Indonesian Journal of Data and Science

Volume 4 Issue 3
ISSN 2715-9936
https://doi.org/10.56705/ijodas.v4i3.102

Research Article

Comparison of Machine Learning Land Use-Land Cover Supervised Classifiers Performance on Satellite Imagery Sentinel 2 using Lazy Predict Library

Muhamad Iqbal Januadi Putra [1,2,*], Vincent Alexander [3]
[1] Universitas Terbuka, Indonesia, 048942434@ecampus.ut.ac.id
[2] Universitas Siber Asia, Jakarta, Indonesia
[3] Universitas Tarumanagara, Jakarta, Indonesia, alex.535220149@stu.untar.ac.id

Correspondence should be addressed to Gst Ayu Vida Mastrika Giri; vida@unud.ac.id

Received 05 November 2023; Accepted 28 November 2023; Published 31 December2023

Abstract:
The utilisation of various supervised classifier algorithms in classifying land use and land cover (LULC) from satellite imagery has been widely used worldwide, yet the implementation using lazy predict library remained unexplored. This study aims to create the LULC supervised classifier model for Sentinel 2 satellite images using lazy predict library and assess its capability for creating multiple machine learning models. The result of this study shows that lazy predict library can generate 26 machine learning models in efficient few lines of code and less time-consuming. Most LULC models generated by lazy predicts has performance metrics above 90% with time computation between 0 and 1 seconds. While lazy predict library has benefits to generate various machine learning models at once, it has drawbacks in terms of its feasibility for the machine learning production, its obstacle running in local environment, and its requirements for the RAM computation.

Keywords: Classification, machine learning, lazy predict library, LULC, remote sensing
Dataset link: https://github.com/iqbal1201/machine_learning_lazy_predict/tree/main/datasets

1. Introduction:

Land use and land cover (LULC) classification from remote sensing satellite imagery has been widely used to understand the physical change of environmental system in various area [1]–[4]. As many papers had explained, LULC (e.g. deforestation, urbanization) can bring a detrimental impact towards the environment, such as flood, landslide, water quality degradation, and climate change [5]. This approach provides a critical information in various spatiotemporal dimension for the decision-makers to properly manage and plan the environmental and ecological system [2] Given this critical information, producing accurate LULC classification from satellite imagery is vital for sustainable management and natural resource management [2].

Several various researches have been conducted to explore the capabilities of each machine learning algorithm/classifier to classify the LULC from satellite imagery. Most of these papers utilize the common machine learning algorithm. For instance, [3] examined the performance of classification and regression tree (CART), random forest (RF), and support vector machine (SVM) to classify the LULC from satellite imagery. Similary, [2] and [6] utilized the support vector machine (SVM), maximum likelihood (ML), and random forest/random tree (RF/RT) for the same goal. Both papers showed that those classifiers result in good accuracy when classifying the LULC from satellite imagery. Furthermore, [7] and [5] proposed another machine learning classifier in their papers, such as gradient tree boosting (GTB), multilayer perceptron neural networks (MLP-ANN), and extreme gradient boosting classification (XGBoost).

While significant papers have discussed various machine learning classifiers capability in classifying LULC from satellite imagery, however, there is insignificant papers explore this capability using the lazy predict library. In fact,

this library can give us more holistic information when it comes to the comparison of machine learning model performance. Unlike in other fields such as medicine [8], [9], there is less papers discussed the application of lazy predict library to create multiple classifiers in remote sensing.

Therefore, this paper aims to create multiple supervised classifiers in LULC classification by leveraging the lazy predict library. On top of that, this paper will also review the capability and feasibility of this library to generate a proper classifier model. A range of supervised classifier algorithms will be used to classify this dataset.

2. Method:

The methodology of this project consists of study area selection, data preparation, data preprocessing, cross-validation, learning simulation (lazy predict library), prediction, evaluation performance, and selecting the best classifier model [10]–[14]. Each step is described in the following paragraph.

A. Study Area

Classifying LULC from satellite imagery (Sentinel 2) requires selecting the area of study as we must collect the data and generate a sampling to choose the label for the machine learning training purpose. The area of study for this paper is in some area in Riau, Indonesia following the availability of Sentinel 2 image with sufficient clear cloud coverage. **Figure 1** depicts the area of study of this research.

