Comparative Analysis of Machine Learning Approaches for Predicting Loan Default Risk

By

Afrin Jahan
ID: CSE2203027136

Supervised By:

Zarin Hadika

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science & Engineering.



Department of Computer Science & Engineering Sonargaon University (SU)

September 2026

APPROVAL

The thesis titled "**Comparative Analysis of Machine Learning Approaches for Predicting Loan Default Risk**" submitted by **Afrin Jahan** (CSE2203027136) to the Department of Computer Science and Engineering, Sonargaon University (SU) has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

---------- **Supervisor**

Zarin Hadika
Lecturer
Department of Computer Science and Engineering
Sonargaon University (SU)

---------- **Examiner 1**

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

---------- **Examiner 2**

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

---------- **Examiner 3**

(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

DECLARATION

We hereby declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Zarin Hadika,** Lecturer, Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this Thesis and thereof has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Zarin Hadika)
Supervisor

Signatures

Afrin Jahan
ID: CSE2203027136

ABSTRACT

Banks and other lenders need to know, before giving out a loan, how likely it is that the borrower will fail to pay it back. This is called credit risk. In the past, banks used simple statistical methods to check this risk. These methods were easy to understand but not always very accurate. Newer machine learning models, such as Random Forest and gradient boosting methods, can often predict default more accurately. However, these models are harder to explain. A bank cannot easily tell a customer why they were rejected if the model works like a black box. This creates a problem: should a lender choose a model that is more accurate, or one that is easier to explain? Most past studies only test one model on one dataset, so it is hard to know if the results would stay the same on different data.

This thesis tries to solve both problems together. Five machine learning models are used: Logistic Regression, Support Vector Machine (with an RBF kernel), Random Forest, XGBoost, and CatBoost. These models are tested on two different loan datasets. The first dataset has 6,843 loan records, with about 35% marked as default. The second dataset has 614 loan applications, with about 31% marked as not approved. All five models are trained and tested using the same method, called nested cross-validation, so the comparison between models is fair. After training, a tool called SHAP is used to explain which features (such as income or credit score) affect each model's decision the most.

On the first dataset, all five models perform in a similar range, with ROC-AUC scores between about 0.73 and 0.74. CatBoost scores the highest (0.743), but the difference between models is small. Across all models, Income, Credit Score, and Loan Amount are found to be the most important features. On the second dataset, results are noticeably better overall. Random Forest gets the best ROC-AUC score (0.754), and SVM-RBF gets the highest accuracy (80.8%). In this dataset, one feature stands out far more than the rest: Credit History. This matches what is already known in banking a person's past record of paying credit is one of the strongest signs of future default risk.
These results suggest that once the data is properly cleaned and prepared, the choice of features matters more than the choice of algorithm. A dataset with a strong, meaningful feature like Credit History leads to better results, no matter which model is used. The thesis also shows that SHAP can be used with both simple models (like Logistic Regression) and complex models (like CatBoost) to explain their decisions in a similar way. This means banks do not have to choose a harder-to-explain model just to get better accuracy — explainability tools can work well even with strong-performing models. The thesis ends with a discussion of the trade-off between accuracy and explainability, and suggests future work such as testing more datasets, checking for bias, and trying deep learning models.

Keywords: loan default prediction, credit risk, machine learning, nested cross-validation, explainable AI, SHAP, XGBoost, CatBoost, Random Forest, Support Vector Machine, Logistic Regression

ACKNOWLEDGMENT

At the very beginning, we would like to express our deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the schedule time.

We are auspicious that we had the kind association as well as supervision of **Zarin Hadika**, Lecturer, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our thesis.

We would like to convey our special gratitude to **Prof. Bulbul Ahamed,** Head, Department of Computer Science and Engineering, Sonargaon University, for his kind concern and precious suggestions.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to our parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AUC	Area Under the Curve
CV	Cross-Validation
F1	F1-Score (harmonic mean of Precision and Recall)
GridSearchCV	Grid Search Cross-Validation (hyperparameter search method)
LR	Logistic Regression
ML	Machine Learning
RBF	Radial Basis Function (an SVM kernel)
RF	Random Forest
ROC	Receiver Operating Characteristic
SHAP	SHapley Additive exPlanations
SVM	Support Vector Machine
XAI	Explainable Artificial Intelligence
XGBoost	Extreme Gradient Boosting

TABLE OF CONTENTS

Tittle

APPROVAL ---- i
DECLARATION ---- ii
ABSTRACT ---- iii
ACKNOWLEDGMENT ---- iv
LIST OF ABBREVIATIONS ---- v
CHAPTER 1 ---- 1
INTRODUCTION ---- 1
1.1Background of the Study ---- 1
1.2 Problem Statement ---- 2
1.3 Motivation ---- 3
1.4 Research Objectives ---- 4
1.5 Scope of the study ---- 4
1.6 Contributions of the thesis ---- 5
1.7 Organization of the thesis ---- 5
Chapter 2 ---- 6
Literature Review ---- 6
2.1Traditional Statistical Approaches to Loan Default Prediction ---- 6
2.2Machine Learning Approaches to Credit/Loan Default Prediction ---- 7
2.3Random Forest in Credit Risk Modelling ---- 9
2.4 Gradient Boosting Methods: XGBoost and CatBoost ---- 10
2.5 Explainable AI and SHAP in Credit Risk Modelling ---- 11
2.6 Research Gap ---- 12
Chapter 3 ---- 13
Methodology ---- 13
3.1 System Architecture ---- 13
3.2 Dataset 1 ---- 15
3.3 Dataset 2 Description ---- 16
3.4 Data Preprocessing ---- 17

3.4.1 Handling of Missing Values ----18
3.4.2 Encoding of Categorical Variables ----18
3.4.3 Scaling of Numeric Variables ----19
3.5 Nested Cross-Validation ----19
3.5.1 Outer loop ----19
3.5.2 Inner Loop ----19
3.6 Model Training and Final Model Fitting ----22
3.7 Evaluation Metrics ----23
3.8 Model Explainability with SHAP ----24
3.9 Tools and Implementation Environment ----25
Chapter 4 ----26
Experimental Results and Analysis ----26
4.1 Cross-Validation Performance (Outer Loop) ----26
4.2 Single train test split performance ----27
4.3 ROC Curve Comparison ----28
4.4 Confusion Matrix ----30
4.5 Explainability Results (SHAP) ----31
4.5.1 Dataset 1- Loan Approval ----32
4.5.2 Dataset 2 Loan Details ----36
Chapter 5 ----39
Discussion ----39
Chapter 6 ----41
Conclusion ----41
6.1 Summary of the Study ----41
6.2 Summary of Key Findings ----41
Chapter 7 ----43
Future Work ----43
References ----44

LIST OF TABLES

Table 3.1:Class Distribution of Dataset 1 (Defaulted Variable) ……. 15

Table 3.2: Dataset 1 feature description……...15

Table 3.3: Class Distribution of Dataset 2 (Loan_Status Variable) … ..…….16

Table 3.4: Dataset 2 feature description……...16

Table 3.5: Hyperparameter search space per model…….20

Table 3.6: Software libraries and their role in this study…….24

Table 4.1: Nested cross-validation results, Dataset 1 (mean ± standard deviation across 5…26 outer folds)

Table 4.2: Nested cross-validation results, Dataset 2 (mean across 5 outer folds)…….27

Table 4.3: Final test-set results, Dataset 1……27

Table 4.4: Final test-set results, Dataset 2……28

LIST OF FIGURES

Figure No.	Tittle	Page No.
Figure1.1	Traditional Decision Tree Framework for Credit Risk Assessment	01
Figure 2.1	System Architecture	14
Figure 4.1	ROC curves, Dataset 1 (all five models, final test set)	29
Figure 4.2	ROC curves, Dataset 2 (all five models, final test set)	29
Figure 4.3	Confusion matrices, Dataset 1 (final test set, n = 1,369)	30
Figure 4.4	Confusion matrices, Dataset 2 (final test set, n = 123)	31
Figure 4.5.1a	SHAP summary (beeswarm) plot, CatBoost, Dataset 1 (Loan Approval)	32
Figure 4.5.1b	SHAP mean |value| bar plot, CatBoost, Dataset 1 (Loan Approval)	33
Figure 4.5.2a	SHAP summary (beeswarm) plot, Random Forest, Dataset 1 (Loan Approval)	34
Figure 4.5.2b	SHAP mean |value| bar plot, Random Forest, Dataset 1 (Loan Approval)	34
Figure 4.5.3a	SHAP summary (beeswarm) plot, XGBoost, Dataset 1 (Loan Approval)	35
Figure 4.5.3b	SHAP mean |value| bar plot, XGBoost, Dataset 1 (Loan Approval)	35
Figure 4.6.4a	SHAP summary (beeswarm) plot, CatBoost, Dataset 2 (Loan Details)	36
Figure 4.6.4b	SHAP mean |value| bar plot, CatBoost, Dataset 2 (Loan Details)	36
Figure 4.5.5a	SHAP summary (beeswarm) plot, Random Forest, Dataset 2 (Loan Details)	37
Figure 4.6.5b	Figure 4.6.5b SHAP mean |value| bar plot, Random Forest, Dataset 2 (Loan Details)	37

Figure 4.5.6a	Figure 4.5.6a SHAP summary (beeswarm) plot, XGBoost, Dataset 2 (Loan Details	38
Figure 4.5.6b	SHAP mean |value| bar plot, XGBoost, Dataset 2 (Loan Details)	38

CHAPTER 1

INTRODUCTION

1.1Background of the Study

Credit risk assessment is a fundamental component of the lending process in banks and other financial institutions. Before approving a loan, lenders must evaluate the likelihood that a borrower will repay the borrowed amount according to the agreed terms. The failure of a borrower to meet these repayment obligations is commonly referred to as loan default. Accurate prediction of loan default is essential because it enables financial institutions to minimize credit losses, improve lending decisions, maintain portfolio quality, and strengthen their overall financial stability. Therefore, the development of reliable and effective loan-default prediction models has become an important area of research in the field of financial risk management.

For much of the twentieth century, this judgement was formed using rule-based scorecards and traditional statistical techniques. A loan officer, or later a statistical model built by a risk analyst, would combine a small number of financial ratios and demographic indicators income, existing debt, employment status, age into a single score using fixed weights, often derived from a technique called discriminant analysis or, somewhat later, logistic regression [1]. These approaches had a significant practical virtue: they were simple enough to audit by hand, and their logic could be explained in plain language to a regulator, an auditor, or the applicant themselves [2]. Their central limitation, however, was that they generally assumed a straightforward, often linear, relationship between each input variable and the outcome being predicted, and they struggled to capture more complex, non-linear interactions between variables for example, a situation where high income only reduces default risk when combined with a stable employment history, but not otherwise [3].

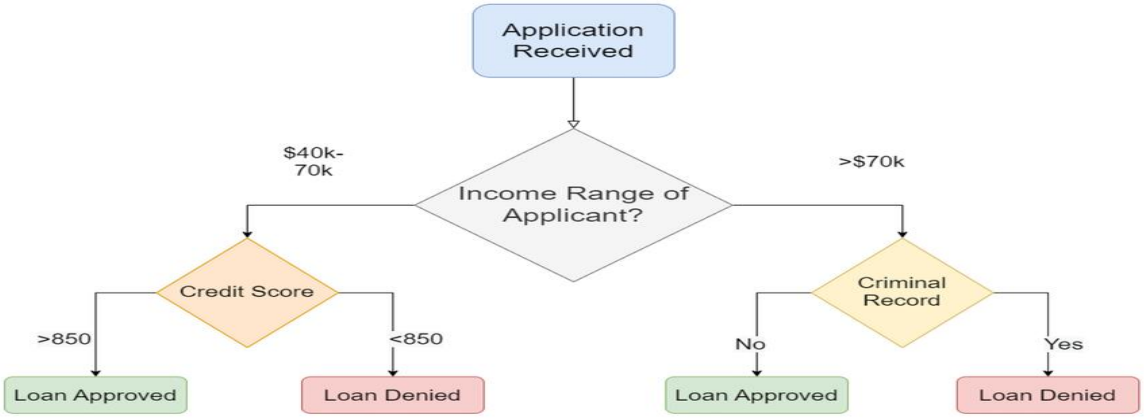


Figure1.1: Traditional Decision Tree Framework for Credit Risk Assessment

Over roughly the last two decades, and accelerating sharply in the last ten years, this landscape has shifted substantially toward machine learning. As computing power has become cheaper

and datasets richer, algorithms such as Random Forest, Support Vector Machines, and gradient boosting frameworks like XGBoost and CatBoost have repeatedly demonstrated an ability to extract more predictive signal from the same underlying data than a traditional statistical scorecard can, precisely because they are built to automatically detect non-linear patterns and interactions between features without requiring an analyst to specify them by hand in advance [4]. This capability has made machine learning an increasingly attractive option for institutions seeking to reduce default losses and make faster, more granular lending decisions, including for borrower populations such as those applying for small, short-term, or emergency loans who may not have the long, well-documented credit histories that traditional scorecards were originally designed around.

This shift, however, has not come without serious complication. The very flexibility that allows a machine learning model to capture more complex patterns is also what makes many of these models difficult, and sometimes practically impossible, to interpret directly [5]. A logistic regression coefficient can be explained to a non-technical audience in a single sentence. A prediction produced by an ensemble of two hundred decision trees, or by a kernel-based Support Vector Machine operating in a transformed, high-dimensional feature space, cannot be explained this way without additional tools designed specifically for that purpose. In a regulated financial context, where lenders in many jurisdictions are legally required to provide applicants with a meaningful reason for a credit decision, and where auditors and regulators need to verify that a model is not relying on inappropriate or discriminatory signals, this opacity is not merely an academic inconvenience it is a genuine barrier to responsible adoption. Predictive accuracy alone, however impressive, is therefore not a sufficient basis for deploying a model in this setting; a model must also be interpretable enough, in some meaningful sense, to satisfy the transparency obligations that come with making decisions about people's access to credit.

1.2 Problem Statement

The specific problem this thesis addresses can be stated precisely. Complex machine learning models gradient boosting ensembles such as XGBoost and CatBoost, and kernel-based methods such as Support Vector Machines with a non-linear kernel frequently achieve better raw predictive performance than simpler models such as Logistic Regression when measured using standard classification metrics. At the same time, these more complex models are structurally harder to interpret: their decision logic is distributed across hundreds of trees, or embedded in a non-linear kernel transformation, rather than expressed as a small set of directly readable coefficients.

