Applied Mathematics & Information Sciences
*An International Journal*

http://dx.doi.org/10.18576/amis/170407

# Binary Classification with Supervised Machine Learning: A Comparative Analysis

*Yassine El aachab* [1,*] , *Mohammed Kaicer* [1] *and Youness Jouilil* [2]

[1]Laboratory of Analysis Geometry and Applications, Department of Mathematics, Faculty of Sciences, Ibn Tofail University Kenitra, Morocco
[2]Department of Economics, Faculty of Economics and Social Sciences of Mohammedia, Hassan II University of Casablanca, Morocco

Received: 14 Mar. 2023, Revised: 08 May 2023, Accepted: 6 Jun. 2023
Published online: 1 Jul. 2023

**Abstract:** Over the past decade, Machine Learning has become a practical approach for simulating and examining social issues, notably poverty, education, and health diseases. This study compares the performance of various machine learning methods especially Support Vector Machines (SVM), Decision Trees (DT), and Logistic Regression (LR) in predicting poverty status. For this purpose, the present contribution employs a micro dataset which has been extracted from the National Survey on Household Consumption and Expenditure 2013/2014. Several evaluation metrics such as accuracy, precision, Cohen's Kappa statistic, F1-score, and recall are used to evaluate the models' outputs. The R results indicate that the three algorithms achieved high accuracy scores. Therefore, the decision trees have more improvements in terms of accuracy (99.61%) compared to LR (91.09%) and linear kernel SVM methods ( 99.24%).

**Keywords:** Prediction, Decision Trees, Logistic Regression, Support Vector Machines, Classification, Poverty.

## 1 Introduction

Machine Learning, a subset of artificial intelligence, has lately gained popularity for its capacity to automate difficult procedures and make data-driven predictions. Machine learning algorithms have been used in a variety of industries, including financial services, healthcare, and marketing [1,2]. The growing amount of data available has been one of the main drivers of the success of ML [3].

The purpose of this document is to compare statistical classical and machine learning approaches in terms of their accuracy, interpretability, and performance on different types of data sets. Furthermore, we aim to provide insights into when to use each algorithm based on the characteristics of the data and the requirements of the problem. More specifically, we focus on comparing three popular machine learning techniques especially the Support Vector Machines (SVM), Decision Trees (DT), and Logistic Regression (LR) for classification tasks.

Logistic Regression, introduced by David Cox in 1958, is a widely used algorithm for binary classification [2]. Meanwhile, decision trees are a prominent machine learning approach that is employed for both classification and regression tasks. In addition, the SVM is a decision support tool that is utilized for both classification and regression problems [4,5]. The previous algorithms have been used extensively in various applications, and each has its own advantages and limitations.

Indeed, classification is a critical topic in machine learning, with the goal of predicting the class label of an instance based on its attributes. Logistic regression, decision trees, and SVM are popular algorithms used for classification tasks.

In this paper, we will compare these algorithms based on their strengths and weaknesses and provide insights into when each one might be more appropriate for a given classification problem. By examining their underlying principles, mathematical formulations, and implementation details, we aim to provide a comprehensive comparison for the purpose of classification [6,7,8,9].

The rest of this work is structured as follows. We present, in section 2, the materials and research methods needed to conduct this work. In section 3, we describe the database used in our paper and the tool used to manipulate it and expose the results. Then, we discuss the results of the comparison in section 4. In section 5, we conclude.

* Corresponding author e-mail: yassine.elaachab@uit.ac.ma

## 2 Materials and Methods

### 2.1 Logistic Regression

Logistic Regression is a well-known statistical method for solving binary classification problems. It is used to simulate the relationship that exists between a dependent variable that is binary and other variables that are independent which can either be continuous or categorical. The logistic regression model maps the structure of the relationship that is established by the independent variables and the dependent variable in the form of a logistic function, which can then be used to make predictions [7,8,9,10,11,12].

For the mathematical reformulation. Let $y$ be the binary dependent variable, and $x_1, x_2, ..., x_n$ be independent variables. The logistic regression is represented as follows [13,14,15,16]:

$$h(y=1|x) = \frac{1}{1+exp^{-(\beta_0+\sum_{j=1}^{n}\beta_j x_j)}} \tag{1}$$

Where $\beta_0, \beta_1, \beta_2, ..., \beta_n$ represent the coefficients of the model that are calculated based on the data and $h(y=1|x)$ is the probability of $y$ being 1 given $x$.

