CUSTOMER CREDIT SCORING SYSTEM USING MACHINE LEARNING AND WEB-BASED DECISION SYSTEM

BY

ILODIUBA CHIEMELIE BERNARD
GOU/U22/CSC/933

DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY
GODFREY OKOYE UNIVERSITY, ENUGU STATE.

IN PARTIAL FULFILLMENT OF THE AWARD OF BACHELOR OF SCIENCE (B.SC), DEGREE IN COMPUTER SCIENCE

SUPERVISOR: MR CAMILUS NJOKU

JUNE, 2026.

APPROVAL PAGE

This research, titled "**CUSTOMER CREDIT SCORING SYSTEM USING MACHINE LEARNING AND WEB-BASED DECISION SYSTEM**," has been assessed and approved by the Department of Computer Science Committees in the Faculty of Computing and Information Technology, Godfrey Okoye University, Enugu, Nigeria.

Mr. Camilus Njoku	....	....
Supervisor	Signature	Date
External Examiner	....	....
External Examiner	Signature	Date
Dr. Frank Okebanama	....	....
HOD, Department of Computer Science	Signature	Date
Prof. I. A. Ajah	....	....
Dean, Faculty of Computing And Information Technology.	Signature	Date

CERTIFICATION

I certify that I completed the research included therein and that it was primarily based on my observations and investigation. Consequently, I will fully accept the University's decision if I am found to have cheated on the assignment.

ILODIUBA CHIEMELIE BERNARD
GOU/U22/CSC/933

DEDICATION

This project is dedicated to Almighty God, the source of all wisdom, strength, and knowledge.

ACKNOWLEDGMENTS

I give all glory and thankfulness to God Almighty, whose unfailing grace and divine direction helped me to accomplish this work. My deepest gratitude is extended to my supervisor, Mr. Camilus Njoku, and also to Dr. Engr. Callistus Ugorji, for their academic supervision, thoughtful and critical guidance, perseverance, and motivation during completing the project.

I am Mostly grateful to the Head of Department, Dr. Frank Okebanama, for his exemplary leadership and dedication to raising the standard of academic achievements in the Computer Science Department. At the same time, I appreciate all the Computer Science lecturers for the knowledge they have shared with me throughout the course of study.

The biggest thanks must be given to my father Mr. Simon Ilodiuba and mother Mrs. Benedette Ilodiuba, for their parental affection, sacrifice, and continuous support; and my dear brothers and sisters Chisom Chimaobi, Ogechukwu, and Chiamaka, for their consistent cheering.

ABSTRACT

In this research work, an implementation of a machine learning algorithm for customer credit scoring has been done for Bernard Supermarket Enugu, Nigeria, with the aim of solving problems related to inconsistencies, poor record keeping, and poor accountability associated with unstructured, judgmental credit scoring process. The main idea behind this research is to come up with a decision support system based on a web application, which includes a predictable model that will help in making credit granting, approval, and rejection decisions. Literature review shows that credit scoring makes decisions better than judgments, and there is a need for small interpretable credit scoring models for retail credit applications in daily business operations apart from formal banking institutions. The current study was carried out using an agile development approach and consists of a web application having three tiers, which have HTML, CSS, and JavaScript for the front end, PHP for the application layer, MySQL for data storage, and a logistic regression machine learning algorithm written in Python as the classifier. Firstly, the customer dataset was cleaned and normalized, after which it was divided into two parts in the ratio of 80/20 and used for training and testing the classifier respectively. The classifier resulted in 89.21% accuracy and had an equal level of precision, recall, and F1-scores for both classes. Truly, research finds that a small, interpretable machine learning model can provide objective, consistent, and auditable credit decision support even with low-cost hardware. That is why, the system could be considered a reliable baseline solution for supermarket credit management. The platform can potentially be improved through the inclusion of larger datasets, comparative modelling, and wider usage in various settings.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

Credit enables trade to continue but its use involves a certain degree of risk. Many trade relationships are such that the seller delivers the goods before the buyer makes the full payment. The trade-credit research regards such a practice as a payment-deferral means that gains even more traction in situations where the buyers are less capable of raising finance from formal channels while, at the same time, the suppliers are in possession of meaningful information on customers' behaviour. Some studies in the area point to Really suppliers are able to extend credit because, in general, they have a very close and continuous eye on customers - sometimes even managing to monitor them more effectively than financial institutions - and for small and medium-sized enterprises (SMEs) trade credit can constitute a very considerable proportion of business assets and liabilities. As a result, customer credit is not just a financing issue; it is equally an information and decision issue (Pattnaik et al. 2020; Peter et al. 2022).

Because of such factors, personal contact relationship informal records and the seller's individual experience with the customer often determine the granting of credit in a retail environment of small and medium enterprises. This may work perfectly for the few customers known well and who have become regulars over the years but it will definitely put a limit on the number of customers if, at one point, the memory of the institution fades due to staff turnover. Results from studies concentrating on SMEs in Nigeria have revealed that a number of businesses grant credit to attract customers but, then again, lack of control of this credit can bring deterioration of the financial situation. A study conducted in Lagos identified that trade-credit practices covering credit checking, credit policy and cash discounts A lot contributed in shaping the performance of small businesses, highlighting the very tight connection that exists between credit-making decisions and the capacity to keep a business alive (Peter et al. 2022).

To cope with this issue, credit scoring emerged as a way of introducing formal assessment into making decisions about lending. Credit scoring refers to the employment of techniques based on statistics or algorithms which help differentiate between candidates who are considered to be highly likely to repay their borrowings (good risk) and those who are expected to fail doing so (bad risk) (Dastile et al. 2020). It has been greatly adopted as a risk management quantitative

instrument and Mainly in the consumer credit sector it has risen to the ranks of the leading ones as it offers a formal method to estimate the risk of default or the likelihood of repayment behaviour, research says accessible. Besides, interestingly that scoring models are much more than simple formulas; currently, they are embedded into the lending business processes not only for support with prospective customer approval but also for determining pricing, monitoring and account management, and the latest research points out that they need to be both transparent and audit-able if they are to be trusted by regulators and prompt use by those who are making decisions on the ground (Bcker et al. 2022).
Initially, credit scoring mainly relied on classical statistical methods like discriminant analysis and logistic regression, and then it expanded to cover the comparison of linear models, logistic models, neural networks, decision trees and other classifiers (Dastile et al. 2020). By the decade of 2000 scholars were no longer raising the question of whether scoring can be used, but which modelling approach is to be preferred for different business conditions. Logistic regression continues to be a long-standing benchmark due to its robustness and explainability, although its predictive power can be improved using machine-learning methods that can model non-linear relationships (Dumitrescu et al. 2022).

The rise of machine learning has taken this process forward in terms of making systems learn patterns from past data instead of applying only pre-defined rules created by experts. Existing literature on credit scoring includes work done using decision trees, neural networks, support vector machines, and ensemble modelling techniques. Large comparative and benchmark studies reveal that more sophisticated classifiers can Actually boost classification performance, whereas other factors (transparency, implementation cost, data quality, and business usefulness) still need to be considered (Gunnarsson et al. 2021). At the same time, different studies show that transaction-level behavioural data combined with conventional credit information can be Quite a bit more accurate in predicting defaults and missed payments (Ala'raj et al. 2022).

These things do not only count for banking. Most of the credit-scoring work is centered on financial institutions. After all, as far as the fundamental problem is concerned, any organization that is running a customer credit policy must inevitably assess the risk of a delayed or failed payment. In the bibliometric reviews of trade credit, it is noted that risk management work targeting SMEs remains really scarce, although trade credit is a main channel through which many small businesses operate (Pattnaik et al. 2020). Since SMEs usually do not have access to the market information which is a prerequisite for large corporate risk modeling their need is for risk assessment tools that will fit their own situation - an observation which, of course, applies most directly to retail stores and supermarkets that give customer credit but do not have access to the same level of data infrastructure as banks (Peter et al. 2022).

A second major trend is the proliferation of decision support systems. Decision support system (DSS) is a computer-based, interactive system that assists users in problem-solving by drawing on data documents knowledge and models. In most cases a scoring model only represents one step forward and within a credit domain it would be disadvantageous to rely solely on its outputs; instead, one should integrate them within a system that would be able to capture customer information, keep records, display recommendations and enable staff to act consistently. DSS that are both web-based and model-driven are the ones that come with the most perks, mainly in situations where users need to interact with quantitative models in particular in real decision-making events through accessible interfaces (Sharda et al. 2020).

Deploying such systems has become more straightforward because of the Web. Web-engineering literature points out the significant ways in which web applications differ from traditional software. First, their users are geographically dispersed second, they should continuously accommodate changes, and third, they need special attention to usability, navigation and evolving requirements. So, a systematic, disciplined design process is very suitable for web applications (Pressman & Maxim, 2020). For example, a supermarket credit application stands to benefit Really from a web-based platform as it can be made available through the existing in-store devices, supports centralised record-keeping and enables the scoring logic to be interacted with rapidly.

This research yet takes a practical Computer Science angle on the problem. Its intent is not to replicate the enormity of industrial banking systems but rather to produce a relatively small, understandable and useful system for the retail sector. The suggested system integrates a web-based interface, a RDBMS and a ML classifier to make credit decisions for customers in a supermarket context. The case example is that of Bernard Supermarket of Enugu that adopts an objective and responsible approach towards giving credit to customers. The case gives an appropriate setting for this research work.

1.2 Statement of the Problem

For some firms that are giving their customers credit facilities, there is no specific procedure of determining the chances of the credit being repaid. In situations like these, the decision-making process is usually dominated by the personal opinions of the employees or incomplete documentation. In these cases, the process of decision making is often based quite heavily on the subjective assessments of the staff members involved, their familiarity with the particular customers, or inadequate documentation. While these three elements can provide some help in reaching a decision, they simultaneously bring about a level of inconsistency to the whole procedure. It also reduces accountability and increases the chance of bad debt.

Such a problem is further compounded when the number of customers increases or the historical information is not properly documented. Literature on SME trade credit points out that poorly managing credit can even lead to business failure (Peter et al. 2022). At the same time, the overall credit-scoring literature reports that formal risk classification leads to more consistent lending decisions (Dastile et al. 2020). Though, most of the existing scoring systems and research are mainly focused on banks and bigger financial institutions rather than on regular retail businesses which usually have limited technological resources (Gunnarsson et al. 2021).

This paper so responds to the lack of a low-key, data-driven and credit-scoring system that can be used in a supermarket setting. It lays out a machine learning-powered, internet-based decision system that can take in appropriate customer data, assess creditworthiness and assist in more objective credit approval decision.

1.3 Aim and Objectives of the Study

This project intends to design and build a customer credit-scoring system with the help of machine learning as well as a web-based decision support platform that may be implemented in a supermarket setting. The precise objectives of the research are to:

1. identify and analyse relevant customer and transaction features for credit-risk evaluation
2. collect and preprocess appropriate dataset for training the system
3. develop a machine-learning model for classifying customers into lower-risk and higher-risk credit groups
4. evaluate model performance using appropriate metrics.

