Analyzing the Performance of Feature Selection on Regression Problems: a Case Study on Older Adults' Functional Profile

Javier Rojo, Lara Guedes de Pinho, César Fonseca, Manuel José Lopes, Sumi Helal, Juan Hernández, Jose Garcia-Alonso, Juan M. Murillo

Abstract—Healthcare systems are capable of collecting a significant number of patient health-related parameters. Analyzing them to find the reasons that cause a given disease is challenging. Feature Selection techniques have been used to address this issue—reducing these parameters to a smaller set with the most "determinant" information. However, existing proposals usually focus on classification problems—aimed to detect whether a person is or is not suffering from an illness or from a finite set of illnesses. However, there are many situations in which health professionals need a numerical assessment to quantify the severity of an illness, thus dealing with a regression problem instead. Proposals using Feature Selection here are very limited. This paper examines several Feature Selection techniques to gauge their applicability to the regression-type problems, comparing these techniques by applying them to a real-life scenario on the functional profiles of older adults. Data from 829 functional profiles assessments in 49 residential homes were used in this study. The number of features was reduced from 31 to 25—with a correlation between inputs and outputs of 0.99 according to the R^2 score and a Mean Square Error (MSE) of 0.11—or to 14 features—with a correlation of 0.98 and MSE of 5.73.

Index Terms—Feature Selection, Regression, Machine Learning, Aging Informatics, Healthcare Data Analytics, eHealth

1 Introduction

Every day, healthcare professionals need to assess large numbers of patients—not only those with chronic or degenerative diseases but also those with long-term or chronic conditions [1]. This requires a constant reassessment of patients, especially for the aging population, among whom chronic and degenerative diseases and dysfunctions are more common and more frequent. Many of these assessments involve collecting large amounts of information regarding different variables associated with the patients' health status [2]. To help health professionals manage data from these assessments, many healthcare systems have IT platforms where their patients' information is digitized [3] into Electronic Health Records, allowing health professionals, with the help of software developers and data scientists, to develop powerful assessment platforms [4].

Nevertheless, the effort needed for health professionals and informal caregivers to perform these assessments remains excessive. Collecting data from several variables in each assessment is time-consuming which infringes on their limited time and reduces their patient-facing care time [5]. Health professionals are thus forced to reduce the number of assessments that they need to perform routinely to care for their patients, thus reducing the sustainability and quality of care.

If a clinical assessment includes redundant or irrelevant variables, then the initial set of inputs could potentially be reduced at a minimal accuracy loss, consequently easing the aforementioned time crunch situation. An example of this is the World Health Organization Disability Assessment Schedule II (WHODAS-II), an assessment instrument for functioning. Although the original version included 36 items, some studies have validated the use of a reduced 12-item version [6], [7]. However, establishing which variables contain redundant information and can therefore be eliminated is not a trivial matter. The selection cannot be done manually, based on intuition or using arbitrary techniques to find relationships between variables. Instead, statistical techniques are needed to calculate which reduced set of variables provides the same information as the original set [8]. After this, calculations done to extract results with these items must be reformulated or replaced by other techniques—since the new set does not contain all items of the original one.

To solve this problem, two types of techniques exist in the literature, which can be used together. First, Feature Selection techniques [8] allow selecting from among a set of features those that best describe the output, eliminating redundant, unrelated, or non-influential variables. Second, Machine Learning and Deep Learning techniques can be used to define models that automatically learn from data to establish relationships between input variables and an output that has not been programmed or defined beforehand—a prediction model. These techniques are already employed in the health domain to solve problems involving the analysis of different kinds of data [9], [10], [11].

- *J. Rojo, J. Hernández, J. Garcia-Alonso, and J. M. Murillo are with the Department of Computer and Telematics Systems, University of Extremadura, Spain. E-mail: {javirojo, juanher, jgaralo, juanmamu}@unex.es*
- *S. Helal is with the University of Florida, USA. Email: helal@cise.ufl.edu*
- *L. Guedes de Pinho, C. Fonseca, and M. J. Lopes are with the University of Évora and the Comprehensive Health Research Centre (CHRC), Portugal. Email: {lmgp, cfonseca, mjl}@uevora.pt*

Manuscript received December XX, 2021; revised December XX, 202X.

Feature Selection and Machine Learning/Deep Learning techniques are closely related. Feature Selection techniques are employed to pre-process the input variables in any of the main types of Machine Learning and Deep Learning problems [10]: classification —the output is one or several among a series of well-known classes—, regression —the output is a continuous value—, and clustering —the output is a series of clusters not known previously, to which the different samples belong. This pre-processing is often conducted as part of the data cleansing phase [9], [12]. Feature Selection is frequently used in classification problems [8], [13], [14]. The use of Feature Selection in unsupervised problems, such as clustering, has also been extensively researched for years [15]. However, as far as the authors are aware, Feature Selection is not commonly applied to regression problems [16], [17] and its potential could be investigated further [18].

This paper aims to offer a technological solution that can be used to reduce the number of items to be measured in a health assessment, allowing the diagnosis of a pathology or condition to be accomplished with almost the same precision but with less data. To do this, different feature selection techniques used in regression problems and available in the literature have been reviewed. In order to compare them, these techniques are applied to a real-world case study focused on the assessment of aging adults' functional profiles. This comparison allows us to propose new models to predict functional profiles' scores with fewer variables than the original formulas. The results obtained by the model in terms of correlation between inputs and outputs are also analyzed. They provide an overview of the performance of feature selection techniques when applied to regression problems in the case study. They also provide insights into their extrapolation to other problems of this type—in other healthcare assessments as well as other application domains.

The paper is organized as follows. First, Section 2 introduces Dimensionality Reduction and its Feature Selection techniques. Section 3 provides an overview of the structure and process followed in the development of the proposal and introduces the case study. Section 4 compares each of the alternative feature selection techniques, by using them—independently or in combination—to predict target variables based on the older adults' functional profile dataset. Section 5 discusses the results in terms of number of features and achieved correlation between inputs and outputs, and compares the initial set of variables with the final set. Section 6 is a brief discussion about Feature Selection on regression problems and the results obtained in the case study focused on in this paper. Section 7 reviews existing literature on the use of Feature Selection on healthcare data. Finally, Section 8 presents the paper's conclusions and an agenda for future work in this research area.

2 Background

The increasingly high dimensionality and complexity of data make it necessary to use Dimensionality Reduction (DR) techniques [19] in exploratory data analysis. Feature Selection is a type of DR technique that allows data scientists to move from an initial high dimensional data set to a more reduced one, known as the intrinsic dimension of data. This intrinsic dimension contains the minimum features that describe the data [20], providing a series of benefits: simplification of prediction models based on these data, improvement of its predictions, or avoidance of the long-standing problem known as the "curse of dimensionality" [21], among others. These benefits are common to all types of DR techniques. However, each one has its own intrinsic advantages, depending on how dimension is reduced in each of them.

Feature Selection methods [22] reduce the data dimensionality by selecting a subset of variables from the input. This subset describes the input data while reducing the adverse effects due to including noise and irrelevant variables in predictions [8]. Therefore, they aim to simplify prediction models, improving their results—namely better generalization and reduced overfitting—and problem understanding. They also reduce the amount of information to be collected as input, and thus the computational load and fit time, while avoiding the above-mentioned curse of dimensionality. For these reasons, Feature Selection methods are particularly useful in domains with many input features and few samples [23]—i.e., analysis of DNA micro-arrays data or handwritten manuscripts—especially in classification problems.

These techniques are based on the premise that, in many models, the input information that allows characterizing an output—for example, discriminating between different classes in a classification problem—can be obtained from a smaller set of variables than initially used. This could be due to mutually-dependent variables—where the value of one is highly correlated with the value of another—or due to irrelevant input variables, whose values do not affect the output—even though they were initially believed that they would. The former introduces redundant information (and complexity) in the models, while the latter could lead to noise and bias in the prediction. Feature Selection eliminates redundant and irrelevant features at a minor loss of information, and hence model accuracy, in both supervised and unsupervised problems.

Feature Selection is different from other DR methods [24] such as Feature Extraction. Feature Extraction methods generate a set of new features based on the initial ones with each new feature containing the most relevant information from several of the initial ones. Such Feature Extraction set might or might not be smaller than the initial set. Feature Selection, on the other hand, filters the most relevant, existing features without generating new ones, only eliminating some. For this reason, Feature Selection is better suited to the healthcare sustainability problem addressed in this work. However, when predictions need to be improved, both methods can be applied together—first Feature Selection and then Feature Extraction on the resulting features.

There are three main types of Feature Selection methods in the existing literature [8], [22]: filter, wrapper, and embedded methods. Filter methods are independent of the prediction model that is subsequently used, where the selection of the most relevant variables is based on a ranking criterion. The ranking is arrived at using a well-defined criterion used to establish the relationships between each input variable and the output variables. Wrapper methods [25] work differently by employing and evaluating a prediction model to each subset of the variables. Performance for each subset is measured as a function of the prediction accuracy provided by the model trained with that subset. This model is then "wrapped" in a search algorithm for the best-performing subset. Finally, embedded methods also employ a prediction model to determine the best subset of input variables. However, in this case, feature selection is performed as part of the model's training process, rather than based on the predictions yielded by an already trained model. Therefore, there is no need to split the data into a training and a testing set—to train first and then evaluate the prediction model's performance later—thus reducing the Wrapper methods' computational cost.

These methods are not mutually exclusive and can be used together [26]. For instance, the filter methods, being less computationally expensive and more independent regarding the prediction model to be subsequently used but also less aggressive in terms of feature sifting, are usually applied first to perform an initial phase of feature selection, which amounts to pre-processing the set of variables that will subsequently be analyzed using one of the other two techniques.

In addition to Feature Selection techniques, this paper also employs Machine Learning models. These techniques can be divided into supervised, unsupervised, and reinforced learning, depending on the type of problem to be addressed. Learning is considered supervised when there is a set of labeled data —both the input and the output are known. Supervised learning includes problems of classification or regression, depending on what is to be learned. This paper will focus on supervised learning and, more specifically, on regression problems.

