Journal of Autonomous Intelligence (2024) Volume 7 Issue 5
doi: 10.32629/jai.v7i5.1544

ORIGINAL RESEARCH ARTICLE

Enhancing banking governance: A machine learning-based credit risk classification

Karima Moumane[1,*], Ikram El Asri[2], Ilham Rharoubi[3], Hafida Ait Abderrahman[4], Sara Faqihi[5]

[1] *Software Project Management Research Team, ENSIAS, Mohammed V University, Rabat 10112, Morocco*

[2] *STRS Laboratory, SEEDS Team, INPT, Rabat 10112, Morocco*

[3] *LERSEM laboratory, FSJES CUAM, Ibn Zohr University, Agadir 80000, Morocco*

[4] *FSJES, Ibn Zohr University, Agadir 80000, Morocco*

[5] *ENSIAS, Mohammed V University, Rabat 10112, Morocco*

*** Corresponding author:** Karima Moumane, karima.moumane@ensias.um5.ac.ma

ABSTRACT

Risk management in the banking sector has gained heightened significance following the 2008 Global Financial Crisis. With the advent of Machine Learning (ML) techniques, financial institutions are increasingly turning to Artificial Intelligence (AI) for enhanced risk assessment and management. This paper introduces a systematic protocol for implementing a decision tree classifier tailored for credit risk classification. Additionally, we develop a user-friendly web application utilizing the Flask framework and Python Pickle library. This application offers customers an intuitive interface to input their attributes and receive predictions regarding their credit risk classification. Our empirical findings demonstrate that the Support Vector Machine (SVM) achieves a commendable accuracy of 77% in classifying customers based on their banking data. Furthermore, the web application proves to be an effective means for customers to interact with the ML model, enhancing accessibility and user engagement. These outcomes underscore the substantial benefits that ML techniques can bring to the banking industry, enabling improved risk detection and management while concurrently enhancing customer service delivery.

Keywords: banks; CNN architecture; credit risk; machine learning; Python

ARTICLE INFO

Received: 16 January 2024
Accepted: 6 February 2024
Available online: 22 March 2024

1. Introduction

After the Global Financial Crisis (GFC) of 2008, banks and financial institutions are invited to put in their strategies new tools that provide them with loan defaults and financial losses. Since this crisis, risk management in banks has gained more prominence and focused on how risks are detected, analyzed, corrected if possible, and managed. Researchers, policymakers, and bankers have focused on the development of risk management and how they can introduce data science (machine learning), one of the hottest technology fields in the world to date. Machine learning techniques (ML) are widely used in several areas, such as healthcare[1–3], supply chain[4,5], and banking[6,7]. In the DealBook newsletter of The New York Times, Royal Bank of Canada (RBC) ranked second overall among North American and European banks at incorporating and advancing artificial intelligence (AI) by Evident, a group of AI data scientists, researchers, and analysts working to strengthen the use of ML not only in detecting loan underwriting but also in analyzing customer behavior.

According to the report of McKinsey & Co[8], risk functions in banks will take a different era from what they are today, and no one can predict all forthcoming disruptions, macroeconomic shocks, geopolitical world changes, or banking scandals. The trend is not to set what will be required of the risk functions of the future but to be prepared for the coming changes by the establishment of a ML protocol to improve the accuracy of risk models by identifying complex nonlinear patterns, and every piece of information can be used to increase the detection of risks. Banks also applied to keep their customers satisfied by upping their game, access to services at any time, and taking strong advocacy of corporate values and principles supported by a robust risk culture reinforced by using a model risk to create efficiency and improve customer experience.

This study not only provides an overview of AI and ML for automating risk assessment and reducing credit risk but also focuses on their application in Moroccan banks to detect patterns and anomalies indicating fraudulent activity. Additionally, we aim to address a critical question faced by bank administrators: How can we determine whether our customer is eligible for a loan or not? To answer this question and establish a meaningful, independent, and operational system tailored to our industry, easily usable by bank employees, we proceed through three main phases: first, founding an open dataset for a Moroccan bank; secondly, experimenting with different machine learning models; and third, deploying our best model into a web application for user convenience.

The remainder of the paper is structured as follows: Section 2 presents related work, Section 3 discusses the proposed architecture, Section 4 presents and discusses classification results as well as the web application implementation, including the considered points for choosing to deploy our model in a web application. Finally, Section 5 concludes the paper, and Section 6 provides insights into future work.

2. Related work

2.1. Risk management at banks

The financial system is made up of many different entities, with assets being transferred through various channels. Banks and financial markets such as equity markets act as intermediaries, connecting lenders and borrowers. Moreover, commercial banks play a crucial role in the performance of the financial system as intermediaries in the flow of funds.

Furthermore, banks are vulnerable to various types of risks that can result in financial instability if not properly managed. The risks associated with banking operations have the potential to lead to their failure. Indeed, the field of risk management in banking has greatly advanced over recent decades; however, there remains a growing need for further improvement and development[9].

The field of banking risk management has evolved to address a wide range of risks, both old and new. Despite the prevalence of traditional risks such as credit, liquidity, and market risks, newer risks such as information security have become increasingly significant. Furthermore, the forms and characteristics of traditional risks have been altered and have become intertwined with newer risks[9].

The financial system depends on a variety of organizations and channels to transfer assets, with banks and equities markets acting as intermediaries to connect lenders and borrowers. Even though the field of banking risk management is always changing, with new threats like information security coexisting with more established risks like credit and liquidity, there is always a need for improvement and development in order to effectively manage the complex interactions between these risks.

2.2. Credit risks

Credit risk refers to the financial loss that results from a counterparty’s inability to meet their

contractual obligations, such as making timely payments of interest or principal, or from the increased likelihood of default over the course of the transaction. According to Milojević and Redzepagic [9], credit risk has remained the predominant risk in the banking industry over the past few decades and has seen the most adoption of AI and ML.

The use of ML techniques in credit risk modeling has been the subject of extensive academic research since the early 2000s[10]. The purpose of these models is to predict the potential financial loss for a credit institution, such as a bank or peer-to-peer lender, in the event of a borrower defaulting on a loan. The key aspect of a credit risk model is the default probability, which is typically estimated using statistical methods and credit scoring models. Moreover, the complexity of evaluating credit risk has led to the adoption of ML, particularly in the expanding credit default swap market, where determining both the probability of a default event and the cost of default in case it occurs involves many uncertain elements[11]. Moreover, Bussmann, Giudici and Marinelli[12] Suggests using a novel multistage deep belief network-based extreme ML approach for credit risk assessment. Additionally, their research indicates that network-based explainable AI models can improve our understanding of the factors that contribute to financial risks, including credit risks, by extracting nonlinear relationships from financial information found in balance sheets.

