Data Science in Finance and Economics

https://www.aimspress.com/journal/dsfe

DSFE, 5(3): 320–354.
DOI:10.3934/DSFE.2025014
Received: 24 November 2024
Revised: 1 June 2025
Accepted: 25 June 2025
Published: 24 July 2025

Research article

Comparative analysis of classical and Bayesian optimisation techniques: Impact on model performance and interpretability in credit risk modelling using SHAP and PDPs

Tatenda Shoko[1], **Tanja Verster**[2,3] **and Lindani Dube**[2,3,*]

[1] African Institute for Mathematical Sciences, 6 Melrose Rd, Muizenberg, Cape Town, 7950, South Africa
[2] Centre for Business Mathematics & Informatics, North-West University, Potchefstroom, 2531, South Africa
[3] National Institute for Theoretical and Computational Sciences (NITheCS), South Africa

* **Correspondence:** Email: Lindani.Dube@nwu.ac.za.

Abstract: Financial institutions often hesitate to use complex models such as random forests and extreme gradient boosting (XGBoost) for credit risk assessments due to challenges in selecting the optimal hyperparameters and the interpretability of these 'black box' models. This study addresses these issues by comparing the effects of classical grid search and Bayesian hyperparameter optimisation techniques on models' performance and interpretability in credit risk modelling. The results indicate that Bayesian optimisation improves recall for XGBoost, making it more effective at identifying defaulters while offering no notable advantage for random forests and even reducing performance for logistic regression. Although Bayesian optimisation reduces the computational time required for finding optimal hyperparameters, it does not improve the models' discriminatory power (area under the curve). The findings also suggest that different optimisation techniques influence feature importance and ranking, with SHapley Additive exPlanation (SHAP) values revealing that slight hyperparameter adjustments can lead to substantial changes in feature importance, particularly for logistic regression. However, partial dependence plots (PDP) for the variable *Rate* under Bayesian and classically optimised models are similar, indicating that using classical or Bayesian optimisation techniques does not alter the relationship between features and default probability. These insights emphasise the importance of selecting an appropriate optimisation approach to balance models' performance and explainability, with significant implications for credit risk modelling decisions.

Keywords: credit; random forest; classical optimisation; interpretability; machine learning; Bayesian optimisation; hyperparameters

JEL Codes: C41, C44, C61, G21, G32, O33

1. Introduction

Credit risk is the loss that the lender (mainly banks) may incur due to the borrower's failure to pay their financial obligations (McNeil et al., 2015). The probability of default (PD), a significant component of credit risk, is the numerical measure of the likelihood that the borrower will fail to honour their financial obligations within an agreed-upon period. Banks are in the business of granting loans to individuals; thus, the main function of credit is to channel the transfer of funds from savers to spenders. According to Zharikova et al. (2023), monitoring credit risk is necessary for economies to achieve an efficient allocation of funds. The Basel Committee on Banking Supervision (1999) states that managing credit risk involves establishing the maximum acceptable exposure to credit risk and ensuring that this exposure is within acceptable means to maximise the bank's risk-adjusted rate of return. Therefore, banks are encouraged to practice prudential risk management standards to reduce losses from defaults, ensuring efficient allocation of funds in the economy (Antony and Suresh, 2023).

Credit risk modelling is assessing and estimating PD and other credit risk parameters. Historically, default probabilities have been estimated using statistical methods such as linear regression and discriminant analysis (Altman, 1968; Hand and Henley, 1997). However, these methods often fail to capture complex and nonlinear relationships in the credit risk data (Helmy et al., 2023). With recent advances in technology and computing power, machine learning models such as random forest and extreme gradient boosting (XGBoost) have improved predictive performance, particularly in capturing and modelling nonlinear relationships (Helmy et al., 2023).

Despite the advantages of random forest and XGBoost models in handling complex nonlinear relationships in data, banks are not willing to use these algorithms to develop credit risk models due to two main challenges: Hyperparameter tuning and model interpretability (Xu, 2024). Hyperparameters such as the learning rate, the depth of the tree, and the number of trees in models' such as XGBoost affect models' performance, and finding the right values for these is critical to obtain the maximal models' performance without overfitting (Owen, 2022; Wu et al., 2019)

Another drawback of random forest and XGBoost is that the models are not interpretable. Random forest, XGBoost, and deep learning models are often called 'black box' models, in the sense that for the user of the model, the only available information is the inputs and outputs of the model, without knowing how the transformations inside the black box occur to convert inputs into outputs (Masís, 2021). In the banking industry, it is mandatory and essential to explain and justify the predictions made by machine learning models so that these models can be accepted by the stakeholders (Bertrand et al., 2024).

While studies have explored the impact of hyperparameter optimisation on the performance of machine learning models (Wu et al., 2019; Hu et al., 2020), and others have focused on exploring the interpretability of machine learning models for decision-making purposes (Rodríguez-Pérez and Bajorath, 2019; Dube and Verster, 2024), limited research has examined the effect of the choice of hyperparameter optimisation techniques on both the model performance and interpretability of the models, especially in credit risk modelling. This research addresses this gap by comparing two hyperparameter optimisation techniques: Bayesian optimisation and grid search. It evaluates their impact on models' performance, measured by accuracy, precision, recall, F1 score, and AUC as well as on model explainability, using SHAP and partial dependence plots (PDPs). The study applies these techniques to three machine learning models: Logistic regression, random forest, and XGBoost.

This work aims to answer the following research questions:

- How does the performance of Bayesian-optimised models differ from classical (grid search) optimised models when applied to credit risk datasets?
- Can SHAP effectively capture and interpret differences in predictions, feature importance, and feature contributions between models optimised using classical (grid search) and Bayesian methods?
- Can PDPs show changes in the relationships between the features and the probability of default (log-odds of defaulting) introduced by different classical and Bayesian optimisation techniques?

Our major contribution is exploring how classical and Bayesian hyperparameter optimisation methods affect models' performance and models' interpretability in credit risk assessment using two explainability techniques: SHAP and PDPs. This work aims to guide the selection of hyperparameter optimisation methods for efficient and understandable credit risk modelling.

The objectives of this work are as follows:

- To compare the performance of classical and Bayesian optimised models regarding accuracy, precision, recall, F1-score, and AUC;
- To examine the interpretability of these models using SHAP, analysing how different optimisation techniques affect features' importance and rankings.
- To use PDPs to determine whether different optimisation techniques change the relationships between the features and default probabilities.

Studying how the choice of hyperparameter optimisation techniques affects performance and model interpretability is crucial for credit risk modellers in banks aiming to build accurate and justifiable models for decision-making purposes, such as loan approvals, interest rate settings, and risk management strategies. This study provides guidance on selecting hyperparameter optimisation methods for efficient and interpretable credit risk modelling. It contributes to academic research and literature in the fields of predictive modelling, machine learning, and explainable machine learning.

The remainder of this study is organised into four sections. Section 2 presents related work and a detailed discussion of the machine learning algorithms, optimisation techniques, and explainability techniques used in this study. Section 3 describes the data, the preprocessing steps, and the experimental setup. The section also discusses the classification metrics that will be used to compare the performance of the optimised models. Section 4 presents models' performance and explainability results and provides a detailed discussion of the findings. Section 5 presents the conclusion, the limitations of our analysis, possible recommendations for risk modellers, and potential areas for future research. The appendices provide information about the hyperparameters, partial dependence plots, and probabilities.

2. Review of the literature

Recent advances in digital technology and data science have significantly influenced the application of machine learning techniques in finance, particularly in areas such as credit risk modelling. As financial institutions increasingly adopt digital infrastructures, the availability of large-scale financial

data and improvements in computational power have enabled the deployment of more complex models, such as random forests and gradient boosting methods, in risk assessment frameworks (Li et al., 2024). However, the adoption of these models remains limited due to challenges surrounding hyperparameter optimisation and models' interpretability, which are critical in regulated environments like credit scoring. Studies examining broader financial and technological linkages—such as the role of digital platforms in global value chains (Li et al., 2024) or the impact of e-commerce on the digital economy (Desalegn et al., 2024)—highlight how digital transformation drives demand for interpretable and effective analytics across domains. These insights echo findings in machine learning literature that emphasise the trade-offs between performance and transparency (Chen, 2024), reinforcing the importance of our study's focus on comparing classical and Bayesian optimisation methods. By situating our work within this evolving digital landscape, we contribute to a growing body of research that seeks to balance predictive accuracy with explainability in financial decision-making. In this literature review, we begin by discussing related work, then examine machine learning models, and finally explainable machine learning techniques.

2.1. Related work

In credit scoring, Xia et al. (2017) applied grid search, manual search, random search, and Bayesian optimisation to improve the performance of the XGBoost algorithm. They successfully compared the accuracy, error rate, and AUC of Bayesian optimised XGBoost to classically optimised (random, grid, and manual search) models. Their findings show that Bayesian hyperparameter tuning performs better than classical search methods with respect to accuracy, error rate, and AUC. The authors highlighted that Bayesian optimised XGBoost also provided interpretable feature importances and decision charts, bridging the gap between performance and transparency.

In their paper, Alonso Robisco and Carbo Martinez (2022) evaluated the impact of the random search and Bayesian optimisation methods on the performance of XGBoost and logistic regression models in corporate risk classification. Their results indicated that Bayesian optimised XGBoost consistently outperformed logistic regression in accuracy, AUC, recall, and F1 score. These findings demonstrate the impact of Bayesian optimisation on models' performance and complexity. Similarly, Wang and Ni (2019) compared random search and Bayesian tree-structured Parzen estimators (TPE) optimisation methods on the performance of XGBoost and logistic regression. They reported that Bayesian TPE-optimised XGBoost outperforms logistic regression in accuracy, AUC, and F1 score, underscoring the need for hyperparameter optimisation to enhance models' performance.

Yang et al. (2022) applied Bayesian optimisation to find the best parameters for XGBoost, random forest, and gradient boosting decision trees (GBDT) in personal credit delinquency prediction. Their study recorded AUC values of 0.92 for the optimised random forest, 0.94 for GBDT, and 0.95 for XGBoost, demonstrating that hyperparameter optimisation improves the discriminatory power of these models. The authors also noted that Bayesian optimisation is computationally efficient, a critical factor in credit risk predictions where time is often a constraint.

Similarly, Kong et al. (2023) used Bayesian optimisation to optimise XGBoost parameters in credit scoring, achieving an AUC of 0.95, significantly higher than logistic regression (an AUC of 0.80) and GBDT (an AUC of 0.92). Their findings align with Yang et al. (2022), confirming that Bayesian optimisation enhances the discriminatory power of XGBoost. To address models' transparency, they applied SHAP to explain predictions, using SHAP summary, decision, waterfall, and force plots to demonstrate an improved interpretability of the Bayesian-optimised XGBoost model. In a related study,

du Toit et al. (2024) evaluated Shapley values as a model-agnostic interpretability technique in credit scoring. They tested Shapley values on simulated datasets with linear and nonlinear relationships, showing that Shapley values are related to the weights of evidence, a well-known measure in scorecard literature, and can effectively explain the direction of the relationships between explanatory variables and outcomes.

Financial institutions require models that are both high-performing and interpretable for decision-making. De Lange et al. (2022) used Shapley values to explain the predictions of a light-gradient model in credit scoring for consumer loans. Their research highlighted the benefits of explainability techniques like Shapley values to improve models' transparency, particularly for stakeholders needing insights into features influencing credit default predictions. They also emphasised the need for interpretability-focused research to bridge the gap between traditional interpretable models (e.g., logistic regression) and complex machine learning models like XGBoost.

In bank churn predictions, Dube and Verster (2024) applied Shapley values, breakdown plots, and partial dependence plots (PDPs) to address the interpretability of random forest models under class imbalance. Their findings suggest that Shapley values are essential for revealing changes in feature importance as class imbalance varies, while PDPs provide consistent insights into feature-target relationships. Their study underscores the importance of using multiple explainability techniques to fully understand complex models. Similarly, Kłosok et al. (2020) addressed the explainability in credit risk models by employing PDPs, feature importance, accumulated local effects, individual conditional expectation (ICE) curves, and Shapley values to interpret random forest predictions. Their results demonstrate that these tools produce reliable explanations, enhancing the transparency and accountability of credit risk systems.

Beyond credit scoring, explainability techniques have been applied in other domains. Gawde-Prabhudesai et al. (2024) explored explainable artificial intelligence (AI) in predictive maintenance of rotating machines, integrating LIME, SHAP, PDP, and ICE to interpret AI models' predictions. Their approach involved multi-sensor data acquisition and frequency-domain feature extraction, with the results showing that these explainability techniques provide human-understandable insights into AI-driven maintenance decisions, improving trust and model validation. In another study, Khan et al. (2025) conducted a comparative analysis of automated machine learning (AutoML) frameworks for hyperparameter optimisation and explainability techniques in predicting the ultimate moment capacity of ultra-high-performance concrete (UHPC) beams. They found that the Optuna AutoML framework achieved the highest predictive accuracy with minimal computational time, while SHAP outperformed LIME, PDP, and permutation importance in providing detailed and actionable insights for feature contributions, particularly in high-stakes engineering applications.