The logistic regression model is trained by optimizing the parameters $\beta_0, \beta_1, \beta_2, ..., \beta_n$ to maximize the likelihood of the observed data. The likelihood function is [14,15,16]:

$$L(\beta) = \prod_{j=1}^{n} h(y_j|x_j)^{y_j}(1-h(y_j|x_j))^{1-y_j} \tag{2}$$

where $n$ is the number of samples, $y_j$ is the observed dependent variable, and $x_j$ is the corresponding independent variables for each sample.

The logistic regression algorithm can be exposed as :

**Algorithm 1** Logistic Regression Algorithm

```
procedure LOGISTIC REGRESSION(X, y)
    m ← number of the training examples in X
    n ← number of the training features in X
    Initialize the weight vector w with zeros
    Repeat until w converge:
        for j = 1 to m:
            z_j = w^T X_i
            h_j = 1/(1+e^{-z_j})
            w ← w + α(y_i − h_j)X_j
    return w
end procedure
```

### 2.2 Decision Trees

Decision trees are a popular machine-learning method used for classification and regression applications. The underlying principle of decision trees is to iteratively partition the feature space into subsets that have characteristics as homogeneous in terms of the target variable as possible [5,17,18].

The decision trees algorithm builds a tree model where each internal node symbolizes a test on a feature and each leaf node symbolizes a predicted class label or target value. The algorithm works by selecting the feature that best splits the data based on some criterion, and then recursively applying the same process to each subset of data defined by the split until some stopping criterion is met (e.g., maximum tree depth, minimum number of samples per leaf) [19,20].

Hence, let $\mathscr{X}$ denotes the feature space, and $\mathscr{Y}$ the target space [6,7,21,22,23] :

$$S = (x_i, y_i)_{i=1}^{n}, \ \ where \ \ x_i \in \mathscr{X} \ and \ y_i \in \mathscr{Y} \tag{3}$$

Each internal node $j$ is associated with a feature $f_j$ and a split point $s_j$, and the test at node $j$ is defined as [6,7,21,22,23]:

$$[f_j < s_j] \tag{4}$$

We can formulate the decision trees learning problem as finding a tree $T$ that minimizes the empirical risk:

$$R(T) = \frac{1}{n}\sum_{i=1}^{n} E(y_i, \hat{y}_i) \tag{5}$$

where the loss function $E(y, \hat{y})$ calculates the difference at a leaf node between the real target value $y$ and the anticipated target value $\hat{y}$. The squared error loss is a popular option for regression issues.

$$E(y, \hat{y}) = (y - \hat{y})^2 \tag{6}$$

While for classification problems, a common choice is the loss 0-1

$$E(y, \hat{y}) = [y \neq \hat{y}] \tag{7}$$

To learn the decision trees, we can use a recursive algorithm that selects the best feature and split point at each internal node based on some splitting criterion and stops when some stopping criterion is met. One common splitting criterion for classification problems is information gain, which measures the reduction in entropy of the target variable due to the split, defined as [6,7,21,22,23]:

$$\mathrm{IG}(S, f) = \mathrm{G}(S) - \sum_{v \in \text{values of}(f)} \frac{|S_v|}{|S|}\mathrm{G}(S_v) \tag{8}$$

where $S$ stands for the present set of training samples, $f$ stands for the feature under test, values of$(f)$ stands for the set of potential feature values, and $S_v$ stands for the subset of samples with feature value $v$.

With :

$$G(S) = -\sum_{y \in \mathscr{Y}} p(y|S) \log p(y|S) \tag{9}$$

is the entropy of the target variable in set $S$, and $p(y|S)$ is the proportion of samples in set $S$ that have target value $y$. The best feature and split point are selected as those that maximize information gain [6,7,21,22,23].

Once, the decision trees are learned, By moving through the tree from the root to a leaf node depending on the results of the tests at each internal node and returning the anticipated target value at the leaf node, we can use it to predict the target value for a new input feature vector $x$ [6,7,21,22,23].

The Decision Trees algorithm could be expressed as follows:

**Algorithm 2** Decision Trees Algorithm

```
procedure BUILDTREE(X, y)
    Initialize an empty tree T
    if stopping criterion is met then
        Assign the majority class label or target value to the leaf node
    else
        Select the best feature f and split point s that maximizes some criterion (like information gain, Gini impurity)
        Add a new internal node to T with test f < s
        Partition the data into two subsets based on the test
        Recursively call BuildTree() on each subset, appending the resulting subtree to the internal node
    end if
    return T
end procedure
```

## 2.3 Support Vector Machines

SVM is a learning algorithm for classification, regression data, and identification the outliers [24]. The theoretical framework of SVM can be divided into three main components:

1. Maximizing the Margin: Finding the hyperplane that divides the classes in the feature space with the largest margin is the basic objective of SVM. The margin is determined by the separation between the nearest support vectors (also known as data points) and the hyperplane [24,25,26,27,28].
2. Finding the Optimal Hyperplane: SVM solves a quadratic optimization model with restrictions to determine the best hyperplane. The constraints ensure that the hyperplane is separated from the closest data points, while the objective is to maximize the margin [24,25,26,27,28].
3. Handling Non-Linearities: If the data cannot be separated linearly, SVM uses a kernel function to map the data into a space with increased dimensions. The data is changed by the kernel function into a space that allows for linear separability. SVM determines the ideal hyperplane in the original feature space by maximizing the margin in the transformed space [24, 25,26,27,28].

