Advances in Artificial Intelligence and Machine Learning; Research 3 (1) 760-777 Received 06-01-2023; Accepted 22-02-2023; Published 03-03-2023

An Experimental Study of Dimension Reduction Methods on Machine Learning Algorithms with Applications to Psychometrics

Sean H. Merritt sean.merritt@cgu.edu
Department of Economics
Claremont Graduate University
150 E 10th St, Claremont, CA, 91711

Alexander P. Christensen alexander.christensen@vanderbilt.edu
Department of Psychology and Human Development
Vanderbilt University
Nashville, TN, 37203

Corresponding Author: Sean H. Merritt

Abstract

Developing interpretable machine learning models has become an increasingly important issue. One way in which data scientists have been able to develop interpretable models has been to use dimension reduction techniques. In this paper, we examine several dimension reduction techniques including two recent approaches developed in the network psychometrics literature called exploratory graph analysis (EGA) and unique variable analysis (UVA). We compared EGA and UVA with two other dimension reduction techniques common in the machine learning literature (principal component analysis and independent component analysis) as well as no reduction in the variables. We show that EGA and UVA perform as well as the other reduction techniques or no reduction. Consistent with previous literature, we show that dimension reduction can decrease, increase, or provide the same accuracy as no reduction of variables. Our tentative results find that dimension reduction tends to lead to better performance when used for classification tasks.

Keywords: Dimension reduction, Exploratory Graph analysis, PCA, ICA, Machine learning, Interpretability

1. INTRODUCTION

Machine learning has proliferated across science and impacted domains such as biology, chemistry, economics, neuroscience, physics, and psychology. In nearly all scientific domains, new technology has allowed for more data to be collected leading to high-dimensional data. With increasingly complex data, the parameters of the machine learning algorithms exponentially increase leading to issues in interpretability. Solutions to this issue requires either careful feature engineering, feature selection, regularization or some combination of them. In this paper, we focus on feature engineering by way of dimension reduction.

Citation: Sean H. Merritt and Alexander P. Christensen An Experimental Study of Dimension Reduction Methods on Machine Learning Algorithms with Applications to Psychometrics. Advances in Artificial Intelligence and Machine Learning. 2022;3(1):49.

The goal of dimension reduction within machine learning is to reduce the number of variables to a refined set of variables that retain the maximum variance explainable in the whole set that then maximizes prediction. The standard method in machine learning has been to apply Principal Component Analysis (PCA). PCA attempts to find a linear combination of dimensions that are uncorrelated (or orthogonal) and adequately explain the majority of variance between all variables in the dataset. The utility of PCA in machine learning contexts is clear: variables are embedded in a reduced dimension space that maximizes their distinct variance from other dimensions. Given the congruence between the goals of dimension reduction within machine learning and the function of PCA, it's not surprising that the method has become the go-to choice for machine learning researchers.

Should PCA be the de facto dimension reduction method? Previous work examining the effects of different dimension reduction techniques within machine learning algorithms is sparse. Reddy and colleagues [1] tested PCA and linear discriminant analysis (LDA) against no dimension reduction on cardiotocography data. They found that PCA performed better than no reduction when the number of features was high. Similar work has found that PCA tends to perform as well as or better than no reduction [2, 3]. These studies, however, have been limited to examining classification tasks only and very specific applications (e.g., cardiotocography, internet of things, bot detection). Whether PCA should be routinely applied to data before using machine learning algorithms is an open question that we aim to address.

Other commonly used dimension reduction techniques include independent component analysis (ICA). ICA is similar to PCA in that it tries to linearly separate variables into dimensions that are statistically independent rather than uncorrelated. This function is the major difference between their goals: PCA seeks to maximize explained variance in each dimension such that dimensions are uncorrelated whereas ICA seeks to identify underlying dimensions that are statistically independent (maximizing variance explained is not an objective). Similar to PCA, there is a strong congruence between the goals of dimension reduction within machine learning and ICA. With statistically independent dimensions, the data are separated into completely unique dimensions. This property ensures that the predicted variance of an outcome is explained uniquely by each dimension. One advantage ICA has over PCA is that it can work well with non-Gaussian data and therefore does not require variables to be normalized. ICA is commonly used in face recognition [4] as well as neuroscience to identify distinct connectivity patterns between regions of the brain [5, 6].

PCA and ICA are perhaps the two most commonly used dimension reduction methods in machine learning. Despite their common usage, few studies have systematically evaluated whether one should be preferred when it comes to classification or regression tasks. Similarly, few studies, to our knowledge, have examined the extent to which dimension reduction improves prediction accuracy relative to no data reduction at all. Beyond PCA and ICA, there are other dimension reduction methods that offer different advantages that could potentially be useful in machine learning frameworks. Supervised methods, such as sufficient dimension reduction techniques [7], are common in literature, but for the purpose of this paper we focus on unsupervised methods from the network psychometrics literature in psychology.

Exploratory graph analysis (EGA) and unique variable analysis (UVA) are methods that have recently emerged in the field of network psychometrics [8]. These techniques build off of graph theory and social network analysis techniques to identify dimensions in multivariate data. EGA is often compared to PCA in simulations that mirror common psychological data structures [9, 10, 11]. UVA,

in contrast, rose out of a need to identify whether variables are redundant (e.g., multicollinearity, locally dependent) with one another and could be reduced to single, unique variables [12]. Given the goal of dimension reduction in machine learning, these two approaches seem potentially useful for reducing high-dimensional data and identifying unique, non-redundant sources of variance (respectively).

In the present study, we compare PCA, ICA, EGA, UVA, and no reduction on 14 different data sets, seven classification tasks and seven regression tasks. The main aims of this paper are to (1) introduce two alternative dimension reduction methods to the machine learning literature, (2) compare these and the other dimension reduction methods against each other as well as no reduction to the data on a variety of data types and tasks, and (3) examine features of data that lead to dimension reduction improving machine learning algorithms prediction over no reduction. The paper is outlined as follows: section two defines and formalizes EGA and UVA, section three explains the data and procedures in detail, section four reports the results, and section five provides our concluding remarks.

2. Psychometric Dimension Reduction

2.1 Exploratory Graph Analysis

Exploratory graph analyses (EGA) begins by representing the relationship among variables with the Gaussian graphical model (GGM) with the graph $G = \{v_i, e_{ij}\}$, where node v_i represents the i^{th} variable and the edge e_{ij} is the partial correlation between variable v_i and v_j. Estimating a GGM in psychology is often done using the EBICglasso [13, 14, 15], which applies the graphical least absolute shrinkage and selection operator (GLASSO) [16, 17] to the inverse covariance matrix and uses the extended Bayesian information criterion (EBIC) [18] to select the model.