In the context of Bayesian optimisation, Rodemann et al. (2024) proposed ShapleyBO, a framework that uses Shapley values to interpret Bayesian optimisation proposals in black box optimisation problems, such as personalising wearable robotic devices. Their approach quantifies each parameter's contribution to the acquisition function, disentangling exploration and exploitation dynamics and enhancing human-AI collaboration through a ShapleyBO-assisted human-machine interface. Their findings suggest that such interpretability tools can reduce regret in human-in-the-loop applications, highlighting the broader applicability of Shapley values in optimisation interpretability.

Our work aimed to assess the impact of different hyperparameter optimisation techniques not only on the performance but also on the interpretability of XGBoost, random forest, and logistic regression

models when applied to credit risk data. We compare two optimisation methods: Bayesian optimisation and grid search. The best hyperparameters obtained from these methods are used to build classification models for credit risk data. Additionally, SHAP and PDPs are used to analyse the effects introduced by the different optimisation techniques, ensuring both performance and transparency are addressed.

2.2. Machine learning models

This study used three classifiers: logistic regression, random forest, and XGBoost. Logistic regression is applied because it is widely used in credit due to its interpretability based on its coefficient (Gatla, 2023). The random forest model is preferred due to its ability to give higher accuracy, handle large data, and work with incomplete data (Wu et al., 2019). XGBoost has been adopted because of its great performance in speed, sensitivity in memory usage, and multiple hyperparameters, making it suitable for comparison purposes with random forest (Chen and Guestrin, 2016).

2.3. Logistic regression

Logistic regression is a widely used classification algorithm in credit risk modelling for predicting the probability that a customer will default or not, given a set of explanatory variables (Bussmann et al., 2021). Let $\mathbf{x}_i = \{x_{i1}, \ldots, x_{ip}\}$ be a set of explanatory variables, and $\mathbf{y}_i \in \{0, 1\}$ be the binary target set representing defaulting (1) and not defaulting (0) then; in this case, the log-odds of probability of defaulting ($P(y_i = 1)$) are given by a linear combination explanatory variables as shown in Equation 1 below (Murphy, 2022):

$$\log \frac{P(y_i = 1)}{1 - P(y_i = 1)} = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}, \tag{1}$$

where β_0 is the intercept term, and β_j represents the feature coefficients obtained by using maximum likelihood estimation to maximise the probability of observed data. Each coefficient β_j is the effect of a 1-unit change in $x_{i,j}$ on the log-odds of defaulting.

The probability of default is obtained by transforming Equation 1 into Equation 2:

$$P(y_i = 1 \mid \mathbf{x}_i) = \frac{e^{\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}}}{1 + e^{\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}}}. \tag{2}$$

The logistic regression model does not have many hyperparameters. In this work, the only one was `C`, which controls how the regularisation strength was tuned (Ahmed Arafa et al., 2022). We trained the logistic regression model using the `LogisticRegression` function from the `scikit-learn` library. We used grid-search and Bayesian optimisation techniques (see Section 2.6) to find the optimal values of `C`. The results of the optimal values of `C` are provided in Table 5 in Appendix A.

2.4. Random forest

Random forest is an ensemble learning method that combines multiple decision trees to create a more robust and accurate model through a process called bagging (Breiman, 2001). Each decision tree in the forest recursively partitions the data into subgroups to minimise node impurity, which is typically measured by the Gini impurity or cross-entropy at each split (Murphy, 2022).

Given a training dataset $D = \{\mathbf{x}, \mathbf{y}\}$, where $\mathbf{x}$ is the feature matrix with p input features and a binary target vector $\mathbf{y}$, where $\mathbf{y} \in \{0, 1\}$ are class labels, each instance is represented as (x_i, y_i). The decision tree splits the feature space into M regions $R_1, R_2, \ldots, R_M$, and the model predicts a constant c_m for each region acco the majority class.

The definitions and equations in this subsection are adapted from Hastie et al. (2009) and Murphy (2022) unless otherwise specified.

The estimated model $\hat{f}(x)$ can be written as

$$\hat{f}(x) = \sum_{m=1}^{M} c_m I(x \in R_m), \tag{3}$$

where $I(x \in R_m)$ is 1 if x belongs to region R_m, and 0 otherwise. The quality of each split is determined by impurity measures, such as the Gini impurity or cross-entropy:

$$\text{Gini impurity} = \sum_{k=0}^{1} p_{mk}(1 - p_{mk}), \tag{4}$$

$$\text{Cross-Entropy} = -\sum_{k=0}^{1} p_{mk} \log(p_{mk}), \tag{5}$$

where p_{mk} is the proportion of class k in region R_m.

According to Wu et al. (2019), the random forest algorithm selects a random subset of features $m \leq p$ at each split (where p is the total number of features) and chooses the best split from that subset. This randomness helps reduce overfitting and improves the model's generalisation to unseen data (the test dataset). After growing B trees, the random forest predicts the class of a new observation by taking a majority vote across all trees.

The prediction at a new point x_i is given by

$$\hat{C}^B_{\text{rf}}(x) = \text{majority vote}\{\hat{C}_b(x_i)\}^B_1, \tag{6}$$

where $\hat{C}^B_{\text{rf}}(x)$ is the new prediction and $\hat{C}_b(x)$ is the class prediction made by the b-th tree in the forest.

We trained the random forest model using the `RandomForestClassifier` from Python's `scikit-learn` library (Pedregosa et al., 2011). The primary hyperparameters considered were the number of decision trees (B), the maximum depth of each tree, and the splitting criteria (Gini or entropy). Both the grid search and Bayesian optimisation techniques were used to determine the optimal hyperparameters for the random forest. Detailed information about the hyperparameters considered, the search space, and the optimal hyperparameters identified can be found in Table 4.

2.5. Extreme gradient boosting (XGBoost)

XGBoost (Chen and Guestrin, 2016) is a machine learning algorithm designed as an improvement over the traditional gradient boosting method (Friedman, 2001). XGBoost sequentially builds multiple decision trees, with each new tree correcting the errors made by the previous ones, creating a stronger predictive model. At each step, XGBoost focuses on the residuals (errors) from previous trees. A key advantage of XGBoost over the traditional gradient boosting tree algorithm is the presence of a

regularisation term in the objective function, which helps to reduce overfitting while improving the model's accuracy.

In XGBoost, the prediction for each instance x_i is represented as

$$\hat{f}(x_i) = \sum_{b=1}^{B} f_b(x_i), \tag{7}$$

where $f_b(x_i)$ is the prediction from the b-th tree, and B is the number of trees.

The objective of XGBoost is to minimise the loss function L_b, which measures the difference between the predicted value and the actual value. The loss function is:

$$L_b = \sum_{i=1}^{n} l(y_i, \hat{y}_i), \tag{8}$$

where $l(y_i, \hat{y}_i)$ is a function measuring the loss between the actual y_i and the predicted $\hat{y}_i$.

To prevent overfitting, XGBoost incorporates a regularisation term $\Omega(f_b)$ into the objective function:

$$L_b = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{b=1}^{B} \Omega(f_b). \tag{9}$$

The regularisation term is defined as:

$$\Omega(f_b) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \tag{10}$$

where T is the number of leaves, w_j is the weight for leaf j, γ controls model complexity, and λ is the regularisation parameter.

We trained the XGBoost model using the `XGBClassifier` from Python's `xgboost` library. Key hyperparameters such as the number of trees, maximum depth, learning rate, and regularisation terms were optimised using both grid search and Bayesian optimisation. Details of the hyperparameters and their optimal values are given in Table 3 in Appendix A.

2.6. Hyperparameter optimisation methods

Hyperparameters are user-defined parameters of the machine learning model whose values control the training of the machine learning algorithm (Wu et al., 2019). Hyperparameter optimisation aims to find the best hyperparameters from the defined domain that give the best score on the test dataset without overfitting. Mathematically, hyperparameter optimisation can be represented as (Owen, 2022):

$$x^\star = \arg\max_{x \in \mathcal{X}} f(x), \tag{11}$$

where $f(x)$ is the objective score to be maximised (recall in this study) in the test dataset, $x^\star$ is the set of hyperparameters that gives the highest score, and x is any possible value which the hyperparameter can take from the user-defined domain set $\mathcal{X}$.

This study considers two methods for tuning the hyperparameters: Bayesian optimisation and grid search. We will apply these two methods to find the optimal hyperparameters for XGBoost, random forest, and logistic regression.

2.7. Bayesian hyperparameter optimisation

Bayesian optimisation is a method designed to improve the performance of the objective function by applying a probabilistic model of the objective function. It then uses this model to select hyperparameters that evaluate the true objective function. The Bayesian optimisation algorithm is based on Bayes' Theorem to search for the maximum of the objective function (which is recall in our study; see Section 3.4).

Definitions and equations in this subsection are acknowledged from work by Overisch (2020) and Bergstra et al. (2011) unless specified.

The Bayes Theorem is given in Equation 12

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}, \tag{12}$$

where $P(y|x)$ is the posterior term, $P(x|y)$ is the likelihood, $P(y)$ is the prior term, and $P(x)$ is a normalising constant. Bayesian optimisation begins with the definition of the set of hyperparameters to be optimised as $\mathbf{X}_i = (x_1, x_2, ..., x_n)$, and the objective function to be maximised $f(\mathbf{X}_i)$, which measures the performance (or cost) associated with each sample. The next step is sequential sampling, where data is collected from each hyperparameter set, and the performance is calculated as given in Equation 13 below

$$D = \{(x_1, f(x_1)), (x_2, f(x_2)), ..., (x_n, f(x_n))\}. \tag{13}$$

These samples are drawn from ranges defined by each hyperparameter's upper and lower bounds. The cost D, and the objective function f, the posterior distribution $p(f|D)$ can be expressed using the Bayes' Theorem as follows:

$$p(f|D) = P(D|f) \cdot P(f). \tag{14}$$

Here, the prior $P(f)$ captures any prior information about the objective function, while the likelihood $P(D|f)$ is the probability of observing the data given the objective function. After the initial sampling, Bayesian optimisation uses the likelihood term in Equation 14 to estimate the objective function to form a surrogate model (the probability representation of the objective function), then uses the surrogate model to select the hyperparameters. Commonly used surrogate models include the Gaussian process, random forest, and tree-structured Parzen estimators (TPE) (see Shahriari et al. (2016). This study chooses TPE as the surrogate model because it can handle both continuous and categorical hyperparameters (Owen, 2022).

The TPE approach uses the Bayes' Theorem in Equation 12 to model the probability distribution of the hyperparameters given the objective function $p(x|y)$. The TPE selects some threshold quantity y^* to be some quantile γ of the observed y. The distributions of the hyperparameters can be modelled as follows:

$$p(x|y) = \begin{cases} l(x) & \text{if } y < y^* \\ g(x) & \text{if } y \geq y^*, \end{cases} \tag{15}$$

where $l(x)$ is the distribution of samples whose objective values are better than the threshold y^*, while $g(x)$ is the distribution formed by using the remaining observations. The quantile γ determines the proportion of samples for which $p(y < y^*) = \gamma$.

After constructing a surrogate model, an acquisition function is used to suggest new hyperparameter values based on the current surrogate model $p(x|y)$ from Equation 15. The acquisition function balances

the trade-off between exploration (searching new regions of hyperparameter space with little known information) and exploitation (evaluating the regions where the objective function performs well). Commonly, acquisition functions include the expected improvement (EI) and the probability of improvement (PI). This study used the EI because it can balance exploitation and exploration efficiently (Akiba et al., 2019).

The EI under the TPE approach is given by:

$$EI_{y^\star}(x) = \int_{-\infty}^{y^*} (y^* - y)p(y|x)dy \tag{16}$$

$$= \int_{-\infty}^{y^*} (y^* - y)\frac{p(x|y)p(y)}{p(x)}dy. \tag{17}$$

By letting $\gamma = p(y \leq y^*)$ and $p(x) = \int_{-\infty}^{\infty} p(x|y)p(y)dy = \gamma l(x) + (1-\gamma)g(x)$ and substituting the last expression into Equation 17, we obtain the following:

$$EI_{y^\star}(x) = \int_{-\infty}^{y^*} (y^* - y)\frac{p(x|y)p(y)}{\gamma l(x) + (1-\gamma)g(x)}dy. \tag{18}$$

Optimising EI means focussing on the 'good' region of the objective function where $y \leq y^*$. Substituting $p(x|y) = l(x)$ from Equation 15 into Equation 18

$$EI_{y^\star}(x) = l(x)\int_{-\infty}^{y^*} (y^* - y)\frac{p(y)}{\gamma l(x) + (1-\gamma)g(x)}dy. \tag{19}$$

Equation 19 can be simplified further into Equation 20

$$EI_{y^\star}(x) = \frac{\gamma y^* l(x) - l(x)\int_{-\infty}^{y^*} p(y)dy}{\gamma l(x) + (1-\gamma)g(x)} \propto \left(\gamma + \frac{g(x)}{l(x)}(1-\gamma)\right)^{-1}. \tag{20}$$

From the last expression of Equation 20, $EI_{y^\star}(x) \propto \frac{l(x)}{g(x)}$. To achieve high EI, the algorithm looks for points where $l(x)$, the likelihood of finding good results, is high, and $g(x)$, the likelihood of bad results, is low. In each iteration, TPE evaluates new sets of hyperparameters, updating $l(x)$ and $g(x)$ on based on the observed results. This iterative process continues until the specified number of evaluations is reached, ultimately converging on a set of hyperparameters that optimises the objective function.