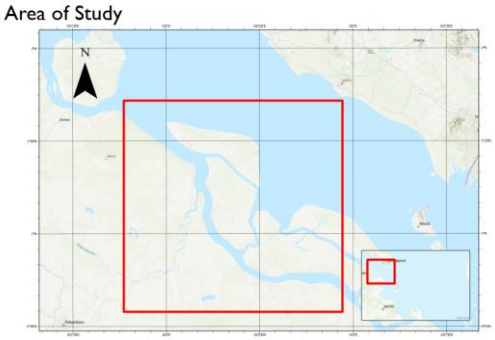

Figure 1. Area of study

B. Dataset

This paper utilizes satellite image Sentinel 2 which collected freely from Scihub Copernicus (https://dataspace.copernicus.eu/). This satellite image contains 13 bands in 10, 20, or 60 meters in pixel size. In detail, these bands are B1 (aerosol), B2 (blue), B3 (green), B4 (red), B5 (red edge), B6 (near infrared), B7 (near infrared), B8A (near infrared), B10 (cirrus), B11 (shortwave infrared), and B12 (shortwave infrared). Each band has specific wavelength spectrum which can highlight specific information according to the reflectance level each object on earth surface. For instance, B5, B6, B7, B8A can be used to extract the vegetation information from the earth surface since its wavelengths are very sensitive to the chlorophyll inside the leaf. The Sentinel 2 image used for this paper is shown in **Figure 2**.

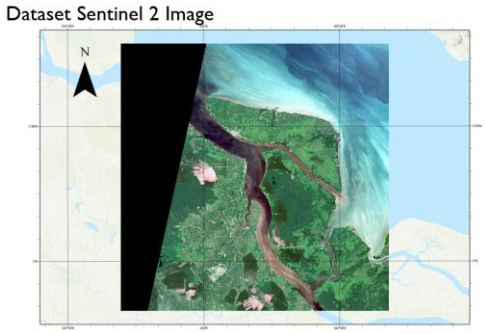

Figure 2. Sentinel 2 image

C. LULC Label Sampling

Classification problem in machine learning model requires the labels for the training as this approach will assign each object to one of the predefined categories/classes. Thus, label sampling is required to collect the categories of LULC from the Sentinel 2 image. There five categories LULC for this paper (shown in **Table 1**). Equally randomized samples were created for each category by creating the digitized point on the Sentinel 2 image. Each point represents the LULC categories and its digital number/reflectance number from the Sentinel 2 image. For instance, if one point digitized/collected from the water body area (sea, river, lake, etc.), this point will be digitized on the water surface and collects the digital number of Sentinel 2 image. The total number of labels was 75.000 sample for all the categories.

Table 1. Label sampling for LULC classification

No	Label Name	Id Class
1.	Bareland	1
2.	Built-Up Area	2
3.	Vegetation	3
4.	Water Body	4
5.	Cloud	5

D. Data Preprocessing

Data are commonly not "clean" due to the presence of redundant, inconsistent, and missing values [15], [16]. "Dirty" data can lead to confusion for the mining and learning process, resulting in inaccurate output [16]. Therefore, data preprocessing is undertaken to deal with this issue by cleaning, transforming, and selecting features [15], [16].

Data preprocessing was started by doing some image processing, such as atmospheric image correction and image index generation. Atmospheric correction aims to convert the Sentinel 2 digital numbers (DN) to the reflectance value. This process can be conducted by implementing the dark object subtraction for each band in Sentinel 2 image.

Then, after all bands image corrected, only selected bands will be transformed to create the image index/raster index. Image index generation aims to transform and select all the features into more meaningful features, following the LULC categories we were interested in. Thus, there were four image indices created for this purpose, namely Normalize Difference Vegetation Index (NDVI), Modified Normalize Difference Water Index (MNDWI), Soil Adjusted Vegetation Index (SAVI), and Modified Bare Soil Index (MBI) with the following formula:

$$NDVI = \frac{(B8A - B4)}{(B8A + B4)} \quad (1)$$

$$MNDWI = \frac{(B3 - B11)}{(B3 + B11)} \quad (2)$$

$$MBI = \frac{(B11 - B12 - B8A)}{(B11 + B12 + B8A)} + 0.5 \quad (3)$$

$$SAVI = \frac{(B8A - B4)}{(B8A + B4 + 0.5)} x\ 1.5 \quad (4)$$

The next process was extracting the all the image indices into the points we created in the previous sampling step. Therefore, all the points will have the label LULC classes/categories following with the NDVI, MNDWI, MBI, and SAVI values. This is noted that all the image indices act as the features for the machine learning models. All the data (label and features) were converted into comma-separated values (csv) format so that it can be easily consumed by the machine learning pipeline.