A second, related problem compounds the first. Much of the existing research comparing machine learning models for loan default or credit-risk prediction evaluates its chosen models on a single dataset, using whatever train-test split or cross-validation scheme that particular study happens to adopt. This makes it genuinely difficult to know whether a reported finding for instance, that XGBoost outperforms Logistic Regression by some margin reflects a general property of these two algorithms on this type of problem, or whether it is a result specific to the particular dataset, feature set, and evaluation method used in that one study. Without a

consistent evaluation protocol applied across multiple model families and multiple datasets, it is difficult to draw conclusions that a practitioner could confidently apply to a new, different lending dataset.

A third problem, connected to the first two, concerns how explainability is typically handled in existing literature. Many studies that do apply an explainability method such as SHAP apply it only to their single best-performing model, rather than to every model under comparison. This makes it impossible to know from those studies alone whether the features identified as important are a genuine property of the underlying data and problem, or whether they are, at least partly, an artefact of the specific model architecture used to generate the explanation.

This thesis is designed to address all three of these problems together, rather than in isolation. It evaluates five classification models Logistic Regression, SVM-RBF, Random Forest, XGBoost, and CatBoost under one identical, carefully separated nested cross-validation protocol; it applies SHAP-based explainability consistently to every one of these five models, rather than only to the best-performing one; and it repeats this entire process independently on two structurally different loan datasets, so that any finding about model performance or feature importance can be checked for consistency, rather than being reported from a single dataset alone.

1.3 Motivation

This topic was chosen because it sits at a genuinely practical intersection between two concerns that are often treated as separate research questions: how well a model predicts, and whether that prediction can be explained and trusted. From a purely commercial perspective, even a small improvement in predictive accuracy can translate into a meaningful reduction in bad-debt losses once applied across a large loan portfolio, which is a strong incentive for lenders to adopt more sophisticated models. From a regulatory and ethical perspective, however, a model that cannot explain why a specific applicant was rejected risks eroding the trust of both the applicant and the wider public in automated decision-making[6], and may in some jurisdictions fall short of legal requirements around the right to an explanation for automated decisions[7].

Explainability, in this framing, is not a secondary nicety to be added on if time permits. It is a practical requirement with several distinct, concrete uses. It allows a loan officer or customer service representative to give an applicant a specific, defensible reason for a decision, rather than an unhelpful no message. It allows an internal auditor or external regulator to check that a model is not relying, even indirectly, on demographic proxies that would raise fairness concerns. And it allows the model developers themselves to sanity-check that a model has learned something sensible and consistent with domain knowledge for example, that a stronger credit history should reduce, not increase, predicted default risk rather than picking up on coincidental patterns or noise in the training data.

The specific decision to design this study around two datasets, rather than one, was a deliberate response to the second problem identified the difficulty of generalizing findings from a single dataset comparison [8]. By deliberately choosing two datasets that differ in size (6,843 records

versus 614 records), in class balance, and in the specific features available (notably, the presence of a Credit_History feature in one dataset but not the other), this study is able to check directly whether conclusions about model ranking and feature importance are stable across meaningfully different data, rather than simply assuming they would be.

Figure 1 below represents one possible decision tree developed for credit risk analysis and loan processing. The three main aspects characteristic of the applicant Income Range, Credit Score, and Criminal Record interact to make a specific binary decision (Loan Approved or Loan Denied) based on the principles of decision-making tree.

For credit-related defaults, one can assume that such a decision tree represents a typical machine learning model for the job. Using the basic logic incorporated into the model, one can comparatively analyze individual characteristics and determine their impact on the final decision. At the same time, more complex machine learning models, such as Random Forest, XGBoost, and Neural Networks, can be implemented to improve the accuracy of predictions.

1.4 Research Objectives

This study has the following objectives:

•To Preprocess and prepare Dataset 1 and Dataset 2 for supervised binary classification, including handling of missing values, encoding of categorical variables, and scaling of numeric variables where required by the model being used.

•To Train and tune five classification models Logistic Regression, SVM-RBF, Random Forest, XGBoost, and CatBoost using a nested cross-validation protocol, applied identically to each dataset.

•To Evaluate and compare these five models using Accuracy, Precision, Recall, F1-score, and ROC-AUC, both under nested cross-validation and on a final held-out test set.

• To Interpret model predictions using SHAP-based explainability, applied consistently to every model rather than only the best-performing one, to identify which features most strongly influence each model's decisions.

•To Compare findings on model performance and on feature importance across the two datasets, to assess how consistent these findings are.

1.5 Scope of the study

This study was limited to two structured, tabular loan-related datasets, namely, a loan-default dataset with a binary target and a loan-approval dataset with a binary target and credit history. The study evaluated five classical and ensemble machine-learning classifiers under a standardized nested cross-validation paradigm and compared their performance using standard classification metrics with SHAP applied consistently for post-hoc model interpretation.

Deep learning architectures, real-time or production deployment considerations, and a dedicated fairness or bias audit across protected demographic attributes are outside the scope of this study. These are identified explicitly as directions for future work in Chapter 7, rather

than as gaps that undermine the comparative aim of this thesis, which is focused specifically on classical and ensemble model families under one consistent evaluation protocol.

Dataset 1 [9] shows a moderate class imbalance of approximately 65% non-default to 35% default. Dataset 2[10] is more imbalanced toward its positive class (approximately 69% approved to 31% not approved) and is considerably smaller in sample size (614 records compared with 6,843). Dataset 2 also contains missing values across several features, addressed through imputation. These characteristics are treated throughout this thesis as genuine data-related limitations rather than errors to be corrected, and their implications for interpreting the comparative results are discussed directly.

1.6 Contributions of the thesis

This study makes two main contributions. First, to researchers, it provides a controlled, like-for-like comparison of five widely used classifier families evaluated under one identical nested cross-validation protocol, which reduces the risk that reported performance differences reflect inconsistencies in evaluation methodology rather than genuine differences between the algorithms themselves. Because this comparison is repeated independently across two datasets that differ substantially in size, class balance, and available features, the study also offers direct evidence on how far a conclusion drawn from one dataset can reasonably be expected to generalize to another a question that is often assumed rather than tested in the existing literature.

Second, to practitioners, this study offers practical guidance on selecting a model under real-world explainability constraints. Rather than treating the choice between a high-performing but opaque ensemble model and a simpler, more transparent linear model as a fixed, binary trade-off, this thesis demonstrates that SHAP can be applied consistently across both types of model, which allows a lending institution to weigh the modest performance differences observed between models against the practical costs of deploying and maintaining an explainability pipeline for a more complex model, rather than assuming that transparency requires abandoning the better performing option outright.

1.7 Organization of the thesis

This thesis is organized into seven chapters. Chapter 1 covers the introduction, including the background, motivation, problem statement, objectives, scope, and significance. Chapter 2 Comparative Analysis of Machine Learning Approaches for Predicting Loan Default Risk. Chapter 3 describes the methodology and model design. Chapter 4 presents the experimental results. Chapter 5 discusses the findings. Chapter 6 concludes the study. Chapter 7 addresses limitations and future work.

Chapter 2

Literature Review

This chapter reviews the existing research relevant to the aims of this thesis. It begins with the older, statistical methods that dominated credit risk assessment before machine learning became widespread, then reviews the broader shift toward machine learning approaches for credit-risk and loan-default prediction. It then examines each of the five model families used in this thesis in turn Logistic Regression, Support Vector Machines, Random Forest, and the two gradient boosting methods XGBoost and CatBoost looking specifically at how each has been applied in credit-risk contexts in prior work .It then turns to Explainable AI generally and SHAP specifically, the explainability tool used throughout this thesis .The chapter closes by drawing these threads together into a specific statement of the research gap this thesis addresses .

Throughout this review, particular attention is paid to two recurring issues in the existing literature that motivate the design of this thesis: first, whether a given study evaluates its models on more than one dataset, and second, whether explainability methods such as SHAP are applied to every model compared, or only to the single best-performing one. These two issues are returned directly.

2.1Traditional Statistical Approaches to Loan Default Prediction

Aslam et al. (2019) presents loan lending has been playing a significant role in the financial world throughout the years. Although it is quite profitable and beneficial for both the lenders and the borrowers. It does, however, carry a great risk, which in the domain of loan lending is referred to as Credit risk. Industry experts and Researchers around the world assign individuals with numerical scores known as credit scores to measure the risk and their creditworthiness. Throughout the years machine learning algorithms have been used to calculate and predict credit risk by evaluating an individual's historical data. This study reviews the present literature on models predicting risk assessment that use machine learning algorithms [11].

Akuji et al. (2026) describes This study optimized loan default prediction by comparing k-nearest neighbor (KNN), random forest (RF), and hybrid methods. The dataset used was preprocessed using simple imputer, label encoder, synthetic minority oversampling technique (SMOTE), and correlation-based feature selection on top seven features while grid search cross-validation (GSCV) and random search cross-validation (RSCV) were employed to optimize models. Before tuning, RF achieved perfect performance (100% accuracy, 99.8% precision, 100% recall, 99.9% F1, 1.000 area under curve (AUC)), outperforming untuned KNN (99.2% accuracy, 96.2% precision, 99.8% recall, 98.0% F1, 0.997 AUC) and hybrid (99.8% accuracy, 99.1% precision, 99.9% recall, 99.5% F1). After tuning, RF maintained same results, confirmed by 10× nested CV stability (F1=0.9997±0.0002) and McNemar tests showing equivalence to RF_RSCV (p=1.0000). KNN improved marginally in precision (96.2%→99.8%) but declined in recall, while hybrid dropped slightly across metrics. Partial dependence plots confirm RF's dominance stems from three key features (lump_sum_payment,

property_value, co-applicant_credit_type), validated by business impact analysis showing minimal errors against KNN/hybrid. RF_GSCV's perfection reflects true generalization, not overfitting, establishing it as the production-ready gold standard. Future work can address static dataset limitation by incorporating dynamic time-series data with online learning, concept drift detection, and real-time macroeconomic features to enhance real-world generalizability.[12]

Tariq et al. (2019) shows the purpose of this study is to provide comprehensive research and to develop a model to predict loan defaults. This kind of model becomes inevitable as the issue of bad loans are very much critical in the financial sector especially in micro financing banks of various underdeveloped and developed countries. To cope with this problem a comprehensive literature review was done to study the significant factors that led to this issue. Moreover, these reviewed studies were critically focused towards applying data mining techniques for the prediction and classification of the loan defaults. This study used methodologies named KDD, CRISP-DM and SEMMA [13].

Bhandary& Ghosh et al. (2025) find out Compares LDA and LR against SVM, XGBoost, RF, and DNN on the UCI Taiwan credit-card dataset (n = 30,000), explicitly framing LDA/LR as the "traditional statistical" baseline.[14]

2.2Machine Learning Approaches to Credit/Loan Default Prediction

Kvamme et al. (2018) investigated mortgage default prediction using machine learning techniques, particularly Convolutional Neural Networks (CNNs). The study utilized consumer transaction data, including account balances and transaction activities, from a Norwegian mortgage portfolio. The findings demonstrated that CNN-based models could effectively identify borrowers at risk of mortgage default, achieving a ROC-AUC of 0.918, while combining the neural network with Random Forest improved the ROC-AUC to 0.926. The study demonstrated the potential of deep learning techniques for extracting useful patterns from financial transaction data for credit-risk assessment.[15]

Sigrist & Hirnschall et al. (2019) show proposed a Gradient Tree-Boosted Tobit model, known as Grabit, for default prediction. Their approach combined gradient tree boosting with a Tobit-based framework to address challenges associated with default prediction and censored data. The study demonstrated that machine learning-based boosting techniques can provide effective predictive performance in credit-risk modelling. This research highlights the growing importance of ensemble learning techniques for improving the accuracy of default prediction compared with traditional statistical approaches.[16]

Albanesi & Vamossy et al. (2019**)** describes examined consumer loan default prediction using a deep learning approach. Their study focused on whether modern machine learning techniques could improve the prediction of consumer defaults compared with conventional credit-scoring approaches. The findings showed that deep learning models could effectively capture complex relationships among borrower characteristics and improve the identification of borrowers who are likely to default. This study provides evidence that advanced machine learning techniques can be useful for improving consumer credit-risk assessment.[17]

Zhou et al. (2019) investigated default prediction in peer-to-peer (P2P) lending using high-dimensional data and machine learning techniques. The study compared several gradient-boosting algorithms, including Gradient Boosting Decision Trees (GBDT), XGBoost, and LightGBM. The researchers focused on the challenges associated with high-dimensional and imbalanced loan datasets. Their findings demonstrated that machine learning-based boosting algorithms can effectively improve the prediction of loan default. This study is particularly relevant to credit-default prediction because it demonstrates the effectiveness of ensemble-based models for handling complex lending data.[18]

Aniceto et al. (2020) show the application of machine learning techniques to consumer creditworthiness using a Brazilian bank's loan database. The study emphasized that accurate credit-risk evaluation is important because incorrect lending decisions can result in significant financial losses. Different machine learning classification techniques were evaluated to determine their ability to distinguish between creditworthy and risky borrowers. The research demonstrated the applicability of machine learning methods to consumer credit-risk assessment and showed that data-driven models can provide valuable support for lending decisions [19].

Turiel et al. (2020) examined loan acceptance and default prediction in peer-to-peer lending using artificial intelligence techniques. The study investigated different machine learning approaches for predicting whether a loan would be accepted and whether the borrower would subsequently default. The researchers demonstrated that artificial intelligence-based models can capture complex relationships within P2P lending data and provide useful predictions for credit-risk management. The study therefore highlights the potential of machine learning as an alternative to conventional lending decision-making methods.[20]

Xu et al. (2021) developed a machine learning methodology for predicting loan defaults in the Chinese P2P lending market. The study used data from Renrendai.com and applied Random Forest (RF), Extreme Gradient Boosting Tree (XGBoost/XGBT), Gradient Boosting Model (GBM), and Neural Network (NN) models. The dataset contained 54,477 observations and 28 features, while SMOTE was used to address class imbalance. The results showed that all four models achieved accuracy and kappa values above 90%, with Random Forest outperforming the other models. The study also identified borrower verification, income, employment, and asset-related information as important factors associated with default risk [21].

