Scientific Journal of Informatics
Vol. 11, No. 4, Nov 2024
p-ISSN 2407-7658 https://journal.unnes.ac.id/journals/sji/index e-ISSN 2460-0040

Classification Performance of Stacking Ensemble with Meta-Model of Categorical Principal Component Logistic Regression on Food Insecurity Data

Dhita Elsha Pangestika[1*], Anwar Fitrianto[2], Kusman Sadik[3]
[1,2,3]Department of Statistics, Institut Pertanian Bogor, Indonesia
[2]Institute of Engineering Mathematics, Universiti Malaysia Perlis, Malaysia

Abstract.
Purpose: Stacking is one type of ensemble whose base-models use different algorithms. The classification results from its base-models are categorical and tend to be associated with each other. They then become input for the stacking meta-model. However, there are no currently definite rules for determining the classifier that becomes the meta-model in stacking. On the other hand, recent research has found that CATPCA-LR can work well on categorical predictor variables associated with each other. Therefore, this study focuses on the classification performance of the stacking algorithm with the CATPCA-LR meta-model.
Methods: The study compared the classification performance stacking with CATPCA-LR meta-model to stacking with other meta-models (random forest, gradient boost, and logistic regression) and its base-models (random forest, gradient boost, extreme gradient boost, extra trees, light gradient boost). This research used food insecurity data from March 2022.
Result: The stacking algorithm with the CATPCA-LR meta-model performs better insecurity data regarding sensitivity, balanced accuracy, F1-Score, and G-Means values. This model offers a sensitivity of 46.28%, a balanced accuracy of 59.82%, an F1-Score of 37.82%, and a G-Means of 58.26%. Meanwhile, regarding specificity values, the light gradient boost (LGB) algorithm gives the highest value compared to other algorithms. This model provides a specificity value of 88.40%. Generally, the stacking with the CATPCA-LR meta-model algorithm provides the best performance compared with other algorithms on food insecurity data.
Novelty: This research has explored a stacking classification performance with CATPCA-LR as meta-model.

Keywords: Classification, Ensemble, Food-insecurity data, Meta-model, Stacking
Received October 2024 / **Revised** January 2025 / **Accepted** February 2025

INTRODUCTION

Machine learning is a part of artificial intelligence that has been widely used, one of which is in classification modeling studies. Classification modeling is a rule to enter an object into a class or certain groups. The rule is obtained based on data from other previously known objects [1]. The ensemble method is a machine learning method commonly used in classification modeling. It combines predictions from multiple base-models to achieve better classification performance than individual base-models [2]. Ensemble methods based on the type of base-models can be divided into two types, namely non-hybrid-ensemble and hybrid-ensemble. Non-hybrid-ensemble is an ensemble method whose base-models use the same algorithm. Classification methods included in non-hybrid-ensemble methods include the random forest (RF) classification method, extra trees (ET), gradient boost (GB), extreme gradient boost (XGB), and light gradient boost (LGB). Hybrid-ensemble is an ensemble method that uses base-models with different algorithms. The classification method included in the hybrid ensemble method is a stacking ensemble learning method (stacking).

The stacking method was first introduced by Wolpert in 1992 [3]. The stacking method can combine several base-models with different algorithms [4]. This method performs learning by cross-validation of multiple base-models. Next, prediction results from several base-models are used as input variables for the combining model (meta-model). The stacking method can be used to improve model performance [4].

*Corresponding author.
Email addresses: dhita.pangestika@apps.ipb.ac.id (Pangestika)*, anwarstat@gmail.com (Fitrianto), kusmans@apps.ipb.ac.id (Sadik)
DOI: 10.15294/sji.v11i4.15315

Several studies on stacking in classification analysis have been conducted in various sectors, including finance [5], health [6], resources [7], agriculture [8], construction [9], and others.

One of the issues with the stacking method is that there is no definite rule for determining the meta-model of stacking ensemble learning as an aggregator of base-model prediction results [10]. Research related to the use of meta-model has been conducted, including research [11] and [12]. Research [11] related to stacking classification performance using GB, logistic regression (LR), RF, and deep neural network (DNN) as meta-model. The results in research [6] show that GB and LR as meta-model provide better classification results than RF and DNN as meta-model on data breast cancer. Research [12] related to classification modeling in heart disease events using RF, LR, SVM, and XGB as meta-model. The results in research [7] show that RF and LR as a meta-model provides better performance compared to SVM, and XGB as meta-model.

In stacking, the classification results of each base model are categorical and tend to be associated with each other. On the other hand, Kemalbay and Korkmazoglu [13] have conducted research related to classification performance using a combination of CATPCA and logistic regression, namely categorical principal component logistic regression (CATPCA-LR), on data with interrelated predictor categorical variables. In addition, Khikmah et al. [14] have conducted research related to comparing CATPCA-LR performance with classification performance using logistic regression on data with associated category predictor variables. The results of this study show that the CATPCA-LR classification results have better sensitivity values than logistic regression without CATPCA. Thus, based on the research results of Kemalbay and Korkmazoglu [13] and Khikmah et al. [14], using CATPCA-LR as a meta-model for stacking can obtain better classification performance.

Data with unbalanced classes (imbalanced data) are often encountered when performing classification analysis. Imbalanced data occurs when the number of one class is more prominent than other classes (minorities). The imbalanced class can affect classification performance because classification results tend to fall into the majority class. There are two main approaches to dealing with imbalanced data conditions on the data level: undersampling and oversampling methods [15]. The undersampling method balances minority classes by removing the majority of the classes until the distribution is balanced [15]. While using the oversampling method, replication of minority class data is carried out randomly and repeatedly until the observations have the same quantity as the majority class data. The oversampling method has higher accuracy and sensitivity values than the undersampling method [16]. However, oversampling methods tend to occur overfitting because data generation is carried out using existing data previously. Chawla et al. have developed the SMOTE method to overcome the weaknesses in the oversampling method by generating synthetic data based on k-nearest neighbors [17].

On the other hand, one of Sustainable Development Goals (SDGs) priorities is addressing food insecurity. Food insecurity is a condition that occurs when individuals lack access to safe and nutritious food in sufficient quantities for normal growth and also an active and healthy life [18]. One government effort to overcome food insecurity is the provision of social assistance [19]. Utilizing classification models can facilitate the identification of food-insecure households, ensuring targeted delivery of social assistance. Food insecurity incidents can be predicted using specific predictors. However, food insecurity data often faces the issue of class imbalance [20], where the majority class dominates, and the minority class becomes difficult to predict. Considering the effectiveness of the SMOTE method, this study applies SMOTE to address class imbalance in food insecurity data.