In the machine learning pipeline, the data cleansing, transformation, and normalisation once again was conducted to ensure that the data have a good quality. Data cleaning will be undertaken to fill in missing values, smooth out outliers, and correct inconsistencies in the data [15], [16]. Filling the missing value in each column in the dataset will be conducted by replacing it with its mean value, one of the most widely used strategies for handling the missing value [16]. Outlier detection will be carried out using a statistical-based method, which assumes the data distribution is normal and the outlier is the attributes with attribute values outside the standard deviation from the mean.

Meanwhile, data normalisation will be applied to avoid the concealing effect of a particular feature on other features due to the varying ranges [16]. This project will apply data normalisation using Max-min normalisation,

which transforms the dataset into a predefined range 0-1 [16]. The normalisation will be applied mainly to the classifier, which highly depends on the distance. Then, the data is divided into training set and test set with the ratio 80:20 for training: test. All the process was conducted using the scikit-learn pipeline to make sure all the datasets treat equally and running in the Google Colab environment (free GPU tiers).

E. Evaluation Performance

Classification performance is required to assess/validate the quality of the classifier model. The classification result from the training set will be compared with its test set to measure how well the trained model fits the test set. This step will be used to determine which classifier is better in classifying LULC categories.

Due to the multiclass classification purpose in this project, a 5×5 confusion matrix or contingency table will be applied to measure the performance level of each classifier [17]. This matrix consists of true positive (TP) if the actual value is positive and predicted as positive, false negative (FN) if the actual value is positive and predicted as a negative value, false positive (FP) if the actual value is negative and predicted as positive, and true negative (TN) if the actual value is negative and predicted as negative [17]. These values (TP, FN, FP, and TN) can be used to calculate the following evaluation metrics [18]–[22].

By default, lazy predict will compute the metrics Accuracy, Balanced Accuracy, and F1 Score [23]–[26]. Accuracy is the ratio of correctly predicted instances to the total number of instances in the dataset. If the category/label is in imbalance, the Balanced Accuracy is more favourable. F1 Score is the harmonic mean between Precision and Recall [17].

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

$$Balanced\ Accuracy\ = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \quad (2)$$

$$F1\ Score\ = \frac{2TP}{n + TP - TN} \quad (3)$$

3. Result and Discussion:

Lazy predict library computed 26 machine learning model/classifiers, sorted according to the performance metrics level shown in **Table 2**. There are seven classifiers with highest accuracy, balanced accuracy, and F1 Score value, namely Random Forest, Extra Tress, KNeighbors, SCV, Bagging, LGBM, and Quadratic Discriminant Analysis. Among these top seven performance, Quadratic Discriminant Analysis is the model with the fastest computation (0.03 s) while Random Forest is slowest one (3.21 s). This indicates that Quadaratic Discriminat Analysis could be the best option for building a machine learning model classifier.

Table 2. Lazy predict result.

Classifier	Accuracy	Balanced Accuracy	F1 Score	Time Taken (s)
Random Forest	0.98	0.98	0.98	3.21
Extra Tress	0.98	0.98	0.98	0.85
KNeighbors	0.98	0.98	0.98	0.31
SVC	0.98	0.98	0.98	0.92
Bagging	0.98	0.98	0.98	0.61
LGBM	0.98	0.98	0.98	0.98
Quadratic Discriminat Analysis	0.98	0.98	0.98	0.03
Extra Tree	0.97	0.98	0.97	0.03
Label Propagation	0.97	0.98	0.97	8.46
Logistic Regression	0.97	0.97	0.97	0.38
Decision Tree	0.97	0.97	0.97	0.12
Label Spreading	0.97	0.97	0.97	16.13
Gaaussian NB	0.95	0.95	0.95	0.03
NuSVC	0.95	0.95	0.95	27.02
Nearest Centroid	0.94	0.95	0.94	0.04
Calibarated Classifier CV	0.94	0.94	0.94	1.87
Passive Aggressive	0.94	0.94	0.94	0.06
Linear SVC	0.94	0.94	0.94	0.50
SGD Classifier	0.93	0.93	0.93	0.13