3 Proposed approach

The proposal in this paper is the result of an experimental study carried out in several phases. All the experiments were conducted using real-life data from a single case study: an evaluation of the aging adults' functional profile. The specific data used were derived from 829 assessments of 716 older adults treated in 49 residential homes and medical centers in Portugal. These assessments are employed to compute five different continuous values that describe the aging adults' functional profile, making this case study a regression study.

The first phase involved examining available examples of using Feature Selection techniques in regression problems. Quantitative comparison was carried out considering different aspects to evaluate the performance of each technique when applied to our case study.

Then, the best performing techniques were selected to create a series of models that, using a reduced set of input features, could predict the functional profile of the older adults studied with almost the same precision as the original set. In order to measure this accuracy, we provide a series of metrics on the correlation between inputs and outputs and the error introduced in the prediction models, compared to the traditional method using all the input features.

3.1 Case Study: Evaluation of Functional Profile using the Elderly Nursing Core Set (ENCS)

The experiment performed is associated with the authors' Multidimensional Integrated Assessment Platform for elderly (MIAPe) [4], which is employed by health professionals and caregivers in different Portuguese socio-geriatric settings. It is used to assess different aspects of the patients' health including the patients' functional profile which is the most relevant to the work in this paper. For this purpose, 31 items or questions—the input variables—are collected, with values filled manually using the Elderly Nursing Core Set (ENCS) form [27].

The ENCS was developed by Lopes and Fonseca in 2013 to assess the quality of life in terms of functioning among older adults [28]. The psychometric properties of the ENCS were later evaluated by Fonseca et al. [27]. This assessment instrument is based on the International Classification of Functioning, Disability and Health (ICF) [29]. Each of the 31 questions is valuated on a Likert scale from 1 to 5 points: (1) No disability: 0–4%; (2) Mild disability: 5– 24%; (3) Moderate disability: 25–49%; (4) Severe disability: 50–95%; and (5) Complete disability: 96–100%. From these 31 items, 10 items measure body functions, 17 items measure body structure, and 4 items measure environmental factors. Full description of these items can be found in the supplementary material. The ENCS is divided into four areas of concern: self-care, learning and mental functions, communication, and social relationships. A higher value is correlated with a worse functional profile of the assessed individual. Applying a series of validated mathematical formulas [27] to the values of these 31 questions, five scores corresponding to the above-mentioned areas of concern are extracted. These represent the general functional profile of the elderly: *General punctuation of functionality* (GPF), *Self-care* (AC), *Learning and memory functions* (LMF), *Communication* (C), and *Relationships with friends and carers* (RFC). These scores are represented as continuous values, from 0 to 100.

- **General punctuation of functionality** corresponds to the overall score obtained in the functional profile for all dimensions. For its calculation, all items are employed.
- **Self-care** corresponds to the basic activities of daily living, such as eating and drinking. For its calculation, all 17 items from body structure are employed.
- **Learning and memory functions** is concerned with cognitive functions, such as memory or orientation. For its calculation, all 10 items from body functions are employed.
- **Communication** is related to the ability to talk or hold a conversation. For its calculation, 4 of the 17 items from body structure are employed.
- **Relationships with friends and carers** corresponds to the ability to maintain relationships with family or obtain support from health professionals, friends, or other caregivers. For its calculation, all 4 items from environmental factors are employed.

Each of these scores can be represented as a regression function that takes as input each of the 31 questions of the ENCS form, making this case study a suitable one for the intended scope. Therefore, in this case study the focus will be put on being able to compute each of these five scores with a reduced number of questions—instead of the 31 original ones.

4 Comparison of Feature Selection Techniques on Functional Profile

To determine the best Feature Selection technique to be applied to the case study, the different techniques available for Feature Selection on regression problems were compared. As far as the authors know, no previous study has compared the application of different Feature Selection techniques on regression problems.

To carry out this comparison, a battery of different Feature Selection techniques has been chosen, following a well-structured process— to make the comparison as fair and reproducible as possible. All the results obtained were meticulously documented to ensure replicability. The techniques selected, the methodology followed, and the results obtained are discussed next.

4.1 Selected Feature Selection Techniques

The Feature Selection techniques used in this study are commonly available in Data Science tools. Specifically, we used the Python programming language, which is widely used in the field of Data Science and Data Mining. Therefore, only techniques offered for this language were sought. Techniques were selected from Scikit-learn[1] and Yellowbrick[2] Data Science libraries. The first is a widely used library for the creation of Machine Learning models that also offers DR tools. The second is a Scikit-learn wrapper that offers visual solutions to facilitate model and data analysis.

Limiting the study to only techniques available in Python and, more specifically, offered by these libraries, may suggest that certain techniques were left out of the comparison—for instance, those that are less common. However, given the large number of techniques offered by these libraries—as shown below—, we consider that the obtained results in the comparison are sufficiently valid—since we have included very common and frequently used Feature Selection techniques such as Recursive Feature Elimination (RFE) and filter methods based on the correlation between features and Mutual Information (MI). Moreover, from a practical viewpoint, using techniques offered by libraries that are widely used in production environments facilitates the availability, reproducibility, and transfer of results of this research.

All Feature Selection techniques available in both libraries were evaluated. For each technique, different parameter combinations were tested. Compared Feature Selection techniques are shown in Table 1.

Type	Technique	Library
Filter	GenericUnivariateSelect	Sklearn
Filter	SelectPercentile	Sklearn
Filter	SelectKBest	Sklearn
Filter	SelectFpr	Sklearn
Filter	SelectFdr	Sklearn
Filter	SelectFwe	Sklearn
Filter	VarianceThreshold	Sklearn
Filter	Rank1D	Yellowbrick
Filter	Rank2D	Yellowbrick
Filter	Feature Correlation	
Embedded	SelectFromModel	Sklearn
Embedded	SequentialFeatureSelector	Sklearn
Embedded	RFE	Sklearn
Embedded	RFECV	Sklearn
Embedded	Feature Importances	Yellowbrick
Embedded	RFECV	Yellowbrick

TABLE 1
List of techniques evaluated

No wrapper method was tested since these libraries offer embedded methods instead.

4.2 Methodology

Our comparison evaluated the different techniques based on the number of features selected, the correlation achieved by Machine Learning models using those features, and the execution time required to select the features. Correlation is measured with the coefficient of determination, R^2 score, with 1.0 being the ideal value—indicating that model that perfectly explains the observed output variation concerning input variation—and 0 being the worst-case scenario—model that does not explain any variation—. Negative values are not taken into account. In contrast to other metrics, the closer the R^2 score is to 1, the better the model explains the variability of the target feature (output). Other metrics, such as Mean Square Error (MSE), must be considered when training a final model. However, to assess how changes in input features affect the model's output, only the R^2 score was considered in this comparison.

To execute the different techniques we used a laptop running Windows 10, with 16 GB of RAM and an Intel Core i7-8550U processor at 1.8 GHz base frequency and 4.0 GHz turbo frequency, equipped with NVMe SSD technology for storage. The following software was used: Python v.3.7.5 (programming language), Anaconda v.1.9.12 (Python distribution for Data Science), conda v.4.10.1 (package management system), Jupyter v.6.0.1 (development environment), Scikit-Learn v.0.23.0 (Data Science library), Yellowbrick v.1.2 (Data Science library), and Pandas v.0.25.3 (data manipulation and analysis library).

The process followed to carry out the comparison was as follows:

Step 1: Transform functional profile data into a suitable format. The Elderly Nursing Core Set (ENCS) data was anonymized and filtered to obtain a dataset containing only information on the form's 31 items and the values for each of the five scores of the functional profile of the older adults (GPF, AC, LMF, C, and RFC).

Step 2: Test the initial feature set data, without applying Feature Selection techniques, on 12 different Scikit-learn regression models. These 12 models were generated with the main Scikit-learn algorithms for regression models —based on the assumption that linear regression, logistic regression, support vector machines, and tree-based algorithms are the most employed for regression problems [30]. All parameters were left as default. Since there are five outputs, five models had to be created with the same inputs—one for each output. In this case, we took advantage of Scikit-learn's MultiOutputRegressor class, which internally creates one

1. https://scikit-learn.org/stable/
2. https://www.scikit-yb.org/en/latest/

model for each output and then encapsulates them in one. The results were obtained with an initial set of 829 samples, corresponding to evaluations of actual patients, dividing them into 555 for training and 274 for testing. This is a typical division of roughly two parts for training and one for testing. Then a hold-out validation was performed—training and testing sets without cross-validation.

Step 3: Select the five models with the best predictions (best R^2) out of the 12 models tested in the previous step. This is possible because the initial feature set is easily computable and allows the testing of different algorithms.

Step 4: Test the Feature Selection algorithms. For each of these methods, different configurations were tested, based on the different hyperparameters that they accept:

- In those that required keeping a number of features, such as K, percentile or the maximum number of features to be selected, the value was initially set to half —15 features or a percentile of 50%. This decision has been taken because the objective in the step is to make an initial comparison between the different techniques and not to find the optimal value of hyperparameters for each of them—or even compare the results for the same one with different hyperparameter values. This is done later in Step 6.
- In those using a threshold, different values were tested.
- Values such as *alpha*, employed by techniques such as SelectFpr or SelectFdr, were left as default.
- For the embedded methods, each of the five models selected in Step 3 was tested as estimators. This estimator model is the one being wrapped by the embedded method to determine the importance of each feature and can be different from the final prediction model where the set of features selected is employed as input.
- Methods using a model as estimator required one estimator for each output and thus were not compatible with MultiOutputRegressor. This required performing a feature selection for each of the five outputs, before merging the results. This merge is done by joining the selected features for each output so that the resulting set of features allows predicting the five functional profile scores —the five outputs.

Step 5: Evaluate the performance of the selected feature set for each technique and hyperparameter configuration, again using the prediction models selected in Step 3. The results are provided in terms of the final number of features, R^2 score for each of the five models evaluated, average R^2 score, and execution time of the Feature Selection process. To calculate the R^2 score for each model, we used the same division of samples and validation as in Step 3.

Step 6: Those methods that could be configured using parameters were executed again, using the whole range of possible values. Results were complemented with a graph per each technique and configuration, showing the evolution of the number of resulting features for each different value of the parameters and the average R^2 score associated with each number of variables. This score was calculated using the same functions as in Step 5.

