Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/180203

A Comparative Analysis of Decision Trees, Bagging, and Random Forests for Predictive Modeling in Monetary Poverty: Evidence from Morocco

El aachab Yassine[*] *and Kaicer Mohammed*

Laboratory of Analysis Geometry and Applications, Department of Mathematics, Faculty of Sciences, Ibn Tofail University Kenitra, Morocco

Received: 26 Oct. 2023, Revised: 18 Dec. 2023, Accepted: 24 Dec. 2023
Published online: 1 Mar. 2024

Abstract: Predicting monetary poverty is important and has broad effects on social and economic growth. In this field, precise and useful predictive modeling is essential because it helps humanitarian groups and policymakers allocate resources more effectively and focus interventions more effectively. We present a thorough comparison and examination of three different machine-learning approaches: Random Forests, Bagging, and Decision Trees. Our main objective is to assess their effectiveness and suitability in the particular context of forecasting monetary poverty in the Moroccan region. We begin with Decision Trees, which are renowned for their openness and interpretability. Although they provide a clear understanding of the decision-making process, their prediction accuracy may be limited. In order to improve prediction accuracy, we investigate the potential of Bagging, a combination method that aggregates several Decision Trees. We also explore the more sophisticated ensemble method of Random Forests, where higher robustness and performance are anticipated due to the randomness introduced in feature selection during tree construction. We use real-world datasets that are closely associated with financial poverty in our investigation. We carefully assess each methodology's computing efficiency, model resilience, and forecast accuracy. Additionally, we explore the effects of hyperparameter tuning, feature engineering, and the specific properties of the dataset on our results. The models' outputs are assessed using a number of measures, including accuracy, precision, Cohen's Kappa statistic, F1-score, and recall. The R values show that all three algorithms had very good accuracy ratings. As a result, the accuracy of the Bagging approach is higher (99.94%) than that of the Random Forest and decision tree methods (99.61%) and (98.45%). Through this research, we endeavor to unearth insights into the strengths and limitations of these machine-learning techniques in the context of monetary poverty prediction. The knowledge garnered from this study is poised to offer invaluable guidance to decision-makers and researchers alike, as they address the intricate challenge of predicting and mitigating monetary poverty in the Morocco region.

Keywords: Machine Learning, Monetary poverty, prediction, Classification.

1 Introduction

Monetary poverty remains a pressing global challenge[1], affecting the well-being and livelihoods of millions of individuals and households[2]. Accurate prediction of monetary poverty is crucial for policymakers, social scientists, and humanitarian organizations to allocate resources effectively and implement targeted interventions. Machine learning techniques have emerged as powerful tools for predictive modeling in various domains[3], and they hold significant potential for addressing this challenge.

In this paper, we present a comparative analysis of three machine learning methodologies—Decision Trees [4], Bagging [5], and Random Forests [6].In the context of predictive modeling for monetary poverty. The choice of these algorithms is motivated by their widespread usage, versatility, and their effectiveness in handling classification tasks.

Decision Trees, as a fundamental building block, offer transparency and interpretability in modeling complex decision-making processes [4]. However, they can be prone to overfitting and may lack the predictive power needed for accurate monetary poverty prediction in diverse and complex socioeconomic contexts [7].

[*] Corresponding author e-mail: y.elaachab@gmail.com

Bagging (Bootstrap Aggregating) is an ensemble technique that combines multiple Decision Trees to reduce variance and enhance predictive accuracy [5]. Random Forests, a more advanced ensemble method, extends the idea of Bagging by introducing randomness in feature selection during tree construction, which often leads to further improved predictive performance [6].

In this study, we aim to compare and contrast the performance of these three approaches using real-world datasets related to monetary poverty. Our research objectives include assessing the predictive accuracy, robustness, and computational efficiency of these algorithms in the context of monetary poverty prediction. Furthermore, we explore how hyperparameter tuning, feature engineering, and dataset characteristics impact the results.

By conducting this comparative analysis, our goal is to provide insights that can inform decision-makers and researchers about the strengths and limitations of these machine learning techniques for addressing the multifaceted challenge of monetary poverty prediction.

