Comparative Analysis of Classical and Bayesian Optimisation Techniques: Impact on Model Performance and Interpretability in Credit Risk Modelling Using SHAP and PDPs

Tatenda Shoko (shoko@aims.ac.za)
African Institute for Mathematical Sciences (AIMS)

Supervised by:
Dr. Lindani Dube
North-West University, South Africa
and
Prof. Tanja Verster
North-West University, South Africa

24 October 2024
Submitted in partial fulfillment of a structured masters degree at AIMS South Africa

Abstract

The performance, hyperparameter optimisation, and explainability of machine learning algorithms have been widely explored in the literature. However, there is limited research on how different hyperparameter optimisation techniques, such as classical methods (grid search, manual tuning, and random search) and more advanced Bayesian approaches, affect both model accuracy and interpretability. Financial institutions are often reluctant to use complex models such as random forest and extreme gradient boosting (XGBoost) to model credit risk due to difficulties in selecting optimal hyperparameters and the challenges of interpreting these 'black-box' models. This study addresses these two challenges by comparing the impact of classical (grid search) and Bayesian hyperparameter optimisation techniques on model performance and interpretability in credit risk prediction. The results show that Bayesian optimisation improves the recall for XGBoost, making it more effective at identifying defaulters while providing no significant benefit for the random forest and reducing the performance of logistic regression. Although Bayesian optimisation decreases the computational time required to find optimal hyperparameters, it does not improve the discriminatory power (AUC) of the models. Moreover, the findings of this study suggest that the choice of the optimisation technique affects feature importance and ranking of the features, with SHAP values showing that minor adjustments of the hyperparameters lead to notable changes in the importance of the features, particularly in logistic regression. However, the partial dependence plots (PDPs) for variable *Rate* under the Bayesian optimised models are similar to those produced by the classically optimised models, showing that the choice of the optimisation technique does not alter the relationship between features and probability of default. These findings highlight the importance of selecting the right optimisation approach to balance model performance and explainability, with significant implications for decision-making in credit risk modelling.

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

Declaration

I, the undersigned, hereby declare that the work contained in this research project is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

Tatenda Shoko, 24 October 2024

Contents

1. Introduction

1.1 Overview of Credit Risk Modelling

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). 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 primary function of credit is to channel the transfer of funds from savers to spenders. According to Zharikova et al. (2023), proper credit risk monitoring is necessary for economies to allocate funds efficiently. 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 essential in banking to estimate PD and make decisions about loan payments, lending rates, and risk management policies (Wang and Ni, 2019). Historically, the default probabilities have been calculated using statistical methods such as linear regression, logistic regression, and discriminant analysis (Altman, 1968; Hand and Henley, 1997). However, these methods often fail to capture complex and nonlinear relationships among variables in credit risk data (Gatla, 2023b). With recent advancements 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).

1.2 Problem Statement and Research Questions

Despite the advantages of random forest and XGBoost models mentioned in Section 1.1, banks are unwilling to use these models to develop credit risk models due to two main challenges: hyperparameter tuning and model interpretability (Xu, 2024). According to Owen (2022), hyperparameters such as learning rate, tree depth, and number of trees in XGBoost affect model performance, and therefore, finding the optimal values for these hyperparameters is necessary to obtain maximal model performance without overfitting.

Another drawback of random forest and XGBoost is that the models are not interpretable. Random forest, XGBoost, and deep learning models are often described as 'black box', 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 transformations inside the black box function occur to convert inputs into outputs (Masís, 2021). Transparency is essential in risk modelling, as it allows stakeholders to understand and justify the predictions made by a machine learning model (Bertrand et al., 2024).

Several studies have explored the impact of hyperparameter optimisation (tuning) 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 effects of the choice of hyperparameter optimisation techniques on both model performance and interpretability, especially in credit risk

modelling. This work addresses this gap by comparing two hyperparameter tuning techniques: Bayesian optimisation and grid search. It evaluates the impact of these techniques on model performance and on model explainability using SHapley Additive exPlanations (SHAP) and partial dependence plots (PDPs).

The following research questions will be addressed in this study:

- How does the performance of Bayesian-optimised models differ from classically (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 relationships between features and probability of default (log-odds of defaulting) introduced by classical and Bayesian optimisation techniques?

1.3 Aim and Objectives

This work aims to assess how Bayesian and classical hyperparameter optimisation techniques impact model performance and interpretability in 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 feature importance and rankings.
- To use PDPs to determine whether different optimisation techniques change the relationships between features and default probabilities.

1.4 Significance of the Study

Studying the impact of the choice of hyperparameter optimisation techniques on performance and model interpretability is vital for credit risk modellers in banks to build accurate and justifiable models for decision-making purposes regarding loan approvals, setting interest rates on loans, and planning for proper risk management strategies. This study guides the selection of hyperparameter optimisation methods for efficient and understandable credit risk modelling. The study might be used for further academic research and literature in predictive modelling, machine learning, and explainable machine learning.

1.5 Structure of the Thesis

Chapter 2 presents related work and a detailed discussion of the machine learning algorithms, optimisation techniques, and explainability techniques used in this study. Chapter 3 describes the data, the preprocessing steps, and the experimental setup. The chapter also discusses evaluation metrics used to compare the performance of the optimised models. Chapter 4 presents model performance and explainability results and provides a detailed discussion of the findings. Chapter 5 presents the conclusion, the limitations of our analysis, possible recommendations to risk modellers, and potential areas for future

research. The appendices provide information about hyperparameters, partial dependence plots, and probabilities.

2. Review of Literature

This chapter reviews related work on hyperparameter optimisation and model interpretability of machine learning models. The chapter also includes a detailed discussion of grid search and Bayesian optimisation methods, machine learning algorithms, and interpretable machine learning techniques used in this study.

2.1 Related Work

This section discusses previous research where hyperparameter optimisation has been used to improve the performance of machine learning models. The section also discusses previous literature on the explainability of machine learning models in credit risk modelling and other domains.

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. The authors successfully compared the accuracy, error rate, and AUC of Bayesian-optimised XGBoost to classical-optimised ones (random, grid, and manual search). Their findings show that Bayesian hyperparameter tuning performs better than classical search methods in accuracy, error rate, and AUC. The authors highlighted that Bayesian-optimised XGBoost also provided interpretable feature importances, bridging the gap between performance and transparency.

In their paper, Alonso Robisco and Carbo Martinez (2022) evaluated the impact of 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 logistic regression in accuracy, AUC, recall, and F1 score. These findings reinforce the trade-off between model complexity and performance that often arises when using advanced machine learning models in credit risk estimation. In a similar study, Wang and Ni (2019) also compared the random search and the Bayesian tree-structured Parzen estimator (TPE) optimisation methods on the performance of XGBoost and logistic regression. The authors also recorded that the Bayesian TPE-optimised XGBoost outperforms logistic regression in accuracy, AUC, and F1 score. Their discovery indicates the need for hyperparameter optimisation to improve model performance.

Yang et al. (2022) also applied Bayesian optimisation to find the best parameters for XGBoost, random forest, and GBDT (gradient boosting decision trees) in personal credit delinquency prediction. Their study successfully applied Bayesian optimisation to improve the performance of these three models, recording an AUC of 0.92 for the optimised random forest, 0.94 for GBDT, and 0.95 for XGBoost. These findings show that hyperparameter optimisation improves the discriminatory power of the models. The authors also recorded that Bayesian optimisation takes less time to find the optimal hyperparameters, particularly credit risk, where computational efficiency can be a significant concern.

The work by Kong et al. (2023) used Bayesian optimisation to find the best parameters for XGBoost in credit scoring. The authors successfully implemented Bayesian optimisation and recorded an AUC value of 0.95 for Bayesian-optimised XGBoost, significantly above the commonly used logistic regression with an AUC of 0.80 and gradient-boosting decision trees with an AUC of 0.92. Their findings agree with Yang et al. (2022), proving that Bayesian optimisation improves the discriminatory power of the XGBoost model. To address the issue of model transparency, the authors applied SHAP to explain the prediction results from Bayesian optimised XGBoost. By using SHAP summary, decision, waterfall, and force plots, the authors demonstrated that Bayesian optimisation also improves the interpretability of the XGB model.

In addition to accuracy, financial institutions require models that are also interpretable for decision-making purposes. In their paper, 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 using explainability techniques such as Shapley values to improve model transparency, particularly for stakeholders who require information on which features most strongly influence credit default predictions. The study also explored the gap between traditional interpretable models, such as logistic regression, and more complex machine learning models, such as XGBoost, expressing the need for interpretability-focused research in credit risk modelling.