This research aims to compare the classification performance of stacking algorithms with CATPCA-LR meta-model against other meta-models (such as LR, GB, and RF) and its base-models (RF, GB, XGB, ET, and LGB). This study uses food insecurity data from the March 2022 Susenas results in Indonesia to analyze the incidence of food-insecure households.

METHODS

Data

The data used in this research was food insecurity data from Susenas data for March conducted by Badan Pusat Statistik (BPS) in 2022. Food insecurity status in households is the response variable (Y) in this research, which consists of the "food insecurity households" category and the "food secure households" category. The predictor variables used in this research can be presented in Table 1. The predictor variables

refer to the research by Dharmawan et al. in 2022 [20]. The empirical data utilized in this study involved a three-step data preparation process. The first step involved integrating or merging the data on household-level units and the data on individual-level units. Second, eliminating observations that exhibit empty values (missing data), are coded "refuse to answer," or are coded "don't know" across eight questions from the Food Insecurity Experience Scale (FIES). Lastly, the data is categorized into "food insecurity households" and "food secure households," which serve as the response variables. A household is categorized into "food insecurity households" whether there is a "Yes" answer across eight questions from FIES.

Table 1. List of predictor variables to be used

Group	Characteristic Variables		Characteristics Measurement	Description
Characteristics of Head of Household	X_1	*Education of household head*	Ordinal	1 : Not in school/ did not finish elementary school (SD) 2 : Elementary school (SD) 3 : Junior high school (SMP) 4 : Senior high school (SMA) 5 : College
	X_2	*Vulnerable households head*	Nominal	0 : No 1 : Yes
Characteristics of Social and Economic Capital	X_3	*Main income from the transferee*	Nominal	0 : No 1 : Yes
	X_4	*Ownership of land assets*	Nominal	0 : No 1 : Yes
	X_5	*Internet access*	Nominal	0 : No 1 : Yes
	X_6	*Access to outpatient treatment*	Nominal	0 : No 1 : Yes
	X_7	*Number of household members having savings accounts*	Ordinal	0 : There aren't any 1 : 1 2 : 2 or more
	X_8	*Number of household members illiterate*	Ordinal	0 : There aren't any 1 : 1 2 : 2 or more
Characteristics of Social Protection Program	X_9	*Grantee of Hopeful Family Program (PKH)*	Nominal	0 : No 1 : Yes
	X_{10}	*Grantee of Prosperous Family Program (KKS)*	Nominal	0 : No 1 : Yes
	X_{11}	*Grantee of Social Assistance from Local Government*	Nominal	0 : No 1 : Yes
	X_{12}	*Grantee of Health Insurance National Program (BPJS)*	Nominal	0 : No 1 : Yes
	X_{13}	*Grantee of Health Insurance Local Program (Jamkesda)*	Nominal	0 : No 1 : Yes
	X_{14}	*Grantee of Scholarship Social Program (KIP/PIP)*	Nominal	0 : No 1 : Yes
	X_{15}	*Grantee of Non-Cash Social Assistance (BPNT)*	Nominal	0 : No 1 : Yes
Characteristics of House Condition	X_{16}	*Type of house roof*	Nominal	1 : Others 2 : Asbestos 3 : Zinc 4 : Concrete 5 : Tile
	X_{17}	*Type of house floor*	Nominal	1 : Others 2 : Tile/ terazzo 3 : Cement/ red brick 4 : Parquet/ vinyl/ carpet 5 : Marble/ ceramic
	X_{18}	*Types of house walls*	Nominal	1 : Others 2 : Log 3 : Bricks/ woven bamboo/ plastering of woven bamboo
	X_{19}	*Source of house lighting*	Nominal	0 : Non-electricity 1 : With electricity
	X_{20}	*House size*	Ordinal	1 : < 50 2 : 50 - 100 3 : 101 - 200 4 : > 200

Group	Characteristic Variables		Characteristics Measurement	Description
Characteristics of Home Activities Support	X_{21}	*Type of cooking fuel*	Nominal	0 : Not cooking in the house 1 : Others 2 : Firewood 3 : Kerosene 4 : LPG 3 kg 5 : LPG 5.5 kg/ LPG 12 kg/ bluegaz
	X_{22}	*Source of drinking water*	Nominal	1 : Others 2 : Drilling well/ pump 3 : Tap water 4 : Branded bottled/ refill water
	X_{23}	*Decent drinking water*	Nominal	0 : No 1 : Yes
	X_{24}	*Decent sanitation*	Nominal	0 : No 1 : Yes

Feature selection with chi-square test

The study [21] stated that selecting predictor variables can improve model performance. In addition, the study's result showed that the chi-square test was an effective method for selecting predictor variables. The selection of predictor variables using the chi-square method was carried out by testing the significance of the relationship between a predictor variable and the response variable. The null hypothesis in the chi-square test is that the feature being tested does not have a significant association with the response variable. Meanwhile, the alternative hypothesis of the chi-square test is that the feature being tested has a significant association with the response variable. The formula for the chi-square statistic is as follows:

$$\chi^2 = \sum_{i=1}^{b} \sum_{j=1}^{k} \frac{(n_{ij} - \mu_{ij})^2}{\mu_{ij}} \quad (1)$$

b is the number of rows (number of categories) in the first variable, and k is the number of columns (number of categories) in the second variable. n_ij is the observation frequency value of the cell in the i-th row and j-th column. Meanwhile, the value of μ_ij is the expected value of the cell in the i-th row and j-th column. The statistical value of χ^2, greater than the values of χ^2 of the table with degrees of freedom (b-1) (k-1), identifies the rejection of the null hypothesis. The test results with the null hypothesis rejected can be interpreted that there was a significant association between the predictor variable and the response variable. Conclusions in the chi-square test can also be drawn through a comparison of the p-value and alpha (α). If the p-value is smaller than α, the null hypothesis is rejected.

Synthetic minority oversampling technique (SMOTE)

Imbalanced class data occurs when the number of one class is much lower than the other class [22]. The imbalanced class can have a negative impact on the classification results because the classification results tend to fall into the majority class. SMOTE (Synthetic Minority Oversampling Technique) handles data imbalance by generating artificial or synthetic data based on the k-nearest neighbors between minority classes [17]. Synthetic data generation from SMOTE for categorical variables uses the modified Value Difference Metric (VDM) formula as follows:

$$\Delta(X, Y) = w_x w_y \sum_{i=1}^{m} \delta(x_i, y_i)^r \quad (2)$$

with Δ(X, Y) is the distance between observations x and y, w_xis the weight of observation x, w_y is the weight of observation y, m is the number of predictors, r has a value of 1 to produce the Manhattan distance and has a value of 2 to produce the Euclidean distance δ(x_i,y_i) which is the distance between observations x and y on the i-th variable calculated using the following equation:

$$\delta(x_i, y_i) = \sum_{j=1}^{k} \left| \frac{C_{xj}}{C_x} - \frac{C_{yj}}{C_y} \right|^d \quad (3)$$

Where k is the number of classes in the response variable, C_xjis the number of categories x in the jth class, C_yjis the number of categories y in the jth class, and d is a constant with a value of 1.