All the software developed for the benchmark is provided as additional material and in a Zenodo repository[3]. The complete comparison can be replicated by running the "Comparison" and "Graphical" notebooks.

4.3 Comparison of Results

The main outcomes of the different steps of the comparison are shown below. A more complete version of the tables presented here, and additional figures with the results of varying the parameters' values when using the different techniques, can be found as additional resources and in the aforementioned Zenodo repository.

From **Step 1**, no outcomes are obtained. This step is only for data processing.

From **Step 2** the Table 2 is obtained. This table shows the results obtained when evaluating the different models, using all 31 features—without Feature Selection. As can be seen, the results in most of the tested models are close to 100% correlation (1.0 R^2 score). Only Support Vector Regressor (SVR) with a non-linear RBF kernel yielded lower correlation results (0.74 R^2 score). On the other hand, the third-degree Polynomial Regression yielded a negative and out-of-bounds result. According to the Scikit-learn library that provides the implementation of the algorithm and the calculation of the coefficient of determination, the negative result is due to the model output being very different from that expected for the same input values [31].

To ensure that the R^2 score obtained by these models is not the result of overfitting and that default hyperparameters are not leading the models to a suboptimal point, a 5-step cross-validation with the training data has been carried out, showing that the results are good and acceptable. The standard deviation between the R^2 score of each step has been analyzed and, except for the SVR with RBF kernel and NuSVR, the values obtained are in the order of 10^{-3} to 10^{-5}. The exact results can be found in the Jupyter notebook *Initial_tests_CV* in the supplementary material.

The five models with a coefficient of determination higher than 0.99—Linear Regression, Ridge Regression, Lasso Regression, Extra Tree Regression, and LinearSVR—were selected in **Step 3**.

As well, Table 2 shows the time needed to train the model and to perform a prediction for each of the tested models. Neural Network Regression is the one offering the worst training time, followed by Random Forest, Extra Tree, and Polynomial Regression of Degree 3. This situation is expected since these are the more complex models. Prediction times are in the order of microseconds in most cases, with the costly ones being the Random Forest and Extra Tree methods again, as well as some implementations of the SVR. These are from the order of tenths of milliseconds. However, times for both cases, train and prediction, are acceptable in any of the models and their consideration is not as important as R^2 for that case—no need for constant retraining or fast prediction exists in our case study.

Step 4 and 5 start at that point, having the results of the five models selected when they are trained with the original features set. In these steps, the aim is to obtain the results for the same models when they are trained with the

3. https://doi.org/10.5281/zenodo.6421873

Model	R^2	Train time (s)	Predict time (s)
Linear Regression	0.999659	0.0102	$7.40x10^{-6}$
Polynomial Regression (Degree=2)	0.885259	0.3103	$3.13x10^{-5}$
Polynomial Regression (Degree=3)	-220.837775	1.0423	$9.73x10^{-5}$
Ridge Regression	0.999657	0.0182	$8.63x10^{-6}$
Lasso Regression	0.990675	0.0124	$1.09x10^{-5}$
Decision Tree Regression	0.957315	0.0201	$1.09x10^{-5}$
Random Forest Regression	0.983631	1.3971	$2.06x10^{-4}$
Extra Tree Regression	0.990431	1.1709	$1.85x10^{-4}$
SVR (kernel=rbf)	0.745581	0.1036	$3.01x10^{-4}$
LinearSVR	0.998442	0.2315	$7.29x10^{-6}$
NuSVR	0.721767	0.0623	$1.53x10^{-4}$
Neural Network Regression	0.978924	2.6033	$3.72x10^{-5}$

TABLE 2
Results of model testing without feature selection

set of features selected by each Feature Selection technique (and each of its parameters' configurations) tested in this study. Tables 3 and 4 show the summary of the results obtained for each of these techniques in different terms: the number of features selected, average train time (in seconds), and R^2 score when the set of features selected is used as the input set to train each of the five models under study. Table 3 shows the results of the filter methods and their configurations while Table 4 shows results for the embedded ones. As can be seen in both tables, the results obtained are related to the number of features considered, execution time (in seconds), and correlation—the R^2 score obtained in Step 5 of comparison with the set of features selected on each of the five models selected of the Table 2 during Step 3 and on average. As was the case in Step 2, a 5-step cross-validation has been carried out here in addition to the traditional validation as well, to check that the models used and their configuration are still valid and do not suffer from overfitting. The results of this cross validation can be found in the Jupyter notebook *Comparison_CV* of the supplementary material. All possible configuration and techniques available in the libraries employed have been tested. However, those configurations or techniques not working have been discarded and are not shown in the tables of results to improve readability. That is the reason why some techniques listed in Table 1 are not shown on them. Their results can be found in the supplementary material attached to this manuscript.

In the tables, tuples containing only results in terms of execution time belong to techniques that provide visual results, that is, that do not automatically filter features. Instead of that, they provide some graphs that experts in the domain must interpret to manually filter the features by themselves. Since such results are subject to expert interpretation, they do not provide direct information in terms of the number of features selected and/or results of the models trained with it. Hence, all these tuples are listed in the tables to indicate that they have been tested. However, none of them will be delved into further in this comparison.

Regarding the results, we will consider first the **time required to perform the feature filtering** —which offers an insight into how computationally costly each of the techniques employed can be. This time is only relevant for the beginning, when the features are selected. After that, it will not be necessary to re-execute these techniques. In this regard, it can be observed that the average time of the filter methods (*Time* column of *Average results* row in Table 3) is much lower than the average time of the embedded methods (*Time* column of *Average results* row in Table 4). However, this is mainly due to the time required for the RFECV technique of the *yellowbrick* library when using Linear SVR as estimator. If the time required by that technique with that configuration is not taken into account, the average time of the embedded methods is significantly reduced to 4.37 seconds. Therefore, it can be concluded that filtering methods are, on average, slightly faster than embedded methods. Specifically, embedded methods' time is 663% that of filter ones if the time of the RFECV with Linear SVR as estimator is discarded. However, and especially taking into account the duration of the execution times obtained by using any of the methods considered—mostly in the order of tenths and hundredths of seconds, the results are manageable and do not affect the decision of which one should be used in this case study.

If we focus only on the times obtained by the embedded methods, the situation arising from using a Linear SVR as estimator in the RFECV technique appears not to be an isolated case —although this is the most outstanding example. In all embedded methods, applying a Linear SVR as estimator notably increases the execution time by 1953% —from 4.61 seconds on average for other techniques and configurations to 90.06 seconds on average for those using a Linear SVR as estimator. Conversely, the model that offers the best estimator times in the embedded methods is the Ridge Regression, although with results very similar to those offered by the other three models.

It is noticeable, however, that the embedded techniques from the Yellowbrick library —Feature Importances technique and RFECV (yellowbrick version)— have much longer execution times than the techniques taken from Scikit-learn. This is probably because this library works as a wrapper over Scikit-learn—taking the same amount of time as if they were executed directly—and then generating graphical solutions based on the results obtained. The time requirements of these techniques are even more remarkable in the case of RFECV, whose fastest configuration — using a Linear Regression as estimator— needs 74 times the time of the slowest Scikit-learn technique with the same configuration—RFE with a Linear Regression as estimator.

On the other hand, if we focus only on the times obtained by the filter methods, we can appreciate that the use of the *mutual_info_regression* as score function leads to a notable increase in the times. Specifically, the use of this score function instead of *f_regression* leads to an increase in time of 11458% in the techniques that allow both score functions— 0.02 seconds on average with *f_regression* and 2.34 seconds on average with *mutual_info_regression*. In addition, as with the embedded methods, the Yellowbrick techniques also require longer times than the others. These are the only methods whose results reach times in the order of seconds or tenths of seconds—without taking into account techniques employing *mutual_info_regression* as score function.

Next, we will discuss the **number of features selected** by each technique. It is important to mention at this point that in most cases the number of features selected by each technique does not match the number of maximum, k best—*k* being the number of best features to be selected— and percentile features indicated for the technique. This is due to the existence of five outputs in our case study.

					R^2 score					
Feature Selection technique	Score func. / Algorithm	Parameters	N. features	Time (s)	LR	Ridge	Lasso	Extra Tree	LinearSVR	Avg.
GenericUnivariateSelect	f_regression	percentile(50)	**26**	**0.030751**	0.999662	0.999661	0.991935	0.959194	0.999570	**0.990004**
GenericUnivariateSelect	mutual_info_regression	percentile(50)	**26**	**2.134323**	0.999662	0.999661	0.991935	0.962304	0.999617	**0.990636**
GenericUnivariateSelect	f_regression	k_best(15)	**26**	**0.018815**	0.999662	0.999661	0.991935	0.957074	0.999595	**0.989585**
GenericUnivariateSelect	mutual_info_regression	k_best(15)	**26**	**1.907155**	0.999662	0.999661	0.991935	0.956841	0.999618	**0.989543**
GenericUnivariateSelect	f_regression	fpr	**29**	**0.020798**	0.999660	0.999659	0.990675	0.959773	0.996560	**0.989265**
GenericUnivariateSelect	f_regression	fdr	**29**	**0.017323**	0.999660	0.999659	0.990675	0.957852	0.997536	**0.989076**
GenericUnivariateSelect	f_regression	fwe	**29**	**0.028273**	0.999660	0.999659	0.990675	0.957720	0.998158	**0.989174**
SelectPercentile	f_regression	percentile(50)	**26**	**0.016828**	0.999662	0.999661	0.991935	0.960177	0.999587	**0.990204**
SelectPercentile	mutual_info_regression	percentile(50)	**26**	**2.680883**	0.999662	0.999661	0.991935	0.960406	0.999616	**0.990256**
SelectKBest	f_regression	k(15)	**26**	**0.021328**	0.999662	0.999661	0.991935	0.962586	0.999604	**0.990690**
SelectKBest	mutual_info_regression	k(15)	**26**	**1.932915**	0.999662	0.999661	0.991935	0.964257	0.999524	**0.991008**
SelectFpr	f_regression	alpha(0.05)	**30**	**0.017359**	0.999659	0.999657	0.990675	0.956662	0.999236	**0.989178**
SelectFdr	f_regression	alpha(0.05)	**30**	**0.016368**	0.999659	0.999657	0.990675	0.963822	0.998723	**0.990507**
SelectFwe	f_regression	alpha(0.05)	**30**	**0.016368**	0.999659	0.999657	0.990675	0.956609	0.998340	**0.988988**
VarianceThreshold		threshold(0)	**31**	**0.011376**	0.999659	0.999657	0.990675	0.951833	0.999485	**0.988262**
VarianceThreshold		threshold(0.5)	**30**	**0.012895**	0.990122	0.990139	0.983033	0.948834	0.987646	**0.979955**
VarianceThreshold		threshold(0.8)	**26**	**0.014385**	0.950137	0.950143	0.940121	0.835992	0.933441	**0.921967**
Rank1D	algorithm="shapiro"			**0.266352**						
Rank2D	algorithm="pearson"			**0.514352**						
Rank2D	algorithm="covariance"			**0.467696**						
Rank2D	algorithm="spearman"			**0.499440**						
Rank2D	algorithm="kendalltau"			**0.775745**						
Feature Correlation	method="pearson"			**1.325842**						
Feature Correlation	method="mutual_info-regression"			**3.091538**						
Average results of Scikit Learn			**28**	**0.52342**	**0.996186**	**0.996187**	**0.987845**	**0.951290**	**0.994462**	**0.985194**
Average results of Yellowbrick				**0.99157**						
Average results			**28**	**0.659963**	**0.996186**	**0.996187**	**0.987845**	**0.951290**	**0.994462**	**0.985194**