The optimization issue can be expressed numerically as follows:

$$\min_{w,b} \frac{1}{2} |w|^2 \tag{10}$$

subject to

$$y_i(w^T x_i + b) \geq 1 \text{ for } i = 1,2,\ldots,N \tag{11}$$

Where $w$ is the normal vector to the hyperplane, $b$ is the bias, $x_i$ is the $i^{th}$ sample in the feature space, $y_i$ is the corresponding class label, and $N$ is the number of samples [29,30,31].

The kernel function is a crucial component in Support Vector Machines (SVM), which allows the algorithm to handle non-linearly separable data. The kernel function turns the data into a space which is characterized by a higher dimension and where it can be separated linearly [32,33,34,35].

The kernel function to be used is determined by the characteristics of the data and the type of problem to be solved. It is common to try different kernel functions and choose the one that gives the best results.

A kernel function can be defined as a function that transforms the data input into a space that is characterized by a higher dimension, where the data becomes linearly separable. The mathematical representation of a kernel function can be defined as follows [32,33,34,35]:

$$Ker(x,x') = \phi(x)^T \phi(x') \tag{12}$$

where $Ker(x,x')$ is the kernel function that converts the data input $x$ and $x'$ in a universe that is characterized by a larger dimension, and $\phi(x)$is the function of mapping that turns the data input in a universe that is characterized by a larger dimension [32,33,34,35].

The linear kernel is defined as [32,33,34,35]:

$$Ker(x,x') = x^T x' \tag{13}$$

The polynomial kernel is defined as [32,33,34,35] :

$$Ker(x,x') = (x^T x' + c)^d \tag{14}$$

with $c$ is a constant and $d$ is the polynomial's degree.

The RBF kernel (radial basis function) is defined as [32,33,34,35]:

$$Ker(x,x') = e^{-\gamma |x-x'|^2} \tag{15}$$

$\gamma$ : is a positive constant.

The sigmoid kernel is defined as follows [32,33,34,35]:

$$Ker(x,x') = \tanh(\gamma x^T x' + r) \tag{16}$$

where $\gamma$ is a scalar parameter and $r$ is a bias term.

The SVM Algorithm can be structured as:

**Algorithm 3** SVM Algorithm with Kernel

1: **procedure** THE($X, y, C$, kernel)
2: $m \leftarrow$ number of training examples in $X$
3: Initialize the Lagrange multipliers $\alpha_1, \alpha_2, \ldots, \alpha_m$
4: Solve the quadratic optimization problem to find the optimal $\alpha$
5: Use the $\alpha$ values to find the support vectors $S$
6: Use the support vectors to find the hyperplane parameters $w$ and $b$
7: For a new sample $x$, evaluate the prediction as $\text{sign}(f(x)) = \text{sign}(\sum_{i \in S} \alpha_i y_i \text{kernel}(x, x_i) + b)$
8: **return** hyperplane parameters $w$ and $b$
9: **end procedure**

### 2.4 Confusion Matrix And metrics

Confusion Matrix: A table that summarizes a classification model's performance. It includes information about the True Positives (TrPo), False Positives (FaPo), True Negatives (TrNe), and False Negatives (FaNe) [32,33,34,35].

| Confusion Matrix | | |
|---|---|---|
| | **Predicted(P)** | **Predicted(N)** |
| **Actual Positives(P)** | TrPo | FaNe |
| **Actual Negatives(N)** | FaPo | TrNe |

Accuracy: The model's percentage of true predictions. It is computed as follows: (TrPo + TrNe) / Total [32,33,34,35].

$$Accuracy = \frac{TrPo + TrNe}{TrPo + TrNe + FaPo + FaNe} \tag{17}$$

Precision: The proportion of True Positives among the cases predicted as positive [32,33,34,35]. It is calculated as TrPo / (TrPo + FaPo).

$$Precision = \frac{TrPo}{TrPo + FaPo} \tag{18}$$

Recall (Sensitivity): The percentage of True Positives successfully detected by the model [32,33,34,35]. It is computed as TrPo / (TrPo + FaNe).