1.4 Significance of the Study

This research is not only important in practice and academics but is also important for many individuals, including:

a. Customers purchasing on credit from the supermarket: The customers purchasing on credit at the supermarket go through a fairer process of selection based on objective criteria.

b. Supermarket Staff Handling Credit Operations: personnel who approve or monitor credit accounts will enjoy the advantage of a guided decision-support tool that lessens the mental workload and inconsistencies of a purely manual evaluation.

c. Small and Medium-Sized Retail Businesses (SMEs): other small retail businesses that encounter credit management problems can implement the approach they propose here. As such, the logic of credit scoring that was originally developed for banks is equally valid wherever credit is extended (Gunnarsson et al. 2021; Dumitrescu et al. 2022).

d. Academic Researchers in Credit Scoring and FinTech: the research adds to the expanding literature on localizing and simplifying formal credit-scoring techniques for small-business and non-banking contexts, proving that analytical value can be maintained even under constraint situations (Bcker et al. 2022; Sharda et al. 2020).

e. Computer Science Students and Educators: this piece of work is an example of interdisciplinary computing practice going live, combining database design, web-application development, system analysis, object-oriented thinking, UML-based design and predictive modelling into one single integrated system.

f. Future Developers and System Designers: software professionals who want to produce simple decision-support or credit-risk tools for informal retail settings will find the architecture, design choices and modelling approach presented in this work very beneficial.

1.5 Scope of the Study

The work is centred on the design and creation of a system for customer credit scoring in a supermarket setting in Enugu Nigeria taking Bernard Supermarket as a pilot project example.

This system will First and foremost handle credit decisions for retail customers who buy on credit for the short term rather than banking on large-scale corporate lending.

For features, this research involves customer detail capturing, keeping record of transactions and repayments, customer credit scoring through machine learning, and approval recommendation display on a web interface. Though, the system will be the staff and administrator's aid not the customer for self-use. The machine-learning problem is simply a binary classification where customers can either be creditworthy or risky, based on a handful of attributes. Strictly speaking, the system uses client-side web technologies, PHP-powered server technology, a MySQL database, JSON/AJAX data exchange, and a simple machine learning component suitable for the level of student's thesis. There is no mention of integration with credit agencies, banking API, mobile money gateway, or any advanced deployment capability, such as cloud orchestration.

1.6 Limitations of the Study

There were multiple constraints encountered in the study process. First of all, there was data availability constraint as a supermarket-focused student project cannot have access to an extensive and completely labelled repayment histories, which banks and large fintech companies can use. Moreover, according to the existing literature, such data availability is crucial for the performance of the model (Dastile et al. 2020; Gunnarsson et al. 2021). Thus, in order to solve the problem, the system uses structured data capture form that consistently records customer, transactions, and repayments' information and, thus, gradually provides a clean data set that contains a limited number of highly relevant features and the binary classification problem. Another constraint had to do with deployment: there was no way to implement a complicated and modern model that would be hard to understand and believe for non-specialists. Thus, the study aimed at implementing a simpler model (Bäcker et al. 2022) with a clear decision support interface. Moreover, since the system was created within one store, the results are specific to this location and cannot be generalized.

1.7 Definition of Key Terms

a. Credit Scoring: Using formal statistical or algorithmic techniques to separate credit applicants/customers into different risk groups based on observed variables. In practice, it is a methodical approach for calculating the probability of a customer making a payment or defaulting (Dastile et al. 2020; Bcker et al. 2022).

b. Credit Risk: is the risk that the borrower or customer may not fulfill the payment agreements. For a retail environment, it also represents the possibility that goods issued on credit terms may not be paid for as anticipated (Pattnaik et al. 2020; Ala'raj et al. 2022).
c. Trade Credit / Customer Credit: deferred-payment terms under which goods and services may be delivered first and paid for later. Here customer credit points to the store's granting of permission to the customer for obtaining goods with payment scheduled for a later time (Pattnaik et al. 2020; Peter et al. 2022).
d. Machine Learning: Statistical learning methods using computational models that can identify patterns from data to make predictions or classifications. The concept is applied here to identify customer creditworthiness by analyzing past customer and repayment data (James et al. 2021).
e. Decision Support System: a computer-based, interactive system that assists users in solving decision problems by using data and models. Here it means the software through which the store staff enter the data, get a score and follow a recommendation (Sharda et al. 2020).
f. Web-Based Information System: A software application that is deployed using web technologies and mostly accessed through a web browser. Such systems must be carefully designed for navigation, usability and the ability to handle change over time, mostly in operational environments (Pressman & Maxim, 2020).

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter reviews the body of knowledge relevant to the development of a customer credit scoring system for a retail environment. The review proceeds from concepts to theories, then to empirical studies, before examining existing systems and identifying the research gap. This does not merely imply listing all the previous works done, but it indicates how this study is a development of earlier research and differs in substance from other researches.

2.2 Conceptual Review

2.2.1 Credit Scoring Systems

The need to move lending decisions from the realm of subjective judgment to a formal risk categorization process led to the emergence of credit scoring systems. Hand and Henley (1997) define credit scoring as a collection of formalized statistical procedures aimed at classifying borrowers into acceptable and non-acceptable risk classes, whereas more comprehensive definitions highlight that credit scoring has turned out to be one of the most successful quantitative techniques in consumer finance. Operationally, a scoring system is defined by a transformation of inputs into a score or probability used to make an accept/reject/revise decision (Thomas et al., 2017).

The history of credit scoring is marked by the usage of linear and discriminant methods. Although Altman's (1968) research on failure prediction is well-known for illustrating the strength of using multiple factors in modelling, it needs to be noted that the initial application was not concerned with retail credit, but with the bankruptcy of corporations. Later, logistic regression became an important addition to the available tools, especially considering the naturalness of this technique when dealing with dichotomous outcome and interpretable coefficients (Crook et al., 2007).

In the course of development, credit scoring became widely applicable in application scoring, behaviour scoring, account management, collections, and profitability analysis. The mentioned diversity is critical since credit scoring is not limited to a single screening decision, but may also be helpful for making subsequent decisions regarding existing customers and their changing risks and needs. However, in the current research, credit scoring will only be viewed from the angle of application-type scoring in the retail store context and the relevant question of whether a client may buy on credit (Thomas et al., 2017; Mester, 1997).

2.2.2 Customer Credit Risk in Retail Environments

First of all, there are significant differences between customer credit risk in retailing and the risks modelled in the banking sector. Datasets, regulatory environment, and products differ significantly between banks and supermarket chains and other retailers. However, the economics of the problem stay the same, products are sold before the payments are completed, thus leaving room for potential defaults.

Literature on trade credits provides some explanation of the importance of this problem. According to Petersen and Rajan (1997), the reason why sellers provide trade credit is that they have information about their clients which could be obtained during the course of their interaction with the firm and they might have an advantage over the financial institutions in terms of either monitoring or liquidation. This is especially relevant for the retail setting, as store employees might have practical experience in getting information about customers through repeated business interactions. However, this information might not be institutionalized.

Trade credit is especially important for small and medium-sized enterprises. Trade credit comprises a large share of the balance sheets of many SMEs and there is insufficient research done on the subject of credit-risk management tools specific to the SMEs. Furthermore, as noted by Terradez et al. (2015), one of the limitations of the most advanced risk-management approaches is the need for market data unavailable to small firms, hence the need for a context-specific scoring system.

Nigerian SME literature provides another perspective on this question. According to Peter et al. (2022), many SMEs give credit to their customers in order to attract and keep them, however, poorly managed trade credit can negatively affect their financial performance. Retail credit can drive up sales, however, only if the company can manage the credit risk. It is exactly this point where the scoring system comes in handy.

2.2.3 Machine Learning in Credit Prediction

Machine learning represents one of the busiest fields in credit-risk prediction because of the nature of the problem that involves binary classification where a client either repays or defaults. In their systematic review of methodologies used for credit scoring, Louzada, Ara, and

Fernandes (2016) discuss a wide range of approaches such as neural networks, support vector machines, logistic regression, decision trees, fuzzy systems, Bayesian networks, and genetic programming. The authors note fast development of the field since the 1990s as well as the predominance of classification tasks in the research on credit risk.

The main advantage of using machine learning is the ability to detect complex patterns of non-linear interactions and relationships that are invisible for traditional scorecards. For instance, Khandani, Kim, and Lo (2010) reveal that machine learning algorithms that use information about transactions and credit bureau can predict delinquency and default better than other algorithms. There are other studies that prove that prediction of credit risk requires comprehensive comparison of various classifiers and sometimes new approaches outperform traditional logistic regression models (Baesens et al., 2003; Lessmann et al., 2015).

However, there are other factors that should be considered in credit scoring. First of all, the process takes place in a decision-making context where interpretability of results is vital because users need to trust the results of the analysis and explain them. Baesens et al. (2003) consider this issue discussing the way in which results of neural networks can be transformed into decision tables and rule sets to make the model easier to understand for decision-makers. This idea is especially important for the small businesses as the model that is difficult to explain and understand will never be used in practice regardless of its efficiency.

Thus, within the current study, machine learning is viewed as a practical tool rather than the goal itself. The key point is not to develop a complicated algorithm but to choose a model that is interpretable, operational, and predictive enough to work in the environment of web-application created by an undergraduate.

2.2.4 Decision Support Systems

Credit scores are of little use unless incorporated into decision-making procedures. Here the concept of Decision Support Systems comes into play. As defined by Power, DSS refers to systems aiding people in using information in various forms such as data, documents, communication, knowledge and models in order to solve problems and take decisions. Most importantly, DSS systems are built to be reused on an ongoing basis rather than being used only once (Power, 2000, 2007).

The definition given by Power (2007) for model-driven DSS describes the latter in terms of one or more models being used as the source of analytical capability and user interaction with these models through convenient interfaces. This makes the proposed credit system a perfect example of model-driven DSS: machine learning classifier is used as a model part and web interface and database as means through which users can enter data, get histories and receive credit scoring results (Power, 2007).

Web-based aspects are also important here. Power (2000) points out that the emergence of web technologies created new opportunities for development of DSS systems since models and decision tools could now be accessed via web and network environments. This becomes especially important for a supermarket scale credit scoring system since it enables usage of browser-based application rather than specialized software requiring high costs and working with centralized records.

2.2.5 Web-Based Information Systems

Firstly, a web-based information system does not consist of nothing more than a transfer of traditional information system to a web page. The literature on web engineering suggests that such a system possesses specific requirements in terms of navigation, usability, evolution and integration. Web engineering is defined by Deshpande et al. (2002) as "the application of systematic, disciplined and quantifiable approaches to the development, operation and maintenance of web-based applications". It becomes particularly relevant in the context of an undergraduate project where the latter can easily transform into an assembly of pages and scripts if its design is not based on the proper systems approach.

The issue of requirements analysis is also essential. According to Escalona and Koch (2004), there are some additional issues in web applications that are related to navigational structure, multimedia, and usability. Therefore, it is necessary to perform careful requirements engineering. In reality, a credit scoring web application has to provide an easy way to enter the data, low cost of training users, reliable navigation, and convenient display of scores and decisions. Even the best scoring model will not help in case of inefficient usability of the interface used by store clerks.

Furthermore, Hennicker and Koch (2001) note that UML design techniques are suitable for designing of web applications as they enable developers to transition from use cases and concept models to navigation and presentation. Thus, the inclusion of UML artifacts in the current research is justified.

2.3 Theoretical Framework

2.3.1 Statistical Credit Scoring Theory

The first theoretical anchor of this study is statistical credit scoring theory. At its core, this theory treats the credit decision as a classification problem: using known characteristics of previous borrowers or customers, the analyst seeks to estimate the likelihood that a new case belongs to a favourable or unfavourable risk class. Hand and Henley (1997) present this logic clearly in their review, and Crook, Edelman, and Thomas (2007) show how central it remains to consumer credit risk assessment. (Hand & Henley, 1997; Crook et al., 2007).