TABLE 3
Results with filter feature selection methods

					R^2 score					
Feature Selection technique	Estimator	Parameters	N. features	Time (s)	LR	Ridge	Lasso	Extra Tree	LinearSVR	Avg.
SelectFromModel	Linear Regression		**25**	**0.019840**	0.999661	0.999660	0.991935	0.956487	0.999528	**0.989454**
SelectFromModel	Ridge Regression		**25**	**0.017359**	0.999661	0.999660	0.991935	0.969842	0.999629	**0.992145**
SelectFromModel	Lasso Regression		**26**	**0.018350**	0.999662	0.999661	0.990675	0.963172	0.998009	**0.990236**
SelectFromModel	Extra Tree Regression		**15**	**0.029721**	0.989018	0.989018	0.981734	0.949199	0.987603	**0.979314**
SelectFromModel	Linear SVR		**25**	**0.472687**	0.999661	0.999660	0.991935	0.965365	0.999622	**0.991249**
RFE	Linear Regression	step=1. n_features_to_select(0.5)	**19**	**0.184014**	0.995868	0.995877	0.988012	0.956765	0.995921	**0.986488**
RFE	Ridge Regression	step=1. n_features_to_select(0.5)	**19**	**0.168639**	0.995872	0.995876	0.988544	0.962146	0.995824	**0.987653**
RFE	Lasso Regression	step=1. n_features_to_select(0.5)	**19**	**0.252960**	0.993647	0.993647	0.985406	0.957668	0.992244	**0.984522**
RFE	Extra Tree Regression	step=1. n_features_to_select(0.5)	**18**	**0.342241**	0.994401	0.994401	0.986994	0.961540	0.994313	**0.986330**
RFE	Linear SVR	step=1. n_features_to_select(0.5)	**18**	**6.788760**	0.994438	0.994437	0.987052	0.958555	0.994463	**0.985789**
Feature Importances	Linear Regression	topn=15. stack=false,relative=true		**1.870455**						
Feature Importances	Ridge Regression	topn=15. stack=false,relative=true		**1.721119**						
Feature Importances	Lasso Regression	topn=15. stack=false,relative=true		**1.874882**						
Feature Importances	Extra Tree Regression	topn=15. stack=false,relative=true		**1.906664**						
Feature Importances	Linear SVR	topn=15. stack=false,relative=true		**2.007809**						
RFECV (yellowbrick)	Linear Regression	step=1. cv=5	**25**	**13.765470**	0.999661	0.999660	0.991935	0.959631	0.999592	**0.990096**
RFECV (yellowbrick)	Ridge Regression	step=1. cv=5	**25**	**14.980706**	0.999661	0.999660	0.991935	0.959212	0.999561	**0.990006**
RFECV (yellowbrick)	Lasso Regression	step=1. cv=5	**25**	**17.979028**	0.999661	0.999660	0.991935	0.959729	0.999610	**0.990119**
RFECV (yellowbrick)	Extra Tree Regression	step=1. cv=5	**27**	**18.757748**	0.999552	0.999550	0.991866	0.961919	0.999535	**0.990484**
RFECV (yellowbrick)	Linear SVR	step=1. cv=5	**29**	**351.008670**	0.999662	0.999660	0.991935	0.951663	0.999586	**0.988501**
Average results of Scikit Learn			**21**	**0.82946**	**0.99619**	**0.99619**	**0.98842**	**0.96007**	**0.99572**	**0.98732**
Average results of Yellowbrick			**26**	**42.58725**	**0.99964**	**0.99964**	**0.99192**	**0.95843**	**0.99958**	**0.98984**
Average results			**23**	**21.708356**	**0.997339**	**0.997339**	**0.989589**	**0.959526**	**0.997003**	**0.988159**

TABLE 4
Results with embedded feature selection methods

Therefore, to select the appropriate features for each output, all techniques must be applied five times on the original set of features. After this, the selected features per output are merged to obtain the total required set of features. This way, selecting the 15 best features (k=15) for each output may result in a final set of 26 features —a result yielded by some of the techniques. It should also be pointed out that, in this case, the embedded techniques yield more remarkable results than the filter techniques. The former selected 23 features on average (*N. features* column of *Average results* row in Table 4), while the latter selected an average of 28 features (*N. features* column of *Average results* row in Table 3).

In the case of embedded methods, the RFE technique yielded sets with fewer than 20 features for any of the estimators tested. Specifically, Extra Tree Regression and Linear SVR are the ones that filtered most features, reducing the set to 18 features compared to the 19 features obtained with the other estimators. The same happens with the SelectFromModel when using the Extra Tree Regression as estimator again. When using this estimator the number of features is reduced to 15, compared to the 25-26 selected when using other estimators, without a remarkable loss of correlation.

Regarding filter methods, the number of features can at best be reduced by five, resulting in sets of 26 features. Several techniques and configurations result in this number of features being selected. However, there are others such as VarianceThreshold—with threshold values such as 0.5 as an intermediate value or 0 as a lower limit value—that filter only one or no features at all, respectively. The same applies to the SelectFpr, SelectFdr, and SelectFwe techniques. All of them are based on the use of an alpha —corresponding to the highest uncorrected p-value for features to be kept— which, if left at its default value, only filters out one feature. However, these techniques also fail to improve correlation by filtering fewer features and retaining more information for the model —since there are still features that are redundant or not linked to the outputs among the resulting 30 features.

Finally, it is necessary to discuss the **results in terms of correlation**. In this case, the embedded techniques offer

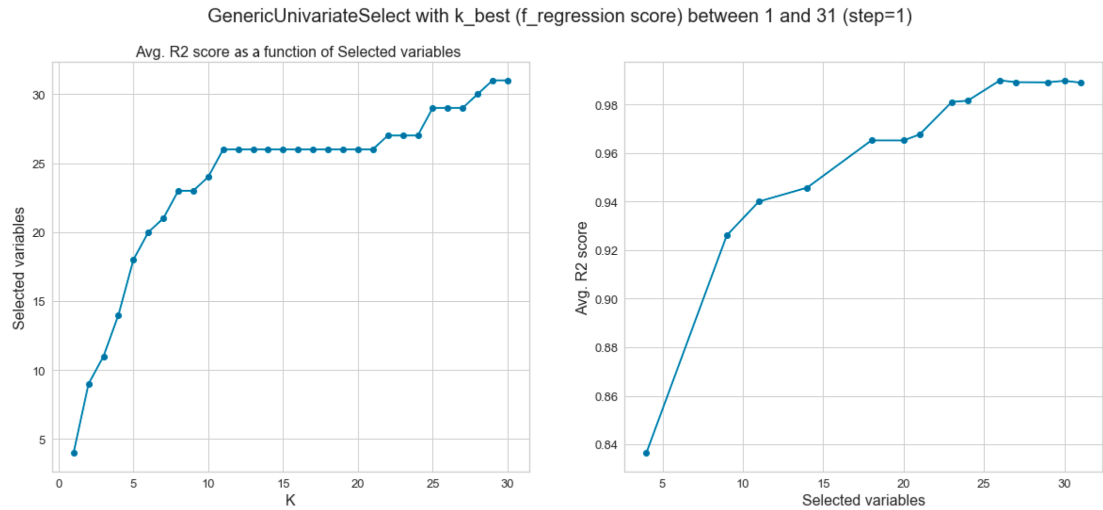

Fig. 1. Number of features selected and R^2 score as a function of the k best selected in GenericUnivariateSelect technique

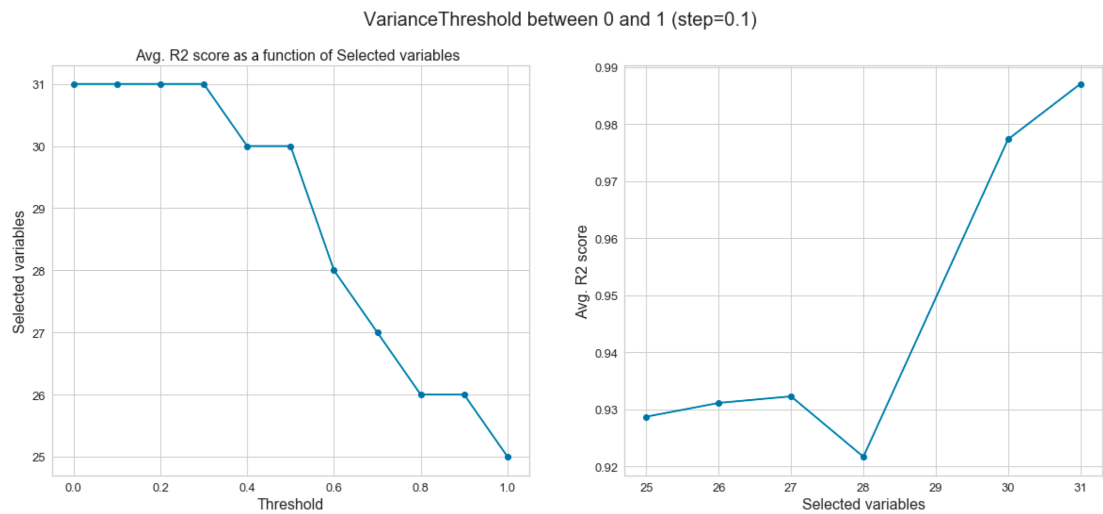

Fig. 2. Number of features selected and R^2 score as a function of the threshold defined in VarianceThreshold technique

more similar results, all achieving a correlation higher than 0.98 (*Avg. R^2 score* column in Table 4). This suggests that they maintain the correlation of the original set of variables, even when considering a smaller number of variables. In the case of the filter methods, the results are similar (*Avg. R^2 score* column in Table 3), although some of the techniques reduce the precision depending on the configuration of the hyperparameters. This is the case of the VarianceThreshold technique when the threshold value is changed from 0 to 0.5 or 0.8. However, even at the risk of lower precision if this particular technique is used, both embedded and filter techniques are successful at filtering features—to a greater degree or lesser degree, as mentioned above—while keeping the original precision close to the ideal value of 1. A correlation of 0.98 or 0.99 is not a decisive factor, both values being very close to the ideal.

