BAREKENG: Journal of Mathematics and Its Applications
June 2025 Volume 19 Issue 2 Page 1421-1432
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol19iss2pp1421-1432

APPLICATION AND PERFORMANCE COMPARISON OF MULTI-OUTPUT MACHINE LEARNING FOR NUMERICAL-NUMERICAL AND NUMERICAL-CATEGORICAL OUTPUTS

Karin Joan[1], Robyn Irawan[2], Benny Yong[3*]

[1,2,3]*Center for Mathematics and Society, Faculty of Science, Parahyangan Catholic University*
Jln. Ciumbuleuit No. 94, Bandung, 40141, Indonesia

*Corresponding author's e-mail: * **benny_y@unpar.ac.id***

Article History:

Received: 30th November 2024
Revised: 2nd February 2025
Accepted: 8th March 2025
Published: 1st April 2025

Keywords:

Logistic Regression;
Multi-Output Machine Learning;
Multivariate Regression Tree;
Multivariate Random Forest;
Multi-Output Neural Network.

ABSTRACT

Multi-Output Machine Learning is an advancement of traditional machine learning, designed to predict multiple output variables simultaneously while considering the relationships between these output variables. Multi-Output Machine Learning is essential as a decision support tool because decision-making in many problems generally considers multiple factors. The use of Multi-Output Machine Learning is more advantageous than conventional machine learning in terms of time efficiency, addressing data limitations, and ease of maintenance. These benefits will significantly impact cost savings for industries utilizing Big Data. The models used in this research include Multivariate Regression Tree, Multivariate Random Forest, and Multi-Output Neural Network. The Multivariate Regression Tree and Multivariate Random Forest are developed by modifying the splitting function using Mahalanobis distance. The topological changes introducing shared and private hidden layers are the key development of the Multi-Output Neural Network. The prediction results indicated a trade-off in error between two output variables when comparing the Multivariate Regression Tree and Multivariate Random Forest with their single output counterparts. Meanwhile, the Multi-Output Neural Network model successfully improved the prediction results for both output variables. This research also introduces Mixed Multi-Output Machine Learning, which can predict numerical and categorical output variables. The Mixed Multi-Output Machine Learning model utilizes the logit values from the Logistic Regression model to extend the range of prediction results beyond the 0 to 1 interval. Multi-Output Neural Network is the sole model that produces predictions with relatively small errors and high accuracy values.

How to cite this article:

K. Joan, R. Irawan and B. Yong., "APPLICATION AND PERFORMANCE COMPARISON OF MULTI-OUTPUT MACHINE LEARNING FOR NUMERICAL-NUMERICAL AND NUMERICAL-CATEGORICAL OUTPUTS," *BAREKENG: J. Math. & App.*, vol. 19, iss. 2, pp. 1421-1432, June, 2025.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

Decision-making is one of the most crucial parts of every business flow to achieve set targets. Decisions are usually made by considering multiple factors based on data. For example, the success of a digital marketing campaign on social media can be measured by program awareness and the revenue generated by the campaign. Machine learning has significantly aided the decision-making process [1]. However, traditional machine learning (hereinafter called single-output machine learning) can only predict one variable, whether a numerical or categorical output. Therefore, multi-output machine learning has been developed from single-output machine learning to predict multiple outputs simultaneously. Using multi-output machine learning offers benefits such as reducing computational time and resources. Additionally, maintaining one multi-output machine learning model is easier than maintaining multiple single-output models since adjustments to fit business interests may be required [2].

Prior to this paper, other researchers have already developed multi-output machine learning models, such as Regression Trees and Neural Networks. A study by Abyaneh used Multivariate Linear Regression and Multi-Output Neural Networks to determine air quality in Iran by predicting the values of Biochemical Oxygen Demand (BOD) and Chemical Oxygen Demand (COD) [3]. In this case, multi-output machine learning was a helpful tool for the research since the measurement process for COD would be costly. Schmid et al. studied the performance of multivariate and univariate random forests [4]. In another study, Multivariate Regression Trees, Multivariate Random Forests, and Multi-Output Neural Networks were used to predict the cost of treatment and length of stay in a hospital [5].

Both cases mentioned above involved predicting multiple numerical output variables of the same type. However, in many cases, the output variables that need to be predicted include numerical and categorical types. The algorithms used to predict numerical and categorical variables differ, so they couldn't be blended bluntly. For instance, Regression Trees use sum squared error as the loss function, while Classification Trees use Gini impurity as the loss function. These loss functions cannot be combined directly due to differences in the units of measurement. In a previous study, Lu He et al. attempted to predict both types of variables by converting the categorical variable into a binary variable and combining the loss functions. However, the numerical output was not accurate [6].

