Media Statistika

p-ISSN 1979 – 3693 e-ISSN 2477 – 0647
**MEDIA STATISTIKA 17(1) 2024: 13-24**
http://ejournal.undip.ac.id/index.php/media_statistika

# ENSEMBLE-BASED LOGISTIC REGRESSION ON HIGH-DIMENSIONAL DATA: A SIMULATION STUDY

**Tintrim Dwi Ary Widhianingsih, Heri Kuswanto, Dedy Dwi Prastyo**
Department of Statistics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

**e-mail**: *dwi.ary@its.ac.id*

**DOI: 10.14710/medstat.17.1.13-24**

**Article Info:**
Received: 8 September 2023
Accepted: 19 September 2024
Available Online: 14 October 2024

**Keywords:**
*Classification; ELR; High-Dimensional Data; Lorens; Affordable Medicine*
.

**Abstract:** Dramatic computation growth encourages big data era, which induces data size escalation in various fields. Apart from huge sample size, cases arise high-dimensional data having more feature size than its samples. High-computing power compels the usage of modern approaches to deal with this typical dataset, while in practice, common logistic regression method is yet applied due to its simplicity and explainability. Applying logistic regression on high-dimensional data arises multicollinearity, overfitting, and computational complexity issues. Logistic Regression Ensemble (Lorens) and Ensemble Logistic Regression (ELR) are the logistic-regression-based alternative methods proposed to solve these problems. Lorens adopts ensemble concept with mutually exclusive feature partitions to form several subsets of data, while ELR involves feature selection in the algorithm by drawing part of features based on probability ranking value. This paper uncovers the effectiveness of Lorens and ELR applied to high-dimensional data classification through simulation study under three different scenarios, i.e., with feature size variation, for imbalanced high-dimensional data, and under multicollinearity conditions. Our simulation study reveals that ELR outperforms Lorens and obtains more stable performance over different feature sizes and imbalanced data settings. On the other hand, Lorens achieves more reliable performance than ELR on a simulation study with a multicollinearity issue.

## 1. INTRODUCTION

The rapid growth of computational power emerges effortless mechanism on collecting and storing data digitally. More advanced and efficient computing technology benefits agile systems to store a huge number of new instances into a database instantly (Thudumu, et al., 2020). This enhancement influences on rapid sample size growth in a dataset (Gao, et al., 2017). Employing more samples inherently reinforces parameter estimation in a statistical model. Natural asymptotic approximation on statistics manages the condition when the sample size approximates infinity. In this case, data distribution would approach Gaussian distribution for which most of the statistical approaches assume. However, in some conditions under fixed sample size, data size can continuously grow through the number of features. For example, data that naturally have limited cases to record, e.g., gene-expression (Alon, et al., 1999; Bhattacharjee & Meyerson, 2003; Sotiriou, et al.,

2003), drug-discovery problem (Widhianingsih, et al., 2020), microarray-data (Haghighi, et al., 2022; Kuswanto, et al., 2018), and rare cases in biology (Johnstone & Titterington, 2009), is more probably to be observed intensely. Accordingly, features in a dataset exceed its sample size. In addition, difficult and expensive data collection compels the collector to prioritize getting more information on a few samples than obtaining numerous observations.

Using too many features in learning a model is most critical for data with a small sample size (Hua, et al., 2005). Advance computation system insists subtle modern techniques (Gao, et al., 2017; Li, et al., 2021; Xu, et al., 2023), including Bayesian approach (Annest, et al., 2009; Wang, et al., 2013), to handle this problem. However, in practice, traditional statistical approaches are yet practical for their simplicity and explainability, although implementing typical statistical approaches, e.g., logistic regression, on high-dimensional data can yield an unsatisfied model. There usually occur four common issues. First, logistic regression solutions on high-dimensional data are not unique due to few samples involved in parameter estimation. Second, it turns out that the multicollinearity issue, which refers to a high correlation, occurs on several couples of features. Increasing feature size could also multiply the probability of multicollinearity issues occurring in a dataset. Third, an overfitting model is prone to exist when the feature size exceeds the sample size (Ayesha, et al., 2020). During model training, an overfitting model can be detected when the performance of training data is significantly higher than when it is applied on a validation (or test) set. Moreover, overfitting can happen if the model is too complex. High dimensionality of training data contributes to the high complexity of the trained model (Romero, et al., 2010). Lastly, an elevated complexity model is computationally costly in obtaining the estimates of model parameters (Ayesha, et al., 2020).

Obtaining favorable data from high-dimensional description is critical for some reasons, especially to cope with the curse of dimensionality implying to the issues (Ayesha, et al., 2020; Destrero, et al., 2009). The well-known solutions are feature extraction and selection. Redundant and irrelevant components can be removed by transforming the original features to new defined features by feature extraction techniques. In this fashion, the new feature set preserves most information of the original dataset. On the other hand, feature selection picks part of features that are most relevant and meaningful to a particular problem (Bolon-Canedo, et al., 2016). In this way, meaningless data containing noise and redundant information are accordingly eliminated.