As result of the last step (**Step 6**), a series of graphs—one per technique, admitting hyperparameters with values in a finite range—are included to illustrate the relationship between the number of filtered features, the average precision achieved by the models with each selected feature set, and the different techniques and the value obtained by their hyperparameters. These graphs illustrate how the size of the selected feature set changes when the value of any of its hyperparameters is modified within its finite range. It is also possible to appreciate changes in correlation caused by changes in the number of selected features. All these graphs are provided as additional material in the Zenodo repository, inside the "images" folder. In addition, each technique and configuration is linked to its associated graphs in the .XLSX results document. For the sake of brevity, we will only discuss those graphs that we consider to be of greater relevance, since they provide the most easily visually understandable graphs and since they are a significant sample of the different types of hyperparameters that can be configured (such as the thresholds, the number of features to be selected K, or the maximum number of features to be selected).

The first one is that of GenericUnivariateSelect with f_regression as score function. Figure 1 shows the evolution

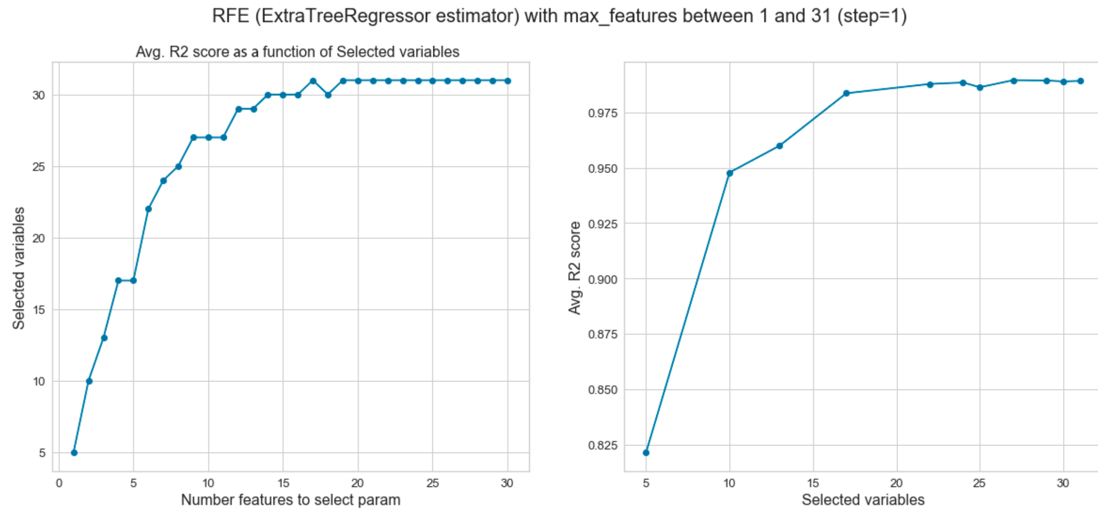

Fig. 3. Number of features selected and R^2 score as a function of the maximum features selected in RFE technique

of the number of features finally selected and average R^2 score for this configuration—taking into account the five models used as estimators during validation—and k best features to be selected as parameter changes. The k best parameter varied in a range from 1 to 31—the total number of features—with an added—i.e., selecting one more feature in each execution. The figure shows how, by selecting one feature per output, a value of 3 features is obtained, with a correlation of almost 0.84. From that point on, correlation increases up to 26 features—especially when k is adjusted to a value of 2 features and selected features increase from 3 to 9. This number of features is obtained by a wide range of k. The figure also shows that the tested models perform better with this number of features than with the original 31, losing correlation when more features are selected.

Among the filter methods, it is also interesting to highlight the VarianceThreshold graph (Figure 2), showing variations as the threshold was changed. After the 0.5 threshold value, the technique begins to filter features. It is interesting to see how far it filters when the entire spectrum of threshold values is tested—from 0.0 to 1.0 with a step of 0.1. The figure shows that, after the 0.4 threshold, it starts to reduce the space of features. In addition, as the threshold increases, the number of features selected decreases. This is expected since by increasing the threshold, the variance of the features that can be selected becomes more restrictive. When the optimum point is reached, it is subjected to the objective pursued during the process of Feature Selection. The 0.0 threshold is obtaining the best correlation but using all features, while 1.0 is obtaining the worst correlation, but with fewer features. Even a middle point, such as the 0.7 threshold—selecting a set of 27 features that provides an increase in correlation with respect to that offered by the 26 or 28 features set offered by other thresholds— can be considered as an optimum point.

Finally, it is also interesting to discuss the graphs corresponding to the embedded RFE method—those that filter out more features. Specifically, Figure 3 shows the evolution of this technique when an Extra Tree Regression model is used as estimator and the maximum number of features to be selected per output is adjusted. In this case, the number of features to be selected is more progressive than in the other techniques. For the first values, the result obtained is very similar to that already yielded by the GenericUnivariateSelect technique in Figure 1. It can also be observed that the optimal correlation point corresponds to 26 features. This number of features is obtained when a maximum number of 8-10 features per output are selected. For any selection higher than 16 features per output the original 31 features are selected—except in the case of 22 features per output when 30 variables are selected. In this case, the R^2 score plot shows that this change slightly improves the precision.

To conclude, it can be argued that the embedded methods select a smaller group of features than the filters, achieving even better correlation on average—at the cost of some of these techniques requiring slightly higher execution times. Almost all embedded techniques filter more features than the best-performing filter method. The most restrictive embedded technique manages to reduce the number of variables to 50% while maintaining correlation above that of some of the filter techniques, such as the variance threshold at 0.8. This last technique only obtains a reduction of approximately 12% and a correlation of 0.92. It is also interesting to indicate that, for filter techniques, the function score *mutual_info_regression* notably increases the time, without improving correlation.

5 Improving Functional Profile Assessments

Taking as a starting point the results obtained in the comparison above, two models have been developed using the features selected by one of the analyzed Feature Selection methods as shown in Figure 4. These models predict the five functional profile scores of the patients considered: one model with higher accuracy—but using more data—, and the other with slightly less accuracy—but using a smaller amount of data.

The creation of two models illustrates the dual potential of Feature Selection techniques. Figure 4 shows the process followed for the creation of the two models and the use for which each model is developed. On the one hand, *Model 1* filters out a smaller number of features while offering better precision. This could be used to measure the functional profile based on the new questionnaires to be fulfilled by health professionals—now with 25 questions instead of 31. This model offers health professionals measurements of the different scores of the functional profile of their aging patients with almost as much precision as the original model, but with fewer questions. On the other hand, *Model 2* filters out a larger number of features (from 31 features to 14), offering slightly less precision. This model could be used by informal caregivers or by the patients themselves to perform periodic self-assessments, thus providing a more constant flow of information on their functional profile. If any irregularities were to be observed whilst applying this model, health professionals could get involved and start using *Model 1* to obtain more accurate measurements.

For the creation of both models, several of the analyzed Feature Selection methods were used rather than just one. First, because each of these models was generated using different methods. Besides, as suggested by researchers such as Chen et al. [26] our models combined filtering and embedded methods. As illustrated in Figure 4, the feature selection phase before the creation of the Machine Learning model has performed in two stages: first a filtering method then an embedded method is applied per model in both stages. This procedure yields better results since the filtering methods applied only filter out the most irrelevant features and the embedded methods used subsequently complete the elimination of features based on the result offered by the first ones.

The filter method used in both models was the GenericUnivariateSelect with an *f_regression* function score and a percentile of 50—since this technique and configuration removes a few features while maintaining the precision of the original set. As an embedded method, in the case of *Model 1* we used the SelectFromModel with the default configuration and RidgeRegression as an estimator. In the comparison, this reduced the feature set to 25 features with an average correlation of 0.992—the highest score of all the tested Machine Learning models. For *Model 2* we used SelectFromModel again with the default configuration but changed the estimator to the Extra Tree Regressor since, in the comparison, it was the one that eliminated the highest number of features.

Finally, as a Machine Learning model for the creation of the prediction model, we applied in both cases Linear Regression—the one offering the best prediction among the five analyzed. As in the comparison, a Linear Regression model was needed for each output. Therefore Scikit-learn's MultiOutputRegressor was used to encapsulate the creation of these five models in a single one. The results obtained in terms of correlation between inputs and outputs and prediction error, using the same dataset as during the comparison, are shown in Table 5.