In bank churn predictions, Dube and Verster (2024) also applied Shapley values, breakdown plots and PDPs to address the interpretability of random forest under class imbalance. Their findings suggest that Shapley values are necessary for showing changes in feature importance as class imbalance changes, while the PDPs showed a consistent relationship between features as the level of class imbalance changes. Their study emphasises the need to use multiple explanability techniques to understand relationships between features and target variables.

Kłosok et al. (2020) also the solved issue of the need for explainability in credit risk models by using PDPs, feature importance, accumulated local effects, individual conditional expectation curves, accumulated local effects, and Shapley values to interpret the predictions of the random forest model. Their findings demonstrate that all the explainability tools they applied produce reliable explanations, leading to transparency and accountability of credit risk systems.

The present work aims to assess the impact of Bayesian and classical hyperparameter optimisation techniques on the performance of XGBoost, random forest, and logistic regression models, as well as on the interpretability of these models when applied to credit risk data. The best hyperparameters obtained from these methods are used to build classification models for credit risk data. Moreover, SHAP and PDPs are used to analyse the effects of Bayesian and classical optimisation techniques to find optimal hyperparameters.

2.2 Machine Learning Models

This section discusses the machine learning models used in this study and their associated hyperparameters, which were tuned to enhance model performance.

This study used three classifiers: logistic regression, random forest, and XGBoost. Logistic regression has been adopted in this study because it is widely used in PD modelling due to its simplicity and interpretability of its coefficients (Gatla, 2023b). The study used the random forest model due to its ability to handle large data and well as its ability to work with incomplete (missing) data (Wu et al., 2019). XGBoost model has been chosen due to its great performance in speed, memory usage sensitivity, and multiple hyperparameters, making it suitable for classification purposes (Chen and Guestrin, 2016).

2.2.1 Logistic Regression

Logistic regression is a widely used classification algorithm in credit risk modelling to predict the probability that a customer will default or not, given a set of explanatory variables (Bussmann et al., 2021). Suppose $\mathbf{x}_i = \{x_{i1}, x_{i2}, \ldots, x_{ip}\}$ is a set of explanatory variables and $\mathbf{y}_i \in \{0, 1\}$ is a binary target set representing default (1) and no default (0), then log-odds of the probability of default are given by a linear combination of explanatory variables as shown in Equation 2.2.1 below (Murphy, 2022):

$$\log \frac{P(y_i = 1)}{1 - P(y_i = 1)} = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}, \tag{2.2.1}$$

where β_0 is the intercept term, and β_j's are the feature coefficients obtained using maximum likelihood estimation to maximise the probability of observed data (Yang et al., 2022). Each coefficient β_j represents the effect of a unit change in x_{ij} on the logarithmic odds of default. The probability of default ($P(y_i = 1)$) is obtained by transforming Equation 2.2.1 into Equation 2.2.2:

$$P(y_i = 1 \mid \mathbf{x}_i) = \frac{e^{\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}}}{1 + e^{\beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j}}}. \tag{2.2.2}$$

The logistic regression model was trained using the `LogisticRegression` class from the `scikit-learn` library in this work. The logistic regression model does not have many hyperparameters. In this work, only the inverse of the regularisation `C`, which controls the regularisation strength, was tuned (Pedregosa et al., 2011a). More details about the search space and the optimal values for `C` are given in in Appendix A, Table A.3

2.2.2 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 based on 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:

$$\hat{f}(x) = \sum_{m=1}^{M} c_m I(x \in R_m), \tag{2.2.3}$$

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:

$$\text{Gini impurity} = \sum_{k=0}^{1} p_{mk}(1 - p_{mk}), \tag{2.2.4}$$

$$\text{Cross-Entropy} = -\sum_{k=0}^{1} p_{mk} \log(p_{mk}), \tag{2.2.5}$$

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 (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:

$$\hat{C}_{\text{rf}}^{B}(x) = \text{majority vote}\{\hat{C}_b(x_i)\}_1^B, \tag{2.2.6}$$

where $\hat{C}_{\text{rf}}^{B}(x)$ 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. The main hyperparameters considered are the number of decision trees (B), the maximum depth of each tree, and the splitting criteria (Gini or entropy). Detailed information on the hyperparameters considered, their functions, the search space, and the optimal hyperparameters obtained are found in Appendix A in Table A.2.

2.2.3 Extreme Gradient Boosting (XGBoost)

XGBoost (Chen and Guestrin, 2016), is a machine learning algorithm designed as an improvement over 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 trees algorithm is that the XGBoost has 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:

$$\hat{f}(x_i) = \sum_{b=1}^{B} f_b(x_i), \tag{2.2.7}$$

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 and actual values. The loss function is given by:

$$L_b = \sum_{i=1}^{n} l(y_i, \hat{y}_i), \tag{2.2.8}$$

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

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

$$L_b = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{b=1}^{B} \Omega(f_b). \tag{2.2.9}$$

The regularisation term is defined as:

$$\Omega(f_b) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \tag{2.2.10}$$

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, their functions and their optimal values are provided in Table A.1 in Appendix A.

2.3 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):

$$x^\star = \arg\max_{x \in \mathcal{X}} f(x), \tag{2.3.1}$$

where $f(x)$ is the objective score to be maximised (recall in this study) on 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 hyperparameters: Bayesian optimisation and grid search. These methods will be applied to find the optimal hyperparameters for XGBoost, random forest and logistic regression models.

2.3.1 Bayesian Hyperparameter Optimisation

This subsection outlines the Bayesian optimisation method employed in this study. Initially, Bayes' Theorem is introduced, followed by a general explanation of how Bayesian optimisation operates. Subsequently, the focus shifts to the specific variant of Bayesian optimisation used in this work: the tree-structured Parzen estimator (TPE) approach.

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

Bayesian optimisation is a method designed to improve the performance of the objective function by applying the probabilistic model of the objective function. It then uses this model to select hyperparameters to choose the hyperparameters that evaluate the true objective function. Bayesian optimisation algorithm is an application of the Bayes Theorem to search for the maximum of the objective function.

The Bayes' Theorem is given in Equation 2.3.2:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}, \tag{2.3.2}$$

where $P(y|x)$ represents posterior distribution, $P(x|y)$ the likelihood, $P(y)$ the prior term and $P(x)$ is a normalizing 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 2.3.3 below:

$$D = \{(x_1, f(x_1)), (x_2, f(x_2)), ..., (x_n, f(x_n))\}. \tag{2.3.3}$$

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:

$$p(f|D) = P(D|f) \cdot P(f). \tag{2.3.4}$$

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 initial sampling, Bayesian optimisation uses the likelihood term in Equation 2.3.4 to estimate the objective function to form a surrogate model (the probability representation of the objective function), then uses the surrogate model to select 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 continuous and categorical hyperparameters (Owen, 2022).

The TPE approach uses the Bayes Theorem in Equation 2.3.2 to model the probability distribution of hyperparameters given the objective function, i.e., $p(x|y)$. This approach selects a threshold y^* to be some quantile γ of the observed y. The distributions of hyperparameters can be modelled as follows:

$$p(x|y) = \begin{cases} l(x) & \text{if } y < y^* \\ g(x) & \text{if } y \geq y^*, \end{cases} \tag{2.3.5}$$

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 2.3.5. 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 used acquisition functions include expected improvement (EI) and probability of improvement (PI). This study used EI because it can efficiently balance exploitation and exploration (Bergstra et al., 2011). The EI under the TPE approach is given by:

$$EI_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y) p(y|x) dy \tag{2.3.6}$$

$$= \int_{-\infty}^{y^*} (y^* - y) \frac{p(x|y)p(y)}{p(x)} dy. \tag{2.3.7}$$

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 2.3.7, we obtain the following:

$$EI_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y) \frac{p(x|y)p(y)}{\gamma l(x) + (1-\gamma)g(x)} dy. \tag{2.3.8}$$

Optimising EI means focussing on the "good" region of the objective function where $y \leq y^*$. Substituting $p(x|y) = l(x)$ from Equation 2.3.5 into Equation 2.3.8:

$$EI_{y^\star}(x) = l(x) \int_{-\infty}^{y^*} (y^* - y) \frac{p(y)}{\gamma l(x) + (1-\gamma) g(x)} dy. \tag{2.3.9}$$

Equation 2.3.9 can be simplified further into Equation 2.3.10

$$EI_{y^\star}(x) = \frac{\gamma y^* l(x) - l(x) \int_{-\infty}^{y^*} p(y) dy}{\gamma l(x) + (1-\gamma) g(x)} \propto \left(\gamma + \frac{g(x)}{l(x)}(1-\gamma)\right)^{-1}. \tag{2.3.10}$$

From the last expression of Equation 2.3.10, $EI_{y^\star}(x) \propto \frac{l(x)}{g(x)}$. to achieve high Expected Improvement (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 $g(x)$ 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.