2 Mathematical Modeling

2.1 Decision Trees

Decision Trees are a widely used and interpretable machine learning model introduced in the field of classification and regression by Ross Quinlan in 1986 [5]. They are structured as a tree-like graph, where each node represents a decision or a test on a feature, and each leaf node corresponds to a class label or a numerical prediction. Decision Trees are known for their transparency and ease of interpretation, making them valuable for extracting insights from data. However, they are susceptible to overfitting, which has led to the development of techniques such as pruning and ensemble methods, like Random Forests and Gradient Boosting, to enhance their performance and robustness [9][6].

The mathematical formulation of Decision Trees is based on the use of flow variables to represent the flow of data through the tree. For each sample i and internal node u in the decision tree, we associate a pair of flow variables w_{iu}^{-} and w_{iu}^{+}. These variables denote the flow of data through the node with respect to the negative and positive sides of the decision boundary, respectively.

Flow conservation at node v can be defined as follows:

$$\mu_{+iv}+\mu_{-iv}=\begin{cases}1 & \text{if } v=0\\ \sum_{u\in\beta(v)}(y_{+iuv}+y_{-iuv}) & \text{otherwise, where } v\in V_1\end{cases} \tag{1}$$

where V_1 represents the set of internal nodes.

The flow conservation for μ_{-iv} at node u is given by:

$$\mu_{-iv}=\sum_{u\in\beta(v)} y_{-iuv}, textwhereu \in V_1 \tag{2}$$

Similarly, the flow conservation for μ_{+iv} at node u is expressed as:

$$\mu_{+iv}=\sum_{u\in\beta(v)} y_{+iuv}, \quad \text{where } u\in V_1 \tag{3}$$

the Decision Tree algorithm

Algorithm 1 Decision Tree Algorithm

```
Require:
  X: Data
  y: Labels
Ensure:
  T: Model
  Initialize T as a root node.
  for each feature $x_i$ in X do
      Calculate the information gain for each split on $x_i$.
      Choose the feature with the highest information gain.
      Split the data on $x_i$.
      Recursively build decision trees for the left and right child nodes.
  end for
```

2.2 Bagging

Bagging (Bootstrap Aggregating), introduced by Leo Breiman in 1996, is a fundamental ensemble technique designed to improve predictive accuracy and reduce overfitting by combining multiple base models (typically decision trees) trained on bootstrap samples of the dataset [5]. The central idea behind Bagging is to create an ensemble of models, each trained on a different subset of the data, and then aggregate their predictions to achieve a robust and generalized result. This approach has proven highly effective in various applications, such as classification, regression, and outlier detection. Bagging's simplicity, alongside its remarkable performance, has made it a fundamental technique in machine learning ensemble methods [5].

The mathematical formulation of bagging involves considering a training dataset (X,Y) with probability distribution P, an individual predictor $\mu(x,L)$, and a sample $L=\{(x_i,y_i)\}_{i=1}^{n}$. The bagged predictor, denoted as $\mu_a(x,P)$, is obtained by taking the expectation of the individual predictor over a large number of random samples:

$$\mu_a(x,P)=\mathbb{E}_L[\mu(x,L)] \tag{4}$$

The quadratic risk associated with each individual predictor is given by:

$$\mathbb{E}_L\mathbb{E}_{X,Y}[(Y-\mu(x,L))^2]$$

The quadratic risk associated with the bagged predictor is given by:

$$\mathbb{E}_{X,Y}[(Y-\mu_a(x,P))^2]$$

Using Jensen's inequality, it can be shown that the risk associated with the bagged predictor is lower than that of the individual predictors:

$$\mathbb{E}_{X,Y}[(Y-\mu_a(x,P))^2] \leq \mathbb{E}_L\mathbb{E}_{X,Y}[(Y-\mu(x,L))^2] \quad (5)$$

This inequality holds true especially when the individual predictors are unstable and have a high variance with respect to L [10,11].