To deal with high-dimensional data issues, ensemble method is one of alternatives that can be adopted. Generally, the concept of an ensemble can be implemented in various cases, e.g., time series (Suhartono, et al., 2012), regression modeling (Shu & Burn, 2004), and Bayesian modeling (Duan, et al., 2007). Ensemble principals basically carry out a collection of several classification models to obtain desired output (Dietterich, 2000). The ensemble method obtains target predictions necessarily by computing the aggregation of tentative outputs from several base models. Accordingly, we can expect more preferably classification performance, especially if the base models are weak in predicting target features (Rokach, 2010). Furthermore, the ensemble approach can address the instability of parameter estimations for classification models (Buhlmann, 2012).

Two logistic regression extensions elaborated with the ensemble-based method are Logistic Regression Ensemble (Lorens) (Lim, 2007) and Ensemble Logistic Regression (ELR) (Zakharov & Dupont, 2011). Both methods are an ensemble-based method that principally works with logistic regression as the base model. Lorens algorithm partitions the input data into several subspaces with no overlapping features so that the number of features on each subspace is less than the sample size. Each subspace emerges as an input of the base

model, in this case, logistic regression, and the desired output is then decided by average probability value or majority voting of predictions from entire base models. On the other hand, ELR is designed with different principal. The ensemble concept of ELR is induced to enhance the base model in an iterative manner. Also, feature selection (Ayesha, et al., 2020; Ray, et al., 2021) is incorporated into this advancement during model training to form the compact technique in dealing with high-dimensional data. Therefore, employing a repetitious algorithm in ELR would consistently generate a single model. The last model obtained from training ELR model is assumed as the optimal one, then it is used to perform prediction to the validation (or test) set.

This paper compares the effectiveness of both methodologies under desired conditions. Some existing work of simulation studies evaluate some important approaches, e.g., SMOTE (Synthetic Minority Over-Sampling Technique) (Chawla, et al., 2002), for high-dimensional data with imbalanced classes (Blagus & Lusa, 2013; Lin & Chen, 2013), investigate typical techniques for dimensionality reduction on high-dimensional data (Chung & Keles, 2010), and explore imputation methods presence in high-dimensional data (Deng, et al., 2016). This article involves the simulation study with different schemes, including the feature size, balance ratio of the classes, and multicollinearity issues in the dataset. Model performance is measured and analyzed by evaluation criteria of classification to show the effectiveness under several conditions of the simulation study. Thus, this paper contributes in: (1) accommodating simulation study of high-dimensional data with various schemes, (2) implementation of Lorens and ELR under various conditions, (3) uncovering the effectiveness of ensemble-based methods constructed by logistic regression technique to high-dimensional data.

## 2. LITERATURE REVIEW

### 2.1. Logistic Regression

Logistic regression is one of the well-known statistical methods that handle classification problems for data with categorical target features. Suppose that matrix $\boldsymbol{X}$ denotes a data of $p$ features with size $n \times p$ with $\boldsymbol{x_i} = \{x_{i1}, x_{i2}, \dots, x_{ip}\}^T$ for $i = 1, 2, \dots, n$ and vector $\boldsymbol{y}$ with size $n \times 1$ contains target feature. For binary problem with target feature $\boldsymbol{y} = \{0, 1\}$, logistic regression is denoted by $\pi(\boldsymbol{x}_i) = \pi(x_{i1}, x_{i2}, \dots, x_{ip}) = P(y_i = 1|\boldsymbol{x_i}) = 1 - P(y_i = 0|\boldsymbol{x_i})$, with $0 \leq \pi(\boldsymbol{x_i}) \leq 1$. If Class 0 acts as a class reference, the probability function of $y_i = 0$ is expressed by Equation (1)

$$P(y_i = 0|\boldsymbol{x_i}) = \frac{1}{1 + e^{-(\beta_0 + \boldsymbol{x_i^T\beta})}} \tag{1}$$

while the probability function for $y_i = 1$ can be calculated by Equation (2)

$$P(y_i = 1|\boldsymbol{x_i}) = \frac{e^{(\beta_0 + \boldsymbol{x_i^T\beta})}}{1 + e^{(\beta_0 + \boldsymbol{x_i^T\beta})}} = \frac{1}{1 + e^{(\beta_0 + \boldsymbol{x_i^T\beta})}} \tag{2}$$

Parameter estimation of logistic regression can be calculated numerically using the Newton-Raphson method with Equation (3)

$$\{\hat{\beta}_0, \widehat{\boldsymbol{\beta}}\}_{t^*} = \{\hat{\beta}_0, \widehat{\boldsymbol{\beta}}\}_{t^*-1} - \boldsymbol{H}^{-1}_{\ell(\hat{\beta}_0, \widehat{\boldsymbol{\beta}})}\{\hat{\beta}_0, \widehat{\boldsymbol{\beta}}\}_{t^*-1} \boldsymbol{\nabla}\ell\{\hat{\beta}_0, \widehat{\boldsymbol{\beta}}\}_{t^*-1} \tag{3}$$

where $t^*$ denotes the index during the estimation iteration. Vector $\boldsymbol{\nabla}\ell(\boldsymbol{\beta})$ and matrix $\boldsymbol{H}^{-1}_{\ell(\widehat{\boldsymbol{\beta}})}$ represent gradient vector and Hessian matrix of ln-likelihood function $\ell(\beta_0, \boldsymbol{\beta}) = \sum_{i=1}^{n} y_i \ln \pi(x_i) + (1 - y_i) \ln(1 - \pi(x_i))$, respectively.