2.3.2 Grid Search

Grid search is a traditional (classical) exhaustive approach that explores every possible combination of hyperparameters. The method involves training the machine learning model with each possible hyperparameter configuration on the training dataset (oversampled in this study) and evaluating its performance using a predetermined metric on a test dataset (Bentéjac et al., 2021).

The grid search process operates 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, record the cross-validation scores along with the best parameter combinations.
- Retrain the model with optimal hyperparameters.
- Assess the model performance on the test dataset.

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

2.4 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 the model predictions but that those decisions must be ethical and human-friendly. Thus, interpretable machine learning means getting useful information from a machine learning model about the relationships, feature importances, and contributions in the given data or learnt by the model from the data 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.

2.4.1 SHapley Additive exPlanations (SHAP)

SHAP is a game-theoretic approach to interpreting the machine learning model's 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 (Molnar, 2020):

$$\phi_j = \frac{1}{|N|!} \sum_{S \subseteq N \setminus \{j\}} |S|!(|N| - |S| - 1)![f(S \cup \{j\}) - f(S)], \tag{2.4.1}$$

where N is a set of consisting of all features, S is a subset N without feature of interest j, $f(S)$ is the model' output when only the features in S are considered, and $f(S \cup \{j\})$ is the output when adding the feature j to subset S.

As noted by Rodríguez-Pérez and Bajorath (2019), Shapley values fairly distribute feature importance for a given prediction. Positive values indicate a positive contribution, negative values indicate a negative contribution, and zero indicates no impact.

In this study, we used Python's `SHAP` library, which provides model-agnostic explainers to visualise visualize feature importance and contribution on the test dataset. For local explanations, SHAP calculates each variable's contribution to an individual prediction, while global explanations can be obtained by aggregating these contributions. For tree-based models (XGBoost and random forest), the study used SHAP's `TreeExplainer`, and for logistic regression models, `LinearExplainer` is used.

2.4.2 Partial Dependence Plots (PDPs)

A PDP (Friedman, 2001) is a global explainability tool that shows how the model 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 acknowledged 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 model output. In that case, the partial dependence function on the model's prediction function $\hat{f}(X)$ on X_S is defined as:

$$f_{\text{pd}}(X_S) = \mathbb{E}_{X_C}[\hat{f}(X_S, X_C)] = \int \hat{f}(X_S, X_C)\, dP(X_C), \tag{2.4.2}$$

where $X_C = X \setminus X_S$ represents the complement 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 2.4.2 is approximated by averaging over the observed values of X_C. The estimated partial dependence function is then expressed as:

$$\widehat{f_{\text{pd}}(X_S)} = \frac{1}{n} \sum_{i=1}^{n} \hat{f}(X_S, x_{iC}), \tag{2.4.3}$$

where x_{iC} denotes the observed values of X_C for the i-th observation. This expression shows that the partial dependence function averages over the values of X_C while keeping the values of X_S constant.

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 2.4.3 (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.

3. Materials and Methods

This chapter discusses the methods that we followed in this study. Section 3.1 discusses the credit risk dataset, Section 3.2 details the preprocessing steps, Section 3.3 discusses the evaluation metrics to compare model performance, and Section 3.4 gives the experimental design.

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 3.1 below.

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

Table 3.1: Loan Applicant Data for Credit Risk Analysis dataset (see Kaggle (2021)).

The dependent variable in this dataset is *Default*, which indicates whether the applicant has defaulted on previous loans. Other variables given in Table 3.1 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 categorical variables were introduced 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 which can handle missing data, but data cleaning and preprocessing steps were applied across all models to ensure fair comparison of results.

As discussed in Section 2.2, random forest and XGBoost are robust models which can handle missing data, but data cleaning and preprocessing steps were applied across all models to ensure fair comparison of results.

Figure 3.1 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 classifiers tend to capture non-defaulters but fail to capture defaulters.

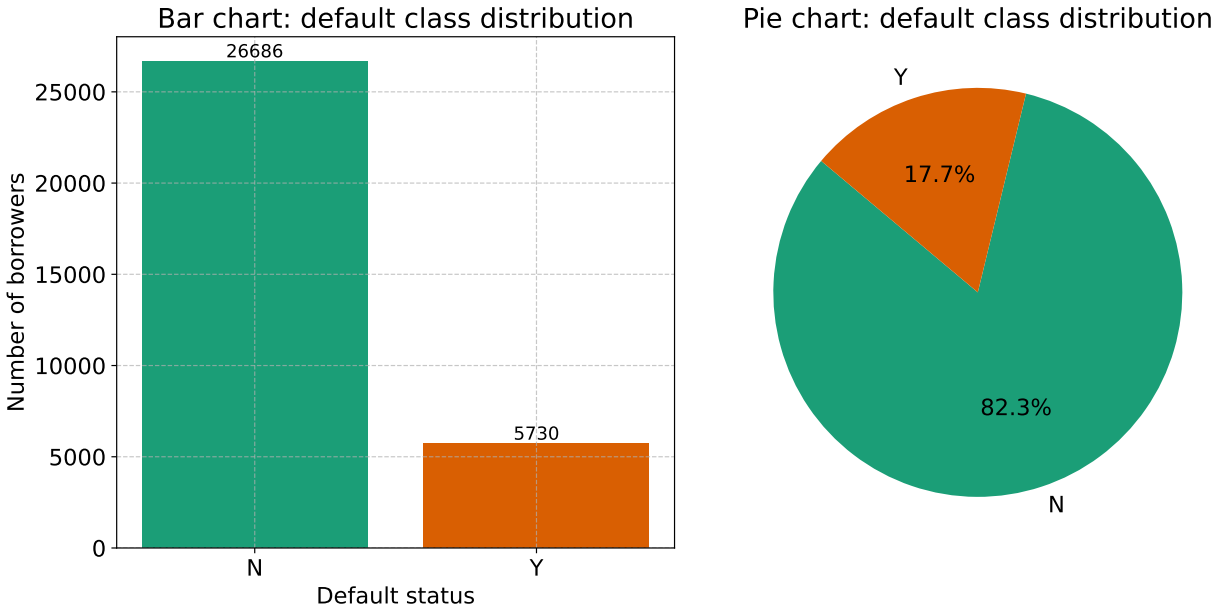


Figure 3.1: Default status: class distribution

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 train and test sets using the `train_test_split` function from Python's `scikit-learn` library. This function performs a random shuffle to ensure an unbiased partition. 80% of the data was used for training, and the remaining 20% was set aside to evaluate model performance.

3.2.1 Searching for Optimal Hyperparameters

The study used `Optuna`, an open-source Python library, to perform Bayesian optimisation and identify optimal hyperparameters for 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 `scikit-learn` library, was used. In both 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 results.

More information on hyperparameter functions, domain space and optimal values for XGBoost, random forest, and logistic regression are given in Tables A.1, A.2 and A.3 respectively in Appendix A.

3.3 Model Evaluation Metrics

The study accessed the classification metrics values to compare the performance of the models optimised using 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:
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{3.3.1}$$
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 towards 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:
$$\text{Precision} = \frac{TP}{TP + FP}. \tag{3.3.2}$$
- **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:
$$\text{Recall} = \frac{TP}{TP + FN}. \tag{3.3.3}$$
- **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:
$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3.3.4}$$
- **AUC and ROC**: The receiver operating characteristics (ROC) is a 2D plot that shows 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.

3.4 Experimental Design

After preprocessing 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 3.2 shows a flow chart for the entire modelling process covering preprocessing, hyperparameter tuning, performance evaluation as well as explainability techniques.