$$Recall = \frac{TrPo}{TrPo + FaNe} \tag{19}$$

F1-Score: Precision and recall are balanced by the harmonic mean. When such classes are imbalanced, it is a good sign of the model's overall performance [32,33,34,35].

$$F1 - Score = 2 \times \frac{Precisionl \times Recall}{Precision + Recall} \tag{20}$$

The ROC curve (receiver operating characteristic) graphically represents the efficiency of a binary classification method by comparing the proportion of real positives to the proportion of false positives [32,33,34,35].

$$TPR = \frac{TrPo}{TrPo + FaNe} \tag{21}$$

$$FPR = \frac{FaPo}{TrNe + FaPo} \tag{22}$$

## 3 Data and Tools

### 3.1 Data

The data used in this study is from Morocco's National Household Living Standards Survey (2013/2014), This is carried out by the High Commission for Planning's household survey section. We started with a pre-processing step, such as cleansing the data, filtering it, and so on, in order to carry out our experiment and analysis of the prediction and classification of the phenomenon. For the 2014 survey year, we picked all relevant data from 12 Moroccan regions. There are 11969 valid data (observations) and 784 variables in total.

### 3.2 Tools

To handle and analyze the data, we used the R programming language. As a result, all of our mathematical and statistical prediction and classification results were generated using the R software.

## 4 Results and Discussion

### 4.1 Result of Logistic Regression

The binary classification model's performance is indicated by an area under the ROC curve (AUC) of 96.07 %. The ROC (Receiver Operating Characteristic) curve plots the true positive rate (TPR) versus the false positive rate (FPR) for different classification thresholds to reflect the model's performance. The AUC calculates the area under the ROC curve to assess the general efficiency of the model. An AUC value of 96.07% indicates that the model is able to accurately differentiate between positive and negative classifications.

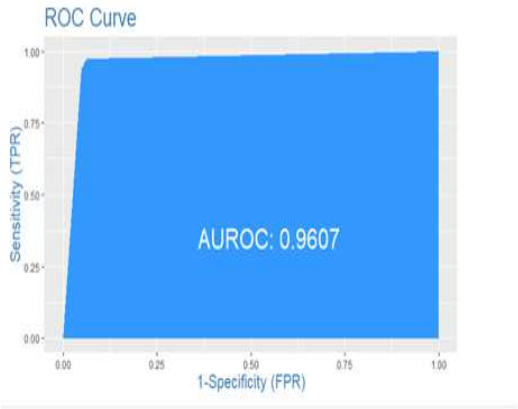


**Fig. 1:** ROC curve of the Logistic Regression.

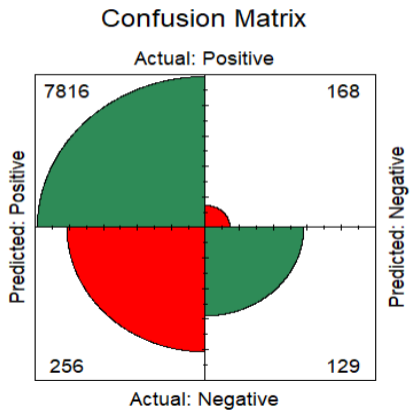


**Fig. 2:** confusion matrix of the Logistic Regression

**Table 1:** Metrics of the Logistic Regression

| Metrics | Value% |
|---|---|
| Accuracy | 91.09 |
| Specificity | 97.78 |
| Precision | 46.59 |
| F1-score | 38.31 |
| Cohen's Kappa statistic | 12.36 |

In the given confusion matrix, the number of true positives (TP) is 7816, which means that the model correctly identified 7816 people as poor. False positives (FP) are 168, which indicates that the model incorrectly classified 168 non-poor individuals as poor. False negatives (FN) are 265, which represents the number of actual poor individuals who were misclassified as non-poor. The number of true negatives (TN) is 129, which indicates that the model correctly classified 129 non-poor individuals as non-poor.

The accuracy of the model is 91.09 % which means that it correctly classifies 91.09% of the instances in the dataset. The precision of the model is 46.59% which means that out of all the positive instances predicted by the model, 46.59% of them are actually positive. The recall of the model is 32.80% which means that out of all the actual positive instances, the model was able to correctly identify 32.80% of them.A Kappa value of 0.1236 suggests that there is only a slight agreement between the true and predicted class labels, indicating that the classification model is not performing well. The interpretation of the Kappa value varies between 0 and 1, where 0 indicates no agreement between the true and predicted class labels and 1 indicates perfect agreement. The F1-score of the model is 38.31% which is the harmonic mean of precision and recall and gives a single score that balances both the precision and recall. A higher F1-score indicates a better balance between precision and recall.

