IJAIT (International Journal
of Applied Information Technology)

http://journals.telkomuniversity.ac.id/ijait/

Simple Machine Learning Architecture as a Service
Case Study: Gender Prediction

Tora Fahrudin [a, *], Dedy Rahman Wijaya [a]

[a] *School of Applied Science, Telkom University, Indonesia*
torafahrudin@telkomuniversity.ac.id, dedyrw@telkomuniversity.ac.id

ARTICLE INFO

Received May 9th, 2023
Revised May 31st, 2023
Accepted June 9th, 2023
Available online July 28th, 2023

Keywords
Machine learning, cloud, machine learning architecture as service, gender prediction model

ABSTRACT

Machine learning (ML) development starting in the 1950s, has shown significant progress. Various fields have used machine learning as an information system element that is useful in assisting data processing, personalization, prediction, and performing anomaly detection of occurring transactions. Along with developments, cloud-based machine learning technology is becoming the choice for ease of implementation and connectivity with various other technology platforms. This paper proposes a machine learning architecture as a service (MLaaS) implemented in a case study of a gender prediction model based on height and weight. The results show that the MLaaS architecture is straightforward to implement and fits the needs of various access environments and the ease of updating models centrally. Our gender prediction model achieved 91.78% in the precision, recall, and F1-score, 91.8% in specificity and NPV, and 91.79% in accuracy.

* Corresponding author at:
School of Applied Science, Telkom University
Jl. Telekomunikasi No. 1, Terusan Buah Batu, Bandung, 40257
Indonesia
E-mail address: torafahrudin@telkomuniversity.ac.id

ORCID ID:
- First Author: 0000-0001-9600-0192
- Second Author: 0000-0003-0351-7331

https://doi.org/10.25124/ijait.v7i01.5991

1. Introduction

Machine learning is a machine related to computational statistics and specializes in making predictions [1]. Nowadays, machine learning plays essential roles in many sectors, such as health [2], farming [3], economics [4], education [5], games [6], and also social [7]. Various data processing technologies using machine learning have developed in such a way. Many algorithms, from traditional to modern, have been developed and implemented, such as SVM [8], Deep Learning [9], Linear Regression [10], and Random Forest [11].

As part of the new incoming technologies [12], cloud computing also contributed to the effectiveness of machine learning implementation. The power of cloud computing in machine learning implementations is meager cost, high performance, and high availability [13]. There are three main types of services available in the cloud: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [14].

Amazon Web Services (AWS) Machine Learning, Google Cloud Machine Learning Engine, Microsoft Azure Machine Learning, IBM Watson Machine Learning, and Databricks are famous platforms in cloud computing.

1.1. Current Challenges

In small enterprises, the price of cloud computing services is still expensive. It also has transaction cost theory: management costs, meta-services, and business process reengineering costs, the uncertainty of managing contracts, investing in monitoring, and high transaction frequency, which compensates for the needed investment [15]. So, in this paper, we proposed a simple framework to implement MLaaS in a gender prediction study case.

1.2. Previous Works

In this section, the secondary studies found that implementing MLaaS is still challenging for researchers. The utilization of ML techniques in the cloud-integrated computing paradigms is the trend; Artificial Neural Networks (ANN), Support Vector Machine (SVM), Linear Regression (LR), Deep Neural Networks (DNN), Random Forest (RF), Decision Tree (DT) and many other algorithms has successfully implemented as ML techniques used in integrated cloud computing [16].

Besides focusing on algorithms such as workload prediction using Deep Learning in Cloud Computing [17], another research also focused on a novel architecture for a scalable, flexible, and non-blocking machine learning as a service based on a service component architecture (SCA) and focusing on predictive modeling using Model-View-Controller [18]. To the best of our knowledge, the paper does not give a detailed explanation of the architecture and library which is used. So, in this paper, we provided a simple architecture and library in more detail.

We implement a gender prediction model based on height and weight data. This study case complements similar research such as gender prediction based on their path in shopping [19], neuroimage data [20], real-time facial images [21], and open domain text [22]. In this manuscript, we implemented simple gender prediction models from weight and height datasets using a machine learning algorithm.

The rest of the paper is organized as follows: Section 2 describes the material and methods, Section 3 presents the result and discussion, whereas Section 4 presents the conclusion.