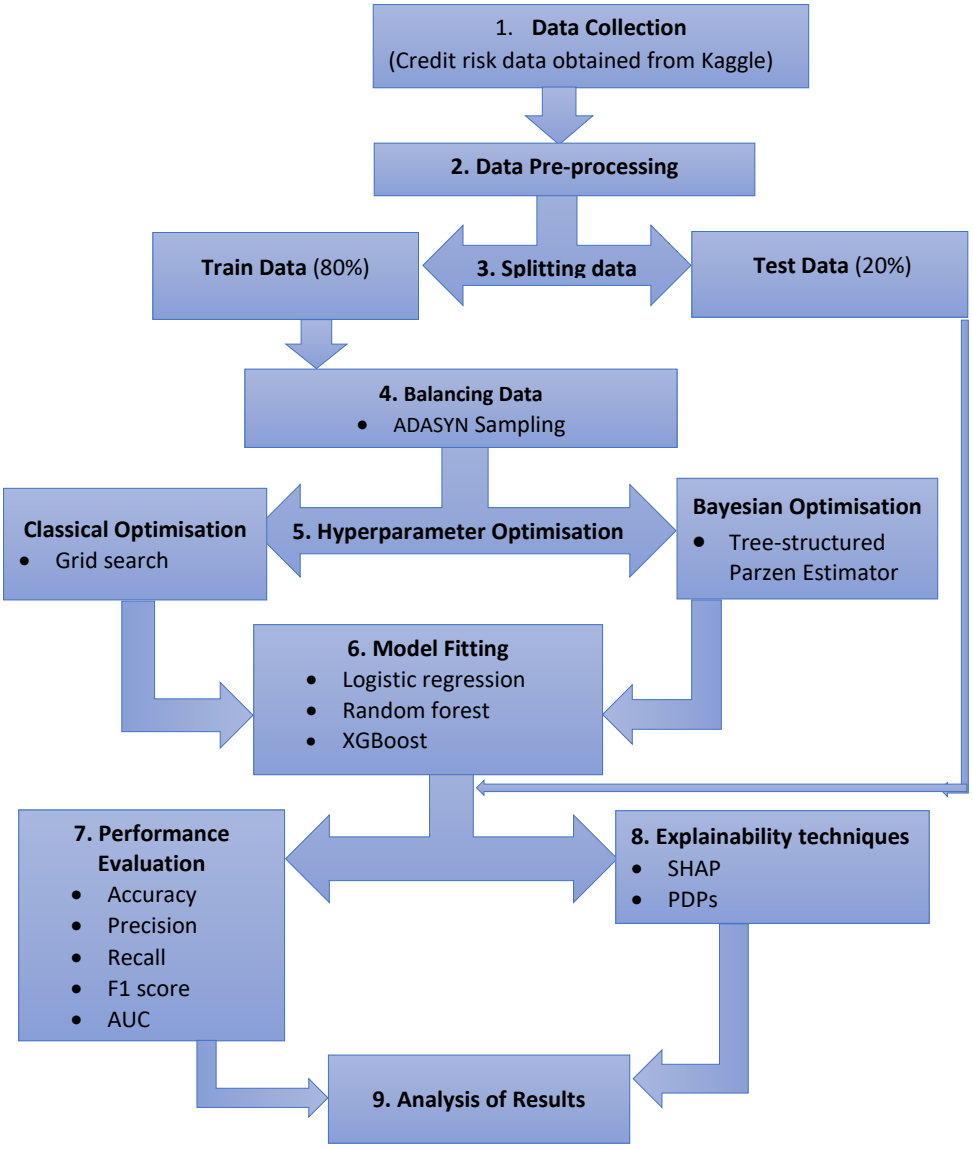


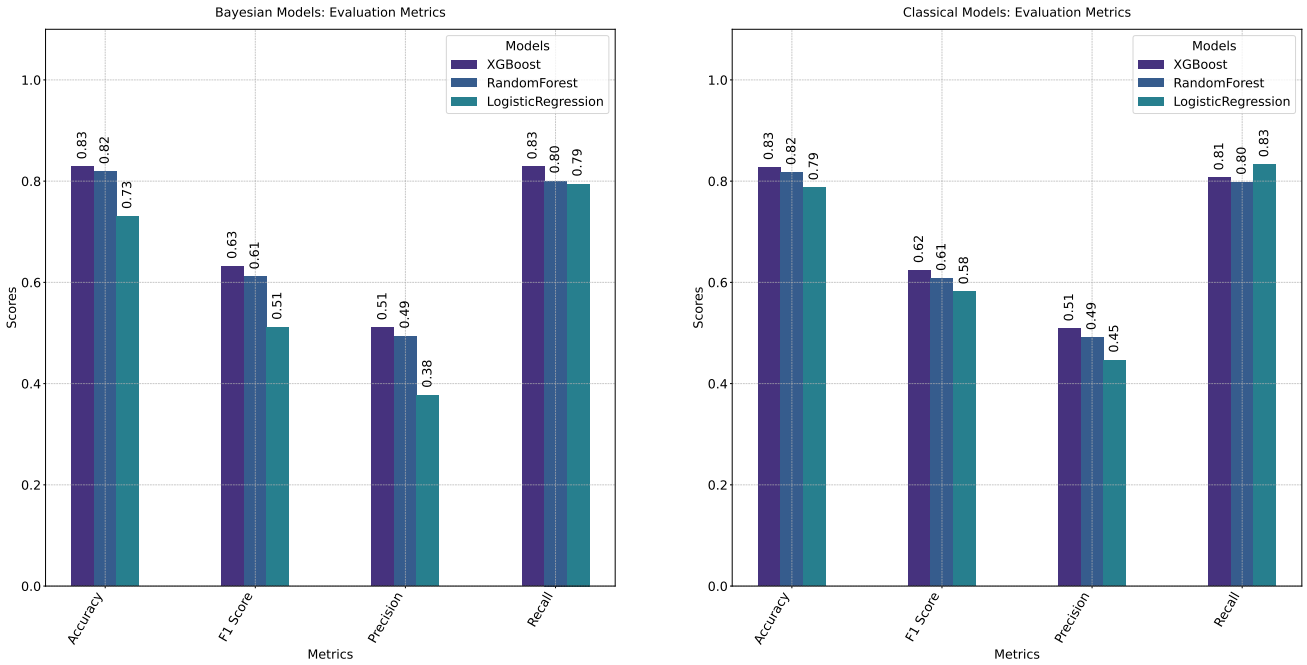
Figure 3.2: Experimental design flow chart.

4. Results and Discussion

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

4.1 Model Performance Measures

Figure 4.1 provides 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 precision, recall, precision, F1 scores, 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.



(a) Bayesian models

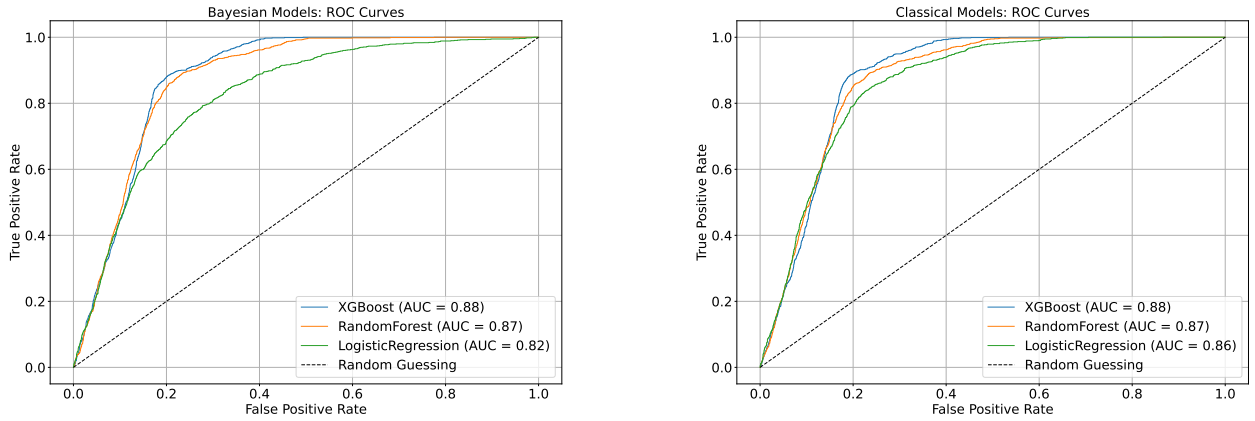
(b) Classical models

Figure 4.1: Performance measures. (a) Bayesian-optimised models and (b) classical-optimised models. These metrics were evaluated on the original test dataset for XGBoost, random forest, and logistic regression, comparing Bayesian optimisation to classical grid search techniques.

Figures 4.1a and 4.1b indicate a slight performance improvement in recall for XGBoost when using Bayesian hyperparameters, with a recall of 83% compared to 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 metrics (accuracy, precision, recall, and F1-score) for both Bayesian optimisation and grid search, 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.

The ROC and area under the curve (AUC) were used to measure the discriminatory power of the models under the two optimisation techniques. The AUC does not require the choice of threshold, making it suitable for classifying defaulters and non-defaulters. Figure 4.2 below shows the ROC curves and their AUC for the three models under Bayesian and grid-search optimisation techniques.



(a) Bayesian Models

(b) Classical Models

Figure 4.2: ROC curves: Classical vs Bayesian Models.

