multi-objective feature selection. In *26th International Conference on Extending Database Technology (EDBT 2023): Ioannina, Greece, March 28-March 31: proceedings*. OpenProceedings, 697–709.

[20] Steven Euijong Whang and Jae-Gil Lee. 2020. Data collection and quality challenges for deep learning. *Proceedings of the VLDB Endowment* 13, 12 (2020), 3429–3432.

[21] Wen Xiao, Ping Ji, and Juan Hu. 2021. RnkHEU: A Hybrid Feature Selection Method for Predicting Students' Performance. *Scientific Programming* 2021 (2021), 1–16.

[22] HongFang Zhou, JiaWei Zhang, YueQing Zhou, XiaoJie Guo, and YiMing Ma. 2021. A feature selection algorithm of decision tree based on feature weight. *Expert Systems with Applications* 164 (2021), 113842.

[23] Yingying Zhu, Junwei Liang, Jianyong Chen, and Zhong Ming. 2017. An improved NSGA-III algorithm for feature selection used in intrusion detection. *Knowledge-Based Systems* 116 (2017), 74–85.