Random forest

Random Forest (RF) is an improvement of the Classification and Regression Tree (CART) method by applying the Bootstrap Aggregating (Bagging) method and random feature (predictor variable) selections [23]. The following are the stages of preparation and estimation using random forest on training data of size N observations and p predictor variables [24]:

1. a. Bootstrap Stages, draw a random sample with recovery of size *n* from *N* observations.

b. Random Sub-setting stages, compile a tree based on data from the bootstrap stage results. During the splitting process, predictor variables are selected randomly with $d < p$.
c. Repeat steps *a* until *b* in *k* times to obtain *k* random trees.

2. Carry out a combined estimate based on *k* trees using majority voting in classification and average in regression.

Extremely randomized trees (extra trees)

Extra Trees was developed by Geurts et al. [25] with the idea of building an ensemble of many trees with large differences between trees by adding a randomization process to the tree formation algorithm. Three essential things in forming trees in Extra Trees are [1] :

1. A classification tree is formed using all the training data as the root node.
2. The selection of sample predictor variables as node separators is selected from the best of a subset of predictor variables, not the best of all predictors.
3. The best node separator candidate for a selected predictor variable is done randomly.

Gradient boost

Gradient boost is an improvement algorithm from a tree-boost algorithm developed by Friedman in 2001 [26]. The following are the stages of the Gradient Boost algorithm

1. Prepare input in the form of training data, loss function, learning rate (v), and number of iterations (M)
2. Stage 1: Create an initial model, namely $F_0(x)$, those obtained from:
$$F_0(x) = log\ (\frac{p}{1-p}) \quad (4)$$
3. Stage 2: for m =1 to M, do the following:
   a. Calculating r_{im}or pseudo residual for each i =1, …, n with formula
   $$r_{im} = y - p \quad (5)$$
   b. Construct a regression tree model using r_{im} as response variables and $\boldsymbol{x_i}$ as a predictor variable, and obtain as many as J_m as the final node.
   c. For each final node j (with j =1, …, J_m) from step 3.b., the calculation is then carried out:
   $$\gamma_{jm} = \frac{\sum_{i=1}^{n*}(y-p_i)}{\sum_{i=1}^{n*} p_i(y-p_i)} \quad (6)$$
   d. The sign n* indicates the addition of only observations at the node or subset R_{jm}.
   e. Do updates
   $$F_{m-1}(x) + v \sum_{j=1}^{J_m} \gamma_{jm} I\ (x \in R_{jm}) \quad (7)$$
4. Stage 3: Output $F_m(\boldsymbol{x})$

Extreme gradient boost (XGboost)

Chen and Guestrin [27] developed an ensemble technique method from gradient boosting to extreme gradient boost. This algorithm's basic concept is to adjust the learning parameters to reduce the loss function.

$$obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)}\right) + \Omega\ (f_t) \quad (8)$$

Where $\sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)}\right)$ is the loss function, and $\Omega(f_t)$ shows the regularization function. This algorithm introduces a regularization term in the objective function to prevent overfitting.

Light gradient boost machine (LightGBM)

LightGBM was first introduced by Ke et al. [28]. This algorithm focuses on model efficiency and performance. LightGBM was developed to increase efficiency, handle large data sets, and reduce training time [29]. LightGBM uses two strategies: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). LightGBM utilizes the Gradient-based One-Side Sampling (GOSS) technique to reduce the amount of data used during the training process. In simple terms, GOSS randomly selects samples with high gradients and discards samples with small gradients. GOSS assumes that data with small gradients have lower errors and are considered well-trained. In addition, the variance gain calculation is carried out by adding weights or constant multipliers to data with small gradients. This technique helps reduce the computational burden and speeds up the training process [28]. In addition, LightGBM also uses Exclusive Feature Bundling (EFB) to combine several variables/features into an exclusive bundle. This technique helps reduce the dimensionality of features and overcome data sparsity [30]. The combination of these concepts makes LightGBM one of the most powerful and efficient ensemble algorithms in machine learning.

The k-fold cross-validation

The *k*-fold cross-validation is a process used to measure model performance. The *k*-fold cross-validation technique to evaluate model performance can be done with the following stages [1]:

1. Perform a random division of the dataset (*D*) into k mutually exclusive parts to produce *D1*, *D2*, …, *Dk*.
2. For *j*=1 to *j*=*k*, the following steps are carried out:
   a. Define *Dj* as testing data
   b. Define *D* - *Dj* as training data
   c. Perform modeling with training data (*D*-*Dj*)
   d. Using the model in step 2.c. to predict the testing data (*Dj*)
3. After stage 2, the class prediction is obtained for all observations. Next, a comparison is made between the results of the class prediction and the actual class using various specific measures. These measures include sensitivity, specificity, balanced accuracy, and so on.

Categorical principal component analysis (CATPCA)

Principal Component Analysis (PCA) is a method commonly used to reduce the dimensions of m correlated variables to p uncorrelated components able to explain the diversity of data, where $p \leq m$. However, PCA has two limitations. First, the relationship between variables is assumed to be linear. Second, the variables are on numerical scales. Alternatively, CATPCA can be used to deal with categorical variables associated with each other [13] [31]. CATPCA is a method used on categorical scale data using optimal scaling, which converts categorical labels into numerical values by maximizing the diversity between variables [32]. There are *n* individuals and *m* variables given by *n* x *m* observations score matrix $\boldsymbol{H}$ where each variable is defined by $\boldsymbol{h_j}$, j =1, 2, …, *m* where *j* is a column of the matrix $\boldsymbol{H}$. If the variable $\boldsymbol{h_j}$ is a nominal or ordinal measurement scale, so a linear transformation of the optimal scale is observed in each score by converting it into a categorical quantification by:

$$\boldsymbol{q_j}=\varphi_j\ (\boldsymbol{h_j}) \tag{9}$$

$\boldsymbol{Q}$ is a matrix of category quantification. Let $\boldsymbol{X}$ be a matrix of *n* x *p* objects, which are the values of the individual values of the principal components obtained by CATPCA. For example, $\boldsymbol{A}$ is an *m* x *p* matrix of component loadings where column *j* is denoted by $\boldsymbol{a_j}$. Then, the loss function (L ($\boldsymbol{Q, A, X}$)) to minimize the original data and principal components can be given as follows:

$$L\ (\boldsymbol{Q, A, X}\ ) = n^{-1} \sum_{j=1}^{m} tr\ \left(\boldsymbol{q_j a_j^T - X}\ \right)^T (\boldsymbol{q_j a_j^T - X}\ ) \tag{10}$$

Stacking ensemble learning (stacking)

Stacking ensemble learning is an ensemble method consisting of several base-models that carry out learning independently. The prediction results from the base-models become input for learning by the meta-model, which produces the final prediction. Furthermore, Zhao et al. [33] adjusted the stacking concept to reduce the risk of overfitting. The idea proposed by Zhao et al. [33] can be presented in Figure 1.