## *4.2 Result of Decision Trees*

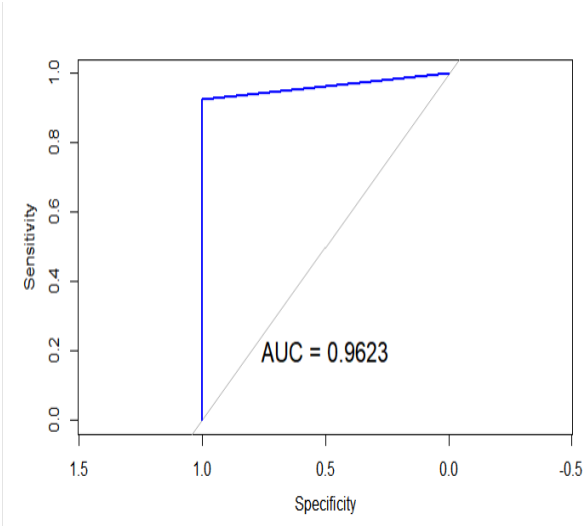


**Fig. 3:** ROC curve of the Logistic Regression.

The AUC (Area Under the Curve) of 0.9623 indicates a high level of predicting poverty. An AUC of 1.0 represents a perfect prediction, while an AUC of 0.5 represents a model that performs no better than random guessing. Therefore, an AUC of 0.9623 indicates that the model has a high level of discrimination power in distinguishing between individuals who are living in poverty and those who are not. In other words, the model properly classifies 96.23 % of the cases, which is a good

indication of the model's performance in predicting poverty.

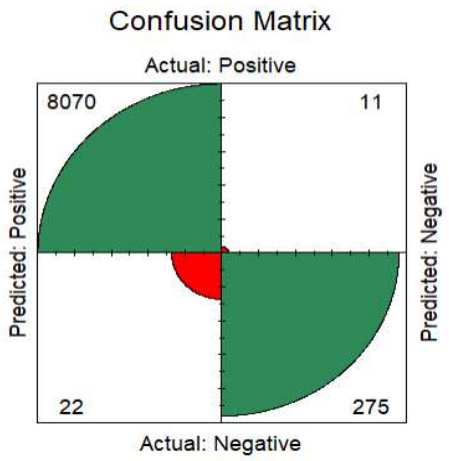


**Fig. 4:** confusion matrix of the Decision Trees

The confusion matrix for the prediction and classification of household poverty indicates that out of 8,378 households, 275 were correctly classified as living in poverty (true positives), while 11 were incorrectly classified as not living in poverty (false negatives). In addition, 8,070 households were correctly classified as not living in poverty (true negatives), while 22 were incorrectly classified as living in poverty (false positives). Overall, the decision trees model achieved high accuracy in predicting household poverty status, with a relatively low number of false positives and false negatives [12].

**Table 2:** Metrics of the Decision Trees

| Metrics | Value% |
|---|---|
| Accuracy | 99.61 |
| Specificity | 92.59 |
| Precision | 99.86 |
| F1-score | 99.79 |
| Cohen's Kappa statistic | 94.14 |

The accuracy of the model is 0.9961, which means that 99.61% of the classifications were correct. The model has a sensitivity of 0.9986, meaning that it correctly identifies 99.86% of households that are living in poverty. The model has a specificity of 0.9259, which means that it correctly identifies 92.59% of households that are not living in poverty. The positive predictive value (PPV) of the model is 0.9973, which means that when the model predicts that a household is living in poverty, there is a 99.73% chance that it is correct. The negative predictive value (NPV) of the model is 0.9615, which means that when the model predicts that a household is not living in poverty, there is a 96.15% chance that it is correct. Finally, the Kappa coefficient of the model is 0.9414, indicating that the model has a strong agreement with the actual classifications.

## *4.3 Result of SVM*

### 4.3.1 SVM using a Linear kernel

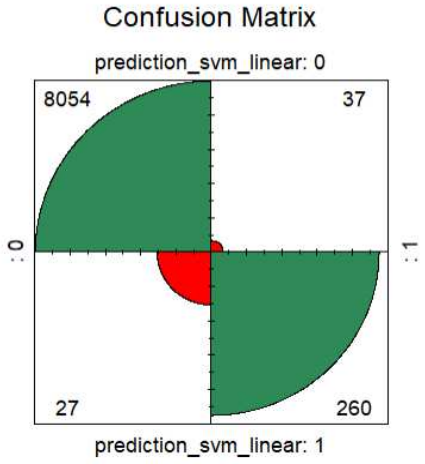


**Fig. 5:** Confusion Matrix of SVM using a Linear kernel

**Table 3:** Metrics of SVM using a Linear kernel

| Metrics | Value% |
|---|---|
| Accuracy | 99.24 |
| Precision | 95.42 |
| Recall | 99.67 |
| F1-score | 89.08 |
| Specificity | 87.54 |
| Cohen's Kappa statistic | 88.65 |