Binary logistic regression is prone to overfitting, which can lead to exploding parameter estimation $\{\hat{\beta}_0, \widehat{\boldsymbol{\beta}}\}$ when training the model. Overfitting also can induce sensitive prediction values. In this case, a small shift of parameter estimation can lead to a significant change in prediction value. This indicates high instability of the prediction obtained by binary logistic regression (Zakharov & Dupont, 2011). Accordingly, an additional term is added to the objective function, as in Equation (4), to regularize the parameter during the estimation of a logistic regression model. For average loss function $L(\beta_0, \boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} f\left(y_i(\beta_0 + x_i^T\boldsymbol{\beta})\right)$, the extended objective function of Logistic Regression is formulated as follows

$$\min_{\beta_0,\boldsymbol{\beta}} \mathcal{L}(\beta_0, \boldsymbol{\beta}) + R(\boldsymbol{\beta}) \tag{4}$$

where $R(\boldsymbol{\beta})$ denotes regularization term. Notation $R(\boldsymbol{\beta})$ can be replaced by $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p}|\beta_j|$ for $L_1$-norm (Tibshirani, 1996) and $\|\boldsymbol{\beta}\|_2^2 = \sum_{j=1}^{p}\beta_j^2$ for $L_2$-norm (Hoerl & Kennard, 2000). Alternatively, a combination of $L_1$- and $L_2$-norm, or elastic net regularization (Zou & Hastie, 2005), can also substitute $R(\boldsymbol{\beta})$ by $\sum_{j=1}^{p}\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j|$, where $\alpha \in [0, 1]$ denotes the term to control the trade-off between two regularization factors.

## 2.2. Logistic Regression Ensemble (Lorens)

The Lorens algorithm is proposed based on CERP (Classification by Ensembles from Random Partitions) principle (Ahn, et al., 2007). The main concept of CERP is to combine the collection of weak models to achieve better performance. Commonly, when the number of features exceeds the sample size, typical data treatments like feature reduction are conducted to avoid any issues during parameter estimation and to prevent poor model. However, Lorens overlooks feature reduction by partitioning data into smaller subsets. These subsets have independent features (See Figure 1). In this fashion, the feature size of each subset can be arranged not to exceed the sample size. More formally, suppose that $\boldsymbol{\Theta}$ denotes feature space, Lorens partitions the data into $m$ independent and balance subspaces $\{\theta_1, \theta_2, \ldots, \theta_m\}$. The base model, e.g., logistic regression, is trained using feature partitions so that there exist combinations of $m$ models to decide the final prediction of Lorens later. So, Lorens does not require feature selection since it already accommodates all features in the dataset. Moreover, partitioning the features can also cut the correlation between classification models (Lim, et al., 2009). The final output of Lorens could be obtained by averaging the prediction probability value or majority voting of the entire base models.

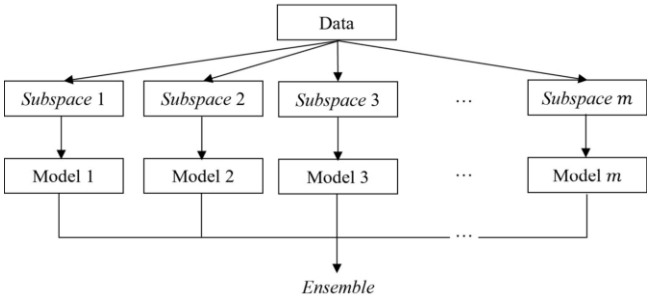



**Figure 1.** Lorens Algorithm with CERP-based Concept

## 2.3. Ensemble Logistic Regression (ELR)

ELR improves logistic regression by combining regularization strategy and feature selection altogether using an iterative ensemble approach for the high-dimensional dataset. ELR integrates $L_2$-norm regularization to obtain a sparse and stable model. To tackle the difficulty in modeling high-dimensional data, ELR performs feature selection by drawing $n$

features among total $p$ features. ELR selects these features by generating probability measures from a ranking method, e.g., $t$-test ranking combined with quality measure and parameter estimation of the trained model. The *t*-test ranking can be calculated based on Equation (5)

$$t_j = \frac{\mu_{j_0} - \mu_{j_1}}{\sqrt{\frac{\sigma_{j_0}^2}{m_0} + \frac{\sigma_{j_1}^2}{m_1}}} \tag{5}$$

The mean expression value of feature $j$ for examples of class 0 ($m_0$) is denoted as $\mu_{j_0}$, and for examples of class 1 ($m_1$), it is denoted as $\mu_{j_1}$. Correspondingly, the standard deviations for these groups are represented as $\sigma_{j_0}$ and $\sigma_{j_1}$. This probability is then updated in every iteration according to Equation (6)

$$prob_{j,t} = \frac{1}{z}\left(prob_{j,t-1} + quality \cdot \beta_j^{2 \cdot sign(quality)}\right) \tag{6}$$

where $t = 1, 2, \ldots$ denotes the iteration index, and $z$ is a normalized constant to manage the updated probability remains in the range [0, 1]. A relative quality measure is obtained from the difference of BCR values, more formally formulated as

$$quality = \log(1 + BCR_t - \overline{BCR}_{t-1}) \tag{7}$$

The $t$-test ranking initialize probability measures of entire input features $prob_{j,0}$. The ELR algorithm specifies convergent criteria based on BCR or Balanced Classification Rate, see Equation (8).

$$BCR = \frac{1}{2}\left(\frac{TP}{P_A} + \frac{TN}{N_A}\right) \tag{8}$$

According to confusion matrix of classification prediction in Table 1, BCR is formally defined as the basically an average value of sensitivity and specificity measures, with total positive class $P_A = T_P + F_N$ and total negative class $N_A = F_P + T_N$.

