STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–24.
Published online in International Academic Press (www.IAPress.org)

Artificial Intelligence and Machine Learning Models for Credit Risk Prediction in Morocco

Asmaa Faris [1,*], Mostafa Elhachloufi [2]

[1]*Laboratory of Applied Modeling for Economics and Management, University Hassan II Casablanca, Morocco*
[2]*Department of Statistics and Applied Mathematics for Economics and Management,University Hassan II Casablanca, Morocco*

Abstract This study investigates the application of artificial intelligence and machine learning models for credit risk prediction using a real-world dataset collected from a Moroccan credit institution. The data reflect clients' demographic, socio-economic, and financial characteristics, as well as behavioral information related to credit history and interactions with the institution. Six supervised learning models Logistic Regression, Random Forest, Support Vector Machine, Decision Tree, k-Nearest Neighbors, and Naïve Bayes were trained and evaluated using key performance metrics such as accuracy, recall, F1-score, AUC, and average precision. Results indicate that Random Forest outperformed all other models, demonstrating strong discriminative power and robustness to class imbalance, while Logistic Regression provided consistent and interpretable baseline performance. These findings highlight the effectiveness of ensemble and margin-based methods in credit scoring applications and emphasize the importance of feature importance analysis for transparent and informed decision-making in financial risk assessment.

Keywords Credit Risk, Machine Learning, Artificial Intelligence, Predictive Modeling

DOI: 10.19139/soic-2310-5070-2486

1. Introduction

Credit risk assessment is a cornerstone of the financial sector, significantly affecting loan approval processes, portfolio management, and the overall stability of banking institutions. Accurate prediction of a borrower's likelihood of default is essential not only for minimizing financial losses but also for ensuring the equitable distribution of credit across individuals and businesses [56]. In traditional credit risk models, statistical techniques such as logistic regression [8] and discriminant analysis [9] have been widely applied. These methods, while effective in many cases, have limitations when dealing with the complexities of modern financial environments. The increasing diversity of data sources, the non-linear nature of creditworthiness, and the growing interdependence of various financial factors make it difficult for these models to fully capture the underlying risk [19].

The introduction of Artificial Intelligence (AI) and Machine Learning (ML) has marked a significant paradigm shift in the field of credit risk evaluation. ML algorithms have the potential to surpass traditional statistical approaches by uncovering complex, hidden patterns in vast and diverse datasets, such as transactional history, behavioral traits, and non-traditional credit indicators [40]. Algorithms like Decision Trees [41], Support Vector Machines [42], Random Forests [43], and Deep Learning models [44] have shown exceptional performance in terms of predictive accuracy and generalizability, offering a more nuanced understanding of credit risk [49]. These models can process structured, semi-structured, and unstructured data, enabling financial institutions to make more informed lending decisions based on comprehensive risk profiles.

[*] Correspondence to: Asmaa Faris (Email: asmaa-faris-etu@etu.univh2c.ma). Laboratory of Applied Modeling for Economics and Management, Faculty of Legal, Economic, and Social Sciences - Ain Sbaa, University Hassan II Casablanca, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

In recent years, several empirical studies have demonstrated the effectiveness of AI models in credit risk prediction. For example, Brown and Mues (2012) compared logistic regression with tree-based classifiers on UK credit data and showed that ensemble methods like Random Forests consistently outperformed traditional models in terms of AUC and misclassification rate [50]. Malekipirbazari and Aksakalli (2015) applied Gradient Boosting and Random Forest to peer-to-peer lending data and achieved over 90% accuracy, highlighting the ability of machine learning to deal with high-dimensional, imbalanced data [51]. More recently, Zhang et al. (2020) demonstrated that deep learning models, such as multilayer perceptrons and convolutional networks, significantly improved default prediction when trained on transactional sequences and behavioral signals [49].

Building on these advances, recent studies from 2020 to 2025 further confirm the growing maturity of AI-based credit scoring. Kim and Sohn (2020) leveraged Long Short-Term Memory (LSTM) networks to handle sequential borrower data and achieved substantial improvements over static models in predicting non-traditional borrower default risks [59]. In a more recent investigation, Ribeiro et al. (2021) explored model-agnostic interpretability techniques in financial services, emphasizing the regulatory necessity of transparent credit decisions [60]. Zhang et al. (2023) introduced a hybrid ensemble framework combining XGBoost with SHAP-based explanations for high-stakes lending environments, achieving high interpretability and performance on real-world datasets [61]. Similarly, Faris et al. (2024) proposed a regionalized deep learning model tailored to local lending contexts in Morocco, enhancing credit access while maintaining robustness and fairness [62].

In addition to improving predictive performance, AI-driven credit risk models can enhance operational efficiency by automating much of the risk assessment process. By reducing reliance on human judgment, these models also help mitigate the biases often found in traditional credit scoring [53], leading to more equitable outcomes in lending decisions. Moreover, AI-based systems can integrate alternative data sources, such as social media activity, online behaviors, and real-time transaction data, to generate a more complete and holistic view of an applicant's financial reliability [1]. This integration of diverse data sources can be particularly beneficial in assessing individuals with limited credit histories or those operating in underserved markets [54].

However, despite their advantages, AI-driven models present several challenges that need to be carefully addressed. One of the most significant concerns is model interpretability. Complex models, such as deep neural networks, are often viewed as ”black boxes,” which makes it difficult to justify their decisions to regulators, consumers, and other stakeholders [55]. This issue of transparency in AI is critical, particularly in the highly regulated field of financial services, where accountability and fairness are paramount [2]. Other challenges include data privacy concerns, the need for robust bias mitigation strategies, and the evolving landscape of regulatory requirements [58]. In this context, explainable AI (XAI) tools such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) have been proposed to improve transparency and trust in AI decisions [52].

Given the evolving complexity of credit risk factors and the limitations of traditional methods, there is a pressing need to explore and benchmark AI and ML techniques that can handle heterogeneous data and complex relationships more effectively. This study is motivated by the demand for more accurate, interpretable, and fair credit risk models that can be trusted by both financial institutions and regulatory bodies. In this context, this paper offers a comprehensive comparison between classical statistical credit risk models and state-of-the-art machine learning algorithms using real-world financial datasets. Furthermore, we evaluate model interpretability techniques aimed at enhancing transparency for complex AI models and investigate feature engineering strategies that leverage alternative data sources to improve predictive power. Finally, we discuss the practical implications and regulatory challenges associated with deploying AI-driven credit risk systems.

Through this work, we aim to contribute to the ongoing discourse on AI-based credit risk assessment by providing valuable insights into model selection, feature engineering, and regulatory considerations. Our goal is to guide financial institutions towards adopting more effective, fair, and transparent lending practices, thereby improving the overall stability and inclusiveness of financial services.

2. Methodology

2.1. Data Description

This study utilizes a real-world dataset obtained from a Moroccan credit institution, covering client credit data for the year 2019. The dataset includes 45,212 financial records, each corresponding to an individual client. These records provide a comprehensive overview of clients' demographic, socio-economic, and financial characteristics, along with their credit behavior and interactions with the institution.

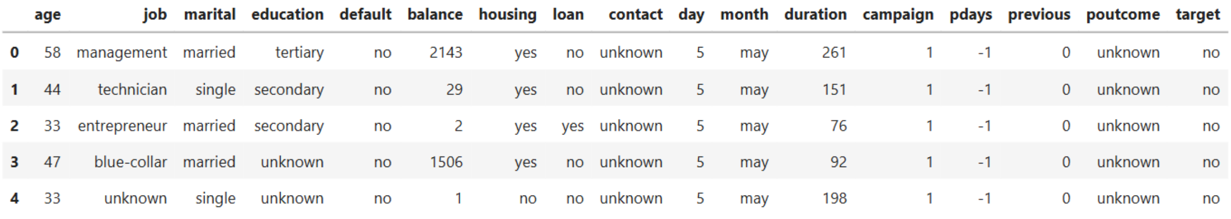
Table 1. Data Attributes and Their Types

Attribute	**Type**
Age	Numerical
Job	Categorical
Marital Status	Categorical
Education	Categorical
Balance	Numerical
Housing Loan	Binary (0/1)
Personal Loan	Binary (0/1)
Contact Type	Categorical
Last Contact Day/Month	Categorical/Numerical
Duration	Numerical
Campaign Contacts	Numerical
Days Since Last Contact	Numerical
Previous Campaign Contacts	Numerical
Previous Outcome	Categorical
Target (Credit Profile)	Binary (0/1)

The attributes in the Table 1 encompass variables such as age, job type, marital status, education level, account balance, loan history, and previous marketing interactions. The dataset is particularly suitable for developing machine learning models for credit risk classification, as it contains both qualitative and quantitative variables.

The target variable, labeled as Credit Profile, is binary, where a value of 1 indicates that the client is considered to have a good credit profile, and 0 indicates a bad credit profile, based on the institution's internal risk assessment policies and past default behavior.

The Figure 1 illustrates the structure of the dataset, which includes both client-related and campaign-specific variables collected by the bank. The input variables used in this study consist of client-related and campaign-specific data.



	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	target
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no

Figure 1. An overview of the dataset (Python result)

Client data include: (1) age (numeric), (2) job type (categorical: ”admin.”, ”unknown”, ”unemployed”, ”management”, ”housemaid”, ”entrepreneur”, ”student”, ”blue-collar”, ”self-employed”, ”retired”, ”technician”,

”services”), (3) marital status (categorical: ”married”, ”divorced”, ”single” – with ”divorced” encompassing both divorced and widowed individuals), (4) education level (categorical: ”unknown”, ”secondary”, ”primary”, ”tertiary”), (5) default status indicating whether the client has credit in default (binary: ”yes”, ”no”), (6) balance, representing the average yearly balance in euros (numeric), (7) housing loan status (binary: ”yes”, ”no”), and (8) personal loan status (binary: ”yes”, ”no”). Variables related to the last contact of the current marketing campaign include: (9) contact communication type (categorical: ”unknown”, ”telephone”, ”cellular”), (10) day of the month of last contact (numeric), (11) month of last contact (categorical: ”jan”, ”feb”, ..., ”dec”), and (12) duration of the last contact in seconds (numeric). Additional attributes include: (13) campaign, which refers to the number of contacts during the current campaign (numeric, including the last contact), (14) pdays, the number of days since the client was last contacted (numeric, with -1 meaning no previous contact), (15) previous, the number of contacts made before the current campaign (numeric), and (16) poutcome, the result of the previous marketing campaign (categorical: ”unknown”, ”other”, ”failure”, ”success”). The target output variable is (17) whether the client subscribed to a term deposit (binary: ”yes”, ”no”).

The Table 2 above presents descriptive statistics for selected quantitative variables in the dataset.