Statistical techniques that can objectively evaluate and analyze credit risk are of great importance because they help lending organizations and especially banks avoid significant losses that may occur due to borrower default.

2.3. Machine learning and its applications

In the 1980s, ML emerged as a subfield of AI that uses statistical techniques to enable computers to learn and improve from experience[13]. AI and ML can aid in overcoming current global economic and financial difficulties, including those brought on by the COVID-19 pandemic. The implementation of these tools, central components of AI, is revolutionizing the way financial risk is managed and evaluated.

Banks and other financial institutions are increasingly using ML tools for prediction and classification purposes. According to Alonso and Carbo[13], recent surveys show that these institutions are adopting an increasing number of ML techniques in various areas of credit risk management, such as regulatory capital, provisions, credit scoring, and monitoring. Several studies have investigated the application of stress testing[14] in credit risk management[14,15].

ML can find significant patterns in data and has become a popular tool for tasks that require extracting meaningful information from large datasets. The complexity of patterns in these datasets makes it difficult for programmers to specify the exact process for extraction. ML solves this problem by providing programs with the ability to learn and adapt, overcoming the limitations of explicit and detailed programming specifications[16].

According to Aziz and Dowling[11], AI-powered solutions are providing new opportunities to make lending decisions, warn traders of potential risks in their positions, detect fraud from both customers and insiders, improve compliance, and reduce the risk of model errors.

Surveys reveal that financial institutions are incorporating ML techniques more frequently in order to improve categorization and prediction in a variety of areas related to credit risk management, including regulatory capital, provisions, credit scoring, and monitoring. This indicates the wide range of applications for ML tools. Furthermore, the ability of ML to identify complex patterns in large datasets is emphasized. This ability overcomes the difficulties associated with explicit programming specifications and provides flexible applications in a variety of activities, from risk assessment and fraud prevention to lending decisions.

2.4. ML to create a good system of banking governance

The idea of creating better ways to control conflicts of interest among corporate stakeholders is not novel, but it became more important after a large financial crisis. This is especially true for the banking sector. The place occupied by banks in the economy as well as the type of work they do, coupled with the substantial costs that can result from issues with their governance, make it crucial to examine the governance mechanisms in the banking sector[17]. Furthermore, banks have particular governance challenges that make their governance structures different from those of nonfinancial organizations[18].

There is a growing trend among financial services organizations to adopt AI and ML technologies to improve their operations and stay ahead of the curve. Currently, central banks are facing a variety of novel and exceptional difficulties, including the advent of distributed ledger technology, advancements in data analytics such as AI and ML, the prevalence of cloud computing, and the expansion of mobile access and improved internet speed and connectivity[19]. Furthermore, Milojević and Redzepagic[9] explain the benefits of AI and ML applications in each credit risk management segment. These major segments are presented in the following figure (**Figure 1**).

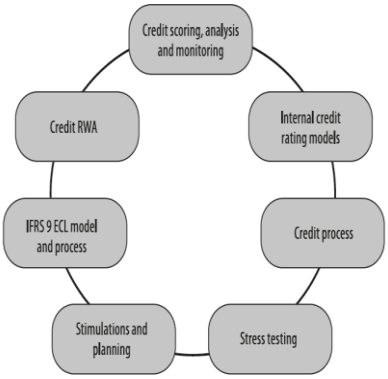

Figure 1. Major segments of the credit risk management AI and ML implementation[9].

However, banks, while implementing ML, face challenges and unresolved issues related to the risk associated with models, such as “black box” problems, access to and protection of data, clarity, ethics, and the availability of qualified staff to develop and apply new techniques[9].

The potential benefits of these technologies are numerous, ranging from improved credit underwriting and compliance processes to better customer interactions and risk management. This trend has been recognized by organizations such as the Institute of International Finance (IIF), which has written about the potential of AI and ML as “RegTech” in the banking industry and as key components of FinTech’s new business models[10]. Risk managers are optimistic about the potential of AI and ML to drive further advancements in the field of banking risk management[9].

According to Leo, Sharma and Maddulety[16], ML, a key technology with significant implications for risk management, can create more precise risk models by recognizing intricate, nonlinear patterns within extensive datasets, and its forecasting capability improves with each additional piece of information, leading to enhanced predictive power over time.

This work focuses on the implementation of a decision tree classifier designed exclusively for credit risk classification, acknowledging the urgent need for sophisticated approaches. The decision to examine credit risk in depth is supported by the fact that it is crucial to banking operations and that making mistakes in this area can have far-reaching effects.

3. General architecture

In this section, we describe the general design of our proposed architecture for credit risk classification. The proposed system's architecture (depicted in **Figure 2**) involves the following components:

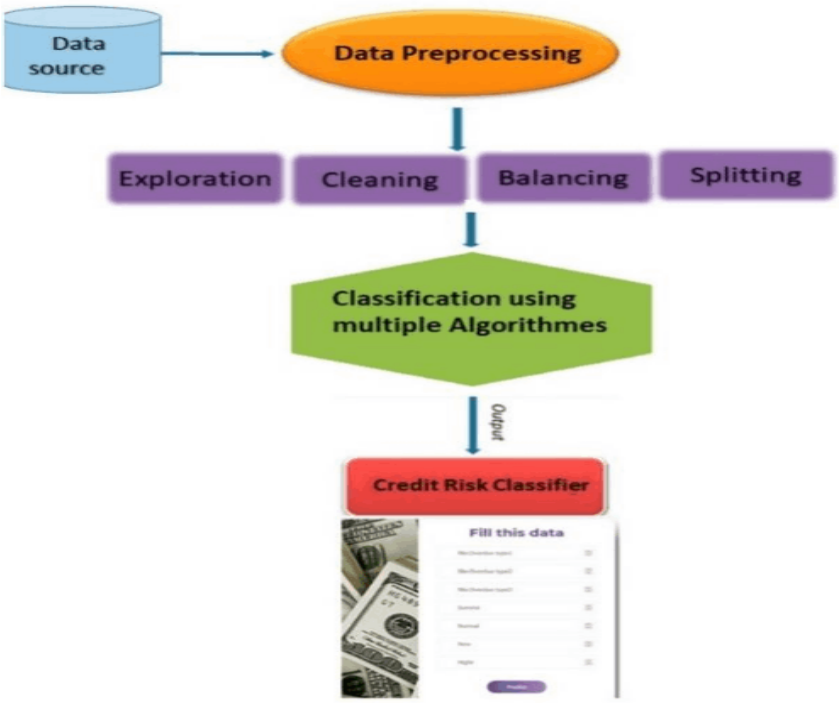

Figure 2. The system architecture.