**Table 1.** Confusion Matrix of Classification

| | Prediction | Positive | Negative |
|---|---|---|---|
| Actual | Positive | True Positive (TP) | False Negative (FN) |
| | Negative | False Positive (FP) | True Negative (TN) |

Steps in ELR is provided in Algorithm 1.

**Algorithm 1.** Ensemble Logistic Regression

*Input* Learning sample and target feature $(\boldsymbol{X}, \boldsymbol{y})$
*Require* Regularization parameter $\lambda$
*Output* Vector $\boldsymbol{prob} \in [0,1]$

Initialize $\boldsymbol{prob}$ based on $t$-test ranking calculation
Initialize $\overline{\text{BCR}} = 0.5$
*Repeat*
    Split learning sample into training and validation set
    Draw $n$ out of $p$ features at random according to $\boldsymbol{prob}$
    Estimate parameters of L2-norm LR model
    Compute BCR of model on validation set by Equation (8)
    Compute quality measurement by Equation (7)
    *For each* $j$ among the $n$ sampled features do
        Update $\boldsymbol{prob}$ by Equation (6)
    Update average $\overline{\text{BCR}}$
*Until* no significant change of $\overline{\text{BCR}}$

BCR value determines the sign of the relative quality with the following condition. If $quality > BCR_{t-1}$ then $sign(quality) = +1$, otherwise $sign(quality) = -1$. The obtained quality from this step is then used to update the $\boldsymbol{prob}$ values based on Equation (6). ELR trains the model iteratively until the average BCR or $\overline{\text{BCR}}$ value approximates to a constant, for example until the change less than $10^{-5}$.

## 3. MATERIAL AND METHOD

Simulation data generation considers some aspects, i.e., number of features (V), imbalance ratio of classes (B), and multicollinearity (M). Simulation data to analyze the effect of feature size on high-dimensional data is generated under univariate feature and balanced data. In this scheme, feature dimension increases based on the ratio of sample and feature size $n{:}p$ over the scenarios. Generated features accommodate two categories, discrete and continuous data with 1:4 proportion, respectively. The sample size is fixed to 100 data for each feature. This simulation arranges discrete features to follow binomial distribution $x_d \sim B(\theta)$ with $\theta = 0.8$ and continuous variables to conform with univariate normal distribution $x_c \sim N(\mu, \sigma^2)$ with $\mu = 0.5$ and $\sigma^2 = 0.8$. Secondly, the simulation scenario to analyze the effect of imbalanced data consists of various class ratio levels. This scheme regularly generates 100 samples and 500 features. For a total of 500 features, the partitions involve 100 discrete and 400 continuous features without the multicollinearity effect. The distribution setting of simulation data is identical to the first scenario. A simulation study to analyze the effect of multicollinearity generates data using four scenarios. The first scenario generates features without multicollinearity for balanced data. Secondly, generated data contains no multicollinear features for imbalanced data. Then, the last two scenarios generate features with multicollinearity for balanced and imbalanced data, respectively. In this simulation study, the feature size is fixed into 200 features, with 50 discrete and 150 continuous features. For imbalanced data, the adjustment of the target feature is 20 and 80 for Class 0 and Class 1, respectively. Multivariate features are generated specifically for continuous data based on multivariate normal distribution with mean vector $\mu \sim N(1,1)$ and covariance matrix $\boldsymbol{\Sigma}$. The covariance of the multivariate normal distribution is generated from a random correlation matrix $\boldsymbol{R}$ following the procedure in (Joe, 2006). This correlation matrix is then multiplied by the random variances of all features $\boldsymbol{\sigma R \sigma}^T$ to form covariance matrix $\boldsymbol{\Sigma}$ (Qiu & Joe, 2020). Finally, binary target feature is generated based on Equation (2) with $y \sim N(0, \sigma^2)$ for random variance $\sigma^2$. The logistic regression parameters $\beta_0$ and $\beta$ are set up from normal distribution with $\mu = 0$ and $\sigma \sim N(5,1)$. The threshold to determine the target feature is fixed from class ratio $n_{y_i} = 1/n$.

Since our simulation studies also include analyzing the effect of imbalance on the models, we evaluate the trained models using AUC (Area Under the Curve). When used with balanced data, AUC performs similarly to accuracy measures; however, it offers more reliable and unbiased results when applied to imbalanced data (Prastyo, et al., 2023). AUC measures the area under the ROC (Receiver Operating Characteristic) curve by comparing the true positive rate TPR = TP/(TP+FN) = $\alpha$ and false positive rate FPR = FP/(FP+TN) = $1 - \gamma$ across different cut-off thresholds. When the decision threshold varies, the AUC can be calculated using trapezoidal integration, formulated as in Equation (9)

$$AUC = \sum_i \left( (1 - \gamma_i \Delta\alpha) + \frac{1}{2} (\Delta(1-\gamma) \Delta\alpha) \right) \tag{9}$$

where $\Delta(1-\gamma) = (1-\gamma_i) - (1-\gamma_{i-1})$ and $\Delta\alpha = \alpha_i - \alpha_{i-1}$ (Bradley, 1997).