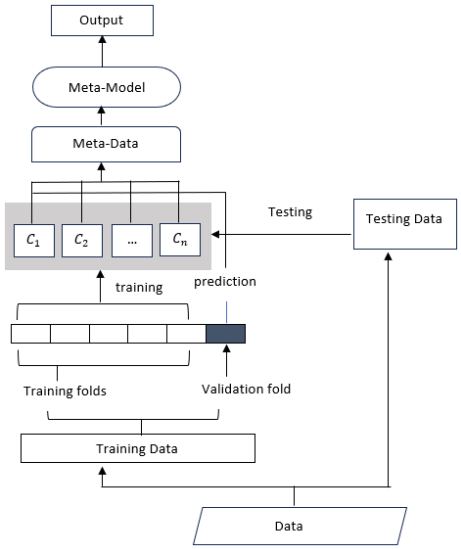

Figure 1. Illustration of the stacking method of Zhao et al. (2020) [33]

Classification modelling

The first step in this research is to divide the data into two parts, namely, 30% test data and 70% training data. Next, treatment is carried out on the training data, handling unbalanced data using SMOTE. SMOTE increases the minority class data point by generating synthetic data based on k-nearest neighbors. The generating synthetic data for the minority class uses five-nearest neighbors to determine the random samples, resulting in the same percentage of data points (50:50) in the SMOTE data as in the majority class or the new training data.

After processing the unbalanced data treatment using SMOTE, the training data is divided into two parts, namely 30% for validation data and 70% of the data used to build the base-models using 5-fold cross-validation. The base-models used in this research are RF, ET, GB, LGB, and XGB. Next, validation data predictions are carried out using the built base-models. The prediction results from the validation data will be used as input for learning using the meta-model.

This research examined the classification performance of the stacking method with the CATPCA-LR meta-model. The classification performance will be compared with the performance of the base-models (namely RF, ET, GB, LGB, and XGB). This research will also compare stacking methods with other meta-models, such as stacking with logistic regression meta-model (stacking_meta_LR), stacking with gradient boost meta-model (stacking_meta_GB), and stacking with random forest meta-model (stacking_meta_RF). The evaluation measures used to measure performance are sensitivity, specificity, balanced accuracy, F1-Score, and G-Means value. The combination of machine learning algorithms used as the base-models and meta-model in this research can be presented in Table 2.

Table 2. Stacking scenarios based on the meta-model used

Base-models	Meta-model	Stacking Name
random forest (RF), xgboost (XGB), lightgb (LGB), extra-trees (ET), gradient boost (GB)	Logistic regression (LR)	stacking_meta_LR
	Categorical principal component logistic regression (CATPCA-LR)	stacking_meta_CATPCA
	Gradient boost (GB)	stacking_meta_GB
	Random forest (RF)	stacking_meta_RF

Classification model evaluation

The method commonly used to describe the performance of classification models is the confusion matrix [34]. Confusion matrix is a cross-tabulation between response variable data that falls into the prediction and observation/actual classes. The confusion matrix for cases with two classes of response variables is as follows:

The method commonly used to describe the performance of classification models is the confusion matrix [34]. Confusion matrix is a cross-tabulation between response variable data that falls into the prediction and observation/actual classes. The confusion matrix for cases with two classes of response variables is as follows:

Table 3. Confusion matrix

Actual	Predictions	
	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

The performance of a classification model can be measured through several values, including sensitivity, specificity, balanced accuracy, F1-Score, and G-Means.

1. Sensitivity
   Sensitivity is the correctness of predicting the positive class compared to the entire positive class [35]. The sensitivity value can be obtained using the formula [36]:
   $$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \quad (11)$$
2. Specificity
   Specificity is the correctness of predicting the negative class compared to the entire negative class [35]. The specificity value can be obtained using the formula [36]:
   $$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \quad (12)$$
3. F1-Score
   F1-Score is the harmonic mean of sensitivity and precision values [37].

$$\text{F1-Score} = 2 \times \left[\frac{\text{Sensitivity} \times \text{Precision}}{\text{Sensitivity} + \text{Precision}}\right] \quad (13)$$

where precision: $\frac{TP}{TP+FP}$ (14)

4. Balanced accuracy
   Balanced accuracy is an accuracy metric for unbalanced samples [38]. Another definition of balanced accuracy is the arithmetic mean of sensitivity and specificity values [39]

$$\text{Balanced accuracy} = \left[\frac{\text{Sensitivity} + \text{Specificity}}{2}\right] \quad (15)$$

5. G-Means
   G-Means is the geometric mean of sensitivity and specificity values [38].

$$\text{G-Means} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \quad (16)$$

Flowchart analysis

The flowchart's input is raw data from BPS. The data analysis begins with data pre-processing as the first step. This step aims to process raw data into data that is ready for analysis in the next stage. Furthermore, data exploration and selection of predictor variables are carried out. The data exploration step aims to explore the data conditions, and selecting predictor variables aims to improve model performance. The dataset is then divided into training data (70%) for model training and test data (30%) using fix random state. Then, SMOTE is applied to the training data, further divided into training folds (70%) and validation folds (30%). A predictive model is developed using the training dataset. The evaluation of its performance is based on metrics such as sensitivity, specificity, balanced accuracy, F1-Score, and G-Means on the testing data. This process is repeated for 30 sampling strategies (validation replications) to ensure reliability. The average performance across the 30 replications is calculated for each algorithm. Next, the comparative analysis uses the Anova test and Tukey to identify the best method with significant performance at $\alpha = 5\%$. Finally, the process concludes with analysis and interpretation as the output.

