IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 2, April 2025, pp. 1220~1231
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i2.pp1220-1231

Per capita expenditure prediction using model stacking based on satellite imagery

Heri Kuswanto, Asva Abadila Rouhan, Marita Qori'atunnadyah, Supriadi Hia, Kartika Fithriasari, Tintrim Dwi Ary Widhianingsih
Department of Statistics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Article Info

Article history:

Received Apr 25, 2024
Revised Nov 5, 2024
Accepted Nov 14, 2024

Keywords:

Convolutional neural network
Java
Model stacking
Poverty
Satellite imagery

ABSTRACT

One of the indicators for measuring poverty is per capita expenditure. However, collecting timely and reliable per capita expenditure data is quite challenging and expensive, as it requires collecting detailed household data directly. One way to deal with this issue is to use satellite image data processed by machine learning methods. This research proposes a method to predict the per capita expenditure of regencies or cities in Indonesia based on satellite imagery using machine learning techniques, such as k-nearest neighbors (KNN), random forest (RF), extreme gradient boosting (XGBoost), and support vector machine (SVM). The predictions are stacked to predict per capita expenditure using least absolute shrinkage and selection operator (LASSO) regression as the meta-learner. The model is trained on Google-Earth-based satellite imagery of Java Island, Indonesia, which provides more update field conditions compared to data collected from Statistics Indonesia (BPS). The research found that the stacked model outperforms the individual methods. However, the R^2 criterion of the stacked method is comparable to that of RF, which is slightly higher than the others.

Corresponding Author:

Heri Kuswanto
Department of Statistics, Institut Teknologi Sepuluh Nopember
St. Raya ITS Sukolilo, Surabaya 60111, Indonesia
Email: heri.kuswanto@its.ac.id

1. INTRODUCTION

Poverty is a crucial issue on a global scale which becomes a big concern of all countries, including Indonesia. Poverty is characterized by an economic incapacity to fulfill essential food and non-food needs as measured through the per capita expenditure. A population is classified as poor if its average monthly per capita expenditure falls below the poverty threshold. Policymakers rely on measures of poverty metrics, assessed at the household or district level, to allocate government resources and assistance effectively. In most countries, household income or household consumption serves as the primary basis for determining the welfare of the poor [1]. Unfortunately, collecting reliable data to measure poverty based on per capita expenditure is both time-consuming and costly, as it requires detailed household surveys [2].

Statistics Indonesia (BPS) is a government agency in Indonesia which is responsible to provide required data for policymaking, as well as to assist statistical activities in other government institutions. One of BPS main activities is to conduct the National Socio-Economic Survey (*Survei Sosial Ekonomi Nasional or SUSENAS*) twice a year, where the consumption expenditure data is collected. Given the long time of data updating process and the high cost of survey for data collection, a new strategy of collecting the required data is indispensible. One of the proposed strategies to estimate poverty is to leverage the sattelite image or mobile data. The satellite image data can be obtained from Google Earth. For instance, it has been showed that light levels observed from sattelite image data is correlated with asset-based wealth measures in 37 African countries [3]. Another research showed that luminuousity from nighttime light can improve the estimates at regional level of countries with the poor statistical system [4]. Although the processing of satellite images can enhance the estimate of economic features, the methods used to capture surface conditions have yet to achieve convincing results, as the level of light captured by the satellite is indistinguishable from the noise especially when measuring low-income areas [4]. Mobile data is introduced as an alternative to infer socioeconomic status and utilized to estimate poverty [5]. Although the method shows convincing results, it can not be generalized to the case of other countries due to different features available in the mobile phone. Several other work examine the implementation of satellite imagery are the following [6]–[8]. Considering the benefits of using satellite imagery, this study aims to evaluate the effectiveness of this data in estimating poverty levels.

Daytime digital images have a higher image resolution than nighttime digital images. This image contains visible objects such as buildings, roads, cars, farm fields, and roofs that allow identifying the well-being of an area. Image color is usually used as the feature for object identification, such as plants have shades of green, urban areas are usually silver or gray which comes from concentration of concrete or other building materials. In addition to color, objects that can be identified are patterns, shapes, and textures. Farms usually have prominent geometric shapes rather than naturally occurring patterns. Deforestation is often square. The straight lines that appear are almost certainly roads, canals, to regional boundaries. It is found that visible visual objects have correlation with expenditure level, so that it can capture the heterogeneity of poverty level in those specific areas [9], e.g., roads are correlated with economic development, which thus provide useful information about the level of development of an area [10]. Related research on the use of digital images can be found in references [9], [11]–[15].

Leveraging sattelite image for poverty estimation is a complex work which requires a fast and realiable method. Machine learning approach is one of the solutions to solve this issue. VGG16 architecture implemented to predict socioeconomic indicators in the context of poverty mapping in Philippine archipelago [12]. More machine and deep learning approaches, e.g., decision tree regression (DTR), support vector machine (SVM), random forest regression (RFR), multi-layer perceptron (MLP), and one-dimensional convolutional neural network (CNN-1D) are utilized to process and analyze multi-source satellite imagery for poverty mapping in East Java, Indonesia [16]. However, performance of those methods needs to be improved. One of the techniques to optimize prediction is by stacking the methods. Prior research implement this technique integrate multiple classifiers to the heart failure study [17]. The advantages of stacking methods in these contexts include improved prediction accuracy and the ability to harness diverse classifier strengths, leading to more robust and reliable models [18]. With these advantages, this research proposes stacking of several machine learning methods to improve the prediction performance. We propose to use least absolute shrinkage and selection operator (LASSO) regression [19] as a meta-learner. LASSO regression has been used in economic [19] and biology fields [20]. Due to frequently update of Google digital imagery, using available satellite imagery data combined with machine learning tools to identify image features is expected to effectively improve the predictability of per capita expenditures. We employ root mean square error (RMSE) and R^2 to evaluate the comparison of the proposed method with other machine learning techniques.