Liu et al. (2022) investigated loan default prediction in the online credit market using machine learning techniques. The study compared several machine learning algorithms, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest, with traditional Logistic Regression. The research demonstrated that machine learning models can effectively capture nonlinear relationships within borrower and loan characteristics and can provide improved predictive performance compared with conventional statistical models. The findings support the use of machine learning techniques for automated credit-risk evaluation and loan-default prediction.[22]

Li and Wu et al. (2024) investigated loan default prediction using explainable machine learning techniques. The study compared nine commonly used machine learning models and found that the Random Forest model demonstrated efficient and stable predictive performance. To

improve the interpretability of the prediction process, the researchers applied SHapley Additive exPlanations (SHAP) to identify and explain the most important factors influencing loan default. Their findings showed that borrower age and work experience were among the significant factors associated with default risk. This study is particularly relevant to modern credit-risk research because it combines predictive performance with model interpretability, addressing the limitations of using complex machine learning models as black boxes.[23]

2.3Random Forest in Credit Risk Modelling

Random Forest builds a large collection of individual decision trees, each one trained on a randomly resampled subset of the training data (a technique known as bootstrap aggregation, or bagging) and, at each split within each tree, considering only a randomly selected subset of the available features rather than all of them. The forest's final prediction is then formed by combining the predictions of all its individual trees, typically by majority vote for classification tasks. This combination of randomness at both the data level and the feature level is specifically designed to reduce the tendency of a single decision tree to overfit its training data, while retaining the ability to model non-linear relationships and interactions between features that a single tree, or a linear model, can capture.

Uddin et al. (2022) applied Random Forest to credit default prediction using Chinese micro-enterprise credit data, focusing on both predictive accuracy and model interpretability. The study employed relative variable importance to identify the contribution of individual predictors and found that Random Forest provided reliable and efficient performance across multiple evaluation measures. The findings also showed that non-traditional variables, including legal-representative information and internal non-financial factors, could significantly improve credit-risk classification, demonstrating the suitability of Random Forest for multidimensional credit risk modelling.[24]

Liu et al. (2022) investigated the application of Random Forest, Support Vector Machine, and K-Nearest Neighbor for predicting default probability in the online credit market. Using data from Chinese P2P lending, the study compared these machine learning models with Logistic Regression based on AUC, accuracy, and Brier score. The results demonstrated that the machine learning approaches achieved significantly better prediction performance than the traditional logistic model, highlighting the effectiveness of Random Forest and other nonlinear algorithms in online credit-risk assessment [25].

Zhou et al. (2023) proposed a two-stage credit-scoring framework based on Random Forest using loan data from 3,045 Chinese small firms. The researchers first applied SMOTE to address class imbalance and subsequently employed Random Forest to construct predictive credit features. Their findings demonstrated that financial, non-financial, and macroeconomic factors all contributed to default assessment, while the proposed methodology achieved stronger performance than other machine learning models in robustness tests [26].

Gao et al. (2023) examined farmers' loan default prediction using Lending Club data combined with severe-weather information. The study compared Artificial Neural Networks, Gradient

Boosting Trees, and Random Forest and found that all three models were effective in predicting loan defaults, with Random Forest achieving the highest accuracy among the evaluated models. The researchers further used SHAP to interpret the model predictions and identified severe weather as an economically relevant factor in predicting farmers' loan default risk [27].

2.4 Gradient Boosting Methods: XGBoost and CatBoost

XGBoost and CatBoost both belong to the family of gradient boosting methods, which build an ensemble of decision trees sequentially, rather than independently and in parallel as Random Forest does. Each new tree added to a gradient boosting ensemble is trained specifically to correct the errors made by the ensemble of trees built so far, rather than being trained independently on a resampled version of the full dataset. This step-by-step, error-correcting training process frequently gives gradient boosting methods an advantage in raw predictive accuracy over bagging-based ensembles such as Random Forest, particularly on structured, tabular data of the kind examined in both datasets used in this thesis.

Chen & Guestrin (2016) introduced XGBoost as a scalable and efficient tree-boosting framework designed to improve both predictive performance and computational efficiency. The method incorporates sparsity-aware learning, regularization, and optimized tree construction, making it suitable for large-scale classification and regression problems [28].

Prokhorenkova et al. (2018) proposed CatBoost, a gradient boosting algorithm specifically designed to handle categorical features effectively. The study introduced order boosting and an innovative categorical feature-processing mechanism to reduce prediction shift and target leakage, demonstrating strong predictive performance compared with existing boosting methods [29].

Odegu et al. (2020) applied Extreme Gradient Boosting (XGBoost) to bank loan default prediction using borrower application and demographic information. The study evaluated the model using accuracy, precision, recall, F1-score, and ROC-AUC, demonstrating the potential of XGBoost for identifying risky borrowers and supporting loan credit approval decisions [30].

Xia et al. (2020) investigated loan default prediction in peer-to-peer lending using CatBoost and incorporated both traditional credit information and narrative data. The findings showed that CatBoost effectively captured information from borrowers' data and outperformed an industry benchmark under different experimental settings, highlighting its potential for credit-risk prediction [31].

Across the literature reviewed in this section, a consistent theme emerges gradient boosting methods tend to produce the strongest raw predictive accuracy among the model families commonly compared in credit-risk research, but this advantage typically comes at the cost of reduced direct interpretability relative to Logistic Regression, again illustrating the accuracy-interpretability trade-off introduced in Chapter 1 and motivating the explainability approach reviewed in the next section.

2.5 Explainable AI and SHAP in Credit Risk Modelling

As machine learning models used in financial decision-making have grown more complex, the practical need to explain their individual decisions has grown correspondingly. This need is not purely academic. Regulators in multiple jurisdictions, auditors conducting internal model risk reviews, and the borrowers directly affected by a lending decision all typically require some clear, specific reason for why a loan application was approved or rejected a requirement that a complex ensemble model or a kernel-based classifier cannot satisfy directly from its own internal structure. This practical requirement has driven the growth of a research field generally referred to as Explainable AI (XAI), which develops tools specifically designed to open up these otherwise opaque, "black box" models after they have already been trained, without requiring any change to the underlying model itself.

Bussmann et al. (2020) proposed an Explainable AI framework for fintech risk management, particularly focusing on credit risk in peer-to-peer lending. The study applied an XGBoost model to data from 15,045 small and medium-sized enterprises and used Shapley values through SHAP to explain individual credit-risk predictions. The XGBoost model achieved higher predictive performance than logistic regression, while SHAP provided feature-level explanations for each prediction. The findings demonstrated that explainable machine learning can improve both the predictive capability and interpretability of credit risk models [32].

de Lange et al. (2022) developed an explainable AI model for credit default prediction using a dataset of unsecured consumer loans from a Norwegian bank. The study combined LightGBM with SHAP to identify and interpret the factors influencing credit-risk predictions. By providing explanations of the model's decisions, the approach addressed the need for transparency and interpretability in banking credit assessment. The study highlighted the practical value of SHAP-based explanations for supporting reliable and accountable credit decision-making [33].

Li and Wu (2024) investigated loan default prediction using nine commonly used machine learning models and applied SHAP to interpret the factors influencing default risk. Their results showed that the Random Forest model provided efficient and stable predictive performance among the evaluated approaches. SHAP analysis identified borrower characteristics such as age and years of employment as important factors associated with loan default probability. The study demonstrated how explainable machine learning can complement predictive modelling by providing meaningful interpretations of the factors driving loan default predictions[34].

Winkel and Reichenbach (2024) applied explainable machine learning survival analysis to credit risk assessment using more than 350,000 loans from the Estonian Bondora peer-to-peer lending platform. The study developed classical and extreme gradient-boosted Cox models to estimate loan default risk and used SHAP to interpret the model outputs. The boosted model outperformed both the classical Cox model and the platform's existing rating system, while SHAP revealed nonlinear relationships and interaction effects among borrower characteristics. The findings demonstrated the potential of explainable survival models for improving both credit risk prediction and interpretability [35].

Lin et al. (2025) investigated automobile loan credit risk using machine learning models and integrated XGBoost with SHAP for model interpretation. Using loan data from a Chinese automobile finance company, the study compared several machine learning algorithms and found that XGBoost achieved the strongest overall predictive performance. SHAP was then employed to quantify the contribution of individual features to loan-risk predictions, identifying credit level, credit score, and disbursed loan amount as important factors. The study demonstrated that integrating SHAP with machine learning can enhance both predictive performance and transparency in automobile loan risk assessment [36].

SHAP has been applied in a range of financial contexts beyond loan default and approval prediction specifically. In credit card fraud detection, for example, SHAP has been used in combination with Random Forest and XGBoost to identify which transaction-level features most strongly signal fraudulent activity, and to compare how consistently different models agree on which features matter most for a given prediction. Studies of this kind generally report that SHAP produces feature-importance rankings that align well with what a domain expert in the relevant field would expect to see, which lends practical credibility to using SHAP as a genuine explanation tool, rather than merely a technical exercise disconnected from real-world decision-making. This body of applied SHAP-in-finance literature directly supports this thesis's decision to use SHAP as its primary explainability method, and its findings are compared directly against the SHAP results generated in this study.

2.6 Research Gap

While machine learning effectively predicts financial risk, existing studies face a key trade-off: high-accuracy ensemble models function as opaque black boxes, whereas transparent statistical models lack predictive power for complex data. Furthermore, most research relies on single datasets and basic validation methods that risk overfitting and biased performance estimates. To address these gaps, this research applies a Nested Cross-Validation framework across two distinct financial datasets evaluating five diverse algorithms to balance generalization with predictive accuracy, while incorporating SHAP interpretability to deliver fully transparent, reliable risk assessment models.

Chapter 3

Methodology

3.1 System Architecture

This study follows a quantitative, comparative, empirical research design. The overall goal is to compare how five different machine learning models behave when asked to solve the same kind of problem predicting whether a loan will end in default, or whether a loan application will be approved on two datasets that differ in size, feature set, and class balance. The word "comparative" matters here: this is not a study that tries to build the single best possible model for one dataset. Instead, it deliberately holds the evaluation method fixed across five models and two datasets, so that any differences observed in the results can be attributed to genuine differences between the models and the datasets, rather than to inconsistencies in how each model happened to be tested.

The research design can be described as a five-stage pipeline, and each stage is described in detail later in this chapter. In the first stage, each dataset is examined and cleaned, with missing values handled, categorical variables identified, and the target variable defined clearly. In the second stage, the data for each model is split using a nested cross-validation scheme, so that the process of choosing the best hyperparameters for a model is kept strictly separate from the process of measuring how well that model performs. In the third stage, once the nested cross-validation process has given a reliable performance estimate and identified typical hyperparameters for each model, one final version of each model is trained for closer analysis. In the fourth stage, this final model is evaluated using a standard set of classification metrics. In the fifth and final stage, SHAP is applied to each final model to explain which features drove its decisions. This same five-stage process is applied independently to Dataset 1 and to Dataset 2, using the same code structure, the same evaluation metrics, and the same explainability method, so that the results for the two datasets can be placed side by side in Chapter 4 without needing to account for methodological differences between them.

It is worth being explicit about one design decision here. Some earlier studies in this area only apply a single train-test split, tune the model on that same split, and report the resulting score. This is simpler to implement, but it risks reporting an optimistic score, because the same data that is used to pick the best settings is also used to judge how good those settings are. This study avoids that shortcut by using nested cross-validation throughout, described in full ,even though it takes considerably more computation time to run five models with a full hyperparameter grid search, twice (once per dataset), each wrapped inside two layers of cross-validation.

This diagram shows how two separate loan datasets, the loan approval dataset (614 rows) and the loan details dataset (6843 rows) both flow through the same nested cross-validation pipeline (5 outer folds, 3 inner folds). That same pipeline trains five models (CatBoost, Random Forest, XGBoost, SVM, Logistic Regression), each of which is evaluated on the outer folds using

accuracy, ROC-AUC, F1, precision, and recall. The results from both datasets are then brought together into a cross-dataset comparison table, and the final step is a SHAP analysis to interpret feature importance (for the tree -based models CatBoost, Random Forest, and XGBoost).

So basically, this flowchart gives an end-to-end view of the whole workflow from raw data all the way to final interpretability.

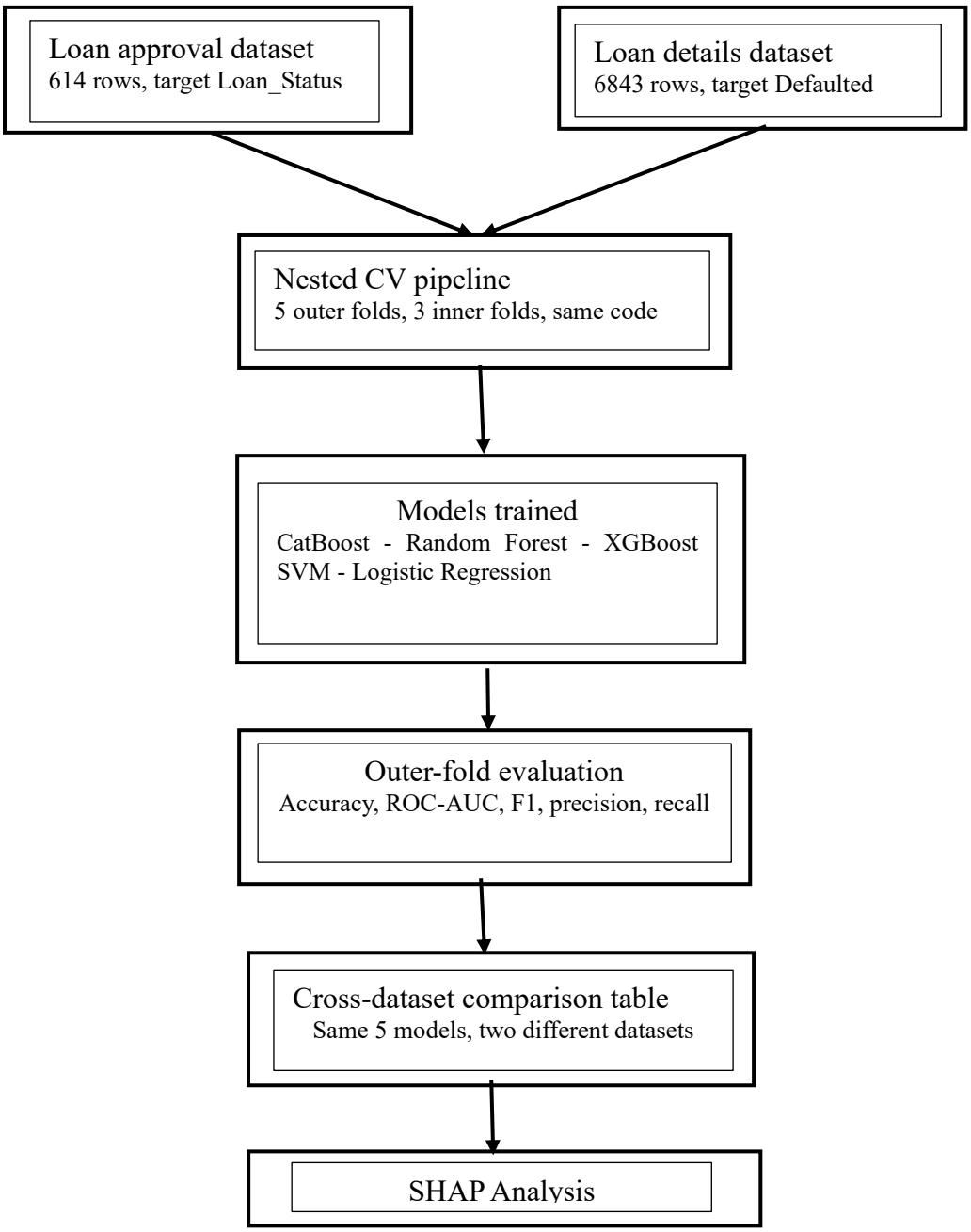


Figure 2.1: System Architecture

3.2 Dataset 1

Dataset 1 is a general-purpose loan default dataset. It contains 6,843 individual loan records, and each record corresponds to one borrower. The dataset includes both demographic information about the borrower (age, marital status, education level, employment status) and financial information related to the loan itself (income, credit score, loan amount). The target variable in this dataset is called Defaulted. It is a binary variable, meaning it can only take one of two values: 1, meaning the borrower defaulted on the loan (failed to repay it as agreed), or 0, meaning the borrower did not default.