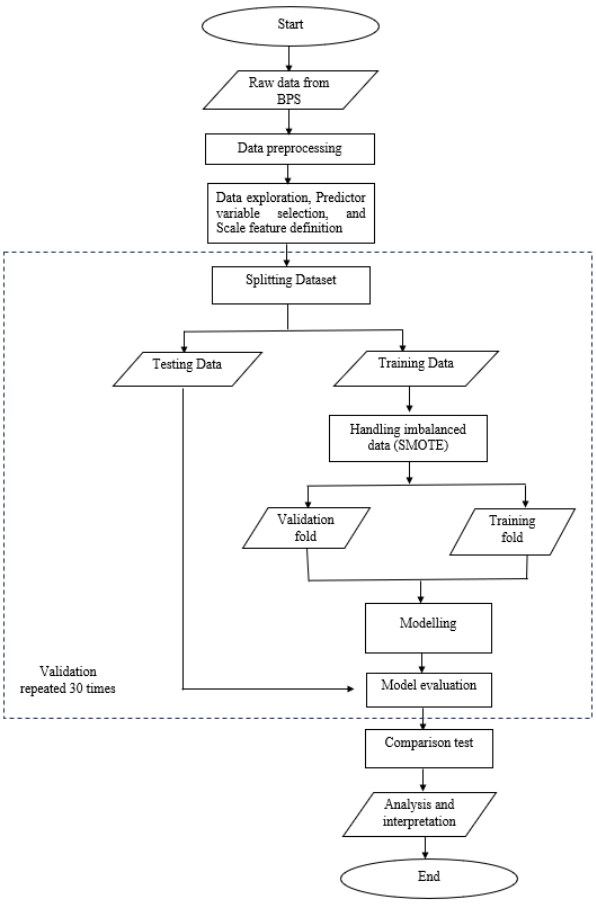

Figure 2. Flowchart of empirical data analysis procedure

RESULTS AND DISCUSSIONS

Data preprocessing result

The data used in this study were obtained from the National Socio-Economic Survey (Susenas) of BPS in 2022. the Susenas data consists of data on household-level units and individual-level units. In this study, the number of predictors from the household-level units consists of 18 variables, including the education of the household head ($X1$), vulnerable household head ($X2$), main income from the transferee ($X3$), ownership of land assets ($X4$), internet access ($X5$), grantee of Hopeful Family Program/PKH ($X9$), grantee of Prosperous Family Program/KKS ($X10$), grantee of Social Assistance from Local Government ($X11$), grantee of Non-Cash Social Assistance/BPNT ($X15$), type of house roof ($X16$), type of house floor ($X17$), type of house walls ($X18$), source of house lighting ($X19$), house size ($X20$), type of cooking fuel ($X21$), source of drinking water ($X22$), decent drinking water ($X23$), and decent sanitation ($X24$). The other six predictors are derived from individual-level units, including access to outpatient treatment ($X6$), number of household members having saving accounts ($X7$), number of household members illiterate ($X8$), grantee of Health Insurance National Program/BPJS ($X12$), grantee of Health Insurance Local Program/Jamkesda ($X13$), and grantee of Scholarship Social Program/KIP/PIP ($X14$). Then, both unit data are merged into the complete dataset.

Next, the observations with missing values or responses coded as “Refuse to answer” or “Don't know” across the eight FIES questions are excluded. In this study, a total of 3,860 households provided such responses to at least one question. Therefore, excluding missing values had reduced the dataset to 335,724 households from the initial 339,584. Finally, the dataset was preprocessed and ready for the next analysis.

Exploration of empirical data (data on food insecurity in Indonesia)

The percentage of households based on food insecurity status is shown in Figure 3. The percentage of food insecure households out of all households that are respondents to the 2022 Susenas was 21.35%, while the percentage of households not food insecure was 78.65%. The comparison of the percentage composition of the response variable classes that are not equal indicates a case of data with imbalanced classes [40]. Thus, food insecurity data was a case of data with imbalanced classes.

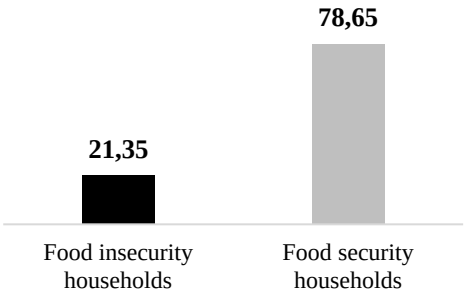

Figure 3. Percentage of households based on food insecurity status (%)

Predictor variable selection

Rahmadeyan and Mustakim [21] stated that predictor variable selection can improve model performance. That study concluded that the chi-square test is an effective predictor variable selection method. Therefore, this study selects predictor variables using the chi-square test before classification analysis. The results of the chi-square test are shown in Table 4. The result of selecting predictor variables using the chi-square test showed that the p-value of one predictor variable, namely, the grantee of Health Insurance Local Program/Jamkesda (X13), was more than α=5%. It can be interpreted that the grantee of the Health Insurance Local Program (Jamkesda) variable does not have a significant association with the response variable (food insecurity status) at α=5%. This result relates to the existence of a program integration policy for Jamkesda into the National Health Insurance (JKN) program under the National Guarantee Council Regulation Number 01 of 2016. The implementation of this policy means that some regions in Indonesia have integrated the Jamkesda program into the National Health Insurance program, namely BPJS. Thus, the grantee of the Health Insurance Local Program (Jamkesda) variable is excluded from the modeling process.

Table 4. Predictor variable selection results using the chi-square test

Variable	p-value	Variable	p-value
X_1	$< 2.200 \times 10^{-16}$	X_{13}	6.827×10^{-2}
X_2	$< 2.200 \times 10^{-16}$	X_{14}	$< 2.200 \times 10^{-16}$
X_3	$< 2.200 \times 10^{-16}$	X_{15}	$< 2.200 \times 10^{-16}$
X_4	$< 2.200 \times 10^{-16}$	X_{16}	$< 2.200 \times 10^{-16}$
X_5	$< 2.200 \times 10^{-16}$	X_{17}	$< 2.200 \times 10^{-16}$
X_6	$< 2.200 \times 10^{-16}$	X_{18}	$< 2.200 \times 10^{-16}$
X_7	$< 2.200 \times 10^{-16}$	X_{19}	$< 2.200 \times 10^{-16}$
X_8	$< 2.200 \times 10^{-16}$	X_{20}	$< 2.200 \times 10^{-16}$
X_9	$< 2.200 \times 10^{-16}$	X_{21}	$< 2.200 \times 10^{-16}$
X_{10}	$< 2.200 \times 10^{-16}$	X_{22}	$< 2.200 \times 10^{-16}$
X_{11}	$< 2.200 \times 10^{-16}$	X_{23}	$< 2.200 \times 10^{-16}$
X_{12}	$< 2.200 \times 10^{-16}$	X_{24}	$< 2.200 \times 10^{-16}$

Algorithm classification performance results on empirical data

The average modeling results on 30 replication validations from nine algorithms are summarized in Table 5. Each algorithm was run 30 times, and the performance metrics were averaged. The stacking algorithm with the CATPCA-LR meta-model achieved the highest values for sensitivity (46.28%), balanced accuracy (59.82%), F1-Score (37.82%), and G-Means (58.26%). In contrast, the LGB algorithm provides the highest specificity at 88.40%. All metrics represent the average performance across 30 replication validations.

