risks

Article

Some Insights about the Applicability of Logistic Factorisation Machines in Banking

Erika Slabber, Tanja Verster and Riaan de Jongh *

Centre for BMI, North-West University, Potchefstroom 2520, South Africa
* Correspondence: riaan.dejongh@nwu.ac.za; Tel.: +27-826651127

Abstract: Logistic regression is a very popular binary classification technique in many industries, particularly in the financial service industry. It has been used to build credit scorecards, estimate the probability of default or churn, identify the next best product in marketing, and many more applications. The machine learning literature has recently introduced several alternative techniques, such as deep learning neural networks, random forests, and factorisation machines. While neural networks and random forests form part of the practitioner's model-building toolkit, factorisation machines are seldom used. In this paper, we investigate the applicability of factorisation machines to some binary classification problems in banking. To stimulate the practical application of factorisation machines, we implement the fitting routines, based on logit loss and maximum likelihood, on commercially available software that is widely used by banks and other large financial services companies. Logit loss is usually used by the machine learning fraternity while maximum likelihood is popular in statistics. Depending on the coding of the target variable, we will show that these methods yield identical parameter estimates. Often, banks are confronted with predicting events that occur with low probability. To deal with this phenomenon, we introduce weights in the above-mentioned loss functions. The accuracy of our fitting algorithms is then studied by means of a simulation study and compared with logistic regression. The separation and prediction performance of factorisation machines are then compared to logistic regression and random forests by means of three case studies covering a recommender system, credit card fraud, and a credit scoring application. We conclude that logistic factorisation machines are worthy competitors of logistic regression in most applications, but with clear advantages in recommender systems applications where the number of predictors typically outnumbers the number of observations.

Keywords: logistic regression; factorisation machines; random forests; machine learning; recommender system; credit scoring; logit loss; maximum likelihood estimation

check for updates

Citation: Slabber, Erika, Tanja Verster, and Riaan de Jongh. 2023. Some Insights about the Applicability of Logistic Factorisation Machines in Banking. *Risks* 11: 48. https://doi.org/10.3390/risks11030048

Academic Editors: Mogens Steffensen and Elisa Luciano

Received: 24 November 2022
Revised: 30 January 2023
Accepted: 16 February 2023
Published: 21 February 2023

1. Introduction

The logistic regression model is one of the most commonly used statistical techniques for solving binary classification problems (Siddiqi 2017; Hand and Henley 1997; Baesens et al. 2016). Banks are heavily reliant on logistic regression when building retail credit scorecards. The latter methodology allows for a straightforward transformation of the estimated default probabilities into scores and facilitates the interpretation of the predictor variables in terms of their importance and effect on default through the odds ratio. Not only are these models applied to assess the creditworthiness of clients but also to other areas, such as debt collection and marketing. In marketing analytics, it has frequently been used, amongst others, to build customer attrition models and campaign response models. There are many other popular classifiers, such as neural networks, XGBoost, and random forests (see, e.g., Lessmann et al. 2015), but these are mostly used for building challenger models and not often implemented as production models. One of the reasons for this is the lack of interpretability of some of these models, which hampers their implementation and acceptance by regulators. Yet, as discussed by Gilpin et al. (2018), there are methods to overcome this limitation, e.g., Shapley values, Local Interpretable Model-agnostic Explanations (LIME), and Accumulated Local Effects (ALE) plots, but these have not yet been adopted widely. In this paper, our focus is on factorisation machines, a modelling technique that is popular among machine learners, but seldom used by statisticians and practitioners. A reason for the lack of interest among statisticians is that they are probably unaware of the many papers on factorisation machines that are mostly published in machine learning conference proceedings. Practitioners, on the other hand, use commercial software for building production systems and fitting algorithms for factorisation machines are restricted or not readily available. Therefore, the main aim of this paper is to make practitioners and statisticians aware of the power of this modelling tool as a binary classifier and to provide fitting algorithms for use in commercial software (SAS). Using the logistic regression model definition, we adapt the standard definition of factorisation machines to include logistic factorisation machines and develop fitting algorithms based on maximum likelihood and logit loss. Logit loss is usually used by the machine learning fraternity, while maximum likelihood is popular in statistics. Depending on the coding of the target variable, we will show that these methods yield identical parameter estimates. Often, banks are confronted with predicting events that occur with low probability. To deal with this phenomenon, we introduce weights in the above-mentioned loss functions. The accuracy of our fitting algorithms is then studied by means of a simulation study and compared with logistic regression. The separation and prediction performance of the methods are then investigated by means of three case studies covering a recommender system, credit card fraud, and credit scoring applications. Like factorisation machines, tree-based learning methods also address the interactive features between variables, and, therefore, we include random forests (Breiman 2001) when we investigate the predictive ability of factorisation machines. In the end, we will show that factorisation machines are worthy competitors of logistic regression and random forests and should be included in the modelling toolkit of statisticians and practitioners.

The paper is organised as follows. In the next section, we introduce logistic factorisation machines, and then, in Section 3, we discuss the algorithms for fitting logistic factorisation machines. In particular, we show that subject to the binary coding of the target variable, identical parameter estimates are obtained when minimising the sum of logit loss or by maximising the logistic log-likelihood. In Section 4, we provide an overview of metrics that are typically considered for evaluating the performance of binary classifiers. In order to test the accuracy of our routines, we conduct a simulation study of which the results are presented and discussed in Section 5. Then, in Section 6, the performance of the techniques is further evaluated in a number of case studies that involve the prediction of customer preferences (a recommender system application), prediction of credit card fraud (a fraud application), and prediction of the probability of default (a scorecard application). The first data set is an artificially constructed one that clearly demonstrates the ability of factorisation machines in a recommender system setting. The second data set is publicly available from the Kaggle website, and the last data set was sourced from one of our client projects. The latter data set is proprietary to the client and cannot be disclosed. Section 7 concludes the paper.

2. Logistic Factorisation Machines

For binary response models, the response Y of an individual or an experimental unit can take on one of two possible values, denoted for convenience by 0 and 1 (for example, $Y = 1$ if a certain event occurred, otherwise, $Y = 0$). The probability of an event occurring, given a set of predictor variables $\boldsymbol{X}$, can be written as $p(\boldsymbol{X}) = P(Y = 1 \mid \boldsymbol{X})$. Then, $p(\boldsymbol{X})$ may be modelled by the logistic function $p(\boldsymbol{X}) = \frac{1}{1+\exp(-\boldsymbol{\beta}'\boldsymbol{X})}$, where $\boldsymbol{\beta}$ is a vector of parameters and p is the linkage function in a generalised linear model. Let $\boldsymbol{x}$ denote a particular realisation of $\boldsymbol{X}$, then the odds of a positive event are $\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})}$. Notice that since $p(\boldsymbol{x}) \in [0, 1]$, the logarithm of the odds ranges in the set of real numbers, thus constituting an unrestricted continuous quantity to which it is possible to fit a linear regression model as

$$logit(p(\boldsymbol{x})) = ln\left(\frac{p(\boldsymbol{x})}{1-p(\boldsymbol{x})}\right) = \boldsymbol{\beta}'\boldsymbol{x}. \tag{1}$$

In order to simplify our notation somewhat, we will drop the conditional dependence notation $p(\boldsymbol{X})$ and use p for the probability of an event occurring. Given an observed data set $(Y_n,\ X_{n1},\ldots,X_{nK}),\ n = 1,\ldots,N$, the binary logistic regression model (LR) can now be written as

$$logit(p_n^{LR}) = \ln\left(\frac{p_n^{LR}}{1-p_n^{LR}}\right) = Z_n^{LR} = \beta_0 + \sum\nolimits_{k=1}^{K}\beta_k X_{nk} \qquad \text{for } n = 1,\ldots,N \tag{2}$$

Here, Z_n^{LR} is the linear regression predictor function, X_{nk} is the n-th observation of the k-th predictor variable, β_k is the parameter or coefficient of the k-th variable, β_0 is an intercept (or bias) term, N is the number of observations, and K is the number of predictor variables. Note from (2) that $p_n^{LR} = \exp\left(Z_n^{LR}\right)/\left(1+\exp\left(Z_n^{LR}\right)\right)$ and $1 - p_n^{LR} = 1/\left(1+\exp\left(Z_n^{LR}\right)\right)$.