Table 2. Descriptive Statistics of Selected Variables

Variable	Count	Mean	Median	Std	Min	Max	Missing
age	45,211	40.936	39.0	10.619	18	95	0
balance	45,211	1362.272	448.0	3044.766	-8019	102,127	0
day	45,211	15.806	16.0	8.322	1	31	0
duration	45,211	258.163	180.0	257.528	0	4918	0
campaign	45,211	2.764	2.0	3.098	1	63	0
pdays	45,211	40.198	-1.0	100.129	-1	871	0
previous	45,211	0.580	0.0	2.303	0	275	0

The **age** variable, with 45,211 observations and no missing values, has a mean of approximately 40.9 years and a median of 39, indicating a relatively symmetric distribution with moderate variability (standard deviation: 10.62). The **balance** variable shows a high level of dispersion (standard deviation: 3044.77), with values ranging from -8019 to 102,127 euros, suggesting the presence of significant outliers. The median balance is 448 euros, much lower than the mean, highlighting a right-skewed distribution.

The **day** variable, indicating the last contact day of the month, is uniformly distributed with a mean around 15.8. The **duration** of the last contact has a mean of 258 seconds and a high standard deviation (257.5), with a maximum value reaching 4918 seconds, again indicating the presence of long outlier calls. The **campaign** variable, representing the number of contacts made during the campaign, shows that most clients were contacted only 2 to 3 times (median: 2.0; mean: 2.76).

For the **pdays** variable, which records the number of days since the last contact before the current campaign, the median is -1, indicating that many clients had not been contacted previously. The standard deviation of over 100 reflects high variability among those who were previously contacted. Lastly, the **previous** variable, showing the number of past contacts before the campaign, has a mean of 0.58 and a median of 0, confirming that most clients had not been contacted before.

Overall, the variables exhibit varying degrees of skewness and dispersion, with some showing strong outlier effects, particularly in *balance* and *duration*, which may require normalization or transformation prior to modeling.

Figure 2 illustrates the client profile and structural characteristics of the dataset. The data reveal a client base predominantly composed of married individuals (60%), working in manual or technical occupations such as blue-collar and technician roles, with a secondary level of education (50–55%), and holding a housing loan (approximately 60%). A marked imbalance is observed in the target variable, where non-subscriptions (85–90%) significantly outweigh subscriptions (10–15%), posing challenges for predictive modeling unless addressed through resampling techniques.

Moreover, the dataset suffers from structural issues: key variables like *contact* (with 45–50% marked as "unknown") and *poutcome* (over 80% "unknown") have large proportions of missing or uninformative values, reducing their predictive usefulness. Additionally, campaign efforts are heavily concentrated in May, accounting for 70–80% of the contacts, which introduces a temporal bias. These features, typical of banking marketing datasets, require targeted data cleaning such as imputation of unknowns and creation of composite variables along with class balancing strategies to unlock the dataset's full predictive potential.

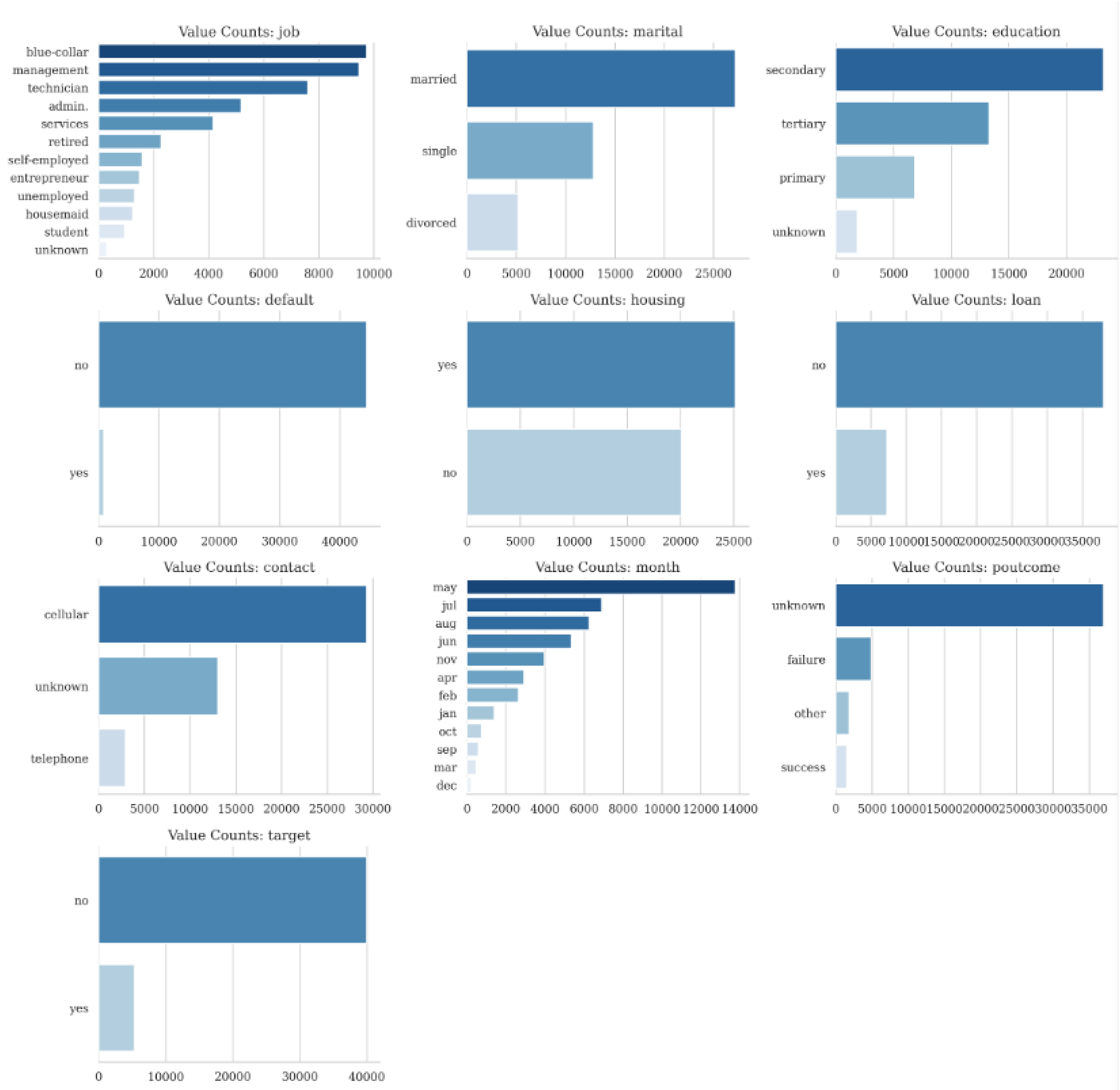
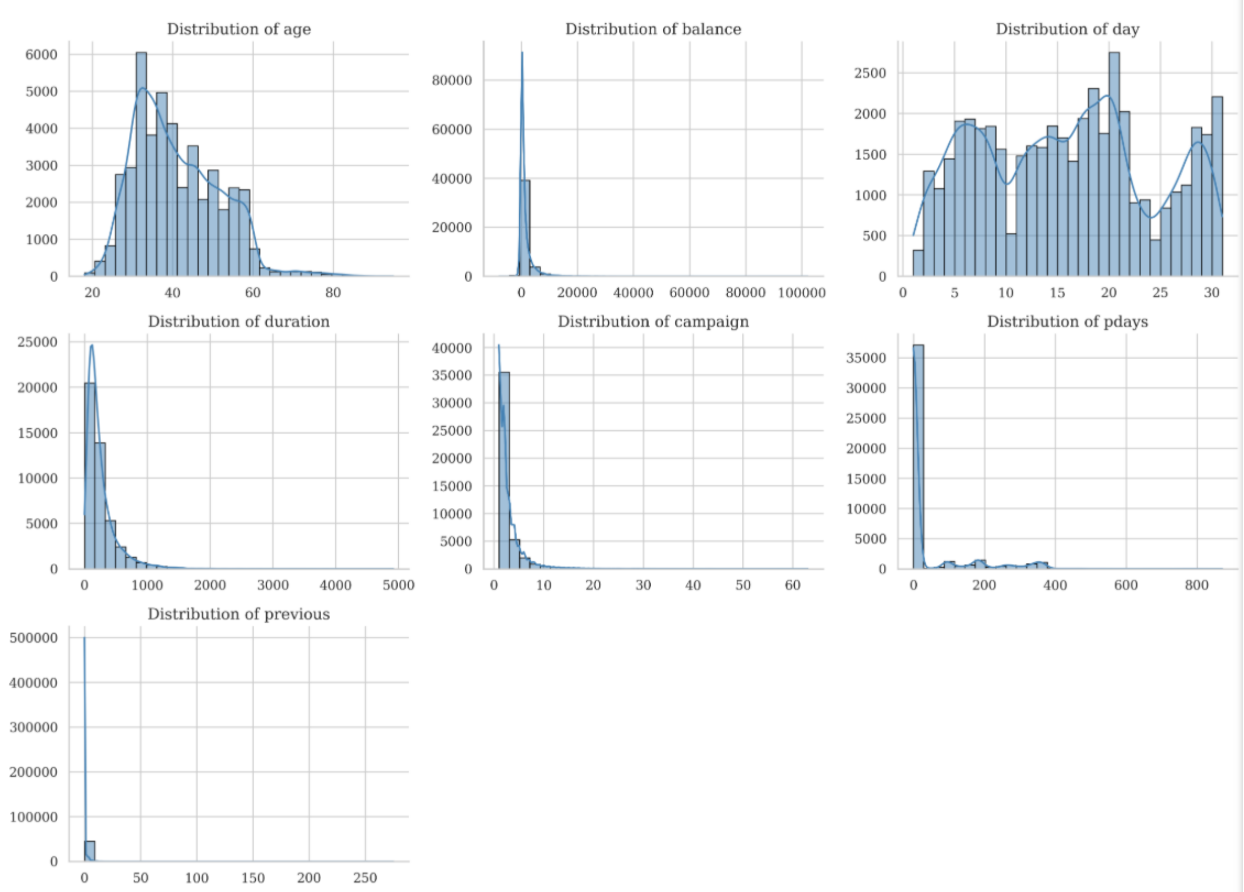


Figure 2. Categorical Variable Distributions

Figure 3 illustrates the distributions of key numerical variables in the dataset, which exhibit pronounced skewness and significant structural challenges. Regarding age, the distribution is approximately normal, centered around 40 years, with the majority of clients aged between 30 and 50. The bank balance shows extreme skewness, where 95% of accounts hold less than 20,000, while outliers reach exceptionally high values up to 100,000. Call duration follows an exponential distribution: 80% of interactions last less than 5 minutes, whereas calls exceeding 40 minutes remain rare.

For current contacts, a sharp peak at one contact characterizes the distribution, with 75% of clients contacted at most three times, although extreme cases reach up to 30 attempts. The variable measuring days since last contact (*pdays*) displays a clear bimodality, contrasting consecutive contacts (*pdays* = 0) with late re-engagements around 400 days. Previous contacts are dominated by an overwhelming proportion of zeros (85–90%), indicating a majority of clients without prior contact history. Finally, the day of the month exhibits an almost uniform distribution, with a slight peak toward the end of the month (days 25–31).