The theory is grounded in the assumption that past behaviour and observable characteristics contain information about future repayment outcomes. Classical models such as discriminant analysis and logistic regression translate these observations into scores or probabilities. Altman's work demonstrated the value of multivariate risk assessment, and later developments consolidated probability-based scoring as a dominant method in lending analytics. In practice, this means that credit decisions need not depend entirely on experience or intuition; they can be supported by estimated probabilities derived from historical evidence. (Altman, 1968; Thomas et al., 2017).

For the present study, statistical credit scoring theory justifies the effort to extract measurable predictors from customer and transaction data and to convert them into a systematic decision aid. It also justifies the use of a threshold-based approval rule, in which estimated probabilities are translated into a practical decision outcome such as approve, reject, or review.

2.3.2 Classification Theory in Machine Learning

The second theoretical anchor is classification theory in machine learning. In statistical learning, classification refers to the prediction of categorical outcomes, especially binary outcomes such as yes/no, good/bad, approve/reject, or default/non-default. James et al. (2021) explain that classification methods seek to learn a mapping from input variables to qualitative class labels, often by estimating class probabilities and applying a decision rule. (James et al., 2021).

This theory is important because it frames credit scoring as a supervised learning task. Historical examples with known outcomes are used to train a classifier; once trained, the model predicts the class of a new observation. In credit applications, the model is learned from past cases where repayment outcomes are already known. The present study follows that logic by treating prior customer records as training examples and new credit requests as cases to be classified.

Classification theory also highlights the need for evaluation. The choice of classifier is not judged simply by intuition but by observed predictive performance, using measures such as accuracy, precision, recall, AUC, and cost-sensitive considerations. Lessmann et al. (2015) show why these matters: different models can perform differently across metrics and data conditions. For a small retail system, this reinforces the need to choose a model not only for statistical performance, but also for managerial usefulness and implementation feasibility.

2.4 Review of Related Literature

There has been a tremendous increase in empirical research related to credit scoring since 2019 thanks to developments in machine learning, availability of alternative data, and increasing demands for the accuracy, explainability, and trustworthiness of the model used. This review focuses on empirical papers published in 2019 through 2026 which have been chosen based on their relevance to the current study and include issues such as using data-driven classification for credit decision making, improvement of modern algorithms compared to previous baseline models, and practical implementation of the technique in retail banking context. Read together, they reveal four recurring themes, the consolidation of the field through systematic reviews, the contest between predictive power and interpretability, the steady advance of deep and ensemble learning, and a belated but important turn towards small and medium-sized enterprises (SMEs) and developing-market contexts. Table 2.1 summarises the selected studies, after which the broader patterns are discussed.

Recent reviews have helped to map a field that had become unwieldy. Dastile, Çelik and Potsane (2020) surveyed the statistical and machine learning literature and concluded that ensemble classifiers consistently outperform single models, while identifying interpretability and class imbalance as the two issues most likely to impede practical adoption. Shi et al. (2022) reached a complementary verdict in their systemic review, ranking model families across public datasets and drawing attention to data imbalance, dataset inconsistency, and the under-use of transparency-preserving methods. Çallı and Coşkun (2021), reviewing more than a hundred studies, shifted the focus from algorithms to predictors, charting how the socioeconomic, demographic, behavioural and situational variables associated with default have changed over time. Collectively, these reviews confirm that the discipline has moved well beyond the question of whether to use scoring, towards finer questions of which model, which features, and under what constraints.

A second cluster of work has probed how much newer algorithms actually gain over the long-standing logistic-regression baseline. Moscato, Picariello and Sperlí (2021) benchmarked widely used classifiers on peer-to-peer lending data and showed that the careful handling of class imbalance is often as decisive as the choice of algorithm. Gunnarsson et al. (2021), in a pointedly titled study, found that deep learning does not automatically justify its added complexity in credit scoring, with gradient-boosted trees frequently matching or exceeding deep networks while remaining easier to deploy. Merćep et al. (2021) demonstrated the converse case, using deep neural networks to exploit very large behavioural datasets where the volume and richness of the data reward more elaborate architectures. The key message of the papers discussed above is that the right choice of a model very much depends on scale of data, quality of features and context of application, not just on the complexity of algorithms used.

Accuracy versus Explainability conflict becomes the third main theme. Ariza-Garzón et al. (2020) showed that Shapley-based methods can render the predictions of otherwise opaque scoring models intelligible to lenders and borrowers on peer-to-peer platforms. Bussmann et al. (2021) extended this idea by applying correlation networks to Shapley values across fifteen thousand SME borrowers, grouping applicants by the reasons behind their scores rather than by the scores alone. Gramegna and Giudici (2021) compared the two most popular explanation techniques, SHAP and LIME, evaluating their ability to discriminate between groups of borrowers. Dumitrescu et al. (2022) reached the same goal by a different route, enriching logistic regression with non-linear decision-tree effects so as to capture interactions while preserving the interpretability that practitioners and regulators value. For a system intended to be understood and trusted by non-specialist staff, this body of work is especially pertinent.

A fourth and, for the present study, most relevant theme is the growing attention to SMEs, small businesses, and developing-market settings, which had long been overshadowed by bank-oriented research. Schalck and Yankol-Schalck (2021) applied machine learning to predict the failure of French SMEs, demonstrating gains over conventional techniques on firm-level data. Yang et al. (2021) identified the credit-risk determinants of SMEs in sustainable supply-chain finance, finding transaction credit and reputation supervision to be especially influential. Hou, Lu and Bi (2024) proposed a feature-selection and resampling framework that improved SME risk prediction in supply-chain finance, while noting that redundant features and imbalanced data remain persistent obstacles. Tarantino and Filomeni (2024) asked directly whether machine learning can be trusted for small-firm credit, and showed that it outperforms traditional techniques precisely when the information available to lenders is limited, a condition typical of informal credit. Most directly of all, Baadel (2025) developed tailored credit-scoring models for informal African merchants using fintech data, arguing that appropriate scoring tools can widen access to working capital in exactly the under-served retail sector that this study addresses.

Table 2.1: Summary of Related Literature Review

Study	Focus	Key findings	Limitations
Ariza-Garzón et al. (2020)	Explainable ML scoring in peer-to-peer (P2P) lending	Shapley-based methods render opaque scoring models intelligible to lenders and borrowers without major loss of accuracy	Confined to P2P loan data; explanation effort grows with model complexity
Dastile, Çelik & Potsane (2020)	Systematic review of statistical and ML scoring	Ensemble and deep models outperform single classifiers; interpretability and class imbalance are the main barriers to adoption	Review only; reliant on common benchmark datasets rather than field deployment
Bussmann et al. (2021)	Explainable ML for SME credit risk on P2P platforms	Correlation networks applied to Shapley values group 15,000 borrowers by the drivers behind their scores	Tested on a single P2P SME dataset; methodologically demanding
Çallı & Coşkun (2021)	Longitudinal review of credit-default predictors	Charts how socioeconomic, demographic, behavioural and situational predictors of default have evolved over time	Predictor-focused rather than system-focused; oriented to consumer credit
Gramegna & Giudici (2021)	Comparison of SHAP and LIME in credit risk	Evaluates the discriminative power of the two leading explanation methods for grouping borrowers	Explanatory study, not a deployable scoring system; limited datasets
Gunnarsson et al. (2021)	Deep learning versus simpler models	Deep learning rarely beats gradient-boosted trees, which are simpler to implement and deploy	Based on two public datasets; results may not transfer to richer behavioural data

Study	Focus	Key findings	Limitations
Merćep et al. (2021)	Deep neural networks for behavioural credit rating	Very large behavioural datasets reward deeper architectures with strong predictive gains	Requires data volumes atypical of small retailers
Moscato, Picariello & Sperlí (2021)	Benchmark of ML classifiers on P2P data	Careful handling of class imbalance is often as decisive as the choice of algorithm	Limited to P2P repayment classification
Schalck & Yankol-Schalck (2021)	ML prediction of French SME failure	ML techniques outperform conventional methods on firm-level SME data	Enterprise-failure focus, not point-of-sale retail credit
Yang et al. (2021)	SME credit risk in sustainable supply-chain finance	Transaction credit and reputation supervision strongly influence SME credit risk	Supply-chain finance setting using Chinese listed-firm data
Dumitrescu et al. (2022)	Interpretable ML credit scoring	Logistic regression enriched with non-linear decision-tree effects gains accuracy while remaining interpretable	Demonstrated on standard credit datasets, not local retail settings
Shi et al. (2022)	Systemic review of ML-driven credit risk	Deep models generally lead on accuracy; transparency and data quality remain open challenges	Review only; rankings depend on public benchmark datasets
Hou, Lu & Bi (2024)	SME credit risk in supply-chain finance	A feature-selection and resampling framework improves prediction over any single algorithm	High data demands; redundant features and imbalance persist
Tarantino & Filomeni (2024)	Trustworthiness of ML for small-firm credit	ML outperforms traditional techniques most clearly when the information available to lenders is limited	Invoice-lending fintech data; corporate rather than retail customers
Baadel (2025)	ML scoring for informal African merchants	Tailored supervised models predict default and widen working-capital access for informal-sector retailers	Single fintech dataset (Lesotho); limited to logistic regression and SVM

Overall, the literature reviewed in the last few years establishes three principles relevant to the present research. First, data-driven scoring always helps to increase credit decision consistency, and, usually, ensembles or boosts yield better accuracy scores. Second, accuracy score is not the decisive factor for the usefulness of a credit model if that model cannot be used in practice and explained to stakeholders. This issue has been at the center of attention of recent studies. Third, despite a clear tendency to consider SME customers and developing markets, formal lenders, fintech, and supply chain finance remain the leading empirical settings of the research while lightweight, easily deployable, and interpretable web-based scoring applications for retail are quite rare. This is precisely the combination of problems, approaches, and solutions that the current paper tries to solve.

2.5 Research Gap

It is evident that the literature supports both the notion that credit scoring is useful and proves that machine learning may help to improve the accuracy of predictive models under certain conditions. The trouble is that most of such research is related either to banks, consumer finance, small business banking, or corporate credits. The typical dataset is collected in formal institutions and the typical task deals with prediction of loan, card, or lending portfolio risk rather than credit in the informal or semi-formal environment of retail. (Hand & Henley, 1997; Crook et al., 2007; Lessmann et al., 2015).

The next research gap is associated with implementation in the particular context. Many authors, who consider the issues of SME trade credit, claim that a smaller business requires its own approach due to the absence of market data and other infrastructure assumed in the risk modeling literature. At the same time, very little attention is paid to the development of cheap and simple web-based system for retail credit and assessment of customer credit using explainable machine learning. (Terradez et al., 2015; Peter et al., 2022).

Finally, systems integration remains the problem as well. In many cases, the literature concentrates on classifiers and predictive accuracy, while comparatively few studies deal with decision support systems and, even rarer – with small business applications. The current research attempts to bridge this gap by integrating three aspects of the process: retail scoring problem, interpretable machine learning, and web-based operational system.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 Introduction

The methodology of the study and design of the proposed system is outlined in this chapter. It covers the research methodology, development methodology, current practice analysis in case study environment, proposed system architecture, machine learning design, main tools used, UML diagrams, sources of data and requirements to hardware and software.