The structure of this paper is organized as follows: section 2 outlines the materials and methods applied in this study, detailing the data, analytical tools, and techniques employed. In section 3, we discuss the results of our analysis, accompanied by a thorough discussion of their implications. Lastly, section 4 presents the conclusion, summarizing the main findings and their potential implications.

2. MATERIALS AND METHOD

2.1. Data

This research uses two datasets to evaluate the method’s performance i.e the per capita expenditure data sourced from the BPS and image data from Google map. The area of study is Java Island, Indonesia, with its spatial location illustrated in Figure 1. Figure 1 depicts raw image of Java Island captured from the Google map. The per capita expenditure data were collected per district or city in 2018. The downloaded image was taken at zoom level 16 and is 400×400 pixels in size. The image will cover an area of 1 km^2. To determine the location of the image to be downloaded, it is necessary to provide a shapefiles file that defines the boundaries of the region. In the shapefiles file used, there are 117 districts/cities for the Java region. An active Maps Static application programming interface (API) is required to download images automatically. In total there are 222,269 images obtained from Google Maps.

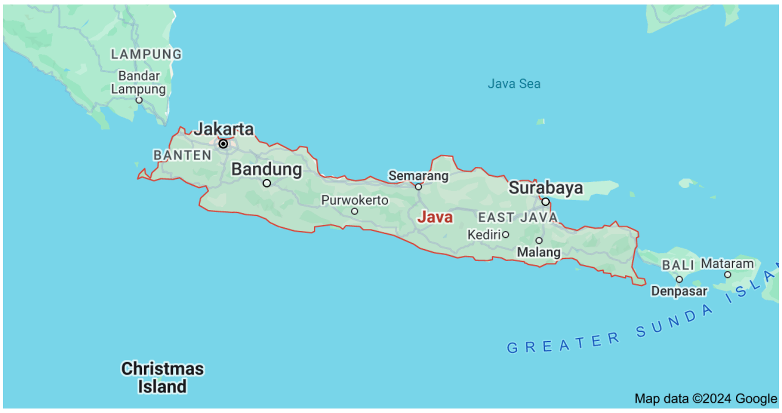


Figure 1. Spatial map of Java Island, Indonesia

2.2. Digital image

Image is represented as a function $f(a,b)$ where a and b are coordinated on a flat plane that represents the value of a pixel in two dimensions [21]. One of the color models is the red, green, blue (RGB) model. RGB representation of images comprises three-color channels i.e., red, green, and blue each with values ranging from 0 to 255. Practically, a color image can be expressed as a vector of three functions as written in (1) [22].

$$f(a,b) = [r(a,b) \quad g(a,b) \quad b(a,b)]) \tag{1}$$

Where a and b indicates latitude and longitude of the specific location respectively.

2.3. Convolutional neural network

A CNN is designed to identify visual patterns directly from images with minimal preprocessing. CNN is described as a neural network that incorporate convolution operations instead of general matrix multiplication in at least one of their layers [23]. The neurons in a CNN are organized in three dimensions. Typically, a CNN composed of three primary layers i.e. convolutional, pooling, and fully connected layer [22].

2.4. Transfer learning

Transfer learning involves transferring knowledge from a pre-trained network to a new network to address similar problems [22]. This technique is implemented to accelerate the training process and enhance model performance. Some examples of pre-trained models are Xception, VGGNet, ResNet, Inception, and MobileNet. Two use cases of transfer learning are fine-tuning and feature extraction [24].

2.5. VGG16

VGGNet was developed by the Visual Geometric Group of Oxford University. This architecture secured the second place in the ImageNet large scale visual recognition challenge 2014 (ILSVRC-2014) competition with a test error of 7.3%. VGG16, one of the best-performing networks, comprises 16 convolutional and pooling layers followed by three fully connected layers and a softmax function [22]. The input to this architecture is a fixed 224×224 RGB image size. It uses a 3×3 a convolution layer with a stride of 1 pixel and a padding of 1 pixel. Spatial pooling is performed by 5 layers of max-pooling layers, each using a 2×2-pixel window with a stride 2. All hidden layers utilize rectified linear unit (ReLu) activation. In total, the VGG16 has 138 million parameters.

2.6. Support vector regression

The suport vector (SV) algorithm is a nonlinear extension of the generalized portrait algorithm introduced by Vapnik [25]. SVM addresses binary classification by framing it as a convex optimization problem. This requires identifying the maximum-margin hyperplane that best separates classes while correctly classifying as many training points as possible. Support vector regression (SVR) defines ϵ-insensitive region around the function, known as ϵ-tube. SVR is also formulated as an optimization problem by establishing a convex insensitive loss function to minimize, aiming to find the flattest tube that contains the majority of training samples. The hyperplane is represented in the form of a support vector, namely the training sample located outside the tube boundary [26]. For multidimensional data, the multivariate regression function is written in (2).

$$f(x) = \begin{bmatrix} w \\ b \end{bmatrix}^T \begin{bmatrix} x \\ 1 \end{bmatrix} = w^T x + b, x, w \in \mathbb{R}^{M+1} \quad (2)$$

