

Comparing linear discriminant analysis and supervised learning algorithms for binary classification - a method comparison study

Ricarda Graf, Marina Zeldovich, Sarah Friedrich

Angaben zur Veröffentlichung / Publication details:

Graf, Ricarda, Marina Zeldovich, and Sarah Friedrich. 2024. "Comparing linear discriminant analysis and supervised learning algorithms for binary classification - a method comparison study." *Biometrical Journal* 66 (1): 2200098.
<https://doi.org/10.1002/bimj.202200098>.

RESEARCH ARTICLE

Comparing linear discriminant analysis and supervised learning algorithms for binary classification—A method comparison study

Ricarda Graf¹  | Marina Zeldovich² | Sarah Friedrich^{1,3} 

¹Department of Mathematics, University of Augsburg, Germany

²Institute of Medical Psychology and Medical Sociology, University Medical Center Göttingen, Göttingen, Germany

³Centre for Advanced Analytics and Predictive Sciences (CAAPS), University of Augsburg, Augsburg, Germany

Correspondence

Ricarda Graf, Department of Mathematics, University of Augsburg, Universitätsstr. 14, Augsburg 86159, Germany.

Email:

ricarda.graf@math.uni-augsburg.de



This article has earned an open data badge “**Reproducible Research**” for making publicly available the code necessary to reproduce the reported results. The results reported in this article could fully be reproduced.

Abstract

In psychology, linear discriminant analysis (LDA) is the method of choice for two-group classification tasks based on questionnaire data. In this study, we present a comparison of LDA with several supervised learning algorithms. In particular, we examine to what extent the predictive performance of LDA relies on the multivariate normality assumption. As nonparametric alternatives, the linear support vector machine (SVM), classification and regression tree (CART), random forest (RF), probabilistic neural network (PNN), and the ensemble k conditional nearest neighbor (Ek CNN) algorithms are applied. Predictive performance is determined using measures of overall performance, discrimination, and calibration, and is compared in two reference data sets as well as in a simulation study. The reference data are Likert-type data, and comprise 5 and 10 predictor variables, respectively. Simulations are based on the reference data and are done for a balanced and an unbalanced scenario in each case. In order to compare the algorithms' performance, data are simulated from multivariate distributions with differing degrees of nonnormality. Results differ depending on the specific performance measure. The main finding is that LDA is always outperformed by RF in the bimodal data with respect to overall performance. Discriminative ability of the RF algorithm is often higher compared to LDA, but its model calibration is usually worse. Still LDA mostly ranges second in cases it is outperformed by another algorithm, or the differences are only marginal. In consequence, we still recommend LDA for this type of application.

KEYWORDS

binary classification, linear discriminant analysis, multivariate normality, simulation study, supervised learning

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial](https://creativecommons.org/licenses/by-nc/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2022 The Authors. *Biometrical Journal* published by Wiley-VCH GmbH.

1 | INTRODUCTION

In psychology and social sciences, linear discriminant analysis (LDA) is a widely applied method for predicting the probability of an individual to be allocated to a specific group (Boedeker & Kearns, 2019; Shayan et al., 2015; Sherry, 2006).

LDA is an extension of Fisher's discriminant analysis (FDA, Fisher, 1936), a multivariate method for finding a linear combination of continuous attributes best separating two classes. While FDA is a descriptive method used to assess the discriminative ability of the variables, LDA is used for class prediction.

LDA is a parametric method requiring the estimation of two parameters, namely, the class means, and the covariance matrix. As such, it is subject to the small sample size problem (Fukunaga, 1990), that is, only applicable in low-dimensional settings, where the number of attributes is smaller than the sample size, since otherwise the covariance matrix may become singular (Chen et al., 2000). LDA furthermore assumes multivariate normality of the data as well as homogeneity of the covariance matrix across the classes. These assumptions are rarely fulfilled by psychological data sets and hard to verify for small sample sizes (Delacre et al., 2017; Rausch & Kelley, 2009).

Nonparametric machine learning algorithms may be preferable when distributional assumptions are not tenable, or the number of features exceeds the number of observations (Malley et al., 2012). These algorithms do not make any assumptions regarding the data but require optimization of one or several hyperparameters. We compared the performance of LDA to multiple nonparametric supervised learning algorithms, which are suitable for binary classification, and for estimation of individual class probabilities. Versions of these algorithms have already previously been applied in psychological research: support vector machine (SVM, Garcia-Chimeno et al., 2015; Liu & Cheng, 2017), classification and regression tree (CART, Hill et al., 2017; Pagan et al., 2005), random forest (RF, Ammerman et al., 2018; Fife & D'Onofrio, 2021; Wallert et al., 2018), k -nearest neighbor (k -NN, Islam et al., 2018; Noh et al., 2012), and probabilistic neural network (PNN, Carson et al., 1999).

Previous research on the application of LDA under nonnormality of the data has shown that it may nevertheless provide stable estimates. This may be due to its good bias–variance trade-off (Hastie et al., 2009): LDA assumes linearly separable classes, and while a linear decision boundary may introduce bias, it may provide better generalizability due to lower variance than a (highly) irregular nonlinear decision boundary. Also, the performance of LDA may only slightly be affected if the measuring range of the nonnormally distributed variables is limited to a narrow interval (Lachenbruch et al., 1973) as in case of Likert-type data.

Despite this awareness, an insecurity about appropriate methods for nonnormal data remains present in the applied sciences, including psychology. Some authors state that inferences about posterior probabilities of group membership inferred using LDA can be quite misleading, especially in case of more severe deviations from normality (McLachlan, 1992; Rausch & Kelley, 2009).

Thus, we conducted a neutral comparison study following recommended guidelines for benchmarking studies by Weber et al. (2019). The paper is organized as follows. In Section 2, we briefly review the supervised classification algorithms applied in our simulation studies and describe the specific approaches we use. In Section 3, we describe the simulation setup and the reference data sets. The results are described in Section 4. Finally, we close with some concluding remarks in Section 5.

2 | SUPERVISED CLASSIFICATION ALGORITHMS

In our simulation study, we focus on the case of binary classification.

The training data set $\mathbf{X} \in \mathbb{R}^{n \times p}$ contains n observations $\mathbf{x} \in \mathbb{R}^p$, where p is the number of variables, also referred to as features, covariates, or dimensionality of the data set. The class label $y \in \{0, 1\}$ of each observation in the training data set is known, and the subset $\mathbf{X}_i \in \mathbb{R}^{n_i \times p}$ contains the n_i observations of group $i \in \{0, 1\}$, where $\sum_{i=0}^1 n_i = n$. The goal of supervised learning algorithms is to create an allocation rule from the training data \mathbf{X} in order to classify new observations with unknown class labels. The performance of the algorithms can be determined from a separate test data set. We will give a brief review of the included algorithms and describe the specific approaches we use.

2.1 | Linear discriminant analysis

We consider LDA as implemented in the MASS package (Venables & Ripley, 2002), and used by researchers in psychology and social sciences. The package uses FDA (Fisher, 1936) for discrimination and its extension, LDA (Rao, 1973), for classification.

FDA finds the linear combination $f(\mathbf{x}) = w_1x_1 + \dots + w_px_p = \mathbf{w}^T\mathbf{x}$, which minimizes the amount of overlap between two classes. The absolute value of the weights $\mathbf{w} \in \mathbb{R}^p$ corresponds to the variables' importance, thus implying an order. According to Fisher, the weight vector \mathbf{w} shall maximize the ratio between the separation of the class means $\mu_i, i \in \{0, 1\}$ and the within-class covariance Σ , that is:

$$\max_{\substack{\mathbf{w} \in \mathbb{R}^p \\ \mathbf{w} \neq 0}} \frac{|\mathbf{w}^T(\mu_0 - \mu_1)|}{(\mathbf{w}^T\Sigma\mathbf{w})^{1/2}}. \quad (1)$$

This is obtained by choosing $\mathbf{w}^* = c \Sigma^{-1}(\mu_0 - \mu_1)$, where $c \neq 0$ is an arbitrary constant (Flury, 1997).

FDA does not depend on multivariate normally distributed data, but the covariance matrices of both groups are assumed to be equal and nonsingular, that is, $\Sigma_0 = \Sigma_1 = \Sigma$ with $|\Sigma| \neq 0$. For $n_0 + n_1 < p + 2$, Σ is always singular, meaning that LDA requires a sufficiently large data set. The unknown parameters are substituted by their sample estimates (Johnson & Wichern, 2007), that is,

$$\hat{\Sigma} = \frac{n_0 - 1}{(n_0 - 1) + (n_1 - 1)} \mathbf{S}_0 + \frac{n_1 - 1}{(n_0 - 1) + (n_1 - 1)} \mathbf{S}_1, \quad (2)$$

where $\mathbf{S}_i = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} (\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)^T$ and $\hat{\mu}_i = \bar{\mathbf{x}}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{x}_{ij}$.

Here, j is the index for a particular observation from class i .

The extension of FDA to LDA (Rao, 1973) assumes that the data of each class follow a p -variate normal distribution, that is, $\mathbf{X}_i \sim \mathcal{N}_p(\mu_i, \Sigma)$ with probability density function

$$f_i(\mathbf{x}) = (2\pi)^{-p/2} |\Sigma|^{-1/2} \exp\{-1/2(\mathbf{x} - \mu_i)^T \Sigma^{-1}(\mathbf{x} - \mu_i)\}. \quad (3)$$

An observation $\mathbf{x} \in \mathbb{R}^p$ is assigned to class i if $\pi_i f_i(\mathbf{x}) \geq \pi_q f_q(\mathbf{x})$ for all $q \neq i$ (Rao, 1973), where π_i is the prior probability of class i . Accordingly, the LDA classification rule is (Johnson & Wichern, 2007):

$$(\mu_0 - \mu_1)^T \Sigma^{-1} \mathbf{x} - \frac{1}{2}(\mu_0 - \mu_1)^T \Sigma^{-1}(\mu_0 + \mu_1) - \log(\pi_1/\pi_0) \begin{cases} < 0 & \text{class 1,} \\ > 0 & \text{class 0.} \end{cases} \quad (4)$$

The posterior probabilities of class membership are computed as (Flury, 1997):

$$\hat{p}(\text{class} = i | \mathbf{x}) = \frac{f_i(\mathbf{x})\pi_i}{\sum_{q=0}^1 f_q(\mathbf{x})\pi_q}. \quad (5)$$

2.2 | Support vector machine

Linear SVM also provides feature weights whose absolute value corresponds to the feature importance (Guyon et al., 2002). In contrast to LDA, which uses the entire training data to find the discriminant function, it only uses a small subset, the support vectors, to find the optimal separating hyperplane. The binary class labels are denoted by $y \in \{-1, +1\}$ in the context of SVM. Based on Mercer's theorem, SVM uses positive-definite kernel functions (Mercer, 1909; Vapnik, 1982) to transform the original data into a potentially higher-dimensional feature space, in which the data become linearly

separable. Using nonlinear kernel functions will result in a nonlinear separating hyperplane in the original space. Variable weights become hard to interpret since they refer to this higher dimension.