2.8. Grid search

According to Bentéjac et al. (2021), grid search is a traditional (classical) exhaustive approach that explores every possible combination of hyperparameters. This method consists of training the machine learning model with each possible hyperparameter configuration in a training dataset and evaluating its performance using a predetermined metric on a test set. In our study, we chose recall as the metric to be optimised in order to develop models that can capture as many defaulters as possible by minimising false negatives.

The grid search algorithm works as follows (Owen, 2022):

- Define $\mathcal{X}$, the search space.

- Form a loop to cycle through each value in $\mathcal{X}$.
- Carry out cross-validation on the training set and record the cross-validation scores along with the best parameter combinations.
- Retrain the model with the optimal hyperparameters.
- Assess the model performance on the test dataset.

The drawback of grid search is that it is possible to miss out on the hyperparameters that are not in $\mathcal{X}$. This study uses grid search because it is easy to implement, and it is the most widely used technique for hyperparameter tuning (Wu et al., 2019).

2.9. Explainable machine learning techniques

According to Masís (2021), interpretability in machine learning is the extent to which the user of the model can understand the reason behind the prediction of the model. The author even distinguished between interpretability and explainability, adding that explainability is not just understanding the reasons behind models' predictions, but these decisions must be ethical and human-friendly. Thus, interpretable machine learning involves getting useful information from a machine learning model about the relationships among the variables in the data or estimated from the model (Molnar, 2020). This study focuses on two post hoc explainability techniques: SHAP and PDPs. These are chosen because they are model invariant, can be applied to interpret any 'black box' model, and are easy to implement. Explainability techniques are used in this study to examine whether Bayesian-optimised models will result in different feature relationships, contributions, and predictions from classical-optimised ones.

2.10. SHapley Additive exPlanations (SHAP)

SHAP is a game-theoretic approach to interpreting machine learning models' predictions, introduced by Lundberg and Lee (2017). It is based on Shapley values from cooperative game theory, proposed by Shapley (1953), which provide a way to fairly allocate the contribution of each feature to a model's prediction (pay-off). Specifically, SHAP values represent the impact of individual features on the model's output (e.g., probability or log-odds of default), by calculating each feature's average marginal contribution across all possible feature combinations (Masís, 2021).

The Shapley value ϕ_j for a feature j is computed as follows (Lundberg and Lee, 2017):

$$\phi_j = \frac{1}{|N|!} \sum_{S \subseteq N \setminus \{j\}} |S|!(|N| - |S| - 1)![f(S \cup \{j\}) - f(S)], \tag{21}$$

where N is the set of all features, S is a subset of features that do not include j, $f(S)$ is the prediction of the model when only the features in S are considered and $f(S \cup \{j\})$ is the prediction when adding the feature j to the subset S.

As noted by Rodríguez-Pérez and Bajorath (2019), the Shapley values can be seen as a fair distribution of feature importance for a given models' prediction. Positive Shapley values indicate features that contribute positively to the prediction, negative values indicate a negative contribution, and values near zero suggest no significant effect on the prediction. Averaging the Shapley values across all instances allows SHAP to serve as a tool for both global and local interpretability.

In this study, we used Python's `SHAP` library, which provides both model-agnostic and model-specific explainers to visualise the importance of characteristics throughout the data set. For local explanations, SHAP calculates each variable's contribution to an individual prediction, while global explanations can be obtained by aggregating these contributions. The SHAP waterfall plot is particularly effective for visualising local explanations, showing how each feature influences a single prediction. For tree-based models (XGBoost and random forest), the study uses SHAP's `TreeExplainer` to interpret predictions, while for logistic regression models, we employed the `LinearExplainer`.

2.11. Partial Dependence Plots (PDPs)

A PDP (Friedman, 2001) is a global explainability tool that shows how the models' output (such as the probability of default or log-odds of defaulting) changes as one feature is varied while the effects of other features are averaged out over their distribution across the dataset (Kłosok et al., 2020). By visualising the relationship between a feature of interest and the model's output, a PDP helps identify whether the relationship is linear, monotonic, or more complex (Molnar, 2020).

Definitions, equations, and related concepts in this subsection are drawn from the work of (Hastie et al., 2009, pages 369–370) unless otherwise specified.

To understand how the PDP is computed, let $X = (X_1, X_2, \ldots, X_p)$ represent a vector of p predictor variables in a dataset with n observations. Suppose we are interested in understanding the effect of a specific subset of features $X_S \subseteq X$ on the models' output. In that case, the partial dependence function on the model's prediction function $\hat{f}(X)$ on X_S is defined as:

$$f_{\text{pd}}(X_S) = \mathbb{E}_{X_C}[\hat{f}(X_S, X_C)] = \int \hat{f}(X_S, X_C)\, dP(X_C), \tag{22}$$

where $X_C = X \setminus X_S$ represents the complementary set of features, and $dP(X_C)$ is the marginal distribution of X_C. Since the true distribution of X_C is typically unknown, the expectation in Equation 22 is approximated by averaging over the observed values of X_C. The estimated partial dependence function is then expressed as:

$$\widehat{f_{\text{pd}}(X_S)} = \frac{1}{n} \sum_{i=1}^{n} \hat{f}(X_S, x_{iC}), \tag{23}$$

where x_{iC} denotes the observed values of X_C for the i-th observation.

For continuous features, a PDP is created by plotting the averaged predictions $\widehat{f_{\text{pd}}(X_S)}$ against the feature values of X_S. For categorical features, a PDP is mainly a bar plot, where each bar corresponds to a unique category, and its height represents the associated partial dependence value given by Equation 23 (Kłosok et al., 2020).

PDPs are most effective when one or two features are visualised, as representing interactions among more than two features can become difficult. When applied to credit risk data, PDPs show how, on average, the probability of default, or log odds, fluctuates as the features of interest vary. In this study, we used `PartialDependenceDisplay` class from Python's `sckit-learn` library to compute PDPs.

Table 1 provides an overview of the machine learning models, hyperparameter optimisation techniques, and explainability tools used in our experiment. It highlights key principles, strengths, weaknesses, and applicable contexts for each method. Logistic regression remains a regulatory-friendly, interpretable baseline model, while ensemble methods like random forest and XGBoost offer improved

predictive power at the cost of transparency. For model tuning, grid search ensures thoroughness but is computationally demanding, whereas Bayesian optimisation offers efficiency for complex models. In terms of explainability, SHAP values offer rigorous, theoretically grounded insights into models' behaviour, whereas PDPs provide quicker, more intuitive visual summaries, albeit with limitations in handling features' interactions. This comparative synthesis informs model selection and deployment decisions by balancing accuracy, interpretability, and regulatory alignment.

Table 1. Summary of the machine learning models, optimisation, and explainability techniques.

Method	Principles	Advantages	Disadvantages	Scope of Application
Logistic regression	A linear model predicting probability via log-odds; assumes linear feature-target relationships.	Interpretable coefficients; computationally efficient; aligns with regulatory needs.	Struggles with nonlinear relationships; sensitive to outliers, multicollinearity.	Ideal for transparent, linear credit risk tasks; suits regulatory reporting.
Random forest	Ensemble of decision trees using bagging; captures nonlinear patterns via random feature splits.	Robust to noise, missing data; provides feature importance; handles nonlinearity.	Less interpretable; computationally intensive; requires extensive tuning.	Suited for nonlinear, complex credit datasets; less ideal for regulatory transparency.
XGBoost	Gradient boosting with sequential trees; optimizes regularised objective for high accuracy.	High predictive accuracy; handles nonlinearity; includes regularisation; feature importance.	Black-box model; complex tuning; longer training times.	Best for maximising accuracy in competitive lending; needs explainability tools.
Grid search	Exhaustive search over predefined hyperparameter combinations; evaluates all via cross-validation.	Simple, reliable; ensures comprehensive search; deterministic results.	Computationally expensive; misses values outside grid; redundant evaluations.	Suitable for simple models or abundant resources; less practical for complex models.
Bayesian optimisation	Probabilistic search using surrogate model and acquisition function; balances exploration-exploitation.	Efficient; fewer evaluations; adapts to complex models; reduces computation time.	Complex to implement; relies on surrogate quality; may miss global optima.	Ideal for complex models like XGBoost; less effective for simple models.
SHAP	Game-theoretic approach; assigns feature contributions based on marginal impact across combinations.	Detailed local/global explanations; consistent; model-agnostic; regulatory-compliant.	Computationally intensive; approximations may reduce accuracy in high dimensions.	Best for detailed audits, regulatory compliance; less suited for real-time applications.
PDPs	Visualises average feature effect on predictions, marginalising other features' effects.	Intuitive; computationally light; shows global feature-target trends; easy to interpret.	Assumes feature independence; limited for interactions; less detailed than SHAP.	Useful for quick, high-level insights into feature effects; less effective for complex interactions.

3. Materials and methods

3.1. Data used

This study utilised the *Loan Applicant Data for Credit Risk Analysis*, an open dataset hosted on Kaggle, a well-known platform for data scientists and machine learning practitioners (Kaggle, 2021). The dataset includes attributes related to loan applicants, such as age and income. It also contains

loan-specific features such as approval status and interest rates. It comprises 11 features and 32,581 instances. The features in the dataset are summarised in Table 2 below.

Table 2. Loan applicant data for credit risk analysis dataset (see Kaggle (2021)).

Feature	Description	Data type
Age	Age of the loan applicant	Numerical
Income	Income of the applicant	Numerical
Home	Home ownership status (Own, Mortgage, Rent)	Categorical
Emp_Length	Employment history in years	Numerical
Intent	Purpose of the loan (e.g. education, home improvement)	Categorical
Amount	Amount taken for loan	Numerical
Rate	Interest rate on the loan	Numerical
Status	Approval status (Fully Paid, Charged Off)	Categorical
Percent_Income	Loan amount as a percentage of income	Numerical
Default	Default history of the applicant (Yes - Defaulted, No - Not Defaulted)	Categorical
Cred_Length	Length of the applicant's credit history	Numerical

The dependent variable in this dataset is *Default*, which indicates whether the applicant has defaulted on previous loans. Other variables given in Table 2 are explanatory variables.

3.2. Preprocessing of data

The data preprocessing phase involved identifying and removing 165 duplicate rows, as they did not provide any additional information to the analysis. For missing data, the *Rate* and *Emp_Length* columns had some missing values affecting less than 30% of the dataset. Mean imputation was used to fill in the missing values (Caton et al., 2022). Dummy variables for the categorical variables were introduced by using one-hot encoding, which was suitable, given the number of categories. The target variable *Default* was converted from categorical values ('Y', 'N') to binary labels (1 for 'Y' and 0 for 'N') to meet the requirements of machine learning models, which require numerical targets. Random forest and XGBoost are robust models that can handle missing data, but data cleaning and preprocessing steps were applied across all models to ensure a fair comparison of the results.

Figure 1 shows the class distribution of defaulters and non-defaulters in the cleaned dataset. The data is imbalanced, with 82.3% (26 686) non-defaulters and only 17.7% (5730) defaulters. According to Dube and Verster (2023), imbalanced classes can lead to skewed results, especially accuracy measures, since the classifiers tend to capture non-defaulters but fail to capture defaulters.

