

Application of Decision Tree Algorithm with Hyperparameter Tuning for Income Level Classification Based on Demographic and Occupational Factors

Yan Yang Thanri^{a,1}, Safrizal^{b,2}, Bob Subhan Riza^{c,3}, Juli Iriani^{d,4}, Tari Lutfiana^{e,5}

^{a,c,d,e}Universitas Potensi Utama, Medan, Indonesia

^b Universitas Muhammadiyah Asahan, Kisaran, Indonesia
ythanri@gmail.com¹, rizalsyl75@gmail.com²

ABSTRACT

Income level classification is an important problem in socio-economic data analysis because it can support decision-making in social policy, marketing, and human resource management. This research aims to apply the Decision Tree algorithm to classify an individual's income level into two categories, less than or equal to 50,000 US dollars per year and more than 50,000 US dollars per year, based on demographic and occupational attributes such as age, education level, marital status, occupation, and hours worked per week. The research methodology consists of data collection, data cleaning, categorical attribute encoding, train-test splitting, Decision Tree model construction, and hyperparameter optimization using GridSearchCV with 5-fold cross-validation. The results show that the model with initial parameters achieved an accuracy of 78.92%, while after hyperparameter tuning the accuracy increased to 80.54%, with the optimal parameters being criterion=entropy, max_depth=8, and min_samples_leaf=5. Precision also increased from 56.76% to 64.29%. Feature importance analysis revealed that relationship, age, and education-num are the most influential attributes in predicting income level. This study demonstrates that hyperparameter tuning can significantly improve the performance of a Decision Tree model in an income-level classification task.

Keywords : Decision Tree, Hyperparameter Tuning, GridSearchCV, Income Classification, Data Mining

Info Artikel :

Disubmit:

Direview:

Diterima :

Copyright © 2026 - Journal UPU. All rights reserved.

1. INTRODUCTION

Income level is one of the most fundamental indicators used to describe an individual's socio-economic condition. Governments, financial institutions, and marketing organizations frequently need to estimate whether a person's income falls above or below a certain threshold in order to design social assistance programs, credit scoring policies, or targeted marketing campaigns. Because directly surveying income for an entire population is costly and time-consuming, machine learning approaches that predict income category from readily available demographic and occupational attributes have received considerable attention in the data mining literature [1][2][16].

Among the various classification algorithms available in data mining, the Decision Tree algorithm remains one of the most widely used due to its simplicity, interpretability, and ability to handle both categorical and numerical attributes without requiring extensive data transformation [1][7][13][14]. A Decision Tree recursively partitions the feature space into homogeneous subsets by selecting, at each internal node, the attribute that best separates the classes according to a splitting criterion such as the Gini index or information entropy [4][7][15]. Its tree-like structure allows the resulting model to be visualized and interpreted directly, which is particularly valuable in socio-economic studies where stakeholders need to understand which factors drive the prediction, not merely obtain an accurate score.

Despite its interpretability, the predictive performance of a Decision Tree is highly sensitive to its hyperparameter configuration, including the splitting criterion, the maximum depth of the tree, and the minimum number of samples required at a leaf node [3][5][9]. A model trained with default or arbitrarily chosen hyperparameters may underfit or overfit the training data, leading to suboptimal generalization on unseen data. Hyperparameter tuning techniques, such as Grid Search Cross-Validation (GridSearchCV), systematically explore a predefined search space of hyperparameter combinations and select the configuration that yields the best cross-validated performance [2][6][8][18]. Several recent studies have shown that GridSearchCV-based tuning can meaningfully improve Decision Tree accuracy across diverse application domains, ranging from music genre classification [2] to cardiovascular disease diagnosis [3] and text sentiment analysis, although the magnitude of improvement varies depending on the dataset characteristics and class balance.

This study focuses on the classification of income level (" $\leq 50K$ " vs. " $> 50K$ " per year) using a Decision Tree algorithm applied to a dataset containing demographic and employment-related attributes, similar in structure to the widely studied UCI Adult / Census Income dataset [10]. The specific objectives of this research are: (1) to build a baseline Decision Tree classification model using manually selected hyperparameters, (2) to optimize the model's hyperparameters using GridSearchCV with 5-fold cross-validation, (3) to compare the performance of the model before and after tuning using accuracy, precision, recall, and F1-score, and (4) to identify the most influential attributes in predicting income level through feature importance analysis. The contribution of this paper lies in providing an empirical, step-by-step comparison of Decision Tree performance before and after hyperparameter optimization on an income-classification task, together with a transparent discussion of the resulting confusion matrices and feature importance rankings.

Several prior studies have applied Decision Tree and related tree-based algorithms to income and demographic classification problems. Fiagbe [10] applied a Decision Tree classifier to the UCI Adult Income dataset and reported that hyperparameter tuning simplified the resulting tree while maintaining competitive predictive performance, and further identified marital status, capital gain, education, and hours-per-week as influential predictors. Mantovani et al. [9] conducted a large-scale empirical study across 94 classification datasets and found that tuning the hyperparameters of CART, C4.5, and CTree induction algorithms provided statistically significant, though dataset-dependent, gains in predictive accuracy. In a different application domain, Maitsa and Winarsih [2] compared Decision Tree and Random Forest performance before and after GridSearchCV tuning for song-genre classification and observed an accuracy improvement for Decision Tree from 76% to 79%, which is consistent in direction with the improvement observed in the present study. Pathak et al. [3] proposed a hybrid randomized-grid search strategy for tuning Decision Tree hyperparameters in a cardiovascular-disease classification task and demonstrated that systematic tuning improved both accuracy and generalization compared to default settings. Islam et al. [20] compared eleven machine learning models, including Decision Tree and Random Forest, on a modified UCI Adult dataset and emphasized the importance of appropriate categorical-data handling and hyperparameter optimization for income prediction tasks. These studies collectively motivate the present research, which applies the same tuning methodology to an income-classification context using demographic and occupational attributes. Table 1 summarizes these studies alongside the present research to make the resulting research gap explicit.