The kernel function of the linear SVM is defined as the inner product of two samples \mathbf{x}_{j_1} and \mathbf{x}_{j_2} :

$$f: \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}, \quad f(\mathbf{x}_{j_1}, \mathbf{x}_{j_2}) = \langle \mathbf{x}_{j_1}, \mathbf{x}_{j_2} \rangle = \mathbf{x}_{j_1}^T \mathbf{x}_{j_2}, \quad j_1, j_2 = 1, \dots, n. \quad (6)$$

Here, we apply the linear soft-margin SVM (Cortes & Vapnik, 1995), a modification of the original algorithm (Vapnik, 1982) designed for linearly separable classes. It allows the classes to overlap depending on the value of the regularization parameter C .

SVM uses convex quadratic programming (QP). The linear separating hyperplane can be defined by a weight vector $\mathbf{w} \in \mathbb{R}^p$ and an intercept b (primal form):

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{j=1}^n \xi_j \\ \text{s.t.} \quad & y_j(\mathbf{w}^T \mathbf{x}_j + b) \geq 1 - \xi_j \text{ and } \xi_j \geq 0 \quad \forall j = 1, \dots, n, \end{aligned} \quad (7)$$

where the value of $\frac{1}{\|\mathbf{w}\|}$ gives the width of the margin. The margin is the distance between the hyperplane and the data samples lying closest to it, the support vectors. The regularization parameter C is a tuning parameter. For the linearly separable case, $C = \infty$ (Hastie et al., 2009). The slack variables ξ_j represent the errors (Cortes & Vapnik, 1995).

The corresponding dual form of the QP maximizes the Lagrange multipliers α :

$$\begin{aligned} \max_{\alpha} \quad & \sum_{j=1}^n \alpha_j - \frac{1}{2} \sum_{j_1=1}^n \sum_{j_2=1}^n \alpha_{j_1} \alpha_{j_2} y_{j_1} y_{j_2} f(\mathbf{x}_{j_1}, \mathbf{x}_{j_2}) \\ \text{s.t.} \quad & 0 \leq \alpha_j \leq C \text{ and } \sum_{j=1}^n \alpha_j y_j = 0 \quad \forall j = 1, \dots, n. \end{aligned} \quad (8)$$

The dual form allows to directly identify the support vectors since their Lagrange multipliers α_j are different from zero and its solution can be computed more efficiently.

The variable weights \mathbf{w} of the linear soft-margin SVM can be computed from the optimal solution α^* (Hastie et al., 2004):

$$\mathbf{w}^* = C \sum_{j=1}^n \alpha_j^* y_j \mathbf{x}_j. \quad (9)$$

The posterior probabilities of class membership are computed using the modified approximation algorithm by Platt (Lin et al., 2007; Platt, 2000).

We use the SSVMP algorithm (Sentelle, 2013), a modification of the SVMpath algorithm (Hastie et al., 2004) to find the optimal regularization parameter C . It optimizes the inverse of the regularization parameter, $\lambda = 1/C$, and uses that only the Lagrange multipliers α_j of the support vectors, the samples on the margin, vary. It starts with a high value of λ , such that all samples are located inside the margin (except of the support vectors). It then successively determines the samples leaving the margin based on the values of α_j and updates the linear decision boundary. A sequence of strictly decreasing values λ is obtained. The algorithm terminates when there are no samples left inside the margin or the next value of λ would be zero.

2.3 | Ensemble k conditional nearest neighbor

The k conditional nearest neighbor (k CNN) algorithm by Gweon (2019) is an extension of the k -NN algorithm by Fix and Hodges (1951), where k is the hyperparameter for the number of neighbors that shall be inspected to classify a new sample.

The algorithm computes the distance between a new sample and all samples of the training data. Although its performance depends on the choice of the distance measure (Abu Alfeilat et al., 2019), usually Euclidean distance is used. The parameter k is sometimes suggested to be set equal to the square root of the number of training observations (Lantz, 2013), but the optimal choice of k largely depends on the data. We tuned the parameter k using grid search, that is, comparing the misclassification error for a range of values of k around the suggested value using cross-validation in the training data (Bergstra & Bengio, 2012; Ghawi & Pfeffer, 2019).

Unlike LDA and linear SVM, the k -NN algorithm (and its extension k CNN by Gweon et al., 2019) does not produce coefficients of the attributes, so the impact of the features on classification cannot be examined (Hastie et al., 2009). Unlike the original version, the k CNN algorithm computes the posterior probabilities of class membership. The posterior probabilities are necessary to compute the classification indices we use to compare the performance. They are computed by (Gweon et al., 2019):

$$\hat{p}_k(\text{class} = i|\mathbf{x}) = \frac{\frac{k}{nV_{k|i}}}{\sum_{q=0}^1 \frac{k}{nV_{k|q}}} = \frac{d(\mathbf{x}, \mathbf{x}_{k|i})^{-p}}{\sum_{q=0}^1 d(\mathbf{x}, \mathbf{x}_{k|q})^{-p}} \quad (10)$$

and converge in probability to the true posterior for $n \rightarrow \infty$. Here, $d(\mathbf{x}, \mathbf{x}_{k|i}) = |\mathbf{x} - \mathbf{x}_{k|i}|$ is the Euclidean distance between a new sample (\mathbf{x}) and its k th nearest neighbour of class i ($\mathbf{x}_{k|i}$), $V_{k|i}$ represents the volume of a hypersphere with center \mathbf{x} and radius $d(\mathbf{x}, \mathbf{x}_{k|i})$, and p the dimensionality of the data.

The algorithm assigns a new sample to the class with the highest posterior probability. The class predictions vary considerably depending on the parameter value k . Therefore, we use the ensemble k CNN (Ek CNN) algorithm by Gweon et al. (2019), which estimates the posterior probabilities $\hat{p}_u(\text{class} = i|\mathbf{x})$ for each value $u = 1, \dots, k$ and assigns a new observation to the class with the highest average

$$\arg \max_{i \in \{0,1\}} \hat{p}_k(\text{class} = i|\mathbf{x}) = \arg \max_{i \in \{0,1\}} \frac{1}{k} \sum_{u=1}^k \hat{p}_u(\text{class} = i|\mathbf{x}) \quad (11)$$

to improve predictive performance. We use the R code of the Ek CNN algorithm available on GitHub (Gweon, 2018).

2.4 | Classification and regression tree

The CART algorithm by Breiman et al. (1984) creates binary decision trees using forward selection and recursive binary splitting in order to minimize dissimilarity of the observations in each node with respect to their class labels y (in case of classification). Starting with the entire data set in the root node, the tree can be grown until each terminal node contains only samples belonging to the same class, but usually a prepruning, or a postpruning strategy is specified to achieve better generalizability in new data. Postpruning is more computationally intensive, but usually gives a more reliable model than prepruning (Trabelsi et al., 2007).

We applied postpruning using the minimum-error cost-complexity pruning strategy, which minimizes the cost-complexity measure

$$R_\gamma(T) = R(T) + \gamma|T| \quad (12)$$

regarding all subtrees $T < T_{\gamma=0}$ ($T_{\gamma=0}$ corresponds to the fully grown tree) for a given cost-complexity parameter γ . The symbols $|T|$ and $R(T)$ indicate the number of terminal nodes, and the misclassification rate of subtree T (Hastie et al., 2009).

There is no universally good choice for setting γ (Cichosz, 2015). We use the recommended approach of choosing γ corresponding to the minimum k -fold cross-validated misclassification error, where $k = 10$ is a conventional choice (Cichosz, 2015; Hastie et al., 2009). We consider all values in the interval $[0, 0.025]$ with step size 0.0001 for γ , which includes the default value of 0.01 in the `rpart` package (Therneau et al., 2019) in R. The CART algorithm determines the best splitting value for each variable $1, \dots, p$ and subsequently chooses the best splitting variable by minimizing an impurity measure

for each node m in tree T . We use the Gini impurity index G_m . It takes the value zero for a split resulting in perfect discrimination between the classes and takes the maximum 0.5 in case of highest impurity. It is defined as:

$$G_m = \sum_{i=0}^1 \hat{p}_{mi}(1 - \hat{p}_{mi}), \quad (13)$$

where \hat{p}_{mi} is the (prior) probability of class i .

The posterior probability of class membership is the portion of samples in the terminal node belonging to the respective class. Variable importance is computed by considering the reduction of the Gini impurity index of the splitting variables.

Disadvantages of the CART algorithm are its sensitivity to small changes in the data set, and the tendency of the Gini importance measure to select variables with many potential splitting values as splitting variables, even if they are random noise (Strobl et al., 2006).

2.5 | Random forest

The likely chance of overfitting in the training data when using fully-grown trees can be reduced by combining several arbitrarily complex trees using the random subspace method, that is, by randomly partitioning the feature space used for growing each tree. With an increasing number of trees, the predictive accuracy increases (Ho, 1995). Breiman's RF algorithm (Breiman, 2001) combines Ho's random subspace method with random subsampling of the training data (with replacement) to further reduce the variance of single tree predictions and to further decorrelate the trees. Class predictions are made according to the majority vote of the ensemble of trees. The most important tuning parameters are the number of trees and the number of features among which the split variable in a tree is chosen. We also considered the sample fraction used for growing a single tree, and minimum node size of the terminal nodes, although the algorithm also performs well without extensive hyperparameter tuning (Boehmke & Greenwell, 2019). Hyperparameter tuning is done by using full grid search. In contrast to the CART algorithm, no test data are needed since the out-of-bag data, the sample fraction not used for growing the tree, are used for assessing the model performance. Variable importance in RFs for classification can be evaluated by the Mean Decrease Impurity:

$$\text{Imp}(X_l) = \frac{1}{n_T} \sum_{t \in T: v(s_t)=X_l} p(t) \Delta z(s_t, t). \quad (14)$$

The decrease in an impurity measure $\Delta z(s_t, t)$ for a variable X_l ($l \in \{1, \dots, p\}$) is summed up over all nodes t , for which it is the splitting variable $v(s_t)$, weighted by the proportion of samples in t , that is, $p(t) = \frac{n_t}{n}$ (Breiman, 2001; Louppe et al., 2013). The Gini impurity measure can be used, with the same potential drawbacks as for the CART algorithm. The posterior probability of class membership equals the fraction of trees that vote for the considered class (Olson & Wyner, 2018).

2.6 | Probabilistic neural network

PNNs by Specht (1966) are four-layer neural networks. Unlike neural networks, their estimates are asymptotically Bayes optimal, they do not require extensive training times, and the only parameter that needs to be optimized is the smoothing parameter σ .