The SVR tries to identify the narrowest tube surrounding the surface while minimizing prediction errors. This condition leads to the formulation of the objective function presented in (3), where the magnitude of the vector is normal to the approximated surface.

$$\min_{w} \frac{1}{2} \|w\|^2 \quad (3)$$

The SVR utilizes an ϵ-insensitive loss function that penalizes predictions deviating from desired output. The value of ϵ determines the width of the tube; decreasing ϵ leads to an increase in the number of support vectors, whereas increasing ϵ results in fewer support vectors. Reducing ϵ will result in more support vectors, while increasing ϵ will result in fewer supporting vectors. The standard formulation of SVR for addressing the approximation problem is represented in (4) [27].

$$f(x) = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i)\, k(x_i, x_j) + b \quad (4)$$

Some kernel functions used in SVR are as follows:

- Linear kernel where $k(x_i, x_j) = x_i^T x_j$
- Polynomial kernel where $k(x_i, x_j) = (x_i^T x_j + 1)^n$, $i, j = 1,2, \dots, n$. Figure 2 shows kernel $k(x_i, x_j)$ for $-\infty < x < +\infty$.

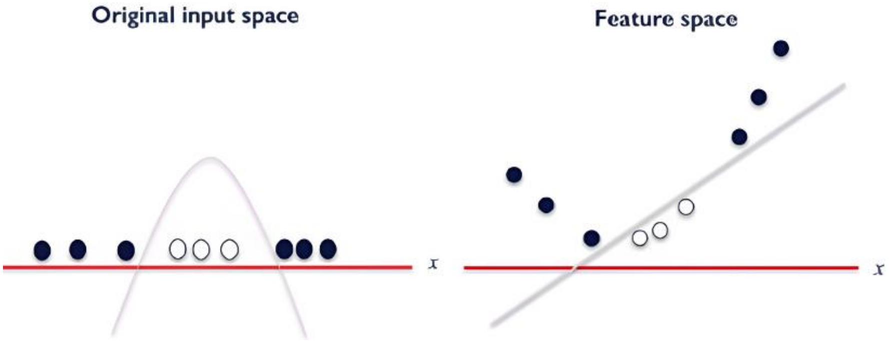


Figure 2. Polynomial kernel illustration

- Radial basis function (RBF) where $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = exp\left(-\gamma \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2\right)$. If $\boldsymbol{x}_i = \boldsymbol{x}$ and $-\infty < z < +\infty$, then the RBF kernel $k(\boldsymbol{x}, \boldsymbol{z})$ can be illustrated in Figure 3. Generally, this kernel has fine performance [28].

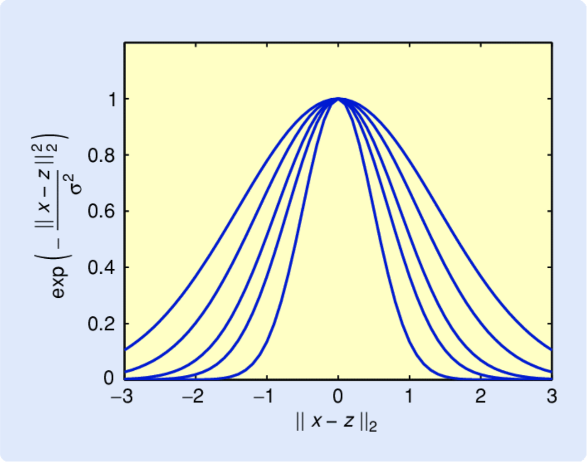


Figure 3. RBF kernel illustration

During the training process, SVR finds the margin hyperplane by estimating sets of parameter α_i and b [29]. SVR performance is also determined by another set of parameters, the so-called hyperparameters. These are the soft margin constant C and the parameter of the kernel, γ, σ, or n.

2.7. Random forest

RFs aims to combine weak learners (e.g., a single decision tree) into strong learners. The RF development process closely follows the bagging method, given a training set contains N samples with M features. The initial step involves bootstrap sampling, where a random sample of N casesis taken with replacement as the training dataset from every single decision tree. At each node, the algorithm randomly selects m variables (where $m < M$), then identifies the predictor variable that gives the best separation among the m variables. A complete tree is then grown without pruning. The predicted outcome is derived from each tree. Consequently, we can obtain predictive results by calculating the mean or weighted average for regression tasks, or through majority voting for classification tasks.

RF utilize two parameters, namely n-tree (the number of trees) and m-try (the number of features used to find the best features), whereas the bagging method employs only uses n-tree as a parameter. Thus, if we set m-try equal to the total number of features in the training dataset, the RF equals tobagging method. The primary advantages of RFs include ease of computation, efficient data processing, and resilience to errors from missing or unbalanced data. A notable disadvantage of a RF is that it cannot predict values beyond the range of the training dataset and are susceptible to overfitting in the presence to noisy data over-fitting [30].

2.8. XGBoost

Extreme gradient boosting (XGBoost) is a machine learning technique employed for regression analysis and classification, derived from gradient boosting decision tree (GBDT). This method was first introduced by [31], who linked boosting and optimization in the development of gradient boosting machine (GBM). The boosting approach involves creating a new model to predict the errors made by the previous model, and this process is used in the boosting method. The addition of new models is carried out until no more error corrections can be made. The gradient boosting method uses gradient descent to minimize errors when creating a new model. The computational process of the XGBoost algorithm is shown in Figure 4 [32]. A new tree is created in the direction of the negative gradient of the loss function. As the number of tree models increases, the loss diminishes progressively.