The results show both models achieve good correlation

Metric	Model 1	Model 2
Coefficient of determination (R^2)	0.999661	0.981238
Mean Square Error (MSE)	0.111246	5.772708
Mean Absolute Error (MAE)	0.178976	1.552575

TABLE 5
Results yielded by different models proposed

measured with the R^2 score. However, the error in the predictions, measured with the Mean Square Error (MSE) and Mean Absolute Error (MAE), is higher in the model with fewer features, as expected. The closeness between MSE and MAE values indicates that errors in *Model 1* are of small magnitude—since the MSE tends to give more weight to large errors and to increase its value concerning the MAE when there are many large errors. Moreover, in this case, the MSE has a small value, indicating errors in the order of less than one unit. The MAE value also shows that the mean difference between the actual value of the functional profile scores and the estimator in *Model 1* is very small. Seeing this value and that the functional profile scores oscillate in a range of values between 0 and 100, the error introduced by the model is minimal. This, together with its high precision, confirms the validity of the model to predict the functional profile of aging patients with confidence. On the other hand, *Model 2*'s MAE shows an error of around 1.55 units on average. However, the MSE does show that, in this case, there are situations where the value of the score oscillates more, producing more significant errors, which penalize this metric compared to that of the MAE. However, the results are positive in both models and adapted to the needs of each one.

Applying a 5-step cross-validation on these models, a mean value of R^2 score of 0.99964 has been obtained for Model 1, with a deviation between the R^2 score obtained in each step of 7.62360x10^{-5}, which shows that the results of the model are reliable, being close to those obtained in the validation with the test set. For Model 2, the results were a mean R^2 score of 0.98166, with a deviation of 4.61616x10^{-3}. This shows that this model is a little less reliable, but still gives very good results to be used as a preventive model.

6 Discussion

We presented a comparison of the different Feature Selection techniques applied to regression problems and showed how the results were applied to the creation of two Machine Learning models for the prediction of the functional profile of older adults. We discuss a few important aspects of our work in this section.

First of all, it is important to note that all the work presented in this paper could be extrapolated to other case studies (within the health or other domains) which require applying Feature Selection on regression problems—i.e., new case studies or other types of health assessments, such as older adults' dementia or loneliness assessments.

Regarding the comparison of the different Feature Selection techniques on regression problems, it has been shown

Fig. 4. Proposal for improving functional profile assessments with Feature Selection and Machine Learning

that these techniques are more efficient to reduce the features within a dataset such as the older adults' functional profile. Some techniques can reduce the initial feature set containing 31 features to 25 or even 14—which is particularly relevant since all these features are currently being used in well-validated clinical procedures for the assessment of functional profiles. This stresses the great potential of these techniques for regression problems.

We have also found that the performance of the filter and embedded methods presented theoretically and practically in previous studies for classification problems is maintained when these techniques are applied to regression problems. In general, filter methods achieved less sifting than embedded methods, but with a much better performance in terms of computational time. Execution time was not a challenge for the present case study where all the results, except those of the Yellowbrick RFECV technique, were obtained in the range of tenths/hundredths of seconds for Scikit Learn methods or a few seconds for the case of Yellowbrick methods—due to the fact of being wrapping Scikit Learn methods. Even more considering the reduced capabilities of the computer where the test has been executed. This, however, could be different in other case studies where execution time may be a potential challenge.

Regarding the filtering percentage, the results obtained could act as a reference for the application of Feature Selection techniques on other regression problems. Our comparison could be used to analyze the performance gaps between the different Feature Selection techniques tested. However, it is important to note that we do not recommend extrapolating the filtering percentage as an exact percentage to other case studies [8]. For each case, the filtering percentage will depend on the quality of the initial data set—i.e., the number of irrelevant features or the number of correlated features—of the new case study.

Another important issue to discuss is whether the features that have been retained in the two models are overlapping. That is, whether all the features of the 14-feature model are included as entries in the 25-feature model. In this regard, it has been verified that this is the case. To be more precise about which items of the original 31 have been retained, the 25-feature model has retained 6 of the 10 body functions items, 16 of the 17 body structure items, and 3 of the 4 environmental factors items. The 14-feature model has retained 4, 7 and 3 items from each group, respectively. As can be seen, the first model filters mostly body function items, so these may be the ones with more overlapping information. The second model continues the path filtering a bit more of these items, but mostly filtering body structure items. The least filtered are the environmental factors, filtering only 1 item for both cases.

On the clinical impact implications of eliminating the number of features we eliminated, we have evaluated the amount of time that can be saved on each functional profile assessment with our proposal. Originally, with the 31 items, it took an average of 20 minutes to assess a patient (an average of 38.70 s per item) when the diagnosis was made by an experienced caregiver. The same caregiver now needs an average of 15 minutes for the same diagnosis, through the new version with only 25 questions (improving the average per item to 36 s). As can be seen, from the caregiver's point of view, their process has been optimized by 25%. As for the self-diagnosis by the elder, we have determined that the elders aged 65 to 80 years need 10 minutes on average to fill in the reduced form of 14 questions (an average of 42 s per item) and that the elders aged more than 80 years need 15 minutes (an average of 64 s per item). For that, a reformulation of the question to make them more user-friendly to the elders is needed.

This improvement is due to the fact that the form used by the caregivers has been reduced by almost 20%, from 31 items to 25 items. However, such filtering does not always have to be achieved, and there may be situations where only a small part of the items can be filtered..

Regarding the impact of filtering few features, from the sanitary point of view we can consider two interesting situations. First, if it is more convenient to keep all the features because the time optimization to be achieved is very small and it is better to keep all the information than to obtain that optimization. Second, if it is better to eliminate those few features because they help to optimize the assessment times (even if only a little). What is best in each case is highly subjective, and depends largely on the interpretation of the health professional in charge and the results obtained in the particular case study. In this case, any filtering obtained is interesting, even if it is small. It does not matter too much to lose certain information, since it is not a case study in which a disease that requires great precision in its diagnosis is being detected.

It is also relevant to discuss the application of these Feature Selection techniques to the assessment of functional profiles. Two Machine Learning models were created to establish the functional profile scores using fewer data. Both the model that selected a smaller number of features and the model that selected a larger number yielded correlation between inputs and outputs close to 100%. For *Model 1*, the error included was so slight that it could be considered negligible. Therefore, the values yielded by this model are as accurate as those provided by the original features set. For the *Model 2*, the error is still not very significant: 1.55 units on average, with some cases where it is slightly more pronounced. As suggested above, this could be used as a preventive model used by informal caregivers to quickly obtain more frequent, approximate functional scores. In case the scores suggest a possible problem, health professionals could then apply the more accurate model.

Our results suggest novel possibilities for the assessment of the functional profile of older adults—one of the most costly issues in terms of human healthcare resources, given that loss of functionality is an inherent consequence of aging. With this proposal, informal caregivers can continuously measure their patients' functional profiles changes, with two advantages: on the one hand, health professionals have to perform fewer assessments while their patients' functional profiles can be continuously monitored. In addition, the assessments performed by health professionals would be faster and more efficient, since fewer parameters would be assessed to obtain the same results. This would benefit healthcare systems, by freeing up time that could be devoted to providing better care. Therefore, the objective of this paper has been achieved and it can be considered a declaration that Feature Selection on regression problems can be employed as the technological tool that can improve the assessment of health pathologies and conditions.

7 Related Work

Dimensionality Reduction (DR) techniques have become very important in Data Science due to the huge amount of data generated every day in the so-called Big Data era [19]—with more and more works applying these techniques to different domains.

Health care is one of the fields where the use of these techniques is imperative. Feature dimension is one of the main challenges in the analysis of healthcare data [9], [12], [32]. Medical tests often generate significant amounts of data [33]—input variables or features—which are difficult to analyze directly by using Data Mining techniques such as Machine Learning or Deep Learning. Thus, proposals such as that of Lee et al. [12] suggested the use of DR techniques to analyze Electronic Medical Records (EMR). They argued that these records were usually composed of hundreds to thousands of medical variables, causing a problem of data high-dimensionality—the so-called curse of dimensionality [21].

Within DR techniques, two types of approaches can be established [12]: Feature Extraction and Feature Selection. Both approaches are not mutually exclusive and are often used together [34]. Feature Extraction methods aim to extract a series of new features from existing features in a dataset. These new features will be more expressive than the original ones, thus allowing a better analysis of the data. This is the approach followed by proposals such as one of Nuñez-Godoy et al. [35], who employ Feature Extraction's Principal Component Analysis (PCA) algorithm to reduce the high-dimensionality of sensor-gathered data to detect patterns in human-sitting-poses. The application of these techniques helps improve the results yielded by the initial prediction models, using the original feature set, while reducing the computational cost. However, the use of these techniques still implies collecting the same number of input variables, so they do not provide a solution—or, at least, not by themselves—to the problem addressed in this paper.

On the other hand, Feature Selection techniques aim to directly reduce the set of features to be collected as input. This way, not only do they improve the performance of prediction models [8]—both in terms of accuracy and computational costs—they also help analyze the features that are collected. Their analytical potential results in several advantages, such as a better understanding of the problem. The Feature Selection process helps eliminate features that are redundant or have no real relation to the output—even if, a priori, healthcare professionals believe that they do.

For these reasons, these types of techniques have been widely used for years. Neumann et al. [36] presented some continuous Feature Selection approaches, measuring their performance in different health datasets. Wei and Billings [37] developed a new unsupervised forward orthogonal search (FOS) for Feature Selection and applied it to different health datasets from the UCI Machine Learning Repository. Maldonado et al. [38] proposed a wrapper method based on the use of kernel functions with Support Vector Machines for classification problems. They compared their method with other existing ones using different datasets, including the Colorectal Microarray (CRMA) and the Wisconsin Breast Cancer (WBC) datasets. Huang et al. [2] applied a filter-based method to an antidepressant medication utilization study, discussing the advantages of this method with respect to a wrapper method on logistic regression. Genuer et al. [39] proposed the use of random forests in wrapper Feature Selection techniques, applying them on four different classification datasets, corresponding to Colon, Leukemia, Lymphoma, and Prostate diseases, and providing some experimental metrics. Jeanneret et al. [40] performed a chemistry-driven Feature Selection in order to analyze urine samples corresponding to dioxin intoxication. Javed et al. [41] demonstrate the performance of filter methods on three different datasets belonging to binary classification (two-class) problems, among them a drug discovery and a heart disease classification dataset. Cateni et al. [42] proposed the combination of different Feature Selection techniques in a two-phase procedure: first using filter methods and then an exhaustive search. This proposal was validated on several UCI datasets, among others. Bodur and Douglas [14] developed and tested a filter algorithm for Feature Selection in healthcare datasets. Uphade et al. [43] make use of these techniques also in classification problems to try to detect which factors should be taken into account in covid patients to try to place them in a range of days necessary for their recovery. Bommert et al. [44] compare the use of different filter methods to select the most relevant features in gene expression within bioinformatics. They focus on this type of methods because they are more computationally efficient than wrappers, being a better choice for high-dimensional data such as the 14 datasets used for the comparison. Outside of the health domain, Guo et al. [45] proposed a method to perform regression feature selection with noised data —or data with outliers— when the distribution of representation error is unknown. Degeest et al. [46] discuss the importance of Feature Selection as a preprocessing step in Machine Learning, highlighting the relevance of filter methods and centering in the context of regression problems.