The input layer contains the elements of the input vector. In the pattern layer, the input vector is compared to each vector from the training data set \mathbf{X} by computing nonparametric Parzen estimators (Parzen, 1962) of the class-specific probability density functions

$$\hat{f}_i(\mathbf{x}) = \frac{1}{(2\pi)^{p/2} \sigma^p} \frac{1}{n_i} \sum_{j=1}^{n_i} \exp \left\{ -\frac{(\mathbf{x} - \mathbf{x}_{ij})^T (\mathbf{x} - \mathbf{x}_{ij})}{2\sigma^2} \right\}, \quad (15)$$

where n_i indicates the number of samples in class i , and \mathbf{x}_{ij} the j th sample of class i (Specht, 1990). In the summation layer, the average of all $\hat{f}_i(\mathbf{x})$ per class are computed. In the output layer, the vector \mathbf{x} is assigned to class i with the maximum value of $\hat{f}_i(\mathbf{x})$.

Finding a good smoothing bandwidth σ for the Parzen estimator is critical, although the misclassification rate does not change considerably with small changes in σ (Specht, 1990). A value too small results in noisy estimations and a value too high in an oversmoothed probability density estimation. The Parzen estimator is known to not adapt well to the local probability density with increasing data dimensionality. This is already the case for a dimensionality higher than 2 or 3 (Boltz et al., 2007, 2009). The optimal value for σ is determined by cross-validation.

Posterior probabilities of class membership are computed as (Specht, 1990):

$$\hat{p}(\text{class} = i|\mathbf{x}) = \frac{\hat{f}_i(\mathbf{x})\pi_i}{\sum_{q=0}^1 \hat{f}_q(\mathbf{x})\pi_q}. \quad (16)$$

Variable importance (per class) is determined by computing the Sobol indices, which is a global sensitivity analysis for scalar outputs (Sobol, 1993).

3 | METHOD COMPARISON STUDY

In order to compare the above-mentioned methods with respect to their performance on psychological data, we conducted a method comparison study. Our simulations are based on two data sets from psychology which we will describe in the following.

3.1 | Reference data sets

Two reference data sets with differing numbers of variables are used. Both contain Likert-type data. In each case, binary classification is considered ($y \in \{0, 1\}$). Data are simulated for an unbalanced ($n_0 = 50$, $n_1 = 100$) and a balanced scenario ($n_0 = n_1 = 500$) in both cases.

Data set 1 (Zeldovich, 2019) is a self-report questionnaire on the “Big Five” personality traits completed by 1109 persons ($n_0 = 535$ men, $n_1 = 574$ women). It uses a 21-item abbreviated version of the Big Five Inventory (BFI-K) by Rammstedt and John (2005) through which the traits extraversion, agreeableness, openness to experience, conscientiousness, and neuroticism are examined on a five-point Likert scale (1 = disagree a lot, 5 = agree a lot). Answers to questions regarding the same trait are summarized in a score, on which the simulations are based. In this simulation, male participants represent group 0, and female participants represent group 1.

Data set 2 (Ma et al., 2015a, 2015b) contains ratings of 597 neutral-expression photographs of persons from diverse racial backgrounds. The 1087 raters (who also come from diverse racial backgrounds) rated these photographs with respect to 15 characteristics on a seven-point Likert scale (1 = not at all, 7 = extremely). In this simulation, Black models represent group 0 ($n_0 = 197$), and White models represent group 1 ($n_1 = 183$). In order to remove multicollinearity, a threshold of 0.6 is applied to the pairwise Pearson correlation coefficients (Dancey & Reidy, 2007), and as a result, the traits “afraid,” “masculine,” “trustworthy,” “threatening,” and “angry” are dropped from the analysis.

In both data sets, the two groups are not distinctively separated from each other. The Euclidean distance between class means is 0.548 in data set 1, and 0.472 in data set 2. The boxplots in Figure 1 show the variables’ distribution.

3.2 | Data-generating scenarios

The methods’ performance is compared in data sets generated from various multivariate distributions, differing in their degree of deviation from multivariate normality. Data-generating scenarios comprise data simulations from the multivariate normal, multivariate skewnormal, multivariate lognormal, multivariate Gamma distribution, and of correlated ordinal and of correlated bimodal variables, respectively. This may help to assess how performance of LDA is affected by

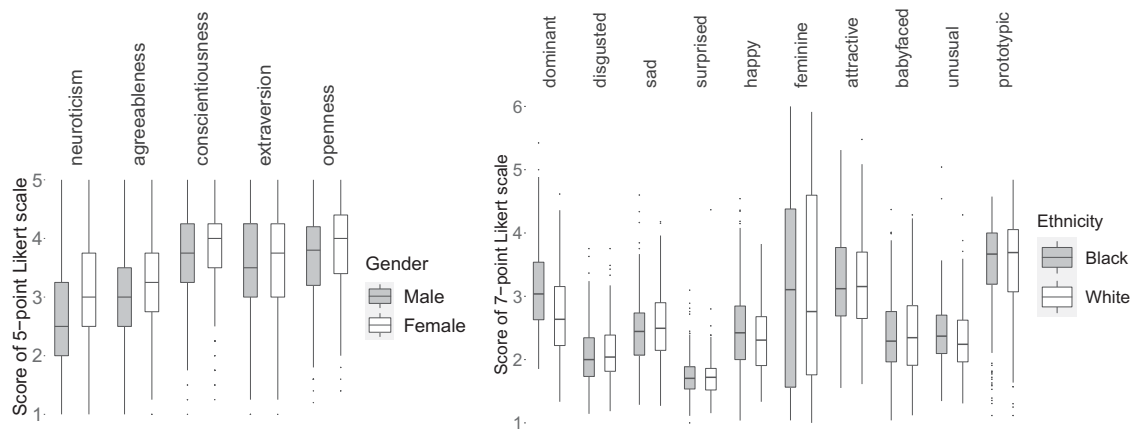


FIGURE 1 Boxplots showing the variables' distribution in reference data sets 1 (left) and 2 (right).

TABLE 1 Parameterizations of the multivariate distributions.

Distribution	Parameterization	
	Class 0	Class 1
Multivariate normal (1)	μ_0, Σ	μ_1, Σ
Multivariate normal (2), increased $d(\mu_0, \mu_1)$	μ_0, Σ	$\mu_1 + 1_p, \Sigma$
Multivariate skewnormal	μ_0, Σ, λ	μ_1, Σ, λ
Multivariate lognormal	μ_0, Σ	μ_1, Σ
Multivariate Gamma	$\mathbf{a}_0, \mathbf{b}_0, \mathbf{R}$	$\mathbf{a}_1, \mathbf{b}_1, \mathbf{R}$
Correlated ordinal variables	$\mathbf{p}_{0,l}^{ord}, \mathbf{R}^{ord}$	$\mathbf{p}_{1,l}^{ord}, \mathbf{R}^{ord}$ ($l = 1, \dots, p$)
Correlated bimodal variables	$(\mu^{bi}, \sigma^{bi}, \nu^{bi}, \tau^{bi})_{0,l}, \mathbf{R}$	$(\mu^{bi}, \sigma^{bi}, \nu^{bi}, \tau^{bi})_{1,l}, \mathbf{R}$ ($l = 1, \dots, p$)

nonnormality. Table 1 shows the parameterizations of the multivariate distributions. As a control, multivariate normally distributed data using a larger distance between the class means are also analyzed. More detailed information about these distributions can be found in the Supplementary Material S.1. Distributional parameters are inferred from the reference data sets.

The parameters $\mu_0, \mu_1 \in \mathbb{R}^p$ symbolize the class mean of class 0 and 1, respectively. The covariance matrix and Pearson correlation matrix are given by $\Sigma \in \mathbb{R}^{p \times p}$ and $\mathbf{R} \in \mathbb{R}^{p \times p}$, respectively. The skewnormal distribution additionally includes the skewness parameter $\lambda \in \mathbb{R}^p$. The skewness of the skewnormal distribution is bounded (Ngunkeng, 2013). Mardia's measure of multivariate skewness (Franceschini & Loperfido, 2019; Mardia, 1970) is used to determine an λ resulting in a high level of skewness. For this, the values $\lambda \in \{0_p, 1_p, \dots, 15_p, 1000_p, 10,000_p, 100,000_p\}$ are tried. The Gamma distribution is defined by a shape parameter $\mathbf{a}_i \in \mathbb{R}^p$ and a scale parameter $\mathbf{b}_i \in \mathbb{R}^p$, obtained from the reference data using the method of moments.

Simulation of correlated ordinal variables is done according to the mean mapping method by Kaiser et al. (2011). It requires the marginal probabilities of each variable ($\mathbf{p}_{0,l}^{ord}$ or $\mathbf{p}_{1,l}^{ord}$, $l = 1, \dots, p$ for group 0 and 1, respectively) and the Kendall correlation matrix of the data rounded to full integers (\mathbf{R}^{ord}) as input parameters. The marginal probabilities must be different from zero, which is satisfied for all variables in data set 1. In data set 2, all variables have at least one empty category after rounding the scores in the original data sets to integer values. A small amount of values (one to four) are randomly added and the same amount of values are subtracted from the two most frequent categories in order to allow simulation of correlated ordinal variables for data set 2. Correlated bimodal variables are obtained by simulation of marginal distributions from bimodal skew-symmetric normal distributions (Hassan & El-Bassiouni, 2016) and subsequent combination via a Gaussian copula using the correlation matrix \mathbf{R} . The parameters of the bimodal skew-symmetric normal distributions (location parameter μ^{bi} , scale parameter σ^{bi} , the second location parameter ν^{bi} , and bimodality parameter τ^{bi}) are not estimated from the data, since they are not noticeably bimodal. Instead, combinations of parameter values are chosen such that the majority of values (97%) lie within the interval range of the original data.

The traits in data set 1 are considered in the following order: extraversion (x_1), agreeableness (x_2), conscientiousness (x_3), neuroticism (x_4), and openness to experience (x_5). The parameter values for simulations based on data set 1 are shown in Table S1 (supplementary material S.2.).

The traits in data set 2 are considered in the order: attractive (x_1), babyfaced (x_2), disgusted (x_3), dominant (x_4), feminine (x_5), happy (x_6), prototypic (x_7), sad (x_8), surprised (x_9), and unusual (x_{10}). The parameter values for simulations based on data set 2 are shown in Table S2 (supplementary material S.2.).

3.3 | Simulation study approach

All simulations and analyses are conducted using the R statistical software (R Core Team, 2021) version 4.1.2 on a Linux system. In order to compare the performance of the classification algorithms, Monte Carlo simulations are performed. Four simulation scenarios are considered—two for each reference data set—by choosing two different sets of sample sizes ($n_0 = 50$, $n_1 = 100$, and $n_0 = n_1 = 500$) common for this type of psychological data. For each simulation scenario, 2000 data sets are simulated using the parameters inferred from the respective reference data set.