In this paper, the solution to predict output variables of different types will be explored with the help of logistic regression. The categorical output will be further converted using logistic regression. The numerical variable will be predicted along with the logit value of logistic regression for each class. The class of the categorical variable will be determined by converting each logit value back into probability values, with the class having the highest probability being chosen as the predicted class.

2. RESEARCH METHODS

Machine learning models such as Regression Trees, Random Forests, and Neural Networks will form the basis of multi-output machine learning, which is used to predict multiple output variables simultaneously. The case study will compare the accuracy of predictions between multi-output machine learning and its single-output counterparts.

2.1 Regression Tree

A Regression Tree is a machine learning algorithm designed for numerical output, aiming to divide observations into several homogeneous partitions by passing them through nodes [7]. The structure of a Regression Tree is shown in **Figure 1**. The nodes are divided into three types: root nodes, internal nodes, and leaf nodes. The root node is the first node that divides the observations into two large groups. The internal node is the node that further divides the observations into several partitions. A leaf node is the final stop where the observations are no longer divided and become the partition itself.

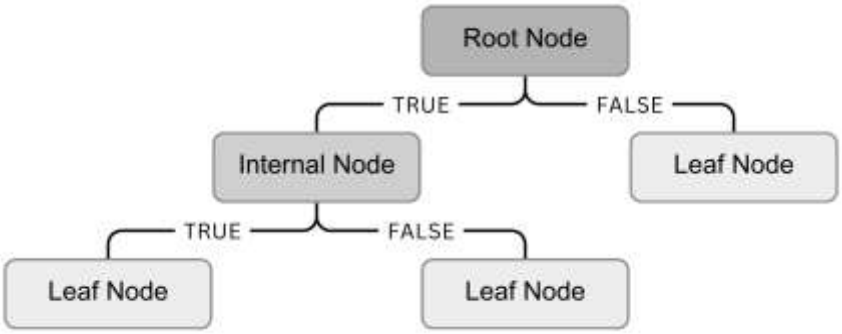


Figure 1. Topology of Decision Tree

At each node, the observation is directed to the right node if it meets the condition of the Boolean expression; otherwise, it is sent to the left node. The Boolean expressions at each node are determined by finding the partition with the smallest error, which is calculated using the equation below:

$$SSE_c = \sum_{I=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

where y_i and $\hat{y}_i$ are the actual and predicted values of the i^{th} observation, respectively. The predicted value is obtained from the average of the observations output within the same partition with the lowest SSE in **Equation (1)**. The tree continues to divide until it meets the predefined stopping criterion.

2.2 Multivariate Regression Tree

The Multivariate Regression Tree is an improvement of the original Regression Tree algorithm, meant to predict multiple output variables simultaneously. The mechanism of the Multivariate Regression Tree is similar to that of the Regression Tree, except for the splitting function. Mahalanobis distance is used as the splitting function, which calculates the distance between two groups of multivariate data, and the squared Mahalanobis distance is stated as:

$$\Delta^2 = (\boldsymbol{\mu_1} - \boldsymbol{\mu_2})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu_1} - \boldsymbol{\mu_2}) \tag{2}$$

where $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$ are the vectors containing the measurement values from 2 different groups, and $\boldsymbol{\Sigma}$ is the non-singular variance covariance matrix. To prevent the singularity of $\boldsymbol{\Sigma}$ matrix, De'Ath proposed a new splitting function **[8]** where $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ from **Equation (2)** are substituted by $\boldsymbol{y}_i$ and $\hat{\boldsymbol{y}}_i$ which they are, respectively, the vectors containing actual output variables and predicted output variables. For simplification as well, the matrix $\boldsymbol{\Sigma}$ is substituted with identity matrix, I, and with some algebraic simplification causing the splitting function is formulated as follows:

$$SSE_c = \sum_{I=1}^{n}\sum_{j=1}^{m}\left(y_{ij} - \widehat{y_{ij}}\right)^2 \tag{3}$$

where y_{ij} and $\widehat{y_{ij}}$ are sequentially the actual and predicted value of the j^{th} output variables of the i^{th} observation. In this method, each output variable carries the same weight in affecting the sum of squared errors. To avoid scale disparities that could lead to imbalances in the calculation of the sum of squared errors by **Equation (3)**, each output variable should be normalized first. Similar to the Regression Tree, the split with the smallest sum of squared errors becomes the boolean expression that divides the data into two partitions. The prediction for each output variable is calculated independently by averaging the observations within the same partition.