To define the GLASSO regularization method, first assume $\mathbf{y}$ is a multivariate normal distribution:

$$\mathbf{y} \sim N(\mathbf{0}, \mathbf{\Sigma}), \tag{1}$$

where $\mathbf{\Sigma}$ is the population variance-covariance matrix. Let $\mathbf{K}$ denote the inverse covariance matrix:

$$\mathbf{K} = \mathbf{\Sigma}^{-1}. \tag{2}$$

$\mathbf{K}$ can be standardized to produce a partial correlation matrix with each element representing the partial correlation between y_i and y_j conditioned on all other variables $(y_i, y_j|\mathbf{y}_{-(i,j)})$ [19]:

$$\text{Cor}(y_i, y_j|\mathbf{y}_{-(i,j)}) = -\frac{\kappa_{ij}}{\sqrt{\kappa_{ii}}\sqrt{\kappa_{jj}}}, \tag{3}$$

where κ_{ij} represents the i^{th} and j^{th} element of $\mathbf{K}$. The GLASSO regularization method aims to estimate the inverse covariance matrix $\mathbf{K}$ by maximizing the penalized log-likelihood, which is defined as [16]:

$$\log\det(\mathbf{K}) - \text{trace}(\mathbf{SK}) - \lambda \sum_{<i,j>} |\kappa_{ij}|, \tag{4}$$

where **S** represents the sample variance-covariance matrix. The λ parameter represents the penalty on the log-likelihood such that larger values (larger penalty) results in a sparser (fewer non-zero values) inverse covariance matrix. Conversely, smaller values (smaller penalty) results in a denser (fewer zero values) inverse covariance matrix. A GLASSO network is represented as a partial correlation matrix using Eq. 3.

Multiple values of λ are commonly used and model selection techniques such as cross-validation [16] are applied to determine the best fitting model. In the psychometric literature, a more common approach has been to apply the extended Bayesian information criterion (EBIC) [18] to select the λ parameter and best fitting model. The EBIC is defined as:

$$\text{EBIC} = -2L + E\log(N) + 4\gamma E\log(P), \tag{5}$$

where L denotes log-likelihood, N the number of observations, E the number of non-zero elements in **K** (edges), and P the number of variables (nodes). Several λ values (e.g., 100) are selected from a expotential set of values between 0 and 1. The default setting of this range is defined by a minimum-maximum ratio typically set to 0.01 [14]. The γ parameter of the EBIC controls how much simpler models (i.e., fewer non-zero edges) are preferred to more complex models (i.e., fewer zero edges). The default setting for this parameter is typically set to 0.50 [15].

After estimating the GGM via the EBICglasso method, EGA estimates the number of dimensions in the network using a community detection algorithm. There are many different community detection algorithms with some of the more commonly applied algorithms being the Walktrap [20] and Louvain [9, 11, 21, 22, 23]. The Walktrap algorithm uses random walks to obtain a transition matrix that specifies how likely one node would be to "step" to another node. On this transition matrix, Ward's hierarchical clustering algorithm [24] is applied to the transition matrix and *modularity* [25] is used to decide the appropriate "cut" or number of clusters should remain.

Modularity is also used as the primary objective function of the Louvain algorithm. Because of its importance for these two algorithms, we define modularity (Q) [26]:

$$d_i = \sum_{i=1}^{p} w_{ij}, \tag{6}$$

$$D = \frac{1}{2}\sum_{i=1}^{p}\sum_{j=1}^{p} w_{ij}, \tag{7}$$

$$Q = \frac{1}{2D}\sum_{i=1}^{p}\sum_{j=1}^{p}\left[w_{ij} - \frac{d_i d_j}{2D}\right]\delta(c_i, c_j), \tag{8}$$

where w_{ij} is the weight (partial correlation) between node i and node j in the network, p is the number of nodes in the network, d_i is the *degree* or sum of the edge weights connected to node i, D is the total sum of all the edge weights in the network (eliminating the double counting of edges in a symmetric network matrix), $\delta(c_i, c_j)$ is the Kronecker delta of the community membership indices (c) for node i and j, respectively.

The Louvain algorithm works by starting with each node in its own community. Each node is then iteratively switched into another community and placed into the community that has the greatest increase in the modularity statistic (if there is no increase, then the node remains in its original community). After the first pass, "latent" nodes for each community are created by summing the edge weights of the nodes belonging to each community. This process then repeats until either modularity cannot be increased further or the resulting community structure is unidimensional (i.e., all nodes belong to a single community). The goal of the Louvain algorithm is to achieve maximum modularity [21, 26].

Communities detected by these algorithms are statistically similar to dimensions in the data (e.g., PCA) [10]. In order to obtain values for each dimension, so-called "network scores" are computed. To obtain network scores, network loadings are first computed. Network loadings are statistically similar to factor and component loadings [27].

Network loadings are computed by taking the standardized node strength (sum of each node's connections; Eq. 6) within and between each dimension. Network loadings are calculated following Christensen and Golino [27]:

$$L_{if} = \sum_{j \in f}^{F} |w_{ij}|, \tag{9}$$

where F is the number of communities defined by a community detection algorithm, L_{if} represents the loading of the node i on community f and $j \in f$ are all nodes j determined to be part of community f (as determined by the community detection algorithm). This measure is standardized by:

$$\aleph_{L_{if}} = \frac{L_{if}}{\sqrt{\sum_{i=1}^{p} L_{if}}}. \tag{10}$$

These standardized network loadings are represented in an $i \times f$ matrix, $\aleph$. The observed data, X, are tranformed into network scores, $\hat{\theta}$, following Golino et al. [28]:

$$V_f = \frac{\aleph_f}{\sqrt{\frac{\sum_{i=1}^{i \in f} (X_i - \bar{X}_i)^2}{n-1}}} \tag{11}$$

and

$$\hat{\theta_f} = \sum_{f=1}^{F} X_{i \in f} \left(\frac{V_{i \in f}}{\sum_{f}^{F} V_{i \in f}} \right) \tag{12}$$

where X_i is the observed values for variable i, $\bar{X}_i$ is the mean of variable i, n is the number of observations, V_f is the standardized network loadings of dimension f ($\aleph_f$) divided by the standard deviation of the variables with non-zero loadings in dimension f, and $\hat{\theta_f}$ are the network scores that are computed by summing the product of each variable X that has a non-zero loading in dimension f and its corresponding relative loading weight.