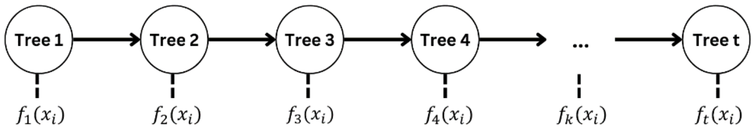


Figure 4. A schematic diagram of XGBoost algorithm

The predicted value at step t is assumed to be y based on (5).

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) \tag{5}$$

For the XGBoost algorithm, determining the number of trees and depth is important. The problem in finding the optimum algorithm can be changed by searching for a new classification that can reduce the loss function, with the target loss function shown in (6).

$$Obj^{(t)} = \sum_{i=1}^{t} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{i=1}^{t} \Omega(f_i) \tag{6}$$

Because the ensemble tree model in (6) is a function as a parameter and cannot be optimized using traditional optimization methods on Euclidean space. So it was replaced with a model trained in an additive way, with y using the i-th prediction and t-th iteration [33]. In minimizing the loss function, f_t is added so that the (7) is obtained as (7).

$$Obj^{(t)} = \sum_{i=1}^{t} l\left(y_i, \hat{y}_i^{(t)}\right) + \left(f_t(x_i)\right) + \Omega(f_t) + constant \tag{7}$$

The regularization term $\Omega(f_i)$ can be calculated using (8) which is used to reduce model complexity and can improve usability on other datasets.

$$\Omega(f_i) = \gamma T + \frac{1}{2}\lambda\|\omega\|^2 \quad (8)$$

2.9. LASSO regression

The linear regression model, in general, can be written as (9).

$$\mathrm{y} = \mathrm{X}\beta + \varepsilon \quad (9)$$

Where y is the observation vector of the response variables with size n×1, $X = (x_1, x_2, \ldots, x_n)^T$, $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$, i=1,2,…, n is the n×p matrix of predictor variables and $\varepsilon = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)^T$ is the error vector. $E(\varepsilon_i)$=0 and $Var(\varepsilon_i)$=σ^2. Then it will be searched $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)^T$ as an estimator for β which will find the sum of the squares of the errors, namely $\varepsilon^T\varepsilon$.

One of the assumptions that must be met when using linear regression is that the predictor variables used should not be multicollinear. LASSO regression is one of the methods used to select predictor variables when multicollinearity occurs in the regression model. By standardizing column X so that $n^{-1}\sum_{i=1}^{n} x_{ij} = 0$ and $n^{-1}\sum_{i=1}^{n} x_{ij}^2 = 1$, the LASSO $\hat{\beta}$ estimator is defined as (10).

$$\hat{\beta} = \arg\min_{\beta}\{\textstyle\sum_{i=1}^{n}(y_i - x_i^T\beta)^2\} \quad (10)$$

With the constraint function $\sum_{j=1}^{p}|\beta_j| \leq s$, where s is a quantity that controls the amount of shrinkage in the LASSO regression coefficient. The (10) can also be written as (11).

$$\hat{\beta} = \arg\min_{\beta}\{\textstyle\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\} \quad (11)$$

With λ is the tuning paramater of LASSO regression. λ has a maximum value of 0. If the value of alpha close to 0, the LASSO estimate is equal to ordinary least squares (OLS) estimate. The LASSO estimation does not have an explicit solution because the LASSO regression constraint function is an absolute function that cannot be derived at the inflection point. One way to get the LASSO estimate is to use the least-angle regression (LARS) algorithm [34].

2.10. K-nearest neighbors

K-nearest neighbours (KNN) was developed with the assumption that similar things exist in close proximity to each other and can be applied to regression and classification. As a parametric model, linear regression has several advantages, including easy to get fit values and simple interpretation, but has a weakness in terms of flexibility, especially in terms of assumptions that must be met. KNN is a non-parametric method, comes with more flexibility and can overcome the problem of assumptions. KNN estimates the relationship between independent variables and sustainable outcomes by averaging observations in the same neighbor. The KNN model can be written as (12).

$$Y = f(X) + \varepsilon \quad (12)$$

Where $\hat{f}(x) = \frac{1}{K}\sum_{x\ in\ N} y_j$ and K are the numbers of neighbors.

The KNN algorithm is: i) determine the metric distance; ii) determine the number of nearest neighbors (k<n); iii) calculate the distance from another data point to the desired point; iv) sort the distance from smallest to largest; and v) calculate the average of the KNN. Generally, KNN uses Euclidean distance in distance calculation. For p dimensions and data samples, the Euclidean distance between 2 points $a_1, a_2, a_3, \ldots, a_p$ and $b_1, b_2, b_3, \ldots, b_p$ is (13) [35].

$$Euclidean = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_p - b_p)^2} \quad (13)$$

While the response variable value for new data (x_{new}) can be calculated by (14).

$$y_{new}(x_{new}) = \frac{1}{k}\textstyle\sum_{i \in k(x_{new})} y_i \quad (14)$$

The value of k determines the accuracy of the model so choosing the optimal value of k is very important.