- Data Preprocessing: Before the data can be analyzed, they need to be preprocessed to remove any missing values, outliers, or inconsistencies. This step is important to ensure that the data are accurate and suitable for analysis.
- Classification: The preprocessed data are then fed into multiple classification algorithm to be trained by learning the relationships between the features (independent variables) and the target class (Risky or Worthy).
- Performance evaluation: the process of evaluating the performance and effectiveness of the used ML model, typically in terms of accuracy.
- Credit Risk Classifier Web Application: To use the trained model to make predictions on new, unseen data by passing the features to the predicted function of the model, a web application has been built to provide an easy and user-friendly interface for users to interact with the model and make predictions.

3.1. Data acquisition and preparation

This subsection presents the data preparation process for the main dataset, which consists of data acquisition and data preprocessing.

Data acquisition:

Obtaining data:

The dataset used for this project contains 981 entries for customer transactions[20] and demographic-related data, with 11 independent variables and one target variable as presented in **Table 1**.

Table 1. Customer data attributes.

Attribute	Description	Type
Loan_ID	Unique Loan ID	Qualitative
Gender	Male/Female	Qualitative
Married	Applicant married (Y/N)	Qualitative
Dependents	Number of dependents	Qualitative
Education	Applicant Education (Graduate/ Under Graduate)	Qualitative
Self_Employed	Self-employed (Y/N)	Qualitative
ApplicantIncome	Applicant income	Quantitative
CoapplicantIncome	Co-applicant income	Quantitative
LoanAmount	Loan amounts in thousands of dollars	Quantitative
Loan_Amount_Term	Term of the loan in months	Quantitative
Credit_History	credit history meets guidelines yes or no	Quantitative
Property_Area	Urban/ Semi Urban/Rural	Qualitative
Loan_Status	Loan approved (Y/N), this is the target variable	Qualitative

In this dataset, each entry represents an individual applying for credit at a bank. Each person is categorized based on their creditworthiness, which is assessed using a set of attributes.

Some individuals are classified as ‘good risks’, indicating that they are considered reliable and likely to repay their credit obligations.

Conversely, ‘bad risks’ are individuals who may pose a higher credit risk due to factors such as a history of late payments or other unfavorable attributes. This classification helps the bank in evaluating and managing its lending decisions.

3.2. Data exploration

In this subsection, we present the data exploration process and results through various types of representations (tables, doughnut charts, histograms, etc.), aiming to visualize the distribution of our dataset instances. Additionally, by referring to the Knowledge Discovery in Databases (KDD) process for building a machine learning model, approximately 80% of the total process is dedicated to data preprocessing, making it the longest phase. We delve into the results of this phase, crucial for observation and consideration as the state-of-the-art approach in processing our dataset with a machine learning algorithm. This exploration is particularly significant for choosing an appropriate model and configuration.

Data exploration is the process of analyzing and summarizing a dataset to understand the underlying patterns and existing relationships within the data. This can involve a variety of techniques, such as visualizing the data, calculating summary statistics, and identifying outliers or anomalies. Data exploration aims to gain a better understanding of the data for further and extensive analysis or modeling.

For each numerical column, the summary statistics are given in **Table 2.** The mean income of the applicants is approximately $5403, with a minimum of $150 and a maximum of $81,000. The mean co-applicant income is approximately $1621, with a minimum of $0 and a maximum of $41,667.

Also, the mean loan amount applied for is approximately $146, with a minimum of $9 and a maximum of $700. Finally, the mean loan term is approximately 342 months, with a minimum of 12 months and a maximum of 480 months.

Table 2. Dataset profile—quantitative attributes.

	Applicant Income	Co-applicant Income	Loan Amount	Loan_Amount_Term	Credit_History
mean	5403.459283	1621.245798	146.412162	342	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150	0	9	12	0
25%	2877.5	0	100	360	1
50%	3812.5	1188.5	128	360	1
75%	5795	2297.25	168	360	1
max	81000	41667	700	480	1

On the other hand, **Figure 3** shows multiple pie plots of categorical variables. Based on these plots, 80% of applicants in the dataset are male, around 65% of the applicants in the dataset are married, about 15% of applicants in the dataset are self-employed and about 85% of applicants have repaid their debts.

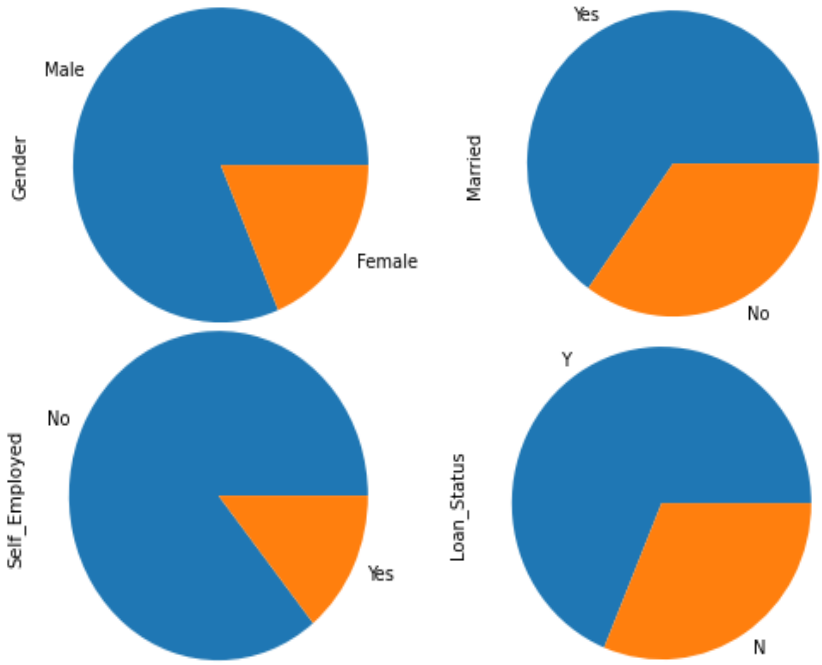

Figure 3. Data exploration visualization—qualitative attributes.