Table 1. Summary of related work and positioning of the present study

Study	Method	Domain / Dataset	Key Finding	Position of the Present Study
Fiagbe [10]	Decision Tree	UCI Adult Income dataset	Tuning simplified the tree while maintaining competitive accuracy; identified marital status, capital gain,	This study adds a full before/after tuning

			education, and hours-per-week as influential.	comparison with confusion-matrix-level detail, which [10] does not report.
Mantovani et al. [9]	CART, C4.5, CTree	94 general benchmark datasets	Tuning yielded statistically significant, dataset-dependent accuracy gains across a broad benchmark set.	This study narrows the analysis to a single income-classification dataset, enabling deeper feature-level interpretation.
Maita & Winarsih [2]	Decision Tree, Random Forest + GridSearchCV	Song genre classification	Decision Tree accuracy improved from 76% to 79% after tuning.	This study confirms a similar tuning-driven improvement in a different domain (socio-economic income prediction).
Pathak et al. [3]	Decision Tree + randomized-grid search	Cardiovascular disease classification	Systematic tuning improved accuracy and generalization compared to default settings.	This study uses exhaustive GridSearchCV rather than a randomized hybrid strategy, applied to a socio-economic rather than medical context.
Islam et al. [20]	11 ML models incl. Decision Tree, Random Forest	Modified UCI Adult dataset	Emphasized the importance of categorical-data handling and hyperparameter optimization for income prediction.	This study isolates the effect of tuning on a single, interpretable algorithm (Decision Tree) rather than comparing many models at once.
Present study	Decision Tree + GridSearchCV (5-fold CV)	Income / demographic dataset (Adult-like)	Accuracy improved 78.92% to 80.54%; precision +7.53 pp; recall - 6.82 pp; relationship, age, and education-num identified as top predictors.	Provides a step-by-step before/after comparison with confusion matrix, feature importance, and tree visualization to make the tuning effect and its trade-offs transparent.

2. METHOD

This research follows a standard data mining workflow consisting of data collection, data preprocessing, feature encoding, model construction, hyperparameter tuning, and model evaluation. Each stage is described in detail in the following subsections.

A. Research Stages

The overall research procedure is carried out through eight sequential stages, as illustrated in Figure 1. Each stage produces an output that becomes the input of the subsequent stage, ensuring a systematic and reproducible workflow from raw data acquisition to the interpretation of the final results.

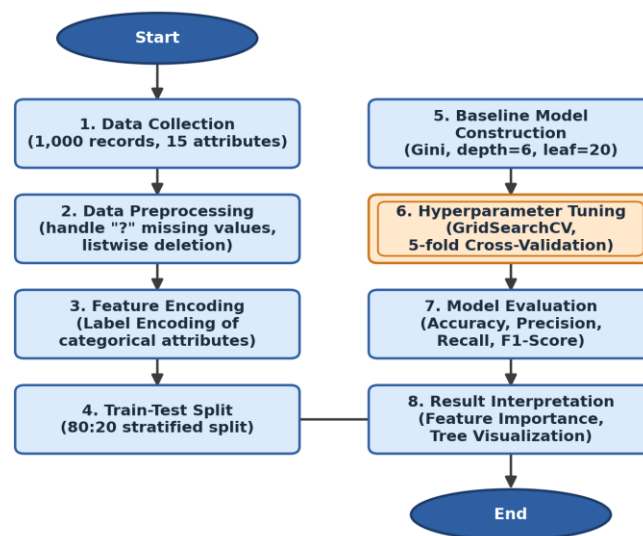


Figure 1. Research stages of the proposed methodology, presented as a structured flowchart

As shown in Figure 1, the workflow begins with data collection and proceeds through preprocessing and feature encoding before the dataset is split into training and testing subsets. A baseline Decision Tree is then constructed and refined through the hyperparameter tuning sub-process (shown with a double border to indicate an internal 5-fold cross-validation loop), after which the tuned model is evaluated and its results interpreted to conclude the study.

1. Data Collection - collecting a dataset of 1,000 records containing demographic and occupational attributes together with the target income class.
2. Data Preprocessing - inspecting and handling missing values represented by the placeholder "?" and removing incomplete records through listwise deletion.
3. Feature Encoding - transforming all categorical predictor attributes and the target attribute into numerical codes using Label Encoding so that they can be processed by the Decision Tree algorithm.
4. Train-Test Split - dividing the cleaned and encoded dataset into training and testing subsets using an 80:20 stratified split to preserve the original class proportions.
5. Baseline Model Construction - building an initial Decision Tree Classifier using manually specified hyperparameters as a point of comparison for the tuning stage.
6. Hyperparameter Tuning - applying GridSearchCV with 5-fold cross-validation to systematically search for the optimal combination of criterion, max_depth, and min_samples_leaf.