We use the implementations of the classification algorithms as available in the following R packages or from the following websites, respectively: MASS (Venables & Ripley, 2002) for LDA, kernlab (Karatzoglou et al., 2004) for the linear soft-margin SVM, rpart (Therneau et al., 2019) for the CART algorithm, ranger (Wright & Ziegler, 2017) for the RF algorithm, the EkCNN algorithm by Gweon (2018), and yap (Liu, 2020) for the PNN.

Data are simulated using the following R packages or approaches, respectively: MASS (Venables & Ripley, 2002) for the multivariate normal distribution, exponential values of the multivariate normally distributed data for the multivariate lognormal distribution, fMultivar (Wuertz et al., 2020) for the multivariate skewnormal distribution, lcmix (Dvorkin, 2019) for the multivariate Gamma distribution, and orddata (Leisch et al., 2010) for the correlated ordinal variables, and gamlssbsn (Hossain et al., 2017) and copula (Hofert et al., 2022) for the correlated bimodal variables.

Data sets are split into training and test data sets in a ratio of 7:3. Data are preprocessed by standardization since SVM and EkCNN rely on the Euclidean distance.

Hyperparameter training for the nonparametric classification algorithms is done as follows. For the SVM, the the simple SVM path (SSVMP) algorithm by Sentelle (2013) available as MATLAB code (Sentelle et al., 2016) is used to determine the optimal regularization parameter C . For this purpose, the code is rewritten in R. For the CART algorithm, we use 10-fold cross-validation to find the optimal cost-complexity parameter value used for postpruning of the fully grown trees. The default value of 0.01 from the function `rpart.control` (Therneau et al., 2019) is included in the search interval. Hastie et al. (2009) and Cichosz (2015) recommend the value $k = 10$ for k -fold cross-validation. For the RF algorithm, we use full grid search of different combinations of values for the number of trees, number of variables considered for splitting the nodes in a tree (m_{try}), the sample fraction used to build a tree, and the minimum size of leaf nodes. We follow the recommendations of Boehmke and Greenwell (2019) with respect to the mentioned default values and the number of considered values. For the EkCNN algorithm, we conduct 10-fold cross-validation to obtain the optimal parameter k and include the recommended default, the square root of the number of training samples (Lantz, 2013), in the search interval. For the PNN algorithm, we use the function `pnn.search_log1` from the R package yap (Liu, 2020) using fivefold cross-validation, since the recommended default value is set to 4 in this function and the small changes in σ do not affect the method's performance (Specht, 1990). It finds the optimal smoothing parameter σ based on cross-entropy.

The algorithms' performance is compared using different performance measures, which can be categorized into overall performance measures, measures of discrimination and measures of calibration (Steyerberg et al., 2010). Measures of overall performance capture both, aspects of discrimination and calibration. They consider the difference between the true class labels and the estimated posterior probabilities (B -, and Q index). Discrimination relates to the algorithm's ability to separate observations of both classes. While sensitivity, specificity, the Youden index, and predictive accuracy only take the accuracy of the class predictions into account, the C index measures the concordance between the algorithm's estimated posterior probabilities and the observed class labels. The C index can be interpreted as the probability that a sample from class 1 will have a higher predicted posterior probability than a sample from class 0, that is, it measures the algorithm's ability to rank observations from high to low probability of belonging to class 1, but it does not measure the accuracy of these predictions unlike measures of calibration. The C index equals the area under the Receiver Operating Characteristic (ROC) curve when applying all possible classification thresholds (Pencina & D'Agostino, 2019). Measures of calibration compare the true and estimated posterior probabilities, where the (unknown) true posterior probabilities are approximated by the respective portion of samples in subsets of the original data (Hosmer–Lemeshow test).

TABLE 2 Mardia measure of multivariate skewness ($b_{1,p}$), value of the corresponding χ^2 test statistic with respective p -value for the reference data sets.

Data set	$b_{1,p}$	Test statistic	p -Value
1	0.944	175	1.9E−20
2	20.305	1286	5.9E−150

Definitions of the C , B , and Q index are given below. The remaining performance measures are described in supplementary material S.3.

For conducting the Hosmer–Lemeshow test, the R function `hoslem.test` from the R package

ResourceSelection (Lele et al., 2019) is applied.

3.4 | C , B , and Q index

As a measure of discrimination, the C index (Goodman & Kruskal, 1954) is considered.

$$C = \sum_{\substack{j_1=1 \\ y_{j_1}=0}}^n \sum_{\substack{j_2=1 \\ y_{j_2}=1}}^n \{ \mathbb{1}_{(\hat{p}_{j_2} > \hat{p}_{j_1})} + \frac{1}{2} \mathbb{1}_{(\hat{p}_{j_2} = \hat{p}_{j_1})} \} / (n_0 n_1) \text{ where } \hat{p} = \hat{p}(\text{class} = 1 | \mathbf{x}). \quad (17)$$

It compares the posterior probabilities \hat{p}_{j_2} of each observation with the true class label $y_{j_2} = 1$ to the posterior probabilities \hat{p}_{j_1} of each observation with the true class label $y_{j_1} = 0$.

The B , and the Q index are considered as measures of overall performance, which measure both, aspects of discrimination as well as calibration.

The B and Q indices (Greenberg & Sen, 1985) take the deviation between the true class label and the estimated posterior probability into account.

$$B = 1 - \sum_{j=1}^n (\hat{p}_j - y_j)^2 / n,$$

$$Q = \sum_{j=1}^n [1 + \log_2 \{ \hat{p}_j^{y_j} (1 - \hat{p}_j)^{(1-y_j)} \}] / n. \quad (18)$$

The Q index is not defined if there are any posterior probabilities \hat{p}_j in the data set equal to zero or one, respectively.

More information about the performance measures can be found in supplementary material S.3.

4 | RESULTS

In this section, we will first analyze the algorithms' performance on the reference data sets and then summarize the results of the simulation study.

4.1 | Analysis of reference data sets

Simulations are based on two reference data sets, comprising 5 and 10 continuous predictor variables, respectively, and a binary outcome variable. Both data sets comprise scores of Likert-type data from psychological questionnaires, measured on a five-point and seven-point Likert scale, respectively. Data of both data sets deviate significantly from multivariate normality with respect to the Mardia measure of multivariate skewness (Mardia, 1970). The results are shown in Table 2.

To quantify the uncertainty of the performance measures in the reference data sets, we follow the procedure proposed by Wahl et al. (2016) using 1000 bootstrap runs. More precisely, the 0.632+ bootstrap estimate (Efron & Tibshirani, 1997) is

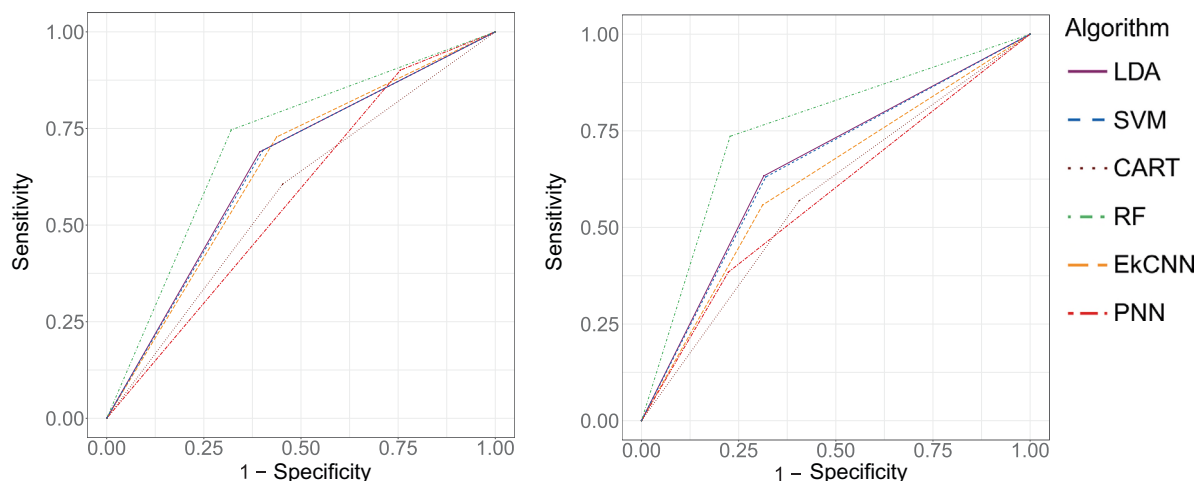


FIGURE 2 ROC curves of the algorithms when applied to reference data sets 1 (left) and 2 (right), illustrating the algorithms' discriminative performance based on the degree of similarity between true and predicted class labels.

computed and its confidence interval derived from the empirical distribution of weights assigned to each bootstrap sample. These weights are the difference between the algorithm's estimated performance obtained by training the model and evaluating its performance in the same data sets, once using the original data and once the bootstrap sample, respectively.

For the ROC curves in Figure 2, the 0.632+ bootstrap estimates of sensitivity and specificity are used.

As a measure of calibration, the Hosmer–Lemeshow test is computed in each of the bootstrap data sets (Huang et al., 2020). The number of p -values indicating a miscalibrated ($p < 0.05$), neither well calibrated nor grossly miscalibrated ($p \in (0.05, 0.1]$), or appropriate model ($p > 0.1$) are shown in Table 3. Missing values for the Hosmer–Lemeshow test occur, since the R function `hoslem.test` fails to obtain cutoffs in case the quantiles are not unique. The Q index computes the logarithm of the product of the posterior probabilities of both classes. It therefore fails in case the posterior probabilities are 0 or 1, respectively. The ROC curves in Figure 2 suggest that the discriminative ability of LDA and the linear soft-margin SVM, the only other linear classifier, are similar. The PNN algorithm tends to assign unknown samples to the larger class in the training data (for data set 1). The reason can be that the Parzen estimator is known to not adapt well to the local probability density with a data dimensionality higher than 2 or 3 (Boltz et al., 2007, 2009). The discriminative ability of the RF algorithm seems to be higher compared to LDA, but the estimates' uncertainty cannot be assessed from these plots.

Table 3 shows the averaged estimated performance measures of the algorithms. For both data sets, the overall performance (B index) of none of the algorithms can be said to be better than the one of LDA, since the CART and RF algorithms tend to have a higher variability than LDA and the other algorithms. The results of the CART algorithm usually have the highest variability. It is known that it is prone to biased estimates of the posterior probabilities (Strobl et al., 2006). The averaged performance measures suggest a better performance of the RF algorithm compared to LDA but its confidence intervals are always wider and its models are the worst calibrated among all of the considered classification algorithms. A worse than random performance cannot be ruled out for the RF algorithm considering the confidence intervals of the overall performance measures (B and Q index). The LDA results are more stable and mostly indicate a better than random performance.