Table 5. Classification performance results based on the learning algorithm (30 replication validations)

Algorithm	Average				
	Sensitivity	Specificity	Balanced Accuracy	F1-Score	G-Means
Stacking _meta_Reglog	31.72	82.70	57.21	32.43	51.22
Stacking _meta_GB	30.31	82.98	56.65	31.37	50.15
Stacking _meta_RF	30.22	83.00	56.61	31.30	50.07
Stacking_meta_CATPCA	46.28*	73.36	59.82*	37.82*	58.26*
XGB	30.22	81.96	56.09	31.54	49.76
RF	35.29	80.09	57.69	33.79	53.16
LGB	26.06	88.40*	57.23	30.86	48.00
ET	31.40	81.86	56.63	31.65	50.70
GB	32.32	79.95	56.13	31.31	50.83

A more in-depth study regarding the influence of method use on performance classification can be determined using the ANOVA test. This test aims to determine whether the influence of the algorithm causes significant differences in model performance measurements. The null hypothesis (H_0) in this test is there are no significant differences in the value of the model performance measures between one another algorithm and other algorithms. The results of the ANOVA test in Table 6 showed that the p-value was smaller than α=5%, so the decision was to reject (H_0). In this matter, it can be interpreted that at least one pair of algorithms has significant differences. Based on ANOVA's results, the Tukey test was then performed to test differences in performance measures based on pairs of algorithms.

Table 6. ANOVA test results

Performance measures	p-value	Results
Specificity	$< 2.200 \times 10^{-16}$	Significant. There are differences in the specificity values between algorithms
Sensitivity	$< 2.200 \times 10^{-16}$	Significant. There are differences in the sensitivity values between algorithms
Balanced accuracy	$< 2.200 \times 10^{-16}$	Significant. There are differences in the balanced accuracy values between algorithms
F1-Score	$< 2.200 \times 10^{-16}$	Significant. There are differences in the F1-Score values between algorithms
G-Means	$< 2.200 \times 10^{-16}$	Significant. There are differences in the G-Means values between algorithms

Tukey's test results are visualized in Figure 4 (specificity) and Figure 5 (sensitivity, balanced accuracy, F1-Score, and G-Means). The algorithm names in the figures are arranged based on performance metrics from the highest (left) to the lowest (right). Groups with similar letter labels indicate no significant difference between their values.

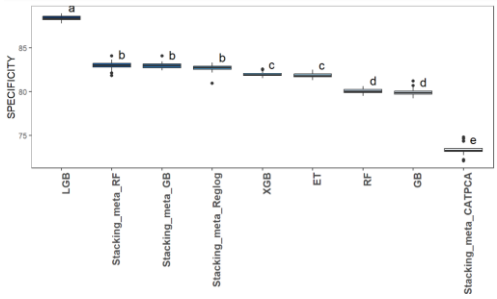

Figure 4. The result and visualization of Tukey's test on specificity values

After condcuting the Tukey test results, the performance of the LGB model was significantly higher than that of other models regarding specificity at α=5%. LGB is a form of gradient boosting that effectively addresses the challenges faced by big data using strategies such as the leaf-wise tree growth method and histogram-based decision tree algorithm [41]. Several studies have also shown that LGB has a higher specificity value than other algorithms, such as XGB [42], RF [43], GB [44], and ET [44].

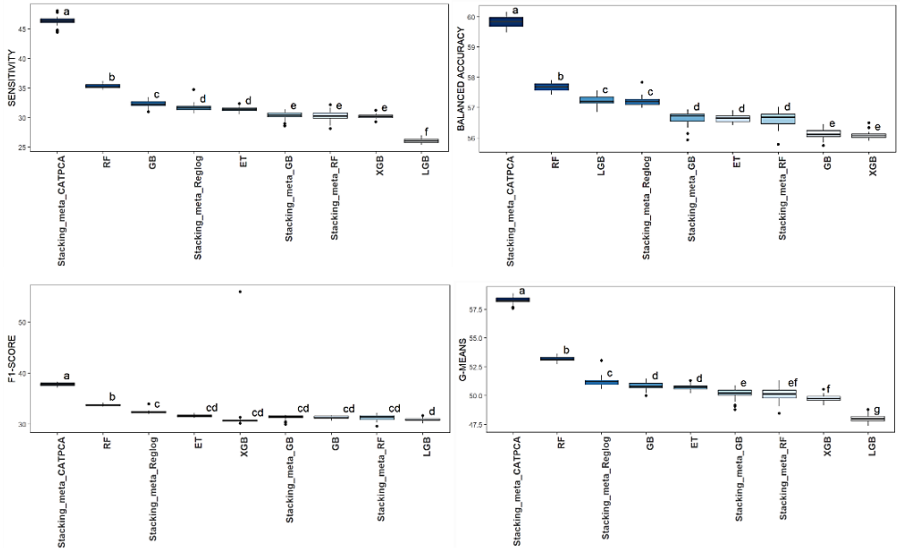

Figure 5 The result and visualization of Tukey's test on sensitivity, balanced accuracy, F1-Score, and G-Means values

On the other hand, stacking with the CATPCA-LR meta-model indicated significantly higher performance than other models regarding sensitivity, balanced accuracy, F1-Score, and G-Means value at α=5%. This result can be explained by the association of base-model prediction results. Based on recent research [14], CATPCA-LR can provide a good sensitivity value on predictor variables associated with each other. The choice of CATPCA-LR as a meta-model can accommodate this association of the base-models prediction results to provide good sensitivity. A high sensitivity value in CATPCA-LR can also improve the balanced accuracy, F1-Score, and G-Means values. Therefore, using CATPCA-LR as a meta-model in stacking ensemble learning can be an alternative to enhance the performance of sensitivity, balanced accuracy, F1-Score, and G-Means values.

Meanwhile, based on a study [45], the sensitivity evaluation measure is essential in the study of food insecurity because the sensitivity evaluation measure can help the government focus more on monitoring and providing appropriate assistance for food-insecure households. So, the consideration of model selection based on sensitivity values is expected to produce more accurate solutions and following government needs. Thus, when compared overall to the model evaluation measure and the sensitivity evaluation measure, the model that provides the best results in the study of food insecurity comes from the stacking algorithm with the CATPCA-LR meta-model with the SMOTE technique.