Checking for missing data:

Handling missing data has been the most common problem faced during the data cleaning and exploratory analysis of this project. Missing data could be a missing sequence, incomplete function, missing files, incomplete information, data entry error, etc.[21,22].

By visualizing missing values, you can quickly identify the columns with missing data, which can help you decide how to handle missing values in our data preprocessing. **Figure 4** shows a heat map representing the presence of missing values across all columns.

The missing map confirms that we have missing values present in our data set and need to be imputed with the appropriate values for further analysis. While doing imputation we will fill Nonapplicable (NA) values present in categorical variables with their modular class, and NA values present in numeric variables will be filled with their median or mean values according to the situation.

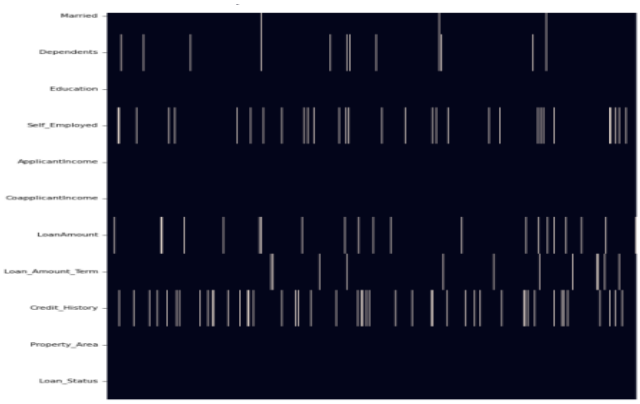

Figure 4. Missing data exploration.

- Gender: "Male" is a modular class in the Gender variable, so we are going to fill NA values present in this categorical variable with the "Male" value.
- Marital status: "Yes" is a modular class in the Marital status variable, so we are going to fill NA values present in this categorical variable with "Yes".
- Dependents: "0" is a modular class in the Dependents variables, so we are going to fill NA values present in this categorical variable with "0".
- Self_Employed: "No" is a modular class in the Self_Employed variable, so we are going to fill NA values present in this categorical variable with "No".
- LoanAmount: missing values in the 'LoanAmount' column will be imputed with the median value "128.0".
- Loan_Amount_Term: missing values in the 'LoanAmount' column will be imputed with the median value "360".
- Credit_History: "1" is a modular class in the Credit_History variable, so we are going to fill NA values present in this categorical variable with "1".

3.3. Data balancing

Checking data imbalance:

Data imbalance usually reflects an unequal distribution of classes within a dataset[23]. For this project, we are interested in specifying data imbalance between the two classes: high credit risk (label = N) and low credit risk (label = Y) customers.

We can observe that:

- Low credit risk Class: 68.7% of the dataset.
- High credit risk Class: 31.3% of the dataset.

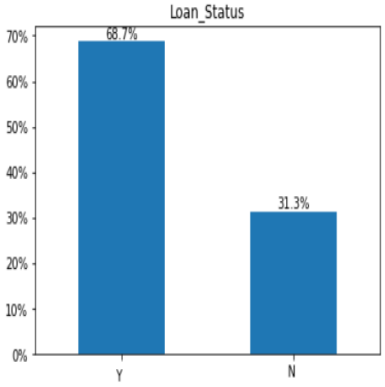

Figure 5. Class distribution plot.

Based on the visualized histogram values of class distribution in **Figure 5**, we can easily observe that the rate of the low credit risk class (class Y) is approximately double the rate of the high credit risk class (class N), given that 68.7/31.3 = 2.18. This indicates a significant imbalance between classes. Therefore, if we train a classification model without addressing this issue, the model will be inherently biased and will not serve as an effective heuristic for us.

Handling the balance of classes:

Under sampling:

To balance the classes, we opted for the use of under sampling as a technique used to balance imbalanced datasets in ML[24,25] and data science for the following reasons:

- The dataset is large: Under sampling, it can be useful in cases where the dataset is large and the minority class (high credit risk class) has a relatively small number of observations. By removing some observations from the majority class (Low credit risk class), the dataset can be balanced without losing too much information.
- The cost of misclassifying the minority class is high: it is important to correctly classify the minority class even if it means misclassifying the majority class. Under sampling can be used to balance the dataset and increase the model's performance on the minority class.
- The model is sensitive to class imbalance: in fact, classification models are sensitive to class imbalance. Under sampling can be used to balance the dataset and increase the model's performance.

Before trying oversampling techniques, we should first split our dataset into test and training sets and then proceed with normalization.

Splitting data:

Dividing the data into training and test sets before applying under-sampling is a standard practice in machine learning and data science, especially when dealing with imbalanced datasets[26]. The primary objective is to prevent the model from overfitting to the training data and to ensure its ability to generalize effectively to new, unseen data. Additionally, the data split into training and test sets allows for an assessment of whether the chosen under-sampling technique functions as intended and whether it introduces excessive bias. By comparing the model's performance on the test set with and without under-sampling, one can discern the impact of under-sampling on the model's overall performance[27].

Normalization (Scaling):

Normalization is a technique used to adjust the range of numeric features to a common scale; it helps to ensure that each feature has an equal impact on the model, and it is necessary since the features have very different ranges and units of measurement.

As a normalization method, we used Standardization, which is a method that scales the data by subtracting the mean value and dividing it by the standard deviation. This method standardizes the data so that it has a mean of 0 and a standard deviation of 1. The standardization method is particularly useful since the data are not normally distributed, and it is used in classification algorithms because they assume that the data are normally distributed.

Data Balancing with Under Sampling:

After splitting and normalizing the data, it is time to use the simple random under sampling technique. We take a sample from the majority class to match the size of the minority class, as described in the previous part.

4. Data classification and implementation

Data classification using ML refers to the use of algorithms and models to automatically categorize data into predefined classes. ML algorithms learn patterns and relationships within the data and use this information to make predictions about the class or category to which new data belong.

The goal of our case study is to classify loan applicants or existing borrowers into different risk categories based on their creditworthiness. The purpose of credit risk classification is to determine the likelihood that a borrower will default on their loan so that lenders can make informed decisions about the risk they are willing to take when granting a loan.