Recommended transformations include logarithmic scaling of *balance*, *duration*, and *campaign* variables to mitigate the impact of extreme values. The creation of composite variables, such as *has_previous_contact* or *contact_strategy* (based on *pdays*), will enhance analysis. Structural zeros in *previous* and *pdays* require specific recoding, either as missing values or as a dedicated category.



Distribution of age

Distribution of balance

Distribution of day

Distribution of duration

Distribution of campaign

Distribution of pdays

Distribution of previous

Figure 3. Distribution of Numerical Variables

2.2. *Data transformation*

Encoding

Label encoding alg. 1 is a widely used preprocessing technique that transforms categorical variables into numeric labels, enabling machine learning algorithms to handle non-numeric data effectively [33, 35]. By assigning a unique integer to each category, this method facilitates the integration of categorical features into predictive models. However, it can unintentionally introduce an ordinal relationship between categories, which may mislead algorithms that interpret numeric values as having magnitude or order [36]. Therefore, while label encoding is appropriate for nominal variables and well-suited for tree-based models such as random forests and gradient boosting [37, 38], other encoding methods like one-hot encoding or embeddings are often preferred when category order should not be implied [39].

Algorithm 1 Label Encoding for a categorical feature

```
Require: Dataset $D$ with categorical feature $F$
Ensure: Encoded numerical feature $F'$
Initialize empty mapping dictionary $M$
Initialize label counter $label \leftarrow 0$
for all unique category $c$ in $F$ do
    $M[c] \leftarrow label$
    $label \leftarrow label + 1$
end for
for all record $r$ in $D$ do
    Replace $r.F$ by $M[r.F]$
end for
return Encoded feature $F'$
```

Normalization

Data normalization alg. 2 is a fundamental preprocessing step in machine learning, particularly when working with numerical features that span different scales. One of the most commonly used techniques is Min-Max normalization, which linearly transforms the original data into a predefined range, typically $[0, 1]$. This method is especially beneficial for algorithms sensitive to feature scaling, such as neural networks, k-nearest neighbors (KNN), and gradient descent-based models, where large differences in value ranges can skew the learning process [66]. The Min-Max normalization technique subtracts the minimum value from each data point and divides the result by the range (maximum minus minimum), thereby rescaling the distribution while preserving the original shape and relationships [67]. Unlike z-score normalization, which centers the data around the mean, Min-Max scaling is often preferred when the dataset lacks significant outliers or when model interpretability in bounded intervals is important [68]. Its effectiveness has been demonstrated in numerous studies involving credit scoring, fraud detection, and time-series analysis where normalized input ensures convergence and stability during model training [69]. In this work, we apply Min-Max normalization to the relevant numerical features to ensure comparability and optimal performance across various machine learning classifiers.

Algorithm 2 Min-Max Normalization of a numerical feature

```
Require: Dataset $D$ with numerical feature $X$
Ensure: Normalized feature $X' \in [0, 1]$
$X_{min} \leftarrow \min(X)$
$X_{max} \leftarrow \max(X)$
for all record $r$ in $D$ do
    $r.X' \leftarrow \frac{r.X - X_{min}}{X_{max} - X_{min}}$
end for
return Normalized feature $X'$
```

Outlier treatment

The analysis of histograms(Figure 4) reveals several significant outliers: account balances exhibit extreme values exceeding 80,000, while the median remains below 20,000. Call durations include isolated cases surpassing 4,000 seconds, even though 95% of calls last less than 500 seconds. The number of previous contacts features prospects who have been contacted more than 200 times, in contrast to a median of zero. Similarly, the variable representing the number of days since the last contact reaches values above 600, despite a concentration at -1, which typically indicates new contacts. These anomalies may correspond to institutional clients, complex cases requiring extended follow-up, or reactivated prospects after long periods of inactivity. Their potential influence on predictive models warrants deeper investigation using statistical techniques such as Grubbs' test or logarithmic transformations to mitigate their impact.

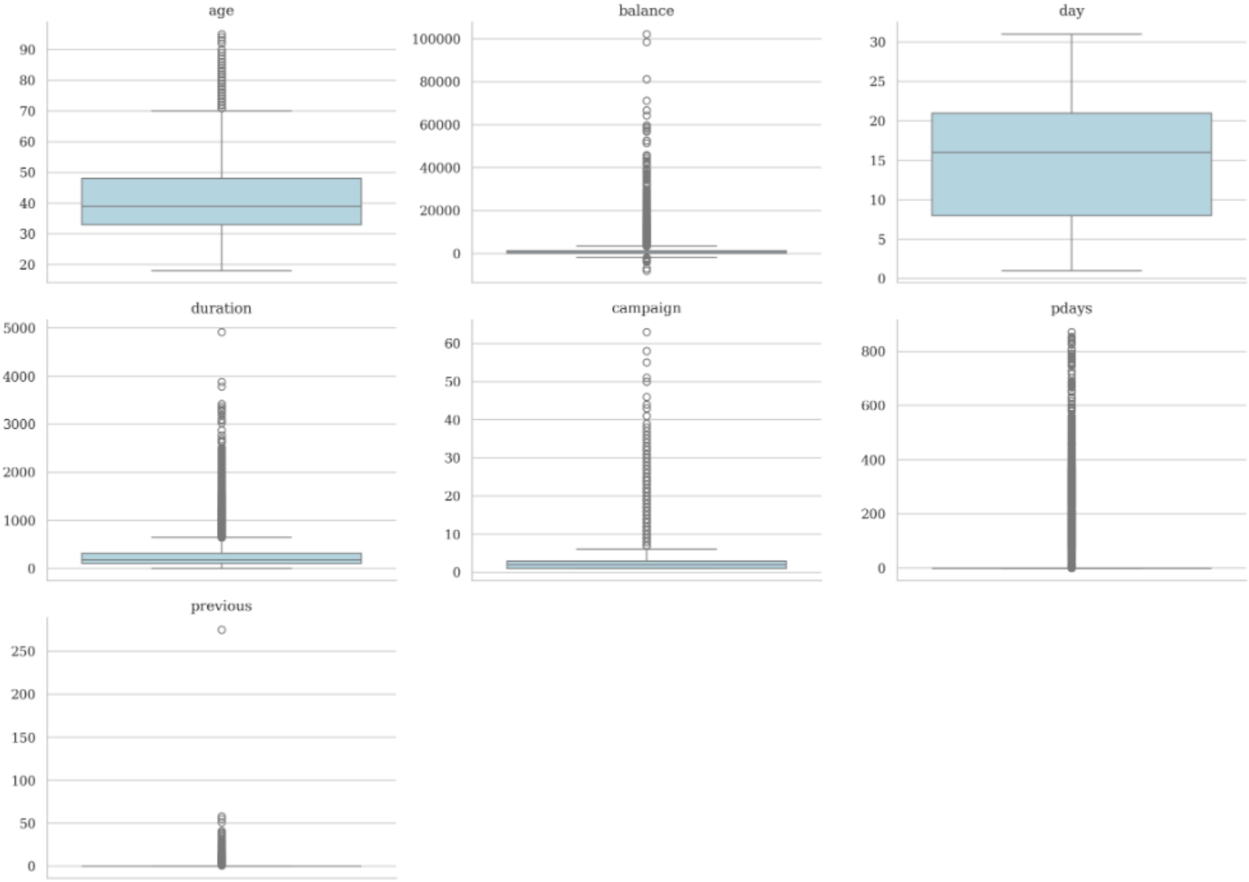


Figure 4. Boxplots for Outlier Detection in Numerical Variables

Outlier detection, as presented in alg. 3, is a crucial component of the data preprocessing pipeline. Extreme values can heavily skew statistical summaries such as mean and variance, leading to misleading insights and degraded predictive model performance. This effect is especially pronounced in sensitive algorithms like linear regression or k-nearest neighbors, where outliers can disproportionately influence the learned parameters or distance calculations. To systematically identify univariate outliers, the Interquartile Range (IQR) method is widely employed. The IQR is computed as the difference between the third quartile (Q_3) and the first quartile (Q_1), effectively capturing the range of the central 50% of the data distribution. Observations lying outside the interval $[Q_1 - 1.5 \times IQR, Q_3 + 1.5 \times IQR]$ are conventionally flagged as outliers [63].

This technique offers several advantages: it is non-parametric and thus robust to skewed or heavy-tailed distributions, unlike standard deviation-based approaches which assume normality and can be distorted by extreme values [68]. By relying on quartiles, the IQR method adapts to the data's intrinsic spread, providing a flexible and interpretable rule for outlier detection. After detection, handling outliers can take various forms depending on the dataset and analytical goals. Common strategies include replacing outlier values with the median (a robust central tendency measure), applying data transformations (e.g., logarithmic or Winsorizing), or outright removal of anomalous records if justified [64].

In the context of our study, we apply the IQR method to numerical features to systematically detect and treat outliers prior to normalization and model training. This step is fundamental to enhance model robustness and ensure that extreme but potentially spurious values do not bias the learning process. Consequently, the predictive models achieve greater stability and generalization performance, especially when dealing with real-world financial data often characterized by heavy tails and rare extreme observations.

Algorithm 3 Detection and Treatment of Outliers Using IQR Method

```
Require: Dataset $D$ with numerical feature $X$
Ensure: Dataset $D$ with outliers treated in feature $X$
Compute first quartile $Q_1 \leftarrow \text{percentile}(X, 25)$
Compute third quartile $Q_3 \leftarrow \text{percentile}(X, 75)$
Compute interquartile range $IQR \leftarrow Q_3 - Q_1$
Define lower bound $LB \leftarrow Q_1 - 1.5 \times IQR$
Define upper bound $UB \leftarrow Q_3 + 1.5 \times IQR$
for all record $r$ in $D$ do
    if $r.X < LB$ or $r.X > UB$ then
        Treat outlier (e.g., replace $r.X$ by median of $X$ or remove $r$)
    end if
end for
return Processed dataset $D$
```

Handling Class Imbalance Class imbalance represents a major challenge in supervised learning, particularly in credit risk assessment where the minority class (typically defaulters) is heavily underrepresented. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ denote a binary classification dataset, with $x_i \in \mathbb{R}^d$ as the feature vector and $y_i \in \{0, 1\}$ the target label, where $y_i = 1$ indicates a credit default. In an imbalanced scenario, the prior probabilities satisfy:

$$P(y = 1) \ll P(y = 0)$$

Standard classifiers trained to minimize empirical risk via overall accuracy:

$$\hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(h(x_i) \neq y_i)$$

tend to favor the majority class ($y = 0$), often producing trivial classifiers that ignore minority instances, leading to poor sensitivity.

where TP, FP, and FN are the true positives, false positives, and false negatives for the minority class, respectively. In financial risk modeling, such as credit scoring, false negatives (missed defaulters) are particularly costly [7, 34].