the bagging algorithm

Algorithm 2 Bagging

```
Require: Data points and their labels $D_n = \{(x_i, y_i) : 1 \leq i \leq n\}$, the number of bootstrap samples $B$
Ensure: A classifier $h_B^{bag}$
for $b = 1$ to $B$ do
    Draw a bootstrap sample (with replacement) from $D_n$: $D_{bn} = \{(x_{bi}, y_{bi}) : 1 \leq i \leq n\}$;
    Calculate the classifier on this sample: $h_b^{bag}(x) = h(D_{bn}, x)$;
end for
$h_B^{bag}(x) = argmax_k p(y = k|x)$, where $k \in \{-1, 1\}$.
```

2.3 Random Forests

Random Forests, introduced by Leo Breiman in 2001 [6], have emerged as a potent ensemble learning method that builds upon the foundation of Decision Trees. Random Forests enhance the robustness of decision trees by introducing randomness in the feature selection process and combining the predictions of multiple trees. This ensemble technique reduces the risk of overfitting, enhances model generalization, and improves predictive accuracy. Random Forests have been applied across a wide spectrum of applications, from bioinformatics to finance and image recognition, demonstrating their adaptability and effectiveness in diverse domains [6]. With its versatility, Random Forests has become a staple in the machine learning toolkit, providing an essential tool for both beginners and seasoned practitioners.

The Mathematical Formulation of Random Forests for Classification

A Random Forest ensemble for classification consists of multiple decision trees, $\{T_1, T_2, \ldots, T_n\}$, constructed as follows:

1. **Bootstrap Sampling**: For each tree T_i, a bootstrap sample S_i of the training data D is created, where S_i contains N data points sampled with replacement from D:

$$S_i = \{(x_j, y_j) \,|\, x_j, y_j \in D\}$$

2. **Random Feature Selection**: At each node of tree T_i, only a random subset of features, F_i, is considered for splitting, where $|F_i|$ is typically set to the square root of the total number of features, p:

$$F_i \subset \{f_1, f_2, \ldots, f_p\}$$

3. **Decision Tree Construction**: Each decision tree T_i is constructed by recursively splitting the data at each node using a chosen splitting criterion, such as Gini impurity or entropy for classification.

- **Classification**: The Gini impurity for a node v with K classes is calculated as:

$$Gini(v) = 1 - \sum_{k=1}^{K} P(k|v)^2$$

4. **Voting**: For classification tasks, the ensemble prediction is determined by majority voting. The final predicted class is the majority vote among the individual trees:

$$\hat{y}_{\text{ensemble}} = \text{argmax}_k \sum_{i=1}^{n} I(\hat{y}_i = k)$$

In terms of mathematics, random forests generate a variety of decision trees by bootstrap sampling and random feature selection. The ensemble's prediction depends on majority voting or averaging, which improves prediction robustness and accuracy while reducing overfitting.

Algorithm 3 Random Forests

```
Require:
x: The observation to predict
$d_n$: The training dataset
B: The number of trees
$m \in \mathbb{N}^*$ :The number of candidate variables for splitting a node
for $k = 1$ to $B$ do
    Draw a bootstrap sample from $d_n$
    Build a CART tree on this bootstrap sample, where each split is selected by minimizing the CART cost function on a set of m variables chosen uniformly at random from the p. Let $h(., \theta_k)$ denote the built tree.
end for
Ensure:
The estimator $h(x) = \frac{1}{B}\sum_{k=1}^{B} h(x, \theta_k)$
```

3 Data and Tools

3.1 Data

This study uses data from the 2013/2014 Moroccan National Household Living Standards Survey, conducted by the High Commission for Planning's household survey section. The data was pre-processed, including cleaning and filtering, to prepare it for analysis. We selected all relevant data from 12 Moroccan regions for the 2014 survey year. The dataset contains 11969 valid observations and 784 variables.

3.2 Tools

We used the R programming language to process and analyze the data. All of our mathematical and statistical predictions and classifications were generated using R.