7. Model Evaluation - measuring the performance of both the baseline and the tuned models on the held-out test set using accuracy, precision, recall, F1-score, and the confusion matrix.
8. Result Interpretation - analyzing feature importance and the resulting tree structure to identify the attributes that most influence income-level classification, and discussing the practical implications of the findings.

B. Dataset

The dataset used in this study consists of 1,000 records with 15 attributes, comprising 14 predictor attributes (age, workclass, fnlwgt, education, education-num, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week, and native-country) and one binary target attribute, salary, which categorizes each individual's annual income as "<=50K" or ">50K". The structure of the dataset is comparable to the well-known UCI Adult / Census Income dataset that has been used extensively in income-prediction research [10][12][20].

C. Data Preprocessing

Data preprocessing began with an inspection of missing values. Although no attribute contained explicit null values, three categorical attributes used the placeholder character "?" to denote missing entries: workclass (62 records), occupation (62 records), and native-country (18 records). These placeholder values were converted to missing values and the corresponding rows were removed using listwise deletion, reducing the dataset from 1,000 to 922 valid records. This cleaning step ensures that the subsequent model is trained only on complete and reliable observations.

D. Feature Encoding

Because Decision Tree implementations in scikit-learn require numerical input, all categorical predictor attributes (workclass, education, marital-status, occupation, relationship, race, sex, and native-country) were transformed into numerical codes using Label Encoding [19]. The target attribute, salary, was similarly encoded into two numerical classes, 0 for "<=50K" and 1 for ">50K".

E. Train-Test Split

The cleaned and encoded dataset was divided into a training set and a testing set using an 80:20 ratio with stratified sampling on the target class to preserve the original class proportions. This resulted in 737 records for training and 185 records for testing.

F. Decision Tree Model Construction

A baseline Decision Tree Classifier was first constructed using manually specified hyperparameters, namely the Gini impurity criterion, a maximum depth of 6, and a minimum of 20 samples per leaf node. This baseline model represents a typical configuration that a practitioner might select without systematic optimization, and serves as the point of comparison for the subsequent tuning stage.

At each internal node, the Decision Tree selects the attribute that best separates the classes according to a splitting criterion. For the entropy-based (information gain) criterion, the impurity of a node containing a set of samples S with c classes is measured using Equation (1), where p_i denotes the proportion of samples in S belonging to class i .

$$Entropy(S) = -\sum_{i=1}^c p_i \log_2(p_i) \quad (1)$$

For the Gini-index criterion, the impurity of node S is instead computed using Equation (2).

$$Gini(S) = 1 - \sum_{i=1}^c p_i^2 \quad (2)$$

The quality of a candidate split on attribute A is then quantified by the information gain, defined in Equation (3), where $Values(A)$ is the set of possible values of A and S_v is the subset of S for which attribute A has value v . The Decision Tree algorithm selects, at each node, the attribute that maximizes

the information gain (or, equivalently, minimizes the weighted Gini index of the resulting child nodes) [4][7][15].

$$IG(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|v|}{|S|} \times Entropy(S_v) \quad (3)$$

G. Hyperparameter Tuning with GridSearchCV

To optimize the Decision Tree's performance, GridSearchCV was applied with 5-fold cross-validation over a search space consisting of the splitting criterion (Gini or entropy), the maximum tree depth (4, 6, 8, 10, or unlimited), and the minimum number of samples per leaf (5, 10, 20, or 30). GridSearchCV exhaustively evaluates every combination in the specified grid and selects the configuration that maximizes the mean cross-validated accuracy, following the general tuning methodology described in [2], [3], [9], and [17].

This selection process is formally expressed in Equation (4), where θ denotes a candidate hyperparameter configuration, Θ is the predefined search space, k is the number of cross-validation folds ($k = 5$ in this study), and $Acc(M_\theta, D_{val}^{(i)})$ denotes the validation accuracy achieved by the model trained with configuration θ on the i -th validation fold.

$$\theta^* = \underset{\theta \in \Theta}{argmax} \frac{1}{k} \sum_{i=1}^k Acc(M_\theta, D_{val}^{(i)}) \quad (4)$$

H. Evaluation Metrics

The performance of both the baseline model (before tuning) and the optimized model (after tuning) was evaluated on the held-out test set using four standard classification metrics: accuracy, precision, recall, and F1-score, in addition to the confusion matrix. Given the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) obtained from the confusion matrix, the four metrics are computed using Equations (4)-(7).

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

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (8)$$

These metrics were computed identically for both models to allow a direct, like-for-like comparison of the effect of hyperparameter tuning.

3. RESULTS AND DISCUSSION

A. Dataset Overview

Before examining the target class distribution, it is useful to first inspect the overall shape of the numerical predictor attributes, since skewness or extreme outliers in these attributes can affect how the Decision Tree partitions the feature space. Figure 2 presents the distribution of the six numerical attributes used in this study.