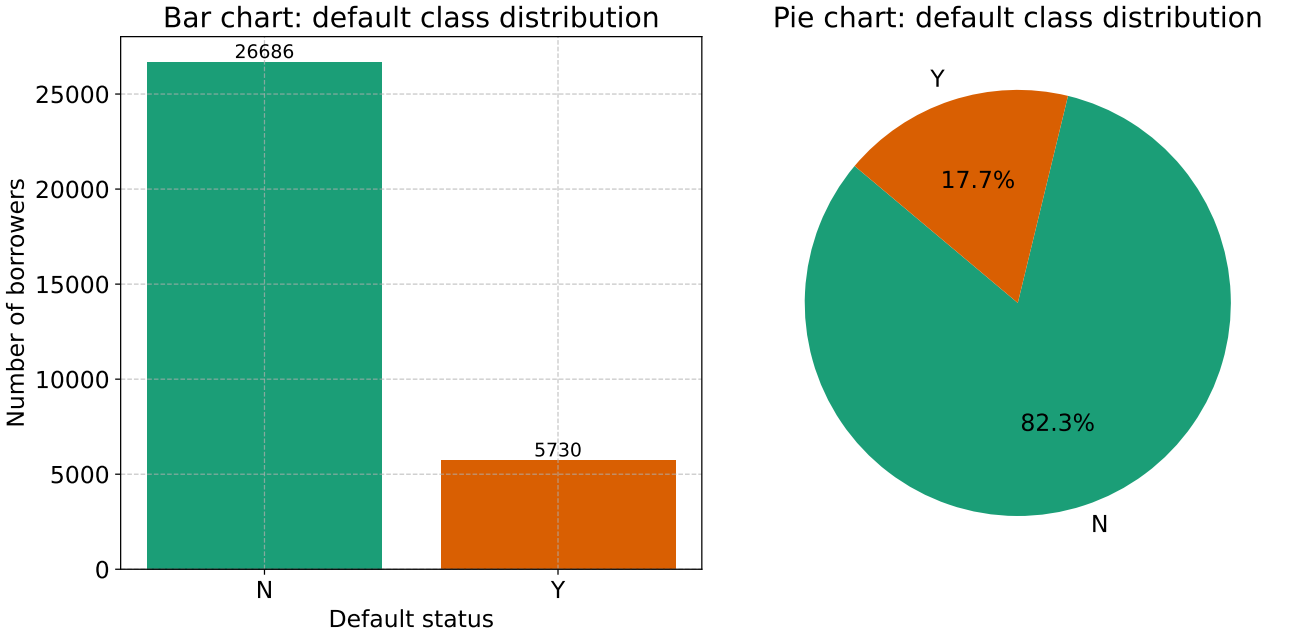

Figure 1. Default status: Class distribution.

Adaptive synthetic sampling (ADASYN), a commonly used method to address class imbalance, was chosen in this work because the method can adjust the weight distribution of the underrepresented class. The algorithm produces synthetic examples for observations that are difficult to classify in the minority class in the underrepresented class (He et al., 2008). Class imbalance is addressed only after the data is split into training and testing sets to avoid data leakage. This ensures that the test dataset represents real-world conditions, with ADASYN sampling applied only to the training set.

Before training the models, the dataset was split into training and testing sets using the `train_test_split` function from Python's `scikit-learn` library. This function performs random shuffling to ensure an unbiased partition. Here, 80% of the data were used for training, and the remaining 20% were left aside to evaluate the performance of the model.

3.3. Searching for optimal hyperparameters

The study used `Optuna`, an open-source Python library, to perform Bayesian optimisation and identify the optimal hyperparameters for the XGBoost, random forest, and logistic regression models. `Optuna` employs the TPE as its default surrogate model, with recall as the objective function. This approach allows for more efficient hyperparameter selection, requiring fewer evaluations than traditional methods (Akiba et al., 2019). For the classical approach, Python's `GridSearchCV` function from the `scikit-learn` library was used. In both the Bayesian and classical optimisation techniques, 5-fold stratified cross-validation was used in this work to reduce model overfitting and to improve the reliability of the results.

More information on the hyperparameter functions, domain space, and optimal values for XGBoost, random forest, and logistic regression is given in Tables 3, 4, and 5, respectively, in Appendix A.

3.4. Model evaluation metrics

The study used the classification metric values to compare the performance of the models optimised using the Bayesian and grid search optimisation techniques. The description and formulas of the five evaluation metrics of the metrics as discussed by Abhishek and Abdelaziz (2023), are given as follows

- **Accuracy**: The proportion of correct predictions out of all model predictions. Accuracy is calculated as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{24}$$

  where, TP, TN, FN, and FP represent true positives, true negatives, false negatives, and false positives, respectively. Accuracy is not a good performance measure when the proportions of the classes are imbalanced, as it tends to be biased toward the majority class (Murphy, 2022).

- **Precision**: The ratio of true positives out of all positive predictions. Precision is especially important when false positives must be avoided at all costs. The formula for precision

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{25}$$

- **Recall**: This represents the value of the actual positives correctly predicted by the model. Recall is preferred if minimising false negatives is a priority. Recall is given by:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{26}$$

- **F1-score**: This balances the precision and recall by taking a weighted average of the two. It is a useful metric when both precision and recall are considered to be equally important. F1 score is given by:

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{27}$$

- **AUC and ROC**: The receiver operating characteristic (ROC) is a two-dimensional plot that shows the true positive rate (recall) on the y-axis against false positive rate on the x-axis for all possible thresholds. The area under the ROC curve (AUC) quantifies the model's ability to distinguish between classes, where 1 indicates perfect performance and 0.5 is same as random guessing.

3.5. Experimental design

After preprocessing the data and finding optimal hyperparameters as outlined in Section 3.2, the next step is to fit the models on oversampled data using optimal hyperparameters and evaluate the performance on the test data. Figure 2 shows a flow chart for the entire modelling process covering preprocessing, hyperparameter tuning, performance evaluation as well as explainability techniques.

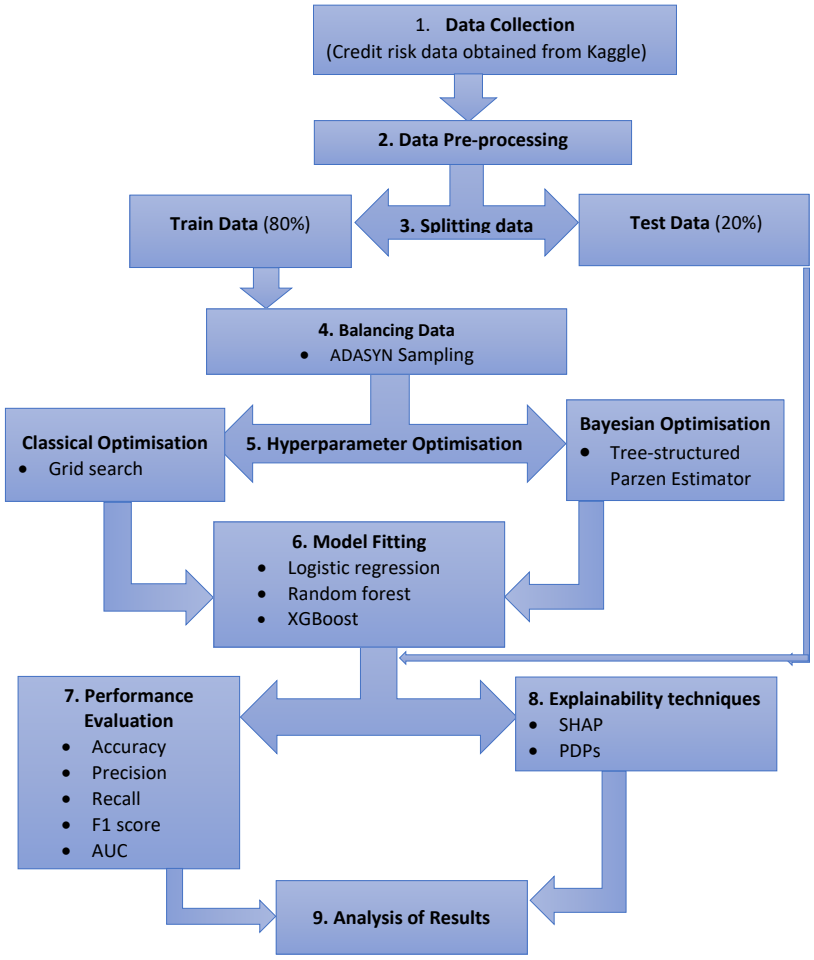

Figure 2. Experimental design flow chart.

4. Results and discussion

This section presents the performance results of our three models, XGBoost, random forest, and logistic regression, optimised with Bayesian and grid search techniques. The section also includes the model explainability techniques, SHAP and PDPs, for both the Bayesian and classical optimised models. The section then discusses the research findings and compares the results with the available literature.

4.1. Model performance measures

Figures 3 and 4 provide an overview of the performance of the XGBoost, random forest, and logistic regression classifiers. These classifiers were optimised using Bayesian and traditional grid search methods, focussing primarily on improving recall. Assessment metrics include accuracy, precision, recall, F1 score, and the area under the ROC curve (AUC), evaluated on the original test data set without applying oversampling techniques. The optimal hyperparameters are detailed in Appendix A.

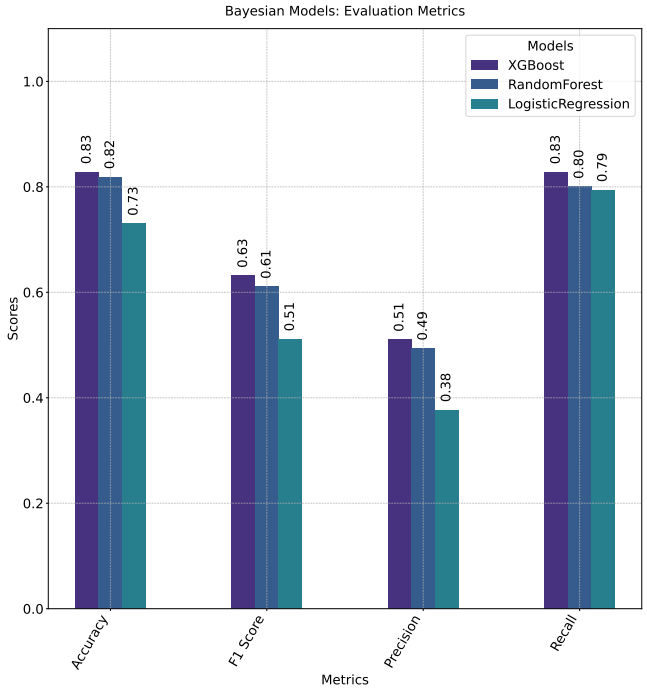

Figure 3. Bayesian models.

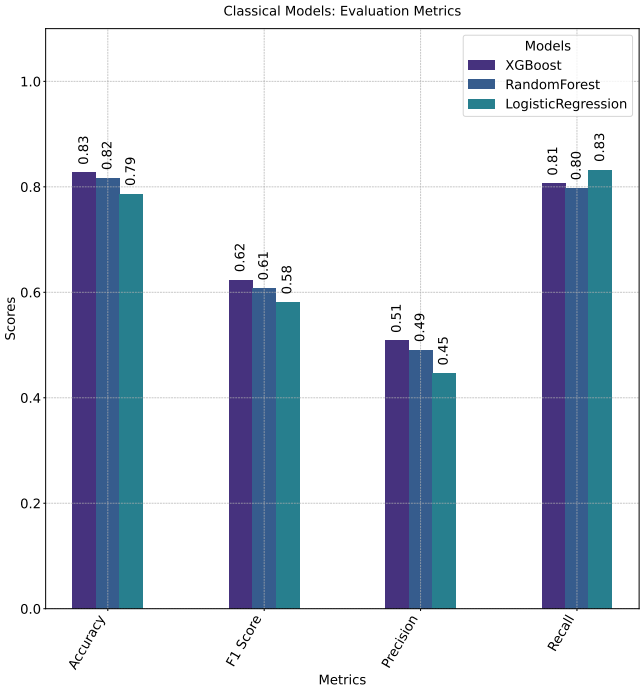

Figure 4. Classical models.

Figures 3 and 4 indicate a slight performance improvement in recall for XGBoost when using Bayesian hyperparameters, with a recall of 83% compared with 81% using grid search. This makes XGBoost the best-performing model for identifying defaulters, which is crucial for applications where minimising false negatives (i.e., missed defaulters) is a priority. The random forest model shows identical performance for both Bayesian and grid search optimised models, indicating that for the random forest, the Bayesian optimisation method does not necessarily offer a performance advantage over grid search. For logistic regression, applying Bayesian optimisation instead of grid search results in decreased accuracy (from 79% to 73%) and recall (from 83% to 79%). This suggests that grid search performs better, especially when correctly identifying defaulters.

Figures 5 and 6 below show the ROC curves and their AUC for the three models under the Bayesian and grid-search optimisation techniques.

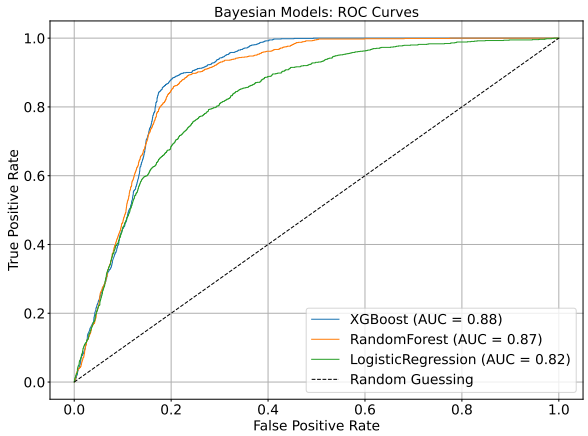

Figure 5. Bayesian models.