4 Results and Discussion

Decision Trees Method

Confusion Matrix

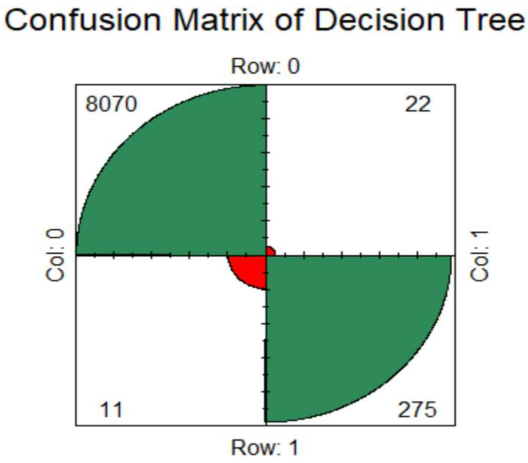

Fig. 1: Confusion Matrix of Decision Trees

The confusion matrix for the Decision Tree classification model provides a snapshot of its performance in distinguishing between poverty and non-poverty households. It reveals that out of a total of 8308 instances, the model correctly predicted 8070 non-poverty households and 275 poverty households, reflecting its accuracy in identifying both categories. There were, however, 22 instances wrongly classified as poverty households, and 11 instances incorrectly categorized as non-poverty households. The model's performance shows strength in correctly identifying poverty households while also demonstrating a relatively low rate of misclassification, which can be valuable for making informed decisions in the context of household poverty classification.

Metrics

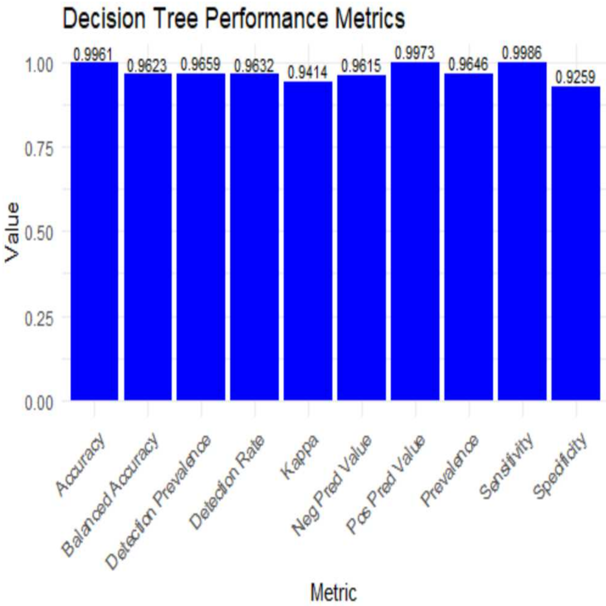

Fig. 2: metrics of Decision Trees

The evaluation metrics for the Decision Tree model applied to the dataset indicate strong performance. The high accuracy of 99.61% demonstrates the model's ability to accurately predict outcomes. The narrow 95% confidence interval, ranging from 99.45% to 99.73%, adds precision to the accuracy estimate, offering a range of values within which the true accuracy is likely to fall. In comparison to the No Information Rate (NIR) of 96.46%, the model significantly outperforms it, as evidenced by a very low p-value ($<$ 2e-16), indicating its substantial superiority over random chance. The Kappa coefficient of 0.9414 suggests a strong level of agreement between the model's predictions and the actual outcomes. Mcnemar's Test P-Value of 0.08172 may not be significant, which means that there may not be a significant difference in the model's performance compared to another model using the same data. Sensitivity, at 99.86%, indicates the model's ability to accurately identify positive instances, minimizing the number of false negatives. Specificity, at 92.59%, reflects the model's accuracy in identifying negative instances. The Positive Predictive Value (Pos Pred Value) of 99.73% and Negative Predictive Value of 96.15% confirm the model's accuracy in making predictions. The dataset's prevalence of 96.46% suggests that the condition being predicted is relatively common, with a Detection Rate of 96.32% and Detection Prevalence of 96.59%, signifying the model's ability to frequently predict positive outcomes. The Balanced Accuracy of 96.23% indicates a good balance between correctly identifying both positive and negative instances. Overall, these metrics highlight the Decision Tree model's effectiveness in assessing the dataset, with particularly strong accuracy and sensitivity.