CONCLUSION

Stacking is an ensemble method where base models with different algorithms yield interrelated categorical classification results, which are then used as input for the meta-model. Currently, there is no definitive rule for determining the meta-model of stacking ensemble learning to aggregate the prediction of base-models. The research on determining meta-models is needed to achieve the best classification performance.

On the other hand, recent studies have found that CATPCA-LR performs well on categorical predictor variables. Selecting the CATPCA-LR as the meta-model in stacking offers the potential to enhance classification performance. Therefore, this study focuses on evaluating the classification performance of the stacking algorithm with the CATPCA-LR meta-model.

The study compared the classification performance stacking with the CATPCA-LR meta-model to stacking with other meta-models (random forest, gradient boost, and logistic regression) and its base-models (random forest, gradient boost, extreme gradient boost, extra trees, light gradient boost). This research uses food insecurity data in March 2022. Regarding balanced accuracy, sensitivity, F1-Score, and G-Means values, the stacking algorithm with the CATPCA-LR meta-model provides the highest value compared to other algorithms. This model presents a sensitivity value of 46.28%, a balanced accuracy value of 59.82%, an F1-Score of 37.82%, and a G-Means of 58.26%. On the contrary, the LGB algorithm provides better performance regarding specificity. This model provides a specificity value of 88.40%. Generally, the stacking with the CATPCA-LR meta-model algorithm provides the best performance compared with other algorithms on food insecurity data.

This research contributes to the field of classification in machine learning by applying stacking with the CATPCA-LR meta-model. Based on this study's findings, the recommendation for future studies is to investigate the classification performance of stacking with the CATPCA-LR meta-model using different base-models to evaluate its applicability.

REFERENCES

[1] B. Sartono dan H. Dharmawan, "Pemodelan Prediktif Berbasis Pohon Klasifikasi," in *PT Penerbit IPB Press*, 2023.

[2] I. D. Mienye dan Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," *IEEE Access*, vol. 10, no. September, hal. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.

[3] Z. Faska, L. Khrissi, K. Haddouch, dan N. El Akkad, "A robust and consistent stack generalized ensemble-learning framework for image segmentation," *J. Eng. Appl. Sci.*, vol. 70, no. 1, hal. 1–20, 2023, doi: 10.1186/s44147-023-00226-4.

[4] A. I. Naimi dan L. B. Balzer, "Stacked generalization: an introduction to super learning," *Eur J Epidemiol*, vol. 33, no. 5, hal. 459–464, 2018, doi: 10.1007/s10654-018-0390-z.

[5] Z. Zhang, Y. Ma, dan Y. Hua, "Financial Fraud Identification Based on Stacking Ensemble Learning Algorithm: Introducing MD&A Text Information," *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/1780834.

[6] H. V. Nguyen dan H. Byeon, "Prediction of Parkinson's Disease Depression Using LIME-Based Stacking Ensemble Model," *Mathematics*, vol. 11, no. 3, 2023, doi: 10.3390/math11030708.

[7] E. Sarmas, E. Spiliotis, V. Marinakis, T. Koutselis, dan H. Doukas, "A meta-learning classification model for supporting decisions on energy efficiency investments," *Energy Build.*, vol. 258, hal. 111836, 2022, doi: 10.1016/j.enbuild.2022.111836.

[8] H. Li, Y. Jin, J. Zhong, dan R. Zhao, "A Fruit Tree Disease Diagnosis Model Based on Stacking Ensemble Learning," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/6868592.

[9] Q. Li dan Z. Song, "Ensemble-Learning-Based Prediction of Steel Bridge Deck Defect Condition," *Appl. Sci.*, vol. 12, no. 11, 2022, doi: 10.3390/app12115442.

[10] N. Kardani, A. Zhou, M. Nazem, dan S. L. Shen, "Improved prediction of slope stability using a hybrid stacking ensemble method based on finite element analysis and field data," *J. Rock Mech. Geotech. Eng.*, vol. 13, no. 1, hal. 188–201, 2021, doi: 10.1016/j.jrmge.2020.05.011.

[11] H. Kwon, J. Park, dan Y. Lee, "Stacking ensemble technique for classifying breast cancer," *Healthc. Inform. Res.*, vol. 25, no. 4, hal. 283–288, 2019, doi: 10.4258/hir.2019.25.4.283.

[12] T. Yoon dan D. Kang, "Multi-Modal Stacking Ensemble for the Diagnosis of Cardiovascular Diseases," *J. Pers. Med.*, vol. 13, no. 2, 2023, doi: 10.3390/jpm13020373.

[13] G. Kemalbay dan Ö. B. Korkmazoğlu, "Categorical Principal Component Logistic Regression: A