The values in the matrix show the number of instances that were correctly and incorrectly classified. The number 8054 in the top left cell indicates that 8054 instances of non-poor were correctly classified. The number 260 in the bottom right cell indicates that 260 instances of poor were correctly classified. The accuracy metric, which is 0.9924, indicates that 99.24 % of the instances were correctly classified. Precision, which is 0.8748, measures the proportion of positive instances that are correctly classified. Recall, which is 0.9967, measures the proportion of actual positive instances that are correctly

classified. The F1-score, which is 0.8908, is the harmonic mean of precision and recall, and provides a balanced view of the performance of the classifier.

The specificity metric, which is 0.8754, indicates the proportion of negative instances that are correctly classified. Cohen's Kappa statistic, which is 0.8865, is a measure of the agreement between the predicted classes and the actual classes, adjusted for chance agreement [12].

4.3.2 SVM using a Radial kernel

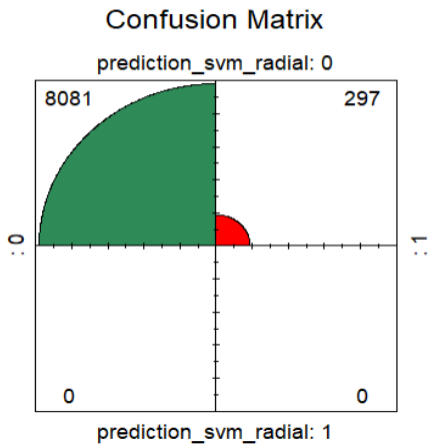


**Fig. 6:** Confusion Matrix of SVM with Radial kernel

**Table 4:** Metrics of SVM using a Radial kernel

| Metrics | Value% |
|---|---|
| Accuracy | 96.46 |
| Precision | 0 |
| Recall | - |
| F1-score | - |
| Specificity | 96.49 |
| Cohen's Kappa statistic | 0 |

The results of the Support Vector Machine (SVM) approach with a radial kernel for the prediction and classification of households as poor and non-poor are evaluated using the confusion matrix. The matrix shows that the SVM approach had 8081 true positive predictions and 297 false positive predictions, while there were no false negatives and 0 true negatives.

Based on the given confusion matrix, the classifier achieved an accuracy of 0.9646 or 96.46 %, indicating that it correctly classified most of the samples. However, the precision for the positive class is 0, which means that the classifier did not correctly identify any of the positive samples.

Since there were no true positive predictions, it is not possible to calculate recall or F1-score for the positive class. The specificity for the negative class is 0.9649, which means that the classifier correctly identified a large majority of the negative samples.

Finally, the Cohen's Kappa statistic for this classifier and dataset is 0, indicating no agreement beyond chance between the classifier and the true labels. This suggests that the classifier may not be performing well on this particular dataset, and further investigation and improvement may be needed to achieve better classification results [12].

4.3.3 SVM using a sigmoid kernel

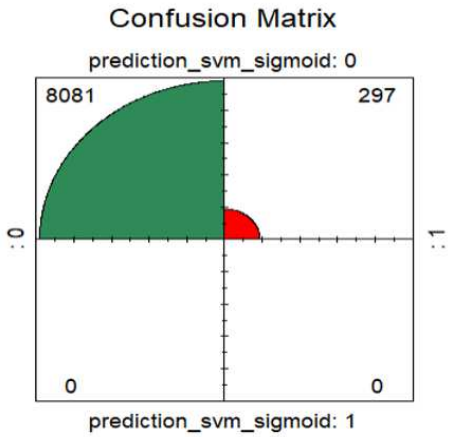


**Fig. 7:** Confusion Matrix of SVM using a sigmoid kernel

**Table 5:** Metrics of SVM using a sigmoid kernel

| Metrics | Value% |
|---|---|
| Accuracy | 96.46 |
| Precision | 0 |
| Recall | - |
| F1-score | - |
| Specificity | 96.49 |
| Cohen's Kappa statistic | 0 |

The confusion matrix for the prediction SVM sigmoid model shows that there were 8081 instances of class 0 and 297 instances of class 1. Out of all the instances of class 0, the model correctly predicted 8081 as class 0, while it

incorrectly predicted 297 instances of class 1 as class 0. There were no instances of class 1 that were correctly predicted by the model.

On what concerns SVM with the sigmoid kernel, we manage to find the same result as the different metrics we compare ourselves with SVM with the radial kernel.