Similarly, the logistic regression model with all two-way interactions (LRI) can be defined as

$$logit(p_n^{LRI}) = \ln\left(\frac{p_n^{LRI}}{1-p_n^{LRI}}\right) = Z_n^{LRI} = \beta_0 + \sum\nolimits_{k=1}^{K}\beta_k X_{nk} + \sum_{k=1}^{K-1}\sum_{j=k+1}^{K}\beta_{kj}X_{nk}X_{nj} \tag{3}$$

for $n = 1,\ldots,N$, where Z_n^{LRI} is the linear regression predictor function that incorporates all two-way interactions between the predictor variables. For a detailed exposition of logistic regression, see Kleinbaum and Regression (2005) and Hilbe (2009). In this paper, we will not be interested in the interpretation of the interaction effects between two variables but rather in whether the presence of interaction terms in the model improves prediction. Although the interpretation of the coefficient of the interaction between two variables is straightforward in linear models, Ai and Norton (2003) showed that this is not the case when fitting non-linear models. In the latter case, they showed for logit and probit models that the interaction effect could not be evaluated simply by looking at the sign, magnitude, or statistical significance of the coefficient of the interaction term. Instead, the interaction effect requires the computation of the cross derivative, and like the marginal effect of a single variable, the magnitude of the interaction effect depends on all the covariates or predictors in the model. In addition, it can have different signs for different observations, making simple summary measures of the interaction effect difficult. In particular, they showed that the sign of the interaction coefficient does not necessarily indicate the sign of the interaction effect between the two variables and that the interaction effect could be non-zero even if the coefficient of the interaction term is zero. We will not pursue this further but felt that it is important to note when interpreting interaction effects.

Rendle (2010) defined factorisation machines as a general-purpose supervised learning algorithm that can be used for both classification and regression tasks. It is an extension of a linear model that is defined to capture interactions or associations between predictor variables even within high dimensional sparse data sets. Combining the definition of factorisation machines in Rendle (2012) with the logistic model with interactions as given in (3), we can define second-order logistic factorisation machines (LFM) as follows

$$\begin{aligned} logit(p_n^{LFM}) = \ln\left(\tfrac{p_n^{LFM}}{1-p_n^{LFM}}\right) = Z_n^{LFM} \\ = \beta_0 + \sum\nolimits_{k=1}^{K}\beta_k X_{nk} + \sum\nolimits_{k=1}^{K-1}\sum\nolimits_{j=k+1}^{K}\sum\nolimits_{g=1}^{G}\phi_{kg}\phi_{jg}X_{nk}X_{nj} \qquad \text{for } n = 1,\ldots,N \end{aligned} \tag{4}$$

where G is the number of factors, and Z_n^{LFM} is the second-order factorisation machine predictor function. Note that as stated by Slabber et al. (2022),

$$\beta_{kj} \approx \sum_{g=1}^{G} \phi_{kg}\phi_{jg} \tag{5}$$

Again, β_0 is the intercept, $\beta_1, \ldots, \beta_K$ are the regression coefficients, and $\boldsymbol{\phi_k}$, $k = 1, \ldots K$ is a G dimensional vector of factor loadings for each variable. Here, $\langle \boldsymbol{\phi_k}, \boldsymbol{\phi_j} \rangle = \sum_{g=1}^{G} \phi_{kg}\phi_{jg}$ denotes the inner product of the vectors $\boldsymbol{\phi_k}$ and $\boldsymbol{\phi_j}$. Note that both the $\boldsymbol{\beta}$'s (1+K parameters) and $\boldsymbol{\phi}$'s (KG factor loadings) have to be estimated and that G is an input parameter for the fitting procedure. Therefore, the LFM model in Equation (4) has $1 + K + KG$ parameters (or coefficients), while the regression model in Equation (3) has $1 + K + K(K-1)/2$ parameters. When $G < (K-1)/2$, LFM requires fewer parameters to be estimated. The advantages and disadvantages of FMs compared to regression models with interaction are discussed in Slabber et al. (2021) and Slabber et al. (2022). As mentioned in the latter papers, when the number of predictors is large, ordinary two-way interaction regression models suffer from a combinatorial explosion of parameters which make them impractical to fit (see, e.g., James et al. (2021)). In such cases, LFM provides an attractive alternative since the number of parameters increases in a linear rather than in a quadratic way.

3. Fitting Logistic Factorization Machines

As stated previously, we have a data set $(Y_n,\ X_{n1}, \ldots, X_{nK})$, $n = 1, \ldots, N$, and we want to estimate the parameters of models (2), (3), and (4). In the machine learning literature, for binary responses $Y_n = +1$ or -1, logit loss (LL) is often used (see, e.g., Rendle (2012)), while in statistics, for binary responses $Y_n = 1$ or 0, maximum likelihood (ML) is the method of choice (see, e.g., Kleinbaum and Regression (2005)). Let Z_n denote any of the above-mentioned predictor functions, and let $l(Y_n, Z_n)$ denote some loss function, then to estimate the parameters contained in Z_n, we can minimise $\sum_{n=1}^{N} l(Y_n, Z_n)$. We will now show identical parameter estimates resulting from minimising the sum of logit loss and maximising the logistic log-likelihood.

Denote the binary encoding $+1/-1$ by Y_n' and the binary encoding $1/0$ by Y_n. Then, for binary responses Y_n', and the omission of a constant factor (see Rendle 2010), the logit loss is defined by

$$l_{LL}(Y_n',\ Z_n) = \ln(1 + \exp(-Y_n' Z_n)) = I(Y_n' = -1)\ln(1 + \exp(Z_n)) + I(Y_n' = 1)\ln(1 + \exp(-Z_n))$$

where $I(.)$ is the indicator function.

Then, since $Y_n' = 2Y_n - 1$, the above equation can immediately be written as

$$l_{LL}(Y_n',\ Z_n) = I(Y_n = 0)\ln(1 + \exp(Z_n)) + I(Y_n = 1)\ln(1 + \exp(-Z_n))\,, \tag{6}$$

We have $p_n = \exp(Z_n)/(1 + \exp(Z_n))$ and $1 - p_n = 1/(1 + \exp(Z_n))$, and, therefore, $\ln(1 - p_n) = -\ln((1 + \exp(Z_n))$ and $\ln(p_n) = -\ln((1 + \exp(-Z_n))$. Substituting this into (6) yields

$$\begin{aligned} l_{LL}(Y_n',\ Z_n) &= -I(Y_n = 0)\ln(1 - p_n) - I(Y_n = 1)\ln(p_n) \\ &= -\{(1 - Y_n)\ln(1 - p_n) + Y_n\ln(p_n)\} = -l_{ML}(Y_n, p_n) \end{aligned} \tag{7}$$

with l_{ML} is the logistic log-likelihood.

Therefore, except for a constant factor, minimising logit loss is the same as minimising the negative of the logistic log-likelihood. So minimising logit loss should provide the same parameter estimates as maximising the logistic log-likelihood.

In the imbalanced case, where the number of non-events (zeros) far outnumbers the number of events (ones), the logistic log-likelihood is usually weighted in the following way

$$\sum_{n=1}^{N} w_n \{Y_n \ln(p_n) + (1 - Y_n) \ln(1 - p_n)\}\ , \tag{8}$$

with $w_n = \frac{1}{2N_1}$ if $Y_n = 1$, $w_n = \frac{1}{2N_0}$ if $Y_n = 0$, $N_1 = \sum_{n=1}^{N} I(Y_n = 1)$, and $N_0 = \sum_{n=1}^{N} I(Y_n = 0)$ and $N_1 + N_0 = N$ (see, e.g., Venter and De Jongh (2023)). Then, the totals for the events and non-events are $\frac{1}{2}$ each, and the overall total of all the weights is 1. Again, maximising (8) will yield the same parameter estimates as minimising

$$\sum_{n=1}^{N} w_n \{\ln(1 + \exp(-Y_n' Z_n))\}.,\tag{9}$$

with $w_n = \frac{1}{2N_1}$ if $Y_n' = +1$ and $w_n = \frac{1}{2N_0}$ if $Y_n' = -1$ and $N_1 = \sum_{n=1}^{N} I(Y_n' = +1)$ $N_0 = \sum_{n=1}^{N} I(Y_n' = -1)$ and $N_1 + N_0 = N$

To fit the logistic regression models given in (2) and (3), we will use SAS PROC GLM for maximising (8) with respect to the parameters contained in Z_n^{LR} and Z_n^{LRI}. The LFMs can then be obtained by using the NLP algorithm in SAS PROC OPTMODEL to minimise (8) or (9) with respect to the parameters contained in Z_n^{FM}. Note that SAS software is widely used in the financial services industry and is often the preferred choice for implementing production systems. Note that SAS Viya incorporates a procedure for fitting factorisation machines (PROC FACTMAC), but this routine has limited application (see Slabber et al. (2022)) and cannot be applied to binary classification problems. It requires a SAS Viya licence that is more expensive than the licence required to run our routines. Note that the NLP algorithm intelligently selects starting values for the algorithm and does not require the specification thereof, although it is provided as an option. To fit the random forests, we used SAS PROC HPFOREST. Unless stated otherwise, we used the default settings when fitting these models and denote them by RF. For the implementation of imbalanced data sets, we used balanced random forests (BFR).