4.2 | Results of the simulation study

The complete simulation results can be found in the supplementary material S.4.

Since the lognormal distribution and the correlated bimodal variables differ most from the multivariate normal distribution with respect to the Mardia measure of multivariate skewness (Mardia, 1970) and the nonparametric Kullback–Leibler divergence (Boltz et al., 2007) as shown in Table 4, only these results are presented here.

Figures 3 and 4 show the summary ROC curves for the unbalanced and balanced scenario, respectively, for the simulations based on data set 1. The mean of the estimated logit-transformed sensitivity and logit-transformed specificity together with the confidence region are shown as black dots and circles. Figure 3 suggests that only the RF algorithm performs better than LDA in nonnormally distributed data in the unbalanced simulation scenario. All algorithms have a

TABLE 3 Performance measures (95% confidence interval) of the supervised classification algorithms when applied to the reference data. Results of the Hosmer–Lemeshow test are given as the portion of p -values which are $<0.05/\in [0.05,0.1)/\geq 0.1$. In case, the Q index or Hosmer–Lemeshow test cannot be computed for all 1000 bootstrap data sets, the number of included data sets (n) is given.

	LDA	SVM	CART	RF	E&CNN	PNN
Data set 1						
Overall performance						
B index	0.779 (0.769, 0.788)	0.779 (0.768, 0.787)	0.907 (0.695, 1.0)	0.792 (0.667, 0.894)	0.773 (0.746, 0.783)	0.756 (0.753, 0.758)
Q index	0.087 (0.054, 0.113)	0.087 (0.054, 0.113)	NA ($n = 0$)	0.114 (−0.344, 0.477) ($n = 85$)	0.069 (−0.015, 0.099)	0.018 (0.008, 0.022)
Discrimination						
C index ^a	0.7 (0.667, 0.729)	0.7 (0.667, 0.73)	0.574 (0.283, 0.864)	0.762 (0.505, 0.986)	0.705 (0.634, 0.736)	0.698 (0.663, 0.725)
Sensitivity	0.689 (0.637, 0.764)	0.694 (0.642, 0.777)	0.606 (0.36, 0.841)	0.746 (0.537, 0.892)	0.728 (0.649, 0.751)	0.901 (0.846, 1.0)
Specificity	0.606 (0.529, 0.678)	0.598 (0.502, 0.663)	0.547 (0.166, 0.915)	0.679 (0.414, 0.873)	0.563 (0.425, 0.569)	0.244 (0.0, 0.445)
Youden index	0.296 (0.236, 0.359)	0.292 (0.227, 0.351)	0.153 (0.0, 0.762)	0.429 (0.0, 0.799)	0.298 (0.144, 0.334)	0.145 (0.0, 0.291)
Predictive accuracy	0.649 (0.62, 0.68)	0.647 (0.614, 0.675)	0.577 (0.266, 0.879)	0.716 (0.493, 0.901)	0.651 (0.577, 0.671)	0.58 (0.502, 0.627)
Calibration						
	0.192/	0.19/	0.3/	0.502/	0.305/	0.999/
Hosmer–Lemeshow test	0.106/	0.107/	0.007/	0.106/	0.142/	0.001
	0.702	0.703	0.693	0.392	0.553	0
Data set 2						
Overall performance						
B index	0.781 (0.757, 0.792)	0.78 (0.756, 0.793)	0.874 (0.656, 1.0)	0.803 (0.673, 0.894)	0.767 (0.729, 0.772)	0.755 (0.747, 0.756)
Q index	0.081 (−0.005, 0.117)	0.086 (0.001, 0.125)	NA ($n = 0$)	0.166 (−0.3, 0.468) ($n = 981$)	0.05 (−0.082, 0.066)	0.013 (−0.013, 0.015)
Discrimination						
C index ^a	0.717 (0.661, 0.754)	0.712 (0.655, 0.747)	0.581 (0.265, 0.863)	0.784 (0.521, 0.999)	0.677 (0.643, 0.691)	0.668 (0.606, 0.674)
Sensitivity	0.633 (0.522, 0.729)	0.631 (0.535, 0.779)	0.569 (0.252, 0.816)	0.736 (0.492, 0.865)	0.558 (0.482, 0.687)	0.384 (0.0, 0.766)
Specificity	0.685 (0.584, 0.783)	0.68 (0.552, 0.774)	0.594 (0.259, 0.873)	0.772 (0.544, 0.9)	0.688 (0.615, 0.771)	0.777 (0.66, 1.0)
Youden index	0.318 (0.195, 0.399)	0.312 (0.202, 0.407)	0.17 (0.0, 0.701)	0.508 (0.08, 0.82)	0.254 (0.175, 0.353)	0.152 (0, 0.417)
Predictive accuracy	0.659 (0.593, 0.698)	0.656 (0.6, 0.7)	0.581 (0.255, 0.847)	0.755 (0.538, 0.913)	0.626 (0.586, 0.671)	0.569 (0.501, 0.656)
Calibration						
	0.387/	0.236/	0.232/	0.89/	0.156/	0.449/
Hosmer–Lemeshow test	0.111/	0.085/	0.007/	0.057/	0.08/	0.161
	0.502	0.679	0.76 ($n = 981$)	0.053	0.764	0.39

^aThe C index is based on rank correlation between predicted posterior probabilities and true class labels. The remaining measures of discrimination only consider the similarity between predicted and true class labels.

TABLE 4 Mardia measure of multivariate skewness ($b_{1,p}$) with the number of significant test results given in parentheses, and nonparametric multivariate Kullback–Leibler (KL) divergence of the continuous distributions for the unbalanced simulation scenario ($n_0 = 50, n_1 = 100$). The results represent the mean values over 2000 simulated data sets.

Distribution	Data set 1			Data set 2		
	$b_{1,p}$		KL	$b_{1,p}$		KL
$\mathcal{N}_p(\mu_i, \Sigma)$ (1)	3.7	(48)	−0.14	23.1	(10)	−0.24
$\mathcal{LN}_p(\mu_i, \Sigma)$	29.8	(2000)	−0.16	66.6	(2000)	−0.31
$\mathcal{SN}_p(\mu_i, \Sigma, \alpha)$	4.4	(206)	0.02	23.7	(14)	0.0
$\Gamma_p(\mathbf{a}_i, \mathbf{b}_i, \mathbf{R})$	4.8	(323)	−0.05	26.0	(165)	−0.07
$\text{Bimod}_p((\mu^{bi}, \sigma^{bi}, \nu^{bi}, \tau^{bi})_{i,l}, \mathbf{R})$	4.9	(342)	0.88	24.9	(31)	7.3

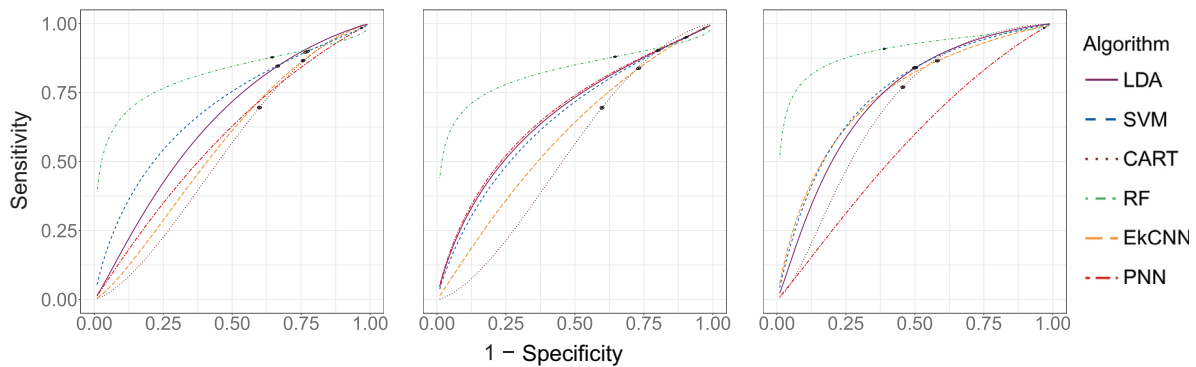


FIGURE 3 Summary ROC curves showing the algorithms' discriminative performance based on the degree of similarity between true and predicted class labels: multivariate normally (left) and multivariate lognormally (middle) distributed data and for correlated bimodal variables (right) simulated based on reference data set 1 ($n_0 = 50, n_1 = 100$). The black dots and circles represent the mean of the estimated logit-transformed sensitivity and logit-transformed specificity with their corresponding confidence region.

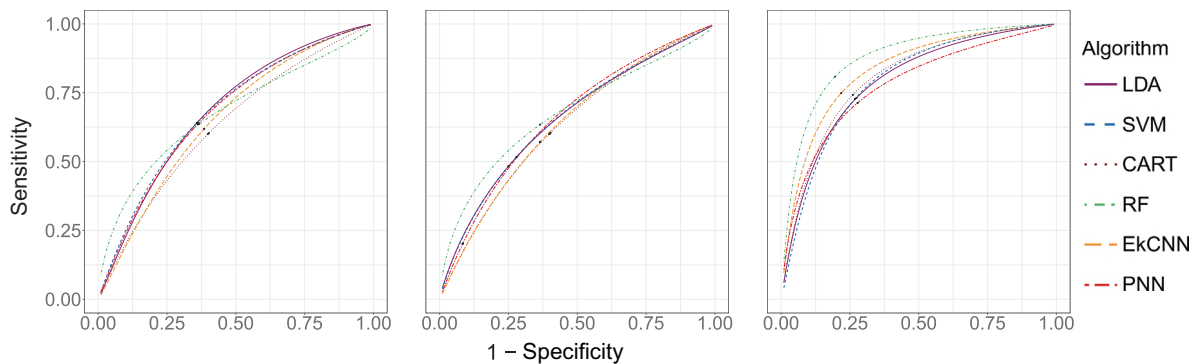


FIGURE 4 Summary ROC curves showing the algorithms' discriminative performance based on the degree of similarity between true and predicted class labels: multivariate normally (left) and multivariate lognormally (middle) distributed data and for correlated bimodal variables (right) simulated based on reference data set 1 ($n_0 = n_1 = 500$). The black dots and circles represent the mean of the estimated logit-transformed sensitivity and logit-transformed specificity with their corresponding confidence region.

rather low sensitivity. They assign new samples predominantly to the larger class from the training data. The performance of the PNN algorithm suffers most from bimodally distributed variables, probably because the more irregular pattern in the data distribution cannot be adequately estimated by the Parzen estimator. Most importantly, the performance of LDA does not decrease in the nonnormally distributed data.

Figure 4 suggests that the performance of the algorithms is almost equal for larger (and equal) sample sizes, which is comparable to the results for data set 2 (Figure S4). Therefore, we will only refer to the performance measures for the unbalanced simulation scenario in the following.