4.1. Classification algorithms

In this subsection of our paper, we expound upon the methodologies employed to select the optimal model.

4.1.1. Machine learning approach

There are numerous classification algorithms available for various types of classification tasks. Here are some of the most commonly used classification algorithms:

Decision tree: is a type of supervised learning algorithm that is mostly used in the field of data mining, ML, and AI[28]. The algorithm creates a tree-like model of decisions and their possible consequences. Each internal node of the tree represents a "test" on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The topmost node in the decision tree is known as the root node. It tests an attribute and splits the dataset into subsets. The subsets are then used as input for the subsequent nodes[29].

Once the decision tree is built, it can be used for prediction by traversing the tree from root to leaf. The algorithm starts at the root node, testing the attribute specified by the node and following the corresponding branch based on the outcome of the test. The process is repeated until a leaf node is reached, at which point the class label of the leaf node is returned as the prediction[30].

Random Forest: is an ensemble learning method used for both classification and regression tasks. It is a combination of multiple Decision Trees, where each tree is built using a random subset of features and data points. The algorithm aims to reduce overfitting and improve generalization by aggregating the predictions of individual trees[31]. During the prediction phase, the final output is determined by taking a majority vote (for classification) or an average (for regression) of the outputs of individual trees. Random Forests are known for their robustness, ability to handle large datasets with high-dimensional features, and resistance to overfitting. They often provide high accuracy and are less sensitive to noise and outliers compared to single Decision Trees.

AdaBoost (Adaptive Boosting): is another ensemble learning method primarily used for binary classification tasks. It builds multiple weak learners (typically Decision Trees with few levels) sequentially, where each subsequent learner focuses on misclassified instances from the previous ones[32]. In each iteration, AdaBoost assigns higher weights to misclassified instances, making them more important for the next iteration. The final prediction is obtained by weighted voting of individual weak learners. AdaBoost excels at improving the performance of weak models and has high accuracy. However, it may be sensitive to noisy data and outliers, and in some cases, it can be prone to overfitting.

Support Vector Machine (SVM): is a powerful supervised learning algorithm used for both binary and multiclass classification. It aims to find the optimal hyperplane that best separates the data points belonging to different classes[33]. The hyperplane is chosen to maximize the margin between the two classes, which

helps improve generalization. SVM can handle high-dimensional data and is effective in situations where the classes are not linearly separable by using the kernel trick to transform the data into a higher-dimensional space. SVM can achieve high accuracy and works well with small to medium-sized datasets. However, SVM's performance may be affected by the choice of the kernel and the regularization parameter, and it can be computationally expensive for large datasets.

Nearest Neighbors (KNN): is a simple and intuitive classification algorithm used for both binary and multiclass classification tasks. It is a non-parametric and lazy learning algorithm, meaning it doesn't learn an explicit model during training. Instead, it stores all the training instances and makes predictions based on the majority class among its k nearest neighbors in the feature space[33]. The value of k is a hyper parameter that needs to be chosen, and larger values of k result in smoother decision boundaries, while smaller values of k may lead to more flexible decision boundaries. The KNN is easy to implement and understand, and it can handle both numerical and categorical features. However, it can be sensitive to irrelevant features and may suffer from the curse of dimensionality when dealing with high-dimensional data.

4.1.2. Deep learning approach

Convolutional Neural Network (CNN): is a specialized deep learning model designed primarily for processing and analyzing visual data. Its architecture is particularly powerful in tasks such as image recognition, classification, and object detection. CNNs are characterized by a unique design that includes convolutional layers, pooling layers, and fully connected layers. This design allows the network to automatically learn hierarchical representations from input data. We can modify the input layer of the CNN to accommodate the structure of the used data, for example we can adapt a CNN architecture to learn from a Comma-Separated Values (CSV) dataset. Even with a CSV dataset, it is essential to use cross-validation to assess the model's performance robustly. Stratified K-fold cross-validation is a suitable technique to maintain class distribution integrity.

Each of these algorithms has its strengths and weaknesses. It is often a good idea to try multiple algorithms and compare their performance to select the most suitable one for the classification task at hand.

4.2. Performance evaluation

Performance evaluation is a critical step in the ML process that helps assess the effectiveness of a classification algorithm or model. It allows us to measure how well the model is performing and how it generalizes to unseen data. Several evaluation metrics can be used depending on the nature of the classification problem (binary or multiclass) and the specific requirements of the application. Since we are dealing with a binary classification problem where there are two classes ("Yes" and "No"), we used the following performance evaluation metrics:

- Accuracy: Accuracy measures the overall correctness of the model's predictions and is defined as the ratio of correctly classified instances to the total number of instances in the dataset. While accuracy is a straightforward and intuitive metric, its practical significance can be limited, especially in the context of imbalanced datasets. In situations where the classes are not evenly distributed, accuracy may not provide a comprehensive understanding of the model's performance.
- Precision: Precision represents the ability of the model to correctly predict positive instances among all the instances it predicted as positive. It is calculated as the ratio of true positives to the sum of true positives and false positives. Precision is particularly important in scenarios where the cost of false positives is high. For instance, in medical diagnoses, a high precision ensures that the positive predictions made by the model are more likely to be true, reducing the chances of unnecessary interventions or treatments.
- Recall (Sensitivity or True Positive Rate): Recall measures the ability of the model to correctly identify

positive instances among all the actual positive instances in the dataset. It is calculated as the ratio of true positives to the sum of true positives and false negatives. Recall becomes crucial when the cost of false negatives is high. In applications like fraud detection or medical screenings, a high recall ensures that the model is effective in capturing all relevant positive instances, minimizing the chances of missing important cases.

- F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is especially useful when dealing with imbalanced datasets. The F1 score is a comprehensive metric that considers both false positives and false negatives. It becomes particularly valuable in situations where achieving a balance between precision and recall is important. This is common in tasks where neither false positives nor false negatives can be tolerated excessively, making F1 score a suitable metric for assessing model performance.

In summary, each performance metric serves a specific purpose, and their practical significance depends on the nature of the problem at hand. Choosing the right metric or combination of metrics is essential for a nuanced evaluation of a machine learning model's effectiveness in different real-world scenarios.

4.3. Classification results

4.3.1. Machine learning approach

Table 3 reports performance evaluation metrics for each algorithm in our binary classification task.

Table 3. Performance evaluation metrics.