Once a model has been fitted, an estimate $\hat{p}_n$ for p_n becomes available. To predict or classify a binary value, an 'optimal' threshold (t) or cut-off must be determined. This will be discussed in more detail in the next section.

4. Performance Measures

As stated previously, our main objective is to evaluate the performance of LFMs in comparison with LR and LRI using typical data sets found in banking. We will discuss the measures that are mostly referenced in the literature. See, for example, Siddiqi (2017); Baesens et al. (2016); Prorokowski (2019); Engelmann and Rauhmeier (2006); and James et al. (2021). Performance measures are commonly broken into discrimination (or separability measures) and calibration (or accuracy or goodness-of-fit measures). Discrimination measures quantify the degree of separability between the two classes assessed, and calibration measures quantify how close the estimated value deviates from the observed value. See Baesens et al. (2016) for a detailed discussion. Before giving our overview of the typical performance measures, note the difference between goodness-of-fit and predictive accuracy. Goodness-of-fit is how well a model can predict data points that were used to estimate its parameters, whereas predictive accuracy is how well a model can predict new data points. To assess the latter, a data set is usually split into a so-called training set and a test or validation set. We have divided the discussion of the metrics or measures into two subsections, namely popular measures that are, in our opinion, frequently used in practice and other measures that seem to be less popular.

4.1. Popular Measures

The confusion matrix, or error matrix, and metrics derived from it are widely used in practice to assess the prediction performance of binary classifiers. The matrix is frequently cited on the internet and in reliable sources such as James et al. (2021) and Baesens et al. (2016). Consider the confusion matrix depicted in Table 1. Since we are working with binary classification, positive and negative corresponds to the target value (an event occurred or not), the columns represent the actual values, and rows represent the predicted values. Then, TP is the true positive where a positive predicted value matches the actual one, TN is the true negative where a negative predicted value matches the actual one, FP is a false positive (Type 1 error) where the predicted value is positive but actually negative, and FN is a false negative (Type 2 error) where the predicted value is negative but actually positive.

Table 1. Confusion or error matrix as a metric of prediction performance.

		Actual	
		Positive	**Negative**
Predicted	**Positive**	True positives (TP)	False positives (FP)
	Negative	False negatives (FN)	True negatives (TN)

Note that for the models considered here, some cut-off values or thresholds have to be determined to obtain the above matrix once the model has been fitted. From this matrix, a number of other metrics may be derived, namely

Accuracy (A) is defined as $A = \frac{TP+TN}{TP+FP+TN+FN}$ and is the proportion of the binary values that were predicted correctly.

Precision (P) is defined as $P = \frac{TP}{TP+FP}$ and gives the proportion of predicted positives that were truly positive. This metric is preferred if we want to be very sure of our prediction; for example, we do not want to predict a customer defaulting on a loan incorrectly.

Recall (R) or the true positive rate is defined as $R = \frac{TP}{TP+FN}$ and is the proportion of actual positives that were predicted correctly. This metric is usually used when we want to capture as many positives as possible; for example, when predicting the presence of cancer, we want to capture the disease even if we are not very sure. Note that in the statistical literature, recall is sometimes referred to as sensitivity (see James et al. 2021).

Specificity (S) is defined as $= \frac{TN}{FP+TN}$ and is the proportion of actual negatives that were predicted correctly. Note that $1 - S = \frac{FP}{FP+TN}$ is the false positive rate which is the proportion of negative cases that were incorrectly predicted as positives.

Probability of correct classification (PCC) is defined as $PCC = \left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right)/2$ and measures the performance of the classifier detecting either of the classes. Again, this measure can also be used to select a threshold which maximises the PCC.

The F1 score $F1 = 2\left(\frac{P*R}{P+R}\right)$ is the harmonic mean of recall and precision and obtains its maximum when recall equals precision. So, F1 tries to find a model with good precision and recall; for example, if you evaluate the credit worthiness of clients applying for a certain product, you want to be sure that the client is creditworthy, and you want to identify as many such clients as possible. The F1 score manages this trade-off. The measure is also frequently used to obtain the best threshold or cut-off value that results in the best F1 score. Since the F1 score gives equal weight to precision and recall, the score is often adjusted by weighting the importance of the two metrics. Many other metrics derived from the confusion matrix are possible, such as variants of the F-score, the Matthews correlation coefficient, Fowlkes-Mallows index, and diagnostic odds ratio (see Prorokowski (2019)).

The Gini coefficient $(Gini)$ is one of the most frequently used metrics for evaluating the classification performance of any model used for binary classification tasks. Contrary to the above-mentioned metrics, the Gini does not rely on the choice of a threshold. The fact that this measure is invariant to threshold choice is important since typically financial institutions employ different thresholds in different applications. For example, the threshold used for assigning credit limits or for deciding on the type of credit card to issue is not the same. The Gini coefficient is defined as $Gini = 2(AUC) - 1$, where AUC is the area under the receiver operating characteristic ROC curve. The latter curve is obtained by plotting the true positive rate (also sensitivity or recall) against the false positive rate (also 1-specificity) for a range of thresholds or cut-off values. This provides a direct link between the confusion matrix and the construction of the Gini coefficient as outlined below. Once the ROC curve is determined, the AUC tells us how much the model, regardless of the chosen threshold, is able to distinguish between positive and negative cases and therefore is a measure of the degree of separability between the two classes. The AUC typically varies between 0.5 (a model with no separation power or a 'random' model) and 1 (a model with perfect separation power), and the Gini typically varies between 0 (no separation) and 1 (perfect separation). The Gini coefficient is a way to adjust the AUC to simplify interpretation; for example, a perfectly random model has a Gini of 0, a perfect model of a Gini of 1, and a perfectly reversing model of a Gini of -1. Negative Ginis are indicative of something that has seriously gone wrong in the modelling process, for example, when predicting the wrong class (predicting the non-events instead of the events). Note that AUC, and therefore Gini, is both threshold and scale invariant and not affected by oversampling. For more detail regarding the use of Gini in practice, the interested reader is referred to Engelmann and Rauhmeier (2006); Siddiqi (2012); and (SAS Institute Inc. 2010).

Remark 1. *The metrics A, P, R, PCC, and F1 should be interpreted carefully when dealing with imbalanced data sets where the event rate is typically small, i.e., when the number of actual positives is much smaller than the actual negatives. Therefore, 'good' values for these metrics are dependent on the event rate, which is equal to the number of events (positives) divided by the number of observations. For example, if the event rate is 2% and the accuracy is 90%, then the accuracy is poor.*

4.2. Other Measures

4.2.1. Goodness-of-Fit

Maximum likelihood estimation is used to estimate the parameters in the logistic regression model and finds the minimal discrepancy between the observed response and the predicted response. The resulting summary measure of this discrepancy is known as the deviance (see McCullagh and Nelder (1989)). The Akaike information criteria (AIC) and the Schwartz-Bayesian Information Criteria (SBIC) can be useful for comparing the goodness-of-fit of models. Both these information criteria are fairly simple derivations from the deviance, adjusted for sample size and a number of predictors. However, the above-mentioned measures have no standard of magnitude, and there are no statistical tests for these indices and no cut-off for what constitutes a good fit. So, they are occasionally used to compare non-nested models, i.e., models that do not have the same cases and where one model has a subset of predictors from the other model. When models are nested, such as in our case, there is no need for AIC and SBIC, and the likelihood ratio (difference in deviances) can be used as a statistical test for goodness-of-fit. Note that AIC and SBIC give the modeller an indication of how well the model fits for a specific misclassification cost, while the Gini gives you an indication of your model's separation performance on average across all misclassification costs. Misclassification costs are mostly assumed to be equal, but this is not the case in imbalanced problems where the misclassification of a (rare) event could be worse than the misclassification of a non-event.