To address this, several strategies have been proposed:

- **Data-level approaches**: Oversampling the minority class or generating synthetic examples using SMOTE [3], or undersampling the majority class to rebalance the dataset.
- **Algorithm-level approaches**: Introducing cost-sensitive learning [4] or adjusting the decision threshold τ of a probabilistic classifier:

  $$\hat{y} = \begin{cases} 1 & \text{if } \hat{P}(y = 1|x) \geq \tau \\ 0 & \text{otherwise} \end{cases}$$

  to favor recall on the minority class.
- **Evaluation-level adaptation**: Using metrics such as AUC-ROC, F1-score, and precision-recall curves [5] instead of raw accuracy, as they provide a more informative evaluation under imbalance.

These approaches aim to improve the model's ability to detect minority class instances without sacrificing generalization, ensuring that models used in financial institutions are not only accurate but also risk-sensitive.

Algorithm 4 Handling Class Imbalance Using SMOTE and Class Weighting

```
Require: Imbalanced dataset $D$ with minority class $C_{min}$ and majority class $C_{maj}$
Ensure: Balanced dataset $D'$ and/or trained model with class weighting
Step 1: Analyze class distribution
Calculate class frequencies: $f_{min} \leftarrow$ count($C_{min}$), $f_{maj} \leftarrow$ count($C_{maj}$)
if $f_{min} \ll f_{maj}$ then
    Step 2: Apply SMOTE (Synthetic Minority Over-sampling Technique)
    for all sample $x_i$ in $C_{min}$ do
        Identify $k$ nearest neighbors of $x_i$ in $C_{min}$
        Generate synthetic samples by interpolating between $x_i$ and neighbors
    end for
    Augment $D$ with synthetic minority samples to obtain balanced dataset $D'$
else
    $D' \leftarrow D$                                          ▷ No resampling needed
end if
Step 3: Define class weights
Compute weights inversely proportional to class frequencies:
    $w_{min} = \frac{f_{maj} + f_{min}}{2 \times f_{min}}, \quad w_{maj} = \frac{f_{maj} + f_{min}}{2 \times f_{maj}}$
Step 4: Train model on $D'$ using class weights $w_{min}$ and $w_{maj}$
Pass class weights to the learning algorithm to penalize misclassification of minority class more heavily
return Balanced dataset $D'$ and trained weighted model
```

To address these issues, oversampling techniques have been widely applied, with the Synthetic Minority Oversampling Technique (SMOTE) being one of the most prominent methods. SMOTE algorithmically generates synthetic examples for the minority class by interpolating between existing minority class samples and their nearest neighbors in feature space, rather than simply duplicating observations. This approach effectively increases the representation of minority instances, producing a more balanced training dataset and helping the model learn a better decision boundary [16]. By creating synthetic points, SMOTE alleviates the risk of overfitting commonly associated with naive oversampling methods.

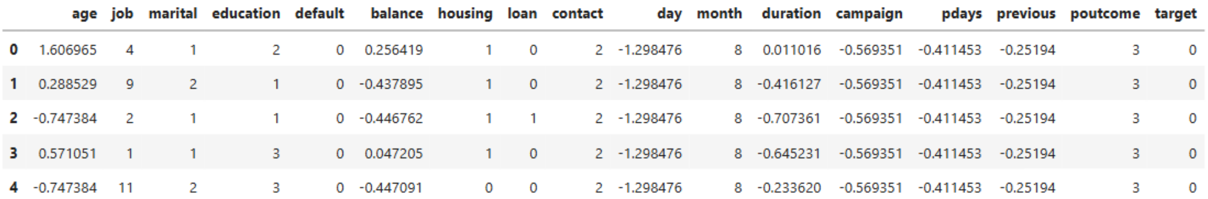
In addition to oversampling, class weighting strategies provide a complementary solution by modifying the model's loss function to penalize misclassification errors on the minority class more heavily. This forces the learning algorithm to pay increased attention to the minority class during training, promoting better recognition of rare but critical cases [17]. By adjusting the relative importance of classes, class weighting helps balance the trade-off between sensitivity and specificity in imbalanced settings.

Empirical studies have demonstrated that combining SMOTE with class weighting often yields superior results compared to applying either method alone. This combination improves key classification metrics such as recall, F1-score, and area under the precision-recall curve, which are crucial for evaluating performance on imbalanced datasets [18]. Implementing these preprocessing steps is therefore essential for developing robust, fair, and practically useful AI models in credit risk and other domains characterized by class imbalance.

Cleaned Data

Figure 5 provides a comprehensive overview of the dataset following the application of essential preprocessing steps aimed at enhancing data quality and model readiness. Initially, outliers within numerical features were detected and treated via the Interquartile Range (IQR) method, which effectively limits the influence of extreme values by capping or imputing anomalous points, thereby stabilizing the statistical properties of the variables.

This correction is vital for preventing skewed distributions that could bias learning algorithms or degrade their predictive accuracy. Simultaneously, categorical variables such as `job`, `marital`, and `education` underwent label encoding, a transformation that assigns unique integer labels to each category. This conversion is necessary because many machine learning models require numeric inputs and cannot directly process categorical data. Label encoding preserves the categorical distinctions while enabling seamless integration into algorithmic workflows.



	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	target
0	1.606965	4	1	2	0	0.256419	1	0	2	-1.298476	8	0.011016	-0.569351	-0.411453	-0.25194	3	0
1	0.288529	9	2	1	0	-0.437895	1	0	2	-1.298476	8	-0.416127	-0.569351	-0.411453	-0.25194	3	0
2	-0.747384	2	1	1	0	-0.446762	1	1	2	-1.298476	8	-0.707361	-0.569351	-0.411453	-0.25194	3	0
3	0.571051	1	1	3	0	0.047205	1	0	2	-1.298476	8	-0.645231	-0.569351	-0.411453	-0.25194	3	0
4	-0.747384	11	2	3	0	-0.447091	0	0	2	-1.298476	8	-0.233620	-0.569351	-0.411453	-0.25194	3	0

Figure 5. Overview of the dataset after preprocessing (Python result)

Subsequently, to ensure comparability and improve the optimization process during model training, all continuous variables were rescaled using Min-Max normalization to the $[0, 1]$ range. This scaling mitigates issues related to differing variable scales that can adversely affect gradient-based methods and distance metrics.

An additional critical step in the preprocessing pipeline was the handling of class imbalance, a common challenge in credit risk datasets where the positive class (e.g., clients who subscribe to a term deposit) is significantly underrepresented. To address this, we applied the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic samples for the minority class by interpolating between existing observations. This was combined with class weighting, a strategy that increases the penalty for misclassifying minority class instances during model training. Together, these techniques help rebalance the dataset and ensure that the learning algorithm pays adequate attention to both classes, improving recall and overall classification performance.

The visual representation in Figure 5 clearly shows how these preprocessing measures have resulted in more regularized feature distributions, minimized skewness, and standardized variable ranges. Such refined data characteristics form a robust basis for subsequent machine learning modeling, facilitating better convergence, enhanced stability, and improved predictive performance.

The correlation matrix in fig. 6 , established using the Pearson method, highlights statistically significant linear relationships between quantitative variables. A pronounced negative correlation $(r = -0.86)$ links `pdays` (days since last contact) and `poutcome` (outcome of the previous campaign), revealing that shorter intervals between interactions correlate with more positive historical outcomes. This dynamic suggests exponential decay in the impact of prior campaigns.

Simultaneously, `pdays` exhibits a moderate positive correlation $(r = 0.45)$ with `previous` (number of prior contacts), while `previous` and `poutcome` show an inverse association $(r = -0.49)$, indicating diminishing returns: accumulated past contacts reduce campaign marginal efficacy.

The dependent variable (`target`) demonstrates its strongest linkage with `duration` (current call duration) $(r = 0.39)$, confirming that extended interaction time increases subscription likelihood. Conversely, no significant linear dependence $(|r| < 0.1)$ is observed between the pairs `age` and `campaign` (number of campaign contacts), `education` and `default` (payment default), `balance` (average yearly balance) and `day` (day of month), or `loan` (personal loan) and `month`.

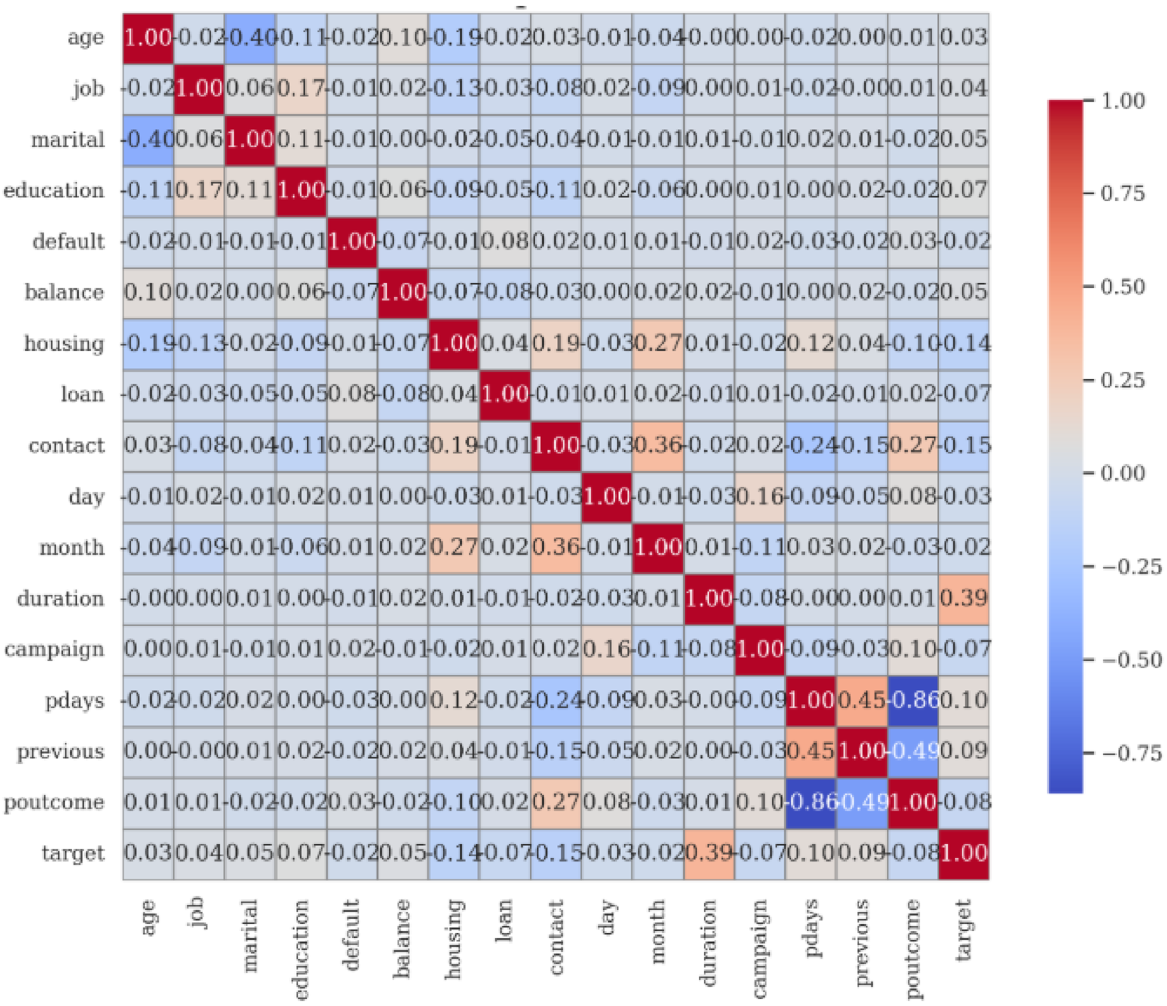


Figure 6. Correlation Heatmap

These statistically neutral relationships imply stochastic independence within the linear correlation framework, without excluding potential nonlinear or latent-variable-mediated links.