Classifier	Accuracy	Balanced Accuracy	F1 Score	Time Taken (s)
Linear Discriminat Analysis	0.93	0.93	0.93	0.06
AdaBoost	0.90	0.90	0.90	1.00
Perceptron	0.90	0.90	0.90	0.05
Ridge Classifier CV	0.82	0.82	0.80	0.04
Ridge Classifier	0.82	0.82	0.80	0.03
Bernoulli NB	0.67	0.64	0.64	0.03
Dummy Classfier	0.21	0.2	0.07	0.02

Figure 3 shows the frequency distribution of all performance metrics for the models generated by lazy predicts. It shows that most models have accuracy, balanced accuracy, and F1 score rate above 90%, unless Ridge Classifier CV and Ridge Classifier which only achieve 82%. In contrast, there are two classifiers that have a low performance, namely Bernoulli NB and Dummy Classifier. Both these classifiers have accuracy, balanced accuracy, and F1 score under 70%.

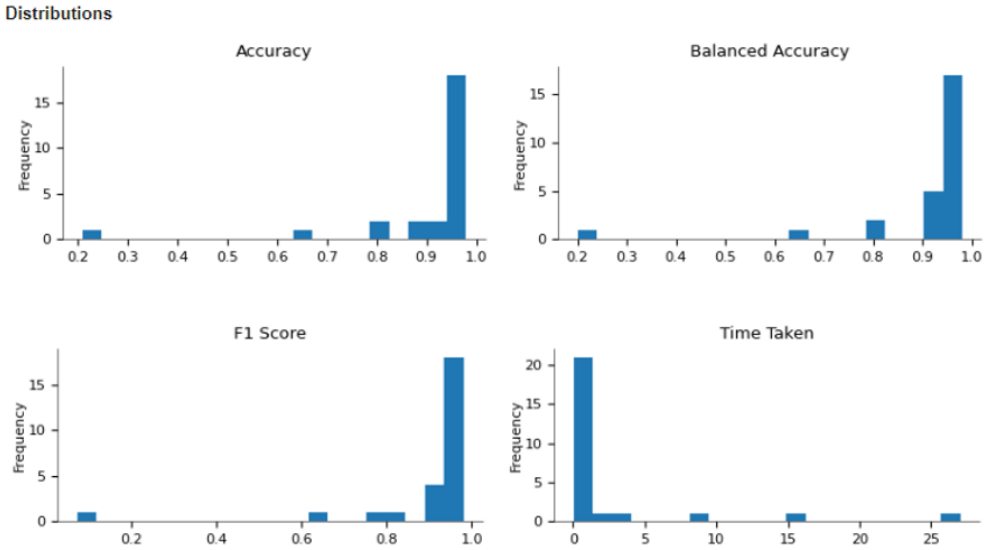

Figure 3. Frequency distribution for performance metrics and time computations

Another interesting part of the lazy predicts is the computation time for each model. Thus, user not only can assess and compare each model based on the performance metrics, but also the computation time which critical in the production of machine learning project. Among all the models, the longest computation time is given to NuSVC model which took 27.02 seconds, followed by Label Spreading that took 16.13 seconds. Meanwhile, there are some classifiers with fast time computation, such as Dummy Classifier (0.02 s), Ridge Classifier (0.03 s), Bernoulli NB (0.03 s), Quadratic Discriminant Analysis (0.03 s), and Extra Tree (0.03 s). **Figure 3** also portrays that most of the model has time computations between 0-1 seconds.

Based on this experiment, using lazy predict to compare the model performance in classifying the LULC datasets, there are some benefits of using lazy predict to generate machine learning supervised classifiers:

a. Lazy predict library is accessible by anyone (open source) and less time-consuming to generate various machine learning models in only few lines of code.
b. The model computation time is relatively faster than regular machine learning method.
c. The code for building lazy predict is easy to understand.
d. Lazy predict provides the performance metrics and time computation for each model so that user can assess and choose the best optimal model based on these informations.