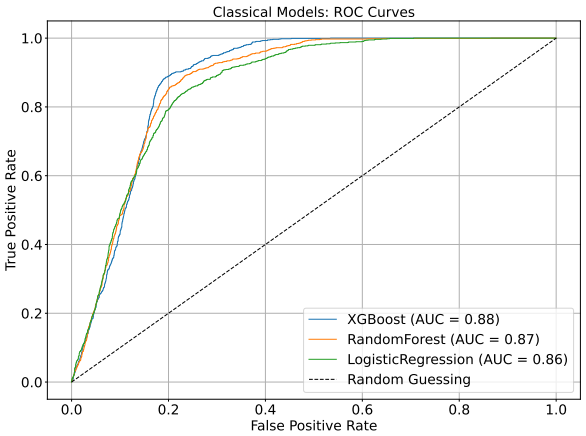

Figure 6. Classical models.

Comparing Figures 5 and 6, we can see that using the Bayesian optimisation method instead of the classical approach does not improve the discriminatory power of XGBoost and random forest, both of which maintain an 88% and 87% ability, respectively, to differentiate between positive (defaulters) and negative classes (non-defaulters). The logistic regression's ability to distinguish between defaulters and non-defaulters drops with the Bayesian approach with 82%, compared to 86% with the classical method.

4.2. Explainability results

SHAP offers global and local interpretability, providing insights into the overall behaviour of the model across the entire dataset (global explainability) and individual predictions (local explainability). PDPs illustrate how a particular feature influences the model's prediction across the dataset. Both SHAP values and PDPs were computed using the test dataset to ensure the evaluation of models' interpretability on unseen data.

4.3. Global Interpretations with SHAP

The SHAP summary plot illustrates the significance of each feature by showing its contribution to the model's predictions (default probability or log odds of defaulting), aggregated across the entire dataset. Each dot represents a Shapley value for a feature in a specific instance, with features ordered by their overall importance. Figures 7 – 12 display the summary plots for both the Bayesian-optimised and classical (grid search) models.

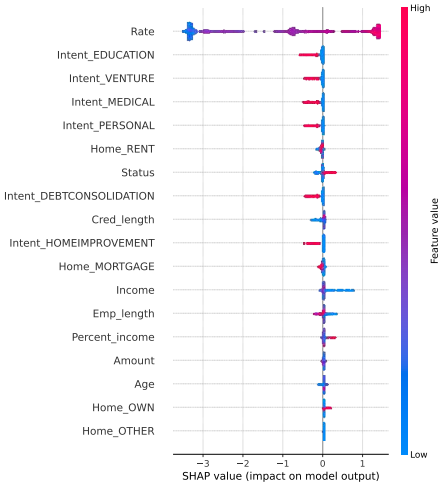

Figure 7. Bayesian optimised XGBoost.

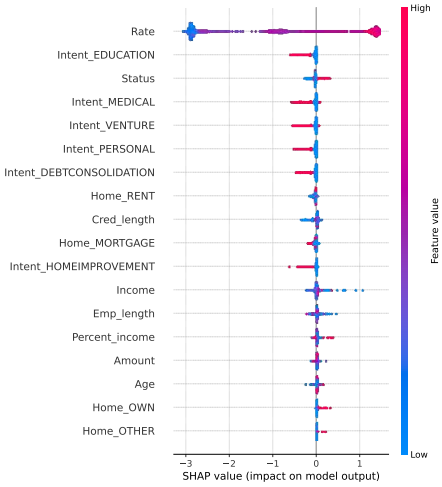

Figure 8. Classical optimised XGBoost.

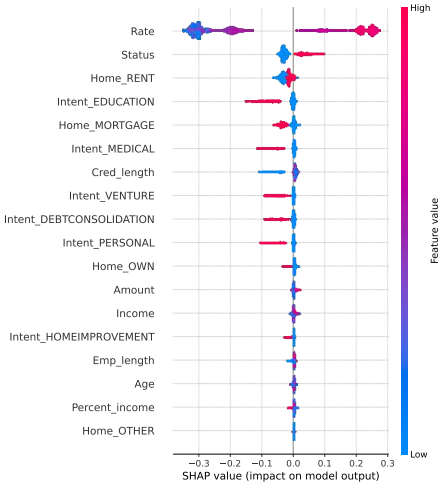

Figure 9. Bayesian optimised random forest.

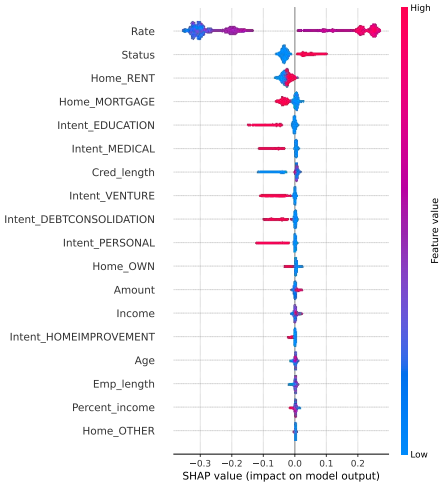

Figure 10. Classical optimised random forest.

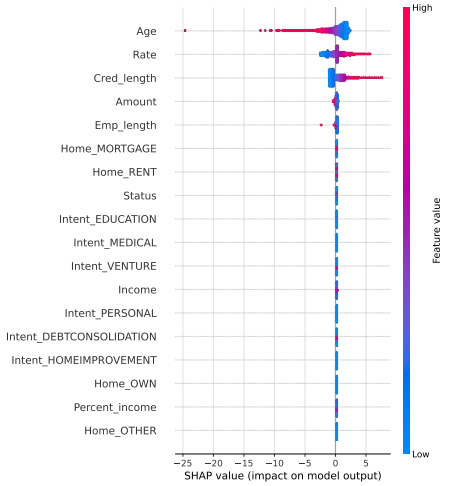

Figure 11. Bayesian optimised logistic regression.

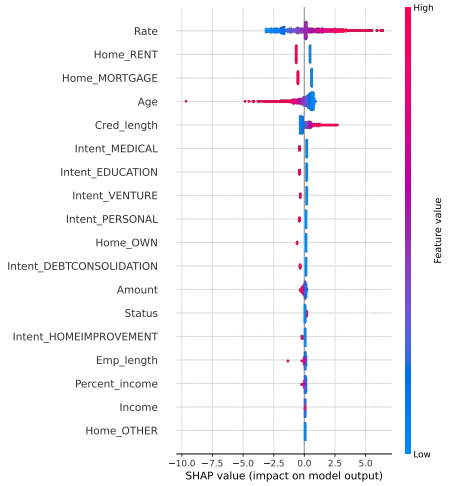

Figure 12. Classical optimised logistic regression.

Figures 7 and 8 show that the top five most important features identified by the Bayesian and classical optimised XGBoost models differ. The Bayesian optimised XGBoost identifies *Rate*, *Intent-purpose of the loan (education)*, *Intent-purpose of loan (venture)*, *Intent-purpose of loan (medical)*, and *Intent-purpose of loan (personal)* as the leading contributors to the prediction of default. In contrast, the classical optimised XGBoost model lists *Rate*, *Intent-purpose of loan (education)*, *Status*, *Intent-purpose of loan (medical)*, and *Intent-purpose of loan (venture)* among its top five most important features. While *Status* is ranked third by the classical approach, the Bayesian approach ranks it seventh. This change in the ranking of features demonstrates that different optimisation techniques can alter the order of importance of variables, even when similar features are selected.

The comparison of feature importance between the Bayesian and classical optimised random forest models, as illustrated in Figures 9 and 10, reveals similar patterns. Both models agree on the significant features, but their rankings vary. The Bayesian optimised random forest ranks *Intent-purpose of loan (mortgage)* as the fourth most important variable. In contrast, under classical optimisation, *Home-ownership status (mortgage)* is ranked as the fourth most important variable. This suggests that, while the same variables may consistently appear across models, the order of their importance can shift depending on the type of hyperparameter optimisation used.

The Bayesian optimised logistic regression model places greater emphasis on *Age*, *Rate*, and *Credit_length* as the most important variables. In contrast, the classical optimised logistic regression prioritises *Rate*, *Home-ownership status (rent)*, and *Home-ownership status (mortgage)*. This further demonstrates that even if hyperparameter tuning influences performance to a lesser extent (see Figures 7 to 12), hyperparameter tuning influences the importance of variables much more significantly, affecting how the models interpret the significance of certain features. The variations across models and optimisation methods suggest that the selection of hyperparameters is a crucial factor in shaping feature importance, and these differences should be considered when interpreting the models' results and their implications for decision-making.

4.4. Local Explainability with SHAP

SHAP waterfall plots offer a detailed explanation of each feature's input to a prediction made by a model for a particular instance, indicating the factors driving a prediction (such as why a borrower defaulted). The base value $E(f(x))$ is the average model prediction across all instances (mean log odds of default). Just like in the SHAP summary plots, the features are also ordered according to importance, from the most important to the least, and the left bar represents the feature values of a particular instance. Here, $f(x)$ is the model's prediction for this specific instance. Figures 13 - 18 show the SHAP waterfall plots for the first defaulting case in the test dataset, illustrating the feature breakdown for both the Bayesian optimised and classical optimised models.

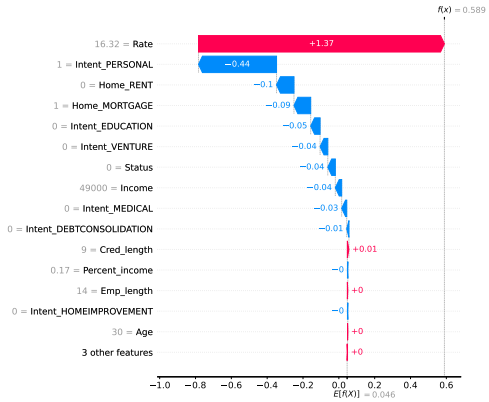

Figure 13. Bayesian optimised XGBoost.

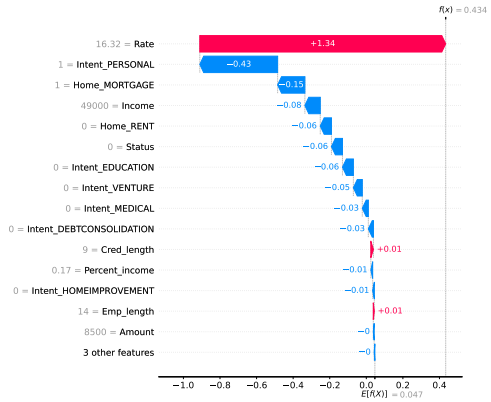

Figure 14. Classical optimised XGBoost.

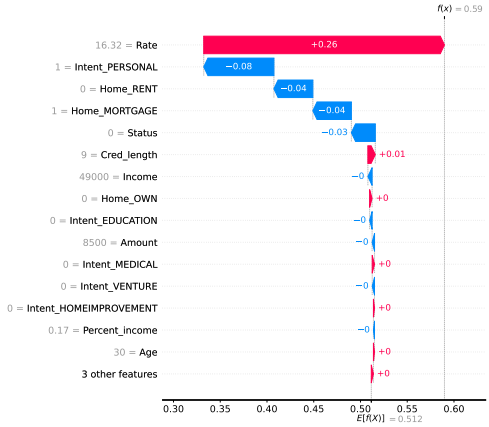

Figure 15. Bayesian optimised random forest.

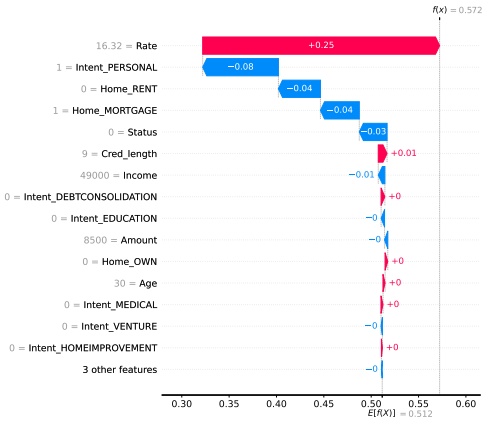

Figure 16. Classical optimised random forest.

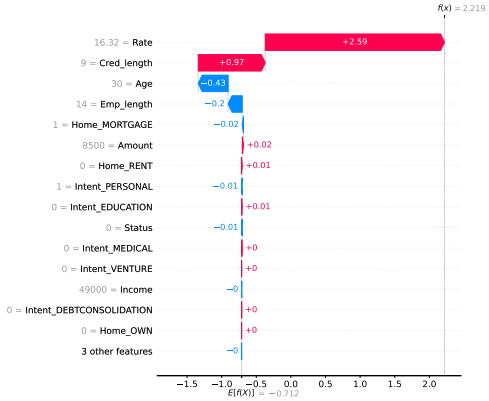

Figure 17. Bayesian optimised logistic regression.