Algorithms	Class	Accuracy	Precision	Recall	F1-score
Decision Tree	Y	0.65	0.53	0.4	0.45
	N		0.7	0.79	0.74
Random Forest	Y	0.68	0.58	0.4	0.47
	N		0.71	0.84	0.77
Ada Boost	Y	0.76	0.93	0.37	0.53
	N		0.73	0.98	0.84
SVM	Y	0.77	1	0.37	0.54
	N		0.73	1	0.58
KNN	Y	0.74	0.78	0.4	0.53
	N		0.73	0.93	0.82

Based on these evaluation metrics, we can observe differences in performance among the algorithms for the binary classification task.

The Decision Tree model shows moderate accuracy, correctly classifying 65% of instances. However, there is room for improvement in both Precision and Recall for both classes. The F1 scores for both classes are relatively low, indicating that the model struggles to strike a balance between correctly identifying risky and not risky credit cases.

Random Forest performs slightly better than the Decision Tree, with a higher accuracy of 68%. It shows better precision, recall, and F1-scores for class N (not risky), indicating that it is better at correctly identifying non-risky credit cases. However, there is still room for improvement in classifying risky credits (class Y).

AdaBoost achieves higher accuracy (76%) than both Decision Tree and Random Forest. It shows impressive precision for class Y (risky credit) at 93%, indicating it has a high ability to correctly identify risky cases. However, the low Recall for class Y (37%) suggests that the model is missing some risky credit

cases, which can be improved.

SVM performs well in terms of Accuracy (77%) and shows perfect Precision for class Y (risky credit). However, like AdaBoost, the Recall for class Y is relatively low (37%), indicating that it misses some risky credit cases. SVM has excellent precision and Recall for class N (not risky), making it robust in identifying non-risky credit cases.

KNN performs well with an accuracy of (74%). It shows balanced precision, recall, and F1 scores for both classes. However, the Recall for class Y (40%) indicates that the model may miss some risky credit cases, which could be further improved.

Considering that the goal is to classify credit as risky or not risky, we want to prioritize both Precision and Recall for class Y. In this context, SVM and AdaBoost stand out with high precision for risky credits. Further analysis, feature engineering, hyper parameter tuning, or employing other advanced techniques could help improve the models' performance for this specific binary classification task of credit risk assessment.

4.3.2. Deep learning approach

This paper introduces a methodology designed for training and evaluating a neural network model tailored to binary classification tasks, demonstrated within the domain of the loan prediction problem. The outlined approach encompasses essential stages, including data loading, preprocessing, and model configuration, culminating in a comprehensive evaluation through the utilization of stratified k-fold cross-validation.

The initial phase of the study involves acquiring and preliminarily exploring both training and testing datasets. Leveraging the Pandas library, CSV files are ingested, and crucial dataset attributes, such as dimensions and missing values, are meticulously examined. To address potential anomalies like empty datasets or parsing errors, robust error-handling mechanisms are implemented.

Following data loading, the training and testing datasets undergo a cleaning process by removing instances with missing values. Additionally, a randomized duplication of 50%, the choice of this rate is a recommendation based on the size of the used dataset, of instances is performed to augment the dataset size, aiming to enhance the model's robustness. In this context, it is assumed that the recommended splitting ratio values are applied: 70% for training, 20% for validation, and 10% for testing processes.

The subsequent phase of the study involves configuring and training neural network models using the Keras library. The model architecture includes a variable number of hidden layers ranging from 1 to 20, with 20 chosen as the final value for the epochs argument based on the size of the available dataset. Each hidden layer houses densely connected nodes activated through rectified linear units (ReLU), and dropout layers are strategically introduced to mitigate overfitting. A stratified k-fold cross-validation approach is employed, ensuring a robust evaluation of the model across diverse subsets of the data.

Upon completion of the training, the optimal model configuration is obtained, as depicted in **Figure 6** below.

Figure 6. Best CNN configuration.

The study concludes by persisting the best-performing model and providing a concise summary of the evaluation metrics. This summary includes details on the superior model configuration, average accuracy, precision, recall, and F1 score across the folds. The reported results serve as a benchmark for future comparative analyses and offer valuable insights into the model's efficacy in the loan prediction domain. **Figure 7** below illustrates the received results.

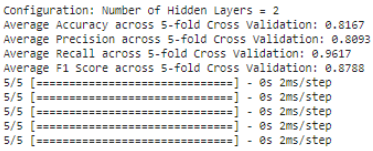

Figure 7. Best CNN configuration.

In summary, this subsection outlines the rigorous methodologies employed to navigate the intricate landscape of model configurations. Our systematic approach, encompassing data preprocessing, model configuration exploration, and thorough cross-validation, ensures the selection of a robust and generalizable model for the loan prediction task.

The resulting neural network model demonstrates remarkable performance, surpassing the efficacy of other employed machine learning models. Across the 5-fold cross-validation, the model achieves an average accuracy of 0.8167, indicative of its ability to correctly classify instances. Notably, the precision metric stands at an impressive 0.8093, emphasizing the model's precision in identifying true positive instances. Furthermore, the average recall, a measure of the model's ability to capture relevant instances, excels at 0.9617. This high recall underscores the model's proficiency in minimizing false negatives. The F1 score, which harmonizes precision and recall, attains an admirable average of 0.8788. Collectively, these results not only establish the neural network as a robust performer but also position it as an outperformer when compared to alternative machine learning models employed in this study.

4.4. Implementation

As we reach this stage, we pose the question: 'What type of application can we use to deploy our machine learning model?'

To answer this question, we thoroughly examined our case.

Given that our field of work is credit risk, closely intertwined with sensitive banking information, it is paramount to ensure the privacy of both customer and bank data, while also taking into account the work environment for bank employees at their respective sites. Consequently, we have opted to deploy our machine learning model in the form of a web application. This strategic choice offers numerous advantages, including enhanced accessibility, providing a user-friendly interface for both bank employees and potential customers. This accessibility facilitates real-time decision-making, ensuring swift and informed assessments of loan applications. Additionally, web applications, while addressing privacy concerns, can be fortified with robust security measures to safeguard sensitive customer and bank data. They also offer scalability to manage a high volume of users and can be readily updated to adapt to evolving needs and regulations. Remote accessibility, data analytics capabilities, seamless integration with existing systems, and the potential for customer engagement further underscore the value of deploying our ML model via a web application.