While lazy predict has benefits, it also comes with some drawbacks which must consider. These are the disadvantages of using lazy predict for creating the machine learning model:

a. The model generated by lazy predict, by default, only primarily used for quick and automated model selection and evaluation. This means the result may not be used for the production purposes.

b. Lazy predict cannot be used in the local environment. The installed version of lazy predict in local machine, somehow, cannot be used properly. Alternatively, running the lazy predict experiment can be conducted in Google Colab.
c. Lazy predict requires a proper memory (RAM) computation. As a result, it is not suitable for big datasets. In this paper, running lazy predict in Google Colab (free GPU tiers) can only be conducted by using 25.000 row datasets, from 75.000 row datasets collected in the training sample.

4. Conclusion:

Lazy predict library can be utilized to generate various LULC classifiers model in efficient few lines of code and less time-consuming. The result of this experiment shows lazy predict library can create multiple 26 classifier models. Most LULC model created by lazy predict have the performance metrics above 90% and time computation 0-1 seconds. While lazy predict has benefits in providing a good information for comparing various machine learning models and selecting the best model, it has drawbacks in term of its feasibility for the machine learning production, its obstacle running in local environment, and its requirements for the RAM computation.

References:

[1] S. Talukdar *et al.*, "Land-Use Land-Cover Classification by Machine Learning Classifiers for Satellite Observations—A Review," *Remote Sens.*, vol. 12, no. 7, p. 1135, Apr. 2020, doi: 10.3390/rs12071135.

[2] S. Basheer *et al.*, "Comparison of Land Use Land Cover Classifiers Using Different Satellite Imagery and Machine Learning Techniques," *Remote Sens.*, vol. 14, no. 19, p. 4978, Oct. 2022, doi: 10.3390/rs14194978.

[3] S. Aldiansyah and R. A. Saputra, "Comparison Of Machine Learning Algorithms For Land Use And Land Cover Analysis Using Google Earth Engine (Case Study: Wanggu Watershed)," *Int. J. Remote Sens. Earth Sci.*, vol. 19, no. 2, p. 197, Jan. 2023, doi: 10.30536/j.ijreses.2022.v19.a3803.

[4] Y. G. Yuh, W. Tracz, H. D. Matthews, and S. E. Turner, "Application of machine learning approaches for land cover monitoring in northern Cameroon," *Ecol. Inform.*, vol. 74, p. 101955, May 2023, doi: 10.1016/j.ecoinf.2022.101955.

[5] E. Alshdaifat, D. Alshdaifat, A. Alsarhan, F. Hussein, and S. M. F. S. El-Salhi, "The Effect of Preprocessing Techniques, Applied to Numeric Features, on Classification Algorithms' Performance," *Data*, vol. 6, no. 2, p. 11, Jan. 2021, doi: 10.3390/data6020011.

[6] Y. O. Ouma, A. Keitsile, B. Nkwae, P. Odirile, D. Moalafhi, and J. Qi, "Urban land-use classification using machine learning classifiers: comparative evaluation and post-classification multi-feature fusion approach," *Eur. J. Remote Sens.*, vol. 56, no. 1, Dec. 2023, doi: 10.1080/22797254.2023.2173659.

[7] A. Jamali, "Land use land cover mapping using advanced machine learning classifiers," *Ekológia (Bratislava)*, vol. 40, no. 3, pp. 286–300, Sep. 2021, doi: 10.2478/eko-2021-0031.

[8] S. Swetanisha, A. R. Panda, and D. K. Behera, "Land use/land cover classification using machine learning models," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 2, p. 2040, Apr. 2022, doi: 10.11591/ijece.v12i2.pp2040-2046.

[9] A. E. Eldin Rashed, A. M. Elmorsy, and A. E. Mansour Atwa, "Comparative evaluation of automated machine learning techniques for breast cancer diagnosis," *Biomed. Signal Process. Control*, vol. 86, p. 105016, Sep. 2023, doi: 10.1016/j.bspc.2023.105016.

[10] K. Nidhul, "Enhanced thermo-hydraulic performance in a V-ribbed triangular duct solar air heater: CFD and exergy analysis," *Energy*, vol. 200, 2020, doi: 10.1016/j.energy.2020.117448.