Other goodness-of-fit statistics that are often used in practice (see Baesens et al. (2016) and Prorokowski (2019)) include the Hosmer–Lemeshow, Spiegelhalter, Kolmogorov–Smirnoff, and Vasicek statistics as well as the Brier score. Note that these quantities do not cater for misclassification cost.

4.2.2. R-Square Measures

In logistic regression, there is no true R-squared value as there is in ordinary least squares regression; however, the deviance (see McCullagh and Nelder (1989)) is often used as the equivalent of the sum of squared errors in linear regression and many R-square measures have been based on it. Shtatland et al. (2000) analysed a number of deviance-based R-square measures and argued that the measures should be used simultaneously when comparing the quality of the fit of logistic regression models. They motivate its use for its similarity to the very popular use of the R-square statistic in linear regression and claim that the measures have some advantages compared to popular information criteria such as AIC and SBIC in terms of interpretability. Allison (2014) discusses the shortcomings of these measures in a predictive power context and recommends that the measures used by McFadden and Zarembka (1974) and Tjur (2009) be used instead.

4.2.3. The H Measure (H)

The H-measure was proposed by Hand (2009) as an alternative for AUC. This measure satisfies a criterion for coherence of performance measures that the AUC does not. According to Hand (2009), the AUC is a coherent measure of separability between classes since it is based solely on the order of scores of the objects and not on the numerical values of those scores. He argues further that separability is not the same as classification performance since separability appears to ignore the central role of the classification threshold. Implicit in the AUC is a hidden assumption that each rank of the objects is equally likely to be chosen as the threshold. This is an unrealistic assumption in most practical applications, and he then proposes the H-measure, which takes into account the context of the application without requiring a rigid value of relative misclassification cost to be set. In a recent paper, Hand and Anagnostopoulos (2022) address queries that users have raised about the measure, including questions about its interpretation, the choice of misclassification cost weighting function, and other practical concerns. Of course, defining a suitable misclassification cost function for a specific problem is not an easy task and is open to criticism. Although this measure has been promoted for some time, it has not gained popularity among practitioners. One of the reasons is that it has not been taken up in regulatory guidelines and that the Gini is often preferred in practice despite the criticism against its use (see Baesens et al. (2016)).

4.3. Summary

Because of the multitude of metrics, we have decided to stick to the widely used Gini coefficient since it provides an indication of a model's ability to separate the classes on average across all classification costs. Although it is not a coherent measure of prediction performance, it is a coherent measure of separability between classes. In our simulation study below, our main objective is to check whether the fitting algorithms converge and estimate the true parameter values accurately. A secondary objective is to compare the binary classifiers in terms of their performance. Since no real application is at the core of the simulation study, misclassification cost is of secondary importance, and we will rely on the Gini as a coherent measure of separability between classes. However, when we analyse practical data sets, we will compare the prediction performance of the models by calculating F1 scores in the following way. Using the training data set, we will fit each model and calculate the F1 score as the maximum F1 score obtained by the particular model over a range of possible cut-off values. Then, the cut-off value corresponding to the maximum F1 score, obtained on the training set, is used to obtain the F1 scores for the particular model on the validation set.

5. Simulation Study

To test the accuracy of our fitting routines in terms of correctly estimating the model parameters, we conducted a simulation study where we generated data according to model (3). We considered sample sizes of $N = 500,\ 1000,\ 1500$, and $K = 10$ predictor variables, each generated independently as standard normal variables. The model parameters were set as follows: $\beta_1 = 1$, $\beta_2 = 2,\ \beta_3 = -3$, $\beta_4 = 1$, $\beta_5 = 1$, $\beta_{12} = 1,\ \ \beta_{13} = 1,\ \beta_{23} = 2,\ \beta_{9\ 10} = -2$, and all other $\beta\prime s$ as zero. The binary responses Y_n values were then generated as follows:

$$Y_n = \begin{cases} 1 \text{ if } u_n \leq\ p_n \\ 0 \text{ otherwise} \end{cases} \text{for } n = 1, \ldots, N$$

where the u_n's are generated as standard uniform variables ($u_n \sim U(0,1)$). Note that when we minimise the sum of logit loss, all the zeros should be replaced by minus ones.

We repeated the process 200 times, and to each of the 200 data sets, we fitted LR, LRI and 2- and 4-factor LFMs. We used maximum likelihood estimation in all cases, but for FMs, in order to numerically check the equivalence of the parameter estimates from (8) and (9), we included minimising the sum of logit loss. It also served as a further check for the accuracy of our algorithmic implementations. The two- and four-factor LFMs fitted with logit loss will be denoted by LLFM2 and LLFM4, respectively, and maximum likelihood fits by MLEFM2 and MLEFM4, respectively.

Tables 2–4 contain the average, standard deviation and mean squared error (MSE) of the fitted coefficients of the predictor variables as obtained over the 200 data sets. Since the results for the different sample sizes were similar, we will only present the results for $N = 1000$ and make some remarks about the other sample sizes. Note that the bias may be obtained as the difference between the true parameter value and the average obtained over simulation runs (see Table 2). Since the data sets were generated according to model (3), one would expect that LRI should perform best, in terms of lowest bias and variance, followed by the LFMs and then LR. However, excluding LR, all LFMs obtained the lowest bias and variance and hence the smallest mean squared error, with the two-factor logistic factorisation machine being the best. The LR estimates have the lowest variance of all but are extremely biased, as is expected. When inspecting the coefficient estimates obtained for the individual simulation runs, we were not surprised, due to the proof in (7), to find that up to eight decimals, the logit loss and maximum likelihood fit of both the two- and four-factorisation machines gave the same coefficient estimates for each of the data sets. This holds for the coefficients of the predictor variables, as well as for the inner product approximations (5) of the coefficients of the interaction terms. At a summative level, the results in Tables 2–4 confirm this as well. An explanation for the better performance of the factorisation machines is that fewer parameters have to be estimated, causing less variation and hence higher accuracy. Note that LR has the lowest MSE for the predictor coefficients but exhibits the most bias for the non-zero predictor coefficients and is, as expected, clearly not the preferred model.

Table 2. Average of predictor coefficient estimates over 200 repetitions for $N = 1000$ for logistic regression (LR), logistic regression with interaction (LRI), two factor factorisation machine based on logit loss (LLFM2) and maximum likelihood (MLEFM2), and four factorisation machines (LLFM4 and MLEFM4).

Betas	True Value	LR	LRI	LLFM2	MLEFM2	LLFM4	MLEFM4
0	0	0.225	0.008	0.014	0.013	0.012	0.013
1	1	0.576	1.157	1.075	1.075	1.136	1.136
2	2	1.046	2.321	2.155	2.155	2.278	2.278
3	−3	−1.578	−3.486	−3.240	−3.240	−3.419	−3.419
4	1	0.561	1.171	1.092	1.092	1.149	1.149
5	1	0.565	1.172	1.080	1.080	1.149	1.149
6	0	0.002	0.009	0.005	0.005	0.008	0.008
7	0	−0.005	0.002	0.002	0.002	0.000	0.000
8	0	0.003	−0.005	−0.007	−0.007	−0.006	−0.006
9	0	−0.008	0.003	−0.004	−0.004	0.003	0.003
10	0	0.005	0.006	0.011	0.011	0.007	0.007

Table 3. Standard deviation of predictor coefficient estimates over 200 repetitions for $N = 1000$ for logistic regression (LR), logistic regression with interaction (LRI), two factor factorisation machine based on logit loss (LLFM2) and maximum likelihood (MLEFM2), and four factorisation machines (LLFM4 and MLEFM4).

Betas	LR	LRI	LLFM2	MLEFM2	LLFM4	MLEFM4
0	0.082	0.158	0.129	0.129	0.148	0.148
1	0.088	0.154	0.136	0.136	0.149	0.149
2	0.111	0.256	0.225	0.225	0.248	0.248
3	0.134	0.345	0.311	0.311	0.328	0.328
4	0.093	0.169	0.153	0.153	0.164	0.164
5	0.095	0.170	0.147	0.147	0.164	0.164
6	0.078	0.148	0.127	0.127	0.142	0.142
7	0.089	0.147	0.130	0.130	0.143	0.143
8	0.080	0.143	0.127	0.127	0.136	0.136
9	0.095	0.154	0.135	0.135	0.149	0.149
10	0.087	0.154	0.138	0.138	0.149	0.149

Table 4. Mean squared error of predictor coefficient estimates over 200 repetitions for $N = 1000$ for the logistic regression and factorisation machine models.