While some proposals offer new Feature Selection algorithms, others delve into the existing ones and their use in the health domain. Remeseiro and Bolon-Canedo [47] reviewed the Feature Selection techniques in medical applications. Suresh et al. [13] defined a full process, beginning with data cleansing and outlier detection, following with dimension reduction using SVM-RFE, and ending with the evaluation of results on the RFS-SVM algorithm and other traditional classifiers. They applied this process on a Prima Indian diabetes dataset. Chen et al. [26] proposed the use of ensemble methods for Feature Selection on health data, consisting of the use of filter, wrapper, and embedded methods together. Chicco and Oneto [16] showed in their research that machine learning could be used to predict sepsis shock, as well as demonstrating that Feature Selection could be employed to identify unexpected symptoms and relevant clinical components of septic shock. Xu et al. [48] proposed a general framework that, using global redundancy minimization in orthogonal regression, emphasized the importance of the correlation between features in the Feature Selection processes. Ali et al. [49] use this type of techniques to enable the creation of Machine Learning models capable of predicting drug response according to omics profiles in precision oncology applications.

In addition, proposals such as that of Arowolo et al. [34] employed these techniques to select the most relevant features within a dataset, before employing Feature Extraction techniques to improve the accuracy of prediction models. This approach was used to analyze in a more efficient way data from the Malaria Vector Gene Expression. Other proposals such as that of Murthy [50] employed the two techniques in a reverse process: first new features were generated with Feature Extraction and then the original features plus the new ones were filtered with Feature Selection. The framework they defined for this is called Minimum Projection error Minimum Redundancy (MPeMR).

However, most applications of Feature Selection are linked to classification problems—aiming to predict to which discrete class from within a dataset a sample belongs, based on a set of inputs. When the expected output is a continuous value—i.e., a regression problem—the number of proposals available in the literature is much smaller. Of the above-mentioned papers, only that of Chicco and Oneto [16] uses Feature Selection on regression problems with health data—ignoring previously published methods, such as that of Huang et al. [2]. Outside the health domain, examples are scarce. Javidi et al. [17] proposed a new Feature Selection technique on regression problems based on the use of the Biology Migration Algorithm (BMA) for Feature Selection and game theory for feature clustering. To test its efficiency, its authors evaluated it on several datasets from different domains belonging to regression problems.

Furthermore, as far as the authors know, no works have reviewed Feature Selection techniques applied to regression problems and compared their performance on a specific case study. For example, Chandrashekar [8] reviews Feature Selection techniques applied to classification problems, offering a comparison of their performance on different health datasets. El Aboudi et al. [25] also provide an overview of different Feature Selection techniques on classification problems, focusing on wrapper methods. Finally, Remeseiro et al. [47] discuss the importance of Feature Selection on medical applications, highlighting the high number of regression problems existing in this domain. However, their techniques' comparison only covers classification problems' datasets.

Finally, our work includes also Machine Learning and Deep Learning tasks. Since there are many Machine Learning algorithms for the definition of prediction models in regression problems, the existing literature was reviewed. Following the recommendations of Jolly et al. [30], tests were conducted on the most common algorithms—linear regression, logistic regression, support vector machines, and tree-based algorithms. Despite using Deep Learning, a Multi-Layer Perceptron Network (MLP) was chosen. Our choice of the most suitable neural network is based on the study by Bhavya and Pillai [10], which discusses the suitability of several neural network types for the analysis of healthcare data.

8 Conclusions and Future Work

This paper aimed to reduce the number of parameters that health professionals must consider to assess an older adult. Feature Selection techniques were used to reduce an initial set of features into a smaller subset containing the relevant information from the initial set while eliminating redundant features.

We created two different Machine Learning models using reduced sets of features to predict functional profile scores: one using fewer features that lose some reliability; and another using more features but with higher reliability. Both models could be used together—one as a preventive model so that the patient can be constantly monitored, and the other to be used only by health professionals. The input features used in each model were obtained by using Feature Selection. For this purpose, an initial comparison of different techniques and configurations was carried out, taking into account the computational time taken in selection, the number of features selected, and the level of correlation between inputs and outputs achieved.

We argue that the use of Feature Selection techniques in the healthcare domain is feasible when working on a regression problem since it can improve the results and help health professionals in performing assessments. We also argue that they can be used in situations where accuracy is very high—close to 100%—but it is necessary to reduce the number of collected variables. This paper does not aim to demonstrate that better predictions are obtained with fewer variables after using Feature Selection. Rather, it shows that predictions of the same quality can be obtained with fewer variables. This should help improve the sustainability of healthcare systems—saving resources and time, which can be then spent on more pressing tasks. Additionally, the comparison provided serves as a starting point for future work seeking to employ these techniques through implementations already available in the most widely used Data Science tools.

The data employed in this paper was taken from the MIAPe platform which collects assessment data of older adults living in assisted living facilities and nursing homes, including functional profile data.

As future work, one important topic is to include wrappers methods into the evaluation, since space constraints have not allowed us to include them in this study. As well, we will aim to create a framework that allows health professionals to interact with the algorithms or feature selection techniques to filter and reduce the features. In this way, they will be able to eliminate features manually based on recommendations, checking how the accuracy of the models is altered. This would allow the model to adapt better to the real needs of healthcare professionals. It would also help to resolve the uncertainties left by the automatic application of Feature Selection. To do this, we could include in the Feature Selection process those techniques tested in the comparison that generated visual results to show correlations, mutual information, and other issues of interest to the users when they filter features.

The need for a flexible and interactive framework such as the one mentioned above is clear since the automatic techniques do not integrate critical domain knowledge or take into account how complicated it can be to establish the value of a specific feature. For instance, they do not consider the relationship that might exist between two features when collecting their values—i.e., the value of two features can be obtained jointly, and it might be interesting to keep both despite one of them providing little information about what is being diagnosed. They do not consider either the importance that a specific feature might have to link the diagnosis of a disease with another one. All these issues could be avoided if users were in the loop of the Feature Selection process.

Acknowledgments

This work was supported by the projects 0499_4IE_PLUS_4_E (Interreg V-A España-Portugal 2014-2020) and RTI2018-094591-B-I00 (MCIU/AEI/FEDER, UE), the FPU19/03965 grant, the Department of Economy and Infrastructure of the Government of Extremadura (GR18112, IB18030), and the European Regional Development Fund.

References

[1] J. R. Gonçalves, I. Ramalhinho, B. L. Sleath, M. J. Lopes, and A. M. Cavaco, "Probing pharmacists' interventions in long-term care: a systematic review," *European geriatric medicine*, pp. 1–21, 2021.

[2] S. H. Huang, L. R. Wulsin, H. Li, and J. Guo, "Dimensionality reduction for knowledge discovery in medical claims database: Application to antidepressant medication utilization study," *Computer Methods and Programs in Biomedicine*, vol. 93, no. 2, pp. 115–123, feb 2009.

[3] D. Kalra, "Electronic health record standards," *Yearbook of medical informatics*, vol. 15, no. 01, pp. 136–144, 2006.

[4] E. Moguel, J. Berrocal, J. M. Murillo, J. García-Alonso, D. Mendes, C. Fonseca, and M. Lopes, "Enriched elderly virtual profiles by means of a multidimensional integrated assessment platform," in *Procedia Computer Science*, vol. 138. Elsevier B.V., jan 2018, pp. 56–63.

[5] C. M. Bell, S. S. Araki, and P. J. Neumann, "The association between caregiver burden and caregiver health-related quality of life in alzheimer disease," *Alzheimer Disease & Associated Disorders*, vol. 15, no. 3, pp. 129–136, 2001.

[6] C. Silveira, R. T. Souza, M. L. Costa, M. A. Parpinelli, R. C. Pacagnella, E. C. Ferreira, J. Mayrink, J. P. Guida, M. H. Sousa, L. Say *et al.*, "Validation of the who disability assessment schedule (whodas 2.0) 12-item tool against the 36-item version for measuring functioning and disability associated with pregnancy and history of severe maternal morbidity," *International Journal of Gynecology & Obstetrics*, vol. 141, pp. 39–47, 2018.

[7] S. H. Park, E. A. Demetriou, K. L. Pepper, Y. J. C. Song, E. E. Thomas, I. B. Hickie, N. Glozier, and A. J. Guastella, "Validation of the 36-item and 12-item self-report world health organization disability assessment schedule ii (whodas-ii) in individuals with autism spectrum disorder," *Autism Research*, vol. 12, no. 7, pp. 1101–1111, 2019.

[8] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers and Electrical Engineering*, vol. 40, no. 1, pp. 16–28, jan 2014.

[9] S. Itani, F. Lecron, and P. Fortemps, "Specifics of medical data mining for diagnosis aid: A survey," pp. 300–314, mar 2019.

[10] S. Bhavya and A. S. Pillai, "Prediction models in healthcare using deep learning," in *Advances in Intelligent Systems and Computing*, vol. 1182 AISC. Springer, dec 2021, pp. 195–204.

[11] J. Rojo, E. Moguel, C. J. Fonseca, M. Lopes, J. G. Alonso, and J. Hernández, "Time series forecasting to predict the evolution of the functional profile of the elderly persons," in *Gerontechnology III: Contributions to the Third International Workshop on Gerontechnology, IWoG 2020, October 5-6, 2020, Évora, Portugal*. Springer, 2021, pp. 11–22.