Case Study for Housing Loan Approval," *Procedia - Soc. Behav. Sci.*, vol. 109, hal. 730–736, 2014, doi: 10.1016/j.sbspro.2013.12.537.
[14] L. Khikmah, H. Wijayanto, dan U. D. Syafitri, "Modeling Governance KB with CATPCA to Overcome Multicollinearity in the Logistic Regression," *J. Phys. Conf. Ser.*, vol. 755, no. 1, 2016, doi: 10.1088/1742-6596/755/1/011001.
[15] J. Hoyos-Osorio, A. Alvarez-Meza, G. Daza-Santacoloma, A. Orozco-Gutierrez, dan G. Castellanos-Dominguez, "Relevant information undersampling to support imbalanced data classification," *Neurocomputing*, vol. 436, hal. 136–146, 2021, doi: https://doi.org/10.1016/j.neucom.2021.01.033.
[16] T. Wongvorachan, S. He, dan O. Bulut, "A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining," *Inf.*, vol. 14, no. 1, 2023, doi: 10.3390/info14010054.
[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, dan W. P. Kegelmeyer, "SMOTE: Syntetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, no. Sept. 28, hal. 321–357, 2002.
[18] FAO, IFAD, dan WFP, *The State of Food Insecurity in the World*, vol. 22, no. 6. 2013.
[19] Badan Ketahanan Pangan, "Panduan Penyusunan Peta Ketahanan dan Kerentanan Pangan," *Bkp*, hal. 1–77, 2020.
[20] H. Dharmawan, B. Sartono, A. Kurnia, A. F. Hadi, dan E. Ramadhani, "a Study of Machine Learning Algorithms To Measure the Feature Importance in Class-Imbalance Data of Food Insecurity Cases in Indonesia," *Commun. Math. Biol. Neurosci.*, vol. 2022, hal. 1–25, 2022, doi: 10.28919/cmbn/7636.
[21] A. Rahmadeyan dan M. Mustakim, "Seleksi Fitur pada Supervised Learning: Klasifikasi Prestasi Belajar Mahasiswa Saat dan Pasca Pandemi COVID-19," *J. Nas. Teknol. dan Sist. Inf.*, vol. 9, no. 1, hal. 21–32, 2023, doi: 10.25077/teknosi.v9i1.2023.21-32.
[22] A. A. Khan, O. Chaudhari, dan R. Chandra, "A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation," *Expert Syst. Appl.*, vol. 244, no. November 2023, 2024, doi: 10.1016/j.eswa.2023.122778.
[23] L. Breiman, "Random Forests," 2001.
[24] B. Sartono dan U. Syafitri, "Metode pohon gabungan: Solusi pilihan untuk mengatasi kelemahan pohon regresi dan klasifikasi tunggal," *ICSA - Int. Conf. Stat. Anal. 2019*, vol. 15, no. 1, 2010.
[25] P. Geurts, D. Ernst, dan L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, hal. 3–42, Apr 2006, doi: 10.1007/s10994-006-6226-1.
[26] A. Natekin dan A. Knoll, "Gradient boosting machines, a tutorial," *Front. Neurorobot.*, vol. 7, no. DEC, 2013, doi: 10.3389/fnbot.2013.00021.
[27] T. Chen dan C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Agu 2016, vol. 13-17-August-2016, hal. 785–794, doi: 10.1145/2939672.2939785.
[28] M. R. Machado, S. Karray, dan I. T. De Sousa, "LightGBM: An effective decision tree gradient boosting method to predict customer loyalty in the finance industry," *14th Int. Conf. Comput. Sci. Educ. ICCSE 2019*, no. Nips, hal. 1111–1116, 2019, doi: 10.1109/ICCSE.2019.8845529.
[29] Q. T. Bui *et al.*, "Gradient boosting machine and object-based cnn for land cover classification," *Remote Sens.*, vol. 13, no. 14, hal. 1–15, 2021, doi: 10.3390/rs13142709.
[30] F. Alzamzami, M. Hoda, dan A. El Saddik, "Light Gradient Boosting Machine for General Sentiment Classification on Short Texts: A Comparative Evaluation," *IEEE Access*, vol. 8, hal. 101840–101858, 2020, doi: 10.1109/ACCESS.2020.2997330.
[31] M. Linting, J. J. Meulman, P. J. F. Groenen, dan A. J. van der Kooij, "Nonlinear Principal Components Analysis: Introduction and Application," *Psychol. Methods*, vol. 12, no. 3, hal. 336–58, 2007, doi: 10.1037/1082-989X.12.3.336.
[32] M. Linting dan A. van der Kooij, "Nonlinear principal components analysis with CATPCA: a tutorial," *J Pers Assess*, vol. 94, no. 1, hal. 12–25, 2012, doi: 10.1080/00223891.2011.627965.
[33] C. Zhao, Y. Xin, X. Li, Y. Yang, dan Y. Chen, "applied sciences for Spam Detection in Social Networks with Imbalanced Data," *Appl. Sci.*, 2020.
[34] O. A. Ebrahim dan G. Derbew, "Application of supervised machine learning algorithms for classification and prediction of type-2 diabetes disease status in Afar regional state, Northeastern Ethiopia 2021," *Sci. Rep.*, vol. 13, no. 1, hal. 1–12, 2023, doi: 10.1038/s41598-023-34906-1.
[35] I. Saputra dan D. A. Kristiyanti, *Machine learning untuk pemula*. BI-OBSES, 2022.
[36] I. M. De Diego, A. R. Redondo, R. R. Fernández, J. Navarro, dan J. M. Moguerza, "General Performance Score for classification problems," *Appl. Intell.*, vol. 52, no. 10, hal. 12049–12063,

2022, doi: 10.1007/s10489-021-03041-7.
[37] D. Ananda, A. Pertiwi, K. Ahmad, J. Unjung, dan M. A. Muslim, "A Performance Comparison of Data Balancing Model to Improve Credit Risk Prediction in P2P Lending," vol. 11, no. 4, hal. 881–890, 2024, doi: 10.15294/sji.v11i4.14018.
[38] W. Wegier dan P. Ksieniewicz, "Application of imbalanced data classification quality metrics as weighting methods of the ensemble data stream classification algorithms," *Entropy*, vol. 22, no. 8, 2020, doi: 10.3390/E22080849.
[39] H. A. Haikal, A. H. Wigena, K. Sadik, dan E. Efriwati, "Comparison of Discriminant Analysis and Support Vector Machine on Mixed Categorical and Continuous Independent Variables for COVID-19 Patients Data," *Sci. J. Informatics*, vol. 11, no. 1, hal. 165–176, 2024, doi: 10.15294/sji.v11i1.48565.
[40] A. Ali, S. M. Shamsuddin, dan A. L. Ralescu, "Classification with class imbalance problem: A review," *Int. J. Adv. Soft Comput. its Appl.*, vol. 7, no. 3, hal. 176–204, 2015.
[41] S. Li, N. Jin, A. Dogani, Y. Yang, M. Zhang, dan X. Gu, "Enhancing LightGBM for Industrial Fault Warning: An Innovative Hybrid Algorithm," *Processes*, vol. 12, no. 1, 2024, doi: 10.3390/pr12010221.
[42] S. I. Serengil, U. G. Tosun, B. Koroglu, S. Imece, dan E. B. Büyükbaş, "A Comparative Study of Machine Learning Approaches for Non Performing Loan Prediction with Explainability," *Int. J. Mach. Learn. Comput.*, vol. 12, no. 5, hal. 208–214, 2022, doi: 10.18178/ijmlc.2022.12.5.1102.
[43] Q. Ni *et al.*, "Establishment and interpretation of the gamma pass rate prediction model based on radiomics for different intensity-modulated radiotherapy techniques in the pelvis," *Front. Phys.*, vol. 11, no. August, hal. 1–8, 2023, doi: 10.3389/fphy.2023.1217275.
[44] P. Xiong, C. Long, H. Zhou, R. Battiston, X. Zhang, dan X. Shen, "Identification of electromagnetic pre-earthquake perturbations from the demeter data by machine learning," *Remote Sens.*, vol. 12, no. 21, hal. 1–27, 2020, doi: 10.3390/rs12213643.
[45] M. SUBIANTO, I. Y. ULYA, E. RAMADHANI, B. SARTONO, dan A. F. HADI, "Application of SHAP on CatBoost classification for identification of variabels characterizing food insecurity occurrences in Aceh Province households," *J. Nat.*, vol. 23, no. 3, hal. 230–244, 2023, doi: 10.24815/jn.v23i3.33548.