2.11. Goodness of fit

In this research, the goodness of fit was measured by R^2 and RMSE. R^2 is used for explanatory purposes and explains how well the independent variable can explain the variability in the dependent variable. While RMSE shows the level of goodness of fit based on error, where the smaller the RMSE value, the better the model used. R^2 is calculated by (15).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (15)$$

Next RMSE is calculated by (16).

$$RMSE = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{n}} \quad (16)$$

3. RESULTS AND DISCUSSION

The geographic distribution of population welfare is a key determinant of economic growth and serves as a crucial basis for decision-making regarding resource allocation. Figure 5 illustrates the distribution of per capita expenditure across cities and regencies on Java Island in 2018. Figure 6 highlights that the distribution of consumption expenditure per capita is right-skewed and identifies seven outliers, represented by points adjacent to the boxplot. Five of these outliers are located in Jakarta's mainland, with the remaining two in the City of Semarang and the City of Bekasi. Families residing in these cities spend, on average, over 20 million rupiahs annually, whereas a typical family in Temanggung Regency spends only 8.2 million rupiahs.

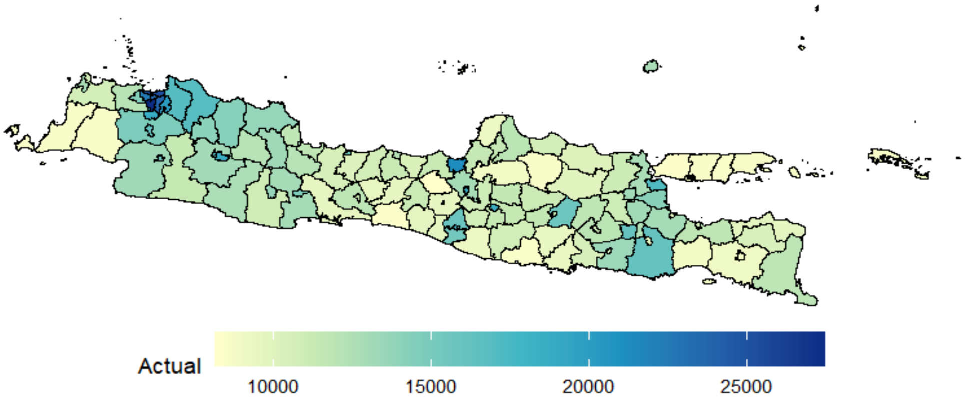


Figure 5. Map distribution of consumption expenditure per capita in 2018

Each image obtained from Google Maps is processed using the VGG16 model, extracting a 4096-dimensional feature vector that serves as the predictor variable. The data set, consisting of 117 observations representing consumption expenditure per capita at the city/regency level, is divided into a training set (80% of the data) and a validation set (20% of the data). Given the small size of the data, we employ bootstrap sampling with 25 repetitions as a cross-validation technique.

Our base models include XGBoost, KNN, RF, and SVM. Each model undergoes hyperparameter tuning, with RF, XGBoost, and KNN having 6, 16, and 10 hyperparameter combinations, respectively. The SVM model is tuned across 20 hyperparameter combinations, distributed equally among four kernels: linear, 2nd-degree polynomial, 3rd-degree polynomial, and RBF. The predictions from these seven base models are used as new predictor variables in a stacked model, where LASSO regression is employed to generate the final prediction. LASSO regression is chosen due to the high correlation among the base model predictions, ensuring a more robust final model. Figure 7 illustrated the performance result of all models'; training set is illustrated in Figure 7(a) and validation set is illustrated in Figure 7(b).

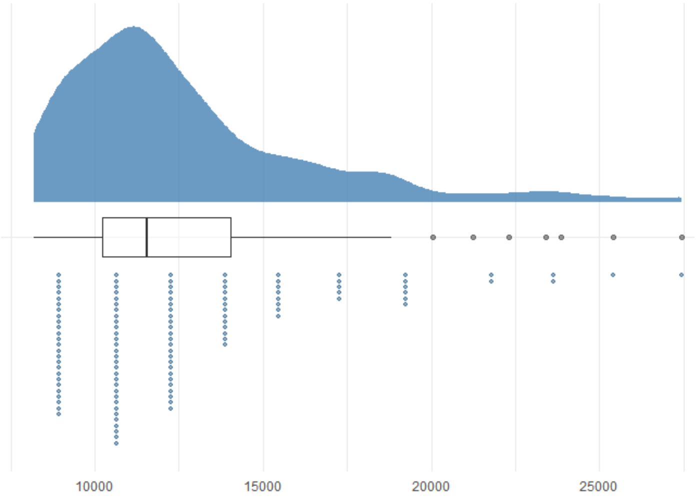
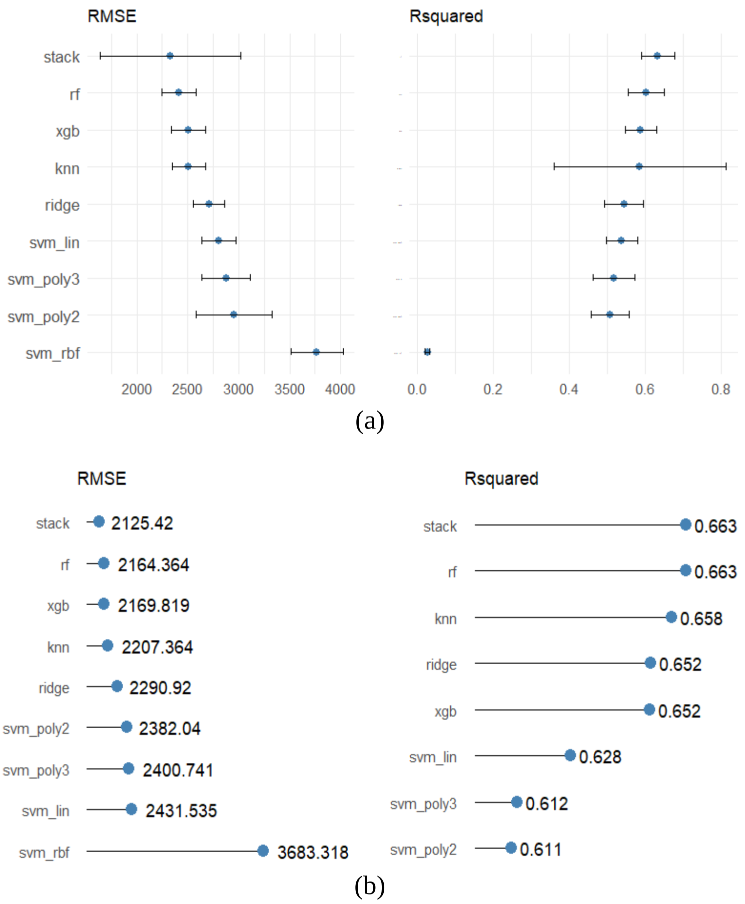