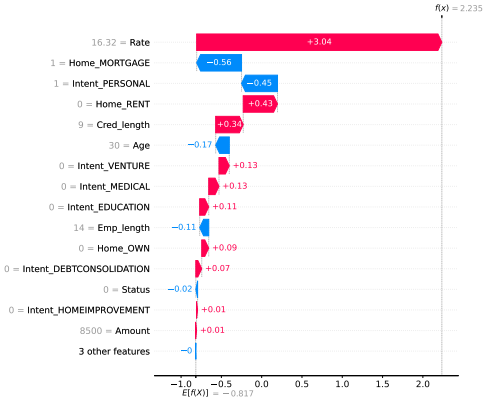

Figure 18. Classical optimised logistic regression.

Figures 13 and 14 show the SHAP waterfall plots for Bayesian and classical-optimised XGBoost models, respectively. The feature contribution hierarchy is consistent for the two most important features (*Rate* of 16.32 and *Intent-purpose of loan (personal)*) across both optimisation approaches. *Rate* with a value of 16.32 appears as the most important factor, increasing the likelihood of defaulting above average

and *Intent-purpose of the loan (personal)* lowers the prediction below the average in the both Bayesian and classical-optimised XGBoost. Bayesian optimised XGBoost considers *Home-ownership status (rent)* to be the third most important variable, while under the classical approach, *Home-ownership status (mortgage)* is the third most important variable. Concerning the model predictions $f(x)$, the Bayesian model estimates the log-odds of defaulting for this individual at 0.589, equating to a 0.643 probability of default, which surpasses the classical model's prediction of 0.434, corresponding to a 0.607 likelihood of default. This uniformity in the order of feature importance indicates that XGBoost does not substantially alter variable importance between the two optimisation methodologies.

Figure 15 reveals that the Bayesian optimised random forest identifies *Rate* of 16.32, *Intent-purpose of loan (personal)*, and *Home-ownership status (rent)* as the three most important drivers influencing default. Here, *Rate* of 16.32 pushes the default above the baseline, while other features push the prediction below the baseline. In contrast, Figure 16 shows that under the classical optimisation approach, *Rate* of 16.32, *Intent-purpose of loan (personal)*, and *Home-ownership status (rent)* are the top three contributors that influence default. *Rate* pushes default above the baseline while the other variables reduce it. As with XGBoost, the log-odds of default vary slightly between the two optimisation techniques (0.59 for Bayesian optimised random forest and 0.57 for the classical approach).

The Bayesian optimised logistic regression (Figure 17) highlights *Rate* of 16.32, Credit_length of 9, and *Age* of 30 as the top three significant variables affecting default, with *Rate* as the main contributor, followed by *Credit_length* of 9 both pushing the log odds of default above the baseline. However, *Age* of 30 reduces the prediction below the baseline. In contrast, classical-optimised logistic regression ranks *Home-ownership status (mortgage)* as the second significant variable pushing default, not *Credit_length*.

The variations observed in the waterfall plots suggest that the choice of the hyperparameter optimisation method influences individual predictions. The probability of defaulting for this customer ranged between 0.6 and 0.9 (refer to Table 6 in Appendix B). This customer ultimately defaulted, indicating that all models effectively estimated a high likelihood of default for this case.

4.5. Partial dependence plots

We computed partial dependence graphs for variables, *Rate* and *Age*, and *Credit_length* for Bayesian and classical optimised XGBoost, random forest, and logistic, based on the test dataset. Figures 19 – 24 below show the PDPs for *Rate* produced by the Bayesian and classical optimised models. The other PDPs for *Age* are shown in Appendix C.

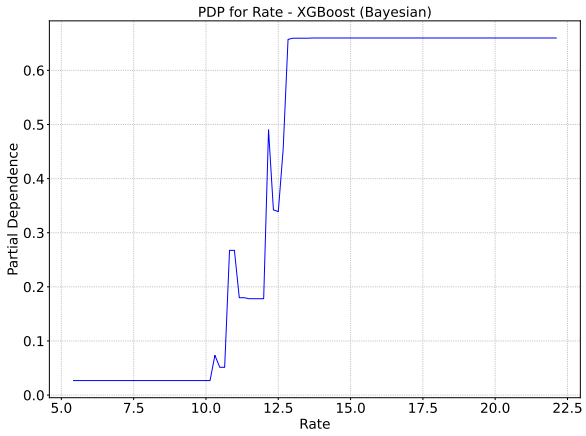

Figure 19. *Rate* under Bayesian optimised XGboost.

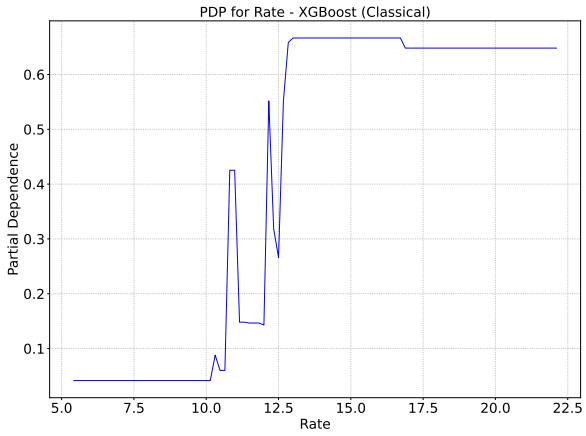

Figure 20. *Rate* under classical optimised XGboost.

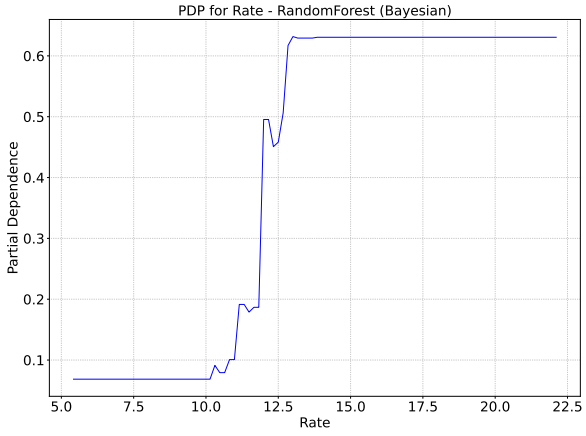

Figure 21. *Rate* under Bayesian optimised random forest.

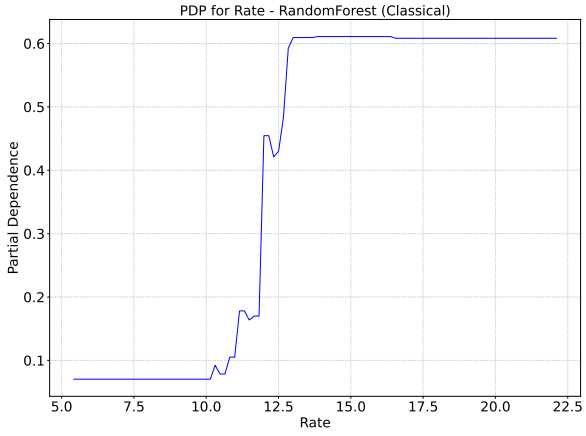

Figure 22. *Rate* under classical optimised random forest.

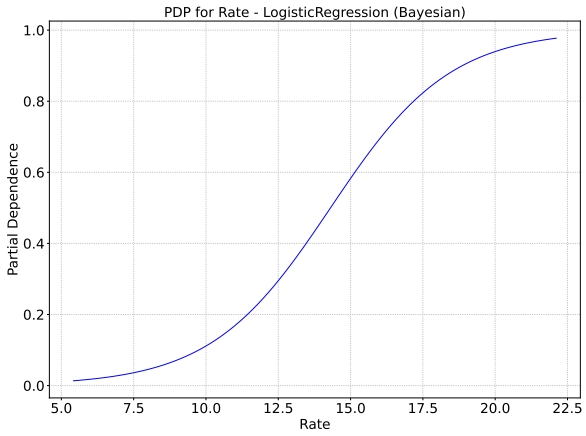

Figure 23. *Rate* under Bayesian optimised logistic regression.

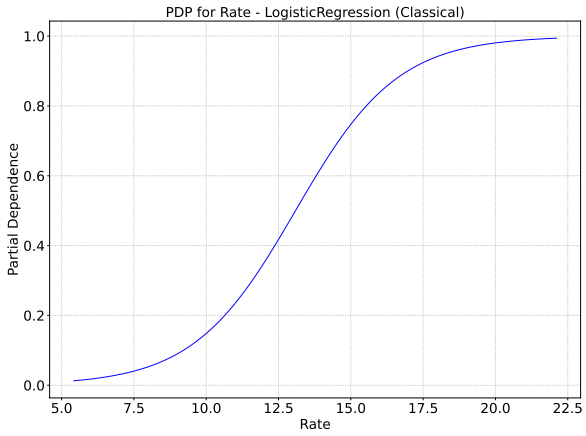

Figure 24. *Rate* under classical optimised logistic regression.

Figures 19 and 20 show the PDPs for the variable *Rate* under the Bayesian optimised XGBoost and classical optimised XGBoost, respectively. In both figures, the default probability sharply rises after a *Rate* of 10. It flattens and remains constant under the Bayesian optimised XGBoost. Still, under classical

optimised XGBoost, the probability of default drops slightly before the *Rate* of 17.5 and thereafter remains constant. The Bayesian model shows smoother changes in the probability of default with rate changes, which could indicate a more stable model with fewer sharp changes in feature importance. The classical-optimised XGBoost also shows more abrupt changes in the probability of default with changes in *Rate*, though the changes are not much different from those under Bayesian optimised XGBoost.

The PDP plots for the variable *Rate* under Bayesian optimised and classical optimised random forest, the PDPs for *Rate* are very similar and show that on average, the probability of a customer defaulting starts to increase after a *Rate* of around 10. Both optimisation techniques show a relatively smooth relationship between the variable *Rate* and the probability of default when applied to a random forest model. However, as with XGBoost, the Bayesian model results in a smoother and more stable constant relationship after *Rate* of 12.5, indicating that Bayesian optimisation stabilises the models' prediction. The minimal difference in performance indicates that Bayesian optimisation does not alter the relationship between *Rate* and the probability of default when using the random forest model.

For logistic regression, the PDPs for *Rate* under the Bayesian approach (Figure 11) and the classical approach (Figure 24) show that the probability of default continuously rises as *Rate* increases. The shape of the two graphs is almost the same, showing that the use of Bayesian hyperparameters does not alter the relationship between *Rate* and the probability of default.

4.6. Discussion of results

Our experiments demonstrated that XGBoost outperformed the other models when optimised using Bayesian methods, as shown by an increase in recall. This improvement may be due to the flexibility and probabilistic nature of Bayesian optimisation (the TPE approach), which favours more complex models such as XGBoost. Unlike grid search, which is deterministic, Bayesian optimisation combines prior knowledge (prior beliefs) with the data (likelihood) to find the next sampling point (posterior). Including prior information enhances XGBoost's performance, allowing it to capture more positive cases (higher recall).

The improvement in XGBoost's performance under Bayesian optimisation compared with grid search is consistent with the findings of Kong et al. (2023), who reported increased precision and recall when using Bayesian optimisation over grid search in credit scoring models. The authors found that Bayesian optimisation was preferable to classical approaches (grid and random search) for hyperparameter tuning, with additional benefits such as reduced computational time. However, we did not observe significant improvements in the performance of the random forest and logistic regression models. In contrast to other studies, such as PK et al. (2023), which recorded improved precision when using Bayesian optimised random forest, our results did not show substantial performance improvements. The limited performance gains in random forest and logistic regression reinforce the idea that Bayesian optimisation's benefits are more pronounced in models with more complex hyperparameter spaces, such as XGBoost. Furthermore, Bayesian optimisation is advantageous regarding computational efficiency, as it requires significantly less time to find the optimal hyperparameters than grid search. Overall, Bayesian optimisation reduces the computational time and improves recall performance while maintaining ROC performance similar to that of classical methods, making it preferable for complex models such as XGBoost.

The similarity in AUC scores between the models under classical optimisation suggests a degree of linear separability within the data. This implies that logistic regression, a simpler model, could be

adequate for this dataset if linear relationships dominate. For practitioners, this could mean that, in cases where the data demonstrate linearity, a linear model such as logistic regression may be preferable due to its interpretability and ease of implementation. This approach can improve models' transparency, which is often crucial in regulatory environments.