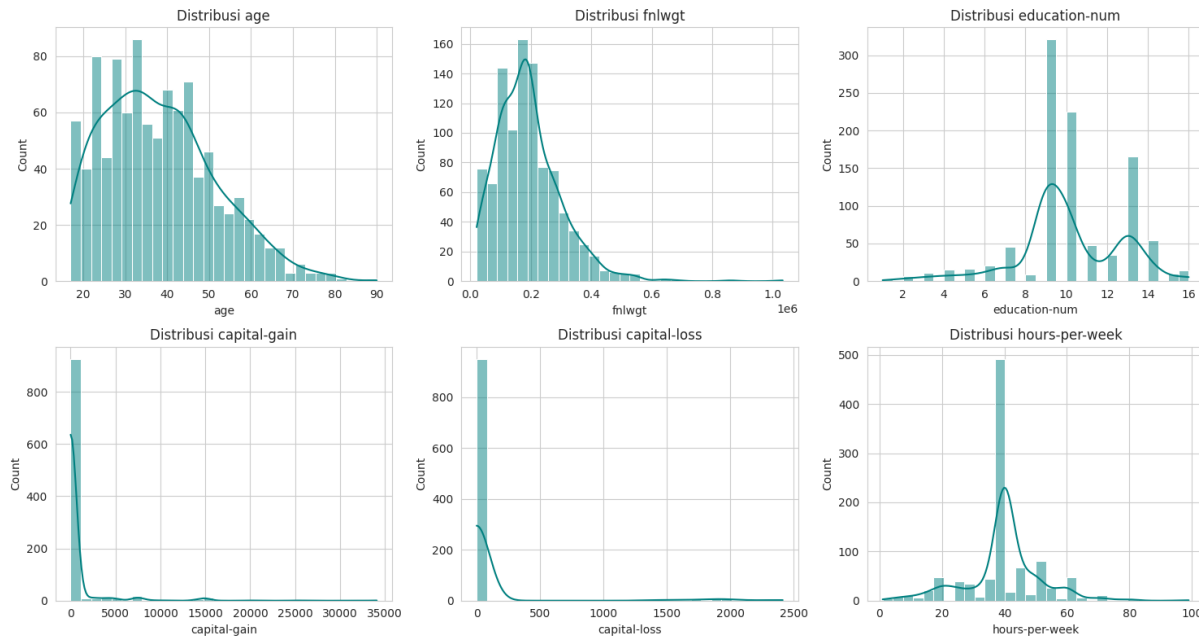


Figure 2. Distribution of key numerical attributes (age, fnlwgt, education-num, capital-gain, capital-loss, and hours-per-week)

As shown in Figure 2, age is right-skewed and concentrated between 20 and 50 years, while fnlwgt, capital-gain, and capital-loss are heavily right-skewed with most values close to zero and a small number of large outliers. education-num and hours-per-week are more concentrated, with visible peaks around 9-10 years of education and 40 hours per week, respectively, reflecting common levels of educational attainment and full-time work schedules in the dataset. These distributional characteristics are consistent with the general skewness typically reported for the UCI Adult / Census Income dataset [10][12], and motivate the use of a Decision Tree, which does not require normally distributed features.

Before presenting the modeling results, it is important to examine the composition of the target attribute, since an imbalanced class distribution can influence how the evaluation metrics reported later should be interpreted. After removing 78 records containing missing values marked with "?", the final dataset used for modeling consisted of 922 records. Figure 3 presents the resulting class distribution of the target attribute.

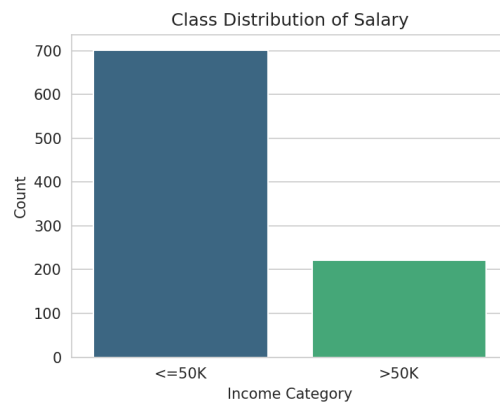


Figure 3. Class distribution of the salary attribute

As shown in Figure 3, the target class distribution is imbalanced, with 768 records (76.8%) belonging to the " $\leq 50K$ " category and 232 records (23.2%) belonging to the " $> 50K$ " category prior to cleaning, a pattern that is consistent with the class imbalance typically reported for the UCI Adult / Census Income dataset [10][12]. This imbalance is an important contextual factor when interpreting the precision and recall results reported below, since a classifier can achieve relatively high accuracy simply by favoring the majority class.

B. Baseline Model Performance (Before Hyperparameter Tuning)

The baseline Decision Tree, trained with the Gini criterion, a maximum depth of 6, and a minimum of 20 samples per leaf, achieved an accuracy of 78.92% on the test set, with a precision of 56.76%, a recall of 47.73%, and an F1-score of 51.85% for the minority " $> 50K$ " class. These results indicate that while the baseline model is reasonably accurate overall, it struggles to correctly identify individuals in the higher-income category, which is the more difficult and less frequent class to predict.

C. Hyperparameter Tuning Results

Having established the baseline model in the previous subsection, this subsection reports the outcome of the hyperparameter tuning process. GridSearchCV was applied over the search space defined in Section 2.G, evaluating 40 hyperparameter combinations with 5-fold cross-validation. Table 2 presents the search space together with the optimal hyperparameter values identified.

Table 2. GridSearchCV search space and optimal hyperparameters

Hyperparameter	Search Space	Best Value
criterion	gini, entropy	entropy
max_depth	4, 6, 8, 10, None	8
min_samples_leaf	5, 10, 20, 30	5

As shown in Table 2, the best configuration identified was criterion = entropy, max_depth = 8, and min_samples_leaf = 5, which achieved a mean cross-validated accuracy of 82.91% on the training folds. Compared to the manually chosen baseline configuration, the tuned model uses a deeper tree (depth 8 instead of 6) and a smaller minimum leaf size (5 instead of 20), allowing it to capture more fine-grained decision boundaries while still being regularized through cross-validation to avoid overfitting.