Figure 6. Distribution of consumption expenditure per capita



(a)

(b)

Figure 7. Performance results of (a) training set and (b) validation set

The stacked model achieved a lower RMSE than RF and the other models in both the training and validation sets. However, it exhibited the highest standard deviation in both RMSE and R², suggesting that the stacked model (LASSO regression) struggled to generalize well to the training data. In terms of R² values, the stacked model was outperformed by RF and performed similarly to XGBoost and KNN in the training set, indicating it did not achieve the best R². In the validation set, the stacked model's performance was comparable to that of RF. On the other hand, the SVM with an RBF kernel was the poorest-performing model, likely due to suboptimal hyperparameter combinations. Optimizing the RBF SVM would require more time and resources.

The model stacking approach yields excellent results when applied to the entire dataset, as shown in Figure 8, with an RMSE of 936.7027 and an R² of 0.9385487. The stacked model appears to handle outliers reasonably well, as 4 out of the 7 outliers have residuals less than twice the RMSE. However, the model tends to underpredict when the consumption expenditure per capita exceeds 15,000. While the results are promising, the validation phase suggests that the model may not be sufficiently robust for predicting new data. Figure 9 presents a map of the stacked model's predictions, which clearly shows that high degree of error observed only for few districts.

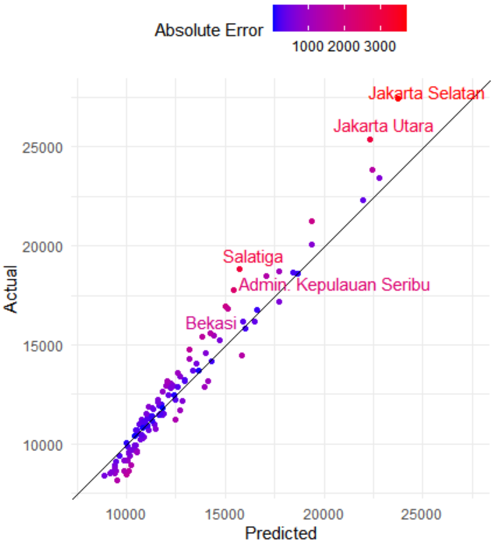


Figure 8. Comparison between actual and predicted value

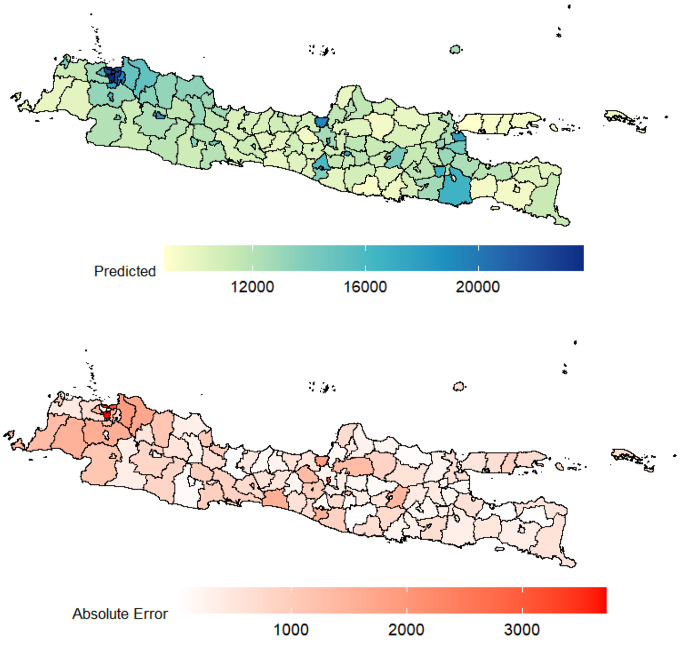


Figure 9. Mapping prediction results and percentage of model error

4. CONCLUSION

The stacked model is the most accurate but also the least precise among the models tested. The stacking method provides only marginal improvements in RMSE and R² compared to individual models, such as RF, suggesting that model selection may yield performance comparable to model stacking. This implies that the benefits of stacking are limited in this context. Our model's performance aligns with findings from various previous studies. However, predicting economic indicators using feature extraction from satellite images does not outperform traditional econometric methods. To enhance the predictive power of satellite images, fine-tuning a CNN through transfer learning could be explored in future work. Although their results did not surpass traditional econometric methods, their approach highlights that selecting the right CNN architecture and optimizing its parameters could potentially close this gap. Fine-tuning the CNN is expected to produce results that are more competitive with traditional methods.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge financial support from the Ministry of Education, Culture, Research and Technology-Indonesia for this work, under project scheme of Fundamental research 2023.