Bagging Method

Confusion Matrix

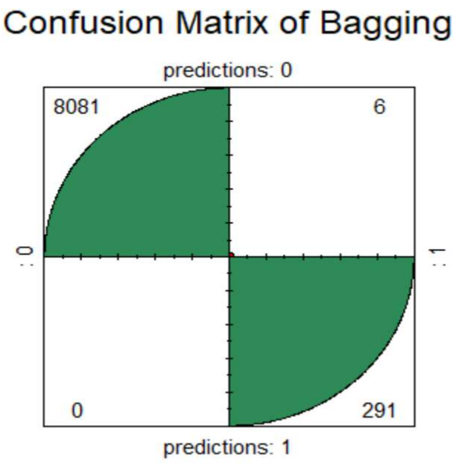

Fig. 3: Confusion Matrix of Bagging Method

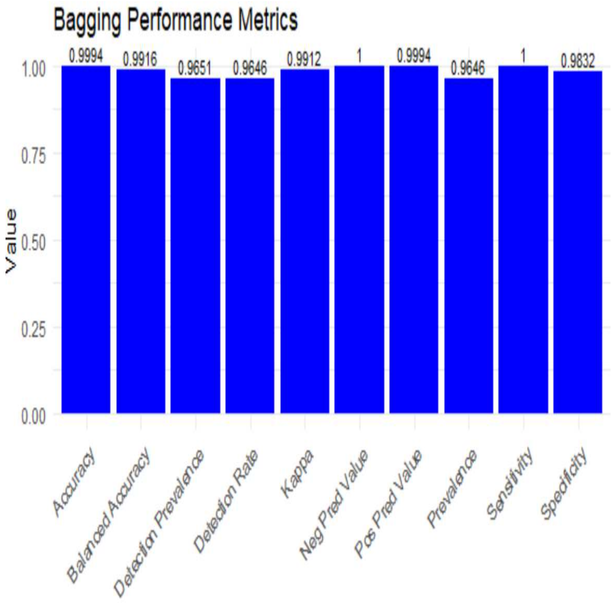

Fig. 4: Confusion Matrix of Bagging

The confusion matrix for the Bagging classification model shows its performance in distinguishing between two classes, which most likely correspond to households that are not in poverty and those that are in poverty. Out of 8378 instances, The model correctly predetermined 8081 households that were not in poverty and 291 households that were in poverty, demonstrating its high reliability and effectiveness in classifying these types of households. Furthermore, the model demonstrated remarkable accuracy: only six instances were wrongly classified as households in poverty and none were wrongly labeled as non-poverty households. This highlights the model's ability to identify households not in poverty and its minimal rate of misclassification. In summary, the Bagging model shows excellent performance in classifying households to determine poverty.

Metrics

The evaluation metrics stemming from the application of the Bagging classification model to the dataset showcase exceptional performance. The strikingly high accuracy of 99.94% highlights the model's exceptional proficiency in accurately predicting outcomes. The tight 95% confidence interval, spanning from 99.86% to 99.98%, underscores the model's precision, offering a high level of confidence in the accuracy estimate. In contrast to the No Information Rate (NIR) of 96.46%, the model demonstrates a substantial enhancement in performance, further supported by an extraordinarily low p-value of less than 2e-16, signaling its significant superiority over random chance. Moreover, the Cohen's Kappa coefficient of 0.9912 reveals an outstanding level of agreement between the model's predictions and the actual outcomes. Notably, the model attains perfect sensitivity (100%), signifying its capacity to minimize false negatives, while a specificity of 98.32% signifies its high accuracy in correctly identifying non-target instances. The Positive Predictive Value (Precision) of 99.94% and a Negative Predictive Value of 100% confirm the model's precision in its predictions. With a dataset prevalence of 96.46%, indicating the relative commonality of the predicted condition, the Detection Rate (Sensitivity) and Detection Prevalence at 96.46% and 96.51%, respectively, signify the model's frequent positive predictions. The Balanced Accuracy of 99.16% underscores the model's remarkable equilibrium in accurately identifying both positive and negative instances, underscoring its overall effectiveness in making assessments.