2.2 Unique Variable Analysis

Another network psychometrics approach that could be valuable in the context of reducing the number of variables used in making predictions with machine learning algorithms is called Unique Variable Analysis (UVA) [12]. The main goal of UVA is to reduce the dataset to a set of unique variables. Rather than the reduction attempting to reduce to a minimal set of variables (like EGA, ICA, and PCA), UVA does not reduce the number of variables unless two or more variables have substantial shared variance.

UVA was developed to solve issues of *local dependence* in traditional psychometrics. Local dependence is defined as two or more variables that possess potentially redundant information [29]. In the context of a PCA model, local dependence would be reflected in substantial correlation(s) between two (or more) variables' residuals after extracting the components [30]. For machine learning, UVA's objective is to maximize the unique variance provided by each feature while minimizing the redundant variance between features.

Like EGA, UVA starts by estimating a Gaussian graphical model using the EBICglasso. With this network, a measure called weighted topological overlap is applied [31]. Weighted topological overlap (ω) is defined as the similarity between a pair of nodes' connections [32, 31]:

$$\omega_{ij} = \frac{\sum_{u=1}^{p} w_{iu} w_{uj} + w_{ij}}{\min\{d_i, d_j\} + 1 - w_{ij}}$$

where u represents a connection that both node i and j have with some third node u.

In simulation studies, UVA is accurate at detecting when variables are statistically redundant (locally dependent) in data structures common in psychology [12]. Based on simulation evidence, a threshold of 0.25 offers an optimal balance between false positives and overall accuracy. Using this threshold, UVA combines variables that are greater than or equal to this threshold.

There are many ways to combine variables (including removing all but one of the locally dependent variables) but the simplest is to sum (or average) them. UVA continues to iteratively re-assess whether any local dependence remains, combining variables along the way. Once no local dependence remains (i.e., all weighted topological overlap values are less than 0.25), then the process stops. The result can run the spectrum of data reduction from no reduction (i.e., the original dataset if no local dependence is identified) to dimension reductions equivalent to PCA, ICA, and EGA if local dependence between variable sets correspond to the dimensions identified by these methods. In sum, UVA offers a flexible middle ground between no data reduction and complete data reduction (i.e., reduction equivalent to dimension reduction methods).

Relative to PCA and ICA, EGA and UVA are data driven. Among applied practitioners and data scientists, this quality represents a substantive advantage and reduces the researcher degrees of freedom. UVA has the additional advantage of acting as a middle ground between complete dimension reduction and no data reduction. Since UVA finds variables that are statistically redundant, not all data is reduced and therefore may preserve some information that would be aggregated in other dimension reduction methods.

3. METHODS

3.1 Data

To evaluate the effectiveness of the different dimension reduction methods, we trained them on 14 different data sets, seven for regression and seven for classification. We chose to limit the data sets by those that were: tabular data, more than 10 attributes (Mean = 125.8462, Min = 14, Max = 785), and had more than 100 instances (Mean = 14752.08, Min = 120, Max = 70000). Additionally, we sampled data from a variety of domains including business, social sciences, physics, and life sciences. For regression these included: blog feedback [33], communities and crime [34], Facebook metrics [35], online news [36], Parkinson's telemonitoring [37], superconductivity [38], and skillcraft data [39]. Classification data included: breast cancer diagnosis, divorce [40], heart disease [41], Modified National Institute of Standards and Technology (MNIST) [42], musical emotion [43], sport articles objectivity [44], and wine data. All of these data, with the exception of the MNIST can be found on the UCI machine learning repository [45]. We accessed a tabular version of the MNIST data via Kaggle.

3.2 Procedure

We started by preproccessing all of our data to remove any missing observations and categorical data (except for the target variables in the classification tasks) as matrices. All data were tabular and were reduced using PCA, ICA, EGA, and UVA. For the non-reduced data, we used the preproccessed data for fair comparison. We used the R statistical software (version 4.13) [46] with the {EGAnet} (version 1.2.0) [47] and {ica} (version 1.0.3) [48] packages. Number of components for PCA and ICA were determined by examining variance explained via a scree plot.

Next, we trained and tuned the hyperparameters of the machine learning models using 75% of each data set with 3-fold cross validation grid search. We used the mean squared error (RMSE) and accuracy (ACC) for the refitting scores for regression and classification, respectively. Then we tested the data on the other 25% of the data using the best parameters. Finally, to compare the methods we used 5-fold cross validation on the full data set with the best selected parameters. We trained on least absolute shrinkage operator (LASSO) and regularized logistic (Logit) for regression and classification, respectively. All data was formatted as tabular {numpy} arrays to be used as input for all machine learning models. For both tasks, we used random forests classifiers (RFC) and random forest regressions (RFR), support vector machine (SVM), and extreme gradient boosted trees (XGB). All machine learning models were done in Python with the {sklearn} (version 1.1.2) [49] and {XGBoost} (version 1.4.2) [50] modules. We compared models with root mean squared error (RMSE) and accuracy (ACC). This process is shown below in FIGURE 1.

All R and python scripts and data used in the analyses are available on the GitHub repository.

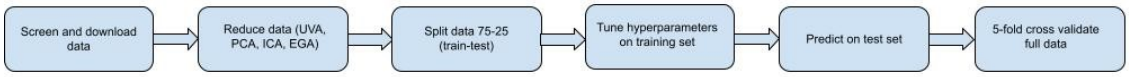

Figure 1: Experimental processes

4. MACHINE LEARNING RESULTS

Using analysis of variance (ANOVA), we compared between each method's performance in regression and classification accuracy. These ANOVAs were followed up using Bonferoni pairwise comparison to determine differences between specific methods. We report the F-statistic (F), p-values (p), and the variance explained (η^2) for the overall ANOVA comparisons. Additionally, we report the p-values (p) and Cohen's D (d) for the pairwise comparisons.