The SHAP summary and waterfall plots reveal the effect of different hyperparameter optimisation techniques on the global and local interpretability in credit risk modelling. These plots illustrate that the choice of optimisation method affects feature importance and contributions. This is particularly evident in logistic regression, where Bayesian optimisation ranks *Age* higher than *Rate*, the most important variable under the classical approach. This difference may be due to logistic regression's regularisation sensitivity (e.g., C). Bayesian optimisation likely fine-tunes the regularisation parameters more precisely, allowing *Age* to be ranked higher than *Rate*. In contrast, grid search, with fewer optimisation steps, may not adjust the regularisation as finely, causing features such as *Rate* and *Home-ownership status (rent)* to dominate. While Bayesian optimisation also influences feature contributions and importance in XGBoost and random forest, the effect is less significant than in logistic regression. In their paper, Kong et al. (2023) also found that the Bayesian optimisation method improves models' interpretability through SHAP analysis, but their study did not attribute the reasons for their results.

The PDP analysis further illustrates how hyperparameter tuning methods impact the relationship between the features and the predicted probability of default, especially in the XGBoost model. Bayesian optimised models produce smoother and more stable relationships between features and probability of default, as observed in the PDP for *Rate* under XGBoost (Figures 19 and 20). The smoother PDP produced by the Bayesian optimised XGBoost model (Figure 19) compared with the classical model (Figure 20) is probably a result of the TPE approach, which tunes the hyperparameters more precisely. In contrast, grid search evaluates pre-defined points in the hyperparameter space. It does not iterate toward a more optimal solution, which can lead to less stable model predictions and more abrupt changes, as seen in the classical optimised XGBoost model. Despite the differences in smoothness of the PDP curves, both the Bayesian and classical approaches produce PDPs that show a general increase in probability as *Rate* increases. This shows that using the Bayesian or classical optimisation method to find the optimal hyperparameters has the minimum effect on the relationship between *Rate* and probability of default.

The optimised logistic regression's PDPs for *Rate* suggest that logistic regression offers better practical explainability with sparse data through interpolation than the XGBoost and random forest models. The smooth transition in the predictions as *Rate* changes provides the bank with a more reliable estimate of risk at intermediate rate levels, which may be beneficial in understanding and managing customer segments with fewer data points. In contrast, the XGBoost and random forest models show more abrupt changes in the predictions, potentially reflecting overfitting in data-scarce regions.

Tools like SHAP plots, waterfall plots, and PDPs are very helpful for model developers and practitioners in credit risk modelling. They make models clearer and easier to understand, which is crucial in regulated fields like banking. For developers, these tools show how Bayesian optimisation fine-tunes the settings to boost performance, like improving recall while keeping the predictions stable. This helps them choose and improve models according to the data, such as whether they are linear or sparse. Practitioners benefit by understanding which features drive predictions, making it easier to explain models to regulators and stakeholders. For example, showing why *Age* or *Rate* matters under different optimisation methods builds trust and meets regulatory needs. Plus, Bayesian optimisation's

speed and clarity help practitioners build and use strong, clear models faster, improving decision-making and risk management in real-world situations. Moreover, these insights can help institutions maintain consistency in their risk management strategies even when updating or recalibrating machine learning models.

5. Conclusions

The main objective of this study was to investigate the impact of Bayesian and classical hyperparameter optimisation techniques on models' performance and explainability. Our findings demonstrate the importance of hyperparameter optimisation in directing models' performance and influencing models' explainability locally and globally, thus playing a role in credit risk modelling and decision-making.

Bayesian hyperparameter optimisation shows potential for improving models' performance, particularly for XGBoost, where it enhances recall. Improving recall is crucial in credit risk modelling, as identifying defaulters and minimising false negatives are a priority. However, Bayesian optimisation has minimal or even negative effects on the performance of random forest and logistic regression models, indicating that different models respond uniquely to hyperparameter optimisation techniques. While Bayesian optimisation improves recall for XGBoost, it does not necessarily improve the model's overall ability to distinguish between defaulters and non-defaulters, as measured by the AUC. This highlights an important consideration: Optimising one performance metric (such as recall) does not always lead to better discrimination between defaulters and non-defaulters. Therefore, the choice of optimisation approach should align with the priority performance metric, whether recall, precision, or AUC, depending on the specific goals of the credit risk model.

Examining the model's explainability through SHAP plots reveals that the choice of hyperparameter optimisation technique can affect the ranking of features (feature importance). We see this in logistic regression, where minor adjustments to hyperparameters due to the introduction of Bayesian optimisation induced significant variations in feature importance. Such findings have implications for models' interpretation and decision-making, as they indicate that optimisation strategies can directly influence the perceived determinants of default risk.

The findings also demonstrate that different hyperparameter optimisation techniques can change the perceived relationships between predictor variables and the target (default), as shown by the partial dependence plots (PDPs). The partial dependence plots for the same feature under different optimisation techniques are found to be different. This is especially relevant in credit risk models used in decision-making, where the ability to understand and trust the model's outputs is as important as the accuracy of its predictions.

One key advantage of Bayesian optimisation is its ability to substantially reduce the computational time required to identify optimal parameters compared with the traditional grid search approach. This efficiency is particularly valuable for large-scale modeling or scenarios demanding rapid results. However, these time savings must be carefully balanced against potential trade-offs, such as reduced interpretability or diminished discriminatory power, which could affect model performance. For instance, while a grid search might exhaustively explore the parameter space and take an entire day to complete (depending on dataset's size), Bayesian optimisation can often achieve comparable results in a fraction of the time, sometimes within minutes.

5.1. Recommendations for risk modellers

These findings are significant for credit risk modellers, highlighting the balance between models' performance, interpretability, and computational efficiency. Understanding how hyperparameter optimisation impacts different modelling aspects is crucial in an industry where decisions based on model outputs can have severe financial and regulatory consequences. Incorrectly optimised models could lead to misinterpreted feature importance, resulting in flawed risk management strategies or non-compliance with regulatory standards.

Therefore, this study emphasises that modellers should select the right optimisation approach. The choice should not be based on the assumption that advanced techniques like Bayesian optimisation will uniformly improve the models or evaluation metrics. It is necessary to understand the effect of the specific characteristics of each model to avoid suboptimal performance or misleading feature importance.

Moreover, the study emphasised that modellers have to balance performance and interpretability. In industries like finance, where understanding the rationale behind predictions is critical, modellers must carefully consider how optimisation affects both model performance and the transparency of its predictions.

The study emphasises the importance of evaluating trade-offs. While Bayesian optimisation offers computational efficiency, modellers should assess whether the speed advantage justifies potential trade-offs in performance or interpretability, ensuring that the chosen method aligns with their application's specific goals and constraints.

In conclusion, hyperparameter optimisation is an essential factor that shapes both performance and the interpretability and practical applicability of models in credit risk. Modellers must take a strategic approach, carefully considering how different optimisation techniques affect predictive accuracy, feature importance, and contributions to ensure that models are actionable, transparent, and aligned with industry standards. In cases where logistic regression performs the same as more complex models, such as XGBoost, modellers should consider prioritising the simpler approach. Logistic regression offers significant advantages in terms of interpretability, ease of deployment, and compliance with regulatory standards. Therefore, when data demonstrate linear relationships, logistic regression may provide a practical and transparent solution for credit risk modelling without sacrificing significant predictive power.

5.2. Limitations of current work

Despite the promising results, our study has several limitations. One key issue is the imbalanced nature of the data, which we mitigated using adaptive synthetic sampling. However, oversampling can introduce bias, and addressing this may lead to better outcomes. In the future, other class imbalance alternatives can be employed, such as undersampling to reduce the majority class, synthetic minority oversampling technique (SMOTE) to generate synthetic minority samples, class weighting to adjust model priorities, ensemble methods like balanced random forest (Agusta et al., 2019; Xie et al., 2009) or Tomek Links (Swana et al., 2022; Elhassan et al., 2016) to refine the decision boundary. Additionally, the specific impact of ADASYN on the models' performance, especially in terms of feature selection and models' interpretability, could be a future research idea.

Another limitation is our choice of models. We selected logistic regression, random forest, and XGBoost. The study limited its analysis to three models because of the large number of results, and the

experiment was designed to be easily replicated. Expanding this to include deep learning models such as deep neural networks could improve the results.

Additionally, our optimisation methods were limited to grid search and Bayesian optimisation. A broader comparison could be achieved by incorporating manual search, random search, heuristic methods, and genetic algorithms.

5.3. Future work

All the listed limitations could be seen as future research ideas. One potential direction is to address the imbalanced data issue further. Although the study used ADASYN, other techniques to correct for the bias introduced by oversampling can lead to better performance.

An additional area for future research would be replicating this study using different datasets or simulated data. This would validate the robustness of the findings and determine whether the conclusions hold across various datasets. By testing the model on a broader range of datasets or synthetically generated data, future studies could explore the generalisability of the results and uncover potential nuances that may arise in different contexts.

Another future avenue involves expanding the range of models used. In addition to the logistic regression, random forest, and XGBoost models used in this study, future work could investigate the impact of incorporating deep learning models, such as deep neural networks, for better comparison purposes.

Optimisation methods also present an opportunity for further expansion of this work. Beyond the grid search and Bayesian optimisation approaches we used, future studies could compare the effectiveness of manual search, random search, heuristic algorithms, and genetic search techniques.

Use of AI tools

The authors declare that they have not used artificial intelligence (AI) tools in the creation of this article.

Acknowledgments

I want to thank AIMS and its sponsors for supporting this work. I want to thank my supervisors, Dr Lindani Dube and Prof. Tanja Verster from North-West University for their guidance, support, knowledge sharing, and expertise. I want to thank the AIMS Academic Director, Prof. Karin, for her support and encouragement. This work is based on the research supported wholly/in part by the National Research Foundation of South Africa (Grant Number 126885).

Conflict of interest

All authors declare no conflicts of interest in this paper.

Code Availability

This project's Python codes and dataset are available at https://github.com/tatendashoko/AIMS-PROJECT.

References

Abhishek K, Abdelaziz M (2023) *Machine Learning for Imbalanced Data: Tackle Imbalanced Datasets Using Machine Learning and Deep Learning Techniques*. Packt Publishing Limited.

Agusta ZP (2019) Modified balanced random forest for improving imbalanced data prediction. *Int J Ad Intell Inf* 5: 58–65.

Ahmed Arafa AH, Radad M, Badawy MM, et al. (2022) Logistic regression hyperparameter optimization for cancer classification. *Menoufia J Electron Eng Res* 31: 1–8. https://doi.org/10.21608/mjeer.2021.70512.1034

Akiba T, Sano S, Yanase T, et al. (2019) Optuna: A next-generation hyperparameter optimization framework, In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2623–2631. https://doi.org/10.1145/3292500.333070

Alonso Robisco A, Carbo Martinez JM (2022) Measuring the model risk-adjusted performance of machine learning algorithms in credit default prediction. *Financ Innov* 8: 70. https://doi.org/10.1186/s40854-022-00366-1

Altman EI (1968) Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *J Financ* 23: 589–609. https://doi.org/10.2307/2978933

Antony TM, Suresh G (2023) Determinants of credit risk: Empirical evidence from Indian commercial banks. *Banks Bank Syst* 18: 88–100. https://doi.org/10.21511/bbs.18(2).2023.08

Basel Committee on Banking Supervision (1999) Principles for the management of credit risk.

Bentéjac C, Csörgő A, Martínez-Muñoz G (2021) A comparative analysis of gradient boosting algorithms. *Artif Intell Rev* 54: 1937–1967. https://doi.org/10.1007/s10462-020-09896-5

Bergstra J, Bardenet R, Bengio Y, et al. (2011) Algorithms for hyper-parameter optimization. *Adv Neur Info Process Syst* 24.

Bertrand A, Eagan JR, Maxwell W, et al. (2024) Ai is entering regulated territory: Understanding the supervisors' perspective for model justifiability in financial crime detection. In: *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1–21.

Breiman L (2001) Random forests. *Mach learn* 45: 5–32. https://doi.org/10.1023/A:1010933404324

Bussmann N, Giudici P, Marinelli D, et al. (2021) Explainable machine learning in credit risk management. *Computat Econ* 57: 203–216. https://doi.org/10.1007/s10614-020-10042-0

Caton S, Malisetty S, Haas C (2022) Impact of imputation strategies on fairness in machine learning. *J Artif Intell Res* 74: 1011–1035. https://doi.org/10.1613/jair.1.13197

Chen KS (2024) Interlinkages between bitcoin, green financial assets, oil, and emerging stock markets. *Data Sci Financ Econ* 4: 160–187. https://doi.org/10.3934/DSFE.2024006

Chen T, Guestrin C (2016) Xgboost: A Scalable Tree Boosting System, In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

De Lange PE, Melsom B, Vennerød CB, et al. (2022) Explainable ai for credit assessment in banks. *J Risk Financ Manage* 15: 556. https://doi.org/10.3390/jrfm15120556

Desalegn G, Tangl A, Boros A (2024) The mediating role of customer attitudes in the linkage between e-commerce and the digital economy. *Natl Account Rev* 6: 245–265. https://doi.org/10.3934/NAR.2024011

Du Toit HA, Schutte WD, Raubenheimer H (2024) Shapley values as an interpretability technique in credit scoring. *J Risk Model Validat*. https://doi.org/10.21314/JRMV.2023.010

Dube L, Verster T (2023) Enhancing classification performance in imbalanced datasets: A comparative analysis of machine learning models. *Data Sci Financ Econ* 3: 354–379. https://doi.org/10.3934/DSFE.2023021

Dube L, Verster T (2024) Interpretability of the random forest model under class imbalance. *Data Sci Financ Econ* 4: 446–468. https://doi.org/10.3934/DSFE.2024019

Elhassan T, Aljurf M (2016) Classification of imbalance data using tomek link (t-link) combined with random under-sampling (rus) as a data reduction method. *Global J Technol Optim S* 1: 2016. https://doi.org/10.4172/2229-8711.S1:111

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Ann Stat* 1189–1232.