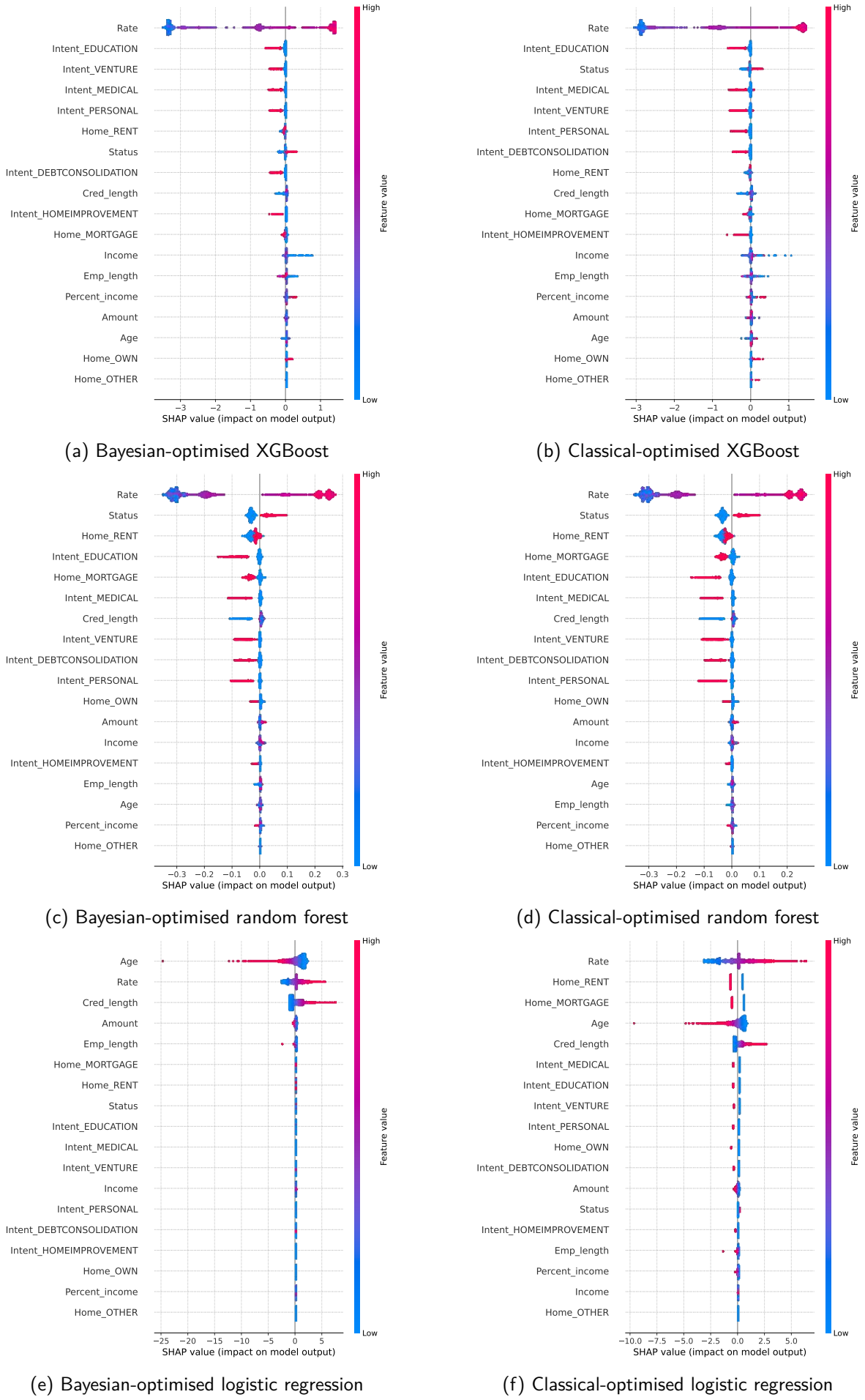
Comparing Figures 4.2a and 4.2b, we can see that using the Bayesian method instead of the classical approach does not improve the discriminatory power of XGBoost and random forest, both of which maintain an 88% and 87% respectively ability to differentiate between positive (defaulters) and negative classes (non-defaulters). The logistic regression's ability to distinguish between defaulters and non-defaulters drops with the Bayesian approach to 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 serve as a global interpretability tool, illustrating how the model's predicted outcome (probability of default) changes as a specific feature(s) varies while keeping other features constant. To ensure reliable evaluation of model interpretability, SHAP values and PDPs were calculated using the test dataset, allowing us to assess model performance on unseen data.

4.2.1 Global Interpretations with SHAP

The SHAP summary plot illustrates the importance of each feature by showing its distribution of Shapley values, which represent the contribution of that feature to the model's predictions (default probability or log odds of defaulting). Each dot corresponds to a single instance in the dataset and represents the Shapley value for a specific feature in that instance. Figure 4.3 displays the summary plots for both the Bayesian-optimised and classical (grid search) optimised models.



(a) Bayesian-optimised XGBoost

(b) Classical-optimised XGBoost

(c) Bayesian-optimised random forest

(d) Classical-optimised random forest

(e) Bayesian-optimised logistic regression

(f) Classical-optimised logistic regression

Figure 4.3: SHAP summary plots showing feature importance for Bayesian and classical-optimised models. Positive SHAP values indicate an enhanced likelihood of default. The feature values are colour-coded according to the scale on the right; for example, a higher *Rate*, denoted in red, is linked with

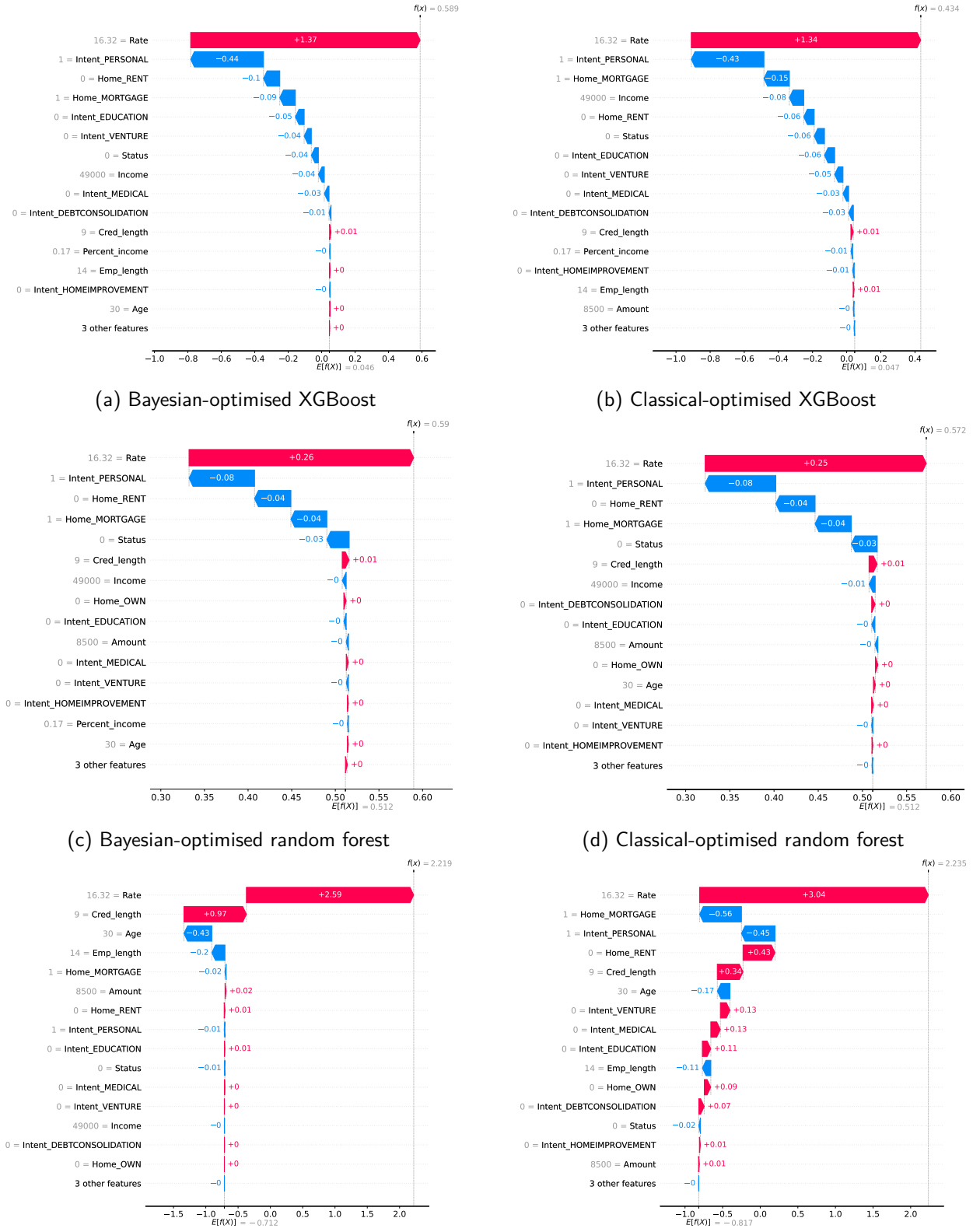
Looking at Figures 4.3a and 4.3b, we can see that the top five most important features identified by the Bayesian and classical-optimised XGBoost models differ. The Bayesian-optimised XGBoost identifies *Rate*, *Intent-purpose of the loan (education)*, *Intent-purpose of loan (venture)*, *Intent-purpose of loan (medical)*, and *Intent-purpose of loan (personal)* as the main 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 under the classical approach, the Bayesian approach ranks *Status* as the seventh most important variable. 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 Bayesian and classical-optimised random forest models, as illustrated in Figures 4.3c and 4.3d, 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, the classical approach ranks *Home-ownership status (mortgage)* in the same position. 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.