4.1 Regression

There were moderate significant differences across reduction methods, $F(4, 686) = 8.20, p < .001, \eta_p^2 = 0.05$. EGA had significantly higher RMSE than ICA across data ($p < .001, d = 0.53$) but did not differ from PCA ($p = 1.00, d = 0.04$), UVA ($p = 1.00, d = 0.05$), or no reduction ($p = 0.59, d = 0.17$). ICA had significantly lower RMSE than PCA ($p < .001, d = 0.56$), UVA ($p < .001, d = 0.57$, and no reduction ($p = 0.03, d = 0.35$. PCA did not significantly differ from UVA ($p = 1.00, d = 0.01$) and no reduction ($p = 0.38, d = 0.21$). UVA did not significantly differ from no reduction ($p = 0.35, d = 0.22$)

On a more granular level, we examine how each method compared on each data set. Results shown in FIGURE 2. There no significant differences in the blog data, $F(4, 92) = 0.34, p = .85, \eta_p^2 = 0.01$ or news data, $F(4, 92) = 0.00, p = 1.00, \eta_p^2 = 0.00$. The best method and algorithm combination for the blog data was UVA with LASSO (RMSE = 38.313) and for the new data was no data reduction with LASSO ($\bar{RMSE} = 11023.107$).

There were significant differences in the crime data, $F(4, 92) = 10.01, p < .001, \eta_p^2 = 0.30$: EGA had lower RMSE than ICA ($p = 0.04, d = 0.90$) and PCA ($p = 0.001, d = 1.26$), ICA had higher RMSE than no reduction ($p < .001, d = 1.38$), PCA had higher RMSE than no reduction ($p < .001, d = 1.74$) and UVA ($p = 0.003, d = 1.17$). The best method and algorithm combination for the crime data was no data reduction with random forest ($\bar{RMSE} = 0.138$).

There were significant differences in the Facebook data, $F(4, 92) = 13.32, p < .001, \eta_p^2 = 0.37$: ICA had lower RMSE than EGA ($p < .001, d = 1.77$), PCA ($p < .001, d = 1.90$), UVA ($p < .001, d = 1.93$), and no reduction ($p = 0.003, d = 1.19$). The best method and algorithm combination for the Facebook data was ICA with random forest ($\bar{RMSE} = 2.151$).

There were significant differences in the Parkinson's data, $F(4, 92) = 24.63, p < .001, \eta_p^2 = 0.52$: EGA had significantly higher RMSE than ICA ($p < .001, d = 2.03$), UVA ($p < .001, d = 1.85$), and no reduction ($p < .001, d = 2.22$), PCA similarly had significantly higher RMSE than ICA ($p < .001, d = 1.99$), UVA ($p < .001, d = 1.81$), and no reduction ($p < .001, d = 2.18$). The best method and algorithm combination for the Parkinson's data was UVA with LASSO ($\bar{RMSE} = 3.680$).

There were significant differences in the superconductor data, $F(4, 92) = 5.70, p < .001, \eta_p^2 = 0.20$: ICA had significantly higher RMSE than EGA ($p = 0.01, d = 2.03$), PCA ($p = 0.03, d = 0.93$), UVA ($p = .002, d = 1.21$), and no reduction ($p < .001, d = 1.36$). The best method and algorithm combination for the superconductor data was no data reduction with random forest ($\bar{RMSE} = 12.680$).

There were significant differences in the skillcraft data, $F(4, 92) = 22.03, p < .001, \eta_p^2 = 0.49$: UVA had significantly higher RMSE than EGA ($p < 0.01, d = 1.81$), ICA ($p < .001, d = 2.00$), PCA ($p < .001, d = 2.40$), and no reduction ($p < .001, d = 2.68$). The best method and algorithm combination for the skillcraft data was no data reduction with random forest ($\bar{RMSE} = 0.994$).

4.2 Classification

There were small-to-moderate significant differences across reduction methods, $F(4, 686) = 5.48, p < .001, \eta_p^2 = 0.03$. EGA had lower accuracy than ICA ($p = 0.02, d = 0.38$) and PCA ($p < .001, d = 0.48$) but did not differ from UVA ($p = 0.68, d = 0.16$) and no reduction ($p = 0.85, d = 0.12$) across data. ICA did not differ in accuracy from PCA ($p = 0.89, d = 0.11$), UVA ($p = 0.36, d = 0.22$), and no reduction ($p = 0.20, d = 0.26$). PCA was more accurate than UVA ($p = 0.05, d = 0.33$) and no reduction ($p = 0.02, d = 0.37$). UVA did not differ from no reduction ($p = 1.00, d = 0.04$).

On a more granular level, we examine how each method compared on each data set. Results shown in FIGURE 3. There no significant differences in the cancer data, $F(4, 92) = 0.23, p = .92, \eta_p^2 = 0.01$. The best method and algorithm combination for the cancer data was EGA and UVA with logit ($\bar{ACC} = 0.970$).

There were significant differences in the divorce data, $F(4, 92) = 3.24, p = .02, \eta_p^2 = 0.12$: ICA was more accurate than no reduction ($p = 0.02, d = 0.97$). Several method and algorithm combinations had perfect accuracy ($\bar{ACC} = 1.000$) for the divorce data: logit with EGA, ICA, PCA, and UVA; random forest with ICA; all methods with SVM; XGB with EGA, ICA, and PCA.

There were significant differences in the heart data, $F(4, 92) = 9.38, p < .001, \eta_p^2 = 0.29$: EGA ($p = 0.002, d = 1.21$), ICA ($p < .001, d = 1.31$), and PCA ($p < 0.001, d = 1.62$), were more accurate than no reduction. ICA ($p = 0.03, d = 0.94$) and PCA ($p = 0.001, d = 1.26$) were more accurate than UVA. The best method and algorithm combination for the heart data was PCA and UVA with random forest ($\bar{ACC} = 0.993$).