2.3 Random Forest

A single Regression Tree has a high chance of overfitting because it tries to fit the training data too precisely. To address this issue, Breiman proposed constructing an ensemble of Regression Trees by sampling the training data using a bootstrapping **[9]**. This sampling method allows for the duplication of observations during the randomization process. In this way, each observation will have a chance to be included in the

training data, making the model more generalized. Each bootstrap data set is used to develop one tree using the same construction algorithm as the Regression Tree. Meanwhile, the data not included in a bootstrap sample become out-of-bag data. The set of several Regression Trees forms a model called Random Forest, where the topology of this model is shown in **Figure 2**.

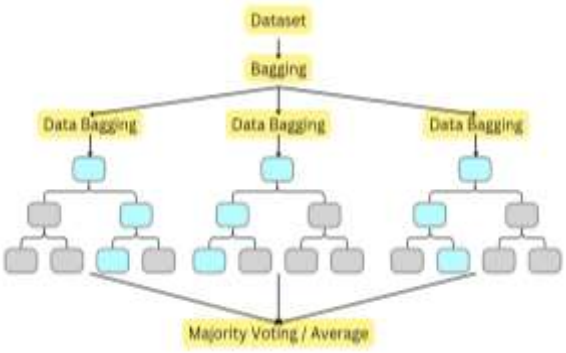


Figure 2. Topology of Random Forest

Each tree will predict the output variables independently. The final prediction is calculated by averaging the predictions from all the trees, as defined by the **Equation (4)** below:

$$\widehat{y_i} = \frac{1}{T}\left(\sum_t^T \widehat{y_{it}}\right) \tag{4}$$

The out-of-bag data that are not selected during bootstrapping are used to calculate feature importance, which helps determine which input variables contribute most to the construction of the Random Forest model **[10]**. Feature importance is calculated using the mean decrease in impurity (MDI), as defined by the following equation:

$$MDI_j = \frac{\sum_c^C p(c) \cdot \Delta SSE_{cj}}{T} \tag{5}$$

where T is the number of trees, $p(c)$ is the sample proportion of the c^{th} partition, and ΔSSE_{cj} is the decrease in impurity for the i^{th} variables in the c^{th} partition. The input variables with the highest MDI values from **Equation (5)** contribute the most to constructing the Random Forest. In a large dataset with many input variables, feature importance can be used to select the most relevant variables, improving model efficiency.

2.4 Multivariate Random Forest

Unlike the original Random Forest, Multivariate Random Forest ensembles multiple Multivariate Regression Trees rather than standard Regression Trees **[11]**. Each tree determines the final predictions by averaging all values predicted by the Multivariate Regression Trees. This approach reduces bias in the predicted values, as the training data is randomly selected using the bootstrapping method. Feature importance can also be extracted from the Multivariate Random Forest using the same formulation, with the decrease in impurity calculated across all output variables.

2.5 Neural Network

In recent studies, neural Networks have been one of the most advanced machine learning models. This model mimics the mechanism of neurons in the human body, where millions of neurons are interconnected **[12]**. A Neural Network consists of three layers: input, hidden, and output. The input layer contains neurons representing the input variables, while the output layer comprises the output variables. In the case of a numeric output, there will be exactly one neuron. In contrast, for a categorical output, the number of neurons will correspond to the number of classes in the category. The user predefines the number of hidden layers and neurons in the hidden layers by considering the needs of the data. For clearer visualization, the architecture of a neural network is shown in **Figure 3**.

Although the individual calculations are simple in every neuron, the model becomes more complex once the neurons are connected. The input (x_{ij}) that enters the neuron is multiplied by a weight value (w_{jp}), which determines the influence of the input. For the connections between the input layer and the hidden layer, the input will be the features of the observations, while in other connections, it will be the output of the preceding neuron. The products of all inputs and their corresponding weights are summed together with a bias value (b_p). The bias value is a value that adjusts the curve to fit the data better. Finally, the entire calculation is passed through an activation function ($f(x)$). The activation function introduces nonlinearity to the model. Several activation functions are used, depending on the requirements of the model. The entire process is defined by the calculation below:

$$y_{ip} = f\left(\sum_{j=1}^{J} x_{ij} \cdot w_{jp} + b_p\right). \tag{6}$$

Constructing a Neural Network aims to find the optimal weight and bias values that produce accurate predictions using **Equation (6)**. Initially, the weights and biases for every neuron connection are randomly initialized. Since these initial values rarely fit the data well, they are refined using the backpropagation algorithm.