Betas	LR	LRI	LLFM2	MLEFM2	LLFM4	MLEFM4
0	0.057	0.025	0.017	0.017	0.022	0.022
1	0.188	0.049	0.024	0.024	0.041	0.041
2	0.921	0.169	0.074	0.074	0.139	0.139
3	2.039	0.355	0.154	0.154	0.284	0.284
4	0.201	0.058	0.032	0.032	0.049	0.049
5	0.198	0.058	0.028	0.028	0.049	0.049
6	0.006	0.022	0.016	0.016	0.020	0.020
7	0.008	0.021	0.017	0.017	0.021	0.021
8	0.006	0.020	0.016	0.016	0.018	0.018
9	0.009	0.024	0.018	0.018	0.022	0.022
10	0.008	0.024	0.019	0.019	0.022	0.022

As far as the estimation of the interaction coefficients is considered, we will present the results for LRI and only MLEFM2 below since the other LFMs performed similarly. The MSE for the interaction coefficients are given in Table 5 for LRI and in Table 6 for MLEFM2. Again, the results for LLFM2 and MLEFM2 were the same, and we observed slightly bigger MSE's for the four-factor LFMs (MLEFM4 and LLFM4). When comparing the MSEs of the LRI and MLEFM2 fit, remarkably, the MSE of the MLEFM2 fits is much smaller than that of LRI despite the fact that the latter model was used to generate the data sets. This is testimony to the remarkable ability of factorisation machines to approximate this model with fewer parameters. In this case, LRI had to estimate 1 + K + K(K−1)/2, i.e., 65 parameters while MLEFM2 only had to estimate 1 + K + KG, i.e., 31 parameters.

Table 5. Mean squared error of interaction coefficient estimates for LRI ($N = 1000$).

	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0.06821	0.08044	0.02662	0.02950	0.02522	0.02432	0.02343	0.02842	0.02549
2		0.20197	0.03057	0.03199	0.03260	0.03029	0.02791	0.03498	0.03719
3			0.04024	0.04144	0.04289	0.03534	0.04172	0.04525	0.04125
4				0.03165	0.02307	0.02285	0.02166	0.02837	0.02934
5					0.02131	0.02465	0.02273	0.02747	0.02052
6						0.02056	0.02078	0.02142	0.02436
7							0.01717	0.02657	0.02172
8								0.02304	0.02339
9									0.16936

Table 6. Mean squared error of interaction coefficient estimates for MLEFM2 ($N = 1000$).

	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0.00510	0.00747	0.00001	0.00003	0.00009	0.00001	0.00001	0.00010	0.00017
2		0.02911	0.00005	0.00008	0.00040	0.00009	0.00005	0.00014	0.00013
3			0.00006	0.00008	0.00031	0.00005	0.00005	0.00034	0.00014
4				0.00000	0.00000	0.00000	0.00000	0.00009	0.00028
5					0.00000	0.00000	0.00000	0.00011	0.00006
6						0.00000	0.00000	0.00009	0.00011
7							0.00000	0.00015	0.00006
8								0.00008	0.00014
9									0.02429

We also recorded the time (in seconds) that the algorithms took to converge, as well as the Gini coefficients obtained for each of the fits. The results appear in Tables 7 and 8. Clearly, the factorisation machines took a longer time to converge than the logistic regression fits. Interestingly, compared to the four-factor factorisation machines, the two-factor factorisation machines took less time on average to converge but seem to struggle more in some cases (see the larger maximum convergence times). As far as the Gini results are concerned, LR is way-off, as expected, while the other techniques give fairly similar results.

Table 7. Time that the estimation algorithms of the models took to converge over the simulation runs.

	Average	**Standard Deviation**	**Minimum**	**Maximum**
LR	0.057	0.015	0.032	0.185
LRI	0.095	0.065	0.062	0.825
LLFM2	5.807	7.158	0.364	28.634
MLEFM2	5.310	6.526	0.336	37.872
LLFM4	8.572	3.820	1.013	21.079
MLEFM4	8.491	3.917	1.101	20.204

Table 8. Gini coefficients obtained by the models over simulation runs.

	Average	Standard Deviation	Minimum	Maximum
LR	0.750	0.023	0.679	0.803
LRI	0.935	0.011	0.904	0.964
LLFM2	0.926	0.014	0.846	0.956
MLEFM2	0.926	0.013	0.844	0.956
LLFM4	0.933	0.011	0.903	0.962
MLEFM4	0.933	0.011	0.902	0.962

To test the predictive ability of the various methods considered here, we repeated the above simulation study by changing the underlying model that was previously used to generate the data set. We also randomly split the generated data sets in a 70% training set and a 30% validation set. When generating the data, we used the same specification as before but included an extra (standard normal) predictor variable with a coefficient of two. We then excluded the variable when fitting the algorithms to the data sets. The algorithms were fitted to the training set only and then used to predict the validation set. We again used Gini as our performance measure. Tables 9 and 10 contains the average fitted and predicted Ginis obtained as well as the standard deviation, minimum and maximum. In order to evaluate the performance of the models, in this setting, with a popular tree-based method, we include standard random forests (RF) as a competitor. Consider the results for the training set in Table 9. Except for LR and RF, the other algorithms give fairly similar Gini estimates with the logit loss, and the maximum likelihood fits again, providing, almost always, exactly the same Ginis for each data set. Here, LR performs the worst and RF the best. For the validation set (Table 10), the predicted Ginis of LRI and the two-factor factorisation machines are best. Interestingly the four-factor factorisation machines break down, which might be due to overfitting the training data set and then being unable to generalise to the validation set.

Table 9. Fitted Ginis on the training set obtained over 200 simulation runs.

	Average	Standard Deviation	Minimum	Maximum
LR	0.689	0.031	0.605	0.763
LRI	0.877	0.018	0.821	0.926
LLFM2	0.859	0.021	0.791	0.913
MLEFM2	0.859	0.022	0.768	0.913
LLFM4	0.873	0.018	0.816	0.924
MLEFM4	0.873	0.018	0.816	0.924
RF	0.999	0.001	0.999	1.000

The same might be true for RF; however, RF slightly outperforms LR on the validation set but performs poorly relative to LRI and the two-factor FMs. This is expected since the data sets are generated from an underlying model that favours LRI and FMs. The results in Table 10 show that the choice of G, the number of factors in the LFM, is crucial and an inappropriate choice can be catastrophic, as indicated by the negative Ginis in the table. A remark about this will be made in the last section.

Table 10. Predicted Ginis for the validation set obtained over 200 simulation runs.

	Average	**Standard Deviation**	**Minimum**	**Maximum**
LR	0.671	0.045	0.546	0.801
LRI	0.800	0.035	0.658	0.886
LLFM2	0.816	0.036	0.657	0.901
MLEFM2	0.816	0.036	0.676	0.901
LLFM4	0.245	0.235	−0.290	0.789
MLEFM4	0.251	0.222	−0.231	0.788
RF	0.696	0.049	0.537	0.798

Remark 2. *The results for the other sample sizes were similar, with the larger sample sizes providing better results in terms of less variance and bias observed.*

6. Analysis of Prediction Performance on Some Data Sets

In this section, we investigate and compare the prediction performance of LFMs with LRI and LR by analysing three data sets. Various studies have confirmed that FMs perform well in recommender system applications and in sparse data settings, but little is known about their performance in the typical binary classification problems banks deal with. In this section, we will investigate two such applications where logistic regression has a stronghold, namely fraud identification and credit scoring. However, before we discuss the latter case studies, we decided to first illustrate the behaviour and prediction performance of FMs on an artificially constructed binary data set that is typically found in recommender system applications. This will set the scene for what is to follow in the banking applications. Note that we now drop the notation that distinguished between logistic loss and maximum likelihood and will subsequently refer to the two-factor LFM as LFM2 and the four-factor LFM as LFM4 and so on.