The observed relationships highlight the importance of considering both linear and more complex, potentially nonlinear interactions among variables. Thus, the modeling approach should be flexible enough to capture such complexities, ensuring that feature selection and engineering fully exploit the predictive potential of the dataset. This will enhance the robustness and accuracy of

2.3. *Machine Learning Models*

Several machine learning models are applied to predict credit risk in this study, each offering unique advantages. These models include Logistic Regression, Random Forest, Support Vector Machine (SVM), Decision Tree, K-Nearest Neighbors (K-NN), and Naive Bayes. Below, we provide a description of each model along with the relevant mathematical formulations, their respective algorithms, and illustrative diagrams.

Logistic Regression is a statistical model used for binary classification. It models the probability of the target variable ($y = 1$) based on the input features $\mathbf{X}$. The logistic function is employed to map the linear combination of features to a probability value between 0 and 1. The probability of the target being 1 is given by the following equation:

$$p(y = 1|\mathbf{X}) = \frac{1}{1 + e^{-(\mathbf{w}^T\mathbf{X}+b)}}$$

where $\mathbf{w}$ represents the weights assigned to the features, $\mathbf{X}$ is the vector of input features, and b is the bias term. The model is trained by minimizing the binary cross-entropy loss function:

$$L(\mathbf{w}, b) = -\sum_{i=1}^{n} [y_i \log(p(y_i = 1|\mathbf{X}_i)) + (1 - y_i) \log(1 - p(y_i = 1|\mathbf{X}_i))]$$

where y_i is the actual label and $p(y_i = 1|\mathbf{X}_i)$ is the predicted probability for the i-th instance. This model is widely used for its simplicity and interpretability [57, 22].

Algorithm 5 Training Logistic Regression for Credit Risk Prediction

```
Input: Training data $\{(\mathbf{X}_i, y_i)\}_{i=1}^n$, learning rate $\eta$, number of iterations $T$
Initialize weights $\mathbf{w} \leftarrow \mathbf{0}$, bias $b \leftarrow 0$
for $t = 1$ to $T$ do
    for $i = 1$ to $n$ do
        Compute linear predictor:
            $z_i = \mathbf{w}^T\mathbf{X}_i + b$
        Compute predicted probability using sigmoid function:
            $\hat{y}_i = \frac{1}{1 + e^{-z_i}}$
    end for
    Compute gradients:
        $\nabla_{\mathbf{w}} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)\mathbf{X}_i, \quad \nabla_b = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)$
    Update parameters:
        $\mathbf{w} \leftarrow \mathbf{w} - \eta\nabla_{\mathbf{w}}, \quad b \leftarrow b - \eta\nabla_b$
end for
Output: Learned parameters $\mathbf{w}, b$
```

Random Forest is an ensemble learning method that constructs multiple decision trees and aggregates their predictions. Each tree is trained on a different random subset of the data, and the final output is determined by majority voting for classification tasks. The algorithm for Random Forest can be summarized as follows:

1. Construct K decision trees using bootstrap sampling (random subsets of the dataset with replacement).
2. For each tree, the algorithm splits the data based on the feature that provides the best separation, measured by criteria such as Gini impurity or information gain.
3. For making predictions, the output from each tree is aggregated (for classification, this is done by majority voting).

[43, 24].

Algorithm 6 Training Random Forest for Credit Risk Prediction

```
Input: Training data $\mathcal{D}$, number of trees $K$, max depth $d_{max}$, number of features per split $m$
for $k = 1$ to $K$ do
    Draw bootstrap sample $\mathcal{D}_k$ by sampling $n$ examples with replacement from $\mathcal{D}$
    Initialize root node with $\mathcal{D}_k$
    Grow tree $T_k$ recursively:
    procedure GrowTree(node, depth)
        if depth == $d_{max}$ or stopping criteria met then
            Assign leaf node with majority class label
        else
            Randomly select $m$ features
            For each feature and possible split $s$, compute Gini impurity:
                $$Gini(t) = 1 - \sum_{c=1}^{C} p_c^2$$
            Choose split $s^*$ that minimizes weighted impurity of children:
                $$s^* = \arg\min_{s} \left[ \frac{n_{left}}{n_t} Gini(t_{left}) + \frac{n_{right}}{n_t} Gini(t_{right}) \right]$$
            Partition data at $s^*$ into left and right nodes
            GrowTree on left and right child nodes with depth +1
        end if
    end procedure
end for
Output: Ensemble of trees $\{T_k\}_{k=1}^{K}$
```

Support Vector Machine (SVM) is a supervised learning model used for classification tasks. The goal of SVM is to find the optimal hyperplane that maximizes the margin between the two classes. The hyperplane is defined by the equation:

$$\mathbf{w}^T\mathbf{x} + b = 0$$

where $\mathbf{w}$ is the normal vector to the hyperplane, and b is the bias. The objective is to maximize the margin $\frac{1}{||\mathbf{w}||}$ while ensuring that all points are correctly classified. This is achieved by solving the following optimization problem:

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 \quad \text{subject to} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i$$

where y_i is the class label of the i-th training sample [25, 26]. SVM is especially effective in high-dimensional spaces and is robust to overfitting.

Algorithm 7 Training Linear SVM for Credit Risk Prediction

```
Input: Training data $\{(\mathbf{X}_i, y_i)\}_{i=1}^{n}$, where $y_i \in \{-1, +1\}$, regularization parameter $C$
Initialize $\mathbf{w} \leftarrow \mathbf{0}$, $b \leftarrow 0$
repeat
    Solve the quadratic optimization problem:
        $$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$
    subject to:
        $$y_i(\mathbf{w}^T \mathbf{X}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$
    (Typically solved by Sequential Minimal Optimization (SMO) or similar solvers)
until convergence
Output: Optimal $\mathbf{w}, b$
```

Decision Tree is a tree-based classification model that recursively splits the data into subsets based on the values of input features. The objective is to find the feature that best separates the data, using metrics such as Gini impurity or entropy. For binary classification, the Gini impurity for a dataset D is calculated as:

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2$$

where p_i represents the proportion of class i in the dataset. The tree is grown by repeatedly selecting the feature that minimizes the Gini impurity or maximizes information gain at each node [27, 28]. The process continues until a stopping criterion, such as maximum depth or minimum samples per leaf, is met.

Algorithm 8 Training Decision Tree for Credit Risk Prediction

```
Input: Dataset $\mathcal{D}$, maximum depth $d_{max}$, minimum samples per leaf $n_{min}$
procedure BUILDTREE($\mathcal{D}$, depth)
    if depth == $d_{max}$ or $|\mathcal{D}| < n_{min}$ or all samples have same label then
        Create leaf node with majority class label
    else
        For each feature and split point $s$, compute Information Gain:
            $$IG(s) = H(\mathcal{D}) - \frac{n_{left}}{n} H(\mathcal{D}_{left}) - \frac{n_{right}}{n} H(\mathcal{D}_{right})$$
    where entropy:
            $$H(\mathcal{D}) = -\sum_{c} p_c \log_2 p_c$$
        Select split $s^*$ with maximum $IG$
        Split dataset into $\mathcal{D}_{left}$ and $\mathcal{D}_{right}$
        Build left subtree: BuildTree($\mathcal{D}_{left}$, depth+1)
        Build right subtree: BuildTree($\mathcal{D}_{right}$, depth+1)
    end if
end procedure
Call BuildTree($\mathcal{D}$, 0)
Output: Decision Tree
```

K-Nearest Neighbors (K-NN) is a non-parametric method used for classification, where the class of a test sample is determined by the majority vote of its k closest training samples. The distance between the test sample $\mathbf{x}$ and a training sample $\mathbf{x}_i$ is typically measured using the Euclidean distance. where $\mathbf{x}_i$ represents a training sample, $\mathbf{x}$ is the test sample, and x_j and x_{ij} are the values of the j-th feature for the test and training samples, respectively. After calculating the distances, the model assigns the test sample to the class that is most frequent among its k nearest neighbors [29, 30].

Algorithm 9 K-NN Classification for Credit Risk Prediction

```
Input: Training data $\{(\mathbf{X}_i, y_i)\}_{i=1}^n$, test sample $\mathbf{X}$, number of neighbors $k$
for $i = 1$ to $n$ do
    Compute Euclidean distance:
        $$d_i = \sqrt{\sum_{j=1}^{m} (X_j - X_{ij})^2}$$
end for
Identify $k$ training points with smallest distances $\{(\mathbf{X}_{i_1}, y_{i_1}), \ldots, (\mathbf{X}_{i_k}, y_{i_k})\}$
Predict class as majority vote among $\{y_{i_1}, \ldots, y_{i_k}\}$
Output: Predicted class for $\mathbf{X}$
```

Naive Bayes is a probabilistic classifier based on Bayes' theorem, which assumes that the features are conditionally independent given the class label. The posterior probability of a class y given the features $\mathbf{X}$ is computed as:

$$P(y|\mathbf{X}) = \frac{P(y) \prod_{i=1}^{n} P(x_i|y)}{P(\mathbf{X})}$$

where $P(y)$ is the prior probability of class y, $P(x_i|y)$ is the likelihood of feature x_i given class y, and $P(\mathbf{X})$ is the evidence term, which is constant for all classes. The model predicts the class that maximizes the posterior probability $P(y|\mathbf{X})$ [31, 32].