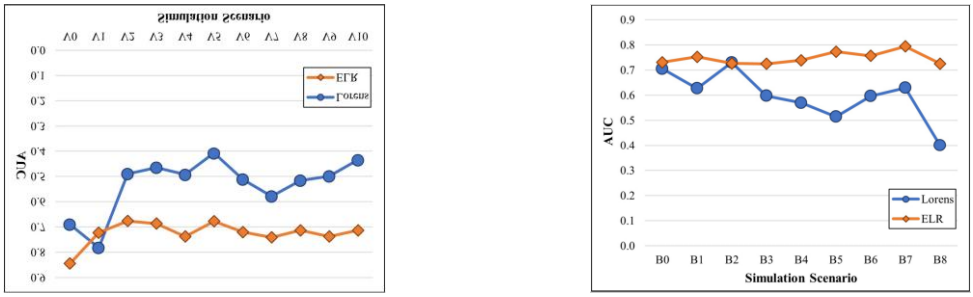
**Table 2.** AUC Score Comparison of Lorens and ELR with 95% CI on Feature Size Variation Scenario

| Scenario | Ratio $(n:p)$ | Discrete $(p_d)$ | Continuous $(p_c)$ | Lorens | ELR |
|---|---|---|---|---|---|
| V0 | 1:1 | 20 | 80 | $0.690 \pm 0.013$ | $0.844 \pm 0.028$ |
| V1 | 1:5 | 100 | 400 | $0.782 \pm 0.028$ | $0.722 \pm 0.028$ |
| V2 | 1:10 | 200 | 800 | $0.490 \pm 0.034$ | $0.677 \pm 0.025$ |
| V3 | 1:15 | 300 | 1200 | $0.465 \pm 0.045$ | $0.687 \pm 0.041$ |
| V4 | 1:20 | 400 | 1600 | $0.493 \pm 0.093$ | $0.738 \pm 0.023$ |
| V5 | 1:25 | 500 | 2000 | $0.408 \pm 0.052$ | $0.678 \pm 0.039$ |
| V6 | 1:30 | 600 | 2400 | $0.512 \pm 0.063$ | $0.720 \pm 0.039$ |
| V7 | 1:35 | 700 | 2800 | $0.579 \pm 0.056$ | $0.741 \pm 0.051$ |
| V8 | 1:40 | 800 | 3200 | $0.516 \pm 0.086$ | $0.713 \pm 0.054$ |
| V9 | 1:45 | 900 | 3600 | $0.499 \pm 0.067$ | $0.738 \pm 0.030$ |
| V10 | 1:50 | 1000 | 4000 | $0.435 \pm 0.097$ | $0.713 \pm 0.041$ |

## 4. RESULTS AND DISCUSSION

Lorens requires feature partitions to run the algorithm. The maximum number of features in each subspace is determined based on the amount of data involved in training base model. For all scenarios, we fix 25 features for each partition to run the Lorens algorithm. For example, in scenario with 1:1 for feature and sample size, there are four partitions to train Lorens' base model. Furthermore, we use 10 replications to obtain the final prediction.

A simulation study to find out the effect of feature size in high-dimensional data when using Lorens and ELR is shown in Figure 2(a). The graphs show that ELR is more stable for the two main scenarios than Lorens. In a scenario with the lowest dimension, ELR achieves 0.84 AUC score, while ELR just makes 0.69 AUC score. This indicates that for common data without high-dimensional problems or when feature size is equal to sample size $p = n$ ELR exceeds Lorens performance. Along with the increment of feature size, ELR performance decreases significantly only when data size ratio 1:5. For the rest scenarios, ELR can maintain its AUC score of around 0.7 with low variation. Compared to ELR, Lorens obtained a more significant performance reduction over the scenarios. For the first two comparable AUC score to ELR. However, it reduces quite far to around 0.5 scores for the rest scenarios with higher variation than ELR. It indicates that Lorens model tends to perform random prediction in data with large feature sizes. More detailed results are revealed in Table 2. It shows that the 95% CI (Confidence Interval) of ELR is more stable than Lorens over the increasing feature size scenarios.




(a) (b)

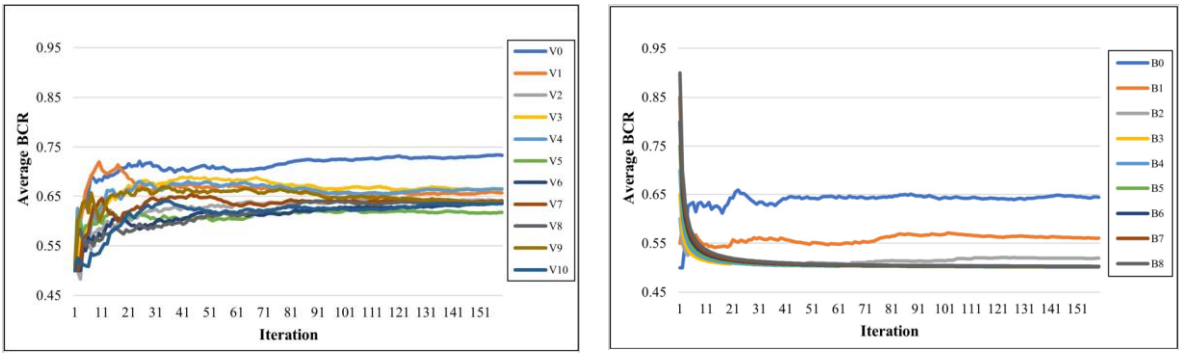
**Figure 2.** AUC Score Comparison of Lorens and ELR in A Simulation Study to Analyze (a) The Effect of Feature Size and (b) The Effect of Imbalanced Data