6.1. Artificial Recommender System Example

In this subsection, an artificially constructed binary data set as given in Table 11 is studied. Theoretically, 20 users (denoted by U_1 to U_{20}) may record their preferences for 20 items (denoted by I_1 to I_{20}). Users could be customers indicating their propensity to buy certain items, recording their preferences for movies, or selecting political candidates they would vote for. Another application could be the identification of the next best product to sell, and Table 11 could then represent the products that a customer owns (indicated by ones) or does not own (indicated by zeros).

The shaded values in Table 11 indicate missing observations, in other words, values that are not available and have to be predicted from the observed values. In our case, the positive responses of the users have been organised in five blocks containing four users each, and assuming no missing values, a clear structure of identical users is visible in Table 11. For example, users U_1 to U_4 all picked items I_1 to I_4, and users U_{13} to U_{16} all picked items I_{13} to I_{16}. Of course, should the labels of users and items be randomised, the block structure of identical users will not be clearly visible. The shaded values represent 120 of the 400 values that were randomly removed from the data set and that we want to predict. Can an LFM model the underlying associations between the users using the 280 observations and then predict the missing values correctly?

Table 11. The full binary ratings data set where the shaded zeros and ones indicate the removed ratings.

	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}	I_{13}	I_{14}	I_{15}	I_{16}	I_{17}	I_{18}	I_{19}	I_{20}
U_1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
U_2	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
U_3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
U_4	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
U_5	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
U_6	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
U_7	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
U_8	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
U_9	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
U_{10}	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
U_{11}	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
U_{12}	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
U_{13}	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
U_{14}	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
U_{15}	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
U_{16}	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
U_{17}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
U_{18}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
U_{19}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
U_{20}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Before we can fit an LFM to the incomplete data set (without the missing values), we have to input it into an appropriate format understood by the algorithms. The data set is therefore encoded using one hot coding (see Slabber et al. (2021) or Rendle (2012)), which means that the two predictor variables (users and items) will escalate to 40 nominal predictors (20 users plus 20 items). If we want to fit an LFM with two factors (LFM2), this will require the estimation of 121 ($1 + K + KG = 1 + 40 + 40 \times 2 = 121$) parameters. This is a significant reduction compared to fitting a full two-way LRI where 821 ($1 + K + \frac{K(K-1)}{2} = 1 + 40 + 20(39) = 1 + 40 + 780 = 821$) parameters need to be estimated. Note that for LRI, the number of parameters to be estimated is more than the number of observations, which will result in failure; however, an unweighted LFM2 model fitted to the 280 observations obtained a near-perfect fit and prediction. For example, the estimated and predicted probabilities ($\hat{p}_n^{LFM2}$) obtained for the actual ones varied between 0.999999981 and 0.999999998 and for the actual zeros between 0.000000000 and 0.000000046. So rounded to the nearest integer, we will estimate and predict Table 11 perfectly. Therefore, if we construct a confusion matrix by using any threshold (say) between 0.0001 and 0.9999, we will obtain 100% for all the confusion matrix-based measures given in Section 4.1. Impressively, with only 70% of the binary values available, LFMs were able to pick up the block structure, and resulting associations between users and items and make a perfect prediction.

To test the power of the method further, we removed 50% of the ratings. When rounding the estimated and predicted probabilities to the nearest integer, we obtained a perfect fit but predicted two of the removed zeros as ones. However, when 70% of the values in the original set were removed, the LFM models fitted and predicted poorly. In all the above-mentioned cases studied, the LR and LRI models performed poorly. LR failed to pick up the associations between users and items because these models do not contain any interaction terms. LRI failed completely due to the input matrix being singular, caused by the fact that the number of predictors outnumbered the number of observations. LFMs perform well because the factor structure enables the individual factors to borrow strength from each other to pick up the associations between users and items (see Rendle (2010) and Slabber et al. (2021)).

Remark 3.

(a) *When inspecting the results of the LFM fits, we noticed that, when all the observations on the edges of a block structure are removed, LFMs struggle to predict those values correctly.*

(b) *A straightforward implementation of RF on this problem provided poor results. This is expected because it is well-known that RFs struggle with sparse data sets. Of course, research on improving RFs to cater for these types of problems are ongoing (see, e.g., Hariharan et al. 2017; Wang et al. 2018). However, given the fact that LFM2 provide a perfect fit on both training and validation sets, it is clearly the winner in this case.*

6.2. Credit Card Fraud Example

The contents and context of this Kaggle data set are described in Kaggle (2021). In brief, it contains data on $N = 284,807$ transactions, of which 492 are fraudulent ($Y_n = 1$) and the remainder clean ($Y_n = 0$). There are $K = 30$ regressors, of which 28 are principal component transformations of features that are not detailed due to confidentiality issues; the other two regressors are described as 'Time' and 'Amount'. Here, we refer to the regressors simply as $X_1, X_2, \ldots, X_{30}$. This is a large data set, and the illustration below follows the modelling paradigm of dividing the data set into training and validation sets. The rows of the data set were permutated randomly, and then half of them were put into the training set and the remainder into the validation set. Thus, the training data set has $N_{train} = 142,404$ observations with $N_{train,1} = 245$ frauds and $N_{train,0} = 142,159$ cleans. This data set is highly unbalanced since the fraction of frauds is only 0.172% of the total. As described in Section 2, expressions (8) and (9), the assignments of the weights in the fitting criterion are used to address this matter. We take $w_n = 1/2N_{train,1}$ if $Y_n = 1$ and $w_n = 1/2N_{train,0}$ if $Y_n = 0$. Then, the totals of the weights for the frauds and the cleans are $1/2$ each, and the overall total of all the weights is 1. This enables the small number of frauds to play a meaningful role in the model training. The results obtained for this data set are given in Tables 12 and 13. The Ginis obtained by the models for the training and validation data sets are given in Table 12, and the F1 scores are in Table 13. Since we are dealing with a highly imbalanced data set, we fitted a balanced random forest (BFR) (see Agusta 2019). Consider the results presented in Table 12. Very high Ginis are obtained by all models on the training set; however, on the validation set, there is a more than 15% drop in separation performance of BRF and the LFMs. This indicates that the models overfit the training set and fail to generalise to the validation set (see, e.g., Engelmann and Rauhmeier (2006) and James et al. (2021). Interestingly, the LRI fit does not exhibit this substantial drop in separation performance and obtains similar results as LR. Standard SAS output provides the 95% confidence interval of the LR Gini on the validation set as (0.8942, 0.9644), and since this model is parsimonious compared to LRI, it will be preferred in practice. However, when the F1 scores in Table 13 are considered, the practitioner could make a different conclusion since the best F1 score on the validation set is obtained by BRF (F1 = 0.860) and the second best by LR (0.803). BRF achieves perfect separation on the training set (F1 = 1.000), while the LR score of 0.815 is close to what LR achieved on the validation set. In the end, the consistency achieved by LR in terms of both Gini and F1 scores, as well as the interpretability advantage, will sway the practitioners' decision in its favour. Note that we did not calculate F1-scores for the LFM4 and LFM6 models since we felt that the parsimonious model (LFM2) is of more interest to the practitioner and gave similar performance to the others as seen in Table 12.

Table 12. Ginis obtained by the various models on the full data set.

Modelling Technique	Train	Validation	Number of Parameters to Estimate
LR	0.987	0.929	31
LRI	0.994	0.925	466
LFM2	0.998	0.824	91
LFM4	0.999	0.835	151
LFM6	0.999	0.741	211
BRF	0.996	0.737	

Table 13. F1 scores obtained by the various models on the full data set.

Modelling Technique	Train	Validation
LR	0.815	0.803
LRI	0.648	0.507
LFM2	0.782	0.752
BRF	1.000	0.860

In retrospect, the better performance of LR is no surprise since 28 of the 30 predictor variables resulted from a principal components analysis (PCA) where each principal component is orthogonal to each other. Therefore, it is highly unlikely that LR will be outperformed by LFMs, LRI, and even BRF since the presence of interaction terms is unlikely.

Since LFMs have distinguished themselves as performing well in data sets where the number of predictors is more than the number of observations, we randomly selected 200 fraud cases and 200 non-fraud cases from the data set. We then split the data set into a training set containing 100 fraud cases and 100 non-fraud cases and allocated the remaining observation to the validation set. The results are given in Table 14 (Gini coefficients) and Table 15 (F1 scores). Note that LRI could not be fitted because the number of parameters to be estimated (466) is much larger than the number of observations (200) resulting in the input matrix being singular.

Table 14. Ginis obtained by the models on the reduced data set.