REFERENCES

[1] World Bank, “Introduction to poverty analysis,” *World Bank Group*, 2014. Accessed: Jan. 07, 2024. [Online]. Available: http://documents.worldbank.org/curated/en/775871468331250546/Introduction-to-poverty-analysis

[2] M. Jerven, “Poor numbers-how we are misled by African development statistics and what to do about it (Uzuazo Etemire),” in *Verfassung in Recht und Übersee*, vol. 46, no. 3, 2013, pp. 336–340.

[3] A. M. Noor, V. A. Alegana, P. W. Gething, A. J. Tatem, and R. W. Snow, “Using remotely sensed night-time light as a proxy for poverty in Africa,” *Population Health Metrics*, vol. 6, 2008, doi: 10.1186/1478-7954-6-5.

[4] X. Chen and W. D. Nordhaus, “Using luminosity data as a proxy for economic statistics,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 21, pp. 8589–8594, 2011, doi: 10.1073/pnas.1017031108.

[5] J. Blumenstock, G. Cadamuro, and R. On, “Predicting poverty and wealth from mobile phone metadata,” *Science*, vol. 350, no. 6264, pp. 1073–1076, 2015, doi: 10.1126/science.aac4420.

[6] R. Engstrom, J. Hersh, and D. Newhouse, “Poverty from space: using high resolution satellite imagery for estimating economic well-being,” *World Bank Economic Review*, vol. 36, no. 2, pp. 382–412, 2022, doi: 10.1093/wber/lhab015.

[7] T. Stark, M. Wurm, X. X. Zhu, and H. Taubenbock, “Satellite-based mapping of urban poverty with transfer-learned slum morphologies,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5251–5263, 2020, doi: 10.1109/JSTARS.2020.3018862.

[8] C. Yeh *et al.*, “Using publicly available satellite imagery and deep learning to understand economic well-being in Africa,” *Nature Communications*, vol. 11, no. 1, 2020, doi: 10.1038/s41467-020-16185-w.

[9] N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, and S. Ermon, “Combining satellite imagery and machine learning to predict poverty,” *Science*, vol. 353, no. 6301, pp. 790–794, 2016, doi: 10.1126/science.aaf7894.

[10] S. M. Pandey, T. Agarwal, and N. C. Krishnan, “Multi-task deep learning for predicting poverty from satellite images,” *Proceedings of the 30th Innovative Applications of Artificial Intelligence Conference, IAAI 2018*, pp. 7793–7798, 2018, doi: 10.1609/aaai.v32i1.11416.

[11] A. Head, M. Manguin, N. Tran, and J. E. Blumenstock, “Can human development be measured with satellite imagery?,” *Proceedings of the Ninth International Conference on Information and Communication Technologies and Development*, pp. 1–11, 2017, doi: 10.1145/3136560.3136576.

[12] I. Tingzon *et al.*, “Mapping poverty in the Philippines using machine learning, satellite imagery, and crowd-sourced geospatial information,” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 42, no. 4/W19, pp. 425–431, 2019, doi: 10.5194/isprs-archives-XLII-4-W19-425-2019.

[13] J. Yin, Y. Qiu, and B. Zhang, “Identification of poverty areas by remote sensing and machine learning: A case study in guizhou, southwest China,” *ISPRS International Journal of Geo-Information*, vol. 10, no. 1, pp. 1-19, Dec. 2021, doi: 10.3390/ijgi10010011.

[14] J. Hersh, R. Engstrom, and M. Mann, “Open data for algorithms: mapping poverty in Belize using open satellite derived features and machine learning,” *Information Technology for Development*, vol. 27, no. 2, pp. 263–292, 2021, doi: 10.1080/02681102.2020.1811945.

[15] B. Babenko, J. Hersh, D. Newhouse, A. Ramakrishnan, and T. Swartz, “Poverty mapping using convolutional neural networks trained on high and medium resolution satellite images, with an application in Mexico,” *arXiv-Statistics*, pp. 1-4, 2017.

[16] S. R. Putri, A. W. Wijayanto, and S. Pramana, “Multi-source satellite imagery and point of interest data for poverty mapping in East Java, Indonesia: Machine learning and deep learning approaches,” *Remote Sensing Applications: Society and Environment*, vol. 29, 2023, doi: 10.1016/j.rsase.2022.100889.

[17] C. C. Chiu, C. M. Wu, T. N. Chien, L. J. Kao, C. Li, and H. L. Jiang, “Applying an improved stacking ensemble model to predict the mortality of ICU patients with heart failure,” *Journal of Clinical Medicine*, vol. 11, no. 21, pp. 1-20, Oct. 2022, doi: 10.3390/jcm11216460.

[18] N. Kardani, A. Zhou, M. Nazem, and S. L. Shen, “Improved prediction of slope stability using a hybrid stacking ensemble method based on finite element analysis and field data,” *Journal of Rock Mechanics and Geotechnical Engineering*, vol. 13, no. 1, pp. 188–201, 2021, doi: 10.1016/j.jrmge.2020.05.011.

[19] R. Tibshirani, “Regression Shrinkage and selection via the LASSO,” *Journal of the Royal Statistical Society. Series B: Methodological*, vol. 58, no. 1, pp. 267–288, 1996, doi: 10.1111/j.2517-6161.1996.tb02080.x.

[20] B. W. Sloboda, D. Pearson, and M. Etherton, “An application of the LASSO and elastic net regression to assess poverty and economic freedom on ECOWAS countries,” *Mathematical Biosciences and Engineering*, vol. 20, no. 7, pp. 12154–12168, 2023, doi: 10.3934/mbe.2023541.