Table 3.1:Class Distribution of Dataset 1 (Defaulted Variable)

Class	**Label (Defaulted)**	**Record Count**	**Percentage**
Non-default	0	4,468	65.3%
Default	1	2,375	34.7%
Total	—	**6,843**	**100%**

The dataset contains eight columns in total before any preprocessing: an identifier column (Name), four numeric predictor columns, three categorical predictor columns, and the target column. The Name column is dropped before modelling, since it is a unique text identifier for each borrower and carries no predictive information about default risk; including it would not help any model and could, in principle, encourage a model to memorize individual borrowers rather than learn general patterns.

Table 3.2: Dataset 1 feature description

Feature	**Type**	**Description**
Age	Numeric	Age of the loan applicant, in years
Income	Numeric	Annual income of the applicant, in the dataset's currency unit
Credit_Score	Numeric	A numeric credit score assigned to the applicant
Loan_Amount	Numeric	The amount of money requested for the loan
Marital_Status	Categorical	Single, Married, Divorced, or Widowed
Education_Level	Categorical	High School, Associate Degree, Bachelor Degree, Master Degree, or Doctorate
Employment_Status	Categorical	Employed, Self-Employed, Unemployed, or Student
Defaulted (target)	Binary (0/1)	1 = the borrower defaulted; 0 = the borrower did not default

An inspection of this dataset found no missing values in any of the eight columns, and no duplicated rows when checking across all columns simultaneously. There is some repetition in the Name column specifically, where the same first-and-last name combination appears attached to different Age, Income, Credit Score, and Loan Amount values; this is treated as a normal occurrence rather than a data quality problem, since real-world populations frequently contain people who share the same name, and since the Name column is dropped from the modelling process regardless. Because the dataset has no missing values, the imputation step described is only relevant for Dataset 2.

3.3 Dataset 2 Description

Dataset 2 is a smaller, publicly available loan-approval dataset, widely known in the machine learning community as the Loan Prediction dataset. It contains 614 loan applications. Unlike Dataset 1, which is framed around predicting default after a loan has already been issued, Dataset 2 is framed around predicting whether a loan application should be approved in the first place. The target variable is Loan _Status, encoded here as a binary variable: 1 means the loan was approved, and 0 means it was not approved.

Table 3.3: Class Distribution of Dataset 2 (Loan_Status Variable)

Class	**Label (Loan_Status)**	**Record Count**	**Percentage**
Approved	1	422	68.7%
Not Approved	0	192	31.3%
Total	—	**614**	**100%**

Dataset 2 contains twelve usable columns after dropping the Loan_ID identifier column, which, like the Name column in Dataset 1, is a unique identifier with no predictive value and is removed before modelling.

Table 3.4: Dataset 2 feature description

Feature	Type	Description
Gender	Categorical	Male or Female
Married	Categorical	Yes or No
Dependents	Categorical	Number of dependents: 0, 1, 2, or 3+
Education	Categorical	Graduate or Not Graduate
Self_Employed	Categorical	Yes or No
ApplicantIncome	Numeric	Monthly income of the primary applicant
CoapplicantIncome	Numeric	Monthly income of a co-applicant, if one exists (0 if none)
LoanAmount	Numeric	Requested loan amount, expressed in thousands
Loan_Amount_Term	Numeric	Length of the loan repayment period, in months
Credit_History	Numeric (0/1)	Whether the applicant has a history of meeting credit obligations (1) or not (0)
Property_Area	Categorical	Urban, Semiurban, or Rural
Loan_Status (target)	Binary (0/1)	1 = loan approved; 0 = loan not approved

A key structural difference between the two datasets, beyond size and class balance, is that Dataset 2 contains missing values in six of its twelve columns: Gender (13 missing), Married (3 missing), Dependents (15 missing), Self_Employed (32 missing), LoanAmount (22 missing), and Loan_Amount_Term (14 missing). Credit_History itself is also missing in 50 records. This is a meaningful fraction of a 614-row dataset for example, the 50 missing Credit_History values represent about 8% of all records so the way these missing values are handled, described in Section 3.4, has a real effect on how much usable information reaches each model. Dataset 2 also includes one feature that has no equivalent in Dataset 1: Credit_History. As Chapter 4 will show, this single feature turns out to have an outsized influence on model performance and on the SHAP-based explanations produced for Dataset 2, which is one of the central findings compared across the two datasets in Section 4.6.

3.4 Data Preprocessing

Both datasets pass through the same three-stage preprocessing pipeline before being used to train any model: handling of missing values, encoding of categorical variables, and scaling of numeric variables. A critical implementation detail, repeated for every one of the five models and both datasets, is that every preprocessing step is fitted only on the training portion of the

data within each cross-validation fold, and then applied unchanged to the corresponding validation or test portion. This avoids what is known as data leakage a subtle error where information from the test set (for example, the median value of a feature calculated across the whole dataset, including test rows) accidentally influences the training process, which would make the reported performance look better than it really is.

3.4.1 Handling of Missing Values

Dataset 1 requires no missing-value handling, since it contains no missing values in any column. Dataset 2, as described, contains missing values across six columns. Two different imputation strategies were used, depending on whether the column is numeric or categorical.

For the numeric columns with missing values (LoanAmount, Loan_Amount_Term, and Credit_History), missing values were replaced with the median value of that column, computed using only the training fold in each cross-validation split. The median was chosen over the mean because it is less sensitive to outliers for example, LoanAmount has a small number of unusually large loan requests, and using the mean here would have pulled the imputed value upward, away from what a typical applicant's loan amount actually looks like.

For the categorical columns with missing values (Gender, Married, Dependents, and Self_Employed), missing values were replaced with the most frequently occurring value in that column, again computed using only the training fold. This is sometimes called "mode imputation." For example, if the vast majority of applicants in the training fold are Male, a missing Gender value is filled in as Male, on the reasoning that this is the single most probable value given no other information.

3.4.2 Encoding of Categorical Variables

Machine learning models generally require numeric input, so the categorical text columns in both datasets (such as Marital _Status in Dataset 1, or Property_ Area in Dataset 2) must be converted into a numeric form before most models can use them. This study uses one-hot encoding for this purpose. One-hot encoding works by creating a new binary (0/1) column for each possible category of a feature, except one category, which is dropped to avoid redundancy this dropped category, sometimes called the reference category, is represented implicitly by all of its one-hot columns being 0 at once.

As a concrete example, the Marital _Status column in Dataset 1 has four possible values: Single, Married, Divorced, and Widowed. After one-hot encoding with the first category dropped, this single column becomes three new columns: Marital_Status_Married, Marital_Status_Divorced, and Marital_Status_Widowed.A record where Marital_Status_Married, Marital_Status_Divorced, and Marital_Status_Widowed are all 0 is understood to represent a Single applicant, since Single was the dropped reference category. The encoder was also configured to handle unknown categories gracefully during prediction if a category appears in the test data that was never seen during training (which can happen with small subsets in cross-validation folds), it is encoded as all-zero rather than causing the model to fail.

One model in this study is treated differently. CatBoost has built-in support for categorical features and does not require them to be one-hot encoded; internally, it uses a different technique for handling categories that is designed to work well without inflating the number of columns. For CatBoost specifically, the categorical columns are passed to the model directly as category labels, with their column positions identified explicitly, rather than being one-hot encoded first.

3.4.3 Scaling of Numeric Variables

Numeric features such as Income, Credit_Score, Age, and Loan_Amount are measured on very different scales Income might range into the tens of thousands, while Age ranges from perhaps 18 to 80. Some models are sensitive to this difference in scale, while others are not.

Logistic Regression and SVM-RBF both fall into the sensitive category, because they rely on distances or weighted sums of feature values, and a feature measured in the tens of thousands (such as Income) would otherwise dominate a feature measured in single or double digits (such as Age) simply because of its larger numeric range, not because it is actually more important. For these two models, numeric features were standardised using a technique called StandardScaler, which transforms each numeric feature so that it has a mean of 0 and a standard deviation of 1 within the training fold. This is done by subtracting the training-fold mean from each value and dividing by the training-fold standard deviation.

Random Forest, XGBoost, and CatBoost, on the other hand, are tree-based models. A decision tree makes its splits by asking questions like the actual scale of the number does not affect how the tree decides where to split, only the order of the values matters. Because of this, scaling was skipped entirely for the three tree-based models, since it would add computational overhead without changing how these models behave.

3.5 Nested Cross-Validation

This section explains, in detail, exactly what "nested" cross-validation means in this study, since the term is sometimes used loosely elsewhere. Nested cross-validation involves two loops, one inside the other, and each loop has a distinct job.

3.5.1 Outer loop

The outer loop exists purely to produce an honest, unbiased estimate of how well a model performs on data it has never seen. The full dataset is split into 5 folds using stratified sampling, meaning the proportion of default (or approved) cases is kept roughly equal across all 5 folds. The outer loop then runs 5 rounds. In each round, one of the 5 folds is set aside completely as the outer test fold, and the remaining four folds are combined into the outer training fold. The outer test fold is only ever used once per round, at the very end, purely to calculate that round's Accuracy, Precision, Recall, F1-score, and ROC-AUC. It plays no role whatsoever in choosing hyperparameters.

3.5.2 Inner Loop

Inner loop exists purely to choose the best hyperparameters for a model, and it operates entirely inside the outer training fold from the current outer round it never sees the outer test fold at

all. Within that outer training fold, the inner loop performs its own 5-fold stratified split, and for every combination of hyperparameters in a predefined grid (Section 3.6), it trains a model on four of these inner folds and measures ROC-AUC on the fifth, repeating this five times and averaging the result. Whichever hyperparameter combination achieves the best average inner-loop ROC-AUC is selected as the "best" configuration for that particular outer round.

Once the inner loop has selected the best hyperparameters for the current outer round, a single model is trained one more time, this time using the full outer training fold (all four of the outer folds combined, not just the inner sub-splits) with the chosen hyperparameters. This model is then evaluated, for the first and only time, on the outer test fold that was set aside at the start of the round. This entire process inner-loop tuning, followed by training on the full outer training fold, followed by a single evaluation on the outer test fold is repeated five times in total, once for each of the five outer folds acting as the test fold. The final performance figures reported for each model in Chapter 4 are the average of these five outer-round results, along with the standard deviation across the five rounds, which gives a sense of how stable that model's performance .

To make the purpose of this two-loop structure concrete: imagine a simpler alternative where a single 5-fold cross-validation is used both to search hyperparameters and to report the final score. In that simpler setup, the fold used to measure a hyperparameter combination's score is the same fold used to decide the score is good the hyperparameter search is effectively allowed to peek at the data it will later be judged on, and it will tend to favour hyperparameter combinations that happen to fit that particular fold especially well, some of which is just luck rather than genuine generalization ability. Nested cross-validation closes this loophole by keeping the outer test fold completely untouched during the entire hyperparameter search process, at the cost of considerably more computation, since an inner 5-fold search must now be repeated fully inside each of the five outer folds.

3.5.2.1 Hyperparameter Tuning

Grid search was the tuning method used for every model in this study. Grid search works by defining a fixed list of candidate values for each hyperparameter of interest, forming every possible combination of these values (the grid), training and evaluating a model for each combination, and selecting whichever combination achieved the best score in this case, the best average ROC-AUC across the inner loop's 5 folds.

ROC-AUC was chosen as the selection metric, rather than Accuracy, because Accuracy can be misleading on the moderately imbalanced datasets used in this study, whereas ROC-AUC evaluates a model's ability to separate the two classes across every possible decision threshold, not just the default threshold of 0.5.

The hyperparameters chosen for each model reflect the settings that are generally understood, from the wider machine learning literature reviewed in Chapter 2, to have the largest effect on that model's behaviour.

Table 3.5: Hyperparameter search space per model

Model	Hyperparameters Searched	What Each Setting Controls
Logistic Regression	C: [0.01, 0.1, 1, 10, 100]	C is the inverse of regularisation strength smaller values apply stronger regularisation, which shrinks the model's coefficients toward zero and can help prevent overfitting.
SVM-RBF	C: [0.1, 1, 10]; gamma: [scale, auto]	C controls how much the model tolerates misclassified points during training; gamma controls how far the influence of a single training example reaches in the RBF kernel a small gamma gives a smoother, simpler decision boundary, while a large gamma allows a more complex, tightly fitted boundary.
Random Forest	n_estimators: [100, 200]; max_depth: [5, 10, None]	n_estimators is the number of individual decision trees in the forest; max_depth limits how many levels deep each tree can grow, which directly controls how much a single tree can overfit to the training data.
XGBoost	n_estimators: [100, 200, 300]; max_depth: [3, 5, 7]; learning_rate: [0.01, 0.05, 0.1]	n_estimators is the number of boosting rounds; max_depth limits the depth of each individual tree added during boosting; learning_rate controls how much each new tree is allowed to correct the errors of the previous trees, with smaller values requiring

		more rounds but often generalising better.
CatBoost	iterations: [200]; depth: [4, 6]; learning_rate: [0.05, 0.1]	iterations is the number of boosting rounds (fixed at 200 here to keep the manual nested search computationally feasible); depth and learning_rate serve the same role as in XGBoost.