Tables 5 and 6 show the estimates of some of the performance measures for the unbalanced simulation scenarios based on data sets 1 and 2, respectively. Full results are given in Tables S3 and S5. The results which hint at a better performance of an algorithm than LDA are printed in bold, but the variance in the results does not always allow for a clear decision. The RF algorithm often outperforms LDA in the lognormal and bimodal data (and also ordinal data). The SVM algorithm also performs slightly better for various simulations of nonnormal distributions based on data set 2 (10 variables, Tables 6 and S5), but not so in data set 1 (five variables, Tables 5 and S3). The low ability of the Parzen estimator to adapt well to multivariate data distributions becomes evident in the sensitivity and specificity estimates of the PNN algorithm. For the unbalanced simulation scenarios and both data sets, its sensitivity is (almost) 1 and its specificity (almost) 0, independent of the data distribution. It assigns (almost) all test samples to the larger class from the training data (Tables S3 and S5). The Sobol sensitivity indices could not be computed for one sample in a particular data set in the unbalanced data simulation scenario based on reference data set 2 for the lognormal data (Table 6).

According to the Hosmer–Lemeshow test, the CART and RF models suffer from miscalibration in simulations based on data set 1 (Table 5), but less in data set 2 (Table 6). For 10 compared to 5 variables in data set 2, a higher number of LDA models become miscalibrated.

DISCUSSION

In this paper, we compared several supervised learning approaches in a neutral simulation study following the recommendations provided by Weber et al. (2019) and Morris et al. (2019). All of the methods under comparison have been developed previously and the authors of this study were neutral with respect to the considered methods and the potential results of the study. In particular, hyperparameter training was done according to the available recommendations to ensure the best possible result for each algorithm. A variety of performance measures was chosen to gain an objective view of the methods' performance. As noted by Niessl et al. (2021) for the context of benchmarking studies, the choice of performance measures can influence the ranking of the proposed methods. We chose performance measures from three categories as described by Steyerberg et al. (2010) through which overall performance, accuracy, and calibration of the model predictions were assessed.

One issue that we came across in our study is the handling of missing data with regard to the performance measures. The Q index cannot be computed in case posterior probabilities of zero or one occur for at least one sample in the data set. The Hosmer–Lemeshow test cannot be computed if the quantiles for deriving the cutoffs are not unique. We dropped these data sets from the results and reported the number of simulated data sets, for which the corresponding measure could be calculated. A more sophisticated handling of the missing values might change the results. However, to the best of our knowledge, there is no guidance available on how to handle missing performance values, see also Niessl et al. (2021) for a related discussion.

Our simulation settings were motivated by real-world data examples from psychology and parameters for the simulations were estimated based on these data examples. To further increase generalizability of the results, we considered two different sets of motivating data with differing numbers of variables.

We conducted both, extensive simulation studies as well as an analysis of two real-world data sets for a comprehensive comparison. The simulated data have the advantage that the true parameters, such as the underlying distributions and the distance between the groups, are known. On the other hand, they might not adequately reflect some important properties of the real-world data. Therefore, we considered both approaches in order to get a better understanding and obtain a fairer comparison between the methods.

Our study has several limitations. First, we only considered two real-world data examples. This is due to the fact that obtaining data still is a major problem despite data-sharing initiatives. Second, simulation studies can, by nature, only ever cover a limited number of scenarios. In adherence with current guidelines (Morris et al., 2019), we motivated the choice of parameters based on real data. Nonetheless and also because of the first limitation, it remains a small subset of possible scenarios and generalization of our results should be done with caution.

A special focus was on nonnormal data distributions with the aim to investigate the limitations of LDA, when its basic assumptions are violated. In accordance with previous considerations (Hastie et al., 2009), we found that LDA is rather robust against violations of the normality assumption in the settings considered in our study. Based on our results, only the RF can be considered a relevant competitor to LDA. It outperformed LDA with respect to several performance measures in highly skewed or bimodal data, but its models suffer more often from miscalibration. Also, the distribution of the reference data was not in fact bimodal. Since LDA is less computationally expensive, easier to conduct, and more

TABLE 5 Performance measures for the simulation of unbalanced data ($n_0 = 50$, $n_1 = 100$) based on data set 1. Table entries indicate the respective mean (SE) of 2000 simulated data sets. The number of analyzed data sets (n) is additionally given, in case the performance measure could not be computed for all data sets. Results of the Hosmer–Lemeshow test are given as the proportion of p -values < 0.05 / $\in [0.05, 0.1]$ / ≥ 0.1 . Results which may indicate a better performance compared to LDA are shown in bold.

	LDA	SVM	CART	RF	EkCNN	PNN
Overall performance						
<i>B</i> index						
$\mathcal{N}_5(\mu_i, \Sigma)$ (1)	0.792 (0.025)	0.789 (0.017)	0.619 (0.071)	0.783 (0.015)	0.781 (0.013)	0.783 (0.003)
$\mathcal{LN}_5(\mu_i, \Sigma)$	0.788 (0.018)	0.781 (0.011)	0.619 (0.071)	0.783 (0.015)	0.775 (0.015)	0.78 (0.004)
$\text{Bimod}_5((\mu^{bi}, \sigma^{bi}, \nu^{bi}, \tau^{bi})_{i,l}, \mathbf{R})$	0.825 (0.03)	0.822 (0.024)	0.72 (0.069)	0.846 (0.015)	0.811 (0.016)	0.791 (0.005)
Discrimination						
<i>C</i> index ^a						
$\mathcal{N}_5(\mu_i, \Sigma)$ (1)	0.672 (0.088)	0.639 (0.127)	0.539 (0.087)	0.624 (0.063)	0.615 (0.093)	0.659 (0.089)
$\mathcal{LN}_5(\mu_i, \Sigma)$	0.643 (0.09)	0.593 (0.139)	0.539 (0.087)	0.625 (0.064)	0.584 (0.098)	0.634 (0.092)
$\text{Bimod}_5((\mu^{bi}, \sigma^{bi}, \nu^{bi}, \tau^{bi})_{i,l}, \mathbf{R})$	0.783 (0.072)	0.779 (0.078)	0.66 (0.09)	0.824 (0.037)	0.767 (0.078)	0.776 (0.074)
Youden index						
$\mathcal{N}_5(\mu_i, \Sigma)$ (1)	0.189 (0.126)	0.13 (0.139)	0.144 (0.105)	0.236 (0.089)	0.121 (0.097)	0 (0)
$\mathcal{LN}_5(\mu_i, \Sigma)$	0.107 (0.103)	0.063 (0.118)	0.143 (0.105)	0.237 (0.091)	0.128 (0.104)	0.002 (0.011)
$\text{Bimod}_5((\mu^{bi}, \sigma^{bi}, \nu^{bi}, \tau^{bi})_{i,l}, \mathbf{R})$	0.361 (0.147)	0.361 (0.152)	0.341 (0.155)	0.525 (0.071)	0.297 (0.145)	0 (0.004)
Predictive accuracy						
$\mathcal{N}_5(\mu_i, \Sigma)$ (1)	0.683 (0.058)	0.678 (0.049)	0.602 (0.07)	0.707 (0.03)	0.662 (0.053)	0.667 (0)
$\mathcal{LN}_5(\mu_i, \Sigma)$	0.671 (0.045)	0.665 (0.036)	0.601 (0.07)	0.706 (0.031)	0.651 (0.059)	0.666 (0.005)
$\text{Bimod}_5((\mu^{bi}, \sigma^{bi}, \nu^{bi}, \tau^{bi})_{i,l}, \mathbf{R})$	0.741 (0.061)	0.741 (0.061)	0.71 (0.07)	0.813 (0.027)	0.729 (0.058)	0.667 (0.001)
Calibration						
Hosmer–Lemeshow test						
$\mathcal{N}_5(\mu_i, \Sigma)$ (1)	0.214/ 0.1/ 0.686/	0.124/ 0.1/ 0.775/	0.729/ 0/ 0.271 ($n = 1593$)	0.689/ 0.084/ 0.228	0.074/ 0.069/ 0.857	0.14/ 0.122/ 0.739
$\mathcal{LN}_5(\mu_i, \Sigma)$	0.168/ 0.095/ 0.738	0.155/ 0.116/ 0.729	0.733/ 0/ 0.267 ($n = 1589$)	0.701/ 0.086/ 0.213 ($n = 1999$)	0.096/ 0.093/ 0.811	0.11/ 0.107/ 0.782
$\text{Bimod}_5((\mu^{bi}, \sigma^{bi}, \nu^{bi}, \tau^{bi})_{i,l}, \mathbf{R})$	0.271/ 0.065/ 0.664	0.116/ 0.088/ 0.796	0.682/ 0.001/ 0.317 ($n = 1570$)	0.295/ 0.111/ 0.594 ($n = 1959$)	0.062/ 0.084/ 0.855	0.352/ 0.19/ 0.458

^aThe *C* index is based on rank correlation between predicted posterior probabilities and true class labels. The remaining measures of discrimination only consider the similarity between predicted and true class labels.

TABLE 6 Performance measures for the simulation of unbalanced data ($n_0 = 50$, $n_1 = 100$) based on data set 2. Table entries indicate the respective mean (SE) of 2000 simulated data sets. The number of analyzed data sets (n) is additionally given, in case the performance measure could not be computed for all data sets. Results of the Hosmer–Lemeshow test are given as the proportion of p -values $< 0.05/ \in [0.05, 0.1)/ \geq 0.1$. Results which may indicate a better performance compared to LDA are shown in bold.