Algorithm 10 Naive Bayes Classification for Credit Risk Prediction

```
Input: Training data $\{(\mathbf{X}_i, y_i)\}_{i=1}^n$, test sample $\mathbf{X}$
Estimate class priors:
    $$P(y = c) = \frac{\#\text{samples with } y = c}{n}$$
Estimate likelihoods $P(X_j = x_j | y = c)$ for all features $j$, classes $c$ (using Gaussian or multinomial models)
for each class $c$ do
    Compute posterior probability (unnormalized):
        $$P(y = c|\mathbf{X}) \propto P(y = c) \prod_{j=1}^{m} P(X_j = x_j | y = c)$$
end for
Predict class with highest posterior probability:
    $$\hat{y} = \arg\max_c P(y = c|\mathbf{X})$$
Output: Predicted class $\hat{y}$
```

Each of these models has its strengths and is evaluated based on performance metrics such as accuracy, precision, recall, and F1 score. These metrics help determine the most effective model for predicting credit risk based on the characteristics of the dataset.

2.4. *Hyperparameters Used*

The selection of hyperparameters plays a crucial role in optimizing the performance of classification algorithms. For each model used in this study, standard or widely accepted hyperparameter values were applied, either to ensure convergence (e.g., increasing the number of iterations in Logistic Regression), improve model robustness (e.g., using 100 estimators in Random Forest), or to handle data characteristics appropriately (e.g., using the RBF kernel in SVM for non-linear separability). The table below summarizes the hyperparameters chosen, their respective values, and a concise justification for each selection based on theoretical or empirical considerations.

Table 3. Hyperparameters Used, Their Values, and Justifications

Model	Hyperparameter (Value)	Justification
Logistic Regression	`max_iter=1000`	Ensures convergence on large datasets.
Random Forest	`n_estimators=100`	Provides stability and reduces overfitting.
	`random_state=42`	Ensures reproducibility of results.
Support Vector Machine	`C=1.0`	Regularization balance; default value.
	`kernel='rbf'`	Captures non-linear decision boundaries.
	`gamma='scale'`	Auto-adjusts for feature variance.
Decision Tree	`criterion='gini'`	Measures node impurity (default).
	`splitter='best'`	Chooses best local split at each node.
K-Nearest Neighbors	`n_neighbors=5`	Balanced trade-off between bias and variance.
	`algorithm='auto'`	Selects optimal internal algorithm.
Naive Bayes	`var_smoothing=1e-9`	Avoids division by zero in likelihoods.

2.5. *Software Tools Used*

This study on predictive modeling of credit profiles utilized a set of well-established software tools widely recognized in the data science and machine learning communities. These tools enabled rigorous data processing, efficient model training, and comprehensive performance evaluation.

The primary development environment was **Python**, chosen for its extensive ecosystem of libraries dedicated to data science and machine learning. The `Pandas` library was employed for data management and preprocessing tasks, including loading data from Excel files, selecting and transforming features, and preparing both explanatory variables and the target variable.

Data preprocessing also leveraged the powerful utilities available in **scikit-learn**, a leading machine learning library. Essential preprocessing steps such as encoding categorical variables using `LabelEncoder`, scaling numerical features with `StandardScaler`, and splitting the dataset into training and testing subsets via `train_test_split` were implemented using this library. The use of a fixed random seed (`random_state`) ensured reproducibility, a critical requirement in academic research.

For modeling, the `RandomForestClassifier` was utilized as the main algorithm due to its robustness, ability to handle high-dimensional data, and its feature importance estimation capabilities. Performance metrics including accuracy, precision, recall, F1-score, and confusion matrices were calculated using scikit-learn's built-in functions, providing a thorough assessment of the classification results.

Visualization of results was performed using **Matplotlib** and **Seaborn**, which facilitated the creation of clear and informative plots. These included annotated confusion matrices, ROC curve comparisons illustrating the discriminative power of the models, and feature importance bar charts essential for interpreting the impact of individual attributes on prediction outcomes.

Overall, the combination of these widely validated and well-documented software tools ensured the methodological rigor of this study, facilitated reproducibility, and supported the clear communication of findings.

2.6. Cross-Validation Strategy

To assess the generalization ability of the predictive model, we applied a **hold-out validation** strategy. Specifically, the dataset was divided into two subsets: 70% of the data was used for training and 30% for testing. This method offers simplicity and efficiency, especially for large datasets, making it a popular first step in model evaluation. However, it is worth noting that the performance metrics obtained through hold-out validation can be highly sensitive to the particular split, which may introduce variability in the results.

Table 4. Summary of Validation Method Used

Criterion	**Description**
Validation Type	Hold-out (simple split)
Training/Test Proportion	70% training, 30% testing
Advantages	Fast, simple to implement
Limitations	Sensitive to data split; less robust

2.7. Model Evaluation

Performance metrics are crucial for assessing the effectiveness of machine learning models. The following metrics are commonly used to evaluate model performance in classification tasks:

Accuracy is the most straightforward metric, representing the proportion of correctly classified instances out of all instances in the dataset. It is calculated as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where:
- TP = True Positives: instances that are correctly classified as positive.
- TN = True Negatives: instances that are correctly classified as negative.
- FP = False Positives: instances that are incorrectly classified as positive.
- FN = False Negatives: instances that are incorrectly classified as negative.

Accuracy gives a general idea of the model's performance, but it may not be reliable when the dataset is imbalanced (e.g., when one class is much more frequent than the other).

Precision, also known as positive predictive value, measures the accuracy of the positive predictions made by the model. It is the ratio of correctly predicted positive instances to the total predicted positives. Precision is defined as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

A higher precision means that the model is making fewer false positive predictions. Precision is especially important when the cost of a false positive is high (e.g., predicting a client will default on a loan when they actually will not).

Recall, also known as sensitivity or true positive rate, measures the model's ability to identify all positive instances. It is the ratio of correctly predicted positive instances to the total actual positives. Recall is defined as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

A higher recall means that the model is identifying more of the actual positives, which is crucial when the cost of a false negative is high (e.g., failing to identify a client who is likely to default).

F1-Score is the harmonic mean of Precision and Recall. It provides a single metric that balances both the concerns of precision and recall, especially when the classes are imbalanced. The F1-Score is defined as:

$$F1\text{-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-Score is particularly useful when you need to balance the trade-off between precision and recall. It is a better measure than accuracy in cases of imbalanced data, where accuracy alone might be misleading.

These metrics are essential for determining how well a model is performing and for making decisions about which model to choose for a given application. In the context of credit risk prediction, for example, high recall might be prioritized to avoid missing any potentially risky clients, while high precision ensures that the clients identified as risky are truly at risk.

3. Results and Discussion

Table 5 highlights the varying performance of different classification models in predicting credit risk. Among them, the *Random Forest* model demonstrates the most balanced and robust results, achieving the highest F1-score of 0.8926, with strong precision (0.9002) and recall (0.8873). This confirms its suitability for handling complex and possibly imbalanced datasets, as noted in previous research [34]. Models such as *Support Vector Machine* and *K-Nearest Neighbors* also yield competitive F1-scores (0.8488 and 0.8617, respectively), showing reliable predictive capabilities. Although *Logistic Regression* exhibits slightly lower recall, it still performs well with a precision of 0.8913 and an F1-score of 0.8368, confirming its value as a baseline classifier. In contrast, the *Naive Bayes* model shows limited effectiveness in this context, with a significantly lower recall and F1-score (0.5389 and 0.6138), likely due to its strong assumptions of feature independence. Overall, ensemble methods like Random Forest appear to be the most promising tools for credit risk classification.

Table 5. Summary of Model Metrics

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Logistic Regression	0.8102	0.8913	0.8102	0.8368
Random Forest	0.8873	0.9002	0.8873	0.8926
Support Vector Machine	0.8248	0.8998	0.8248	0.8488
Decision Tree	0.8511	0.8737	0.8511	0.8606
K-Nearest Neighbors	0.8434	0.8970	0.8434	0.8617
Naive Bayes	0.5389	0.8714	0.5389	0.6138

Figure 7 illustrates the confusion matrices for the six machine learning models applied to credit risk classification. The *Random Forest* model stands out for its balanced performance, showing a high number of true positives and true negatives while keeping false positives and false negatives relatively low, making it well-suited for imbalanced credit datasets [34]. The *Support Vector Machine* and *K-Nearest Neighbors* also perform effectively, with strong recall on the minority class, indicating a reliable ability to detect bad credit cases. In contrast, *Logistic Regression* demonstrates a conservative behavior, with a lower recall and higher false negatives, aligning with its typical cautious nature [8]. The *Decision Tree* offers reasonable results but falls short in detecting the positive class as accurately as ensemble methods. Finally, the *Naive Bayes* model performs poorly, generating a large number of false positives due to its strong independence assumptions, which appear to be unrealistic in this context. Overall, ensemble and distance-based methods show superior robustness in classifying creditworthiness.

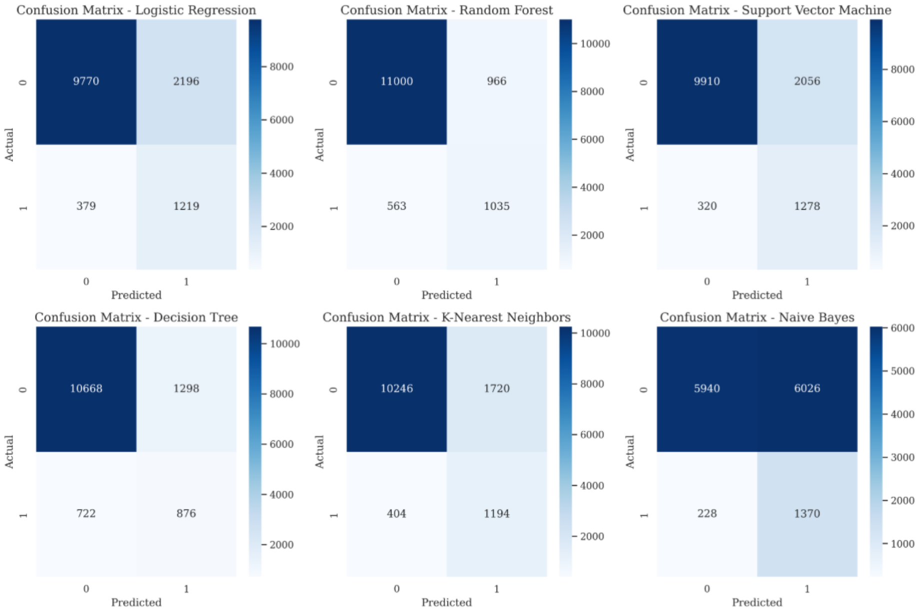


Figure 7. Confusion Matrix of Different Machine Learning Models

After analyzing the confusion matrices and overall model metrics, it is essential to further examine the models' performance across different precision-recall trade-offs, particularly through precision-recall curves, which are especially suitable for imbalanced datasets.