A practical implementation issue arose specifically with CatBoost. CatBoost's constructor accepts a cat_features argument that identifies which columns should be treated as categorical, and this argument does not interact well with the standard grid search tool used for the other four models, because that tool internally creates copies of the model object in a way that conflicts with how CatBoost stores this setting. To work around this, the inner-loop search for CatBoost was implemented as a manual loop: for every candidate hyperparameter combination, the code manually splits the outer training fold into 5 inner folds, trains a CatBoost model on each combination, records the average inner-loop ROC-AUC, and selects the combination with the best average score reproducing the exact same nested logic used for the other four models, just without relying on the automated grid search tool.

3.6 Model Training and Final Model Fitting

Nested cross-validation, as described above, produces five separate models per algorithm per dataset one for each outer fold each potentially trained with slightly different hyperparameters, since the inner loop is free to choose different best settings in different outer rounds depending on which data happened to fall into that round's inner folds. This is by design: the goal of nested cross-validation is to estimate expected performance, not to produce a single deployable model directly. For the purposes of downstream analysis in this thesis in particular, generating SHAP explanations in.

To produce this final model, the hyperparameter combination that was selected most frequently across the five outer folds during nested cross-validation is identified for each algorithm. A fresh 80/20 stratified train-test split is then created from the full dataset, and one final model is trained on the 80% training portion using this most-common hyperparameter combination. This final model is the one used for the SHAP explanations presented in Section 4.5, and for the overfitting check described next.

For the two boosting models, XGBoost and CatBoost, an additional diagnostic step was performed beyond the standard train-test split. The 80% training portion was further split into a smaller training subset and a validation subset, and the model was trained while recording its loss (log loss) and accuracy on both the training subset and the validation subset after every boosting round. This produces a round-by-round record of how the model's performance evolves as more trees are added. If training performance keeps improving while validation

performance plateaus or gets worse, this is a direct sign of overfitting the model is starting to fit patterns specific to the training subset rather than patterns that generalize. This diagnostic is discussed with concrete numbers.

3.7 Evaluation Metrics

Several evaluation metrics were used to compare the performance of the five classification models. Since the datasets have some class imbalance, no single metric was considered sufficient.

- **Accuracy:** Accuracy is defined as the proportion of correctly classified observations (both defaulters and non-defaulters correctly identified) out of the total number of observations evaluated:

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

where TP represents true positives, TN represents true negatives, FP represents false positives, and FN represents false negatives.

Although Accuracy is easy to understand, it can be misleading when the target classes are imbalanced. Therefore, it was considered together with other metrics.

- **Precision:** Precision measures the proportion of predicted positive cases that are actually positive.

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

A higher Precision means that the model produces fewer false positive predictions.

- **Recall:**Recall measures the proportion of actual positive cases that are correctly identified by the model.

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

Recall is particularly important in loan default prediction because failing to identify a borrower who will default can result in financial loss

- **F1-score :**F1-score is the harmonic mean of Precision and Recall.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

It provides a balance between Precision and Recall.

- **ROC-AUC:** ROC-AUC measures the ability of a classifier to distinguish between the two classes across different classification thresholds. A higher ROC-AUC indicates better discrimination ability.
  ROC-AUC was used as the primary model-selection metric during the inner-loop hyperparameter tuning and was also used for the final model comparison.
  The selected metrics are consistent with the methodology specified in the thesis outline.

3.8 Model Explainability with SHAP

SHAP, short for SHapley Additive exPlanations, is the explainability method used throughout this study. It was applied identically to the final model of every one of the five algorithms, on both datasets, so that the feature-importance findings can be fairly compared across model families rather than being available only for whichever model happened to perform best.

The underlying idea behind SHAP comes from a branch of mathematics called cooperative game theory, and specifically from a concept called the Shapley value, named after the economist Lloyd Shapley. The original problem the Shapley value was designed to solve has nothing to do with machine learning: imagine a group of people working together on a task, and a reward is produced at the end. The Shapley value gives a fair way to split that reward among the group members, based on how much each member's presence actually contributed, averaged over every possible order in which the members could have joined the group. SHAP, introduced by Lundberg and Lee (2017), adapts this idea to machine learning by treating each input feature of a prediction as one of these group members, and the model's prediction for a specific case as the "reward" to be fairly divided [37]. The result is a SHAP value for every feature, for every individual prediction, indicating how much that feature pushed the prediction above or below the model's average prediction across the whole dataset.

This gives two complementary views of a model's behaviour, both of which are used. The global view, produced by averaging the absolute SHAP value of each feature across every prediction in the test set, shows which features matter most to the model overall this is the basis for the feature-importance rankings presented. The local view, available for any single prediction, shows exactly which features pushed that specific prediction toward or away from the positive class, and by how much this kind of explanation is what would let a loan officer say to an individual applicant, "your application was affected most strongly by X and Y," rather than only being able to describe the model's behaviour in the abstract.

The specific SHAP implementation used depends on the type of model, because different model structures allow for different calculation shortcuts. For the three tree-based models (Random Forest, XGBoost, and CatBoost), this study uses TreeExplainer, a fast, exact method that takes advantage of a tree's internal structure to calculate SHAP values efficiently, without needing to approximate anything. For Logistic Regression, LinearExplainer was used, which similarly takes advantage of the model's simple linear structure to compute exact SHAP values

quickly. For SVM-RBF, no such structural shortcut is available once a non-linear RBF kernel is applied, so KernelExplainer was used instead a slower, model-agnostic method that works by systematically perturbing feature values and observing how the model's output changes, using a background sample of the training data to represent a realistic set of "typical" feature values to compare against.

3.9 Tools and Implementation Environment

All experiments in this study were implemented in Python 3. The following libraries were the main tools used at each stage of the pipeline.

Table 3.6: Software libraries and their role in this study

Library	Role in this work
pandas / NumPy	Loading, cleaning, and manipulating the two datasets as tabular data structures
scikit-learn	Building the preprocessing pipelines, stratified k-fold splitting, grid search, and implementing Logistic Regression, SVM-RBF, and Random Forest
xgboost	Implementing the XGBoost model, including its round-by-round evaluation history used in Section 4.4
catboost	Implementing the CatBoost model, including its built-in categorical feature handling
shap	Generating TreeExplainer, LinearExplainer, and KernelExplainer explanations for all five models
matplotlib / seaborn	Producing confusion matrices and other visual summaries used internally during analysis

All experiments were run in a cloud-based Jupyter notebook environment (Google Colab). Datasets were stored on Google Drive and mounted into the notebook environment at the start of each session, which kept the data available across separate working sessions without needing to be re-uploaded each time. No GPU acceleration was required for any of the five models used in this study, since all five operate efficiently on tabular data of this size using standard CPU-based computation.

Chapter 4

Experimental Results and Analysis

This chapter follows a fixed sequence: the experimental setup; nested cross-validation performance for model selection; final held-out test-set performance, kept explicitly separate from the cross-validation figures (4.2); ROC curve comparison and confusion matrices for the final test-set models (4.3); SHAP-based explainability results (4.4); and a closing comparison between the two datasets (4.5). report two genuinely different things and should not be confused with one another: reports cross-validation results, used for model comparison and selection during the research process; Section 4.3 reports how the final, already-selected models actually performed on data held out and untouched until this point.

4.1 Cross-Validation Performance (Outer Loop)

Table 4.1 and Table 4.2 report the mean and standard deviation of each metric across the five outer folds of nested cross-validation, for Dataset 1 and Dataset 2 respectively. These are nested cross-validation results, used here for model comparison and selection they are not the final held-out test-set results.

Table 4.1 shows the nested cross-validation results for Dataset 1. CatBoost performs best overall, with the highest Recall (0.4072), F1-score (0.4949), and ROC-AUC (0.7431), making it strongest at identifying defaulters with a balanced trade-off. XGBoost leads slightly on Accuracy (0.7118), while Random Forest achieves the highest Precision (0.6957) but at the cost of much lower Recall (0.2792). Overall, CatBoost offers the most balanced and reliable performance for Dataset 1.

Table 4.1: Nested cross-validation results, Dataset 1 (mean ± standard deviation across 5 outer folds)

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Logistic Regression	0.7112 ± 0.0135	0.6342 ± 0.0480	0.3979 ± 0.0328	0.4887 ± 0.0225	0.7316 ± 0.0198
SVM-RBF*	0.7079 ± 0.0099	0.6258 ± 0.0262	0.3962 ± 0.0242	0.4847 ± 0.0194	0.7315 ± 0.0157
Random Forest	0.7071 ± 0.0091	**0.6957 ± 0.0331**	0.2792 ± 0.0348	0.3982 ± 0.0362	0.7378 ± 0.0122
XGBoost	**0.7118 ± 0.0169**	0.6503 ± 0.0486	0.3739 ± 0.0490	0.4726 ± 0.0403	0.7360 ± 0.0138
CatBoost	0.7117 ± 0.0172	0.6326 ± 0.0425	**0.4072 ± 0.0282**	**0.4949 ± 0.0290**	**0.7431 ± 0.0165**

Table 4.2: Nested cross-validation results, Dataset 2 (mean across 5 outer folds)

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Logistic Regression	0.8030	0.7901	0.9716	0.8714	**0.7461**
SVM-RBF	**0.8078**	0.7880	**0.9858**	**0.8758**	0.7344
Random Forest	0.7801	0.7889	0.9288	0.8531	0.7539
XGBoost	0.7769	**0.7911**	0.9195	0.8499	0.7407
CatBoost	0.7866	0.7906	0.9383	0.8579	0.7417

Table 4.2 shows the nested cross-validation results for Dataset 2. SVM-RBF performs best overall, with the highest Accuracy (0.8078), Recall (0.9858), and F1-score (0.8758), making it the strongest at correctly identifying approved applications. Logistic Regression leads on ROC-AUC (0.7461), while XGBoost achieves the highest Precision (0.7911). Overall, SVM-RBF offers the most balanced and reliable performance for Dataset 2.

On Dataset 1, all five models cluster tightly together, with ROC-AUC ranging from 0.7315 (SVM) to 0.7431 (CatBoost). The standard deviations shown in Table 4.1are generally small relative to the differences between models for example, CatBoost's ROC-AUC of 0.7431 ± 0.0165 and Random Forest's 0.7378 ± 0.0122 overlap once their variability is accounted for, meaning the two models cannot be confidently distinguished as different in ROC-AUC on this dataset alone. On Dataset 2, overall performance is markedly higher across every metric, with Random Forest achieving the best ROC-AUC (0.7539) and SVM-RBF achieving the highest Accuracy (0.8078) and Recall (0.9858).

4.2 Single train test split performance

This section reports the performance of the final trained models, described in Section 3.7, on the held-out 20% test set that was set aside before any tuning took place and was not used anywhere in the nested cross-validation process in Section 4.1. This is the only section in this thesis that reports test-set results in this sense, and it is kept structurally separate from Section 4.1 so the two are never confused.

Table 4.3: Final test-set results, Dataset 1

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Logistic Regression	**0.7100**	0.6226	0.4168	0.4994	0.7277
SVM-RBF	0.7027	0.6504	0.3095	0.4194	0.7193
Random Forest	0.7034	**0.6751**	0.2800	0.3958	0.7316
XGBoost	0.7027	0.6104	0.3958	0.4802	0.7340
CatBoost	0.7093	0.6185	**0.4232**	**0.5025**	**0.7356**

Table 4.4: Final test-set results, Dataset 2

Model	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.8618	0.8400	0.9882	0.9081
SVM-RBF	0.8455	0.8235	0.9882	0.8984
Random Forest	0.8455	0.8438	0.9529	0.8950
XGBoost	0.8374	0.8495	0.9294	0.8876
CatBoost	0.8374	0.8421	0.9412	0.8889

Table 4.3 (Dataset 1) On the final test set, CatBoost performs best overall, with the highest Recall (0.4232), F1-score (0.5025), and ROC-AUC (0.7356). Logistic Regression achieves the highest Accuracy (0.7100), while Random Forest leads on Precision (0.6751). CatBoost remains the most balanced performer, consistent with the cross-validation results.

Table 4.4 (Dataset 2) Logistic Regression performs best overall, with the highest Accuracy (0.8618), F1-score (0.9081), and tied-highest Recall (0.9882). Random Forest achieves the highest Precision (0.8438). Overall, Logistic Regression offers the strongest and most balanced performance on Dataset 2's test set.

Comparing the final test-set figures in Tables 4.3 and 4.4 against the nested cross-validation figures in Tables 4.1 and 4.2 gives a check on how well each model generalises beyond the cross-validation process itself. On Dataset 1, the test-set ROC-AUC for every model falls within about one percentage point of its cross-validation ROC-AUC (for example, CatBoost: 0.7431 in cross-validation versus 0.7356 on the test set; XGBoost: 0.7360 versus 0.7340), which is a sign of good, consistent generalisation rather than overfitting to the cross-validation folds. On Dataset 2, the test-set ROC-AUC scores are noticeably higher than the corresponding cross-validation scores for every model (for example, Logistic Regression: 0.7461 in cross-validation versus 0.8517 on the test set). Given the much smaller size of Dataset 2 (614 records, with a test set of only 123 records), this gap most likely reflects the particular 20% test split drawn for final evaluation being somewhat easier than the average cross-validation fold, rather than indicating a problem with the models themselves; this sensitivity to the specific split is a direct consequence of Dataset 2's small sample size, discussed further as a limitation.

4.3 ROC Curve Comparison

Figure 4.1 shows the ROC curves for all five final models on Dataset 1's test set, plotted on a single set of axes for direct visual comparison.

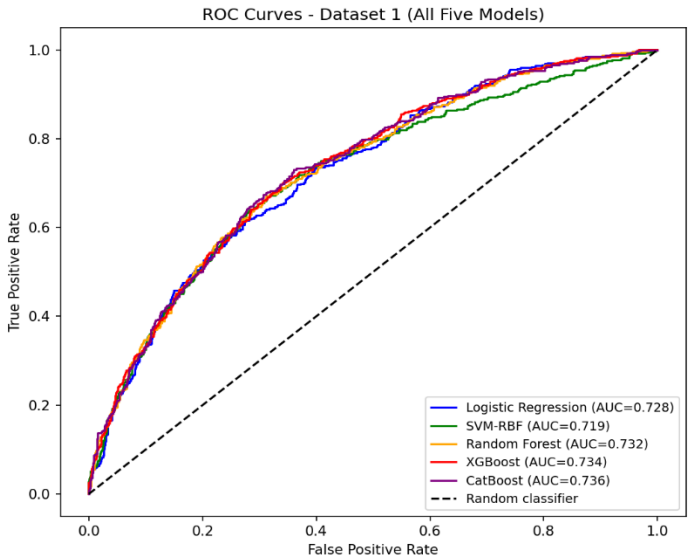


Figure 4.1: ROC curves, Dataset 1 (all five models, final test set)

On Dataset 1, CatBoost, XGBoost, and Random Forest trace almost identical curves across nearly the entire false-positive-rate range, sitting slightly above Logistic Regression, with SVM-RBF trailing marginally below the other four models across most of the curve. None of the five curves dominates clearly and consistently at every threshold; the ordering among the top four models shifts slightly depending on which part of the curve is examined, which is consistent with how closely bunched their ROC-AUC values are in Table 4.4.