		LDA	SVM	CART	RF	EkCNN	PNN
Overall performance							
B index							
	$\hat{\mathcal{N}}_{10}(\mu_i, \Sigma) (1)$	0.793 (0.032)	0.795 (0.017)	0.606 (0.071)	0.786 (0.013)	0.781 (0.016)	0.783 (0.004)
	$\mathcal{LN}_{10}(\mu_i, \Sigma)$	0.78 (0.031)	0.786 (0.014)	0.607 (0.071)	0.786 (0.013)	0.781 (0.018)	0.781 (0.007)
							($n=1999$)
	Bimod ₁₀ (($\mu^{bi}, \sigma^{bi}, \nu^{bi}, \tau^{bi}$) _{i,l} , R)	0.807 (0.034)	0.807 (0.02)	0.672 (0.071)	0.828 (0.015)	0.797 (0.017)	0.79 (0.006)
Discrimination							
C index ^a							
	$\hat{\mathcal{N}}_{10}(\mu_i, \Sigma) (1)$	0.693 (0.086)	0.676 (0.107)	0.539 (0.085)	0.62 (0.067)	0.599 (0.096)	0.631 (0.094)
	$\mathcal{LN}_{10}(\mu_i, \Sigma)$	0.646 (0.092)	0.622 (0.123)	0.54 (0.085)	0.619 (0.067)	0.594 (0.098)	0.616 (0.097)
							($n=1999$)
	Bimod ₁₀ (($\mu^{bi}, \sigma^{bi}, \nu^{bi}, \tau^{bi}$) _{i,l} , R)	0.739 (0.081)	0.73 (0.089)	0.61 (0.087)	0.781 (0.044)	0.682 (0.089)	0.725 (0.084)
Youden index							
	$\hat{\mathcal{N}}_{10}(\mu_i, \Sigma) (1)$	0.238 (0.137)	0.227 (0.142)	0.144 (0.109)	0.215 (0.092)	0.103 (0.089)	0 (0.001)
	$\mathcal{LN}_{10}(\mu_i, \Sigma)$	0.17 (0.118)	0.155 (0.121)	0.144 (0.109)	0.215 (0.092)	0.099 (0.087)	0.011 (0.025)
	Bimod ₁₀ (($\mu^{bi}, \sigma^{bi}, \nu^{bi}, \tau^{bi}$) _{i,l} , R)	0.307 (0.147)	0.297 (0.149)	0.253 (0.144)	0.433 (0.082)	0.162 (0.114)	0.001 (0.006)
Predictive accuracy							
	$\hat{\mathcal{N}}_{10}(\mu_i, \Sigma) (1)$	0.689 (0.062)	0.686 (0.062)	0.599 (0.071)	0.71 (0.028)	0.661 (0.049)	0.667 (0)
	$\mathcal{LN}_{10}(\mu_i, \Sigma)$	0.674 (0.058)	0.673 (0.056)	0.6 (0.071)	0.71 (0.028)	0.662 (0.047)	0.667 (0.011)
	Bimod ₁₀ (($\mu^{bi}, \sigma^{bi}, \nu^{bi}, \tau^{bi}$) _{i,l} , R)	0.713 (0.065)	0.709 (0.065)	0.666 (0.071)	0.785 (0.029)	0.687 (0.051)	0.667 (0.002)
Calibration							
Hosmer–Lemeshow test							
	$\hat{\mathcal{N}}_{10}(\mu_i, \Sigma) (1)$	0.4/ 0.095/ 0.504	0.105/ 0.083/ 0.812	0.574/ 0/ 0.426 ($n=1717$)	0.433/ 0.117/ 0.45	0.086/ 0.082/ 0.832	0.096/ 0.102/ 0.802
	$\mathcal{LN}_{10}(\mu_i, \Sigma)$	0.416/ 0.098/ 0.486	0.114/ 0.114/ 0.796	0.581/ 0/ 0.419 ($n = 1707$)	0.438/ 0.112/ 0.45	0.142/ 0.093/ 0.765	0.104/ 0.108/ 0.788 ($n = 1999$)
	Bimod ₁₀ (($\mu^{bi}, \sigma^{bi}, \nu^{bi}, \tau^{bi}$) _{i,l} , R)	0.428/ 0.086/ 0.486	0.097/ 0.079/ 0.824	0.601/ 0/ 0.399 ($n = 1710$)	0.207/ 0.14/ 0.653	0.06/ 0.068/ 0.871	0.21/ 0.134/ 0.657

^aThe C index is based on rank correlation between predicted posterior probabilities and true class labels. The remaining measures of discrimination only consider the similarity between predicted and true class labels.

established in the field, our study provides reassurance for applied researchers in the following sense: Even if there are scenarios where LDA is outperformed with respect to some performance measures, the differences between the methods are marginal (except for the situations described above). Thus, it seems reasonable to apply LDA even if violations of the normality assumption are suspected. Nevertheless, nonparametric classification algorithms may also work well and have been applied to psychological data (Ammerman et al., 2018; Carson et al., 1999; Fife & D'Onofrio, 2021; Garcia-Chimeno et al., 2015; Hill et al., 2017; Islam et al., 2018; Liu & Cheng, 2017; Noh et al., 2012; Pagan et al., 2005; Wallert et al., 2018).

Finally, we compare our results to previous research findings from studies which compared the performance of LDA and nonparametric supervised classification algorithms.

Lee and Jun (2008) compared several supervised classification algorithms, including LDA, CART, SVM with nonlinear kernels and the k -NN algorithm, based on 14 data sets from the University of California, Irvine (UCI) repository. The number of variables (median = 14.5 compared to 5 and 10 variables in the current study) and total sample sizes (median = 807 compared to 1109 and 380 in the current study) differed a bit from our reference data sets. They found that LDA outperformed CART and k -NN in the majority of cases. The SVM algorithm better discriminated between the classes than LDA using the cross-validated misclassification error. Feldesman (2002) compared the performance of LDA and CART in a four-class classification problem with 10 predictor variables. Using 10-fold cross-validated predictive accuracy, he found that results of the LDA and CART algorithms were relatively similar, with LDA performing slightly better. Finch et al. (2014) examined misclassification rates of several algorithms for inhomogeneous class sizes, and found that CART performed better than LDA. Holden et al. (2011) compared the performance of LDA and CART, among others, in simulations with four independent variables. They considered varying distributional parameters and relationships between the independent and dependent variables. They found that CART almost always outperformed LDA. They used a single covariance structure, exhibiting high multicollinearity among some variables. LDA is not suitable in case of high correlation among the predictor variables.

In summary, the good performance of LDA has also been found in other method comparison studies, but in which only the discriminative ability of the algorithms was considered.

ACKNOWLEDGMENT

Open access funding enabled and organized by Projekt DEAL.

CONFLICT OF INTEREST


The authors declare that there is no conflict of interest that could be perceived as prejudicing the impartiality of the research reported.

DATA AVAILABILITY STATEMENT

Dataset 1: The data are available on request from the corresponding author.

Dataset 2: The data that support the findings of this study were derived from the following resources and are publicly available: <https://www.chicagofaces.org/download/>

OPEN RESEARCH BADGES

 This article has earned an Open Data badge for making publicly available the digitally-shareable data necessary to reproduce the reported results. The data is available in the [Supporting Information](#) section.

This article has earned an open data badge “**Reproducible Research**” for making publicly available the code necessary to reproduce the reported results. The results reported in this article could fully be reproduced.

ORCID

Ricarda Graf  <https://orcid.org/0000-0002-0149-479X>

Sarah Friedrich  <https://orcid.org/0000-0003-0291-4378>

REFERENCES

- Abu Alfeilat, H. A., Hassanat, A. B., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., & Prasath, V. S. (2019). Effects of distance measure choice on k -nearest neighbor classifier performance: A review. *Big Data*, 7(4), 221–248.
- Ammerman, B., Jacobucci, R., & McCloskey, M. (2018). Using exploratory data mining to identify important correlates of non-suicidal self-injury frequency. *Psychology of Violence*, 8(4), 515–525.

- Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, 281–305.
- Boedeker, P., & Kearns, N. (2019). Linear discriminant analysis for prediction of group membership: A user-friendly primer. *Advances in Methods and Practices in Psychological Science*, 2(3), 250–263.
- Boehmke, B., & Greenwell, B. (2019). *Hands-on machine learning with R*. Chapman and Hall/CRC.
- Boltz, S., Debreuve, E., & Barlaud, M. (2007). kNN-based high-dimensional Kullback-Leibler distance for tracking. *Proceedings of the Eighth International Workshop on Image Analysis for Multimedia Interactive Services*.
- Boltz, S., Debreuve, E., & Barlaud, M. (2009). High-dimensional statistical measure for region-of-interest tracking. *IEEE Transactions on Image Processing*, 18, 1266–1283.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. (1984). *Classification and regression trees*. Chapman and Hall/CRC.
- Carson, A. D., Bizot, E. B., Hendershot, P. E., Barton, M. G., Garvin, M. K., & Kraemer, B. (1999). Modeling career counselor decisions with artificial neural networks: Predictions of fit across a comprehensive occupational map. *Journal of Vocational Behavior*, 54(1), 196–213.
- Chen, L.-F., Liao, H.-Y. M., Ko, M.-T., Lin, J.-C., & Yu, G.-J. (2000). A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33, 1713–1726.
- Cichosz, P. (2015). *Data mining algorithms: Explained using R*. Wiley.
- Cortes, C., & Vapnik, V. N. (1995). Support-vector networks. *Machine Learning*, 20, 273–297.
- Dancey, C., & Reidy, J. (2007). *Statistics without maths for psychology*. Pearson/Prentice Hall.
- Delacre, M., Lakens, D., & Leys, C. (2017). Why psychologists should by default use Welch's t-test instead of student's t-test. *International Review of Social Psychology*, 30, 92–101.
- Dvorkin, D. (2019). lcmix: Layered and chained mixture models. *R-Forge*. <https://rdrr.io/rforge/lcmix/>
- Efron, B., & Tibshirani, R. (1997). Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, 92, 548–560.
- Feldesman, M. (2002). Classification trees as an alternative to linear discriminant analysis. *American Journal of Physical Anthropology*, 119, 257–275.
- Fife, D., & D'Onofrio, J. (2021). Common, uncommon, and novel applications of random forest in psychological research. *PsyArXiv*. <https://doi.org/10.31234/osf.io/ebsmr>
- Finch, W. H., Bolin, J. H., & Kelley, K. (2014). Group membership prediction when known groups consist of unknown subgroups: A Monte Carlo comparison of methods. *Frontiers in Psychology*, 5, 337.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 17, 179–188.
- Fix, E., & Hodges, J. L. (1951). *Discriminatory analysis: Nonparametric discrimination: Consistency properties*. USAF School of Aviation Medicine.
- Flury, B. (1997). *A first course in multivariate statistics*. Springer.
- Franceschini, C., & Loperfido, N. (2019). MaxSkew and multiskew: Two R packages for detecting, measuring and removing multivariate skewness. *Symmetry*, 11, 970.
- Fukunaga, K. (1990). *Introduction to statistical pattern recognition*. Academic Press.
- Garcia-Chimeno, Y., Garcia-Zapirain, B., & Rogers, H. L. (2015). Support vector machine classification using psychological and medical-social features in patients with fibromyalgia and arthritis. *Scholars Journal of Engineering and Technology*, 3(5A), 567–571.
- Ghawi, R., & Pfeffer, J. (2019). Efficient hyperparameter tuning with grid search for text categorization using kNN approach with BM25 similarity. *Open Computer Science*, 9, 160–180.
- Goodman, L. A., & Kruskal, W. H. (1954). Measures of association for cross classifications. *Journal of the American Statistical Association*, 49(268), 732–764.
- Greenberg, B. G., & Sen, P. K. (1985). *Biostatistics: Statistics in biomedical, public health and environmental sciences: the Bernard G. Greenberg volume*. Elsevier Science Pub. Co.
- Guyon, I., Weston, J., Barnhill, S. D., & Vapnik, V. N. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46, 389–422.
- Gweon, H. (2018). kCNN.r. *GitHub*. <https://github.com/hgweon/kcnn>
- Gweon, H., Schonlau, M., & Steiner, S. H. (2019). The *k* conditional nearest neighbor algorithm for classification and class probability estimation. *PeerJ Computer Science*, 5, e194.
- Hassan, M., & El-Bassiouni, M. (2016). Bimodal skew-symmetric normal distribution. *Communications in Statistics - Theory and Methods*, 45, 1527–1541.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference and prediction*. Springer.
- Hastie, T. J., Rosset, S., Tibshirani, R., & Zhu, J. (2004). The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5, 1391–1415.
- Hill, R. M., Oosterhoff, B., & Kaplow, J. B. (2017). Prospective identification of adolescent suicide ideation using classification tree analysis: Models for community-based screening. *Journal of Consulting and Clinical Psychology*, 85, 702–711.
- Ho, T. K. (1995). C4.5 decision forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition* (pp. 278–282).
- Hofert, M., Kojadinovic, I., Maechler, M., & Yan, J. (2022). copula: Multivariate dependence with copulas. CRAN. <https://CRAN.R-project.org/package=copula>
- Holden, J. E., Finch, W. H., & Kelley, K. (2011). A comparison of two-group classification methods. *Educational and Psychological Measurement*, 71(5), 870–901.