In our work, we have chosen to use the Flask framework and the Python Pickle library for deploying a ML model for credit risk prediction, and this decision can be attributed to several compelling reasons:

- Simplicity and Lightweight: Flask, a micro web framework in Python, is renowned for its simplicity and minimalism. It offers precisely what is needed to build a web application without introducing unnecessary features. This makes it a lightweight and efficient choice for deploying an ML model, ensuring that the web application remains streamlined and not overly complex.
- Ease of Use: Flask is celebrated for its user-friendly nature and ease of use. Developers, regardless of their experience levels, can swiftly get started with Flask, making it an excellent fit for a wide range of projects, including ML model deployment.
- Integration: Flask seamlessly integrates with other Python libraries and tools. This compatibility is invaluable when working with ML models as it enables straightforward integration with data

preprocessing, model training, and prediction components.

- Web Application Development: Flask is purpose-built for web application development, rendering it an ideal framework for constructing a user-friendly interface through which users can interact with the ML model for credit risk prediction.
- Python Pickle Library: Python Pickle is a library used for serializing and deserializing Python objects, including ML models. It facilitates the saving and reloading of trained ML models, a critical requirement when deploying an ML model within a web application where the model needs to be loaded and utilized for predictions.
- Security: Flask affords the flexibility to implement essential security features such as authentication, authorization, and data encryption. These measures are paramount when handling sensitive customer and financial data in the context of credit risk prediction.
- Community and Documentation: Flask boasts an active and supportive community of developers, accompanied by extensive documentation and resources. This abundance simplifies the process of finding solutions to common challenges and troubleshooting issues.
- Scalability: Flask applications can be readily scaled to manage increased user traffic and accommodate growing data volumes. This scalability is pivotal for a credit risk prediction system that may need to cater to a substantial number of users and data points.
- Customization: Flask offers extensive customization capabilities, enabling developers to tailor the web application to the specific needs and requirements of the credit risk prediction project.

In summary, Flask and Python Pickle were selected for their simplicity, user-friendliness, integration capabilities, security features, and suitability for web application development. These technologies provide a robust foundation for constructing a user-friendly and efficient solution for credit risk assessment.

After training the decision tree classifier, a web application has been designed as shown in **Figure 8** where the customer will enter all the attribute values and the data will be given to the model to predict the classification of the concerned customers. It was based on the use of the Flask framework[34] and the Python Pickle library[35] to save the ML model. Given the significance of mobile applications in today's world[36], it is pertinent to develop a mobile application for credit risk classifications that adheres to software quality standards[37]. This ensures that the application meets non-functional requirements such as usability, reliability, security, etc[38].

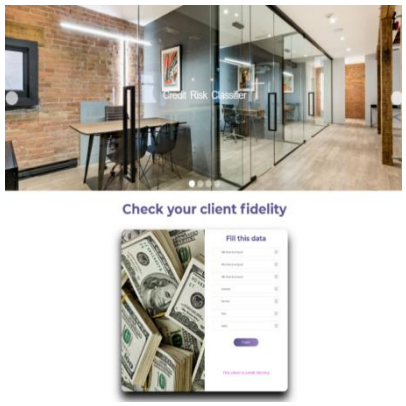

Figure 8. Credit Risk classifier web application.

5. Conclusion

In this contribution, the authors present a theoretical review of machine learning (ML) and its application in banking, specifically focusing on credit risk classification. They introduce an intelligent protocol that utilizes multiple classification algorithms to categorize customers based on their attributes.

Additionally, a user-friendly web application is designed to allow customers to input their attribute values and receive predictions regarding their classification, while providing bank managers with tools for credit risk management. The results highlight the effectiveness of the Support Vector Machine (SVM) classifier, achieving an accuracy of 77%, and the web application's ability to facilitate seamless interaction between customers and bank managers with the model.

The experiments in loan prediction demonstrate that a meticulously configured neural network surpassed alternative models. With a carefully designed architecture involving variable hidden layers and dropout mechanisms, the neural network achieved outstanding results during 5-fold cross-validation. Notably, its average accuracy, precision, recall, and F1 score reached 81.67%, 80.93%, 96.17%, and 87.88%, respectively, outperforming other models. This underscores the efficacy of complex neural network architectures in addressing intricate binary classification tasks, particularly evident in the nuanced domain of loan prediction. The identified best-performing model not only serves as a benchmark for future research but also holds practical significance for applications in predictive modeling.

6. Perspectives

Based on the size of the provided dataset for the Moroccan bank, we applied a variety of classification models. Additionally, we configured a CNN architecture chosen from a range of configuration states. In future works, we anticipate that other research teams will explore larger datasets to leverage the performance of more advanced architectures, such as transformers.

Author contributions

Conceptualization, KM, IEA, IR and SF; methodology, KM, IEA and SF; software, KM, IEA and SF; validation, KM, IEA and SF; formal analysis, KM, IEA and SF; investigation, KM, IEA, IR, HAA and SF; resources, KM, IEA, IR and SF; data curation, KM, IEA and SF; writing—original draft preparation, KM, IEA, IR, HAA and SF; writing—review and editing, KM, IEA, IR, HAA and SF; visualization, KM, IEA and SF; supervision, KM and IEA; project administration, KM. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Habehh H, Gohel S. Machine Learning in Healthcare. Current Genomics. 2021; 22(4): 291-300. doi: 10.2174/1389202922666210705124359
2. Javaid M, Haleem A, Pratap Singh R, et al. Significance of machine learning in healthcare: Features, pillars and applications. International Journal of Intelligent Networks. 2022; 3: 58-73. doi: 10.1016/j.ijin.2022.05.002
3. Weissler EH, Naumann T, Andersson T, et al. The role of machine learning in clinical research: transforming the future of evidence generation. Trials. 2021; 22(1). doi: 10.1186/s13063-021-05489-x
4. Tirkolaee EB, Sadeghi S, Mooseloo FM, et al. Application of Machine Learning in Supply Chain Management: A Comprehensive Overview of the Main Areas. Mathematical Problems in Engineering. 2021; 2021: 1-14. doi: 10.1155/2021/1476043
5. Yang M, Lim MK, Qu Y, et al. Supply chain risk management with machine learning technology: A literature review and future research directions. Computers & Industrial Engineering. 2023; 175: 108859. doi: 10.1016/j.cie.2022.108859
6. Lagasio V, Pampurini F, Pezzola A, et al. Assessing bank default determinants via machine learning. Information Sciences. 2022; 618: 87-97. doi: 10.1016/j.ins.2022.10.128
7. Boukherouaa EB, AlAjmi K, Deodoro J, et al. Powering the Digital Economy: Opportunities and Risks of Artificial Intelligence in Finance. Departmental Papers. 2021. doi: 10.5089/9781589063952.087.A001
8. Härle P, Havas A, Samandari H. The future of bank risk management McKinsey Working Papers on Risk. McKinsey & Company. 2015. Available online: https://www.mckinsey.com/business-functions/risk/our-