Figure 4.2 shows the equivalent comparison for Dataset 2.

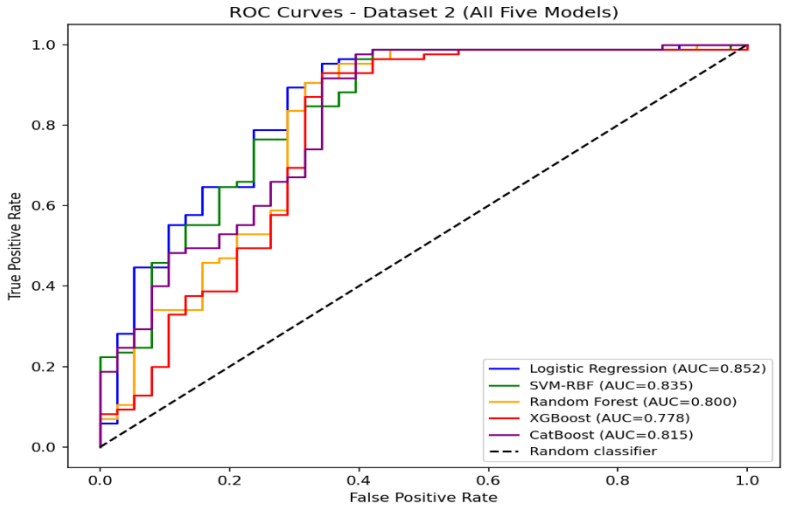


Figure 4.2: ROC curves, Dataset 2 (all five models, final test set)

On Dataset 2, Logistic Regression's curve sits above every other model across almost the entire false-positive-rate range, which matches its highest test-set ROC-AUC in Table 4.5 (0.8517). SVM-RBF follows closely behind. XGBoost's curve is visibly the lowest through the low-to-middle false-positive-rate range, consistent with it recording the lowest test-set ROC-AUC of

the five models on this dataset (0.7777), although all five curves converge toward the top-right corner at higher false-positive rates, where every model eventually catches almost all positive cases at the cost of also raising more false alarms.

4.4 Confusion Matrix

Figure 4.3 and Figure 4.4 show the confusion matrices for all five final models on the Dataset 1 and Dataset 2 test sets respectively.

Figure 4.3 presents the confusion matrices of the five final models evaluated on the held-out Dataset 1 test set (n = 1,369), with 894 actual "No Default" instances and 475 actual "Default" instances held constant across all five panels, permitting direct comparison of classification behaviour independent of test-set composition.

All five models exhibit a consistent asymmetry: correct classification of the majority class (No Default) substantially exceeds correct classification of the minority class (Default), reflecting the moderate class imbalance inherited from the training distribution. Random Forest attains the highest true-negative count (830/894, 92.8%) but the lowest true-positive count (133/475, 28.0%), indicating a decision boundary that is conservative with respect to the default class. SVM-RBF exhibits a similar profile, correctly identifying 147/475 (30.9%) defaults. CatBoost achieves the highest true-positive count among all five models (201/475, 42.3%), accompanied by the lowest false-negative count (274), indicating comparatively superior sensitivity to the minority class. XGBoost (188/475, 39.6%) and Logistic Regression (198/475, 41.7%) occupy an intermediate position between the tree-ensemble extremes.

Overall accuracy is closely clustered across all five models (approximately 70.3–70.9%), such that accuracy alone does not meaningfully differentiate model quality; the more diagnostic distinction lies in recall on the default class, where CatBoost and Logistic Regression outperform Random Forest and SVM-RBF, consistent with the class-imbalance argument advanced in Section 3.8 for reporting Precision, Recall, F1-score, and ROC-AUC alongside Accuracy.

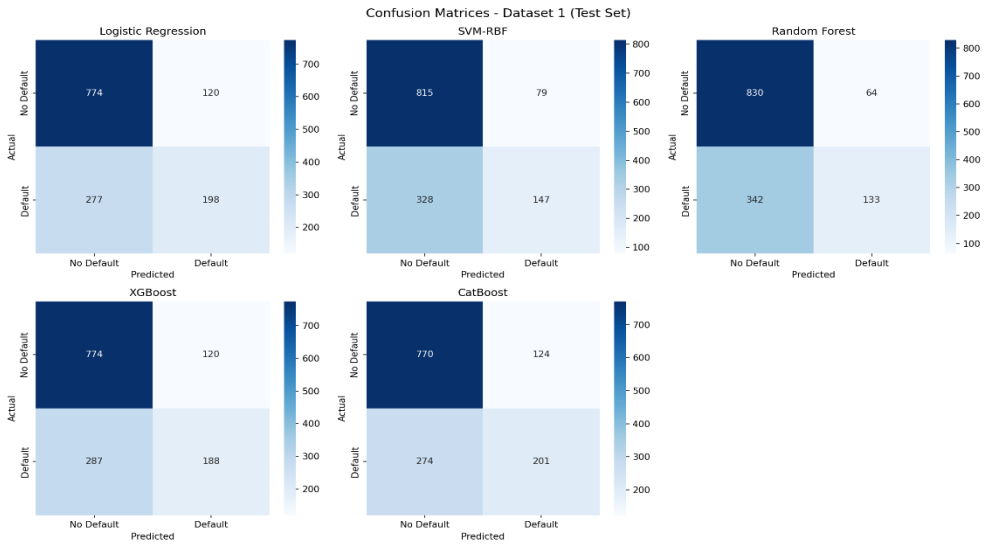


Figure 4.3: Confusion matrices, Dataset 1 (final test set, n = 1,369)

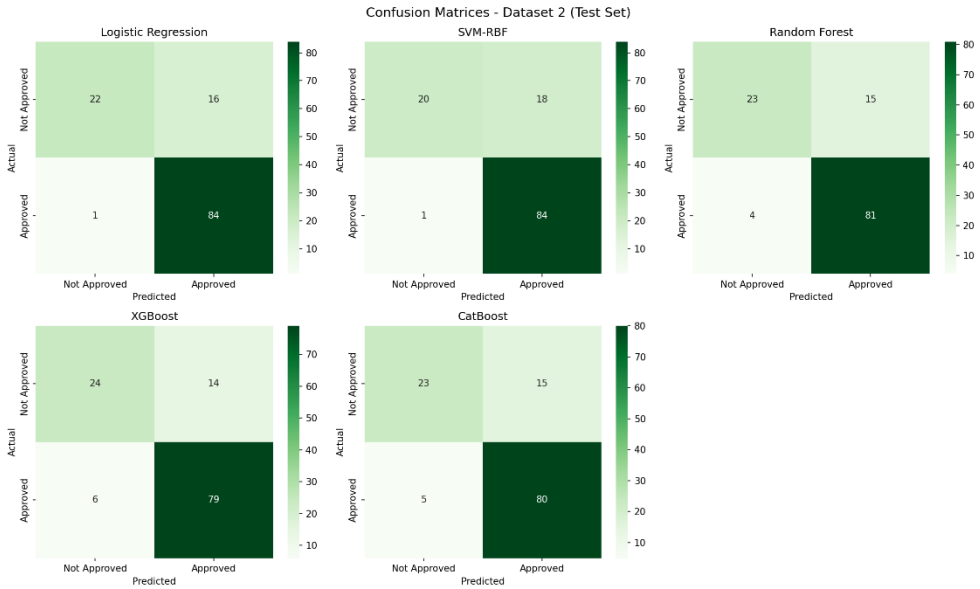


Figure 4.4: Confusion matrices, Dataset 2 (final test set, n = 123)

Figure 4.4 presents the confusion matrices of the five final models evaluated on the held-out Dataset 2 test set (n = 123), comprising 38 actual Not Approved instances and 85 actual "Approved" instances, held constant across all five panels.

All five models exhibit markedly higher sensitivity to the majority Approved class than to the minority Not Approved class, a pattern consistent with the 68.7%/31.3% class imbalance established in Section 3.4. Logistic Regression and SVM-RBF jointly attain the highest recall on the positive class, each correctly classifying 84/85 (98.8%) approved applications with only a single false negative, though this is accompanied by comparatively weaker specificity, correctly identifying only 22/38 (57.9%) and 20/38 (52.6%) of non-approved applications respectively. CatBoost achieves the strongest balance between the two classes, correctly classifying 80/85 (94.1%) approvals alongside 23/38 (60.5%) non-approvals, the highest specificity among the tree-based models. Random Forest and XGBoost show a broadly comparable profile, with specificity of 60.5% and 63.2% respectively, but at a moderate cost to positive-class recall (95.3% and 92.9%).

The consistently elevated false-positive rate relative to the false-negative rate across all five models indicates a shared tendency toward optimistic misclassification of non-approved applicants as approved, an asymmetry with direct practical relevance in a credit-risk context, where such errors carry materially different downstream costs than the converse, and which further substantiates the rationale for reporting Precision, Recall, and F1-score rather than relying on Accuracy alone.

4.5 Explainability Results (SHAP)

SHAP was applied to the final, tuned version of every one of the five models on Dataset 1. Across all five models, the same three features consistently rank as the most influential: Income, Credit_Score, and Loan_Amount. Higher Income and higher Loan_Amount are both associated with an increased predicted probability of default, while a higher Credit_Score is

associated with a decreased predicted probability of default, matching the expected direction from domain knowledge. The three categorical demographic features (Marital_Status, Education_Level, Employment_Status) consistently contribute far less to model predictions than the four numeric features, across every model tested.

4.5.1 Dataset 1- Loan Approval

Figure 4.5.1a and Figure 4.5.1b jointly characterize CatBoost's decision structure on Dataset 1. Credit_History yields a mean absolute SHAP value of 0.853, an order of magnitude greater than any other predictor, with the beeswarm plot indicating that its absence induces a large negative contribution to the predicted log-odds of approval, whereas its presence induces a comparably large positive contribution. Property _Area (0.217) and Education (0.071) constitute a distant second tier of relevance, while CoapplicantIncome, LoanAmount, Married, ApplicantIncome, Loan_Amount_Term, Gender, Self Employed, and Dependents contribute only marginally to model output, indicating that CatBoost's decision boundary is dominated almost exclusively by a single categorical feature.

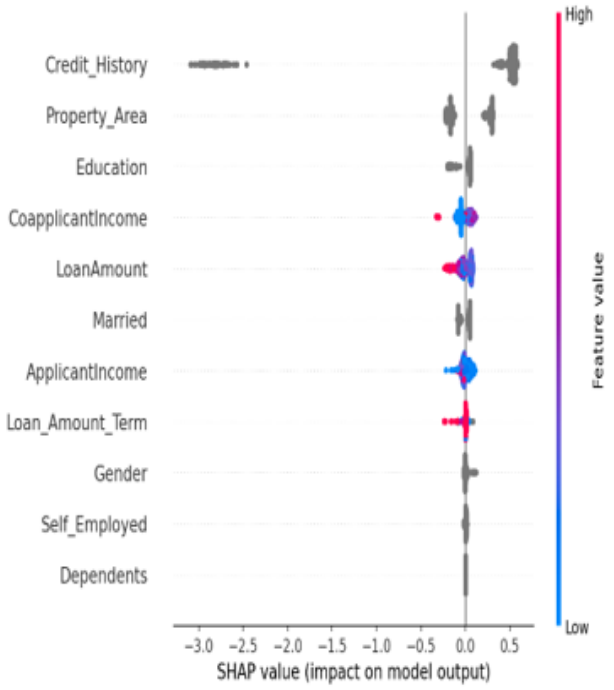


Figure 4.5.1a SHAP summary (beeswarm) plot, CatBoost, Dataset 1 (Loan Approval)

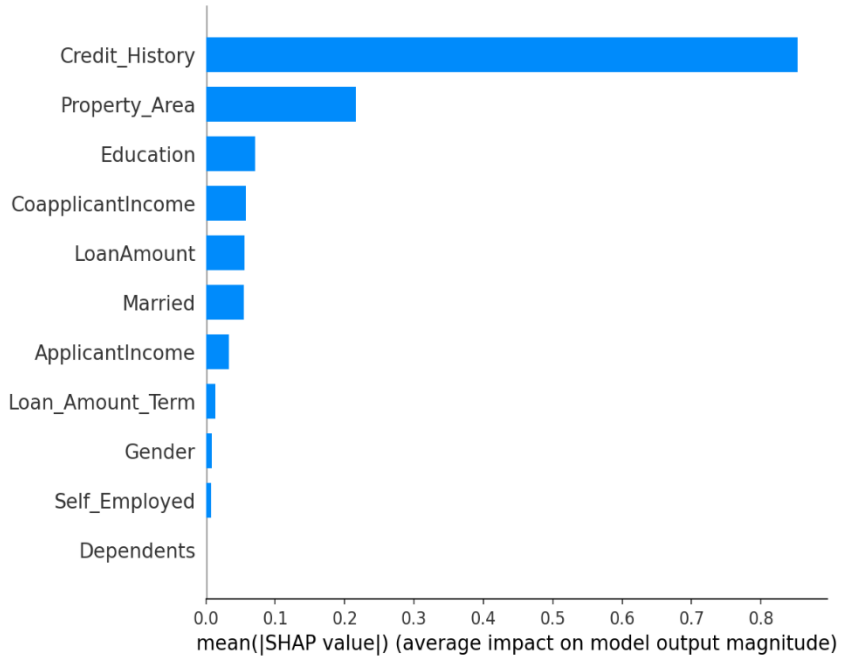


Figure 4.5.1b SHAP mean |value| bar plot, CatBoost, Dataset 1 (Loan Approval)

Figure 4.5.2a and 4.5.2b For Random Forest, the one-hot-encoded Credit_History_0.0 and Credit_History_1.0 indicators jointly dominate feature attribution (mean |SHAP| of 0.125 and 0.055 respectively), reproducing the same directional pattern observed for CatBoost, though with attribution magnitudes distributed across the two binarised indicator columns rather than a single native categorical feature. LoanAmount and ApplicantIncome follow as secondary contributors (0.037 and 0.037), with the remaining predictors Married, Gender, Self_Employed, Dependents, and the one-hot Property_Area and Education categories exerting comparatively minor influence. The convergence with the CatBoost result, despite the differing feature encoding, supports the conclusion that this ranking reflects genuine signal in the data rather than an artefact of a particular model's categorical handling.