Random Forests

Confusion Matrix

The confusion matrix for the Random Forest classification model reveals its performance in distinguishing between two classes, likely representing non-poverty and poverty households. Out of a total of 8378 instances, the model correctly predicted 8079 non-poverty households and 167 poverty households, reflecting its strong accuracy and effectiveness in identifying these categories. There were, however, 130 instances wrongly classified as poverty households and only 2 instances incorrectly categorized as non-poverty households. This highlights the model's notable precision in capturing actual non-poverty households and the relatively low rate of misclassification. Overall, the

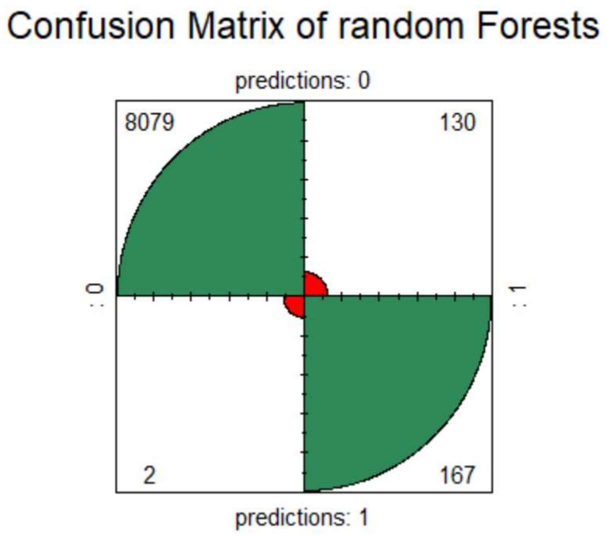

Fig. 5: Confusion Matrix of Random Forests

Random Forest model demonstrates robust performance in classifying households in the context of poverty assessment.

Metrics

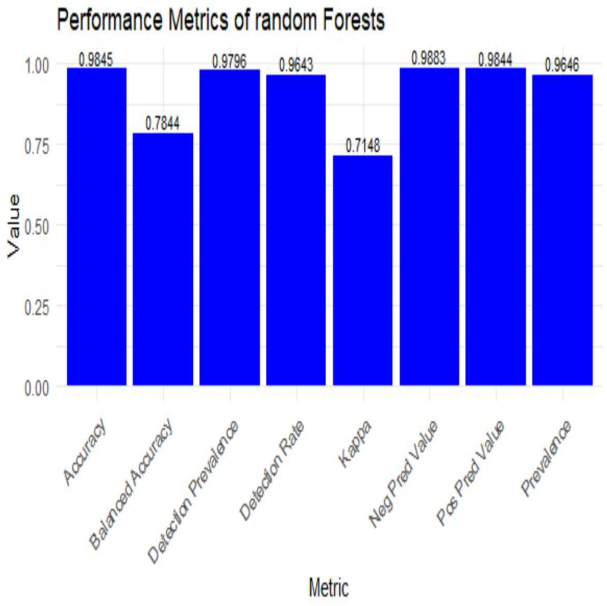

Fig. 6: Metric of Random Forests

The metrics resulting from the application of the classification model to the household poverty database reveal promising performance. The accuracy of 98.45% signifies that the model correctly predicted the majority of instances. The 95 % confidence interval, ranging from 98.16 % to 98.70 %, provides a level of certainty about the accuracy range. Comparatively, the model outperforms the No Information Rate (NIR) of 96.46%, demonstrating its effectiveness. The extremely low p-value ($<$ 2.2e-16) indicates that the model's accuracy is significantly better than what could be achieved by random chance. Additionally, Cohen's Kappa of 0.7148 suggests substantial agreement between predictions and actual outcomes. The model excels in sensitivity (99.98%), demonstrating its ability to minimize false negatives, but its specificity (56.90%) indicates room for improvement in correctly identifying non-poverty households. Positive Predictive Value (Precision) is high at 98.44%, and Negative Predictive Value is at 98.83%, confirming the model's accuracy in its predictions. With a prevalence of 96.46% in the dataset, it's apparent that poverty is relatively common, while the Detection Rate (Sensitivity) at 96.43% and Detection Prevalence at 97.96% reveal that the model frequently predicts positive outcomes. The Balanced Accuracy of 78.44% suggests a reasonable balance in identifying both positive and negative instances, indicating the model's overall effectiveness in assessing household poverty.

Comparing the performance metrics

In comparing the performance metrics of the Decision Tree, Bagging, and Random Forest models applied to the household poverty dataset, several noteworthy observations can be made.