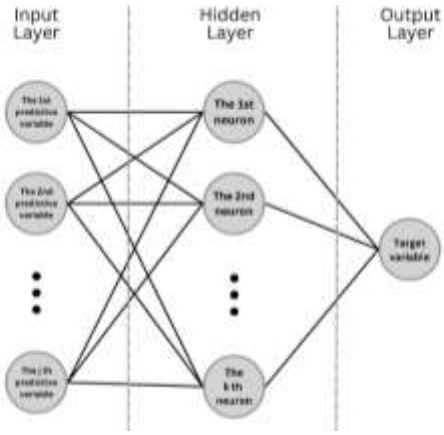


Figure 3. Topology of Neural Network with One Hidden Layer

The backpropagation algorithm employs the chain rule of derivatives and gradient descent to adjust the weights and biases for optimal output variable prediction **[13]**. The adjustments to the weights and biases are calculated using the derivative of the Sum of Squared Errors (SSE) with respect to the weight or bias value. The chain rule is applied backward through the layers to find the necessary derivatives. For example, the change in the weight value for the connection between the neuron in the output layer and the p^{th} neuron in hidden layer is calculated as follows:

$$\Delta w_p = \frac{\partial SSE}{\partial w_p} = \frac{\partial SSE}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial \hat{x}_i} \cdot \frac{\partial \hat{x}_i}{\partial w_p} \tag{7}$$

The new weight value will be calculated using the following formula:

$$w_p{}' = w_p - \Delta w_p \cdot \alpha \tag{8}$$

where α is a constant known as the learning rate, which determines and controls how large a step is taken in each iteration. The learning rate in **Equation (8)** is predefined and is typically kept small to prevent the iterations from diverging. The prediction value is obtained by passing the input variables through the Neural Network model using the refined weights and biases.

2.6 Multi-Output Neural Network

The development of Neural Networks that can predict multiple outputs simultaneously is known as a Multi-Output Neural Network. This model alters the topology of a standard Neural Network by dividing the

hidden layers into two types: shared and private [14], as shown in Figure 4. The shared hidden layer connects to all output variables, while the private hidden layer is dedicated to specific output variables. In simpler terms, the 'signal' sent from the input layer first goes to the neurons in the shared hidden layer. This signal then continues to the neurons in the private hidden layer before being sent to the neurons representing each output variable in the output layer.

The "signal" is transmitted using the same calculation as in **Equation (7)**. The refinement process for the weights and biases is also performed using gradient descent and the chain rule. The difference occurs in the neuron connections from the input to the shared hidden layer and from the shared hidden layer to the private hidden layer, as the calculations consider the sum of squared errors for all output variables. For example, the bias value in the connection between the shared and private hidden layers is calculated as follows:

$$\Delta b_{l_q} = \frac{\partial SSE}{\partial b_{l_q}} = \sum_{l=1}^{L} \frac{\partial SSE}{\partial \widehat{y_{li}}} \cdot \frac{\partial \widehat{y_{li}}}{\partial \widehat{x_{li}}} \cdot \frac{\partial \widehat{x_{li}}}{\partial s_{il_q}} \cdot \frac{\partial s_{il_q}}{\partial r_{il_q}} \cdot \frac{\partial r_{il_q}}{\partial b_i} \tag{9}$$

Equation (9) shows that the difference in bias value is taken from the sum of the sum squared error derivative to the predicted value of all output variables.

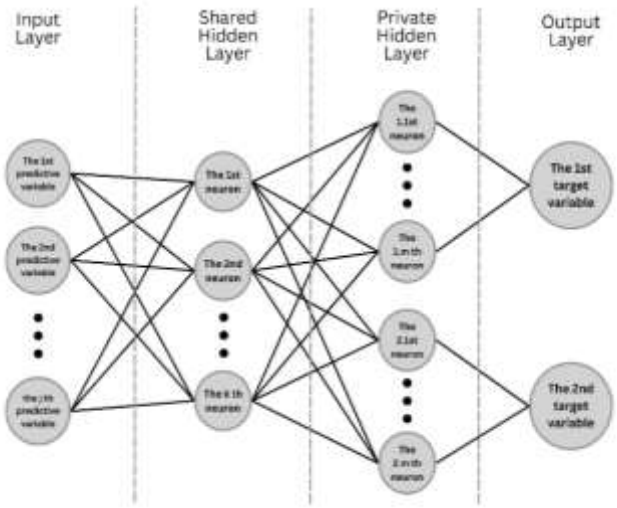


Figure 4. Topology of Multi-Output Neural Network with Shared and Private Hidden Layers