Figure 4.5.2a SHAP summary (beeswarm) plot, Random Forest, Dataset 1 (Loan Approval)

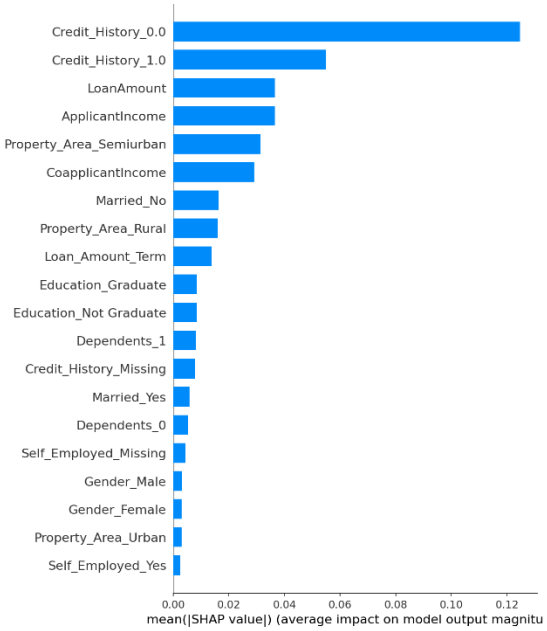


Figure 4.5.2b SHAP mean |value| bar plot, Random Forest, Dataset 1 (Loan Approval)

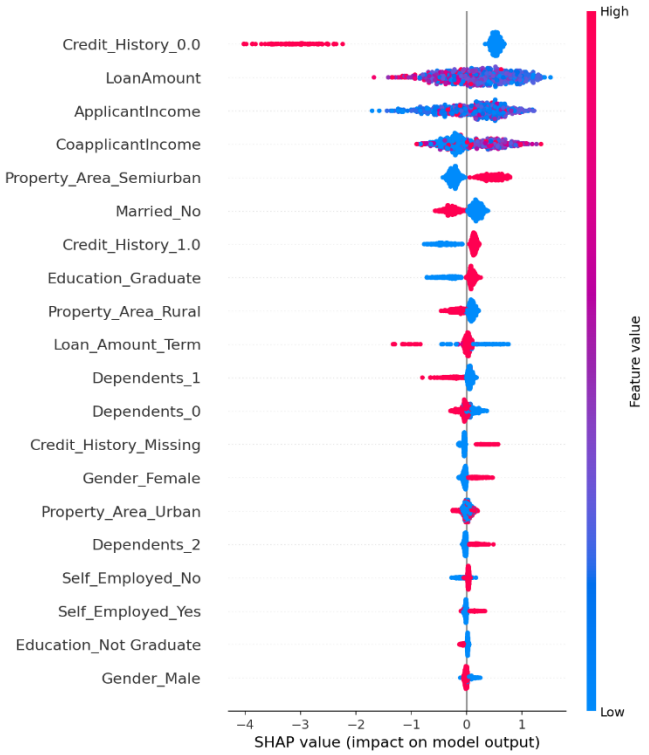


Figure 4.5.3a SHAP summary (beeswarm) plot, XGBoost, Dataset 1 (Loan Approval)

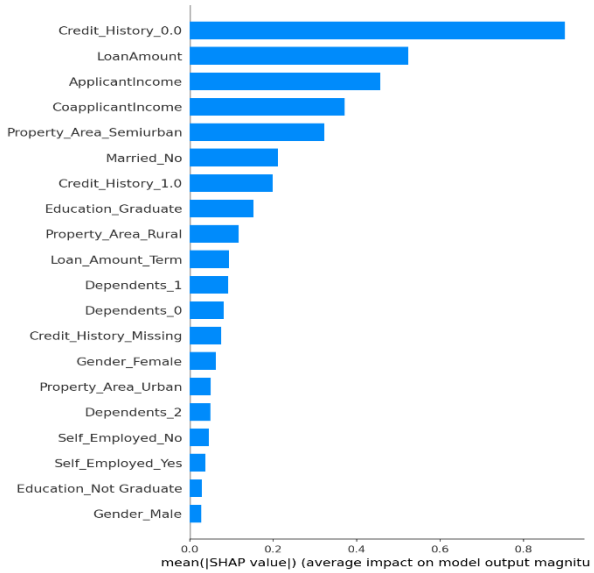


Figure 4.5.3b SHAP mean |value| bar plot, XGBoost, Dataset 1 (Loan Approval)

XGBoost exhibits the same dominant-feature structure, with Credit_History_0.0 attaining a mean |SHAP| of 0.899, again substantially exceeding all remaining predictors. Unlike CatBoost and Random Forest, however, LoanAmount (0.523), ApplicantIncome (0.456), and CoapplicantIncome (0.371) retain comparatively higher secondary importance in the XGBoost fit, suggesting that this model's tree splits extract a larger marginal contribution from the continuous financial variables even though Credit_History remains the primary determinant of predicted approval probability.

Taken together, Figures 4.5.1–4.5.3 provide convergent evidence across three structurally distinct tree ensembles that Credit_History is the principal determinant of predicted loan approval, with Property_Area, Education, and the applicant's financial variables (income and loan amount) contributing secondary, model-dependent effects, while demographic variables such as Gender, Self_Employed, and Dependents remain uniformly unimportant across all three models.

4.5.2 Dataset 2 Loan Details

On Dataset 2, CatBoost attributes the greatest mean |SHAP| to Income (0.619), followed by Credit_Score (0.435), Loan_Amount (0.260), and Age (0.126). The three categorical predictors Employment_Status, Marital_Status, and Education_Level each register mean |SHAP| values below 0.013, indicating that CatBoost's default-risk predictions on this dataset are governed almost entirely by the applicant's continuous financial and demographic attributes rather than by categorical status variables.

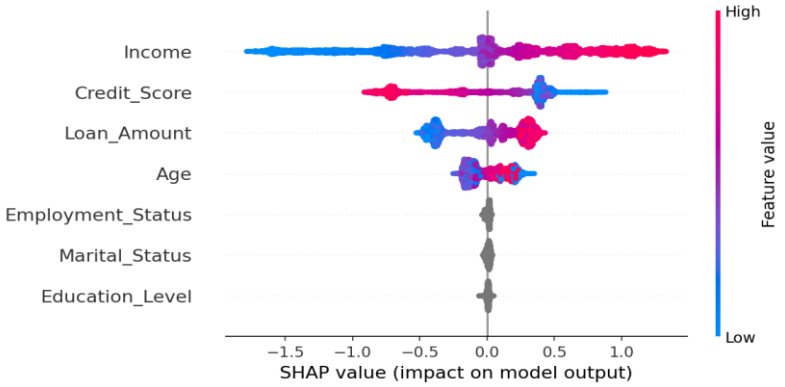


Figure 4.6.4a SHAP summary (beeswarm) plot, CatBoost, Dataset 2 (Loan Details)

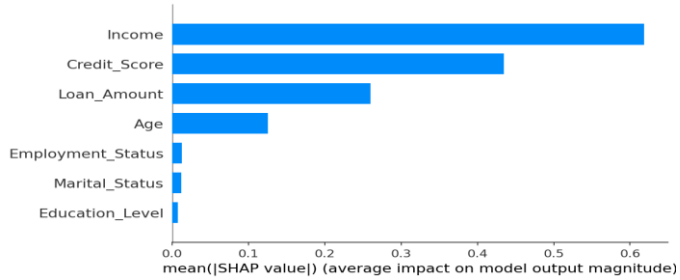


Figure 4.6.4b SHAP mean |value| bar plot, CatBoost, Dataset 2 (Loan Details)

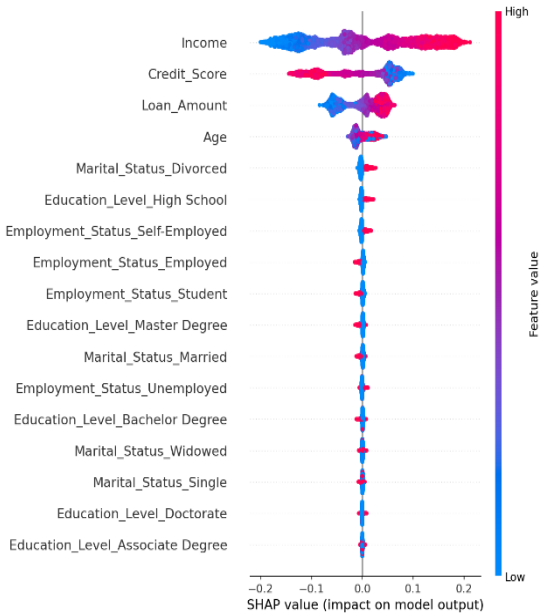


Figure 4.5.5a SHAP summary (beeswarm) plot, Random Forest, Dataset 2 (Loan Details)

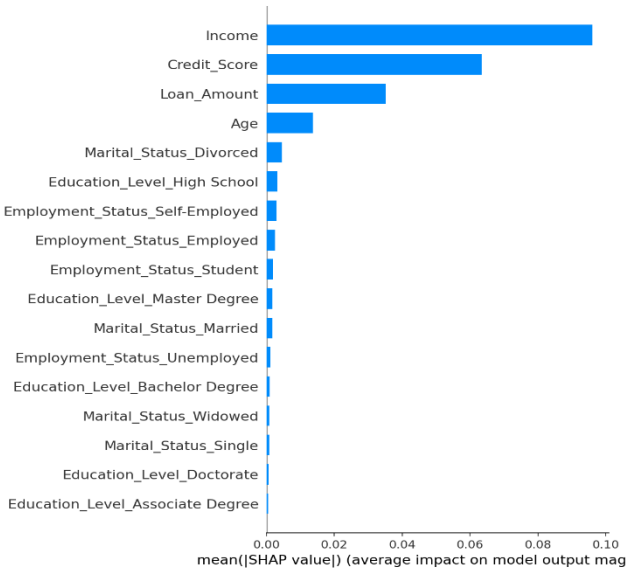


Figure 4.6.5b SHAP mean |value| bar plot, Random Forest, Dataset 2 (Loan Details)

Figure 4.5.5a and 4.5.5b for Random Forest reproduces an identical feature ranking to CatBoost on Dataset 2, with Income (0.096), Credit_Score (0.063), Loan_Amount (0.035), and Age (0.014) as the four leading predictors, followed by a long tail of one-hot-encoded categorical indicators (Marital_Status, Education_Level, Employment_Status) each contributing under 0.005. The absolute magnitudes are smaller than those of CatBoost owing to differences in the underlying prediction scale, but the relative ordering of feature importance is preserved.

Figure 4.5.6a and 4.5.6b for XGBoost likewise ranks Income (0.670), Credit_Score (0.441), Loan_Amount (0.288), and Age (0.179) as the dominant predictors of default risk, with Education_Level, Marital_Status, and Employment_Status categories each contributing mean |SHAP| values below 0.05. The consistent primacy of Income and Credit_Score across all three tree-based models, irrespective of ensemble architecture, indicates that these two variables carry the principal predictive signal for default risk in Dataset 2

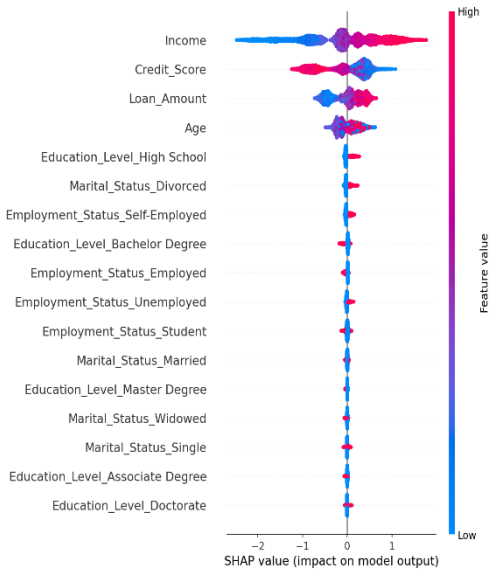


Figure 4.5.6a SHAP summary (beeswarm) plot, XGBoost, Dataset 2 (Loan Details)

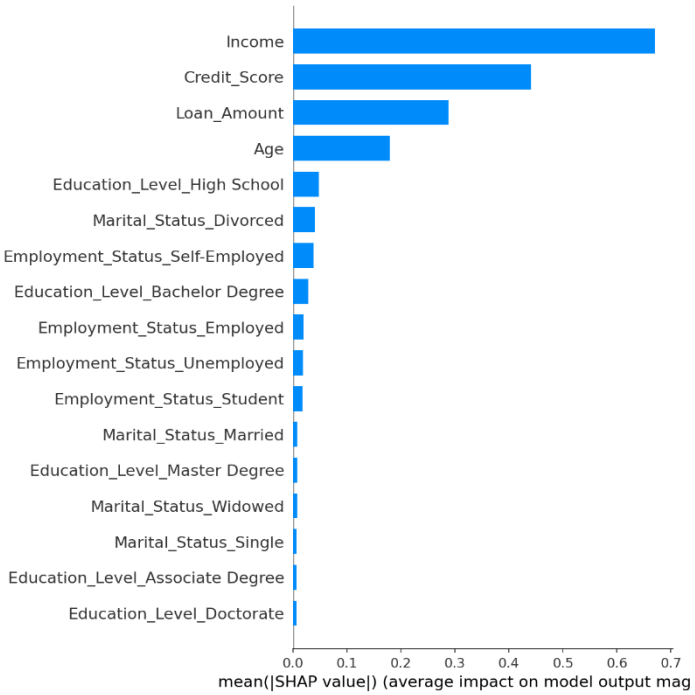


Figure 4.5.6b SHAP mean |value| bar plot, XGBoost, Dataset 2 (Loan Details)

Chapter 5
Discussion

The comparative analysis of machine learning approaches for predicting loan default risk highlights the significant potential of data-driven techniques in enhancing the accuracy and reliability of credit-risk assessment. Loan default prediction is inherently a complex classification problem because borrowers exhibit diverse financial, demographic, and behavioral characteristics, while the relationships between these factors and default outcomes are often nonlinear. Traditional statistical approaches, such as Logistic Regression, generally assume a predefined relationship between predictor variables and the probability of default. Although such models offer simplicity and interpretability, their predictive capability may be constrained when dealing with high-dimensional datasets, complex feature interactions, and nonlinear patterns. In contrast, machine learning algorithms are capable of learning intricate relationships directly from historical lending data, making them increasingly valuable for modern credit-risk management.