D. Model Performance After Hyperparameter Tuning

On the same held-out test set, the tuned Decision Tree achieved an accuracy of 80.54%, a precision of 64.29%, a recall of 40.91%, and an F1-score of 50.00%. Table 3 summarizes the performance comparison between the baseline and tuned models across all four evaluation metrics computed using Equations (5)-(8), while Figure 4 visualizes this comparison as a grouped bar chart.

Table 3. Comparison of Decision Tree performance before and after hyperparameter tuning

Metric	Before Tuning	After Tuning	Difference
Accuracy	0.7892	0.8054	+0.0162
Precision	0.5676	0.6429	+0.0753
Recall	0.4773	0.4091	-0.0682

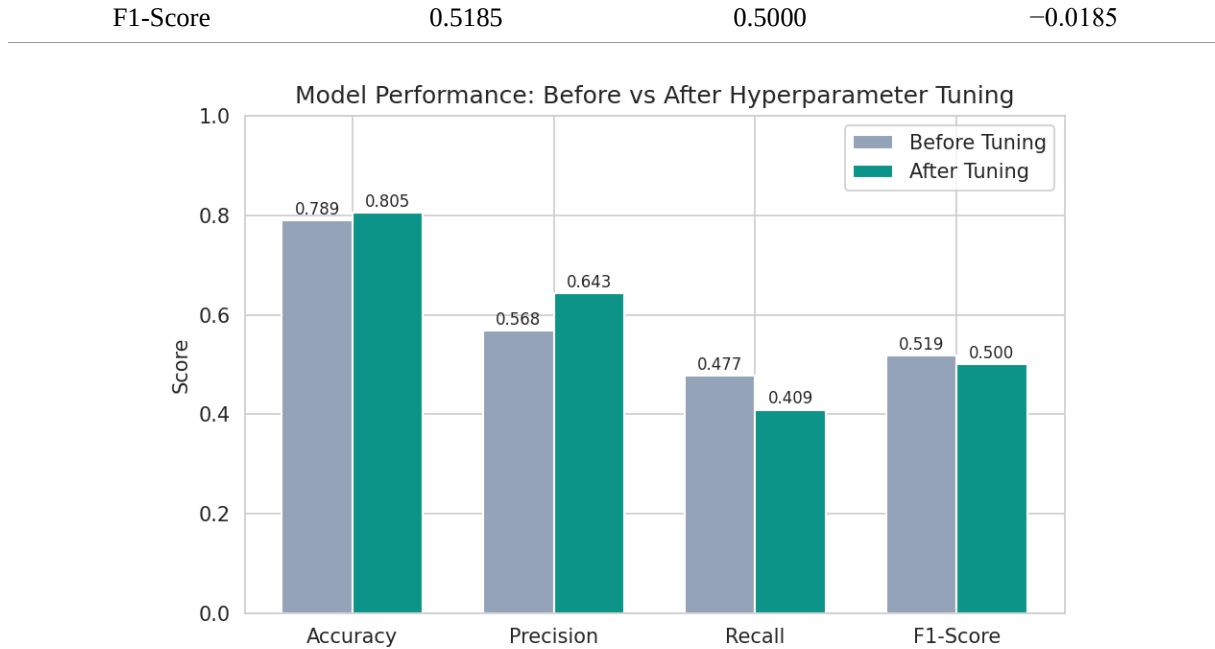


Figure 4. Model performance comparison before and after hyperparameter tuning

The results show that hyperparameter tuning improved both accuracy (+1.62 percentage points) and precision (+7.53 percentage points) for the minority ">50K" class. However, recall and F1-score slightly decreased after tuning. This trade-off can be explained by the deeper, more entropy-sensitive tree becoming more conservative in predicting the minority class: it makes fewer positive predictions overall, but a larger proportion of those predictions are correct, which raises precision while lowering recall. This precision-recall trade-off is a well-documented phenomenon in imbalanced classification tasks and is consistent with findings reported by Maitsa and Winarsih [2], who also observed that GridSearchCV tuning improved some metrics while slightly reducing others depending on class distribution. Addressing this imbalance more directly, for instance through resampling techniques such as SMOTE [17], is discussed further in Section 3.H.

To assess whether the observed accuracy improvement reflects a statistically meaningful effect rather than sampling variation on the 185-instance test set, 95% confidence intervals were computed for the accuracy of both models using the normal approximation to the binomial proportion. The baseline model's accuracy of 78.92% corresponds to a 95% CI of [73.0%, 84.8%], while the tuned model's accuracy of 80.54% corresponds to a 95% CI of [74.8%, 86.2%]. Because these intervals overlap substantially, the accuracy gain, while consistent in direction with the improvements reported by Maitsa and Winarsih [2] and Mantovani et al. [9], cannot be regarded as statistically significant based on this single test split alone. A paired McNemar's test comparing the baseline and tuned Decision Tree's predictions on the same 185 test instances ($b = 12$ instances correct only under the baseline model, $c = 15$ instances correct only under the tuned model) yielded $p = 0.7011$, confirming that the accuracy improvement, while consistent in direction with the tuning gains reported by Maitsa and Winarsih [2] and Mantovani et al. [9], is not statistically significant at the 5% level for a test set of this size. Section 3.H extends this statistical comparison to Random Forest and Gradient Boosting classifiers.