insights/the-future-of-bank-risk-management (accessed on 15 January 2024).
9. Milojević N, Redzepagic S. Prospects of Artificial Intelligence and Machine Learning Application in Banking Risk Management. Journal of Central Banking Theory and Practice. 2021; 10(3): 41-57. doi: 10.2478/jcbtp-2021-0023
10. van Liebergen B. Machine learning: A revolution in risk management and compliance? Journal of Financial Transformation. 2017; 45: 60-67.
11. Aziz S, Dowling MM. AI and Machine Learning for Risk Management. SSRN Electronic Journal. 2018. doi: 10.2139/ssrn.3201337
12. Bussmann N, Giudici P, Marinelli D, et al. Explainable Machine Learning in Credit Risk Management. Computational Economics. 2020; 57(1): 203-216. doi: 10.1007/s10614-020-10042-0
13. Alonso A, Carbo JM. Machine Learning in Credit Risk: Measuring the Dilemma Between Prediction and Supervisory Cost. SSRN Electronic Journal. 2020. doi: 10.2139/ssrn.3724374
14. Alexander C, Sheedy E. Developing a stress testing framework based on market risk models. Journal of Banking & Finance. 2008; 32(10): 2220-2236. doi: 10.1016/j.jbankfin.2007.12.041
15. Vasilopoulos C. Financial Stress Testing: A model based exploration under deep uncertainty. 2013.
16. Leo M, Sharma S, Maddulety K. Machine Learning in Banking Risk Management: A Literature Review. Risks. 2019; 7(1): 29. doi: 10.3390/risks7010029
17. Pathan S, Faff R. Does board structure in banks really affect their performance? Journal of Banking & Finance. 2013; 37(5): 1573-1589. doi: 10.1016/j.jbankfin.2012.12.016
18. Adams RB, Mehran H. Bank board structure and performance: Evidence for large bank holding companies. Journal of Financial Intermediation. 2012; 21(2): 243-267. doi: 10.1016/j.jfi.2011.09.002
19. Bossu W, Liu Y, Rossi ADP, et al. The Impact of Fintech on Central Bank Governance. FinTech Notes. 2021; 2021(001): 1. doi: 10.5089/9781513592473.063
20. Loan prediction problem dataset. Available online: https://www.kaggle.com/datasets/altruistdelhite04/loan-prediction-problem-dataset (accessed on 15 January 2024).
21. Lagani V, Karozou AD, Gomez-Cabrero D, et al. A comparative evaluation of data-merging and meta-analysis methods for reconstructing gene-gene interactions. BMC Bioinformatics. 2016; 17(S5). doi: 10.1186/s12859-016-1038-1
22. Roy B. All About Missing Data Handling. Medium. 2023. Available online: https://towardsdatascience.com/all-about-missing-data-handling-b94b8b5d2184 (accessed on 5 February 2023).
23. Missingness - an overview | ScienceDirect Topics. Available online: https://www.sciencedirect.com/topics/mathematics/missingness (accessed on 5 February 2023).
24. Badr W. Having an Imbalanced Dataset? Here Is How You Can Fix It. Medium. 2020. Available online: https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb (accessed on 6 February 2023).
25. Bach M, Werner A, Palt M. The Proposal of Undersampling Method for Learning from Imbalanced Datasets. Procedia Computer Science. 2019; 159: 125-134. doi: 10.1016/j.procs.2019.09.167
26. Mohammed R, Rawashdeh J, Abdullah M. Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. 2020 11th International Conference on Information and Communication Systems (ICICS); April 2020. doi: 10.1109/icics49469.2020.239556
27. Joseph VR, Vakayil A. SPlit: An Optimal Method for Data Splitting. Technometrics. 2021; 64(2): 166-176. doi: 10.1080/00401706.2021.1921037
28. Rokach L, Maimon O. Decision Trees. In: Maimon O, Rokach L (editors). Data Mining and Knowledge Discovery Handbook. Springer US, Boston, MA; 2005. pp. 165-192. doi: 10.1007/0-387-25465-X_9
29. Oetama RS. Enhancing Decision Tree Performance in Credit Risk Classification and Prediction. Ultimatics : Jurnal Teknik Informatika. 2015; 7(1). doi: 10.31937/ti.v7i1.349
30. Charbuty B, Abdulazeez A. Classification Based on Decision Tree Algorithm for Machine Learning. Journal of Applied Science and Technology Trends. 2021; 2(01): 20-28. doi: 10.38094/jastt20165
31. Hemanth J, Fernando X, Lafata P, et al. International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018. Springer International Publishing; 2019. doi: 10.1007/978-3-030-03146-6
32. Schölkopf B, Luo Z, Vovk V, et al. Empirical Inference. Springer Berlin Heidelberg; 2013. doi: 10.1007/978-3-642-41136-6
33. Taunk K, De S, Verma S, et al. A Brief Review of Nearest Neighbor Algorithm for Learning and Classification. 2019 International Conference on Intelligent Computing and Control Systems (ICCS); May 2019. doi: 10.1109/iccs45141.2019.9065747
34. Welcome to Flask - Flask Documentation (2.2.x). Available online: https://flask.palletsprojects.com/en/2.2.x/ (accessed on 15 February 2023).
35. pickle - Python object serialization. In: Python documentation. Available online: https://docs.python.org/3/library/pickle.html (accessed on 15 February 2023).
36. Moumane K, Idri A. Software quality in mobile environments: A comparative study. In 2017 4th International

Conference on Control, Decision and Information Technologies (CoDIT) (pp. 1123–1128). IEEE. 2017.
37. Moumane K, Idri A, Abran A. Usability evaluation of mobile applications using ISO 9241 and ISO 25062 standards. SpringerPlus. 2016; 5: 548.
38. Moumane K, Idri A. Using ISO 9126 with QoS DiffServ model for evaluating software quality in mobile environments. In 2014 Second World Conference on Complex Systems (WCCS) (pp. 134–139). IEEE. 2014.