Among the commonly applied machine learning techniques, tree-based ensemble models such as Random Forest, Gradient Boosting, XGBoost, and LightGBM have demonstrated considerable effectiveness in loan default prediction. These models can capture nonlinear dependencies and interactions among borrower attributes without requiring extensive assumptions about the underlying data distribution. Random Forest is particularly advantageous because of its robustness, ability to handle heterogeneous features, and relatively low sensitivity to overfitting when an appropriate number of trees is used. However, boosting-based algorithms such as XGBoost and LightGBM can provide stronger predictive performance in many credit-risk scenarios by sequentially focusing on observations that are difficult to classify. Their ability to optimize complex decision boundaries makes them particularly suitable for identifying subtle patterns associated with financially vulnerable borrowers.

Nevertheless, higher predictive performance does not necessarily imply that a model is more appropriate for real-world lending decisions. Credit-risk prediction is a high-stakes application in which erroneous classifications may result in financial losses for lending institutions or unfair denial of credit to potentially eligible borrowers. Therefore, model evaluation should extend beyond accuracy alone. Metrics such as precision, recall, F1-score, ROC-AUC, and confusion matrices provide a more comprehensive assessment of classification performance, particularly when the dataset contains an imbalanced distribution between defaulters and non-defaulters. In highly imbalanced credit datasets, a model may achieve high overall accuracy while performing poorly in identifying actual defaulters. Consequently, greater attention should be given to minority-class detection and the balance between false-positive and false-negative predictions.

Another important consideration is the interpretability of machine learning models. While complex ensemble and boosting algorithms can achieve superior predictive performance, their decision-making processes are often difficult to understand. This creates a significant challenge

in financial applications where lenders, regulators, and borrowers may require explanations for credit-related decisions. Explainable Artificial Intelligence (XAI) techniques, particularly SHAP and LIME, can address this limitation by identifying the contribution of individual features to a model's prediction. Such explanations can reveal whether variables such as income, employment status, loan amount, repayment history, debt-to-income characteristics, or credit history are influencing default predictions. Consequently, integrating explainability into machine learning-based credit-risk assessment can improve transparency, increase stakeholder confidence, and support more accountable lending practices.

Overall, the comparative evaluation suggests that no single machine learning algorithm can be considered universally optimal for every loan default prediction problem. Model effectiveness depends on dataset characteristics, feature quality, class distribution, preprocessing strategies, and the evaluation criteria adopted. Therefore, a reliable credit-risk prediction framework should consider not only predictive performance but also robustness, interpretability, and the potential implications of automated decisions. A comparative machine learning approach, combined with explainability techniques, can provide a more comprehensive foundation for developing transparent and reliable loan default prediction systems. This perspective is particularly important for financial institutions seeking to balance predictive accuracy with responsible and trustworthy credit decision-making.

Chapter 6

Conclusion

6.1 Summary of the Study

This thesis set out to address a specific gap identified in the credit-risk machine learning literature: that most comparative studies of classification models for loan default or loan approval prediction evaluate their models on a single dataset, under evaluation protocols that are not always kept strictly separate from the hyperparameter tuning process, and often apply explainability tools such as SHAP to only the single best-performing model rather than consistently across every model compared.

To address this, the study compared five supervised classification models Logistic Regression, Support Vector Machine with a Radial Basis Function kernel (SVM-RBF), Random Forest, XGBoost, and CatBoost on two structurally different loan datasets. Dataset 1 comprised 6,843 loan records with a binary default label, drawn from applicant demographic and financial attributes, and showed a moderate class imbalance of approximately 65% non-default to 35% default. Dataset 2 comprised 614 loan applications with a binary approval label, drawn from a publicly available loan-approval dataset that additionally included a Credit_History feature, and showed a class balance of approximately 69% approved to 31% not approved.

Every model, on every dataset, was trained and tuned using an identical nested cross-validation protocol, with an outer loop providing an unbiased estimate of expected performance and an inner loop performing a grid search for the best hyperparameters, guided by ROC-AUC. A single final model per algorithm per dataset was then trained on a fresh 80/20 train-test split, using the most commonly selected hyperparameters from the nested cross-validation process, and evaluated on this held-out test set using Accuracy, Precision, Recall, F1-score, ROC-AUC, and Log Loss. ROC curves and confusion matrices were produced for all five final models on both datasets, and a dedicated check for overfitting was carried out by tracking training and validation performance directly. Finally, SHAP was applied consistently to the final model of every algorithm, on both datasets, to identify which features most influenced each model's predictions, and to check whether these explanations were consistent both across model families within a dataset and across the two datasets.

6.2 Summary of Key Findings

This work consists of the findings as follows:

1. On Dataset 1, all five models achieved broadly similar performance under nested cross-validation, with ROC-AUC ranging from 0.7315 (SVM-RBF) to 0.7431 (CatBoost), a spread of just over one percentage point.

2. On Dataset 2, overall performance was noticeably higher across every model and every metric than on Dataset 1, with Random Forest achieving the best nested cross-validation ROC-AUC (0.7539) and SVM-RBF achieving the highest Accuracy (0.8078) and Recall (0.9858).

3. Final held-out test-set results were broadly consistent with the nested cross-validation results on Dataset 1 (within about one percentage point of ROC-AUC per model), supporting good generalisation; on Dataset 2, test-set ROC-AUC scores were noticeably higher than the corresponding cross-validation scores for every model, a pattern attributed to the small sample size of Dataset 2 making the specific test split more sensitive to which records it happened to contain.

4. Model ranking by ROC-AUC was not identical across the two datasets: CatBoost led on Dataset 1 and Random Forest led on Dataset 2, though the same top three models (CatBoost, Random Forest, XGBoost) occupied the top three positions on both datasets, and Logistic Regression and SVM-RBF occupied fourth and fifth place on both datasets.

5. Random Forest, when trained with unrestricted tree depth, showed clear signs of overfitting on Dataset 1, with training accuracy near 100 percent and validation accuracy dropping to around 69 percent; once hyperparameters were tuned through the nested cross-validation inner loop, this gap narrowed substantially, to around two percentage points for a similarly tuned XGBoost model.

6. SHAP identified Income, Credit_Score, and Loan_Amount as the three most influential features on Dataset 1, consistently across all five models, with Income and Loan_Amount increasing predicted default risk and Credit_Score decreasing it.

7. SHAP identified Credit_History as the single dominant feature on Dataset 2, with a SHAP contribution substantially larger than any other feature, consistently across all five models, followed by Property_Area and ApplicantIncome.

8. Reading the confusion matrices for the final test-set models, false negatives (missed defaulters) on Dataset 1 remained high across all five models, ranging from about 56.5 percent (CatBoost) to about 70.5 percent (Random Forest) of actual default cases missed at the standard 0.5 probability threshold; false negatives (missed approvable applicants) on Dataset 2 were considerably lower across all five models, ranging from about 1.2 percent (Logistic Regression, SVM-RBF) to about 7.1 percent (XGBoost).

9. The consistent improvement in performance on Dataset 2 relative to Dataset 1 corresponded directly with the presence and dominant SHAP importance of the Credit_History feature, which was available in Dataset 2 but had no equivalent in Dataset 1.

Chapter 7

Future Work

The following areas are proposed for future research :

1. **Dataset Expansion:** Future studies may incorporate larger, more diverse, and longitudinal datasets from multiple financial institutions to improve model generalizability.
2. **Advanced Models:** Deep learning architectures and hybrid machine learning approaches may be investigated to further enhance loan default prediction performance.
3. **Fairness and Bias Analysis:** Future research may evaluate potential demographic and socioeconomic biases to ensure fair and equitable credit-risk predictions.
4. **Explainable AI:** Advanced XAI techniques may be incorporated to improve the transparency and interpretability of complex prediction models.
5. **Real-World Deployment:** The developed models may be integrated into practical credit assessment systems, with consideration of scalability, computational efficiency, data privacy, and continuous model monitoring.
6. **Cost-Sensitive Evaluation:** Future studies may employ cost-sensitive learning and evaluation metrics that account for the different financial consequences of false-positive and false-negative predictions.
7. **Model Robustness:** Further research may examine model stability under changing economic conditions and across different lending environments.

References

1. Beaver, W. H., Correia, M., & McNichols, M. F. (2011). Financial statement analysis and prediction of financial distress. *Foundations and Trends® in Accounting*, *5*(2), 99-173.
2. Dastile, X., Celik, T., & Potsane, M. (2020). Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing*, *91*, 106263.
3. Karimova, N. (2024). Application of AI in Credit Risk Scoring for Small Business Loans: A case study on how AI-based random forest model improves a Delphi model outcome in the case of Azerbaijani SMEs. *arXiv preprint arXiv:2410.05330*.
4. Xie, Y., & Tummala, S. P. (2025). Machine learning for sensor analytics: a comprehensive review and benchmark of boosting algorithms in healthcare, environmental, and energy applications. Sensors (Basel, Switzerland), 25(23), 7294.
5. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, *1*(5), 206-215.
6. Goodman, B., & Flaxman, S. (2017). European Union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, *38*(3), 50-57.
7. Recht, B., Roelofs, R., Schmidt, L., & Shankar, V. (2019, May). Do imagenet classifiers generalize to imagenet?. In *International conference on machine learning* (pp. 5389-5400). PMLR.
8. Bücker, M., Szepannek, G., Gosiewska, A., & Biecek, P. (2022). Transparency, auditability, and explainability of machine learning models in credit scoring. *Journal of the Operational Research Society*, *73*(1), 70-90.
9. https://www.kaggle.com/datasets/rameshmaity/loan-dataset
10. https://www.kaggle.com/datasets/altruistdelhite04/loan-prediction-problem-dataset
11. Aslam, U., Aziz, H. T., Sohail, A., & Batcha, N. K. (2019). An empirical study on loan default prediction models. *Journal of Computational and Theoretical Nanoscience*, *16*(8), 3483-3488.
12. Akuji, I. I., Olanrewaju, A. B., Ahmed, T. A., Alabi, A. J., & Adeyemi, I. B. (2026). A comparative study of classical, bagging, and hybrid methods for optimizing loan default prediction. *Computer Science and Information Technologies*, *7*(2), 179-195.
13. Tariq, H. I., Sohail, A., Aslam, U., & Batcha, N. K. (2019). Loan default prediction model using sample, explore, modify, model, and assess (SEMMA). *Journal of Computational and Theoretical Nanoscience*, *16*(8), 3489-3503.
14. Bhandary, R., & Ghosh, B. K. (2025). Credit card default prediction: An empirical analysis on predictive performance using statistical and machine learning methods. *Journal of Risk and Financial Management*, *18*(1), 23.
15. Kvamme, H., Sellereite, N., Aas, K., & Sjursen, S. (2018). Predicting mortgage default using convolutional neural networks. *Expert Systems with Applications*, *102*, 207-217.
16. Sigrist, F., & Hirnschall, C. (2019). Grabit: Gradient tree-boosted Tobit models for default prediction. *Journal of Banking & Finance*, *102*, 177-192.

17. Albanesi, S., & Vamossy, D. F. (2019). Predicting consumer default: A deep learning approach. *arXiv preprint arXiv:1908.11498*.
18. Zhou, J., Li, W., Wang, J., Ding, S., & Xia, C. (2019). Default prediction in P2P lending from high-dimensional data based on machine learning. *Physica A: Statistical Mechanics and its Applications*, *534*, 122370.
19. Aniceto, M. C., Barboza, F., & Kimura, H. (2020). Machine learning predictivity applied to consumer creditworthiness. *Future Business Journal*, *6*(1), 37.
20. Turiel, J. D., & Aste, T. (2020). Peer-to-peer loan acceptance and default prediction with artificial intelligence. *Royal Society open science*, *7*(6), 191649.
21. Xu, J., Lu, Z., & Xie, Y. (2021). Loan default prediction of Chinese P2P market: a machine learning methodology. *Scientific Reports*, *11*(1), 18759.
22. Liu, Y., Yang, M., Wang, Y., Li, Y., Xiong, T., & Li, A. (2022). Applying machine learning algorithms to predict default probability in the online credit market: Evidence from China. *International Review of Financial Analysis*, *79*, 101971.
23. Li, H., & Wu, W. (2024). Loan default predictability with explainable machine learning. *Finance Research Letters*, *60*, 104867.
24. Uddin, M. S., Chi, G., Al Janabi, M. A., & Habib, T. (2022). Leveraging random forest in micro-enterprises credit risk modelling for accuracy and interpretability. *International Journal of Finance & Economics*, *27*(3), 3713-3729.
25. Liu, Y., Yang, M., Wang, Y., Li, Y., Xiong, T., & Li, A. (2022). Applying machine learning algorithms to predict default probability in the online credit market: Evidence from China. *International Review of Financial Analysis*, *79*, 101971.
26. Zhou, Y., Shen, L., & Ballester, L. (2023). A two-stage credit scoring model based on random forest: Evidence from Chinese small firms. *International Review of Financial Analysis*, *89*, 102755.
27. Gao, W., Ju, M., & Yang, T. (2023). Severe weather and peer-to-peer farmers' loan default predictions: Evidence from machine learning analysis. *Finance Research Letters*, *58*, 104287.
28. Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).
29. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems*, *31*.
30. Odegua, R. (2020). Predicting bank loan default with extreme gradient boosting. *arXiv preprint arXiv:2002.02011*.
31. Xia, Y., He, L., Li, Y., Liu, N., & Ding, Y. (2020). Predicting loan default in peer-to-peer lending using narrative data. *Journal of Forecasting*, *39*(2), 260-280.
32. Bussmann, N., Giudici, P., Marinelli, D., & Papenbrock, J. (2020). Explainable AI in fintech risk management. *Frontiers in Artificial Intelligence*, *3*, 521340.
33. De Lange, P. E., Melsom, B., Vennerød, C. B., & Westgaard, S. (2022). Explainable AI for credit assessment in banks. *Journal of Risk and Financial Management*, *15*(12), 556.

34. Li, H., & Wu, W. (2024). Loan default predictability with explainable machine learning. *Finance Research Letters*, *60*, 104867.
35. Li, H., & Wu, W. (2024). Loan default predictability with explainable machine learning. *Finance Research Letters*, *60*, 104867.
36. Lin, S., Song, D., Cao, B., Gu, X., & Li, J. (2025). Credit risk assessment of automobile loans using machine learning-based SHapley Additive exPlanations approach. *Engineering Applications of Artificial Intelligence*, *147*, 110236.
37. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*.