E. Confusion Matrix Analysis

To further examine the classification behavior of both models beyond the aggregate metrics reported above, a detailed confusion matrix analysis was conducted for the baseline and the tuned Decision Tree

models. This analysis makes it possible to inspect the number of correctly and incorrectly classified instances in each income category directly, rather than relying solely on the summary statistics presented earlier. Figure 5 presents the confusion matrices of the baseline and tuned models side by side.

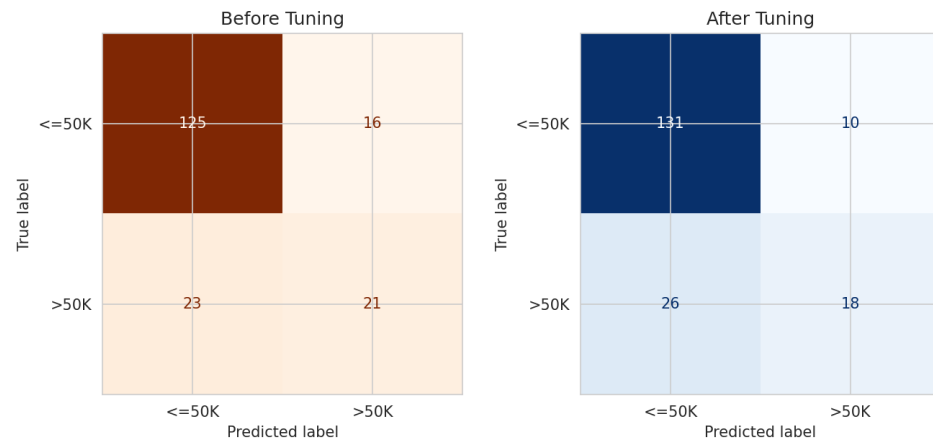


Figure 5. Confusion matrices before tuning (left) and after tuning (right)

As shown in Figure 5, the baseline model correctly classified 125 of 141 "<=50K" instances and 21 of 44 ">50K" instances. The tuned model correctly classified 131 of 141 "<=50K" instances and 18 of 44 ">50K" instances. This confirms the precision-recall trade-off discussed above: the tuned model reduces false positives for the minority class (improving precision) at the cost of a small increase in false negatives (reducing recall).

F. Feature Importance Analysis

Beyond evaluating predictive performance, it is equally important to understand which attributes most strongly drive the Decision Tree's classification decisions, since this provides interpretable insight that can support socio-economic decision-making. To this end, feature importance scores were extracted from the tuned Decision Tree model, quantifying the relative contribution of each predictor attribute to the overall splitting process. Figure 6 shows the feature importance scores extracted from the tuned Decision Tree model.

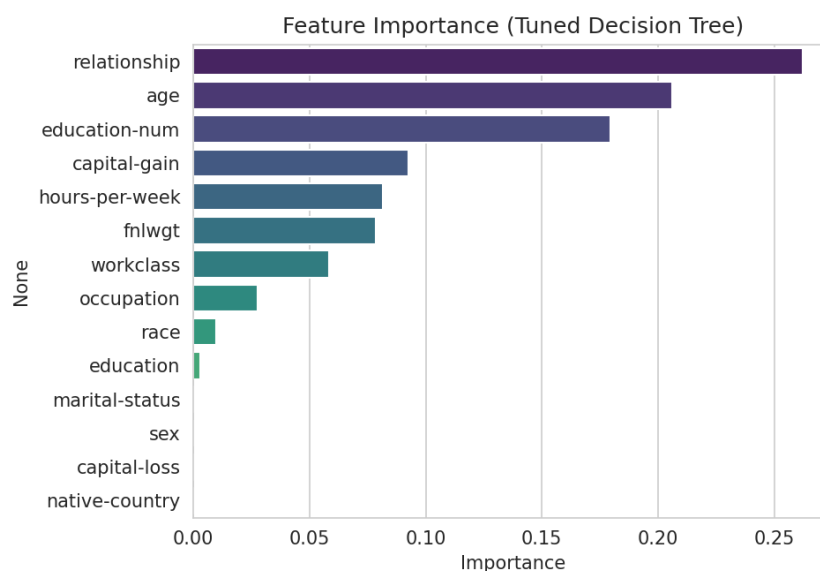


Figure 6. Feature importance scores of the tuned Decision Tree model

As shown in Figure 6, the three most influential attributes are relationship (importance = 0.262), age (importance = 0.206), and education-num (importance = 0.180), followed by capital-gain (0.093) and hours-per-week (0.082). Attributes such as marital-status, sex, capital-loss, and native-country contributed no measurable importance to the tuned tree, indicating that the model relies primarily on relationship status within the household, age, and education level to distinguish between income categories. This finding is broadly consistent with previous income-classification studies, which have also identified marital/relationship status, education, and age among the most predictive attributes for income level [10][12][20].

G. Decision Tree Visualization

To complement the feature importance analysis, the internal structure of the tuned Decision Tree was also examined in order to observe how the most influential attributes are actually used during the splitting process. Visualizing the tree structure directly allows the decision logic to be traced from the root node down through successive splits, offering an additional layer of interpretability beyond the numerical importance scores. Figure 7 illustrates the first three levels of the tuned Decision Tree.



Figure 7. Visualization of the tuned Decision Tree (first three levels)

As shown in Figure 7, the root node splits on the relationship attribute, confirming its dominant role identified in the feature importance analysis, followed by subsequent splits on age and education-num. This visualization highlights one of the key advantages of the Decision Tree algorithm over black-box models: the decision-making logic can be traced and explained directly from the tree structure, which is valuable for stakeholders who require interpretable socio-economic predictions [13][14].