[11] S. W. Sharshir, "Performance enhancement of stepped double slope solar still by using nanoparticles and linen wicks: Energy, exergy and economic analysis," *Appl. Therm. Eng.*, vol. 174, 2020, doi: 10.1016/j.applthermaleng.2020.115278.

[12] M. E. Zayed, "Performance prediction and techno-economic analysis of solar dish/stirling system for electricity generation," *Appl. Therm. Eng.*, vol. 164, 2020, doi: 10.1016/j.applthermaleng.2019.114427.

[13] P. Sharma, "Performance analysis of deep learning CNN models for disease detection in plants using image segmentation," *Inf. Process. Agric.*, vol. 7, no. 4, pp. 566–574, 2020, doi: 10.1016/j.inpa.2019.11.001.

[14] S. Rahman, "Performance analysis of boosting classifiers in recognizing activities of daily living," *Int. J.*

Environ. Res. Public Health, vol. 17, no. 3, 2020, doi: 10.3390/ijerph17031082.

[15] T. K. Nguyen *et al.*, "Machine learning-based screening of MCF-7 human breast cancer cells and molecular docking analysis of essential oils from Ocimum basilicum against breast cancer," *J. Mol. Struct.*, vol. 1268, p. 133627, Nov. 2022, doi: 10.1016/j.molstruc.2022.133627.

[16] E. Alshdaifat , D. Alshdaifat, A. Alsarhan, F. Hussein and S. M. F. El Salhi, "The Effect of Preprocessing Techniques, Applied to Numeric Features, on Classification Algorithms' Performance," *Data*, vol. 6, no. 2, p. 11, 2021. https://doi.org/10.3390/data6020011.

[17] A. Tharwat, "Classification assessment methods," *Appl. Comput. Informatics*, vol. 17, no. 1, pp. 168–192, Jan. 2021, doi: 10.1016/j.aci.2018.08.003.

[18] D. Kim, "Classification of surface settlement levels induced by TBM driving in urban areas using random forest with data-driven feature selection," *Autom. Constr.*, vol. 135, 2022, doi: 10.1016/j.autcon.2021.104109.

[19] H. Nhat-Duc, "Comparison of histogram-based gradient boosting classification machine, random Forest, and deep convolutional neural network for pavement raveling severity classification," *Autom. Constr.*, vol. 148, 2023, doi: 10.1016/j.autcon.2023.104767.

[20] M. Mafarja, "Classification framework for faulty-software using enhanced exploratory whale optimizer-based feature selection scheme and random forest ensemble learning," *Appl. Intell.*, vol. 53, no. 15, pp. 18715–18757, 2023, doi: 10.1007/s10489-022-04427-x.

[21] O. S. Djandja, "Random forest-based modeling for insights on phosphorus content in hydrochar produced from hydrothermal carbonization of sewage sludge," *Energy*, vol. 245, 2022, doi: 10.1016/j.energy.2022.123295.

[22] M. Salem, "Random Forest modelling and evaluation of the performance of a full-scale subsurface constructed wetland plant in Egypt," *Ain Shams Eng. J.*, vol. 13, no. 6, 2022, doi: 10.1016/j.asej.2022.101778.

[23] A. A. Ewees, "Performance analysis of Chaotic Multi-Verse Harris Hawks Optimization: A case study on solving engineering problems," *Eng. Appl. Artif. Intell.*, vol. 88, 2020, doi: 10.1016/j.engappai.2019.103370.

[24] B. Cao, "Performance analysis and comparison of PoW, PoS and DAG based blockchains," *Digit. Commun. Networks*, vol. 6, no. 4, pp. 480–485, 2020, doi: 10.1016/j.dcan.2019.12.001.

[25] A. Das, "Assessment of peri-urban wetland ecological degradation through importance-performance analysis (IPA): A study on Chatra Wetland, India," *Ecol. Indic.*, vol. 114, 2020, doi: 10.1016/j.ecolind.2020.106274.

[26] D. İzci, "Comparative performance analysis of slime mould algorithm for efficient design of proportional–integral–derivative controller," *Electrica*, vol. 21, no. 1, pp. 151–159, 2021, doi: 10.5152/ELECTRICA.2021.20077.