Gatla TR (2023) Machine learning in credit risk assessment: Analyzing how machine learning models are transforming the assessment of credit risk for loans and credit cards. *J Emerg Techno Innov Res* 10: 746–750.

Gawde-Prabhudesai S, Patil S, Kamat P, et al. (2024) Explainable predictive maintenance of rotating machines using lime, shap, pdp, ice. *IEEE Access* 12: 29345–29361. https://doi.org/10.1109/ACCESS.2024.3367110

Hand DJ, Henley WE (1997) Statistical classification methods in consumer credit scoring: a review. *J R Stat Soc A* 160: 523–541. https://doi.org/10.1111/j.1467-985X.1997.00078.x

Hastie T, Tibshirani R, Friedman JH, et al. (2009) *The elements of statistical learning: data mining, inference, and prediction*, 2. Springer. https://doi.org/10.1007/978-0-387-21606-5

He H, Bai Y, Garcia E, et al. (2008) Adasyn: Adaptive synthetic sampling approach for imbalanced learning, In: *Proceedings of the International Joint Conference on Neural Networks*, 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969

Helmy A, Elnaghy S, Ramadan N (2023) Predicting unsettled debts in imbalanced data using resampling methods. In: *2023 Eleventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, 337–344. IEEE. https://doi.org/10.1109/10.1109/ICICIS58388.2023.10391162

Hu L, Chen J, Vaughan J, et al. (2020) Supervised machine learning techniques: An overview with applications to banking. *Int Stat Rev* 89: 573–604. https://doi.org/10.1111/insr.12448

Kaggle (2021) Loan applicant data for credit risk analysis dataset.

Khan MS, Peng T, Akhlaq H, et al. (2025) Comparative analysis of automated machine learning for hyperparameter optimization and explainable artificial intelligence models. *IEEE Access* 13: 84966–84991. https://doi.org/10.1109/ACCESS.2025.3566427

Kłosok M, Chlebus M (2020) *Towards better understanding of complex machine learning models using Explainable Artificial Intelligence (XAI): Case of Credit Scoring modelling*. University of Warsaw, Faculty of Economic Sciences Warsaw.

Kong Y, Wang Y, Sun S, et al. (2023) XGB and SHAP credit scoring model based on Bayesian optimization. *J Comput Electronic Inf Manage* 10: 46–53.

Li Z, Lai Q, He J (2024) Does digital technology enhance the global value chain position? *Borsa Istanb Rev* 24: 856–868. https://doi.org/10.1016/j.bir.2024.04.016

Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. *Adv Neural Inf Process Syst* 30.

Masís S (2021) *Interpretable Machine Learning with Python: Learn to build interpretable high-performance models with hands-on real-world examples*. Packt Publishing Ltd.

McNeil AJ, Frey R, Embrechts P (2015) *Quantitative Risk Management: Concepts, Techniques and Tools-Revised Edition*, Princeton University Press.

Molnar C (2020) *Interpretable Machine Learning*. Leanpub.

Murphy KP (2022) *Probabilistic Machine Learning: An introduction*. MIT Press.

Overisch M (2020) A conceptual explanation of Bayesian model-based hyperparameter optimization for machine learning.

Owen L (2022) *Hyperparameter Tuning with Python: Boost your machine learning model's performance via hyperparameter tuning*. Packt Publishing Ltd.

Pedregosa F, Varoquaux G, Gramfort A, et al. (2011) Scikit-learn: Machine Learning in Python. *J Mach Learn Res* 12: 2825–2830.

Rajesh PK, Shreyanth S, Sarveshwaran R (2023) Enhanced credit card fraud detection: A novel approach integrating Bayesian optimized random forest classifier with advanced feature analysis and real-time data adaptation. *Int J Innov Eng Manage Res* 12: 537–561. https://doi.org/10.48047/IJIEMR/V12/ISSUE05/52

Rodemann J, Croppi F, Arens P, et al. (2024) Explaining bayesian optimization by shapley values facilitates human-ai collaboration. *arXiv preprint*. https://doi.org/10.48550/arXiv.2403.04629

Rodríguez-Pérez R, Bajorath J (2019) Interpretation of compound activity predictions from complex machine learning models using local approximations and Shapley values. *J Med Chem* 63: 8761–8777. https://doi.org/10.1021/acs.jmedchem.9b01101

Shahriari B, Swersky K, Wang Z, et al. (2016) Taking the human out of the loop: A review of Bayesian Optimization. *P IEEE* 104: 148–175. https://doi.org/10.1109/JPROC.2015.2494218

Shapley LS (1953) *A value for n-person games*, Contributions to the Theory of Games 2, 28: 307–317.

Swana EF, Doorsamy W, Bokoro P (2022) Tomek link and smote approaches for machine fault classification with an imbalanced dataset. *Sensors* 22: 3246. https://doi.org/10.3390/s22093246

Wang Y, Ni XS (2019) A XGBoost risk model via feature selection and Bayesian hyper-parameter optimization. *arXiv preprint*. https://doi.org/10.48550/arXiv.1901.08433

Wu J, Chen XY, Zhang H, et al (2019) Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. *J Electron Sci Technol* 17: 26–40. https://doi.org/10.11989/JEST.1674-862X.80904120

Xia Y, Liu C, Li Y, et al. (2017) A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Syst Appl* 78: 225–241.

Xie Y, Li X, Ngai E, et al. (2009) Customer churn prediction using improved balanced random forests. *Expert Syst Appl* 36: 5445–5449. https://doi.org/10.1016/j.eswa.2008.06.121

Xu T (2024) Comparative Analysis of Machine Learning Algorithms for Consumer Credit Risk assessment. In: *7th International Conference on Computer Engineering, Information Science Application Technology* 4: 60–67. https://doi.org/10.62051/r1m3pg16

Yang J, Yin H (2022) Application of Bayesian optimization and stacking integration in personal credit delinquency prediction, In: *CS & IT Conference Proceedings*, 12.

Zharikova O, Pashchenko O, Smalyuh M (2023) Ensuring effective management of the credit portfolio of a commercial bank in the conditions the modern crisis. *Bioecon J* 14: 46–66.

Appendix

A. Hyperparameter Tuning Results

This section shows the hyperparameter tuning results obtained from Bayesian and classical approaches. Table 3 below shows the optimal parameters for the XGBoost model, the search space and the function of each hyperparameter.

Table 3. Hyperparameters, their descriptions, search space, and optimal values for XGBoost.

Hyperparameter	Description	Search Space	Optimal (Classical)	Optimal (Bayesian)
`learning_rate`	Contribution of each tree (step size)	0.01–2	0.029	0.05
`max_depth`	Depth of each decision tree	3 – 10	5	7
`n_estimators`	Number of decision trees trained	50–115	103	50
`subsample`	Observations used for each decision tree	0.8–1	0.824	0.9
`mean_child_weight`	Depth limit for each tree	1–10	5	1
`Gamma`	The smallest decrease in loss required to split at each node	0.005–1	0.54	0.05

Table 4 below shows the optimal parameters for the random forest model, search space, and the function of each hyperparameter.

Table 4. Hyperparameters, their descriptions, search space, and optimal values for random forest model.

Hyperparameter	Description	Search Space	Optimal (Classical)	Optimal (Bayesian)
`n_estimators`	Number of decision trees trained	50–115	57	50
`max_depth`	Depth limit of each decision tree	3–10	7	7
`min_sample_split`	Minimum number of samples required to split a node	2–10	3	3
`Criterion`	Splitting criteria to determine quality of split	`Gini, Entropy`	`Entropy`	`Entropy`

Table 5 below shows the optimal values of `C` for the logistic regression model and search space for both Bayesian and classical optimisation.

Table 5. Hyperparameters, their descriptions, search space, and optimal values logistic regression.

Hyperparameter	Purpose	Search Space	Optimal (Classical)	Optimal (Bayesian)
`C`	Inverse of regularisation (Ahmed Arafa et al., 2022)	0.01–10	7.9	2

B. Probability of Default Results

Table 6 below shows the PD estimates of the first defaulting case in the test data. The estimates are obtained by using Bayesian and classical-optimised models.

Table 6. PD estimates for Bayesian and classical models based on the first defaulting observation in the test dataset

Model	Log(odds)	Probability of Default
Bayesian XGBoost	0.589	0.643
Bayesian XGBoost	0.434	0.607
Bayesian Random Forest	0.590	0.643
Classical Random Forest	0.572	0.639
Bayesian Logistic Regression	2.219	0.902
Classical Logistic Regression	2.235	0.903

C. Partial dependence plots

Figures 25 - 30 below show the partial dependence plots for *Age* for the three classifiers under the Bayesian and classical approaches.

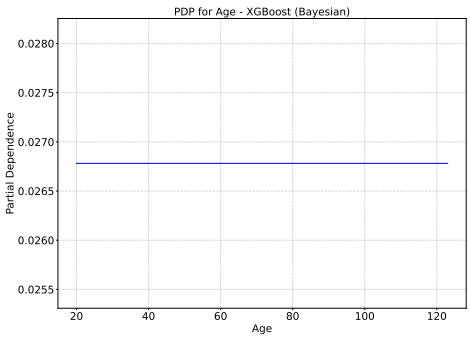

Figure 25. Bayesian-optimised XGBoost

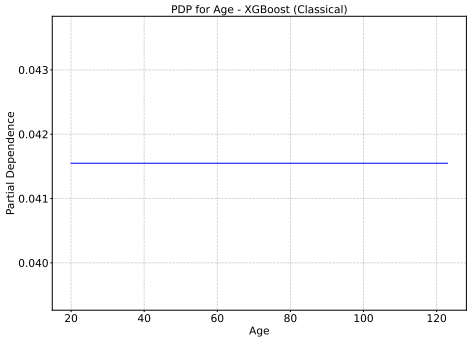

Figure 26. Classical-optimised XGBoost

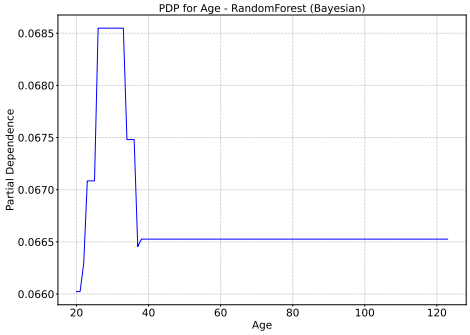

Figure 27. Bayesian-optimised random forest

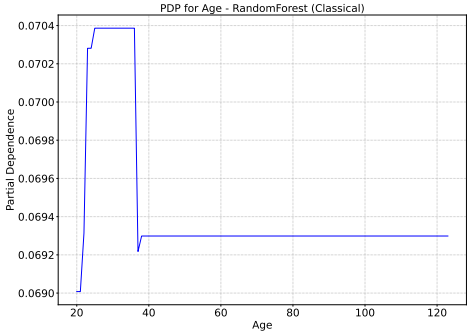

Figure 28. Classical-optimised random forest

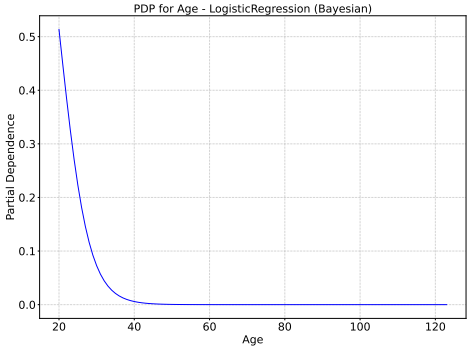

Figure 29. Bayesian-optimised logistic regression

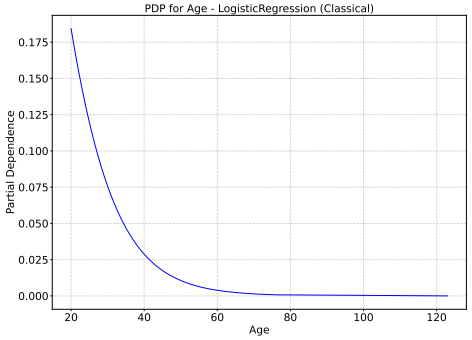

Figure 30. Classical-optimised Logistic Regression