There were significant differences in the MNIST data, $F(4, 92) = 95.94, p < .001, \eta_p^2 = 0.81$: EGA had lower accuracy than ICA ($p < .001, d = 4.34$), PCA ($p < .001, d = 4.18$), UVA ($p < .001, d = 5.27$), and no reduction ($p < .001, d = 5.26$). UVA had higher accuracy than ICA ($p = 0.03, d = 0.94$) and PCA ($p = 0.007, d = 1.10$). Similarly, no reduction had higher accuracy

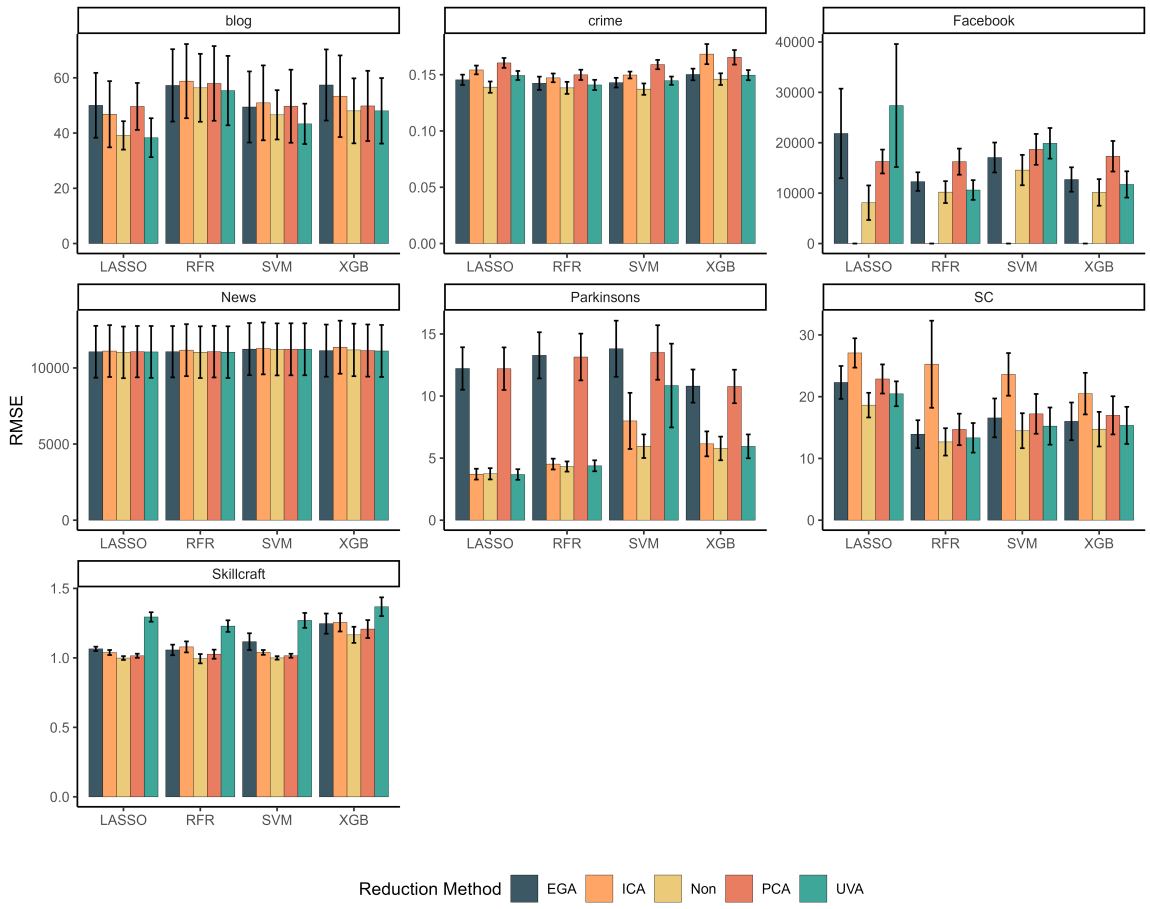

Figure 2: Regression Results. Lower values means better performance.

than ICA ($p = 0.03, d = 0.92$) and PCA ($p = 0.007, d = 1.09$). The best method and algorithm combination for the MNIST data was ICA with SVM ($A\bar{C}C = 0.982$).

There were significant differences in the music data, $F(4, 92) = 11.09, p < .001, \eta_p^2 = 0.33$: EGA ($p = 0.002, d = 1.23$), ICA ($p = 0.006, d = 1.12$), and PCA ($p < .001, d = 1.49$) had higher accuracy than UVA. Similarly, EGA ($p < .001, d = 1.34$), ICA ($p = 0.002, d = 1.23$), and PCA ($p < .001, d = 1.60$) had higher accuracy than no reduction. The best method and algorithm combination for the music data was EGA with XGB ($A\bar{C}C = 1.00$).

There were significant differences in the sports data, $F(4, 92) = 8.51, p < .001, \eta_p^2 = 0.27$: EGA ($p = 0.01, d = 1.04$), ICA ($p = 0.01, d = 1.04$), and PCA ($p = .002, d = 1.23$) had higher accuracy than UVA. Similarly, EGA ($p = 0.003, d = 1.19$), ICA ($p = 0.003, d = 1.19$), and PCA ($p < .001, d = 1.38$) had higher accuracy than no reduction. Several method and algorithm combinations had perfect accuracy ($A\bar{C}C = 1.000$) for the sports data: random forest with EGA, ICA, and PCA as well as all methods with XGB.