4.3.4 SVM using a polynomial kernel

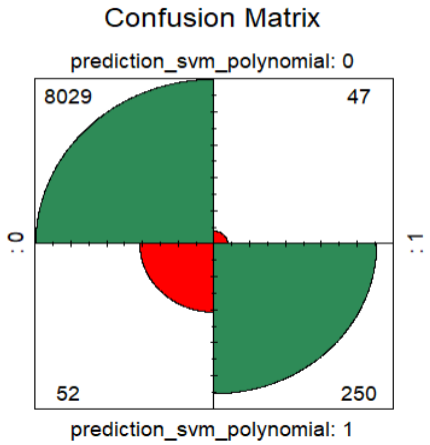


**Fig. 8:** Confusion Matrix SVM using a polynomial kernel .

**Table 6:** Metrics of SVM using a polynomial kernel

| Metrics | Value% |
|---|---|
| Accuracy | 98.82 |
| Precision | 84.12 |
| Recall | 99.36 |
| F1-score | 83.45 |
| Specificity | 84.18 |
| Cohen's Kappa statistic | 82.86 |

The confusion matrix for the prediction polynomial SVM model shows that there are 8029 instances of class 0 (non-poor) and 52 instances of class 1(poor). Out of all the instances of class 0, the model correctly predicted 8029 as class 0, while it incorrectly predicted 47 instances of class 0 as class 1. On the other hand, the model correctly predicted 52 instances of class 1, and 250 instances were incorrectly predicted as class 0.

The prediction results for poor and non-poor Modeling with SVM using a polynomial kernel have achieved an accuracy of 0.9882. This means that the model correctly predicted the poor and non-poor status for approximately 98.82 % of the individuals.

The precision score of 0.8412 implies that 84.12 percent of the people the model predicted to be poor actually were poor. On the other hand, recall assesses how well the model can identify every poor person, and it has a score of 0.9936, suggesting that 99.36 % of the poor people were accurately recognized by the model.

The harmonic mean of recall and precision, also known as the F1-score, shows the equilibrium efficacy of the model's predictions in terms of accuracy and recall with a value of 0.8345. The specificity score, which has a quantity of 0.8418 and indicates how well the model can identify non-poor people, indicates that 84.18 % of non-poor people were properly recognized by the model.

Finally, the model's predictions and the ground truth have a good agreement, as shown by Cohen's Kappa, a statistic that measures inter-annotator agreement, which has a value of 0.8286.

Subsection: Comparison between DT, LR, and SVM

The comparison between the DT, LR, and SVM for the prediction of poor and non-poor households is presented. All models were evaluated using a confusion matrix as well as different metrics for evaluation such as precision, recall, accuracy, F1-score, specificity, and Cohen's Kappa. Most of these algorithms gave us good results based on comparing metrics for DT, LR, and SVM with polynomial and linear kernels.

## 5 Conclusion

This manuscript compares the results of the prediction and classification of households as poor and non-poor using three machine-learning approaches namely DT, LR, and SVM. The evaluation was performed using confusion matrices and various metrics, such as Cohen's Kappa statistic, F1-score, recall, accuracy, and precision.

The results show that DT, LR, and SVM with polynomial and linear kernels performed well in classifying households as poor and non-poor. In addition, the F1-score, which is a balance between precision and recall, indicates that good performance is observed in logistic regression and SVM models in classifying households.

In conclusion, DT, LR, and SVM are effective methods for classifying households. The choice between the models will be determined by the problem's specific requirements, and the trade-off between precision and accuracy desired. In our case, we can say that we were able to identify the poor and non-poor with excellent classification and decision tree metrics.

## References

[1] Y. LeCun,Y. Bengio, and G.Hinton, Deep learning, Nature,7553,521, 436-444, (2015).