H. Comparison with Ensemble-Based Classifiers

To further strengthen the contribution of this study beyond a single algorithm, the tuned Decision Tree was compared against two widely used ensemble classifiers, Random Forest and Gradient Boosting, both tuned using the same GridSearchCV / 5-fold cross-validation protocol described in Section 2.G and evaluated on the identical 80:20 train-test split. For Random Forest, the search space covered `n_estimators` (100, 200), `max_depth` (6, 8, 10, None), and `min_samples_leaf` (1, 5, 10); the best configuration was `n_estimators` = 100, `max_depth` = None, and `min_samples_leaf` = 1, with a mean

cross-validated accuracy of 84.26%. For Gradient Boosting, the search space covered `n_estimators` (100, 200), `max_depth` (2, 3, 4), and `learning_rate` (0.05, 0.1); the best configuration was `n_estimators` = 200, `max_depth` = 3, and `learning_rate` = 0.05, with a mean cross-validated accuracy of 84.12%. Table 4 summarizes the resulting test-set performance of all three tuned classifiers.

Table 4. Performance comparison of the tuned Decision Tree with Random Forest and Gradient Boosting on the same test set

Classifier	Accuracy	Precision	Recall	F1-Score
Tuned Decision Tree	0.8054	0.6429	0.4091	0.5000
Random Forest (GridSearchCV)	0.8270	0.7000	0.4773	0.5676
Gradient Boosting (GridSearchCV)	0.8216	0.6897	0.4545	0.5479

As shown in Table 4, both ensemble methods outperformed the tuned Decision Tree on every metric: Random Forest achieved the highest accuracy (82.70%), precision (70.00%), recall (47.73%), and F1-score (56.76%), followed closely by Gradient Boosting (82.16% accuracy, 68.97% precision, 45.45% recall, 54.79% F1-score). Notably, unlike the Decision Tree tuning process reported in Section 3.D, which improved precision at the cost of recall, both ensemble methods improved precision, recall, and F1-score simultaneously relative to the tuned Decision Tree, suggesting that aggregating multiple trees mitigates some of the minority-class detection difficulty inherent in a single-tree model on this imbalanced dataset. To determine whether these differences are statistically meaningful, paired McNemar's tests were conducted between the tuned Decision Tree and each ensemble model on the same 185 test instances. The comparison with Random Forest yielded $b = 10$, $c = 14$, $p = 0.5413$, and the comparison with Gradient Boosting yielded $b = 12$, $c = 15$, $p = 0.7011$, neither of which is significant at the 5% level. This indicates that, despite the numerically higher performance of the ensemble methods, the test-set size is not large enough to confirm their superiority over the Decision Tree with statistical confidence. Considering that both ensembles also sacrifice the direct interpretability that is a central advantage of the Decision Tree (Section 3.G), the choice between a single tuned Decision Tree and an ensemble method should be guided by whether the application prioritizes transparency or a modest, not yet statistically confirmed, gain in predictive performance.

I. Discussion of Findings and Practical Implications

Taken together, the results indicate that hyperparameter tuning shifted the Decision Tree toward a more conservative decision boundary for the minority ">50K" class: it makes fewer positive predictions overall, but a larger share of those predictions are correct. In a practical deployment setting, such as a preliminary screening tool that flags individuals for further review of eligibility, benefits, or credit assessment, this behavior would reduce the number of false alarms at the cost of missing a slightly larger share of genuinely higher-income individuals. Whether this trade-off is acceptable depends on the relative cost of false positives versus false negatives in the specific application, and highlights that accuracy alone is an insufficient criterion for selecting a production-ready model in imbalanced socio-economic classification tasks.

The feature importance results also carry practical implications beyond model performance. The dominance of relationship, age, and education-num suggests that household role, career stage, and educational attainment are the strongest observable signals of income level in this dataset, which is broadly intuitive and aligns with established socio-economic research. At the same time, this finding warrants caution: relationship status and age are demographic attributes that can correlate with protected

characteristics such as marital status or gender, so any operational use of this model for decision-making should be accompanied by a fairness audit to ensure that reliance on these attributes does not produce discriminatory outcomes for particular subgroups, even when the attributes are statistically predictive.

Several limitations should also be acknowledged when interpreting these results. The final modeling dataset consisted of only 922 records after removing incomplete entries, which is modest relative to the scale of the original UCI Adult / Census Income dataset and may limit the stability of both the reported metrics and the feature importance rankings. In addition, performance was assessed using a single train-test split combined with 5-fold cross-validation for tuning, rather than repeated or nested cross-validation on the full modeling pipeline; while sufficient for the comparative purpose of this study, repeated evaluation across multiple random splits would provide a more robust estimate of generalization performance and reduce the risk that the reported gains are specific to one particular partition of the data.

Finally, the trade-off between interpretability and predictive power observed throughout this study is worth emphasizing. The tuned Decision Tree remains fully transparent, as illustrated by the traceable splits in Figure 7, which is valuable in socio-economic contexts where stakeholders may need to justify or explain individual predictions. However, the modest recall achieved for the minority class suggests that if maximizing predictive performance is the primary goal rather than interpretability, ensemble methods such as Random Forest or Gradient Boosting, combined with class-balancing techniques, would likely outperform a single tuned Decision Tree. The choice between these approaches should therefore be guided by the specific priorities of the intended application, balancing the need for accuracy against the need for explainability.