Looking at Figure 4.3e, the Bayesian-optimised logistic regression model places greater emphasis on *Age*, *Rate*, and *Credit_length*. In contrast, the classical-optimised logistic regression (Figure 4.3f) 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 Figure 4.2), hyperparameter tuning influences the importance of variables much more significantly, affecting how 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 model results and their implications for decision-making.

4.2.2 Local Explainability with SHAP

SHAP waterfall plots offer a detailed explanation of each feature's input on 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). The blue features indicate a reduction in the prediction, while the red features indicate an increase. Just like in 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. Figure 4.4 shows 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.



(a) Bayesian-optimised XGBoost

(b) Classical-optimised XGBoost

(c) Bayesian-optimised random forest

(d) Classical-optimised random forest

(e) Bayesian-optimised logistic regression

(f) Classical-optimised logistic regression

Figure 4.4: SHAP waterfall plots for local explainability with SHAP values derived from the individual who defaulted in the test data. Model predictions, denoted as $f(x)$, are expressed as log odds, with the model's average prediction represented by $E[f(x)]$. Features in red increase the log odds of defaulting above the average $E[f(x)]$, whereas those in blue signify a decrease.

Figures 4.4a and 4.4b show the SHAP waterfall plots for Bayesian and classical-optimised XGBoost models. 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 both Bayesian-optimised XGBoost and classical-optimised XGBoost. Bayesian-optimised XGBoost considers *Home-ownership status (rent)* as the third most important variable, while under the classical approach, the third most important variable is *Home-ownership status (mortgage)*. Thus, the two optimisation techniques result in different feature importances and contributions. Concerning model predictions $f(x)$, the Bayesian model estimates the log odds of defaulting for this individual at 0.589, that is, 0.643 probability of default, which surpasses the classical model's prediction of 0.434 (0.607 probability of default).

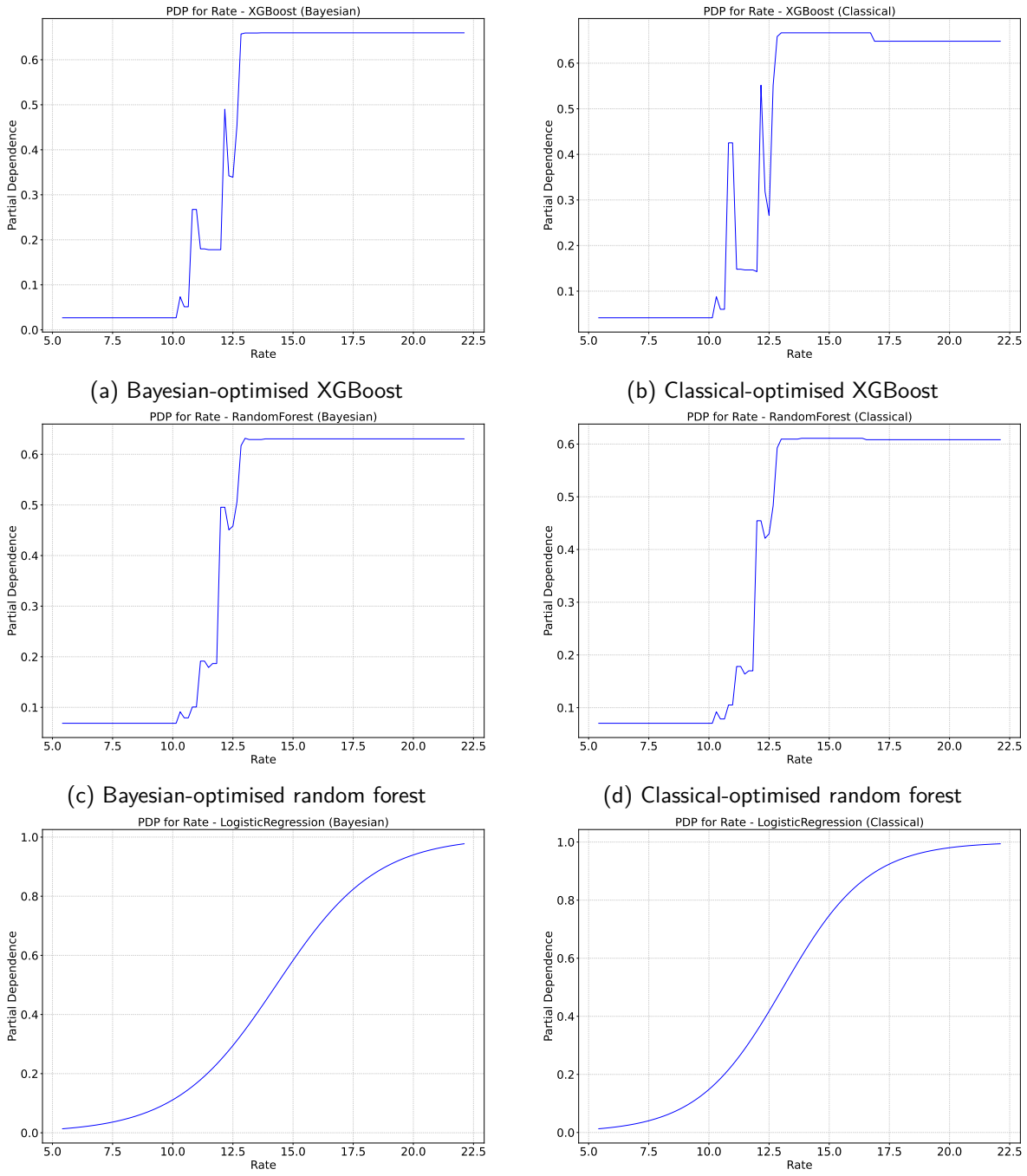
Figure 4.4c reveals that the Bayesian-optimised random forest identifies *Rate* of 9, *Intent-purpose of loan (personal)* and *Home-ownership status (rent)* as the three most important drivers influencing default for this individual. Here, a *Rate* of 16.32 pushes the default above the baseline, while other features push the prediction below the baseline. In contrast, Figure 4.4d shows that under the classical optimisation approach, the *Rate* of 16.32, *Intent-purpose of loan (personal)*, and *Home-ownership status(rent)* are the top three contributors influencing default. *Rate* pushes default above the baseline while the other variables reduce it. The log odds of defaulting 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 4.4e) highlights *Rate* of 16.32, Credit_length of 9, and *Age* of 30 as the top three significant variables affecting default. *Rate* is the main contributor, followed by *Credit_length* of 9, both pushing the log odds of default above the baseline, whereas *Age* of 30 reduces the prediction below the baseline. In contrast, the 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 hyperparameter optimisation method affects model performance and influences how individual predictions are explained. The probability of defaulting this customer ranged between 0.6 and 0.9 (refer to Table B.1 in Appendix B). This customer actually defaulted, indicating that all models effectively estimated a high likelihood of default for this individual.

4.2.3 Partial Dependence Plots

We computed partial dependence plots for*Rate* and *Age* for XGBoost, random forest, and logistic regression under Bayesian and classical optimisation techniques. Figure 4.5 below shows the PDPs for *Rate* produced by Bayesian and classical optimised models. The PDPs for *Age* are shown in Appendix C. The PDPs are plotted using the test data.



(a) Bayesian-optimised XGBoost

(b) Classical-optimised XGBoost

(c) Bayesian-optimised random forest

(d) Classical-optimised random forest

(e) Bayesian-optimised logistic regression

(f) Classical-optimised logistic regression

Figure 4.5: Partial dependence plots for variable *Rate*: Bayesian and classical models.

Figures 4.5a and 4.5b show the PDPs for variable *Rate* under the Bayesian optimised XGBoost and classical optimised XGBoost, respectively. In both situations, the default probability sharply rises after a *Rate* of 10. It flattens and remains constant under the Bayesian-optimised XGBoost. Still, under the classical-optimised XGBoost, there are some inconsistencies; 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.