Modelling Technique	Train	Validation	Number of Parameters to Estimate
LR	0.995	0.827	31
LRI			
LFM2	0.999	0.816	91
LFM4	0.999	0.857	151
LFM6	0.999	0.853	211
BRF	0.999	0.843	

Table 15. F1 scores obtained by the models on the reduced data set.

Modelling Technique	Train	Validation
LR	0.989	0.900
LRI		
LFM2	1.000	0.881
BRF	1.000	0.916

When we compare the results in Table 12 to that in Table 14, it is interesting to note that LR separation performance drops dramatically. Although all models seem to overfit, the separation obtained is good, with LFM4 having the highest Gini but by a small margin. This supports the finding by Rendle that FMs generally perform very well when the number of predictors outnumbers the number of observations, i.e., when $K > N$. As far as Table 14 is concerned, the F1 scores obtained by the models on the training and validation sets are close, with BRF marginally the best performer.

Remark 4.

(a) *To run PROC OPTMODEL, we had to adapt the SAS Config file by changing MEMSIZE from 2G to 10G; otherwise, one gets stuck in memory problems, especially when fitting LFM6.*

(b) *We considered various other data sets where the number of observations was reduced by keeping the number of frauds and non-frauds equal. As the sample size gets smaller, the advantage of the FMs over LR, LRI, and BRF becomes more prominent.*

6.3. Credit Scoring Example

In this section, we compare logistic factorisation machines and logistic regression with respect to their ability in separating the defaulters from non-defaulters, and we use a large proprietary data set obtained from a bank. Note that the information on the predictor variables is classified as confidential. The data set was previously analysed by De Jongh et al. (2015), and we will use a reduced data set resulting from their study that contains 38 predictor variables and 335,523 observations, of which 10,267 (3.06%) defaulted. Because of the larger number of zeros, we found that the unweighted fit performed as well as the weighted fit and will only present the results for the unweighted fits. For this data set, we have available both the original and the weights-of-evidence (WoE) transformed variables. The original data set comprised 1,294,811 observations and 802 variables that were reduced through a process of variable selection and elimination of correlated variables. The Gini statistic obtained after fitting an LR model to the WoE transformed variables of the reduced data set was 0.829, as seen in Table 8 of the above-mentioned paper. Below, we obtain similar values on a random 50% split of the reduced data set in a training and validation set.

Before fitting the models, we standardised the original variables, which is recommended by Frost (2019) if one expects the presence of interaction terms. The fits on the original standardised and the WoE transformed data set are given in Tables 16 and 17 below. As mentioned previously, we fitted the model on the training data set while the performance is evaluated on the holdout portion (validation data set). The Ginis obtained for the data sets are given in Table 16, and the F1 scores are in Table 17. As in the previous example, because we are dealing with an imbalanced data set, we fitted a balanced random forest (BFR).

Table 16. Ginis obtained by the models for the credit scoring data set.

Modelling Technique	Standardised Original Variables		WoE Transformed Variables		Number of Parameters
	Train	Validation	Train	Validation	
LR	0.772	0.783	0.814	0.824	39
LRI	0.805	0.800	0.847	0.825	742
LFM2	0.797	0.802	0.828	0.829	115
LFM4	0.804	0.804	0.832	0.830	191
LFM6	0.809	0.807	0.836	0.830	267
BRF	0.999	0.883	0.999	0.862	

Table 17. F1 scores obtained by the models for the credit scoring data set.

Modelling Technique	Standardised Original Variables		WoE Transformed Variables	
	Train	Validation	Train	Validation
LR	0.383	0.386	0.398	0.399
LRI	0.400	0.386	0.433	0.401
LFM2	0.393	0.391	0.411	0.406
BRF	0.999	0.391	0.977	0.402

Consider the results in Table 16. In both the original standardised and the WoE case, the highest Gini on the validation data set is obtained by BRF, followed by the LFM models and then the LR and LRI models. Except for BRF, where the Gini on the training set is considerably larger than that obtained on the validation set, all models seem to generalise well and exhibit no signs of overfitting. The consistency of the results of the latter models on both the training and validation set indicates the stability of the models (see, e.g., Engelmann and Rauhmeier (2006)). Now consider the results in Table 17 where the F1 scores are presented. As in the previous example, we omitted the four- and six-factor LFMs since their performance are similar to the two-factor LFM, (refer to Table 16). As seen in Table 17, BRF clearly achieve the best F1 score on the training data set (for the original and WoE variables) but perform similar to the other models on the validation sets. So, as far as predictive performance is concerned, there is little to choose between these models.

As motivated below, in the case of the WoE transformed data, one would expect that no interactions between variables are present; hence one would expect that the models studied here should perform similarly. However, in this example, this also seems to hold for the standardised original variables. This concurs with a remark made by Crook (2014) that he seldomly found significant interaction terms when building retail credit scorecards. So, in this case, practitioners will most probably select the easily interpretable and simpler LR model for further analysis and implementation.

Remark 5. *On WoE Transformation.*

Banks almost always use the WoE transformation to build retail credit scorecards using ordinary logistic regression. This so-called balance scorecard methodology (see, e.g., Siddiqi (2017)) relies on fitting logistic regression models where the predictor variables are typically binned in categories, and then the target variable is used to obtain a WoE value for each category of the predictor variables. Here, the WoE value, in a specific category, is obtained as the logarithm of the ratio of the percentage non-events and the percentage events. There are three main advantages of this transformation in credit scoring: (a) outliers and missing values are grouped in classes and binned separately, (b) no dummy variables are needed since categorical variables are handled by WoE, and (c) WoE eliminates the need for linear transformations of non-linear relationships, such as log and square root. The WoE variables are constructed by means of a transformation applied to the categories resulting from the discretization, typically by means of classification trees, of the original inputs. In our opinion, logistic factorisation machines do not have clear advantages over the above-mentioned methods when the predictor variables are WoE transformed. The main reason is that the WoE should be monotonic for each variable. According to Zeng (2014), one criterion used for a good binning algorithm is that a logistic regression run on a single WoE variable should provide a slope of close to one and an intercept close to $\ln\left(\frac{N_0}{N_1}\right)$. Schaeben (2020) further argued that WoE is a special case of logistic regression when the predictor variables are jointly conditional independent, given the target variable. It is a required modelling assumption of weights-of-evidence to ensure its features and proper estimates of probabilities. Therefore, WoE-transformed predictor variables cannot reasonably include interaction terms, providing little scope for LRI or FMs to provide better prediction performance than LR. However, if the original (untransformed variables) are used, the inclusion of proper interaction terms in logistic regression compensates for the lack of joint conditional independence completely (see Schaeben (2014)).

According to Sharma (2011), WoE transformations usually work well in logistic models not containing interaction terms, and this lack of adaptation with respect to interacting variables is one of the main criticisms that may be made regarding the mentioned methodology. However, recently, Giner-Baixauli et al. (2021) extended the WoE-based methodology to include new WoE variables that capture interaction effects. The authors also show that the extended WoE approach enables the improvement of the predictive capability over the standard WoE-based methodology.

In conclusion, if the WoE transformation is applied, we are of the opinion that there is little scope for the application of LFMs in retail credit-scoring contexts. However, in applications where the original untransformed data is used or when the number of predictors is more than the number of observations, its use as a challenger model is recommended.

7. Conclusions

In this paper, we introduced the reader to LFMs and compared their performance with logistic regression and random forests. We have implemented LFM fitting routines in SAS using PROC OPTMODEL, and since SAS is widely used by large companies, we trust that these routines will facilitate the application of LFMs commercially. A simulation study confirmed the accuracy of our routines, and by using two bank-related data sets, we illustrated the strengths and weaknesses of these models. In the process, we hope that we have succeeded in convincing the reader that LFMs are worthy competitors to random forests (BRF) and logistic regression (LR and LRI). The LFMs have clear advantages over the other methods in high-dimensional and sparse data settings and performed competitively on the banking data sets. In comparison to logistic regression (i.e., LR without interaction terms), it comes at the cost of many more parameters, which would be impractical from the model monitoring point of view. However, their inclusion as a challenger model is highly recommended.