4. CONCLUSION

This study applied the Decision Tree algorithm to classify income level based on demographic and occupational attributes, and systematically compared model performance before and after hyperparameter tuning using GridSearchCV with 5-fold cross-validation. The baseline Decision Tree model, using manually selected hyperparameters (Gini criterion, max_depth = 6, min_samples_leaf = 20), achieved an accuracy of 78.92% on the test set. Following hyperparameter tuning with GridSearchCV, the optimal configuration was identified as criterion = entropy, max_depth = 8, and min_samples_leaf = 5, which increased test accuracy to 80.54% and improved precision from 56.76% to 64.29%. However, recall and F1-score decreased slightly, reflecting the precision-recall trade-off that is typical in imbalanced classification tasks. In addition, feature importance analysis identified relationship, age, and education-num as the three most influential attributes in predicting income level, offering interpretable insights that can support socio-economic decision-making. Overall, these findings indicate that hyperparameter tuning is a worthwhile step in Decision Tree-based classification pipelines, although its benefits remain constrained by dataset size and class imbalance. A supplementary comparison with Random Forest and Gradient Boosting, tuned under the same GridSearchCV protocol, showed that both ensemble methods achieved numerically higher accuracy, precision, recall, and F1-score than the Decision Tree (82.70% and 82.16% accuracy, respectively, versus 80.54%); however, paired McNemar's tests indicated that these differences were not statistically significant at this sample size, so the practical choice between a single interpretable Decision Tree and a higher-capacity ensemble should be guided by the application's relative priorities of transparency versus predictive performance. Future research should therefore explore combining hyperparameter tuning with class-balancing techniques and ensemble methods, evaluated on a larger sample, to further improve recall for minority income classes.

ACKNOWLEDGEMENT

The author would like to express sincere gratitude to Universitas Potensi Utama and Universitas Muhammadiyah Asahan for their institutional support during the preparation of this research. The author also extends appreciation to both institutions for providing the academic resources, facilities, and encouragement that contributed to the successful completion of this study.

REFERENSI

- [1] Mao, R., Shi, X., & Shi, Z. (2025). A decision tree classification algorithm based on two-term RS-entropy. *Entropy*, 27(10), 1069. <https://doi.org/10.3390/e27101069>
- [2] Maitsa, A., & Winarsih, N. A. (2025). Comparison of hyperparameter tuning in decision tree and random forest algorithms for song genre classification. *Journal of Applied Informatics and Computing (JAIC)*, 9(4), 1825–1834.
- [3] Pathak, A. K., Chaubey, M., & Gupta, M. (2024). Randomized-grid search for hyperparameter tuning in decision tree model to improve performance of cardiovascular disease classification. *arXiv preprint*, arXiv:2411.18234.
- [4] Zhang, Y., Li, Q., & Xin, Y. (2024). Research on eight machine learning algorithms' applicability on different characteristics data sets in medical classification tasks. *Frontiers in Computational Neuroscience*, 18, 1345575.
- [5] Aaboub, F., Chamlal, H., & Ouaderhman, T. (2023). Statistical analysis of various splitting criteria for decision trees. *Journal of Algorithms & Computational Technology*, 17, 1–12.
- [6] Ramadhan, R. F., & Ashari, W. M. (2024). Performance comparison of random forest and decision tree algorithms for anomaly detection in networks. *Journal of Applied Informatics and Computing*, 8(2).
- [7] Aulianoor, A. B., & Koprari, M. (2024). Comparative analysis of the performance of decision tree and random forest algorithms in SQL injection attack detection. *Journal of Applied Informatics and Computing*, 8(1).
- [8] Damayanti, A., & Baita, A. (2024). Comparison of support vector machine (SVM) and random forest (RF) algorithm performance with random undersampling technique to predict gestational diabetes mellitus risk. *Journal of Applied Informatics and Computing*, 8(2).
- [9] Mantovani, R. G., Cerri, R., Vanschoren, J., Horváth, T., Barbon Junior, S., & de Carvalho, A. C. P. L. F. (2018). Better trees: An empirical study on hyperparameter tuning of classification decision tree induction algorithms. *arXiv preprint*, arXiv:1812.02207.
- [10] Fiagbe, R. (2023). Classification of adult income using decision tree. *Data Science and Data Mining*, Paper 3. UCF STARS Repository.
- [11] Saputra, F. D., & Budiman, F. (2026). Comparison of random forest and LSTM for Tokopedia sentiment analysis. *Journal of Applied Informatics and Computing*, 10(1).
- [12] Kohavi, R. (1996). Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 202–207.
- [13] Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, 2(01), 20–28. <https://doi.org/10.38094/jastt20165>

- [14] Song, Y.-Y., & Lu, Y. (2015). Decision tree methods: Applications for classification and prediction. *Shanghai Archives of Psychiatry*, 27(2), 130–135.
- [15] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81–106. <https://doi.org/10.1007/BF00116251>
- [16] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- [17] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357. <https://doi.org/10.1613/jair.953>
- [18] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, 2, 1137–1143.
- [19] Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, 2(3), 160. <https://doi.org/10.1007/s42979-021-00592-x>
- [20] Islam, M. A., Nag, A., Roy, N., Dey, A. R., Fahim, S. M. F. A., & Ghosh, A. (2023). An investigation into the prediction of annual income levels through the utilization of demographic features employing the modified UCI Adult dataset. *Proceedings of the 2023 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. IEEE. <https://doi.org/10.1109/ICCCIS60361.2023.10425394>