3.2 Research Methodology

3.2.1 Research Approach

An applied system development approach is taken for this study. It implies addressing an operational problem by analysing and designing a system of software artifacts. This approach corresponds to the aim of the research, as it implies the development of a useful credit decision-making tool in response to the local problem, rather than generating any abstract theoretical findings.

The current research combines elements of information systems development, web engineering and predictive analytics. Web engineering literature supports a systematic approach to development of web applications, while decision support literature supports integration of data models into operation systems and not their isolation as separate analytic activities. (Deshpande et al., 2002; Power, 2000).

Accordingly, the current research procedure includes the following steps:

1. identification and analysis of the problem;
2. analysis of existing credit practices in the selected store;
3. determination of the data and functionality requirements;
4. design of database, interfaces and UML models;
5. selection and design of a machine learning classifier;
6. implementation of the classifier within a web application.

3.2.2 Development Methodology

Agile development approach has been selected as a methodology for development of the software component of the system. Agile methodology stresses such principles as delivery of a functional software artifact early in the project, ability to respond to changing requirements, frequent iterations and attention to usability. All these principles appear to be very relevant for the current project as it requires integration of interface design, database, business rules and predictive model into a single product. (Agile Manifesto, 2001).

In practical terms, the work is conducted in increments of manageable scope. One increment is devoted to simple customer record management, another – to transactions and repayments history. Third increment integrates the scoring engine and the fourth one works on the decision dashboard and report pages.

It is also essential to note that Agile in this case relates to the development method, not the programming approach. The design of the system still utilizes the top-down approach in going from the architecture level to modules, and from there to the class, table, and interface level. Similarly, the classes such as Customer, Credit Request, Credit Score, Transaction, and Admin can also be organized using the object-oriented approach despite developing the application in iterations.

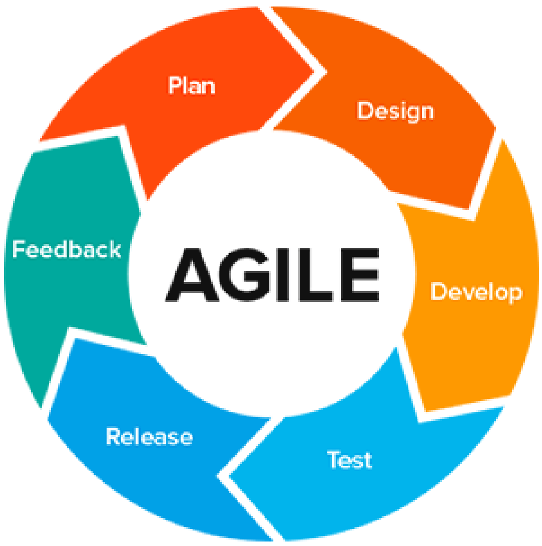


Figure 3.1: Proposed Development Methodology (Agile)

3.3 System Analysis

3.3.1 Analysis of the Existing System

A preliminary analysis of the existing practice in the case study environment shows the following features:

a. Credit decisions about clients are manual, based on the knowledge of the client, previous repayment behaviour, and transaction notes.
b. Client records are not processed via the formal risk model; there is no centralized scoring that helps to make objective decisions about the client.
c. The current system works as the expert system without the knowledge base. Staff knowing the client well may make objective decisions, but it is a subjective process, and it depends mostly on the memory.
d. This system is inconsistent. Two different staff members might have different opinions about the client and vice versa – two clients similar in their behaviour might get different decisions. The reasoning behind the decisions is not documented.
e. 3.3.2 Problems of the Existing System

The problems of the existing system can be described as follows:

a. The lack of objective and standardized criteria for credit decisions;
b. Poor traceability of reasons for credit granting or rejection;
c. Over-reliance on the experience and memory of the individuals;
d. The low usage of the historical data in prediction of the repayment in the future;
e. Difficulty in spotting the patterns of delays in payments between the clients;
f. Time consumption in manual checks of previous transactions; and
g. Higher chances of the formation of the bad debt because of the inconsistency in risk evaluation.

These problems align with the general concerns regarding the SME trade credit management and the need for the formal scoring in lending environment, despite the fact that the actual application might differ from place to place. (Peter et al., 2022; Terradez et al., 2015).

3.3.3 Analysis of the Proposed System

The proposed system brings new elements of consistency to the client credit evaluation process and it is based on three major ideas:

a. Data-driven assessment of clients: Client variables are gathered, put into the central database, and then are processed using classification model for estimation of creditworthiness of the client.

b. Human-assisted decision-making: Proposed system does not replace humans in making decisions; it just proposes a score for the client.

c. Transparency and traceability: With inputs being recorded, scores based on logic, and results linked to specific cases, the process moves away from a memory-dependent exercise towards a completely transparent decision-making process.

3.3.4 Advantages of the Proposed Process

The benefits of the proposed process include:

1. Consistency: Similar cases are processed with the same input and scoring logic.
2. Efficiency: Historical data and predictive results are instantly available.
3. Traceability: Each individual decision can be traced back to the data stored by the system.
4. Better awareness of the risk: Previous repayment behavior can be taken into account in the current decision.
5. Scalability: The process is not limited to the memory of a single employee.
6. Integration of analytics and operations: The machine learning model is embedded within an existing business process rather than applied independently.

These benefits are consistent with the wider context of why credit-scoring and model-driven decision support systems were developed.

3.4 System Design

3.4.1 System Architecture Design

The proposed architecture is a three-layered web architecture along with the inclusion of a machine learning component.

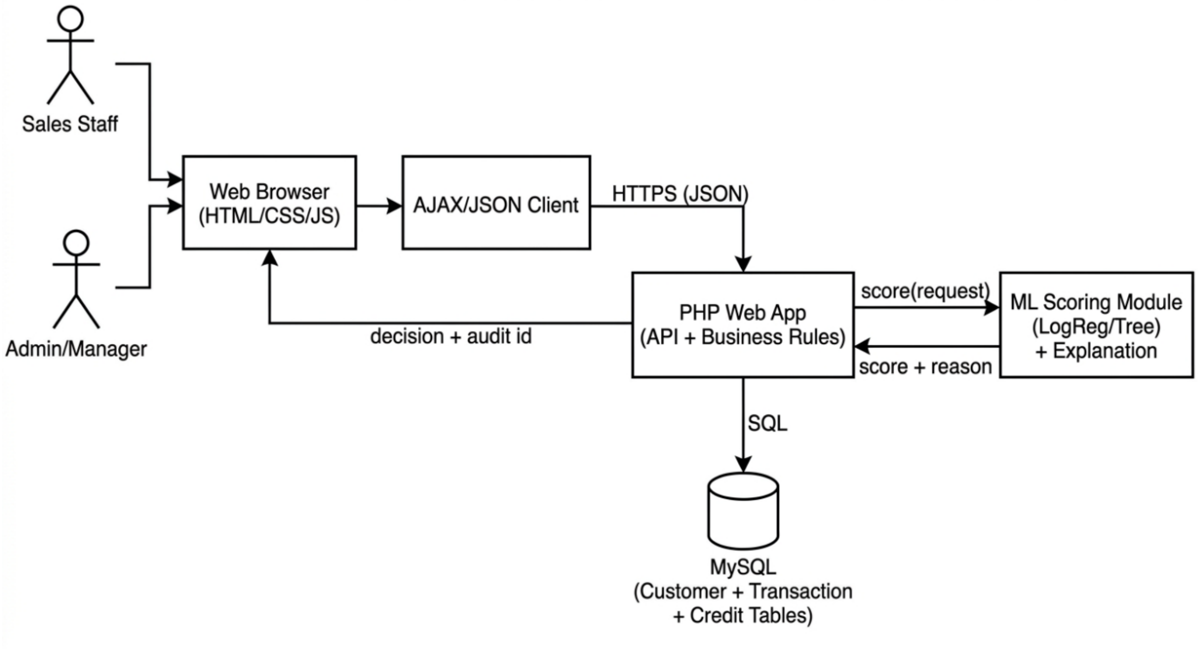


Figure 3.2: Proposed System Architecture

The architecture is intentionally simple. The user will interact with the system through a web browser. The PHP layer will validate the inputs provided, request for scoring, store results, and provide feedback from the backend through the interface. The MySQL database is used for storage of customer and transaction records. The machine learning component can be trained in Python using a light weight library and accessed by the PHP layer through the means of a serialized file, command line call or local API endpoint providing JSON response.

3.4.2 Input Design

The proposed input design will include structured information about customers and credits. In order to allow for accurate predictions, the system will avoid unstructured, free text format input as much as possible and use fields like text boxes, drop down menu, check box and numeric input validation.

Proposed input categories are:

i. customer identification details;

ii. contact information;

iii. occupation or business type;

iv. estimated monthly income or earning range;

v. average purchase value;

vi. frequency of purchase;

vii. prior repayment behaviour;

viii. current outstanding balance;

ix. requested credit amount; and

x. repayment period requested.

A good input design is essential because classification quality depends strongly on the quality of the observed features. The input should be complete enough for scoring purposes without being too stringent to make their use unappealing to the store personnel.

3.4.3 Output Design

The design of the output is such that it facilitates actions instead of just showing figures. The outputs include:

i. a predicted credit class (for example, low risk or high risk);

ii. a score or probability value;

iii. a suggested decision such as approve, reject, or review;

iv. a short explanation field highlighting major contributing factors where possible;

v. customer credit history summary; and

vi. Administrative reports on accepted, denied, and overdue applications.

Such outputs should be displayed through an interface in a non-technical way. Operational users care more about the next step in the workflow rather than the mathematical nature of the classification algorithm.

3.4.4 Database Design

The proposed database structure is relational and based on a small number of interrelated tables. The conceptual description of such tables is given below.

Table 3.1. Database Design Table

Table	Purpose	Selected fields
customers	Holds customer identification and attributes	customer_id, full_name, phone, address, occupation, income_range, created_at
transactions	Holds transactions related to purchases and repayments	transaction_id, customer_id, transaction_type, amount, transaction_date, status
credit_requests	Holds each credit request/application	request_id, customer_id, requested_amount, request_date, requested_due_date, officer_id
credit_scores	Holds model outputs and decision results	score_id, request_id, probability_good, predicted_class, decision, decision_date
users	Holds system users and administrators	user_id, username, password_hash, role
audit_logs	Holds system activity history	log_id, user_id, action, log_time

Relationships are defined as follows: one customer has many transactions, many credit requests, and many score entries related to those requests; one user processes many requests and produces many logs.

3.5 Machine Learning Model Design

3.5.1 Data Description and Features

Variables chosen in the model design represent those that are realistic for a retail store to gather. Unlike in institutional credit databases that may include bureau score, detailed liabilities and regulatory data, the present approach puts more emphasis on a combination of profile variables and behaviour-based variables that can be observed in the store environment.

The suggested set of features includes:

a. age group;
b. occupation/way of living;
c. estimated income level;
d. customer loyalty with the store;
e. frequency of purchases;

f. amount of average purchase;
g. historical sum of credit taken;
h. quantity of repayment made;
i. quantity of late repayment;
j. current overdue;
k. repayment ratio of all credit received

The use of transaction behaviour variables is justified from literature on credit prediction, where historical payment information and transactions proved to bring a lot of predictive power. Khandani et al. (2010) show the benefit of using transaction level data for predicting delinquencies. Moreover, research on SME trade-credit reveals that both behaviour and financial variables are necessary for building models of default risk. (Khandani et al., 2010; Terradez et al., 2015).