[21] R. C. Gonzalez and R. E. Woods, *Digital image processing*. New York: Pearson, 2018.

[22] M. R. Karim, M. Sewak, and P. Pujari, *Practical convolutional neural networks: implement advanced deep learning models using Python*. Packt Publishing, 2018.

[23] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning in adaptive computation and machine learning*. The MIT Press, 2016.

[24] J. Patterson and A. Gibson, *Deep learning: a practitioner's approach, First edition*. O'Reilly, 2017.

[25] V. N. Vapnik, *The nature of statistical learning theory*. New York: Springer, 1995.

[26] M. Awad and R. Khanna, "Efficient learning machines: Theories, concepts, and applications for engineers and system designers," *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, pp. 1–248, 2015, doi: 10.1007/978-1-4302-5990-9.

[27] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Information Processing-Letters and Reviews*, pp. 203–224, 2008.

[28] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: concerns and ways forward," *PLoS ONE*, vol. 13, no. 3, 2018, doi: 10.1371/journal.pone.0194889.

[29] J. S. Racine, L. Su, A. Ullah, W. K. Härdle, D. D. Prastyo, and C. M. Hafner, "Support vector machines with evolutionary model selection for default prediction," *The Oxford Handbook of Applied Nonparametric and Semiparametric Econometrics and Statistics*, 2014, doi: 10.1093/oxfordhb/9780199857944.013.011.

[30] Y.-W. Chiu, *Machine learning with R cookbook: explore over 110 recipes to analyze data and build predictive models with the simple and easy-to-use R code*. 2015.

[31] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001, doi: 10.1214/aos/1013203451.

[32] H. Mo, H. Sun, J. Liu, and S. Wei, "Developing window behavior models for residential buildings using XGBoost algorithm," *Energy and Buildings*, vol. 205, 2019, doi: 10.1016/j.enbuild.2019.109564.

[33] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.

[34] W. K. Härdle and L. Simar, *Applied multivariate statistical analysis*. Berlin, Heidelberg: Springer, 2019.

[35] V. Kumar and M. Sahu, "Evaluation of nine machine learning regression algorithms for calibtion of low-cost PM2.5 sensor," *Journal of Aerosol Science*, vol. 157, 2021, doi: 10.1016/j.jaerosci.2021.105809.

BIOGRAPHIES OF AUTHORS

Heri Kuswanto is a Professor in the Department of Statistics at Institut Teknologi Sepuluh Nopember (ITS), Surabaya. He obtained his bachelor's and master's degree in Statistics from ITS and his Ph.D. from Leibniz University Hannover, Germany, in 2009. After completing his Ph.D., he worked as a Postdoctoral Research Associate at Laval University, Canada for a year. He has received numerous international research grants to promote international collaboration, including the Partnership for Enhanced Engagement in Research (PEER), funded by the National Science Foundation (NSF) and USAID, The World Academy of Science (TWAS) and UNESCO, and Degree Modelling Funds (DMF). His research focuses on machine learning, computational statistics, time series analysis, forecasting, and advanced statistical modeling. He has published many papers on machine learning, weather forecasting, and computational statistics fields. He can be contacted at email: heri.kuswanto@its.ac.id.

Asva Abadila Rouhan earned his bachelor's and master's degrees in statistics from Institut Teknologi Sepuluh Nopember (ITS), Indonesia. He is interested in statistical computation, machine learning, and large language models. He has gained notable experience as a research assistant in the Covid-19 behavior change survey. He can be contacted at email: abadilaasva@gmail.com.

Marita Qori'atunnadyah received a bachelor's degree in statistics (2020) and a master's degree in statistics (2022) from the Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Her research interests encompass statistical computations, statistical business and industry. She can be contacted at email: mrtndyh23@gmail.com.

Supriadi Hia earned his Bachelor of Applied Science degree in Statistics (2015) from Sekolah Tinggi Ilmu Statistik (STIS), Jakarta, Indonesia, and Master of Statistics (2022) from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia. He started working at BPS-Statistics since 2016 until now. During his work at BPS-Statistics, he was part of the 'Regional balance and statistical analysis' team and the 'Integration of statistical processing and dissemination' team. Currently working at BPS-Statistics of Nias Utara Regency as a 'young statistician', and team leader of 'Integration of statistical processing and dissemination'. He can be contacted at email: supriadi.hia@bps.go.id.

Kartika Fithriasari is a lecturer at Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia. She earned her bachelor's degree in Statistics from ITS, master's degree in the same field from Institut Pertanian Bogor (IPB), and doctoral degree from statistics ITS. Her research interests include computational statistics, data science, neural network, and Bayesian statistics. She has received recognition for her teaching and research contributions. She can be contacted at email: kartika_f@its.ac.id.

Tintrim Dwi Ary Widhianingsih received her bachelor's degree in statistics from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, in 2016, followed by a master's degree in statistics from the same institution in 2018. She completed her Ph.D. in computer engineering at Dongseo University, South Korea, in 2021. Her doctoral research focused on the application of machine learning and deep learning techniques in text mining, high-dimensional data analysis, and image processing. From 2016 to 2018, she served as a research assistant in the Computational Statistics and Data Science Laboratory. Since 2022, she has been an Assistant Professor in the Department of Statistics, ITS. Her research interests encompass machine learning, deep learning, and computational statistics. She can be contacted at email: dwi.ary@its.ac.id.