2.7 Logistic Regression

Logistic regression is an extension of the General Linear Model (GLM) designed to predict a dependent variable that is categorical in nature, typically binary [15]. Unlike linear regression, which models a continuous outcome, logistic regression models the probability that a given observation belongs to a particular class. The model estimates the probability of an observation belonging to a specific class by fitting a logistic curve. This curve is an S-shaped function known as the sigmoid function, which outputs values between 0 and 1, representing probabilities. The logistic regression equation is defined as:

$$ln\left(\frac{p(h)}{1-p(h)}\right) = \beta_0 + \beta_1 \cdot x_1 + \ldots + \beta_j \cdot x_j \tag{10}$$

where $p(h)$ is the probability that the observation belongs to the h^{th} class, $\frac{p(h)}{1-p(h)}$ is the odds of the event occurring, $ln\left(\frac{p(h)}{1-p(h)}\right)$ is the logit function. In Logistic Regression from **Equation (10)**, the coefficients $\beta = \{\beta_0, \beta_1, \ldots, \beta_j\}$ are estimated using the Maximum Likelihood Estimation (MLE) method, which finds the values that maximize the likelihood of the observed data.

2.8 Mixed Multi-Output Machine Learning

In some cases, the output variables that need to be predicted are of numerical and categorical types. Most machine learning algorithms can predict both numerical and categorical variables, but there are

differences in their construction algorithms. The loss functions for numerical and categorical variables can't be directly combined because their units differ. The simplest method to predict categorical variables in a machine learning algorithm designed for numerical variables is to convert categorical variables into dummy variables. This conversion usually uses the one-hot encoding method, which transforms categorical variables into a binary matrix, where the values become 0 or 1. However, this method may not produce accurate predictions since the values do not reflect the probability of an observation being classified into a particular class.

Categorical variables can be further converted to logit values using Logistic Regression. First, the data is used to fit a Logistic Regression model. Then, for every observation, whether it's in the training or testing data, the logit value is calculated using the model. For binary types, logit 0 and logit 1 can be predicted along with other numerical output variables. The numerical variables and logit values can then be predicted using the desired machine learning model. After predicting the logit values, they are converted back to probability values, and the class with the highest probability becomes the predicted class of the categorical variable.

2.9 Data Description

Both multi-output and single-output machine learning models are applied to data about digital marketing advertisements sourced from Kaggle. The data consists of 8 variables and 4,571 observations. The variables are described below in **Table 1**. In each advertisement, the company tracks the number of impressions, clicks, and transactions generated by the ads. These three metrics exhibit a funnel-like characteristic, where the number of users performing each action decreases as users move through the stages. The sample data is shown in **Table 2**.

Table 1. Variables Description

Variables Name	Type	Description
Date	Date	Date the ad is displayed
Adgroup	Categorical	Code of adgroup
Ad	Categorical	Code of ad
Impressions	Numerical	Frequency of ad shown
Clicks	Numerical	Frequency clicks by user
Cost	Numerical	Expenses spent to show the ad
Conversions	Numerical	Number of transactions occurred
Revenue	Numerical	Revenue generated from the ad

Table 2. Sample Data

Date	Adgroup	Ad	Impressions	Clicks	Cost	Conversions	Revenue
01-08-2020	adgroup 1	ad 2	247	126	1.29	4	925.71
01-08-2020	adgroup 3	ad 6	253	128	1.32	2	339.83
01-08-2020	adgroup 4	ad 1	5	4	0.08	0	0.0
01-08-2020	adgroup 2	Ad 3	4	0	0.00	0	0.0
01-08-2020	adgroup 1	ad 9	329	130	1.40	3	397.58

***Data source**: data is taken from Kaggle gathered in 2021 of 5 months paid search campaign of a US shopping mall*

From the sample in **Table 2**, it can be concluded that the cost is affected by the number of users who click the ads, meaning the advertiser pays for each click on the ads, commonly known as Cost per Click (CPC). There are several other billing options, such as Cost per View (CPV) and Cost per Mille (CPM). While CPV charges for each individual impression, CPM charges the advertiser for every thousand impressions on the ads. Advertisers can choose the billing option that best generates the desired outcome.

Since the desired output is the number of transactions (conversions) and revenue, both variables are used as output variables to be predicted. These variables are strongly correlated, as a higher number of conversions generally leads to increased revenue from the ads. Therefore, this data is well-suited for multi-output machine learning models.

3. RESULTS AND DISCUSSION

Multi-output machine learning will be applied to the data in **Table 2**, and its results will be compared to single-output machine learning models by examining the sum of squared errors and computational time. The "Conversions" variable in **Table 2** will be transformed into a categorical variable named "Conversions Level." A mixed multi-output machine learning approach will be used to simultaneously predict both the "Conversions Level" and "Revenue." To compare the predictions for "Conversions Level," the accuracy of the models will be reviewed.