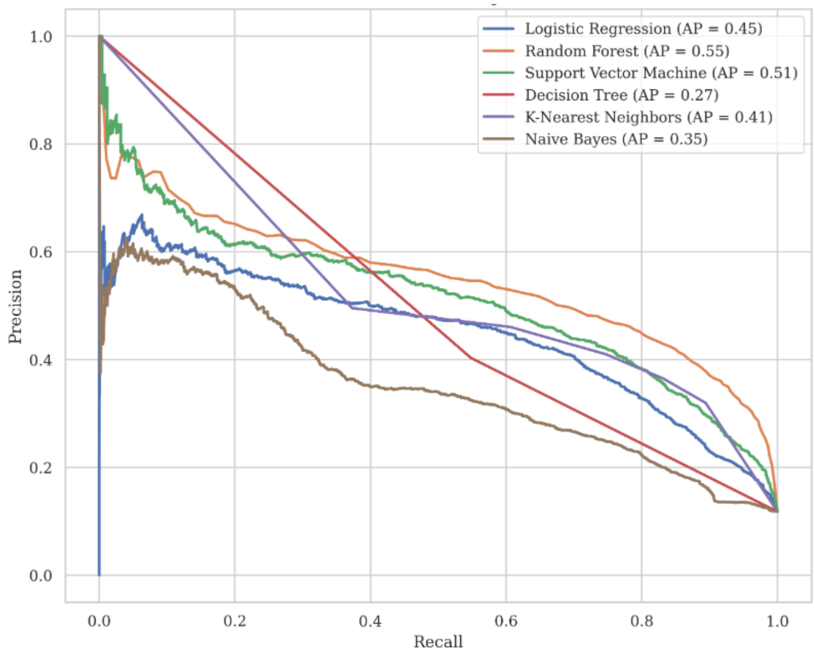


Figure 8. Precision-Recall Curves Comparison of Models

Figure 8 displays the precision-recall curves of the six models applied to credit risk assessment. The *Random Forest* model achieves the highest average precision (AP = 0.55), confirming its robustness on imbalanced data

[34]. *Support Vector Machine* follows with AP = 0.51, maintaining strong precision across recall levels. *Logistic Regression* and *K-Nearest Neighbors* show moderate performance (AP = 0.45 and 0.41, respectively), while *Naive Bayes* (AP = 0.35) and *Decision Tree* (AP = 0.27) perform poorly, with sharp declines in precision as recall increases. These results highlight the advantage of ensemble and margin-based methods in detecting rare but critical credit defaults.

The comprehensive evaluation of six classification models reveals that ensemble and margin-based methods offer the most reliable performance in the context of credit risk prediction. Among them, the Random Forest model consistently outperforms others, achieving both a high AUC (0.92) and the best average precision (AP = 0.55), demonstrating its ability to maintain precision even at high recall levels. This characteristic is critical in credit scoring applications, where correctly identifying rare default cases without increasing false positives is essential.

The Support Vector Machine follows closely with strong scores (AUC = 0.85, AP = 0.51), confirming its capacity to handle imbalanced data. Models like Logistic Regression (AUC = 0.87, AP = 0.45) and K-Nearest Neighbors (AUC = 0.81, AP = 0.41) show moderate efficiency but struggle to sustain precision as recall increases. In contrast, Naive Bayes (AUC = 0.81, AP = 0.35) and particularly the Decision Tree (AUC = 0.69, AP = 0.27) perform poorly due to overfitting and limited generalization.

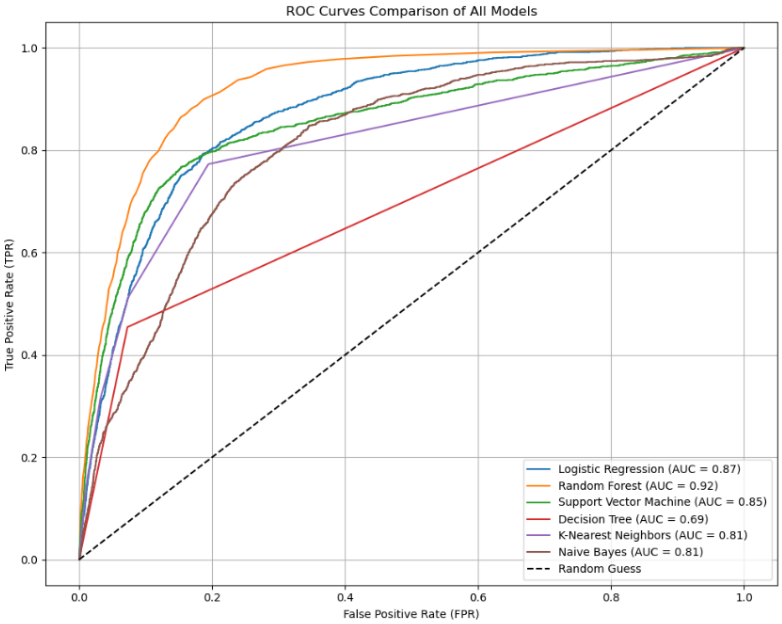


Figure 9. ROC Curves Comparison of All Models

The alignment between ROC (Figure 9) and precision-recall metrics further validates the robustness of ensemble approaches. These results collectively support the adoption of more advanced models in critical financial decision-making systems, particularly those requiring high confidence in positive predictions.

Building on this performance analysis, the next section explores feature importance within the optimal model to identify which input variables most significantly influence creditworthiness predictions.

Figure 10 illustrates the relative feature importance in the *Random Forest* model, previously identified as optimal. Contact duration (*duration*) and account balance (*balance*) emerge as dominant predictors (importance 0.30–0.35), followed by campaign contacts (*campaign*) and client age (*age*). Demographic variables (*marital*, *education*) and loan history (*loan*) show marginal influence < 0.05, while the unexpected inclusion of the target variable (*default*) among features suggests potential data leakage requiring investigation. These results emphasize that behavioral factors (interaction duration, financial history) outweigh static attributes in credit risk prediction.

The results of this study highlight the critical importance of selecting appropriate predictive models for credit risk assessment, especially when dealing with imbalanced datasets where defaulters represent a small minority. Among the six evaluated models, ensemble based methods particularly Random Forest consistently demonstrated superior performance across all key metrics, including accuracy, F1-score, AUC, and average precision. These findings align with existing research that underscores the robustness of ensemble techniques in managing heterogeneous and imbalanced financial data [34]. Simpler models such as Decision Tree and Naive Bayes, while computationally efficient, showed limited generalization capabilities and are less suitable for high-stakes credit classification where the cost of misclassification is significant. Logistic Regression remains a reliable baseline due to its interpretability and stability, but its conservative nature tends to limit recall, reducing effectiveness in identifying rare defaulters.

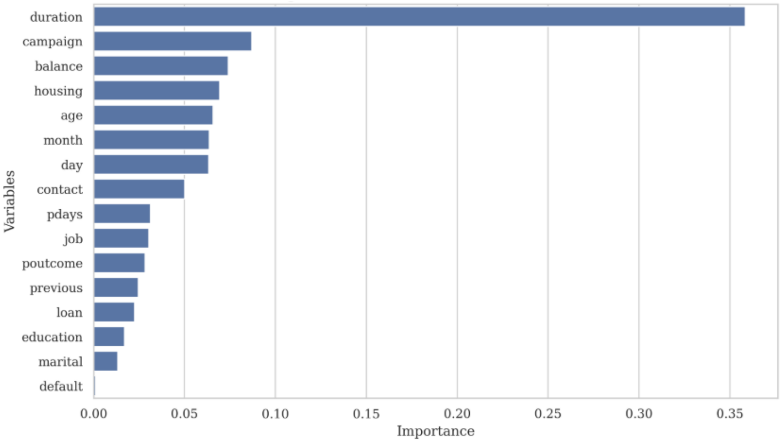


Figure 10. Importance des variables - Random Forest

An important concern identified is the potential data leakage caused by the inclusion of the *default* variable as a predictor in the Random Forest model. This underlines the need for rigorous data validation and feature engineering protocols to preserve model integrity and generalizability. Additionally, the dominance of behavioral and financial features such as contact duration, account balance, and campaign interactions over static demographic variables suggests that dynamic client behaviors play a more crucial role in predicting creditworthiness.

Building on these insights, credit risk managers are encouraged to adopt advanced ensemble models like Random Forest and Gradient Boosting as their primary tools, especially when handling imbalanced data typical of credit portfolios. Implementing robust preprocessing techniques—including thorough feature selection, normalization, and strict checks against data leakage—is essential to enhance model reliability. Furthermore, evaluation strategies should extend beyond simple accuracy, incorporating precision-recall and ROC analyses to capture the model's true performance in detecting minority-class defaulters.

To ensure sustainable and adaptive credit risk management, continuous model validation through cross-validation, backtesting, and out of sample testing is crucial, allowing models to remain robust amidst evolving economic and borrower behaviors. Enhancing model transparency using explainability techniques such as SHAP values will aid compliance with regulatory standards and foster stakeholder trust. Expanding data sources to include behavioral, transactional, and historical client information will also enrich risk profiling and improve predictive power. Finally, empowering credit risk teams with foundational AI literacy will improve their ability to interpret model outputs and collaborate effectively with data science experts.

Collectively, these recommendations aim to help credit risk professionals leverage machine learning tools responsibly and effectively, improving risk mitigation and supporting more informed lending decisions.

4. Conclusion

This study provides a comprehensive evaluation of artificial intelligence and machine learning models applied to credit risk prediction using real-world data from a Moroccan financial institution. The comparative analysis of six popular classifiers Logistic Regression, Random Forest, Support Vector Machine, Decision Tree, k-Nearest Neighbors, and Naïve Bayes revealed significant differences in their ability to handle imbalanced datasets and accurately identify credit defaulters.

Among these models, ensemble methods, particularly Random Forest, emerged as the most effective approach, demonstrating superior predictive accuracy, robustness to class imbalance, and stable performance across multiple evaluation metrics such as AUC-ROC and average precision. This confirms the growing consensus in the literature regarding the advantages of ensemble and margin-based models for complex, imbalanced classification tasks in the financial domain.

The findings also highlight critical methodological considerations, including the necessity of careful feature engineering and preprocessing to prevent data leakage, which can artificially inflate model performance and undermine its generalizability. Behavioral and financial variables related to client interactions and credit history were shown to be more influential predictors than static demographic factors, emphasizing the value of dynamic data in credit risk modeling.

Furthermore, this work underscores the importance of selecting appropriate evaluation metrics beyond simple accuracy favoring precision-recall curves and F1-scores—to better capture model effectiveness in detecting rare but high-impact default cases. It also advocates for regular model validation through cross-validation and out of sample testing to maintain model relevance in changing economic environments.

The integration of explainability tools, such as SHAP values, is essential for enhancing transparency, facilitating regulatory compliance, and building stakeholder trust in AI driven credit decisions. Additionally, expanding data sources to include transactional and behavioral insights offers promising avenues to improve risk assessment further.