The second scenario to analyze the effect of imbalanced data on high-dimensional data for Lorens and ELR performance is shown in Figure 2(b). The line graph practically shows that ELR obtains stable AUC value, while Lorens performance deteriorates upon a higher imbalanced ratio of the binary target feature. More specifically, for balanced data represented by Scenario B0, Lorens, and ELR yield a model with close performance. It is like Scenario B2 when the balanced ratio between Class 0 and 1 reaches 2:3. Hereafter, Lorens performance starts being unstable with high fluctuation until reaching the least AUC score at the last scenario B8, which consists of an imbalanced data ratio 1:9 for Class 0 and Class 1, respectively. It is interestingly shown in Table 3 that 95% CI of Lorens has a slower increasing trend than ELR. ELR performs stability in AUC score, but its 95% CI jumps significantly into a wider range along the schemes from Scenario B0 to B8.

**Table 3.** AUC Score Comparison of Lorens and ELR with 95% CI on Imbalanced Data Scenario

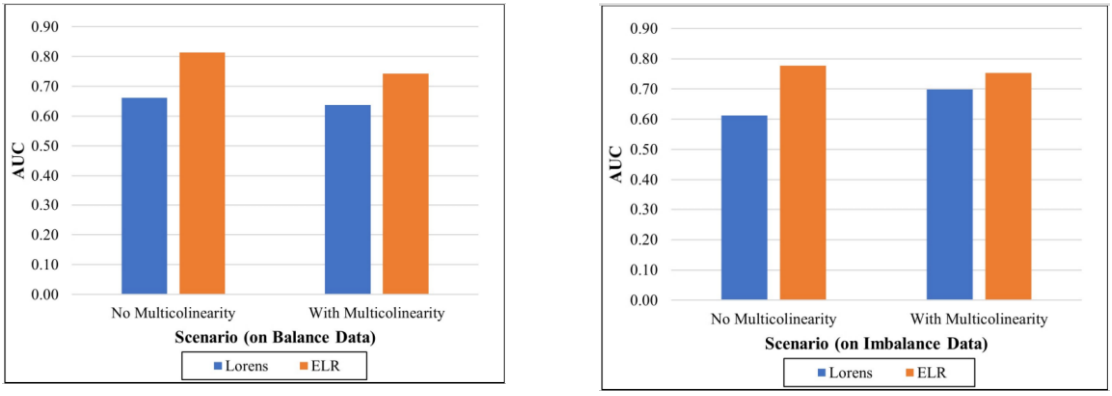
| Scenario | Ratio | Class 0 | Class 1 | Lorens | ELR |
|---|---|---|---|---|---|
| B0 | 1:1 | 50 | 50 | $0.705 \pm 0.007$ | $0.731 \pm 0.028$ |
| B1 | 1:1.22 | 45 | 55 | $0.627 \pm 0.015$ | $0.753 \pm 0.061$ |
| B2 | 1:1.5 | 40 | 60 | $0.730 \pm 0.018$ | $0.727 \pm 0.046$ |
| B3 | 1:1.86 | 35 | 65 | $0.598 \pm 0.025$ | $0.725 \pm 0.094$ |
| B4 | 1:2.33 | 30 | 70 | $0.570 \pm 0.025$ | $0.739 \pm 0.093$ |
| B5 | 1:3 | 25 | 75 | $0.515 \pm 0.030$ | $0.773 \pm 0.086$ |
| B6 | 1:4 | 20 | 80 | $0.597 \pm 0.033$ | $0.757 \pm 0.104$ |
| B7 | 1:5.67 | 15 | 85 | $0.620 \pm 0.039$ | $0.794 \pm 0.121$ |
| B8 | 1:9 | 10 | 90 | $0.400 \pm 0.017$ | $0.725 \pm 0.137$ |

Furthermore, we show the progression of BCR value when training ELR on the first two main scenarios. Figure 3(a) reveals obvious improvements with rough fluctuation for all scenarios during preliminary iterations. It subsequently varies steadily, converging into a constant. This visualization clearly shows that low-dimension data yields the highest average BCR value on almost all iterations with quite a big difference. This indicates that high-dimensional data affect model performance shown by the significant descent of average BCR in the subsequent scenarios. Interestingly, high escalation of feature size does not affect ELR performance to decrease even more. Accordingly, ELR benefits stability and robustness for feature size variation in high-dimensional data.




(a) (b)

**Figure 3.** Average BCR Progression of ELR Training Iterations in A Simulation Study with (a) Various Feature Size and (b) Imbalanced Data




(a) (b)

**Figure 4.** AUC Score Comparison of Lorens and ELR in Simulation Study under Multicollinearity Issue on (a) Balanced and (b) Imbalanced Data

Similarly, the progression of ELR on training the model under imbalanced data scenarios is shown in Figure 3(b). It exhibits that the high imbalance ratio of the target feature notably affects the ELR model. In a scenario containing balanced data, ELR has positive progression, moving from a low to a higher level and then reaching 0.65 of the average BCR value. Secondly, data, with imbalance ratio 1:1.22 consisting of low imbalanced data, carries out reasonable progression, although the average BCR falls into 0.55. Subsequently, the increasing imbalanced ratio can hurt the ELR model training process, reaching an unfavorable performance of 0.50 from originally around 0.90 of the average BCR value. It indicates that at the beginning of training iterations, ELR can predict minority classes. However, ELR performance fades, approaching 0.5 when the trained model ignores the minority class and predicts the majority class for nearly all samples.