3.1 Result of Multi-Output Machine Learning for Numerical-Numerical Outputs

Data from **Table 2** are applied to all the machine learning models discussed above. Prediction accuracy is measured using mean squared error. Additionally, the computational time for constructing multiple single-output models is compared with that of a single multi-output model. The results are presented in **Table 3**.

Table 3. Mean Squared Error and Computational Time Comparison

Variables Name	MSE of Conversions	MSE of Revenue	Computational Time
Single-Output Regression Tree	3.006	166,989.373	0.087
Multivariate Regression Tree	2.650	180,505.158	0.040
Single-Output Random Forest	2.292	104,698.780	1.528
Multivariate Random Forest	2.158	117,868.495	0.812
Single-Output Neural Network	2.269	128,535.058	23.542
Multi-Output Neural Network	1.995	93,125.706	21.609

The Multivariate Regression Tree can significantly reduce the error for "Conversions"; however, this reduction leads to a slight increase in the error for "Revenue." A similar pattern is observed with Multivariate Random Forest and Single-Output Random Forest, where reducing the error for one variable comes at the expense of increased error for another. This trade-off occurs because the splitting function aims to find a Boolean expression that minimizes error for both variables, trying to find the equilibrium point.

In contrast, the Multi-Output Neural Network improves prediction accuracy for both "Conversions" and "Revenue" compared to Single-Output Neural Networks, making it the most effective model among those tested. This advantage arises because the Multi-Output Neural Network uses a shared hidden layer to utilize the relationships between output variables while also employing a private hidden layer to independently refine predictions for each variable. This approach is crucial because, even with high correlation, some degree of independence among output variables remains.

In terms of computational time, all multi-output machine learning models effectively reduce both the processing time for model construction and prediction. While the reduction is not substantial in this case, the time savings would be more significant with larger datasets, considering that machine learning is commonly applied in the big data industry.

3.2 Result of Features Importance

To choose the best billing option between CPC and CPV, the importance of the feature from the Random Forest model will be used to determine which variables contribute the most to the target variables. Feature importance will be extracted from both the Multivariate Random Forest and Single-Output Random

Forest for each target variable. The feature importance results from the Multivariate Random Forest are shown in **Figure 5**.

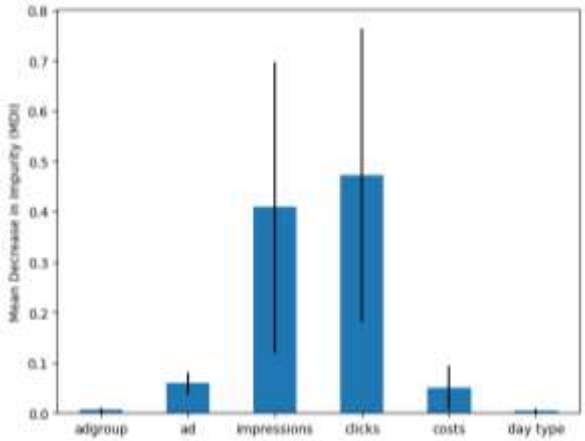
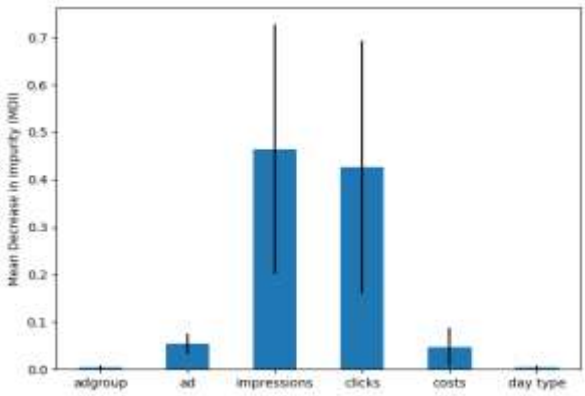
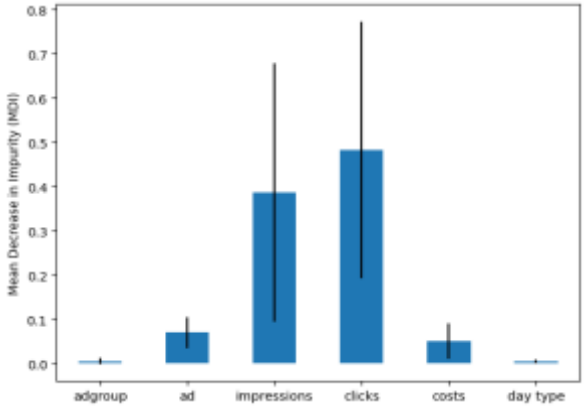


Figure 5. Features Importance from Multivariate Random Forest

The feature's importance obtained from the Multivariate Random Forest shows that the "Clicks" variable contributes the most to the prediction of "Conversions" and "Revenue," followed by "Impressions." This means that to achieve both "Conversions" and "Revenue," the number of clicks by users is the most crucial variable. For that reason, using CPC as the billing option is the correct choice since it can predict both target variables.