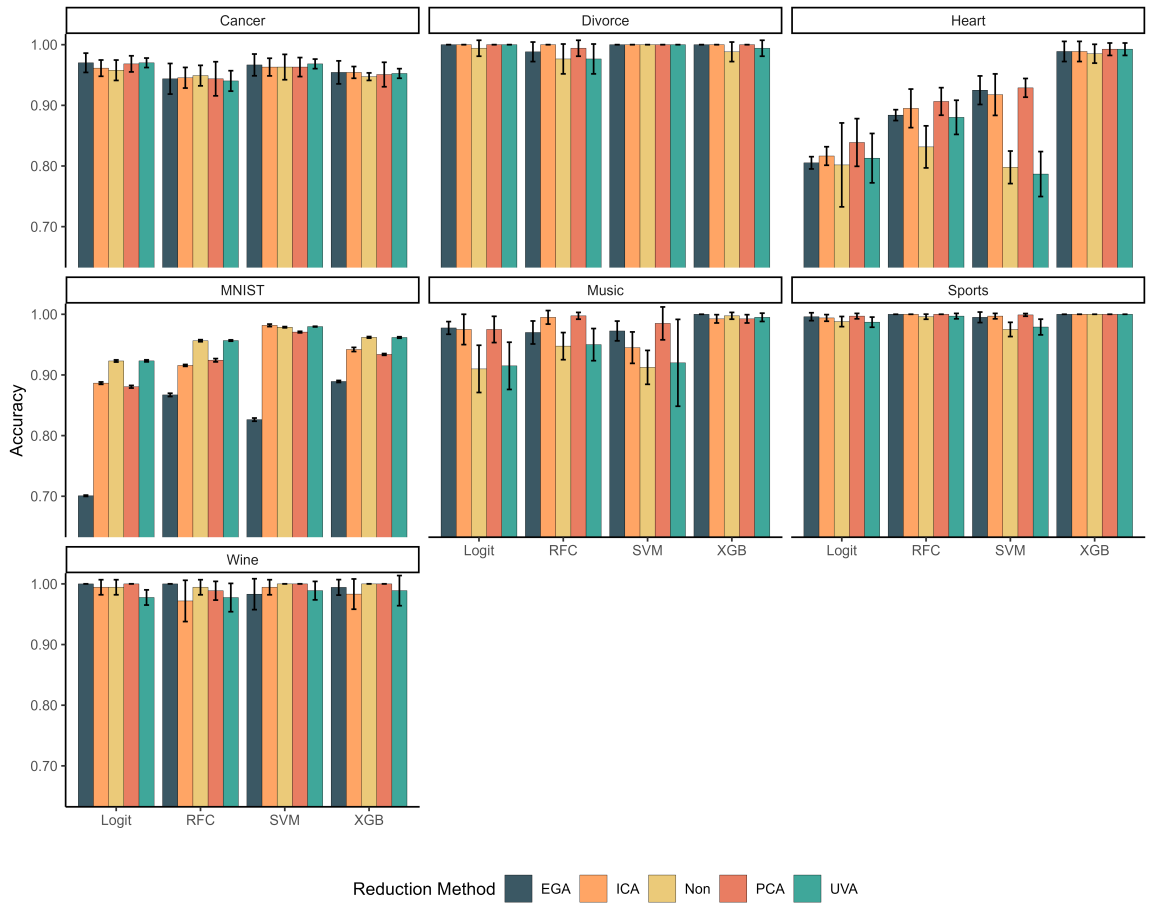

Figure 3: Classification Results. Error bars are standard errors. *y*-axis begins at 0.70.

There were significant differences in the wine data, $F(4, 92) = 3.42, p = .01, \eta_p^2 = 0.13$: PCA ($p = 0.05, d = 0.88$) and no reduction ($p = 0.05, d = 0.88$) had higher accuracy than UVA. Several method and algorithm combinations had perfect accuracy ($\bar{ACC} = 1.000$) for the wine data: logit with EGA and PCA; random forest with EGA; SVM with PCA and no reduction; XGB with PCA and no reduction.

4.3 Comparing Attributes

To better understand when certain reduction methods perform best, we regressed data attributes interacting with each reduction method on the accuracy and RMSE of the data. We standardized the RMSE so as to be comparable across data sets. We tested the number of attributes, number of observations, and the mean kurtosis (kurt) using multiple regression. We report the estimated parameters (b) and the p-values (p). We used no reduction as the reference group compared to simple coded dummy variables corresponding to each reduction method (Table 1).

In classification, we find that EGA performed better than no reduction when the number of attributes ($b = 0.001, p < .001$), the sample size($b = 0.000001, p < .001$), and the average kurtosis ($b = -0.001, p < .001$) increased. ICA was found to perform worse with increase in attributes ($b = -0.001, p < .001$), sample size ($b = -0.00001, p < .001$), and mean kurtosis ($b = -0.00002, p < .001$). UVA and PCA was no different than no reduction on number of attributes, sample size, and kurtosis.

We were not able to replicate these results in the regression data. ICA performed better than no reduction when number of attributes classification ($b = 0.008, p < .01$) and sample size ($b = 0.00001, p < .05$) increased, but not kurtosis ($b = -0.0001, p > .05$). PCA performed worse with increase in attributes ($b = -0.019, p < .01$) and sample size ($b = -0.00002, p < .001$), but not kurtosis ($b = -0.001, p > .05$). UVA performed better with an increase in attributes ($b = 0.005, p < .05$) and EGA performed worse with an increase in kurtosis ($b = -0.0002, p < .05$).

We believe that the results for the classification tasks were skewed by the MNIST data (much larger sample and number attributes and no reduction performed was best). We re-ran these regressions without the MNIST and found that there was no difference between reduction methods and no reduction methods with increasing attributes, sample size, or kurtosis.

5. CONCLUSION

Feature engineering is one of the first steps toward maximizing prediction in machine learning algorithms. Our work examined the effects of dimension reduction on data features using two techniques common to the machine learning literature, PCA and ICA, and two techniques from the network psychometrics literature, EGA and UVA. Overall, we find EGA and UVA perform just as well as PCA, ICA, and no reduction. The predictive performance of each of these methods, however, varied greatly depending on the data. EGA, PCA, and ICA tended to perform similarly while UVA and no reduction tended to perform similarly. As a general trend, we found the best method and algorithm pairs tended to be EGA, ICA, and PCA for the classification tasks, and UVA and no reduction for the regression tasks.

Beyond task type, we examined this variability of performance in the attributes of the data. We failed to replicate the results from Reddy and colleagues [1] that found PCA performed better than no reduction when there is more attributes in the data. However, we did find evidence that EGA performed better than no reduction when the number attributes, sample size, and kurtosis increased on classification tasks. Given that we did not find this to be the case in the regression tasks or replicable without the MNIST, it is not likely that these effects were much more than chance. While our general trend found that dimension reduction may be more effective for classification tasks than regression tasks, more research needs to be done. Future research should also examine how different dimension reduction affects results of textual, visual, and audio data.

One surprising result was that ICA did not perform better with greater kurtosis in the data. Given that ICA was designed for non-Gaussian, we might expect to it to perform well on this type of data. On the other, hand we would expect EGA and PCA to perform worse given that both are designed with Gaussian assumptions. We find that EGA performed better than no reduction method with