In the multicollinearity scenario, we compare Lorens and ELR with two conditions, i.e., on balanced data and on imbalanced data. As shown in Figure 4, ELR obtains better AUC value in all schemes. However, it is obvious that multicollinearity slightly weakens ELR performance in either balanced or imbalanced data. On the other hand, AUC score of Lorens rises quite significantly for the scenario under imbalanced data. Table 4 shows that Lorens results on shorter 95% CI, which means that although on balanced data it yields decreasing AUC score, overall, Lorens obtains more consistent performance than ELR. This is reasonable due to the feature partitions used in Lorens. Splitting data features into several mutually exclusive subspaces benefits eliminating multicollinearity issues in high-dimensional data that is absent in the ELR algorithm.

**Table 4.** AUC Score Comparison of Lorens and ELR with 95% CI for Scenario under Multicollinearity Issue

| Multicollinearity | Balance Ratio | Lorens | ELR |
|---|---|---|---|
| No | 1:1 | $0.661 \pm 0.018$ | $0.813 \pm 0.036$ |
| No | 1:4 | $0.613 \pm 0.019$ | $0.778 \pm 0.109$ |
| Yes | 1:1 | $0.637 \pm 0.007$ | $0.742 \pm 0.038$ |
| Yes | 1:4 | $0.698 \pm 0.014$ | $0.753 \pm 0.132$ |

## 5. CONCLUSION

The implementation of logistic regression on high-dimensional data can lead to challenges that may adversely affect model performance. Several methods have been proposed, such as Lorens and ELR, that are designed based on ensemble approaches in a

different fashion. The Lorens algorithm is designed based on the CERP concept, while ELR brings about the iterative algorithm of the ensemble method. This paper compares the effectiveness of those approaches on high-dimensional data through a simulation study. Simulation data are generated under three main scenarios: (1) with different feature sizes, (2) for imbalanced data, and (3) under a multicollinearity problem. Our simulation study on various feature-size scenarios reveals that ELR obtains better and more stable performance than Lorens. In the second scenario, ELR outperforms Lorens with steady performance. However, the imbalanced class ratio increment leads to a wider confidence interval of ELR performance. Lastly, under the multicollinearity condition, Lorens defeats ELR as it yields more reliable performance in both balanced and imbalanced data settings.

## ACKNOWLEDGMENT

The authors gratefully acknowledge financial support from the Institut Teknologi Sepuluh Nopember for this work, under project scheme of the Publication Writing and IPR Incentive Program (PPHKI) 2024

## REFERENCES

Ahn, H., Moon, H., Fazzari, M. J., Lim, N., Chen, J. J., & Kodell, R. L. (2007). Classification by Ensembles from random Partitions of High-Dimensional Data. *Comput. Stat. Data Anal*., *15*(12), 6166-6179.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by oligonucleotide Arrays. *Proceedings of the National Academy of Sciences*, *96*(12), 6745-6750.

Annest, A., Bumgarner, R. E., Raftery, A. E., & Yeung, K. Y. (2009). Iterative Bayesian Model Averaging: A Method for the application of survival Analysis to high-Dimensional Microarray Data. *BMC Bioinformatics*, *10*(72).

Ayesha, S., Hanif, M. K., & Talib, R. (2020). Overview and Comparative Study of Dimensionality Reduction Techniques for High Dimensional Data. *Inf. Fusion*, *59*, 44-58.

Bhattacharjee, A. & Meyerson, M. (2003). Classification of Human Lung Carcinomas by mRNA Expression *Profiling. Expression Profiling of Human Tumors,* New York: Springer

Blagus, R. & Lusa, L. (2013). SMOTE for High-Dimensional Class-Imbalanced Data. *BMC Bioinformatics*, *14*(1), 106

Bolon-Canedo, V., Sanchez-Marono, N., & Alonso-Betanzos, A. (2016). *Feature Selection for High-Dimensional Data*. New York: Springer.

Bradley, A. P. (1997). The Use of the area under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognition*, *30*, 1145-1159.

Buhlmann, P. (2012). *Bagging, Boosting and Ensemble Methods*. Berlin: Springer Berlin Heidelberg.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal Of Artificial Intelligence Research*, *16*, 321-357.

Chung, D. & Keles, S. (2010). Sparse Partial Least Squares Classification for High Dimensional Data. *Statistical Applications in Genetics and Molecular Biology, 9*(1).

Deng, Y., Chang, C., Ido, M. S., & Long, Q. (2016). Multiple Imputation for general Missing Data Patterns in the presence of High-Dimensional Data. *Scientific Reports, 6*: 2021-2035.

Destrero, A., Mosci, S., Mol, C. D., Verri, A., & Odone, F. (2009). Feature selection for high-dimensional data. *Computational Management Science, 6*, 25-40.

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. In Multiple Classifier Systems, *Proceedings First International Workshop, MCS 2000*, 1-15.

Duan, Q., Ajami, N. K., Gao, X., & Sorooshian, S. (2007). Multi-Model Ensemble Hydrologic Prediction using Bayesian Model Averaging. *Advances in Water Resources, 30*(5), 1371-1386.