2. Material and Methods

2.1. Material

In this section, we provide a list of all materials that we used. We divided the kinds of material into three parts:

1. Building Model. The prediction model was developed from Python using Scikit-learn (sklearn) [23]. Library sklearn is one of the Python libraries used to perform Machine Learning tasks. Sklearn provides various algorithms for data classification, regression, grouping, and dimensionality reduction. Sklearn also provides features for data pre-processing, such as normalization, transformation, and feature selection. Sklearn also provides a module for model evaluation, such as cross-validation and measurement of accuracy, precision, and recall metrics. The dataset used to build a model was a dummy dataset from the Kaggle repository data set [24], containing weight and height from 500 male and 500 female data. The model generated by our experiment will be implemented as MLaaS using Flask, sklearn library, and PHP language. Gender prediction based on weight and height can involve privacy issues. While weight and height are easily measurable physical information, associating them directly with a person's gender can raise issues of sensitivity and privacy. The gender prediction model is not to draw conclusions or assumptions about their identity based solely on weight and height, but this prediction model is only for showing the approximation from our models of what gender it must be.
2. Web Services (Flask). Flask is a minimalistic web framework written in Python [25]. Flask is designed to make it easy to build simple, flexible web applications that can be integrated with various databases and development tools. This web service handles user requests via API, load model, and making predictions and gives the prediction response in JSON format. This flask was configured in default configuration ports (5000).
3. API Client. The client on the API is a program or application that utilizes the services an API provides. In this case, the API is a service provider that clients can access to perform certain operations. The client acts as an API user requesting access to resources or services provided by the API. The client sends requests to the API via HTTP requests, such as GET, POST, PUT, and DELETE. This request contains the information the API needs to identify the action requested by the client. In this context, the API Client gets prediction results from Web Service and is displayed in the client platform, either web-based, mobile, or desktop applications.

The simulation architecture was done in x-64 Windows 11 with Intel core i-7 and 8 GB RAM. The version of Python is 3.7.4.

2.2. Methods

In this section, we explain the method for determining the architecture of MLaaS. Figure ***1*** shows the simple architecture of MLaaS in a gender prediction case study. The prediction model was built using sklearn in Python. Figure 2 shows an experimental design to get the best model by comparing three algorithms with hyperparameter optimization. We use Decision Tree, Random Forest, and also Adaboost. Table 1 shows the hyperparameter optimization setting to find the right combination and achieve maximum performance. The development phase was done using jupyter notebook, and the model was saved into a save format using the Pickle library. The pickle library is one of the libraries in Python that is used to convert Python objects into a format that can be stored or transmitted over the network. Python objects can be saved to a file in byte format, then read back and converted into Python objects. Usually, the machine learning model can be saved (using the dump command) and loaded using this library (load).