The target variable in the classification problem is binary. The historical cases will be marked depending on the result of repayment, e.g.,

1 = good credit (repayment within the period acceptable to the company);

0 = bad credit (default or lateness above the acceptable period).

Thus, the task is suited to a classification model.

3.5.2 Model Selection

The main model chosen for the experiment will be Logistic Regression. It was chosen deliberately, despite the fact that benchmarking studies prove some complex classifiers to perform better. Nevertheless, logistic regression is still widely used in credit scoring applications owing to stability, interpretability, and its applicability for the task. Crook et al. (2007) state that logistic regression is the most often used tool to classify customers as reliable and unreliable credit-takers, and comparative studies continue to take logistic regression as one of the main benchmarks (Crook et al., 2007; Hand & Henley, 1997).

There are several reasons to use logistic regression in this project. First, it suits well for binary classification tasks. Secondly, output values are probability scores that can be used in threshold-based decisions. Thirdly, it provides easy-to-understand coefficients' interpretation in contrast with some black-box methods. Fourthly, it is relatively simple to train and implement in an undergraduate web application.

Other candidate models could include such approaches as decision trees, neural networks, support vector machines, random forests and ensemble methods. Benchmarking studies prove that those approaches provide additional predictive power in certain cases, but they increase the complexity of the solution, thus might decrease interpretability or simplicity of deployment in an underdeveloped small and low-cost system (Baesens et al., 2003; Lessmann et al., 2015; Bensic et al., 2005).

Therefore, logistic regression will be used as the main classifier, whereas the other models could be considered later as a comparative option during the experiment.

3.5.3 Model Training and Prediction Process

The process of training will consist of the following steps:

1. Data collection and collation: the historical data about customers and their transactions will be collected from the case study environment.
2. Data cleaning: problems with missing data, duplication, inconsistencies will be handled.
3. Feature preparation: encoding of categorical variables and standardization of numeric variables if required.
4. Labelling: translation of historical outcomes into 'good credit'/'bad credit' classes.
5. Train-test split: splitting of dataset into training and testing parts.
6. Model fitting: training the logistic regression model on training set.
7. Evaluation: calculating accuracy, confusion matrix, precision, recall, AUC on test set.
8. Deployment step: deploying trained model.
9. Live prediction: when the new customer's credit application arrives, the input is translated into feature space and prediction is applied using the trained model.

In this scenario, the most important thing is the operational adequacy of the system. The classifier must have sufficient precision to assist in informal decision-making but at the same time it needs to be stable so as to provide updates once more data are available. According to credit scoring research, the use of a wide range of measures in evaluation is crucial because various performance measures highlight different strengths and weaknesses of the classifier (Lessmann et al., 2015).

3.6 System Implementation Tools

The system implementation is based on the following tools and technologies:

a. HTML – page structuring;
b. CSS – presentation and user interface design;
c. JavaScript – client-side programming;
d. AJAX – asynchronous communication between client and server;
e. JSON – lightweight data exchange;
f. PHP – server-side programming and integration;
g. MySQL – data storage and manipulation; and
h. Python with simple machine learning libraries – model training and scoring.

This particular set of technologies is suitable for the current project due to the fact that it is lightweight, well-known and feasible within an undergraduate level development environment.

3.7 UML Diagrams

3.7.1 Use Case Diagram

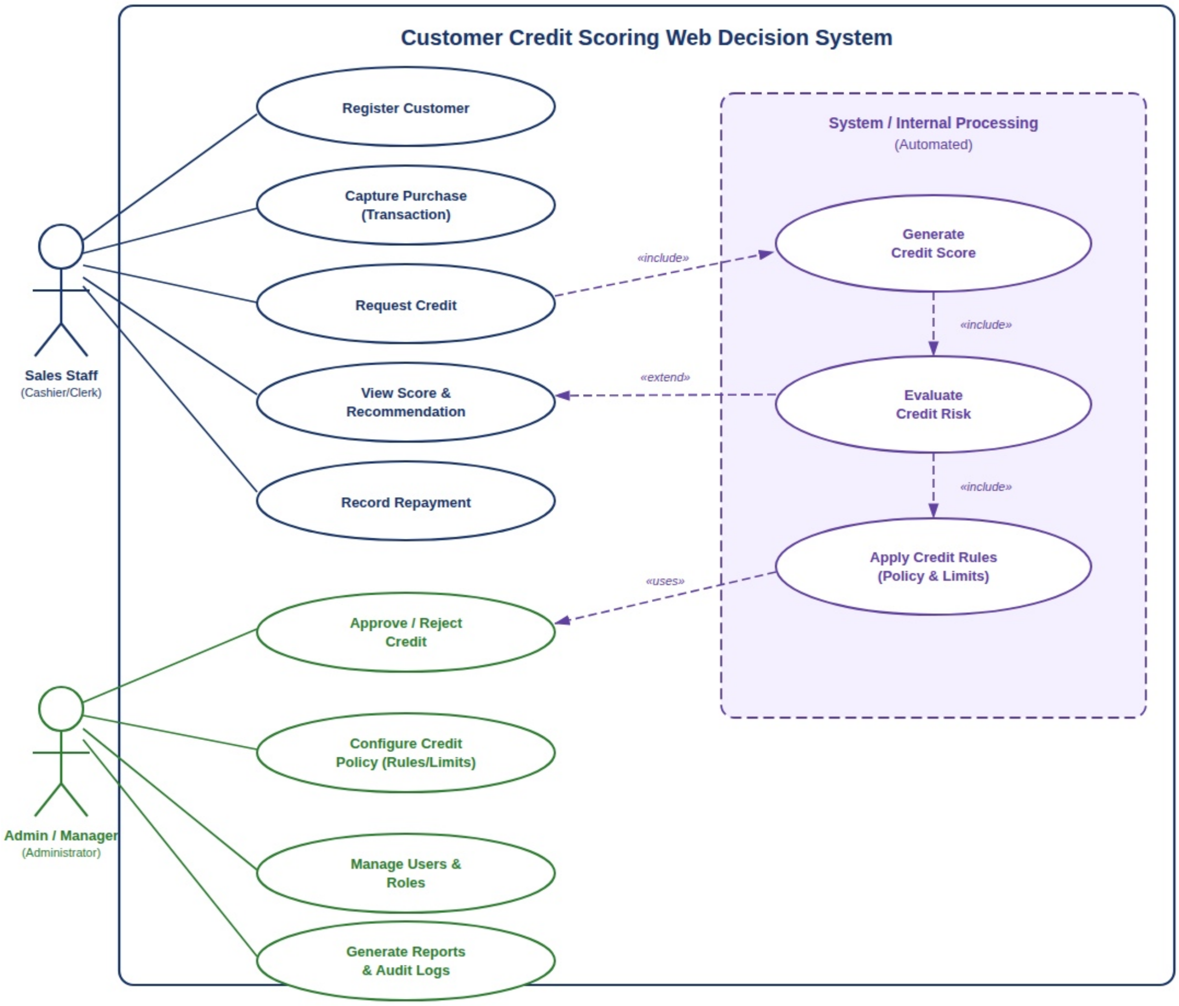


Figure 3.3: System Use Case Diagram

The use case diagram describes how two main actors namely, the Sales Staff (Cashier/Clerk) and the Admin/Manager (Administrator) interact with the Customer Credit Scoring Web Decision System. The Sales Staff handles all operation-related tasks, which include registering customers, capturing the transaction details, applying for credits, accessing scoring suggestions and repayment records. The Admin/Manager handles supervision-related tasks, which include granting or rejecting credit, setting the credit policies and limits, handling user permissions and producing reports. When a credit request is initiated by the Sales Staff, the system automatically triggers an internal processing chain, beginning with generating a credit score, which includes evaluating credit risk, which in turn includes applying the defined credit rules and policy limits. The evaluation process extends to produce a score and recommendation that is made visible to the Sales Staff, while the applied credit rules feed into the approval or rejection decision managed by the Admin. Together, these interactions demonstrate a structured, role-based workflow where human actors initiate and finalise decisions while the automated system handles the underlying computation and risk assessment transparently in the background.

3.7.2 Activity Diagram

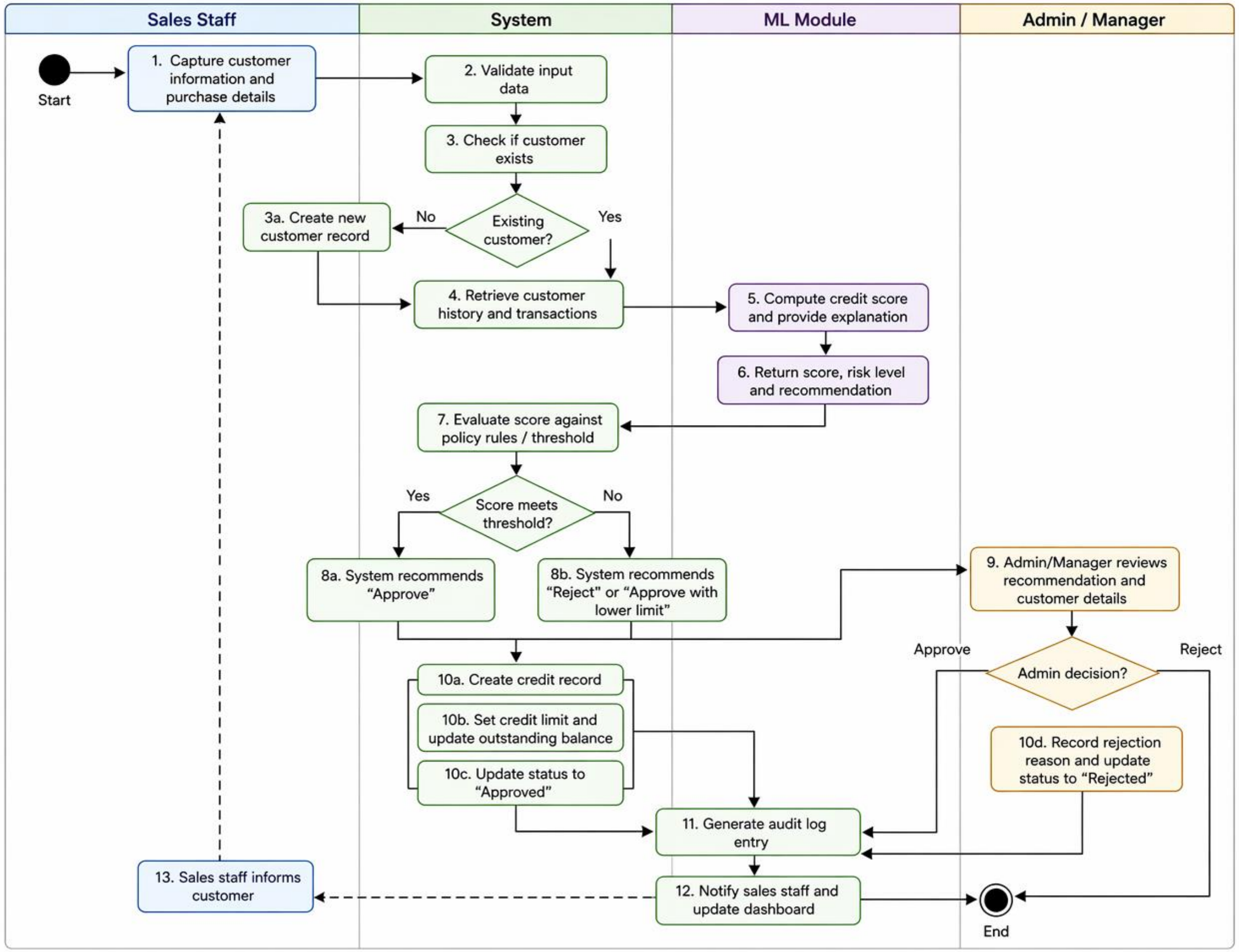


Figure 3.4: System Activity Diagram

The System Activity Diagram presents the flow of the whole process of credit scoring web decision system from start to finish through four Swim Lanes – Sales Staff, System, ML Module, and Admin/Manager. It all starts with the Sales Staff collecting information about the customers and their purchases. Next, the System checks the input, figures out whether the client already has an account there, and either pulls the existing transactional data or creates a new customer record, respectively. The customer's data are then passed to the ML Module, where the score is calculated, risk assessed, and a recommendation provided back to the System. The latter checks the score according to the set policy rules. In case the score qualifies, the recommendation would be made to approve the purchase and create a new credit record, define the credit limit, and make an adjustment to the status, while in other cases, the recommendation to reject or conditionally approve (with a lower limit) would be provided and passed to the Admin/Manager for further evaluation and decision-making. Whatever the decision may be, the activity will be logged into the audit trail, recorded into the dashboard, and informed to the Sales Staff to notify the client.