Gao, L., Song, J., Liu, X., Shao, J., Liu, J., & Shao, J. (2017). Learning in High-Dimensional Multimedia Data: The State of The Art. Multimedia Systems, *23*, 303-313.

Haghighi, M., Caicedo, J. C., Cimini, B. A., Carpenter, A. E., & Singh, S. (2022). High-dimensional Gene Expression and morphology Profiles of Cells Across 28,000 Genetic and Chemical Perturbations. *Nature Methods, 19*(12), 1550-1557.

Hoerl, A. E. & Kennard, R. W. (2000). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics, 42*(1), pp. 80-86

Hua, J., Xiong, Z., Lowey, J., Suh, E., & Dougherty, E. R. (2005). Optimal Number of Features as a Function of sample Size for Various Classification Rules. *Bioinformatics, 21*(8), 1509-1515.

Joe, H. (2006). Generating random Correlation Matrices based on partial Correlations. *Journal of Multivariate Analysis, 97*(10), 2177-2189.

Johnstone, I. M. & Titterington, D. M. (2009). Statistical Challenges of High-Dimensional Data. *Philosophical Transactions of the Royal Society, 367*, 4237-4253.

Kuswanto, H., Melasasi, J. N., & Ohwada, H. (2018). Enzyme Classification on DUD-E Database Using Logistic Regression Ensemble (Lorens). Innovative Computing, Optimization and Its Applications: Modelling and Simulations, *741*, 93-109.

Li, Y., Chai, Y., Yin, H., & Chen, B. (2021). A Novel Feature Learning Framework for High-Dimensional Data Classification. *International Journal of Machine Learning and Cybernetics, 12*, 555-569.

Lim, N. (2007). Classification by Ensembles from Random Partitions Using Logistic Regression Models. *Dissertation.* New York: State University of New York at Stony Brook.

Lim, N., Ahn, H., Moon, H., & Chen, J. J. (2009). Classification of High-Dimensional Data with Ensemble of Logistic Regression Models. Journal of Biopharmaceutical Statistics, *20*, 160-171.

Lin, W.-J., & Chen, J. J. (2013). Class-Imbalanced Classifiers for High-Dimensional Data. Briefings in Bioinformatics, *14*(1), 13-26.

Prastyo, D. D., Savera, R. N., & Adiwibowo, D.. Corporate Financial Distress Prediction Using Statistical Extreme Value-Based Modeling and Machine Learning. *Media Statistika*, 16(1), 1–12. https://doi.org/10.14710/medstat.16.1.1-12.

Qiu, W., & Joe, H. (2020). clusterGeneration: Random Cluster Generation (with Specified Degree of Separation. *Journal of Classification*, *23*(2), 315-334.

Ray, P., Reddy, S. S., & Banerjee, T. (2021). Various Dimension Reduction Techniques for High Dimensional Data Analysis: A Review. *Artificial Intelligence Review*, *54*, 3473-3515.

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, *33*, 1-39.

Romero, C., Ventura, S., Pechenizkiy, M., & Baker, R. S. (2010). *Handbook of Educational Data Mining*. London: CRC Press.

Shu, C., & Burn, D. H. (2004). Artificial neural Network Ensembles and Their Application in Pooled Flood Frequency Analysis. *Water Resources Research*, *40*(9).

Sotiriou, C., Neo, S.-Y., McShane, L. M., Korn, E. L., Long, P. M., Jazaeri, A., & Liu, E. T. (2003). Breast Cancer Classification and Prognosis Based on Gene Expression Profiles from a Population-Based Study. *Proceedings of the National Academy of* Sciences, *100*(18), 10393-10398.

Suhartono, Faulina, R., Lusia, D. A., Otok, B. W., Sutikno, & Kuswanto, H. (2012). Ensemble Method Based on ANFIS-ARIMA for Rainfall Prediction. *Proceedings of 2012 International Conference on Statistics in Science, Business and Engineering (ICSSBE)*, 1-4.

Thudumu, S., Branch, P., Jin, J., & Singh, J. J. (2020). A Comprehensive Survey of Anomaly Detection Techniques for High Dimensional Big Data. *Journal of Big Data*, *7*(1), 42.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267-288.

Wang, W., Baladandayuthapani, V., Morris, J. S., Broom, B. M., Manyam, G., & Do, K.-A. (2013). iBAG: Integrative Bayesian Analysis of High-Dimensional Multiplatform Genomics Data. *Bioinformatics*, *29*(2), 149-159.

Widhianingsih, T. D., Kuswanto, H., & Prastyo, D. D. (2020). Logistic Regression Ensemble (LORENS) Applied to Drug Discovery. *MATEMATIKA: Malaysian Journal of Industrial and Applied Mathematics*, *36*(1), 43-49.

Xu, Y., Yu, Z., Cao, W., & Chen, C. L. (2023). A Novel Classifier Ensemble Method Based on Subspace Enhancement for High-Dimensional Data Classification. *IEEE Transactions on Knowledge and Data Engineering*, 16-30.

Zakharov, R., & Dupont, P. (2011). Ensemble Logistic Regression for Feature Selection. *Proceedings of Pattern Recognition in Bioinformatics - 6th IAPR International Conference*, *7036*, 133-144.

Zou, H., & Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, *67*(2), 301-320.