(a)



(b)

Figure 6. Features Importance from Single-Output Random Forest
(a) Conversions, (b) Revenue

Feature importance will also be extracted from Single-Output Random Forest models independently for each target variable, as shown in **Figure 6**. The feature importance for "Conversions" indicates that the number of impressions affects the prediction of the "Conversions" variable more than the number of clicks. Therefore, if the company's goal is to achieve more conversions, the best billing option would be CPV or CPM. Meanwhile, the feature importance for "Revenue" shows similar results to the Multivariate Random

Forest. From this, it can be concluded that as a target variable, "Revenue" has more influence on the construction of the Multivariate Random Forest compared to "Conversions."

3.3 Result of Multi-Output Machine Learning for Numerical-Categorical Outputs

The categorical variable predicted using Mixed Multi-Output Machine Learning is "Conversions Level," which is derived from the "Conversions" variable. This variable has two classes: "Low" for observations with values below the average and "High" for those with values above the average. To determine which model provides the best prediction, we will use accuracy as the evaluation metric instead of mean squared error, as the latter is not suitable for categorical variables.

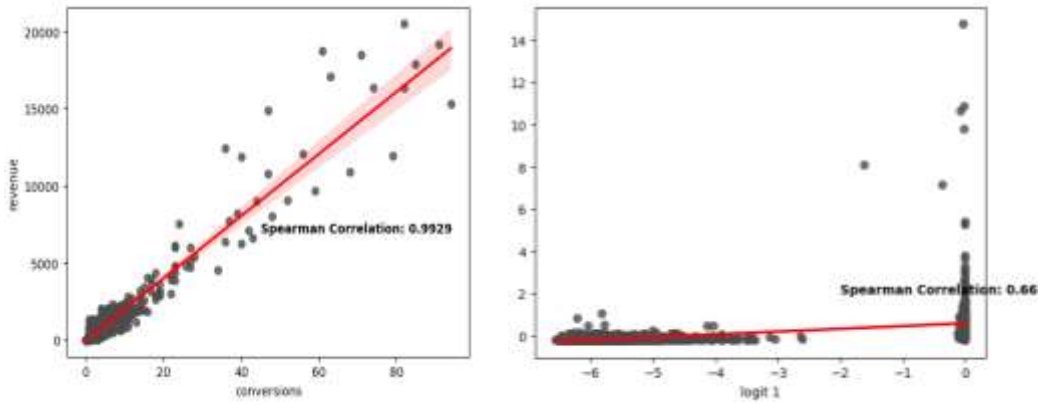
Table 4. Accuracy and Mean Squared Error Comparison

Variables Name	Accuracy of Conversions Level	MSE of Revenue
Single-Output Regression Tree	100%	350,176.587
Multivariate Regression Tree	100%	506,698.012
Single-Output Random Forest	100%	367,763.513
Multivariate Random Forest	100%	430,754.120
Single-Output Neural Network	100%	417,917.709
Multi-Output Neural Network	100%	358,501.499

Table 4 shows that the accuracy of "Conversions Level" across all models achieved 100%. From this, we can conclude that the use of logit values to predict the categorical variable was successfully implemented. However, the mean squared error for the Multivariate Regression Tree and Multivariate Random Forest increased compared to their single-output counterparts. The Multi-Output Neural Network emerged as the best model, outperforming all other models tested in this study for predicting both "Conversation level" and "Revenue."

3.4 The Influence of Correlation in Multi-Output Machine Learning

The correlation between target variables is crucial to ensure that multi-output machine learning can accurately predict both variables, whether the correlation is positive or negative. A correlation above 0.5 or below −0.5 is generally considered strong enough. The closer the correlation value is to 1 or -1, the stronger the relationship between the output variables.