Table 1: Attribute Results

	Dependent variable:					
	Accuracy			RMSE		
	(1)	(2)	(3)	(4)	(5)	(6)
EGA	−0.018***	−0.015***	−0.015***	0.502***	0.397***	0.392***
	(0.005)	(0.005)	(0.005)	(0.147)	(0.103)	(0.089)
ICA	0.013***	0.007	0.007	−0.518***	−0.307***	−0.240***
	(0.005)	(0.005)	(0.005)	(0.147)	(0.103)	(0.089)
PCA	0.006	0.007	0.007	0.959***	0.383***	0.183**
	(0.005)	(0.005)	(0.005)	(0.147)	(0.103)	(0.089)
UVA	0.011**	0.011**	0.011**	−0.360**	−0.224**	−0.183**
	(0.005)	(0.005)	(0.005)	(0.147)	(0.103)	(0.089)
Attributes	−0.0001***			0.000		
	(0.00001)			(0.001)		
N		−0.00000***			−0.000	
		(0.00000)			(0.00000)	
mean_kurt			−0.00001***			−0.000
			(0.00000)			(0.0001)
Algorithm1	−0.023***	−0.023***	−0.023***	0.002	0.002	0.002
	(0.004)	(0.004)	(0.004)	(0.067)	(0.068)	(0.069)
Algorithm2	−0.002	−0.002	−0.002	0.107	0.107	0.107
	(0.004)	(0.004)	(0.004)	(0.067)	(0.068)	(0.069)
Algorithm3	−0.0001	−0.0001	−0.0001	−0.030	−0.030	−0.030
	(0.004)	(0.004)	(0.004)	(0.067)	(0.068)	(0.069)
EGA:Attributes	0.0001***			−0.004		
	(0.00002)			(0.003)		
ICA:Attributes	−0.0001***			0.008**		
	(0.00002)			(0.003)		
PCA:Attributes	0.00001			−0.019***		
	(0.00002)			(0.003)		
UVA:Attributes	−0.00000			0.005*		
	(0.00002)			(0.003)		
EGA:N		0.00000***			−0.00001	
		(0.00000)			(0.00001)	
ICA:N		−0.00000***			0.00001*	
		(0.00000)			(0.00001)	
PCA:N		0.00000			−0.00002***	
		(0.00000)			(0.00001)	
UVA:N		−0.00000			0.00001	
		(0.00000)			(0.00001)	
EGA:Mean kurt			0.00001***			−0.0002*
			(0.00000)			(0.0001)
ICA:Mean kurt			−0.00002***			0.0001
			(0.00000)			(0.0001)

Continued on next page

Table 1: Continued

	Dependent variable:					
	Accuracy			RMSE		
	(1)	(2)	(3)	(4)	(5)	(6)
PCA:Mean kurt			0.00000			−0.0001
			(0.00000)			(0.0001)
UVA:Mean kurt			−0.00000			0.0001
			(0.00000)			(0.0001)
Constant	0.963***	0.961***	0.961***	−0.000	0.000	−0.000
	(0.002)	(0.002)	(0.002)	(0.073)	(0.051)	(0.045)
Observations	600	600	600	600	600	600
R^2	0.288	0.292	0.293	0.125	0.073	0.054
Adjusted R^2	0.273	0.278	0.278	0.108	0.054	0.034
Residual Std. Error (df = 587)	0.051	0.051	0.051	0.941	0.969	0.979
F Statistic (df = 12; 587)	19.791***	20.193***	20.236***	7.017***	3.855***	2.772***

Note: *p<0.1; **p<0.05; ***p<0.01

an increasing kurtosis in the data. One caveat to this might be that we used kurtosis instead of negentropy where we might find ICA to perform better when negentropy is higher.

Broadly, dimension reduction appears to perform better than no reduction on classification tasks. One view might be that with simplified data structures there is less noise stemming from unique variance across different data features that make precise classification difficult. Conversely, this same unique variance may improve performance on regression tasks. Indeed, regression performance in the field of personality psychology has seen consistent and robust effects of individual variables, termed personality nuances, outperforming more global traits that are identified by dimension reduction methods [51, 52, 53].

Our results demonstrated that EGA and UVA are robust methods that can be applied in machine learning. They both provide the advantage of reducing researcher degrees of freedom by estimating the number of dimensions and assigning features to those dimensions automatically. EGA is an additional dimension reduction tool that can be added the machine learning practitioner's toolbox while UVA offers the reduction of features if there are features that may be redundant with one another (e.g., multicollineary, locally dependent). An added benefit of modeling variables as a network is that graph theory measures can be applied to multivariate representations of data features that could provide new features that are extracted from the relationships between them [54]. Whether researchers should reduce the number of features of the data using dimension reduction methods is specific to each dataset. We provide evidence that classification tasks can be improved with dimension reduction methods while regression tasks are less affected. The combination of the task type, method and algorithm combination, and attributes of the data all contribute to performance. Untangling these components and their effects on performance remains an important direction for machine learning.

6. ACKNOWLEDGEMENTS

No funding was received to complete this project.

References

[1] Reddy GT, Reddy MPK, Lakshmanna K, Kaluri R, Rajput DS, et al. Analysis of Dimensionality Reduction Techniques on Big Data. IEEE Access. 2020;8:54776-54788.

[2] Bahşi H, Nõmm S, La Torre FB. Dimensionality Reduction for Machine Learning Based IoT Botnet Detection. In: 15th International Conference on Control, Automation, Robotics and Vision (ICARCV). IEEE Publications; 2018:1857-1862.

[3] Vizárraga J, Casas R, Marco Á, Buldain JD. Dimensionality Reduction for Smart IoT Sensors. Electronics. 2020;9:2035.

[4] Bartlett MS, Movellan JR, Sejnowski TJ. Face Recognition by Independent Component Analysis. IEEE Trans Neural Netw. 2002;13:1450-1464.

[5] Calhoun VD, Adali T. Unmixing Fmri With Independent Component Analysis. IEEE Eng Med Biol Mag. 2006;25:79-90.

[6] McKeown MJ, Sejnowski TJ. Independent Component Analysis of Fmri Data: Examining the Assumptions. Hum Brain Mapp. 1998;6:368-372.

[7] Li B. Sufficient Dimension Reduction: Methods and Applications With R. Chapman and Hall. New York: CRC Press. 2018.

[8] Epskamp S, Maris G, Waldrop LJ, Borsboom D. Network Psychometrics. In: Irwing P, Hughes D, Booth T, editors. The Wiley Handbook of Psychometric Testing, 2 volume set: A multidisciplinary reference on survey, scale and test development. New York: Wiley; 2018.

[9] https://psyarxiv.com/hz89e/

[10] Golino HF, Epskamp S. Exploratory Graph Analysis: A New Approach for Estimating the Number of Dimensions in Psychological Research. PLOS ONE. 2017;12:e0174035.

[11] Golino H, Shi Dingjing, Christensen AP, Garrido LE, Nieto MD, et al. Investigating the Performance of Exploratory Graph Analysis and Traditional Techniques to Identify the Number of Latent Factors: A Simulation and Tutorial. Psychol Methods. 2020;25:292-320.

[12] https://psyarxiv.com/4kra2/

[13] Epskamp S, Cramer AOJ, Waldorp LJ, Schmittmann VD, Borsboom D. Qgraph: Network Visualizations of Relationships in Psychometric Data. J Stat Softw. 2012;48:1-18.