[2] J. D. Kelleher, B. Mac Namee, and A. D'Arcy, Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies, MIT Press, (2015).
[3] D.R. Cox, The regression analysis of binary sequences (Doctoral dissertation, University of London), (1958).
[4] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J.Stone, Classification and regression trees, CRC press, (1984).
[5] J. R. Quinlan, Induction of decision trees. Machine learning, 1st ed,vol.1, 81-106, (1986).
[6] T. Hastie, R. Tibshirani, and J. Friedman, The elements of statistical learning: data mining, inference, and prediction, Springer Science and Business Media,(2009).
[7] G. James, D. Witten, T. Hastie, and R. Tibshirani, An introduction to statistical learning , New York: Springer,112, (2013).
[8] M. N. Murty, Data mining techniques for soft computing, Intelligent Systems and Soft Computing, Proceedings, International Conference on IEEE ,1-8, (1997).
[9] L. Breiman, Random forests, Machine learning, 45(1), 5-32, (2001).
[10] J.Youness, The individual determinants of Financial Inclusion in Morocco: An Exploratory Study in the case of a credit application, 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, 1-6, (2022).
[11] H. Lechheb, H. Ouakil, and Y. Jouilil, Economic growth, poverty, and income inequality: Implications for lower-And middle-income countries in the era of globalization. The Journal of Private Equity 23(1), 137–145, (2019).
[12] Y. El Aachab, and M. Kaicer, Study on Determining Household Poverty Status: Evidence from SVM Approach, International Conference on Digital Technologies and Applications,vol. 669, 3–10, (2023).
[13] Y. El Aachab, and M. Kaicer, Mathematical Modeling of Monetary Poverty: Evidence from Moroccan Case,The International Conference on Artificial Intelligence and Smart Environment,vol.635,615–620, (2023).
[14] C. Bishop, Pattern Recognition and Machine Learning by, Chapter 4, Linear Models for Classification,106-144,(2006).
[15] G. James, D. Witten, T. Hastie, and R. Tibshirani,An Introduction to Statistical Learning, Chapter 4, Classification, 121-145, (2013).
[16] K. P. Murphy, Machine Learning: A Probabilistic Perspective, Chapter 17, Discriminative Models, pp. 475-488, (2012).
[17] E. Alpaydin,Introduction to machine learning. MIT Press. Chapter 4, Decision Trees, (2010).
[18] C. M. Bishop,Pattern recognition and machine learning. Springer. Chapter 14, Decision Trees, (2006).
[19] J. Han, M. Kamber, and J. Pei,Data mining: Concepts and techniques. Morgan Kaufmann. Chapter 8, Classification: Basic Concepts, Decision Trees, and Model Evaluation,(2011).
[20] K.P. Murphy,Machine learning: A probabilistic perspective. MIT Press. Chapter 18, Decision Trees and Ensemble Learning, (2012).
[21] P. Domingos, and M. Pazzani, On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 29(2-3), 103-130,(1997).
[22] E. Alpaydin, Introduction to machine learning (2nd ed.). Cambridge, MA: MIT Press, (2010).
[23] C. M. Bishop, Pattern recognition and machine learning (1st ed.). Springe, (2006).
[24] V.N. Vapnik,The nature of statistical learning theory. Springer-Verlag New York, Inc, (1995).
[25] C. Cortes, and V. Vapnik, Support-vector networks. Machine learning, 20(3), 273-297,(1995).
[26] N. Cristianini, and J. Shawe-Taylor, An introduction to support vector machines and other kernel-based learning methods. Cambridge university press, (2000).
[27] B. Schölkopf, and A.J. Smola, Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, (2002).
[28] N. Görnitz, K.R. Müller, and G. Rätsch, An introduction to kernel-based learning algorithms. Journal of Machine Learning Research, 14(Jan), 1-48, (2013).
[29] B.Scholkopf, A. J.Smola, and K. R. Müller, Input space vs feature space in kernel-based methods, Proceedings of the Sixteenth International Conference on Machine Learning,320-327, Morgan Kaufmann, (1999).
[30] I. Steinwart, and A. Christmann, Support vector machines. Springer Science and Business Media, (2008).
[31] Od. Maimon and L. Rokach, Data Mining and Knowledge Discovery Handbook, Section 5.2, Evaluation of Classification Models, Springer, (2010).
[32] H. Chen, N. Petropulu, and P.S.P. Wang, Handbook of Pattern Recognition and Computer Vision, e. World Scientific, Chapter 5, Evaluation and Comparison of Classifiers, (2000).
[33] M. Steinbach, and V. Kumar, Introduction to Data Mining, by Pang-Ning Tan, Addison-Wesley. Chapter 5, "Evaluation, (2006),
[34] McGraw-Hill, Machine Learning by Tom Mitchell, Chapter 3, Evaluating Hypotheses: The Strategy of Experimental Design, (1997).
[35] R. Duda and P. Hart, Pattern Classification. John Wiley and Sons, Chapter 4, Evaluation of the Classifier,(1973).

**El aachab Yassine** Researcher in mathematics, at the Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco. His research areas are applied mathematics, statistical modeling and its applications, artificial intelligence, applied econometrics, optimization, and data science.

**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.

**Jouilil Youness** Researcher in econometrics and applied mathematics, Faculty of Economics and Social Sciences of Mohammedia, Hassan II University of Casablanca, Morocco. His research areas focus on applied econometrics, public policy evaluation, development economics, artificial intelligence, and data science. He has many publications in national and international scientific journals and has contributed to several collective studies. y.jouilil@gmail.com