(a) **(b)**

Figure 7. Correlation Plot
(a) Conversions vs Revenue, (b) Logit 1 vs Revenue

Figure 7 (a) shows the correlation between "Conversions" and "Revenue" predicted in the case study of numerical-numerical outputs. The correlation between these two variables reaches 0.99, indicating an almost perfect relationship. In contrast, the correlation between "Logit 1" and "Revenue" in **Figure 7 (b)**, predicted together in the case study, only reaches 0.66, which is considered high but not as strong as the

correlation between "Conversions" and "Revenue." This difference in correlation influences the accuracy of "Revenue," as seen in **Table 3** and **Table 4**, where the mean squared error in multi-output machine learning models is significantly higher for numerical-categorical outputs compared to numerical-numerical outputs.

4. CONCLUSIONS

This study shows that multi-output machine learning models, especially Multi-Output Neural Networks, enhance accuracy and efficiency in a large manner compared to their single-output versions. Several approaches, like Multivariate Regression Trees and Multivariate Random Forests, usually involve a trade-off of prediction accuracy in some target variables with others. The most successful architectures combine shared and private hidden layers, which are best for modeling complex relationships among multiple targets while maintaining independence among them. This progress highlights that choosing suitable models for complex datasets with multi-outputs is important since these neural networks perform better compared to traditional approaches in terms of accuracy and computation time; here, the accuracy is 100%, and MSE is 12.5%, approximately better than the single-output model. The results have broad implications for studies needing high-accuracy, multi-variable forecasting, especially from big and complex datasets. Further refinement of these models in future research and comparison with advanced techniques such as tree-boosting methods and support vector machines will make them all more capable.

REFERENCES

[1] R. Sharma, S. Mithas and A. Kankanhalli, "Transforming decision-making processes: a research agenda for understanding the impact of business analytics on organisations," *European Journal of Information Systems*, vol. 23, no. 4, pp. 433-441, 2014.

[2] G. Li, W. Tian and H. a. C. B. Zhang, "Building energy models at different time scales based on multi-output machine learning," *Buildings*, vol. 12, no. 12, p. 2109, 2022.

[3] H. Zare Abyaneh, "Evaluation of multivariate linear regression and artificial neural networks in prediction of water quality parameters," *Journal of Environmental Health Science and Engineering*, vol. 12, pp. 1-8, 2014.

[4] L. Schmid, A. Gerharz, A. Groll and M. Pauly, "Tree-based ensembles for multi-output regression: Comparing multivariate approaches with separate univariate ones," *Computational Statistics & Data Analysis*, vol. 179, p. 107628, 2023.

[5] L. Cui, X. Xie, Z. Shen, R. Lu and H. Wang, "Prediction of the healthcare resource utilization using multi-output regression models," *IISE Transactions on Healthcare Systems Engineering*, vol. 8, no. 4, pp. 291-302, 2018.

[6] L. He, S. C. Madathil, G. Servis and M. T. Khasawneh, "Neural network-based multi-task learning for inpatient flow classification and length of stay prediction," *Applied Soft Computing*, vol. 108, p. 107483, 2021.

[7] V. G. Costa and C. E. Pedreira, "Recent advances in decision trees: An updated survey," *Artificial Intelligence Review*, vol. 56, no. 5, pp. 4765-4800, 2023.

[8] G. O. Ndong, J. Villerd, I. Cousin and O. Therond, "Using a multivariate regression tree to analyze trade-offs between ecosystem services: Application to the main cropping area in France," *Science of The Total Environment*, vol. 764, p. 142815, 2021.

[9] S. J. Rigatti, "Random forest," *Journal of Insurance Medicine*, vol. 47, no. 1, pp. 31-39, 2017.

[10] S. Gharsalli, B. Emile, H. Laurent, X. Desquesnes and D. Vivet, "Random forest-based feature selection for emotion recognition," in *2015 International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Orléans, 2015.

[11] M. López, O. Antonio, A. Montesinos López and J. Crossa, Multivariate statistical machine learning methods for genomic prediction, Springer Nature, 2022.

[12] M. Heidarpanah, F. Hooshyaripor and M. Fazeli, "Daily electricity price forecasting using artificial intelligence models in the Iranian electricity market," *Energy*, vol. 263, p. 126011, 2023.

[13] C. C. Aggarwal, "The Backpropagation Algorithm," in *Neural networks and deep learning*, Springer, 2018, pp. 29-71.

[14] J. Wu and Y. Sun, "Evolving deep parallel neural networks for multi-task learning," in *International Conference on Algorithms and Architectures for Parallel Processing*, Xiamen, 2021.

[15] G. James, D. Witten, T. Hastie and R. Tibshirani, An Introduction to Statistical Learning: with Application in R, 2nd ed., New York: Springer Nature, 2021.