[14] Epskamp S, Fried EI. A Tutorial on Regularized Partial Correlation Networks. Psychol Methods. 2018;23:617-634.

[15] Foygel R, Drton M. Extended Bayesian Information Criteria for Gaussian Graphical Models. Adv Neural Inf Process Syst. 2010:604-612.

[16] Friedman J, Hastie T, Tibshirani R. Sparse Inverse Covariance Estimation With the Graphical Lasso. Biostatistics. 2008;9:432-441.

[17] Jhttps://rdrr.io/cran/glasso/

[18] Chen J, Chen Z. Extended Bayesian Information Criteria for Model Selection With Large Model Spaces. Biometrika. 2008;95:759-771.

[19] Lauritzen SL. Graphical Models. Oxford, UK: Clarendon Press; 1996.

[20] Pons P, Latapy M. Computing Communities in Large Networks Using Random Walks. J Graph Algor Appl. 2006;10:191-218.

[21] Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E. Fast Unfolding of Communities in Large Networks. J Stat Mech Theor Exp. 2008.

[22] Gates KM, Henry T, Steinley D, Fair DA. A Monte Carlo Evaluation of Weighted Community Detection Algorithms. Front Neuroinform. 2016;10:45.

[23] Yang Z, Algesheimer R, Tessone CJ. A Comparative Analysis of Community Detection Algorithms on Artificial Networks. Sci Rep. 2016;6:30750.

[24] Ward JH. Hierarchical Clustering to Optimise an Objective Function. J Am Stat Assoc. 1963;58:238-244.

[25] Newman MEJ. Modularity and Community Structure in Networks. Proc Natl Acad Sci U S A. 2006;103:8577-8582.

[26] Brusco MJ, Steinley D, Watts AL. On Maximization of the Modularity Index in Network Psychometrics. Behav Res Methods. 2022:1-17.

[27] Christensen AP, Golino H. On the Equivalency of Factor and Network Loadings. Behav Res Methods. 2021;53:1563-1580.

[28] Golino H, Christensen AP, Moulder R, Kim Seohyun, Boker SM. Modeling Latent Topics in Social Media Using Dynamic Exploratory Graph Analysis: The Case of the Right-Wing and Left-Wing Trolls in the 2016-Us Elections. Psychometrika. 2022;87:156-187.

[29] Chen W-H, Thissen D. Local Dependence Indexes for Item Pairs Using Item Response Theory. J Educ Behav Stat. 1997;22:265-289.

[30] Ferrando PJ, Hernandez-Dorado A, Lorenzo-Seva U. Detecting Correlated Residuals in Exploratory Factor Analysis: New Proposals and a Comparison of Procedures. Struct Equ Model Multidiscip J. 2022;29:630-638.

[31] Zhang B, Horvath S. A General Framework for Weighted Gene Co-expression Network Analysis. Stat Appl Genet Mol Biol. 2005;4:17.

[32] Ravasz E, Somera AL, Mongru DA, Oltvai ZN, Barabási AL. Hierarchical Organization of Modularity in Metabolic Networks. Science. 2002;297:1551-1555.

[33] Buza K. Feedback Prediction for Blogs. In: Data Analysis, Machine Learning and Knowledge Discovery. Springer; 2014:145-52.

[34] Redmond M, Baveja A. A Data-Driven Software Tool for Enabling Cooperative Information Sharing Among Police Departments. Eur J Oper Res. 2002;141:660-678.

[35] Moro S, Rita P, Vala B. Predicting Social Media Performance Metrics and Evaluation of the Impact on Brand Building: A Data Mining Approach. J Bus Res. 2016;69: 3341-3351.

[36] Fernandes K, Vinagre P, Cortez P. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. In: Portuguese Conference on Artificial Intelligence. Springer.2015:535-546.

[37] Tsanas A, Little M, McSharry P, Ramig L. Accurate Telemonitoring of Parkinson's Disease Progression by Non-invasive Speech Tests. Nat Prec. 2009.

[38] Hamidieh K. A Data-Driven Statistical Model for Predicting the Critical Temperature of a Superconductor. Comp Mater Sci. 2018;154:346-354.

[39] Thompson JJ, Blair MR, Chen L, Henrey AJ. Video Game Telemetry as a Critical Tool in the Study of Complex Skill Learning. PLOS ONE. 2013;8:e75129.

[40] Yöntem MK, Kemal A, Ilhan T, Kiliçarslan S. Divorce Prediction Using Correlation Based Feature Selection and Artificial Neural Networks. Nevşehir Hacı Bektaş Veli Univ SBE Derg. 2019;9:259-273.

[41] Detrano R, Janosi A, Steinbrunn W, Pfisterer M, Schmid J-J, et al. International Application of a New Probability Algorithm for the Diagnosis of Coronary Artery Disease. Am J Cardiol. 1989;64:304-310.

[42] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-Based Learning Applied to Document Recognition. Proc IEEE. 1998;86:2278-2324.

[43] Bilal Er MB, Aydilek IB. Music Emotion Recognition by Using Chroma Spectrogram and Deep Visual Features. Int J Comp Intell Syst. 2019;12:1622-1634.

[44] Hajj N, Rizk Y, Awad M. A Subjectivity Classification Framework for Sports Articles Using Improved Cortical Algorithms. Neural Comput Appl. 2019;31:8069-8085.

[45] http://archive.ics.uci.edu/ml

[46] R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. 2022.

[47] https://cran.r-project.org/web/packages/EGAnet/EGAnet.pdf

[48] https://cran.r-project.org/web/packages/ica/index.html

[49] Kramer O. Scikit-Learn. In: Machine Learning for Evolution Strategies. Springer; 2016: 45-53.

[50] Brownlee J. XG Boost with Python: Gradient Boosted Trees With XGBoost and Scikit-Learn. Mach Learn Mastery. 2016.

[51] Brownlee J. XG Boost with Python: Gradient Boosted Trees With XGBoost and Scikit-Learn. Mach Learn Mastery. 2016.

[52] https://psyarxiv.com/4q9gv/

[53] Seeboth A, Mõttus R. Successful Explanations Start With Accurate Descriptions: Questionnaire Items as Personality Markers for More Accurate Predictions. Eur J Pers. 2018;32:186-201.

[54] Goh PK, Martel MM, Jones PJ, Bansal PS, Eng AG, et al. Clarifying Relations Between ADHD and Functional Impairment in Adulthood: Utilization of Network and Machine Learning Approaches. Assessment. 2023;30:316-331.