[12] C. Lee, Z. Luo, K. Y. Ngiam, M. Zhang, K. Zheng, G. Chen, B. C. Ooi, and W. L. J. Yip, "Big Healthcare Data Analytics: Challenges and Applications." Springer, Cham, 2017, pp. 11–41. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-58280-1{_}2

[13] A. Suresh, R. Kumar, and R. Varatharajan, "Health care data analysis using evolutionary algorithm," *Journal of Supercomputing*, vol. 76, no. 6, pp. 4262–4271, jun 2020. [Online]. Available: https://doi.org/10.1007/s11227-018-2302-0

[14] Bodur and Atsa'am, "Filter Variable Selection Algorithm Using Risk Ratios for Dimensionality Reduction of Healthcare Data for Classification," *Processes*, vol. 7, no. 4, p. 222, apr 2019. [Online]. Available: https://www.mdpi.com/2227-9717/7/4/222

[15] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *Journal of machine learning research*, vol. 5, no. Aug, pp. 845–889, 2004.

[16] D. Chicco and L. Oneto, "Data analytics and clinical feature ranking of medical records of patients with sepsis," *BioData Mining*, vol. 14, no. 1, dec 2021.

[17] M. M. Javidi, "Feature selection schema based on game theory and biology migration algorithm for regression problems," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 2, pp. 303–342, feb 2021.

[18] R. F. Xu and S. J. Lee, "Dimensionality reduction by feature clustering for regression problems," *Information Sciences*, vol. 299, pp. 42–57, apr 2015.

[19] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava, and T. Baker, "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54 776–54 788, 2020.

[20] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: a comparative," *J Mach Learn Res*, vol. 10, no. 66-71, pp. 1–36, 2009.

[21] F. Y. Kuo and I. H. Sloan, "Lifting the curse of dimensionality," *Notices of the AMS*, vol. 52, no. 11, pp. 1320–1328, 2005.

[22] B. Venkatesh and J. Anuradha, "A review of feature selection and its methods," *Cybernetics and Information Technologies*, vol. 19, no. 1, pp. 3–26, 2019.

[23] G. Wei, J. Zhao, Y. Feng, A. He, and J. Yu, "A novel hybrid feature selection method based on dynamic feature importance," *Applied Soft Computing*, vol. 93, pp. 1–13, 2020.

[24] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.

[25] N. El Aboudi and L. Benhlima, "Review on wrapper feature selection approaches," in *Proceedings - 2016 International Conference on Engineering and MIS, ICEMIS 2016*. Institute of Electrical and Electronics Engineers Inc., nov 2016.

[26] C. W. Chen, Y. H. Tsai, F. R. Chang, and W. C. Lin, "Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results," *Expert Systems*, vol. 37, no. 5, pp. 1–10, oct 2020. [Online]. Available: https://doi.org/10.1111/exsy.12553

[27] C. Fonseca, M. Lopes, D. Mendes, P. Parreira, L. Mónico, and C. Marques, "Psychometric properties of the elderly nursing core set," in *International Workshop on Gerontechnology*. Springer, 2018, pp. 143–153.

[28] M. J. Lopes and C. Fonseca, "Processo de construçao do elderly nursing core set," *Journal of Aging & Inovation*, vol. 2, no. 1, pp. 121–131, 2013.

[29] W. H. Organization *et al.*, "Ifc: International classification of functioning, disability and health," 2001. [Online]. Available: https://www.who.int/standards/classifications/international-classification-of-functioning-disability-and-health

[30] K. Jolly, *Machine Learning with scikit-learn Quick Start Guide: Classification, regression, and clustering techniques in Python*. Packt Publishing Ltd, 2018.

[31] "sklearn.metrics.r2score | scikit−learn 1.0.2 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html

[32] J. Rojo, J. Hernandez, and J. M. Murillo, "A Personal Health Trajectory API: Addressing Problems in Health Institution-Oriented Systems," *Lecture Notes in Computer Science*, vol. 12128, pp. 519–524, 2020. [Online]. Available: http://link.springer.com/10.1007/978-3-030-50578-3{_}37

[33] J. Rojo, J. Hernández, J. M. Murillo, and J. Garcia-Alonso, "Blockchains' federation for integrating distributed health data using a patient-centered approach," in *2021 IEEE/ACM 3rd Interna-*

tional Workshop on Software Engineering for Healthcare (SEH). IEEE, 2021, pp. 52–59.

[34] M. O. Arowolo, M. O. Adebiyi, A. A. Adebiyi, and O. J. Okesola, "A Hybrid Heuristic Dimensionality Reduction Methods for Classifying Malaria Vector Gene Expression Data," *IEEE Access*, vol. 8, pp. 182 422–182 430, oct 2020.

[35] S. Nunez-Godoy, V. Alvear-Puertas, S. Realpe-Godoy, E. Pujota-Cuascota, H. Farinango-Endara, I. Navarrete-Insuasti, F. Vaca-Chapi, P. Rosero-Montalvo, and D. H. Peluffo, "Human-sitting-pose detection using data classification and dimensionality reduction," in *2016 IEEE Ecuador Technical Chapters Meeting (ETCM)*. IEEE, oct 2016, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/7750822/

[36] J. Neumann, C. Schnörr, and G. Steidl, "Combined SVM-based feature selection and classification," *Machine Learning*, vol. 61, no. 1-3, pp. 129–150, nov 2005. [Online]. Available: https://link.springer.com/article/10.1007/s10994-005-1505-9

[37] H. L. Wei and S. A. Billings, "Feature subset selection and ranking for data dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 162–166, jan 2007.

[38] S. Maldonado and R. Weber, "A wrapper method for feature selection using Support Vector Machines," *Information Sciences*, vol. 179, no. 13, pp. 2208–2217, jun 2009.

[39] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2225–2236, oct 2010. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167865510000954

[40] F. Jeanneret, J. Boccard, F. Badoud, O. Sorg, D. Tonoli, D. Pelclova, S. Vlckova, D. N. Rutledge, C. F. Samer, D. Hochstrasser, J.-H. Saurat, and S. Rudaz, "Human urinary biomarkers of dioxin exposure: Analysis by metabolomics and biologically driven data dimensionality reduction," *Toxicology Letters*, vol. 230, no. 2, pp. 234–243, oct 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S037842741301374X

[41] K. Javed, H. A. Babri, and M. Saeed, "Impact of a metric of association between two variables on performance of filters for binary data," *Neurocomputing*, vol. 143, pp. 248–260, nov 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231214007358

[42] S. Cateni, V. Colla, and M. Vannucci, "A Hybrid Feature Selection Method for Classification Purposes," in *2014 European Modelling Symposium*. IEEE, oct 2014, pp. 39–44. [Online]. Available: http://ieeexplore.ieee.org/document/7153972/

[43] D. Uphade and A. Muley, "Identification of parameters for classification of covid-19 patient's recovery days using machine learning techniques," *J. Math. Comput. Sci.*, vol. 12, pp. Article–ID, 2022.

[44] A. Bommert, T. Welchowski, M. Schmid, and J. Rahnenführer, "Benchmark of filter methods for feature selection in high-dimensional gene expression survival data," *Briefings in Bioinformatics*, vol. 23, no. 1, pp. 1–13, 2022.

[45] Y. Guo, W. Wang, and X. Wang, "A robust linear regression feature selection method for data sets with unknown noise," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[46] A. Degeest, B. Frénay, and M. Verleysen, "Reading grid for feature selection relevance criteria in regression," *Pattern Recognition Letters*, vol. 148, pp. 92–99, 2021.

[47] B. Remeseiro and V. Bolon-Canedo, "A review of feature selection methods in medical applications," pp. 1–9, sep 2019.

[48] X. Xu, X. Wu, F. Wei, W. Zhong, and F. Nie, "A general framework for feature selection under orthogonal regression with global redundancy minimization," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[49] M. Ali and T. Aittokallio, "Machine learning and feature selection for drug response prediction in precision oncology applications," *Biophysical reviews*, vol. 11, no. 1, pp. 31–39, 2019.

[50] C. A. Murthy, "Bridging Feature Selection and Extraction: Compound Feature Generation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 4, pp. 757–770, apr 2017.

Javier Rojo is a PhD student and research fellow at the University of Extremadura, Spain, through a grant for the formation of university professors from the Spanish Ministry of Science, Innovation, and Universities. He obtained his MSc. in Computer Science from the University of Extremadura in 2021. His research interests include software engineering, eHealth, pervasive computing, mobile computing, and gerontechnology.

Lara Guedes de Pinho is a Researcher at the University of Évora and the Comprehensive Health Research Centre (CHRC). She obtained a PhD in Nursing and Health from the University Rovira i Virgili, Spain, in 2021 and a PhD in Psychology from the University of Aveiro, Portugal, in 2018. Her research interests include aging, eHealth, functioning and mental health.

César Fonseca completed his PhD in Nursing at the University of Lisbon, Portugal. He is currently Professor at the University of Évora, and researcher at the Comprehensive Health Research Center (CHRC) - Research, Education, Training and Innovation in Clinical research and Public Health – Universidade Nova de Lisboa.

Manuel José Lopes is a Full Professor at the Nursing School of the University of Évora, Portugal, and a researcher at the Comprehensive Health Research Centre (CHRC). His research interests are aging and functionality, and models of care.

Sumi Helal is a Full Professor at the University of Florida, USA, and Director of its Mobile and Pervasive Computing Laboratory. His areas of research span Digital Health and the Internet-of-Things with applications to active and healthy living and aging, assistive environments (e.g., smart homes) and independent living. Professor Helal is a Boilermaker (Ph.D., Purdue University, class of 1991), Fellow of the IEEE, Fellow of the AAAS, Fellow of the IET, and a member of Academia Europaea.

Juan Hernández obtained a PhD degree in Computer Science at the Technical University of Madrid, Spain, in 1995. He is currently a Full Professor at the University of Extremadura, Spain, and the Head of the Quercus Software Engineering Group. His research interests include service-oriented computing, ambient intelligence, and model-driven development.

Jose Garcia-Alonso is an Associate Professor at the University of Extremadura, Spain, where he complete his PhD in software engineering in 2014. He is the co-founder of Gloin, a software consulting company, and Health and Aging Tech, an eHealth company. His interests include software engineering, mobile computing, pervasive computing, eHealth, and gerontechnology.

Juan M. Murillo is Full Professor at the University of Extremadura, Spain, and co-founder of the start-up Gloin. His research interests include software architectures, Pervasive and Mobile Computing and Quantum Computing. Murillo holds a PhD in computer science from the University of Extremadura.