Figures 4.5c and 4.5d show PDPs for *Rate* under Bayesian and classical-optimised random forest, respectively. The PDPs are quite 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 variable *Rate* and 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 model's prediction. The minimal difference in performance indicates that the Bayesian optimisation does not alter the relationship between *Rate* and probability of default when using the random forest model.

For logistic regression, the PDPs for *Rate* under the Bayesian approach (Figure 4.3e and classical approach (Figure 4.5f) show that the probability of default continuously rises as *Rate* increases. The shape of the two graphs is almost the same, showing that using Bayesian hyperparameters does not alter the relationship between *Rate* and probability of default.

4.3 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 can be due to the flexibility and probabilistic nature of Bayesian optimisation (TPE approach), favouring 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 performance under Bayesian optimisation compared to 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 a significant improvement 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. Limited performance gains in random forest and logistic regression reinforce the idea that the benefits of Bayesian optimisation are more pronounced in models with more complex hyperparameter spaces, such as XGBoost. Overall, Bayesian optimisation reduces computational time and improves recall performance while maintaining ROC performance similar to 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 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 model transparency, which is often crucial in regulatory environments.

The SHAP summary and waterfall plots reveal the effect of different hyperparameter optimisation techniques on global and local model 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 regularisation parameters more precisely, allowing *Age* to be ranked higher than *Rate*. In contrast, grid search, with fewer optimisation steps, may not adjust 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 more significant in logistic regression. In their paper, Kong et al. (2023) also found that the Bayesian optimisation method improves model interpretability through SHAP analysis, but their study did not attribute the reasons for their results.

The partial dependence plots analysis further illustrates how hyperparameter tuning methods impact the relationship between 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 4.5a and 4.5b). The smoother PDP produced by the Bayesian-optimised XGBoost model (Figure 4.5a) compared to the classical model (Figure 4.5b) 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 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 in sparse data by interpolating better than XGBoost and random forest models. The smooth transition in 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, XGBoost and random forest models show more abrupt changes in predictions, potentially reflecting overfitting in data-scarce regions.

5. Conclusion

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

Bayesian hyperparameter optimisation shows potential for improving model performance, particularly for XGBoost, where it enhances recall. Improving recall is crucial in credit risk modelling, as identifying defaulters and minimising false negatives is 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 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 model 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 model interpretation and decision-making, as they indicate that optimisation strategies can directly influence the perceived determinants of default risk.

The findings show that hyperparameter optimisation has minimal impact on the relationships between predictor variables and the target (default), as illustrated by partial dependence plots (PDPs). Despite differences in smoothness, PDPs under Bayesian and classical optimisation both indicate that the probability of default increases with the variable *Rate*. This is especially relevant in credit risk models used in decision-making, where the ability to understand relationships is as important.

5.1 Recommendations to Risk Modellers

These findings are significant for credit risk modellers, highlighting the balance between model 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 models or evaluation metrics. It is necessary to understand the effect of specific characteristics of each model to avoid suboptimal performance or misleading feature importance.

Moreover, modellers have to balance performance and interpretability. In credit risk, 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 prioritizing the simpler approach. Logistic regression offers significant advantages in terms of interpretability, ease of deployment, and compliance with regulatory standards. Therefore, when data demonstrates 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.

Another limitation is our choice of models. We selected logistic regression, random forest, and XGBoost. 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 limitations listed could be considered for 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.

Code Availability

The Python codes and dataset for this project are available at https://github.com/tatendashoko/AIMS-PROJECT.

Acknowledgements

I want to acknowledge AIMS and its funders 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, and my tutor, Anicet Hounkanrin, for his support and guidance in this work.

References

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

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

Alibrahim, H. and Ludwig, S. A. Hyperparameter optimization: Comparing genetic algorithm against grid search and Bayesian optimization. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 1551–1559. IEEE, 2021.

Alonso Robisco, A. and Carbo Martinez, J. M. Measuring the model risk-adjusted performance of machine learning algorithms in credit default prediction. *Financial Innovation*, 8(1):70, 2022.

Altman, E. I. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, 23(4):589–609, 1968.

Antony, T. M. and Suresh, G. Determinants of credit risk: Empirical evidence from Indian commercial banks. *Banks and Bank Systems*, 18(2):88–100, 2023.

Basel Committee on Banking Supervision. Principles for the management of credit risk, July 1999. URL https://www.bis.org/publ/bcbs54.htm. Accessed: May 17, 2023.

Bentéjac, C., Csörgő, A., and Martínez-Muñoz, G. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54:1937–1967, 2021.

Bergstra, J. and Bengio, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(2), 2012.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. Algorithms for Hyper-Parameter Optimization. *Advances in Neural Information Processing Systems*, 24, 2011.

Bertrand, A., Eagan, J. R., Maxwell, W., and Brand, J. 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*, pages 1–21, 2024.

Bhatt, T. K., Ahmed, N., Iqbal, M. B., and Ullah, M. Examining the determinants of credit risk management and their relationship with the performance of commercial banks in Nepal. *Journal of Risk and Financial Management*, 16(4):235, 2023. doi: $10.3390/\mathrm{jrfm16040235}$. URL https://doi.org/10.3390/jrfm16040235.

Bishop, C. M. and Nasrabadi, N. M. *Pattern Recognition and Machine Learning*, volume 4. Springer, 2006.

Bramer, M. Avoiding overfitting of decision trees. *Principles of Data Mining*, pages 119–134, 2007.

Breiman, L. Random forests. *Machine learning*, 45:5–32, 2001.

Bussmann, N., Giudici, P., Marinelli, D., and Papenbrock, J. Explainable machine learning in credit risk management. *Computational Economics*, 57(1):203–216, 2021.

Caton, S., Malisetty, S., and Haas, C. Impact of imputation strategies on fairness in machine learning. *Journal of Artificial Intelligence Research*, 74:1011–1035, 2022.

Chen, H., Yang, C., Du, M., and Zhang, Y. Research on Credit Risk Prediction under Unbalanced Dataset Based on Ensemble Learning. *Mathematical Problems in Engineering*, 2023. doi: 10.1155/2023/2927393.

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

Crook, J. Credit scoring and its applications. *Journal of the Operational Research Society*, 52:997–1006, 2002.

De Lange, P. E., Melsom, B., Vennerød, C. B., and Westgaard, S. Explainable ai for credit assessment in banks. *Journal of Risk and Financial Management*, 15(12):556, 2022.

Dube, L. and Verster, T. Enhancing classification performance in imbalanced datasets: A comparative analysis of machine learning models. *Data Science in Finance and Economics*, 3(4):354–379, 2023.

Dube, L. and Verster, T. Interpretability of the random forest model under class imbalance. *Data Science in Finance and Economics*, 4(3):446–468, 2024.

Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

Gatla, T. Machine learning in credit risk assessment: Analyzing how machine learning models are transforming the assessment of credit risk for loans and credit cards. *Journal of Emerging Technologies and Innovative Research*, 10:k746–k750, 06 2023a.

Gatla, T. R. Machine learning in credit risk assessment: Analyzing how machine learning models are transforming the assessment of credit risk for loans and credit cards. *Journal of Emerging Technologies and Innovative Research*, 10(6):746–750, 2023b.

Hand, D. J. and Henley, W. E. Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 160(3):523–541, 1997.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

He, H., Bai, Y., Garcia, E., and Li, S. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1322 – 1328, 07 2008. doi: 10.1109/IJCNN.2008.4633969.

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

Hu, L., Chen, J., Vaughan, J., Yang, H., Wang, K., Sudjianto, A., and Nair, V. N. Supervised machine learning techniques: An overview with applications to banking. *International Statistical Review*, 89: 573 – 604, 2020. URL https://api.semanticscholar.org/CorpusID:221090618.

Inga, J. and Sacoto-Cabrera, E. Credit default risk analysis using machine learning algorithms with hyperparameter optimization. In *International Conference on Science, Technology and Innovation for Society*, pages 81–95. Springer, 2022.

Kaggle. Loan applicant data for credit risk analysis dataset, 2021. URL https://www.kaggle.com/datasets/nanditapore/credit-risk-analysis.

Kaggle. Kaggle: Your home for data science, 2024. URL https://www.kaggle.com/. Accessed: 2024-10-19.

Kłosok, M., Chlebus, M., et al. *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, 2020.

Kong, Y., Wang, Y., Sun, S., and Wang, J. XGB and SHAP credit scoring model based on Bayesian optimization. *Journal of Computing and Electronic Information Management*, 10(1):46–53, 2023.