- Hossain, A., Rigby, R., & Stasinopoulos, M. (2017). *gamlssbsn: Bimodal skew symmetric normal distribution*. CRAN. <https://CRAN.R-project.org/package=gamlssbsn>
- Huang, Y., Li, W., Macheret, F., Gabriel, R. A., & Ohno-Machado, L. (2020). A tutorial on calibration measurements and calibration models for clinical prediction models. *Journal of the American Medical Informatics Association*, 27, 621–633.
- Islam, M. R., Kamal, A. R. M., Sultana, N., Islam, R., Moni, M. A., & Ulhaq, A. (2018). Detecting depression using K-nearest neighbors (KNN) classification technique. *International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)* (pp. 1–4).
- Johnson, R., & Wichern, D. (2007). *Applied multivariate statistical analysis*. Pearson Prentice Hall.
- Kaiser, S., Träger, D., & Leisch, F. (2011). *Generating correlated ordinal random values* (Technical Report Number 94).
- Karatzoglou, A., Smola, A., Hornik, K., & Zeileis, A. (2004). kernlab—An S4 package for kernel methods in R. *Journal of Statistical Software*, 11, 1–20. <http://www.jstatsoft.org/v11/i09/>
- Lachenbruch, P. A., Sneeringer, C. P., & Revo, L. T. (1973). Robustness of the linear and quadratic discriminant function to certain types of non-normality. *Communications in Statistics*, 1, 39–56.
- Lantz, B. (2013). *Machine learning with R*. Packt Publishing.
- Lee, S.-J., & Jun, S.-R. (2008). A comparison study of classification algorithms in data mining. *International Journal of Fuzzy Logic and Intelligent Systems*, 8, 1–5.
- Leisch, F., Kaiser, A. W. S., & Hornik, K. (2010). *orddata: Generation of artificial ordinal and binary data*. R-Forge. <https://R-Forge.R-project.org/projects/orddata/>
- Lele, S. R., Keim, J. L., & Solymos, P. (2019). *ResourceSelection: Resource selection (probability) functions for use—availability data*. <https://CRAN.R-project.org/package=ResourceSelection>
- Lin, H.-T., Lin, C.-J., & Weng, R. (2007). A note on Platt's probabilistic outputs for support vector machines. *Machine Learning*, 68, 267–276.
- Liu, C., & Cheng, Y. (2017). An application of the support vector machine for attribute-by-attribute classification in cognitive diagnosis. *Applied Psychological Measurement*, 42(1), 58–72.
- Liu, W. (2020). yap: Yet another probabilistic neural network. CRAN. <https://CRAN.R-project.org/package=yap>
- Louppe, G., Wehenkel, L., Sutura, A., & Geurts, P. (2013). Understanding variable importances in forests of randomized trees. *Advances in Neural Information Processing Systems*, 26 (pp. 431–439).
- Ma, D., Correll, J., & Wittenbrink, B. (2015a). The Chicago face database: A free stimulus set of faces and norming data. *Behavior Research Methods*, 47, 1122–1135.
- Ma, D., Correll, J., & Wittenbrink, B. (2015b). *CFD 3.0 norming data and codebook*. The University of Chicago. <https://www.chicagofaces.org/download/>
- Malley, J., Kruppa, J., Dasgupta, A., Malley, K., & Ziegler, A. (2012). Probability machines consistent probability estimation using nonparametric learning machines. *Methods of Information in Medicine*, 51, 74–81.
- Mardia, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57(3), 519–530.
- McLachlan, G. J. (1992). *Discriminant analysis and statistical pattern recognition*. Wiley.
- Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Proceedings of the Royal Society A*, 209, 415–446.
- Morris, T. P., White, I. R., & Crowther, M. J. (2019). Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, 38, 2074–2102.
- Ngunkeng, G. (2013). *Statistical analysis of skew normal distribution and its applications* (Publication No. 63) [Doctoral dissertation, Bowling Green State University]. https://scholarworks.bgsu.edu/math_diss/63
- Niessl, C., Herrmann, M. P., Wiedemann, C., Casalicchio, G., & Boulesteix, A.-L. (2021). Over-optimism in benchmark studies and the multiplicity of design and analysis options when interpreting their results. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12, e1441.
- Noh, Y.-K., Park, F., & Lee, D. (2012). Diffusion decision making for adaptive *k*-nearest neighbor classification. *Advances in Neural Information Processing Systems*, 3, 1934–1942.
- Olson, M., & Wyner, A. (2018). Making sense of random forest probabilities: a kernel perspective. CoRR abs/1812.05792.
- Pagan, J. L., Oltmanns, T. F., Whitmore, M. J., & Turkheimer, E. (2005). Personality disorder not otherwise specified: Searching for an empirically based diagnostic threshold. *Journal of Personality Disorders*, 19(6), 674–689.
- Parzen, E. (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3), 1065–1076.
- Pencina, M. J., & D'Agostino, R. B. (2019). *JAMA guide to statistics and methods*. McGraw-Hill Education. [jamaevidence.mhmedical.com/content.aspx?aid=1184195233](https://jamanetwork.com/jamaevidence.mhmedical.com/content.aspx?aid=1184195233)
- Platt, J. (2000). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, MIT Press.
- R Core Team. (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>
- Rammstedt, B., & John, O. P. (2005). Kurzversion des big five inventory (BFI-K):. *Diagnostica*, 51, 195–206.
- Rao, C. R. (1973). *Linear statistical inference and its applications*. Wiley Series in Probability and Statistics.
- Rausch, J., & Kelley, K. (2009). A comparison of linear and mixture models for discriminant analysis under nonnormality. *Behavior Research Methods*, 41, 85–98.

- Sentelle, C. (2013). `svmincrementalpath.m`. *GitHub*. <https://github.com/csentelle/simplesvmpath/commit/3bfd4abb735fa220af659917d8de930809b7540f>
- Sentelle, C., Anagnostopoulos, G. C., & Georgiopoulos, M. (2016). A simple method for solving the SVM regularization path for semidefinite kernels. *IEEE Transactions on Neural Networks and Learning Systems*, 27, 709–722.
- Shayan, Z., Kamyari, N., Shayan, L., & Naseri, P. (2015). Prediction of depression in cancer patients with different classification criteria, linear discriminant analysis versus logistic regression. *Global Journal of Health Science*, 8, 41–46.
- Sherry, A. (2006). Discriminant analysis in counseling psychology research. *Counseling Psychologist*, 34(5), 661–683.
- Sobol, I. M. (1993). Sensitivity estimates for nonlinear mathematical models. *Mathematical Modelling and Computational Experiments*, 1(4), 407–414.
- Specht, D. F. (1966). *Generation of polynomial discriminant functions for pattern recognition* [Doctoral dissertation, Stanford University]. <https://dl.acm.org/doi/book/10.5555/905199>
- Specht, D. F. (1990). Probabilistic neural networks. *Neural Networks*, 3(1), 109–118.
- Steyerberg, E. W., Vickers, A. J., Cook, N. R., Gerds, T., Gonen, M., Obuchowski, N., Pencina, M. J., & Kattan, M. W. (2010). Assessing the performance of prediction models: A framework for traditional and novel measures. *Epidemiology*, 21, 128–138.
- Strobl, C., Boulesteix, A., Zeileis, A., & Hothorn, T. (2006). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8, 25.
- Therneau, T., Atkinson, B., & Ripley, B. (2019). `rpart`: Recursive partitioning for classification, regression and survival trees. *CRAN*. <https://CRAN.R-project.org/package=rpart>
- Trabelsi, S., Elouedi, Z., & Mellouli, K. (2007). Pruning belief decision tree methods in averaging and conjunctive approaches. *International Journal of Approximate Reasoning*, 46, 568–595.
- Vapnik, V. (1982). *Estimation of dependences based on empirical data: Empirical inference science*. Springer.
- Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with S*. Springer. <https://www.stats.ox.ac.uk/pub/MASS4/>
- Wahl, S., Boulesteix, A.-L., Zierer, A., Thorand, B., & Wiel, M. (2016). Assessment of predictive performance in incomplete data by combining internal validation and multiple imputation. *BMC Medical Research Methodology*, 16, 144.
- Wallert, J., Gustafson, E., Held, C., Madison, G., Norlund, F., Essen, L., & Olsson, E. (2018). Predicting adherence to internet-delivered psychotherapy for symptoms of depression and anxiety after myocardial infarction: Machine learning insights from the U-CARE Heart Randomized Controlled Trial. *Journal of Medical Internet Research*, 20, e10754.
- Weber, L., Saelens, W., Cannoodt, R., Soneson, C., Hapfelmeier, A., Gardner, P., Boulesteix, A.-L., Saeys, Y., & Robinson, M. (2019). Essential guidelines for computational method benchmarking. *Genome Biology*, 20, 125.
- Wright, M. N., & Ziegler, A. (2017). `ranger`: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77, 1–17.
- Wuertz, D., Setz, T., & Chalabi, Y. (2020). `fMultivar`: Rmetrics—Analysing and modeling multivariate financial return distributions. *CRAN*. <https://CRAN.R-project.org/package=fMultivar>
- Zeldovich, M. (2019). *Statistics exercises III - Big five data [Unpublished data]*. Institute of Psychology, Alpen-Adria-Universität Klagenfurt, Austria.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Graf, R., Zeldovich, M., & Friedrich, S. (2024). Comparing linear discriminant analysis and supervised learning algorithms for binary classification—A method comparison study. *Biometrical Journal*, 66, 2200098. <https://doi.org/10.1002/bimj.202200098>