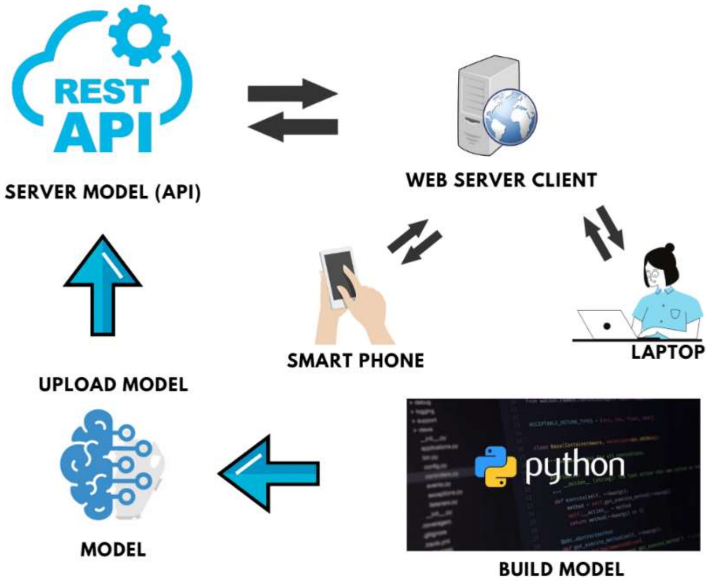

Figure 1 Simple Architecture of MLaaS in Gender Prediction Case

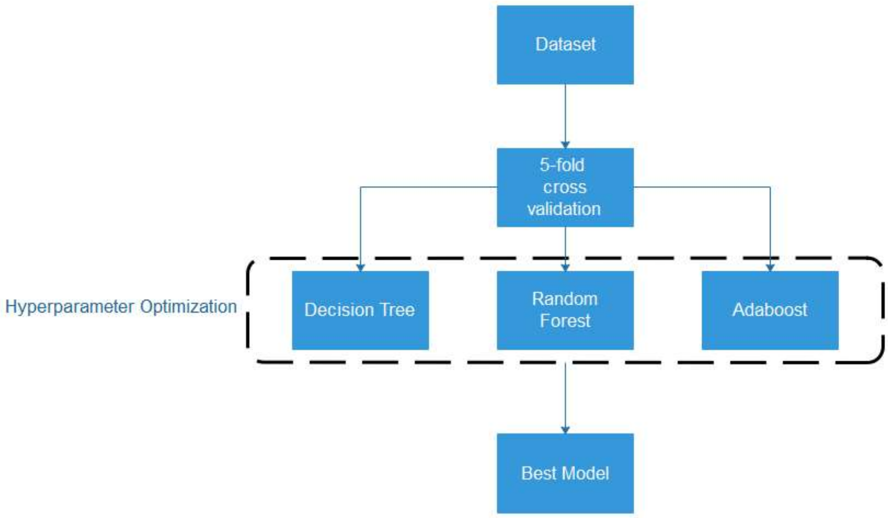

Figure 2 Experimental Design

Table 1 Hyperparameter Optimization of Three Algorithms

Algorithms	Parameter	Parameter Value
Decision Tree	max_depth min_samples_split	3,5,10 2,5,10
Random Forest	n_estimators max_depth	100,200,300 3,5,10
Adaboost	n_estimators learning_rate	100,200,300 0.1,0.5,1.0

After the model was saved into the same format, that model needed to be saved in one directory, which was later loaded into Flask web services. The objective of the flask is to load the model, wait for the prediction request, predict the result from the model, and format the prediction results into json format back to the requester.

```
import os

dir_path = os.path.dirname(os.path.realpath(__file__))
model_jk_classification = dir_path+'/model/jenis_kelamin_model.sa

global classifier
classifier_jk = joblib.load(model_jk_classification)

app = Flask(__name__)

#prediksi jenis kelamin
@app.route('/prediksi_jk', methods = ['POST'])
def disp4():
    content_type = request.headers.get('Content-Type')
    if (content_type == 'application/json'):
        json_req = request.json
        data=[]
        for i in json_req['data']:
            data.append(i)
        df = pd.DataFrame(data, columns = ['Tinggi','Berat'])
        hasil = pd.DataFrame(columns = ['class'])
        kelas = classifier_jk.predict(data)
        for i in kelas:
            new_row = {'class':i}
            hasil = hasil.append(new_row,ignore_index=True)
        json_rep = hasil.to_json(orient='index')
        return json_rep
    else:
        return 'Content-Type not supported!'

if __name__ == '__main__':
   app.run()
```

Figure 3 Simple Code for Loading The Model and Response API Request

```
(musiman) PS D:\Konfigurasi_Program\Rest API Python\API\API> python .\modelprediksijk.py
 * Serving Flask app 'modelprediksijk'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a proc
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [09/May/2023 14:08:15] "POST /prediksi_jk HTTP/1.1" 200 -
```

Figure 4 Log Request in Flask App

This architecture is inspired by Azured machine learning, where the end point of machine learning models can be accessed through various platforms. This request process was done by using the curl mechanism in php languages. It can also be implemented in many PHP frameworks like Laravel, Code Igniter, and Node Js.

Figure 5 Gender Prediction Interface

2.3. Evaluation Metrics

Generally, the classification model must be evaluated by determining precision / positive prediction value (Equation 1), recall/sensitivity (Equation 2), specificity / true negative rate (Equation 3), negative predictive value (NPV) (Equation 3), accuracy (Equation 4), and F1-score (Equation 5). Precision is the degree to which positive predictions made by the model are correct. Recall measures the degree to which the model correctly identifies all true positive samples. Specificity measures the degree to which the model correctly identifies all true negative samples. NPV measures the extent to which the negative predictions made by the model are correct. Accuracy measures the extent to which the model correctly predicts the overall positive and negative classes. Lastly, F1-score combines precision and recall into one score that reflects the balance between the two. The F1-score is the harmonic average between precision and recall and assigns equal weight to both.

$$Precision = \frac{TP}{(TP+FP)} \quad \text{Equation 1}$$

$$Recall = \frac{TP}{(TP+FN)} \quad \text{Equation 2}$$

$$Specificity = \frac{TN}{(TN+FP)} \quad \text{Equation 3}$$

$$NPV = \frac{TN}{(TN+FN)} \quad \text{Equation 4}$$

$$Accuracy = \frac{TN+TP}{(TP+TN+FP+FN)} \quad \text{Equation 5}$$

3. Result and Discussion

Table 2 shows the resulting classification outcomes based on the best parameter for each algorithm. We can see that Random Forest is superior compared to Decision Tree and Adaboost in all performance metrics. Random Forest performed best in the three hundred number estimator and five depth of tree parameters. The accuracy obtained from the three algorithms shows that the resulting model can predict gender based on weight and height somewhat because the accuracy values of all matrices are above 90% of all algorithms.