As far as future research is concerned, we suggest that a detailed study be designed to compare the predictive performance of classifiers, including discrete choice models that address interactive effects, such as proposed by Ai and Norton (2003) and Jiang (2021). Ai and Norton (2003) studied the estimation and inference problems for interaction terms in logit and probit models. Jiang (2021) proposed a semiparametric-ordered response in which explanatory variables can interactively affect the ordered response dependent variable of interest. Such a study should include a plethora of metrics for balanced and imbalanced data sets. Additionally, as far as the predictive accuracy of LFMs is concerned, the choice of the number of factors G is important. This is worth pursuing further, along with the reduction in computational demands as well as the applicability in high-dimensional sparse data settings.

Author Contributions: Conceptualization, E.S., T.V. and R.d.J.; methodology, E.S., T.V. and R.d.J.; software, E.S.; validation E.S., T.V. and R.d.J.; formal analysis, E.S., T.V. and R.d.J.; investigation, E.S., T.V. and R.d.J.; resources, E.S., T.V. and R.d.J.; writing—original draft preparation, E.S., T.V. and R.d.J.; writing—review and editing, E.S., T.V. and R.d.J.; visualization, E.S., T.V. and R.d.J.; supervision, T.V. and R.d.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work is based on research supported in part by the Department of Science and Innovation (DSI) of South Africa. The grant holder acknowledges that opinions, findings, and conclusions or recommendations expressed in any publication generated by DSI-supported research are those of the authors and that the DSI accepts no liability whatsoever in this regard.

Data Availability Statement: The credit card fraud data set analysed in Section 6.2 can be found here: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud (accessed on 20 February 2023).

Acknowledgments: This research benefitted from insightful discussions with Hennie Venter and comments made by Charl Pretorius and two referees.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

Agusta, Zahra Putri. 2019. Modified balanced random forest for improving imbalanced data prediction. *International Journal of Advances in Intelligent Informatics* 5: 58–65. [CrossRef]

Ai, Chunrong, and Edward C. Norton. 2003. Interaction terms in logit and probit models. *Economics Letters* 80: 123–29. [CrossRef]

Allison, Paul D. 2014. Measures of fit for logistic regression. Paper presented at SAS Global Forum 2014 Conference, Washington, DC, USA, March 23–26.

Baesens, Bart, Daniel Roesch, and Harald Scheule. 2016. *Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS*. Hoboken: John Wiley & Sons.

Breiman, Leo. 2001. Random forests. *Machine Learning* 45: 5–32. [CrossRef]

Crook, Jonathan. 2014. Kruger National Park, Skukuza, South Africa. Personal communication.

De Jongh, Riaan, Erika De Jongh, Marius Pienaar, Heather Gordon-Grant, Marien Oberholzer, and Leonard Santana. 2015. The impact of pre-selected variance inflation factor thresholds on the stability and predictive power of logistic regression models in credit scoring. *ORiON* 31: 17–37. [CrossRef]

Engelmann, Bernd, and Robert Rauhmeier. 2006. *The Basel II Risk Parameters: Estimation, Validation, and Stress Testing*. Berlin/Heidelberg: Springer Science & Business Media.

Frost, Jim. 2019. *Introduction to Statistics: An Intuitive Guide for Analyzing Data and Unlocking Discoveries*. State College: Jim Publishing. ISBN 978-1-7354311-0-9.

Gilpin, Leilani H., David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. Paper presented at 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), Turin, Italy, October 1–3; pp. 80–89.

Giner-Baixauli, Carlos, Juan Tinguaro Rodríguez, Alejandro Álvaro-Meca, and Daniel Vélez. 2021. Modelling Interaction Effects by Using Extended WOE Variables with Applications to Credit Scoring. *Mathematics* 9: 1903. [CrossRef]

Hand, David J. 2009. Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning* 77: 103–23. [CrossRef]

Hand, David J., and Christoforos Anagnostopoulos. 2022. Notes on the H-measure of classifier performance. *Advances in Data Analysis and Classification*. [CrossRef]

Hand, David J., and William E. Henley. 1997. Statistical classification methods in consumer credit scoring: A review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 160: 523–41. [CrossRef]

Hariharan, Siddharth, Siddhesh Tirodkar, Alok Porwal, Avik Bhattacharya, and Aurore Joly. 2017. Random forest-based prospectivity modelling of greenfield terrains using sparse deposit data: An example from the Tanami Region, Western Australia. *Natural Resources Research* 26: 489–507. [CrossRef]

Hilbe, Joseph M. 2009. *Logistic Regression Models*. Boca Raton: Chapman and Hall/CRC.

James, Gareth, Witten Daniela, Hastie Trevor, and Tibshirani Robert. 2021. *An Introduction to Statistical Learning with Applications in R*, 2nd ed. New York: Springer. [CrossRef]

Jiang, Yixiao. 2021. Semiparametric Estimation of a Corporate Bond Rating Model. *Econometrics* 9: 23. [CrossRef]

Kaggle. 2021. Credit Card Fraud Detection Dataset. Available online: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud (accessed on 12 June 2021).

Kleinbaum, David, and Mitchel Klein Regression. 2005. *Logistic Regression: A Self-Learning Text*. New York: Springer, p. 22.

Lessmann, Stefan, Bart Baesens, Hsin-Vonn Seow, and Lyn C. Thomas. 2015. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research* 247: 124–36. [CrossRef]

McCullagh, Peter, and John A Nelder. 1989. Monographs on statistics and applied probability. *Generalized Linear Models* (second edition), Chapman and Hall (London and New York). Available online: https://www.utstat.toronto.edu/~brunner/oldclass/2201s11/readings/glmbook.pdf (accessed on 12 February 2023).

McFadden, Daniel, and Paul Zarembka. 1974. *Frontiers in Econometrics*. New York: Academic Press.

Prorokowski, Lukasz. 2019. Validation of the backtesting process under the targeted review of internal models: Practical recommendations for probability of default models. *Journal of Risk Model Validation* 13: 109–47. [CrossRef]

Rendle, Steffen. 2010. Factorization machines. Paper presented at 2010 IEEE International Conference on Data Mining, Sydney, NSW, Australia, December 13–17; pp. 995–1000.

Rendle, Steffen. 2012. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3: 1–22. [CrossRef]

SAS Institute Inc. 2010. *Predictive Modelling Using Logistic Regression (SAS Course Notes)*. Cary: SAS Institution Inc.

Schaeben, Helmut. 2014. A mathematical view of weights-of-evidence, conditional independence, and logistic regression in terms of Markov random fields. *Mathematical Geosciences* 46: 691–709. [CrossRef]

Schaeben, Helmut. 2020. Comment on "Modified Weights-of-Evidence Modeling with Example of Missing Geochemical Data". *Complexity* 2020: 1–4. [CrossRef]

Sharma, Dhruv. 2011. Evidence in favor of weight of evidence and binning transformations for predictive modeling. Available online: https://ssrn.com/abstract=1925510 (accessed on 12 February 2023).

Shtatland, Ernest S., Sara Moore, and Mary. B. Barton. 2000. Why we need an R-square measure of fit (and not only one) in PROC LOGISTIC and PROC GENMOD. Paper presented at Twenty-Fifth Annual SAS®Users Group International Conference, Indianapolis, Indiana, April 9–12; Cary: SAS Institute Inc., pp. 256–25.

Siddiqi, Naeem. 2012. *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. Hoboken: John Wiley & Sons.

Siddiqi, Naeem. 2017. *Intelligent Credit Scoring: Building and Implementing Better Credit Risk Scorecards*. Hoboken: John Wiley & Sons.

Slabber, Erika, Tanja Verster, and Riaan De Jongh. 2021. Advantages of Using Factorisation Machines as a Statistical Modelling Technique. *South African Statistical Journal* 55: 125–44. [CrossRef]

Slabber, Erika, Tanja Verster, and Riaan De Jongh. 2022. Algorithms for estimating the parameters of factorisation machines. *South African Statistical Journal* 56: 69–89. [CrossRef]

Tjur, Tue. 2009. Coefficients of determination in logistic regression models—A new proposal: The coefficient of discrimination. *The American Statistician* 63: 366–72. [CrossRef]

Venter, Hennie, and Riaan De Jongh. 2023. Variable selection by searching for good subsets. *South African Statistical Journal*. *Accepted*.

Wang, Qiang, Thanh-Tung Nguyen, Joshua Z. Huang, and Thuy Thi Nguyen. 2018. An efficient random forests algorithm for high dimensional data classification. *Advances in Data Analysis and Classification* 12: 953–72. [CrossRef]

Zeng, Guoping. 2014. A necessary condition for a good binning algorithm in credit scoring. *Applied Mathematical Sciences* 8: 3229–42. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.