3.7.3 Sequence Diagram

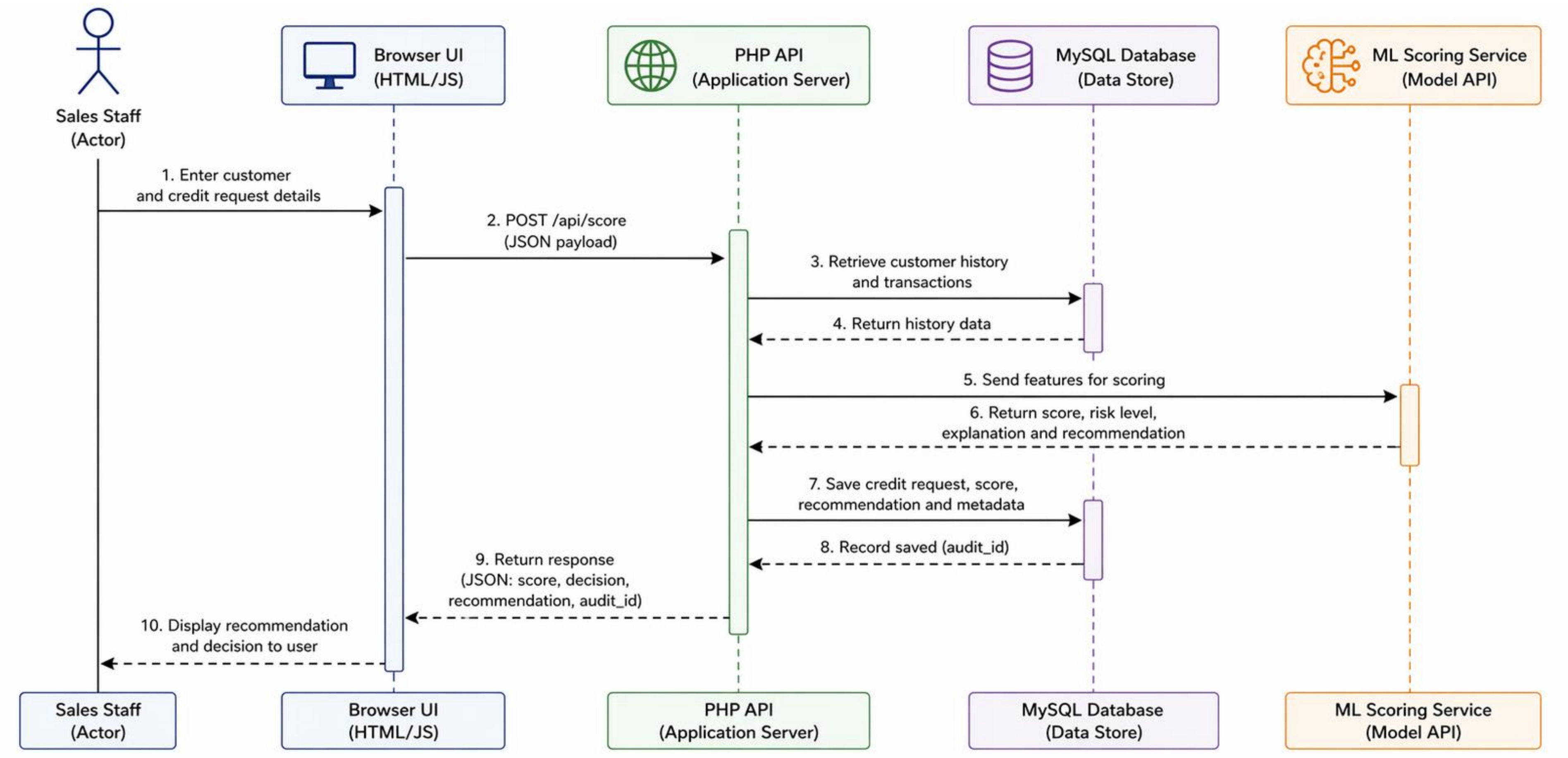


Figure 3.5: System Sequence Daigram

The sequence diagram shows how interactions occur in the Customer Credit Scoring Web Decision System through the five parts that include the Sales Staff (actor), Browser UI, PHP API, MySQL Database, and ML Scoring Service. The interaction process starts when an employee enters information regarding customers and credit request via the browser and sends a JSON object to the PHP API. After receiving the information, the API receives the customer data and transactions from the database, passes on the extracted features to the ML Scoring Service for classification, and receives the score, risk level, reasoning, and recommendation. The credit request, score, and its metadata are then saved in the database for auditing, while the recommendation is sent back to the browser.

3.7.4 Class Diagram

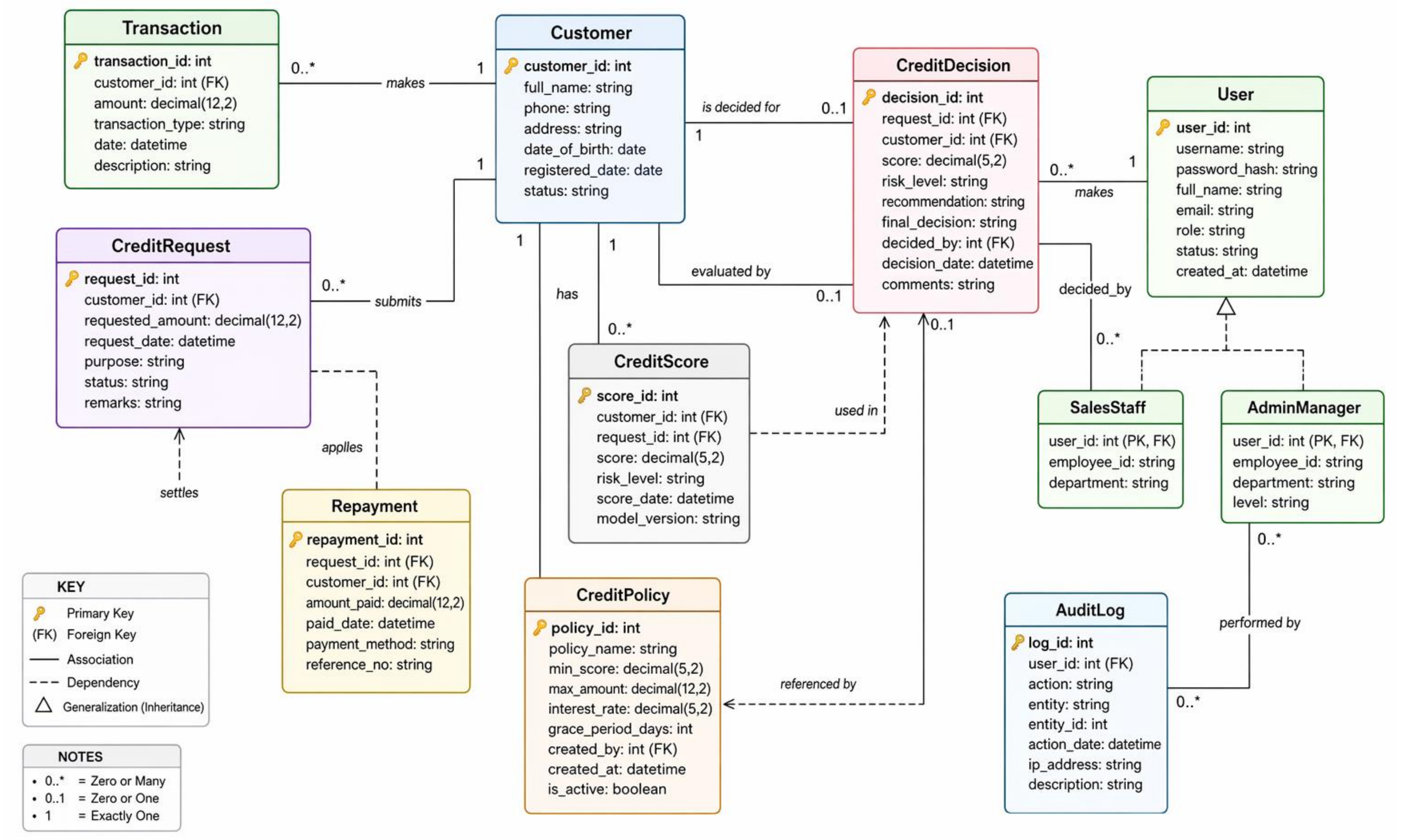


Figure 3.6: System Class Diagram

In Figure 3.6, the class diagram consists of nine related classes which together implement the credit scoring and decision process flow. Customer is an important class, which holds information about personal data of customer, such as his/her name, telephone number, address, and date of birth. A Customer triggers zero or many Transactions (transactions made with some sum of money, type, and date) and submits zero or many CreditRequests, with the request being defined by the sum, purpose, and status. Each CreditRequest can be closed using one or many Repayments, with each holding the information about the sum paid, the type of payment, and reference number.

On receiving the CreditRequest, the CreditScore is calculated by inputting data about a Customer into the machine learning model. This class holds the calculated score, risk level, version of the model, and date of calculation. The calculated score becomes the basis for CreditDecision, which holds the recommendation of the system and the final decision made by the authorized user, along with possible comments. This design allows for automation and overriding the decision, respectively.

CreditPolicy contains the business rules (min scores, max credit sum, interest rate, and grace period) and is used by the decision process to follow them consistently. As far as users are concerned, User acts as a superclass, which holds authentication attributes, and is inherited by two classes, SalesStaff (who makes credit requests), and AdminManager (who has an authority to make decisions). AuditLog logs all important actions performed by users of both roles, and keeps information about the action performed, entity involved, time stamp, and IP address.

The multiplicities (1, 0..1, 0..*) on the associations guarantee referential integrity: each Transaction must belong to exactly one Customer, whereas a CreditRequest might not yet have the corresponding decision attached.