Table 2 Best Result for Each Algorithm

Algorithms	Best Parameter	Precision	Recall	Specificity	NPV	Accuracy	F-1 Score
Decision Tree	max_depth : 5 min_samples_split: 2	91.5%	91.49%	91.49%	91.5%	91.5%	91.49%
Random Forest	n_estimators: 300 max_depth : 5	**91.78%**	**91.78%**	**91.8%**	**91.8%**	**91.79%**	**91.78%**
Adaboost	n_estimators: 200 learning_rate: 0.5	91.76%	91.71%	91.77%	91.76%	91.77%	91.76%

After we get the best model, the model is saved and used in MLaaS architecture. Figure 3 shows a simple code implementation using Flask as MLaaS architecture. It contains the model path, and model load, gets the request, predicts the result, and transforms the prediction into json format. Every request was logged into the flask screen as shown in Figure 4. The implementation of the interface in the client was shown in Figure 5. Equation 6 is the prediction result format for 45 weight and 150 height request data. The class "L" represents male, and "P" represents female. So, in this case, the model predicts that the user’s gender is female.

array(1) { [0]=> array(1) { ["class"]=> string(1) "P" } } Equation 6

The prediction can also be conducted in many rows. The prediction results are also returned by web services in a json array. So it depends on the requirement. Another advantage of using this architecture is as follows:

1. Scalability: if the user access increased, it scaled up the capacity.
2. Effectiveness: the model can be updated centrally, so the client hardware does not need to be updated individually. In the IOT platform, this architecture is very suitable.
3. Collaboration: the computational workload of a job can be separated into different platforms and machines.

With these advantages, the MLaaS architecture with a centralized machine learning model is a popular solution for implementing machine learning models in modern applications, such as web and mobile applications.

4. Conclusions

This study aimed to implement MLaaS in simple ways for predicting gender based on height and weight. This architecture contains some parts, such as a machine learning development model using sklearn, web services using Flask to load the model and process the request to get a prediction, and the front end of applications to display the prediction results using PHP language and curl mechanisms. The results show that the MLaaS architecture is very easy to implement and appropriate in various access environments and the ease of updating models centrally. The gender predictions model was successfully implemented by using Random Forest in n_estimators: 300 and max_depth: 5. That algorithm achieved 91.78% precision, recall, and F1-score, 91.8% in specificity and NPV, and 91.79% accuracy.

Bibliography

[1] N. Thomas Rincy and R. Gupta, "A Survey on Machine Learning Approaches and Its Techniques:" in *2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science, SCEECS 2020*, Institute of Electrical and Electronics Engineers Inc., Feb. 2020. doi: 10.1109/SCEECS48394.2020.190.

[2] M. Javaid, A. Haleem, R. Pratap Singh, R. Suman, and S. Rab, "Significance of machine learning in healthcare: Features, pillars and applications," *International Journal of Intelligent Networks*, vol. 3, pp. 58–73, Jan. 2022, doi: 10.1016/j.ijin.2022.05.002.

[3] R. Togneri, R. Prati, H. Nagano, and C. Kamienski, "Data-driven water need estimation for IoT-based smart irrigation: A survey," *Expert Systems with Applications*, vol. 225. Elsevier Ltd, Sep. 01, 2023. doi 10.1016/j.eswa.2023.120194.

[4] A. Prasad and A. Seetharaman, "Importance of Machine Learning in Making Investment Decision in Stock Market," *Vikalpa*, vol. 46, no. 4, pp. 209–222, Dec. 2021, doi: 10.1177/02560909211059992.

[5] M. E. Webb *et al.*, "Machine learning for human learners: opportunities, issues, tensions, and threats," *Educational Technology Research and Development*, vol. 69, no. 4, pp. 2109–2130, Aug. 2021, doi: 10.1007/s11423-020-09858-2.

[6] G. Edwards, N. Subianto, D. Englund, and J. W. Goh, "The Role of Machine Learning in Game Development Domain - A Review of Current Trends and Future Directions," in *Proceedings of the Digital Image Computing: Techniques and Applications (DICTA)*, Australia: IEEE, 2021, pp. 1–7.