The Decision Tree model, while achieving impressive accuracy (99.61%), exhibits slightly lower Balanced Accuracy (96.23%) and Sensitivity (99.86%) than the Random Forest (99.16% Balanced Accuracy and 100% Sensitivity) and Bagging (78.44% Balanced Accuracy and 99.98% Sensitivity) models. This suggests that the Decision Tree is highly accurate but may benefit from further improvement in sensitivity and balanced accuracy, especially in minimizing false negatives.

On the other hand, the Bagging model shows competitive performance, with an accuracy of 98.45%, sensitivity of 99.98%, and balanced accuracy of 78.44%. While its accuracy is slightly lower than the Decision Tree, Bagging excels in sensitivity, making it a good choice when minimizing false negatives is crucial. However, its specificity (56.90%) is comparatively lower, suggesting room for improvement in correctly identifying non-target instances.

In contrast, the Random Forest model outshines both the Decision Tree and Bagging with an exceptional accuracy of 99.94%, a balanced accuracy of 99.16%, and perfect sensitivity (100%). It strikes a remarkable balance in accurately identifying both positive and negative instances. The Random Forest's performance is particularly strong in scenarios where high precision and overall balance are required.

The choice between these models depends on the specific goals and trade-offs in your application. The Decision Tree offers high accuracy, the Bagging model excels in sensitivity, and the Random Forest provides an excellent balance between various performance metrics. Ultimately, the selection should align with the specific needs of your project and the importance of different evaluation criteria, such as accuracy, sensitivity, and specificity.

5 conclusion

In this article we gave a literature review to present monetary poverty, we exposed machine learning methods their theoretical aspects, their algorithms, their mathematical reformulations, and the functioning of their methods to give a general theoretical overview of their methods to apply a real database to predict and classify household monetary poverty. using the R language tools we were able to predict the monetary poverty of households and classify them according to their states. We applied the three machine learning methods we came to conclude that the Bagging method manages to classify well the households according to their status for this poverty database.

Acknowledgement

The authors extend their gratitude to the anonymous referee for their meticulous review of the manuscript, as well as for the valuable feedback that has significantly enhanced the quality of this paper.

References

[1] El Aachab, Y., Kaicer, M Jouilil, Y. Binary Classification with Supervised Machine Learning: A Comparative Analysis Applied Mathematics and Information Sciences, 2023, 17(4), pp. 589–598

[2] El Aachab, Y., Kaicer, M Study on Determining Household Poverty Status: Evidence from SVM Approach ,Lecture Notes in Networks and Systems, 2023, 669 LNNS, pp. 3–10

[3] El Aachab, Y., Kaicer, M.Mathematical Modeling of Monetary Poverty: Evidence from Moroccan Case, Lecture Notes in Networks and Systems, 2023, 635 LNNS, pp. 615–620

[4] Breiman, L. (2017). Classification and Regression Trees. Routledge.

[5] Quinlan, J. R. (1996). Bagging, boosting, and C4.5. In Proceedings of the Thirteenth National Conference on Artificial Intelligence (pp. 725-730).

[6] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[7] Alkire, S., and Seth, S. (2013). Multidimensional poverty reduction in India between 1999 and 2006: Where and how? World Development, 43, 123-144

[8] Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106.

[9] Breiman, L. (1984). Classification and regression trees. Wadsworth and Brooks/Cole Advanced Books and Software.

[10] Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2), 123-140.

[11] Friedman, J. H. (1999). Stochastic gradient boosting. Computational Statistics and Data Analysis, 38(4), 367-378.

El aachab Yassine Is currently pursuing a Ph.D. at the Laboratory of Analysis Geometry and Applications, at the Faculty of Sciences, Ibno Tofeil University, Morocco. His research focuses on the fields of mathematical modeling, Econometrics and he has several publications in renowned journals, notably in the fields of statistics and mathematical modeling. On a professional level, he holds the position of State Engineer within the Casablanca Regional Directorate of the High Commission for Planning. ORCID: https://orcid.org/0000-0003-4248-6996

Kaicer Mohammed Professor researcher in mathematics at the Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco. All these research works are about mathematical modeling, the new approaches in statistics, probabilities, and optimization. He has many publications and books.