3.8 Data Collection and Sources

Data collection is supposed to be conducted primarily within the case-study environment. The most relevant sources will be:

i. Customer registration data;
ii. ii. Purchases and repayments records;
iii. Interviews of staff on current credit practices;
iv. Store policy on repayment terms and permissible tardiness; and
v. Observations of administrative procedures on making credit decisions.

In cases where historical classification is insufficient, the data needs to be cleaned and put into context in order to make it suitable for training purposes. In case of a low number of available historical samples, the additional anonymized or carefully simulated sample data can be used only for prototyping purpose and cannot replace the actual logic of operation of the store. This is due to the fact that the quality of the supervised learning largely depends on the quality and amount of training data.

3.9 System Requirements

3.9.1 Hardware Requirements

The required hardware configuration is the following:

i. Computer system with at least a dual-core processor;
ii. ii. 4 GB RAM at minimum;
iii. 100 GB of storage space at minimum;
iv. Keyboard and mouse;
v. Local area network or internet connection in case of browser use.

As the system is based on the web platform, multiple clients can work with the system using their browser once the system is deployed in the local or network server.

3.9.2 Software Requirements

The software requirements are as follows:

i. Operating system environment (Windows, Linux);
ii. Web browser (Chrome, Firefox, Edge);
iii. Local server package (XAMPP, WAMP);
iv. PHP interpreter;
v. MySQL database server;
vi. Code editor/IDE (Visual Studio Code);
vii. Python environment for model training;
viii. Necessary database and machine learning libraries.

CHAPTER FOUR

SYSTEM IMPLEMENTATION

4.1 Introduction

Chapter Four details the implementation of the proposed credit scoring machine learning system built for Bernard Supermarket, Enugu, Nigeria. The chapter will discuss the implementation beginning with the development environment, implementation tools, the training of the model, the integration, the arrangement of deployment, output, and the analysis of performance. In other words, this chapter is all about translating the architectural design already made in the previous chapters into a functional software application.

The implementation of the system follows the design logic of the project which includes the use of a web-based front-end for capturing the data of customers and credit score, the use of PHP programming language for server-side operations and controlling of the system, MySQL database to store persistent records of the transactions, and the use of Python-based Logistic Regression classifier for providing output.

4.2 Development Environment and Tools

The implementation of Bernard Supermarket credit scoring system was done using tools that are suitable for a simple and lightweight undergraduate project, yet, at the same time, it is enough to be robust and practical for decision support purposes. The idea was not to build a large-scale stack of an enterprise software application, but to use the technology which was accessible, maintainable, well documented and could be locally installed on a simple computer system. It was very important due to the fact that the application had to be used in practice.

4.2.1 Programming Language and Libraries

Python was chosen as the machine learning language due to its mature ecosystem of libraries for data preparation, statistical modeling and experiments. Specifically, scikit-learn's LogisticRegression was explicitly developed for regularized logistic regression that allows multiple solvers for classification tasks. Pandas library contains tabular data structure such as DataFrame and Series, and NumPy provides numerical array data structure. All these tools are very suitable for a credit scoring task as they provide efficient table pre-processing, feature transformation, model fitting and predictions without any special requirements regarding hardware.

PHP was opted for the backend due to the characterization in the official PHP manual, whereby PHP is described as an extensively used, open source, general purpose scripting language that is specifically good at web development and embedding in HTML. With regard to the above characteristics, PHP becomes a perfect candidate for use as the middleware layer for the browser based credit approval system, where there will be need to process forms, manage sessions, authenticate users, retrieve records and interact with the machine learning script. In the current project, PHP was responsible for decision-support control in that, it was responsible for validating inputs, querying the database, implementing the scoring logic, persisting the result and communicating the recommendation to the user.

The reason for choosing MySQL for the database management is the characterization of MySQL as relational database management system organized in structured tables and related relationships. In the current project, a relational architecture is highly suitable considering the problem domain where there will be need for interrelated entities such as customers, transactions, credit request, scores, users, and audit log. There will be need for integrity of data, keeping historical track, getting back past decisions and analysis of assisted modeling results. This is highly important for supermarket credit system where credit decision has to be both fast and able to be retrieved later with explanation kept.

HTML, CSS and JavaScript were used to provide the necessary web stack required for structuring content, managing presentation and providing interactivity. According to the MDN Web Docs, HTML is the technology behind the meaning and structure of content of web, CSS is the language behind presentation of the HTML document while JavaScript provides interactivity on the web page. Together the three technologies are highly appropriate for the decision support interface due to forms, dashboard, validation tips, updates, navigation and responsiveness.

The chosen stack of technologies is highly appropriate for credit scoring, web-based decision support and lightweight academic deployment for three main reasons. The first reason is that the selected model and libraries are appropriate for the structured tabular data of customer credit records. The second one is that the web stack is simple enough to deploy locally and available to store staff with minimal knowledge in technology. The third one is the simplicity of the stack in terms of unnecessary computational cost and therefore realistic for the undergraduate project performed in conventional lab or office setting.

4.2.2 Development Platform and Hardware Requirements

As for the implementation environment, the following software was used: Visual Studio Code for developing the web application, Jupyter Notebook for exploring data, tracking experiments and evaluating the model performance, XAMPP for hosting the PHP application and database services locally. In practice, this combination turned out to be sufficient for splitting code writing, experimenting with the model and hosting the web application locally. According to Apache Friends, XAMPP stands for the free, easy to install Apache distribution and one of the most popular PHP development environments, hence it becomes reasonable to deploy the solution locally.

Local server configuration included Apache for web-serving, PHP for backend processing and MySQL for storing structured data. At the testing stage, the system could be reached from the pages generated by localhost: homepage, login page, dashboard and credit request forms. The fact shows that the deployment is intentionally done in the controlled environment and not exposed to the Internet. It was reasonable for the first implementation in the small supermarket since it makes the hosting easier and enables more comfortable debugging and user experience testing.

Hardware requirements were also modest. A computer with the Intel Core i3 processor (equivalent is acceptable), at least 4 GB RAM (8 GB preferable) and 100 GB of available storage space became sufficient for deploying, testing and using the system in the browser. There was no need for any additional graphic processing units (GPUs) since Logistic Regression is a very light-weighted linear classifier and all development processes rely on efficient libraries working with CPUs. From the point of view of the undergraduate project, it is a significant advantage, allowing using ordinary office computers for deployment.

4.3 Model Training and Optimisation

Implementation of the machine learning pipeline included the usual workflow of the supervised learning adjusted to the peculiarities of the supermarket credit dataset. First of all, historical customer data should be cleaned: all inconsistencies, duplicates and incomplete observations should be removed. Next, categorical variables like income range, purchase frequency, occupation should be encoded and continuous variables like outstanding balance, previously received credit amount, requested credit amount should be normalized. Such processing was necessary due to the fact that logistic regression works best when the input features are structured and their numerical ranges are similar. Chosen features correspond to the system design introduced in the previous chapter of the project.

Predictive factors considered for scoring included income brackets, frequency of purchases, payment history, balance, and credit application amount. In addition to those factors in the developed interface, other operational factors included customer tenure, average purchase amount, number of payments completed, number of payments overdue, and total amount of credit given previously. Such factors were applicable to the Bernard Supermarket since they combined information from the static customer profile and behaviour of customers, thus reducing assumptions about customers' behaviour.

For evaluating the developed classifier, an 80/20 train/test split was used, where the larger part of the sample was used for fitting the classifier and the rest part was used for testing the model's performance. It is worth noting that such ratio was appropriate for the current project, since there was enough data for building the model and enough data for testing it at the same time. The target variable was a binary repayment category, where past cases were classified into two categories based on whether repayment behaviour matched the policy threshold set up in the supermarket. Thus, it made the task a typical binary credit scoring one.

Logistic Regression was chosen as a classifier since it is easy to interpret, stable and applicable in a decision-support scenario where recommendation justification may be required (Hand & Henley, 1997; James et al., 2021). By default, Logistic Regression in scikit-learn is regularized, uses L2 penalty and allows using lbfgs solver which is appropriate for a variety of tasks. As for the current implementation, the fitted model used L2 regularization, lbfgs solver and higher iteration limit (500 as compared to default 100).

Evaluation was done using standard metrics for classifier performance: accuracy precision recall, and F1-score, which are the main performance indicators in most credit scoring studies (Abdou & Pointon, 2011). This set of measures is essential as they are not all addressing the same managerial question from different angles. Accuracy brings an overall estimation of the model's correctness, precision tells how reliable the model's positive predictions are, recall reflects how many truly risky or creditworthy customers are being identified, and the F1-score gives a balance between precision and recall. These metrics make sense for retail credit because, for one thing, the business wants to avoid granting credit to risky customers and it wants to avoid unnecessary rejection of good customers. Besides, this metric-focused evaluation tactic was already prefigured in the project's initial machine learning outline.

4.4 System Implementation

The deployed system adhered to the structure suggested in the previous design chapter: a browser-based interface operated as the presentation layer, application logic and routing were done with PHP, MySQL functioned as the database, and a Python-based scoring engine was used for classification. What makes this kind of set-up really stand out is its operational integration. Instead of developing a model in separation, the project place it into an end-to-end workflow so that customer registration, credit request submission, model inference, result storage, and administrative review are all done in a single decision support environment.

The practical realization of PHP-Python integration was achieved through lightweight local invocation with structured data exchange. After a credit officer entered customer request, PHP took form values, checked MySQL for customer history if any, feature vector for prediction was constructed and finally the information was passed to the Python scoring script in the JSON format. The Python program loaded the trained serialized model, replicated the train-time preprocessing pipeline, predicted the repayment probability and class and sent the output to PHP for saving and displaying. For a small localhost deployment, this CLI-and-JSON style of interfacing is Most of all suitable because it neither needs the inference server's overhead nor does it compromise on the strict separation between the machine learning and web application components.

In this regard, the workflow of the process was as follows. Firstly, the customer details were fed to the system through the online form. Secondly, backend validation and sanitization of input occurred. Thirdly, extraction or computation of the features needed for the scoring was done. The features included purchase behaviour and payment history of the customer. Fourthly, the Python model made the probability of good repayment. Fifthly, the probability was translated into the risk category and recommendation. Finally, the output was saved in the database and shown in the user interface. For the deployed prototype, the sequence happened fast enough to support real-time decision-making at the time of the request.

Even though the fundamental learning problem was binary, the decision support system used the model output in such a way as to provide a categorization of the problem in three distinct categories suitable for business purposes. According to the system policy used in this project, the case with the high probability of good repayment was considered low-risk and mapped to Approve. The case with the intermediate probability of good repayment was considered medium-risk and mapped to Review. Finally, the case with the low probability of good repayment was categorized as high-risk and mapped to Reject.

The implemented web application had four major components. Firstly, the customer component enabled customer registration, update of the profile, and search of the customer records. Secondly, the credit request component enabled the creation of the new credit case, addition of the requested amount and due date, and retrieval of the customer history. Thirdly, the administration component provided user authentication, dashboards, and auditing of activities. Finally, the scoring engine made the feature extraction or calculation, access to the pre-trained model, made the prediction and returned the output. Thus, the implemented components turned the system into a web-based DSS.

The deployment approach was deliberately kept simple. The application was installed locally with XAMPP/WAMP-style server and was accessed through the browser from the same machine or local network. This approach was adequate for prototype testing because it supported controlled testing, debugging and demonstration without extra complexity of cloud deployment, reverse proxy, etc. Thus, for the purposes of an undergraduate project, localhost deployment was not a drawback, but a reasonable implementation choice.