[7] R. Serban, A. Kupraszewicz, and G. Hu, "Predicting the characteristics of people living in the South USA using logistic regression and decision tree," *IEEE International Conference on Industrial Informatics (INDIN)*, pp. 688–693, 2011, doi: 10.1109/INDIN.2011.6034974.

[8] Y. Lai, "A Comparison of Traditional Machine Learning and Deep Learning in Image Recognition," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Nov. 2019. doi 10.1088/1742-6596/1314/1/012148.

[9] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Market*, vol. 31, no. September, pp. 685–695, 2021, doi: 10.1007/s12525-021-00475-2/Published.

[10] S. J. Kim, S. J. Bae, and M. W. Jang, "Linear Regression Machine Learning Algorithms for Estimating Reference Evapotranspiration Using Limited Climate Data," *Sustainability (Switzerland)*, vol. 14, no. 18, Sep. 2022, doi 10.3390/su141811674.

[11] M. Schonlau and R. Y. Zou, "The random forest algorithm for statistical learning," *Stata Journal*, vol. 20, no. 1, pp. 3–29, Mar. 2020, doi: 10.1177/1536867X20909688.

[12] H. Alzahrani, "A Brief Survey of Cloud Computing," *Global Journal of Computer Science and Technology*, vol. 16, no. 3, 2016.

[13] A. Salim, Juliandry, L. Raymond, and J. V Moniaga, "General pattern recognition using machine learning in the cloud," *Procedia Comput Sci*, vol. 216, pp. 565–570, 2023, doi: 10.1016/j.procs.2022.12.170.

[14] N. Dimitri, "Pricing cloud IaaS computing services," *Journal of Cloud Computing*, vol. 9, no. 1, Dec. 2020, doi: 10.1186/s13677-020-00161-2.

[15] R. Makhlouf, "Cloudy transaction costs: a dive into cloud computing economics," *Journal of Cloud Computing*, vol. 9, no. 1, Dec. 2020, doi: 10.1186/s13677-019-0149-4.

[16] D. Soni and N. Kumar, "Machine learning techniques in emerging cloud computing integrated paradigms: A survey and taxonomy," *Journal of Network and Computer Applications*, vol. 205, p. 103419, Sep. 2022, doi: 10.1016/j.jnca.2022.103419.

[17] Z. Ahamed, M. Khemakhem, F. Eassa, F. Alsolami, and A. S. A. M. Al-Ghamdi, "Technical Study of Deep Learning in Cloud Computing for Accurate Workload Prediction," *Electronics (Switzerland)*, vol. 12, no. 3, Feb. 2023, doi 10.3390/electronics12030650.

[18] M. Ribeiro, K. Grolinger, and M. A. M. Capretz, "MLaaS: Machine learning as a service," in *Proceedings - 2015 IEEE 14th International Conference on Machine Learning and Applications, ICMLA 2015*, Institute of Electrical and Electronics Engineers Inc., Mar. 2016, pp. 896–902. doi 10.1109/ICMLA.2015.152.

[19] O. Dogan and B. Oztaysi, "Genders prediction from indoor customer paths by Levenshtein-based fuzzy kNN," *Expert Syst Appl*, vol. 136, pp. 42–49, Dec. 2019, doi 10.1016/j.eswa.2019.06.029.

[20] Y. Bi, A. Abrol, Z. Fu, J. Chen, J. Liu, and V. Calhoun, "Prediction of gender from longitudinal MRI data via deep learning on adolescent data reveals unique patterns associated with brain structure and change over two years," *J Neurosci Methods*, vol. 384, p. 109744, Jan. 2023, doi: 10.1016/j.jneumeth.2022.109744.

[21] B. Abirami, T. S. Subashini, and V. Mahavaishnavi, "Gender and age prediction from real-time facial images using CNN," in *Materials Today: Proceedings*, Elsevier Ltd, 2020, pp. 4708–4712. doi: 10.1016/j.matpr.2020.08.350.

[22] E. E. Abdallah, J. R. Alzghoul, and M. Alzghool, "Age and Gender prediction in Open Domain Text," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 563–570. doi: 10.1016/j.procs.2020.03.126.

[23] Anonymous, “Sklearn,” *2021*.

[24] P. Shrestha, "Gender classification: Based on Height and Weight," *Kaggle*, 2019.

[25] Pallets, "Flask," Sphinx 4.5.0., 2010. https://flask.palletsprojects.com/en/2.3.x/#user-s-guide (accessed May 09, 2023).