Levesque, J., Gagné, C., and Sabourin, R. Bayesian hyperparameter optimization for ensemble learning. *CoRR*, abs/1605.06394, 2016. URL http://arxiv.org/abs/1605.06394.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 2017.

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

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

Melsom, B., Vennerød, C. B., de Lange, P., Hjelkrem, L. O., and Westgaard, S. Explainable artificial intelligence for credit scoring in banking. *Journal of Risk*, 25(2), 2022.

Misheva, B. H., Osterrieder, J., Hirsa, A., Kulkarni, O., and Lin, S. F. Explainable ai in credit risk management. *arXiv preprint arXiv:2103.00949*, 2021.

Molnar, C. *Interpretable Machine Learning*. Leanpub, 2020. ISBN 9780244768522. URL https://books.google.co.za/books?id=jBm3DwAAQBAJ.

Murphy, K. P. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. URL probml.ai.

Nohara, Y., Matsumoto, K., Soejima, H., and Nakashima, N. Explanation of machine learning models using Shapley additive explanation and application for real data in hospital. *Computer Methods and Programs in Biomedicine*, 214:106584, 2022.

Overisch, M. A conceptual explanation of Bayesian model-based hyperparameter optimization for machine learning, 2020. URL https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f. Accessed: 2024-10-19.

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

Oxford Academic. *Credit Risk Management: Basic Concepts: Financial Risk Components, Rating Analysis, Models, Economic and Regulatory Capital*. Oxford University Press, 2008. URL https://doi.org/10.1093/acprof:oso/9780199545117.002.0007. online edn, Oxford Academic, 1 Jan. 2009.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011a.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine Learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011b.

PK, R. et al. Enhanced credit card fraud detection: A novel approach integrating Bayesian optimized random forest classifier with advanced feature analysis and real-time data adaptation. *International Journal for Innovative Engineering & Management Research, Forthcoming*, 2023.

Rodríguez-Pérez, R. and Bajorath, J. Interpretation of compound activity predictions from complex machine learning models using local approximations and Shapley values. *Journal of medicinal chemistry*, 63(16):8761–8777, 2019.

Scikit-Learn. Decision trees. https://scikit-learn.org/stable/modules/tree.html, 2023. Accessed: 2024-02-12.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the human out of the loop: A review of Bayesian Optimization. *Proceedings of the IEEE*, 104:148–175, 2016. URL https://api.semanticscholar.org/CorpusID:14843594.

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

Wang, Y. and Ni, X. S. A XGBoost risk model via feature selection and Bayesian hyper-parameter optimization, 2019. URL https://arxiv.org/abs/1901.08433.

Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., and Deng, S.-H. Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019.

XGBoost Documentation. *XGBoost Official Documentation*, 2024. URL https://xgboost.readthedocs.io/en/stable/tutorials/model.html.

Xia, Y., Liu, C., Li, Y., and Liu, N. A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, 78:225–241, 2017. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2017.02.017. URL https://www.sciencedirect.com/science/article/pii/S0957417417301008.

Xu, T. Comparative Analysis of Machine Learning Algorithms for Consumer Credit Risk assessment. *Transactions on Computer Science and Intelligent Systems Research*, 4:60–67, 2024.

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

Zhang, C. and Zhou, X. Forecasting credit risk of smes in supply chain finance using Bayesian optimization and xgboost. *Mathematical Problems in Engineering*, 2023(1):5609996, 2023.

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

AppendixA. Hyperparameter Tuning Results

This section shows the hyperparameter tuning results obtained from Bayesian and classical approaches. Table A.1 below shows the optimal parameters for the XGBoost model, the search space and the function of each hyperparameter.

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 A.1: Hyperparameters, their descriptions, search space, and optimal values for XGBoost.

Table A.2 below shows the optimal parameters for the random forest model, search space, and the function of each hyperparameter.

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 A.2: Hyperparameters, their descriptions, search space, and optimal values for random forest model.

Table A.3 below shows the optimal values of `C` for the logistic regression model and search space for both Bayesian and classical optimisation.

Hyperparameter	Purpose	Search Space	Optimal (Classical)	Optimal (Bayesian)
C	Inverse of regularisation	0.01 – 10	7.9	2

Table A.3: Hyperparameters, their descriptions, search space, and optimal values logistic regression.

AppendixB. Probability of Default Results

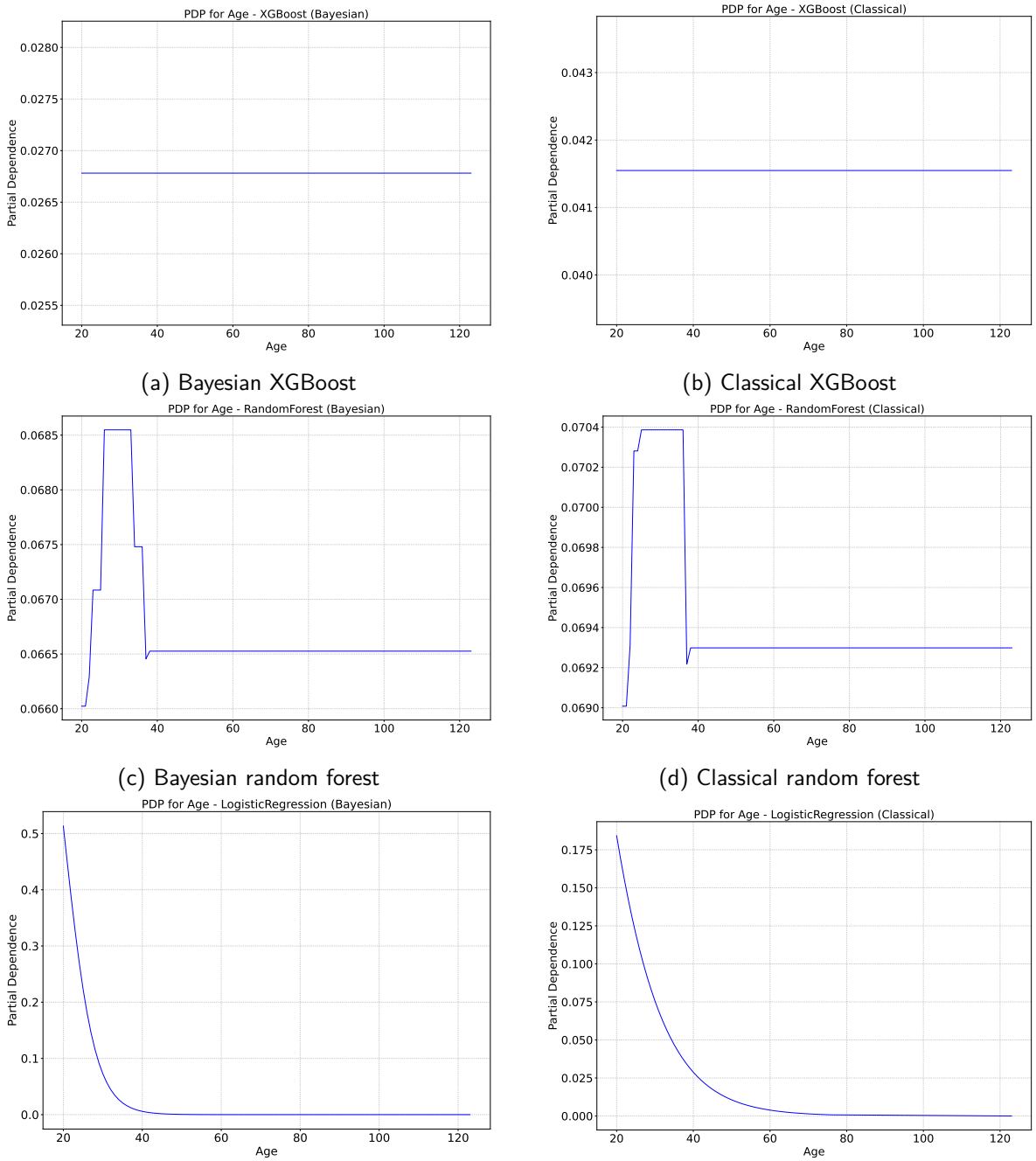
Table B.1 below shows the PD estimates of the first defaulting case in the test data. The estimates are obtained by using Bayesian and classical-optimised models.

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

Table B.1: PD estimates for Bayesian and classical models based on the first defaulting observation in the test dataset

AppendixC. Partial Dependence Plots

Figures C.1a - C.1f below show the partial dependence plots for *Age* for the three classifiers under the Bayesian and classical approaches.



(a) Bayesian XGBoost

(b) Classical XGBoost

(c) Bayesian random forest

(d) Classical random forest

(e) Bayesian logistic regression

(f) Classical Logistic Regression

Figure C.1: Partial dependence plots for *Age*, Bayesian and classical models