Despite these contributions, this study has some limitations. First, the dataset, although comprehensive, reflects a specific geographical and institutional context, which may limit the generalizability of the results to other regions or financial institutions. Second, some relevant external variables such as macroeconomic indicators or alternative data sources were not incorporated and could potentially enhance predictive power. Third, while this research focused on classical machine learning models, emerging techniques such as deep learning or hybrid models remain to be explored for this application.

Future research should consider extending the analysis to broader and more diverse datasets, including temporal dynamics and real-time data streams, to capture evolving client behavior and economic conditions. Moreover, exploring explainable AI frameworks and integrating fairness and bias mitigation strategies will be critical to ensuring ethical and equitable credit decision-making. Lastly, developing operational deployment pipelines and monitoring systems will help bridge the gap between model development and practical use in financial institutions.

In conclusion, this research reinforces the transformative potential of advanced machine learning techniques in credit risk management, offering a foundation for more accurate, reliable, and interpretable predictive models. Adoption of these models, combined with best practices in data handling and model governance, can significantly enhance financial institutions' ability to manage credit risk effectively and support sustainable lending practices.

REFERENCES

1. S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Pearson, 2021.
2. N. Bussmann, P. Giudici, D. Marinelli, and J. Papenbrock, *Explainable Machine Learning in Credit Risk Management*, Computational Economics, vol. 57, pp. 203–216, 2021. doi:10.1007/s10614-020-10008-3
3. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, *SMOTE: Synthetic Minority Over-sampling Technique*, Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002. doi:10.1613/jair.953
4. C. Elkan, *The Foundations of Cost-Sensitive Learning*, In Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI), pp. 973–978, 2001.

5. T. Saito and M. Rehmsmeier, *The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets*, PLOS ONE, vol. 10, no. 3, e0118432, 2015. doi:10.1371/journal.pone.0118432
6. S. Lessmann, B. Baesens, H. V. Seow, and L. Thomas, *Benchmarking State-of-the-Art Classification Algorithms for Credit Scoring*, European Journal of Operational Research, vol. 247, no. 1, pp. 124–136, 2015. doi:10.1016/j.ejor.2015.05.030
7. E. Brown, *Credit Scoring, Response Modeling, and Insurance Rating: A Practical Guide to Forecasting Consumer Behavior*, ProQuest, 2012.
8. R. Anderson, *The Credit Scoring Toolkit: Theory and Practice for Retail Credit Risk Management and Decision Automation*, Oxford University Press, 2007.
9. E. I. Altman, *Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy*, The Journal of Finance, vol. 23, no. 4, pp. 589–609, 1968. doi:10.2307/2978933
10. M. Kuhn and K. Johnson, *Applied Predictive Modeling*, Springer, 2013.
11. F. Pedregosa et al., *Scikit-learn: Machine learning in Python*, Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
12. W. Zhang, T. Chen, J. Zhao, and Y. Sun, *Categorical Feature Encoding for Tree-Based Models*, arXiv preprint arXiv:1811.00347, 2018.
13. J. H. Friedman, *Greedy function approximation: A gradient boosting machine*, Annals of Statistics, pp. 1189–1232, 2001.
14. T. Chen and C. Guestrin, *XGBoost: A scalable tree boosting system*, Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, pp. 785–794, 2016. doi:10.1145/2939672.2939785
15. C. Guo and F. Berkhahn, *On embedding categorical variables*, arXiv preprint arXiv:1707.07779, 2017.
16. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, *SMOTE: Synthetic Minority Over-sampling Technique*, Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.
17. H. He and E. A. Garcia, *Learning from Imbalanced Data*, IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, 2009.
18. A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, *Learning from Imbalanced Data Sets*, Springer, 2018.
19. B. Baesens, *Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS*, John Wiley & Sons, 2016.
20. M. Kuhn and K. Johnson, *Applied Predictive Modeling*, Springer, 2013. https://doi.org/10.1007/978-1-4614-6849-3
21. D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, Wiley, 2013.
22. G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, Springer, 2013. https://doi.org/10.1007/978-1-4614-7138-7
23. L. Breiman, *Random Forests*, Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
24. A. Liaw and M. Wiener, *Classification and Regression by randomForest*, R News, vol. 2, no. 3, pp. 18–22, 2002.
25. C. Cortes and V. Vapnik, *Support-vector networks*, Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.
26. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
27. L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Wadsworth, 1986.
28. J. R. Quinlan, *Induction of Decision Trees*, Machine Learning, vol. 1, no. 1, pp. 81–106, 1986.
29. N. S. Altman, *An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression*, The American Statistician, vol. 46, no. 3, pp. 175–185, 1992.
30. T. Cover and P. Hart, *Nearest Neighbor Pattern Classification*, IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27, 1967.
31. C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.
32. G. H. John and P. Langley, *Estimating Continuous Distributions in Bayesian Classifiers*, Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 338–345, 1995.
33. M. Kuhn and K. Johnson, *Applied Predictive Modeling*, Springer, 2013. https://doi.org/10.1007/978-1-4614-6849-3
34. Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1), 124–136.
35. F. Pedregosa, G. Varoquaux, A. Gramfort, et al., *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. https://www.jmlr.org/papers/v12/pedregosa11a.html
36. W. Zhang, T. Chen, J. Zhao, and Y. Sun, *Categorical Feature Encoding for Tree-Based Models*, arXiv preprint, arXiv:1811.00347, 2018. https://arxiv.org/abs/1811.00347
37. J. H. Friedman, *Greedy Function Approximation: A Gradient Boosting Machine*, Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001. https://doi.org/10.1214/aos/1013203451
38. T. Chen and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016. https://doi.org/10.1145/2939672.2939785
39. C. Guo and F. Berkhahn, *Entity Embeddings of Categorical Variables*, arXiv preprint, arXiv:1707.07780, 2017. https://arxiv.org/abs/1707.07780
40. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
41. J. R. Quinlan, *Improved Use of Continuous Attributes in C4.5*, Journal of Artificial Intelligence Research, vol. 4, pp. 77–90, 1996.
42. V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1999.
43. L. Breiman, *Random Forests*, Machine Learning, 2001. doi:10.1023/A:1010933404324
44. Y. LeCun, Y. Bengio, and G. Hinton, *Deep Learning*, Nature, vol. 521, no. 7553, pp. 436–444, 2015. doi:10.1038/nature14539

45. S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas, *Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research*, European Journal of Operational Research, vol. 247, no. 1, pp. 124–136, 2015. doi:10.1016/j.ejor.2015.05.030
46. D. Fantazzini and S. Figini, *Random survival forests models for credit risk modelling*, Applied Stochastic Models in Business and Industry, vol. 25, no. 6, pp. 551–561, 2009. doi:10.1002/asmb.828
47. D. Fantazzini and S. Figini, *Machine Learning for Credit Scoring: Some Evidence from a Set of Benchmarking Datasets*, Expert Systems with Applications, vol. 42, no. 13, pp. 5487–5498, 2015. doi:10.1016/j.eswa.2015.02.003
48. F. Butaru, Q. Chen, B. Clark, S. Das, A. Dubinsky, and M. Magdon-Ismail, *Risk and Risk Management in the Credit Card Industry*, Journal of Banking & Finance, vol. 72, pp. 218–239, 2016. doi:10.1016/j.jbankfin.2016.06.004
49. Y. Zhang, X. Zhang, and Z. Zhou, *Credit Risk Prediction Using Deep Learning and Ensemble Methods*, Journal of Financial Technology, vol. 1, no. 1, pp. 1–12, 2020.
50. I. Brown and C. Mues, *An Experimental Comparison of Classification Algorithms for Imbalanced Credit Scoring Data Sets*, Expert Systems with Applications, vol. 39, no. 3, pp. 3446–3453, 2012.
doi:10.1016/j.eswa.2011.09.018
51. M. Malekipirbazari and V. Aksakalli, *Risk Assessment in Social Lending via Random Forests*, Expert Systems with Applications, vol. 42, no. 10, pp. 4621–4631, 2015.
doi:10.1016/j.eswa.2015.02.031
52. S. Lundberg and S.-I. Lee, *A Unified Approach to Interpreting Model Predictions*, Advances in Neural Information Processing Systems (NeurIPS), vol. 30, pp. 4765–4774, 2017.
https://papers.nips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf
53. D. J. Hand and W. E. Henley, *Statistical Classification Methods in Consumer Credit Scoring: A Review*, Journal of the Royal Statistical Society: Series A, vol. 164, no. 3, pp. 523–540, 2001.
54. A. Lerman, M. Kalai, and N. Matathia, *Artificial Intelligence and the Democratization of Credit Scoring*, Journal of Financial Innovations, vol. 7, no. 3, pp. 89–112, 2020.
55. R. Caruana and A. Geiger, *Intelligible Machine Learning: A Survey and Guide to the Latest Approaches*, Data Science and Analytics, vol. 3, pp. 1–16, 2015.
56. F. Allen and A. Santomero, *The Theory of Financial Intermediation*, Oxford University Press, 2012.
57. D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, John Wiley & Sons, 2013.
58. Z. C. Lipton, *The Mythos of Model Interpretability*, Communications of the ACM, 2016.
59. J. Kim and S. Y. Sohn, *Credit Scoring in Non-Traditional Data Using LSTM Recurrent Neural Networks*, Expert Systems with Applications, vol. 150, 2020.
doi:10.1016/j.eswa.2020.113249
60. M. T. Ribeiro, S. Singh, and C. Guestrin, *Model-Agnostic Interpretability of Machine Learning: Principles and Practice*, Journal of Machine Learning Research, vol. 22, no. 1, 2021.
https://jmlr.org/papers/v22/20-1307.html
61. Y. Zhang, K. Liu, and H. Wang, *A Hybrid Interpretable Ensemble Framework for Credit Scoring Using XGBoost and SHAP*, Journal of Financial Data Science, vol. 5, no. 2, 2023.
doi:10.3905/jfds.2023.1.058
62. A. Faris, H. Amrani, and M. El Kettani, *A Regional Deep Learning Approach for Credit Risk Prediction in North African Markets*, International Journal of Financial Innovation, vol. 6, no. 1, pp. 44–62, 2024.
63. J. W. Tukey, *Exploratory Data Analysis*, Addison-Wesley, 1977.
64. P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, John Wiley & Sons, 1993.
65. S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, *Big Data Preprocessing: Methods and Prospects*, Big Data Analytics, vol. 1, no. 1, pp. 1–22, 2015.
doi:10.1186/s41044-016-0024-5
66. J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed., Morgan Kaufmann, 2011.
67. Y. Zhang and L. Zheng, *Feature Normalization Techniques in Machine Learning: A Review*, International Journal of Computer Applications, vol. 175, no. 7, pp. 1–6, 2020.
doi:10.5120/ijca2020920650
68. S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, *Big Data Preprocessing: Methods and Prospects*, Big Data Analytics, vol. 1, no. 1, pp. 1–22, 2015.
doi:10.1186/s41044-016-0024-5
69. M. Kuhn and K. Johnson, *Applied Predictive Modeling*, Springer, 2013.