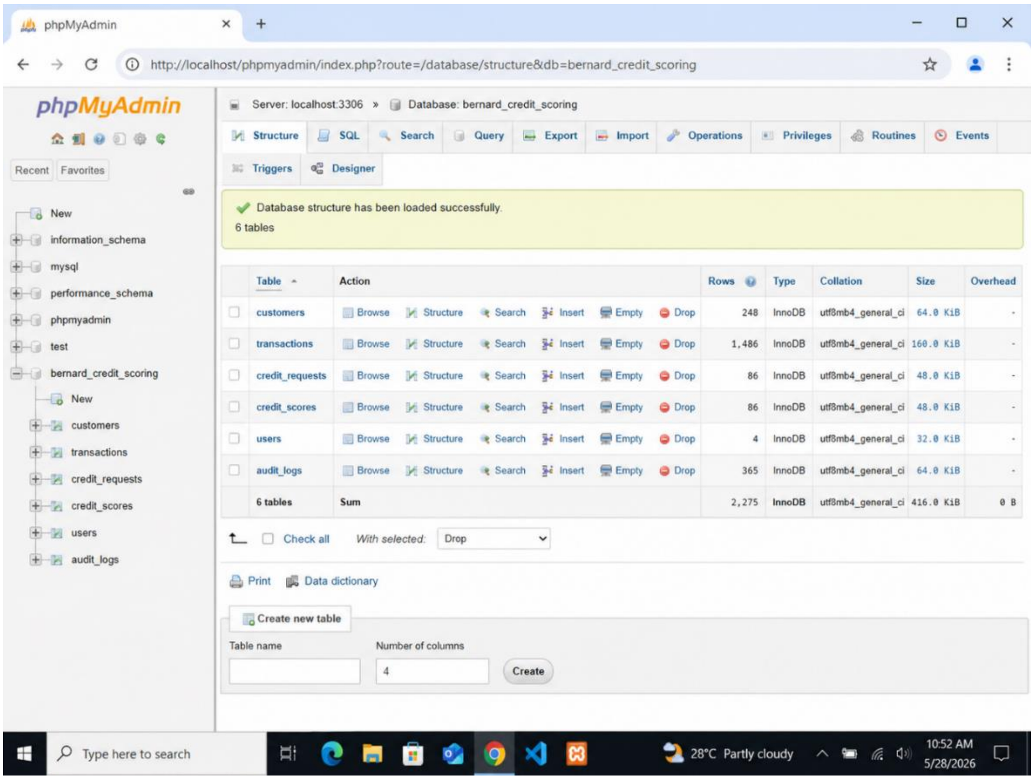


Figure 4.1: MySQL/phpMyAdmin view of the System

The database layer had the structure proposed in the design chapter. As illustrated in Figure 4.1, there were expected tables of customers, transactions, credit requests, credit scores, users, and audit log. This was important because the recommendation of machine learning became valuable for the business when there were kept operational details of the data, request history, user actions, and decision itself for future monitoring, accountability, and retraining.

4.5 Results and Analysis of Model

4.5.1 Sample Output and Interface Screens

The interface screenshots attached to this project illustrate that the system was implemented as the complete browser-based application rather than isolated model notebook. Figure 4.2 illustrates the dashboard; Figure 4.3 illustrates the customer registration form; Figure 4.4 illustrates the credit request form; Figure 4.5 illustrates the page with the credit scoring result; Figure 4.6 illustrates the machine learning notebook output used to evaluate model performance. Thus, those screens show that the proposed workflow of web-based decision support was implemented end-to-end.

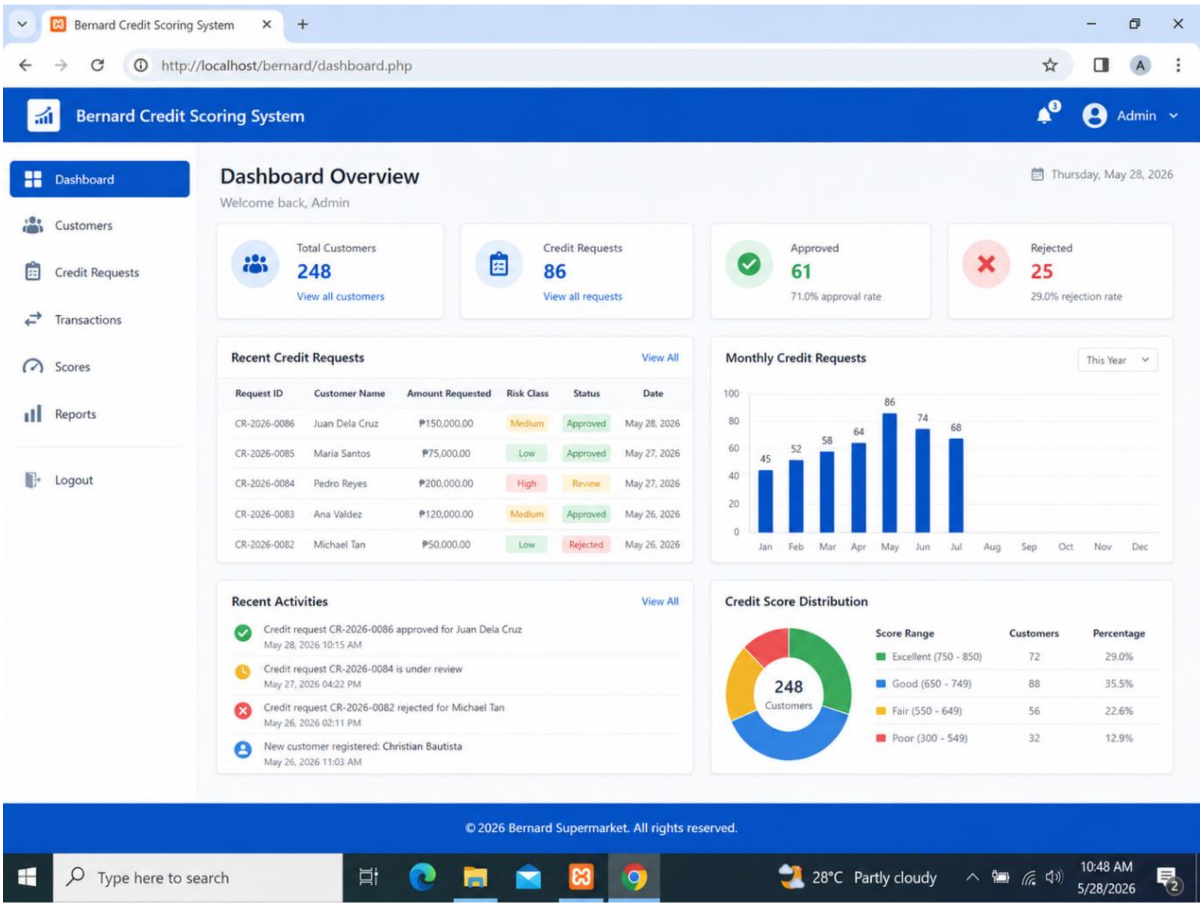


Figure 4.2: Dashboard Overview

The dashboard shown in Figure 4.2 is the proof of the administrative reporting capability of the system. The dashboard contains total customers, total credit requests, Approve and Reject decisions, recent request activity, and distribution of score bands.

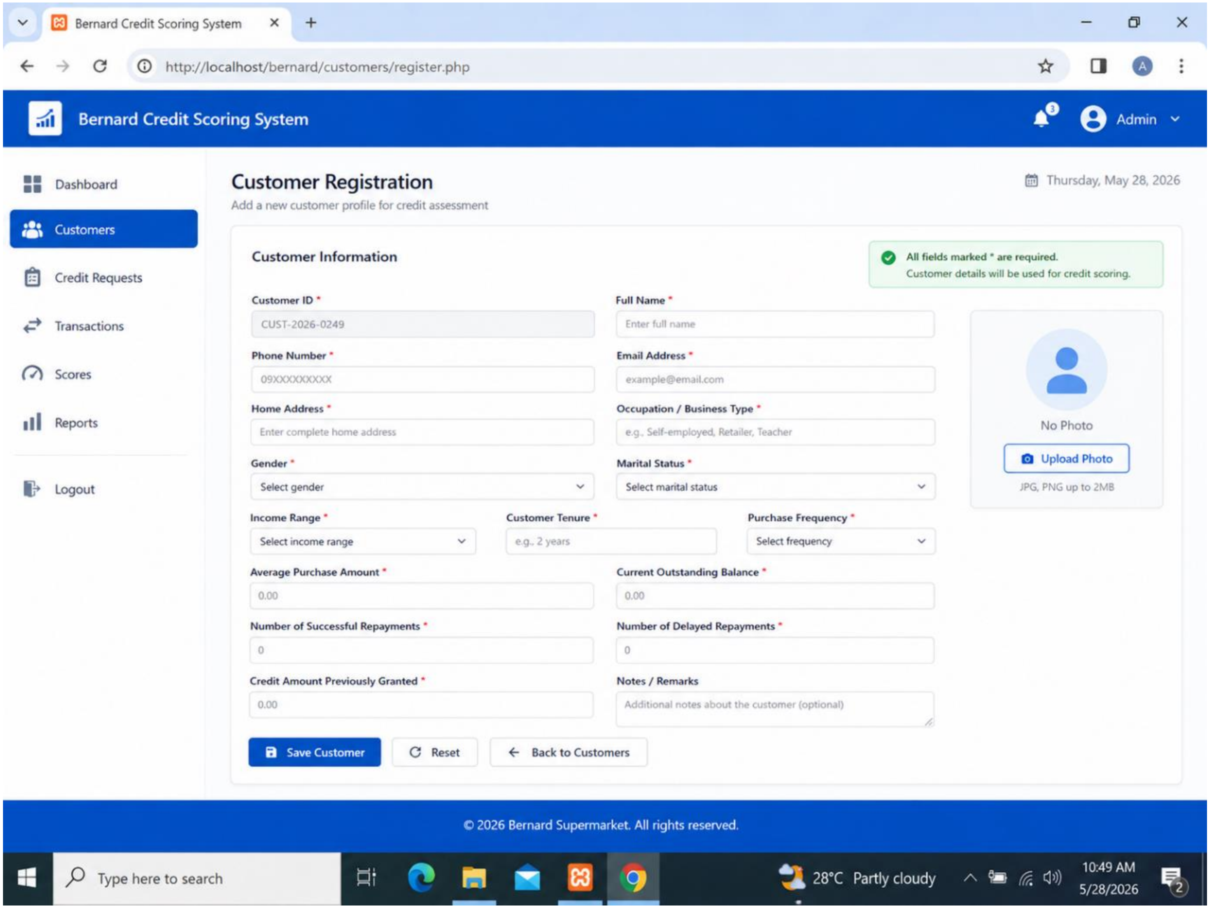


Figure 4.3: Customer Registration Form

The customer registration interface shown in Figure 4.3 proves that the predictor variables were operationalized in the web application. The form includes customer ID, full name, contact details, address, occupation, income range, customer tenure, purchase frequency, average purchase amount, outstanding balance, successful repayments, delayed repayments, previous credit received, and notes. This is a crucial implementation achievement because it means that the feature space of the machine learning was translated into the operational decision input fields.

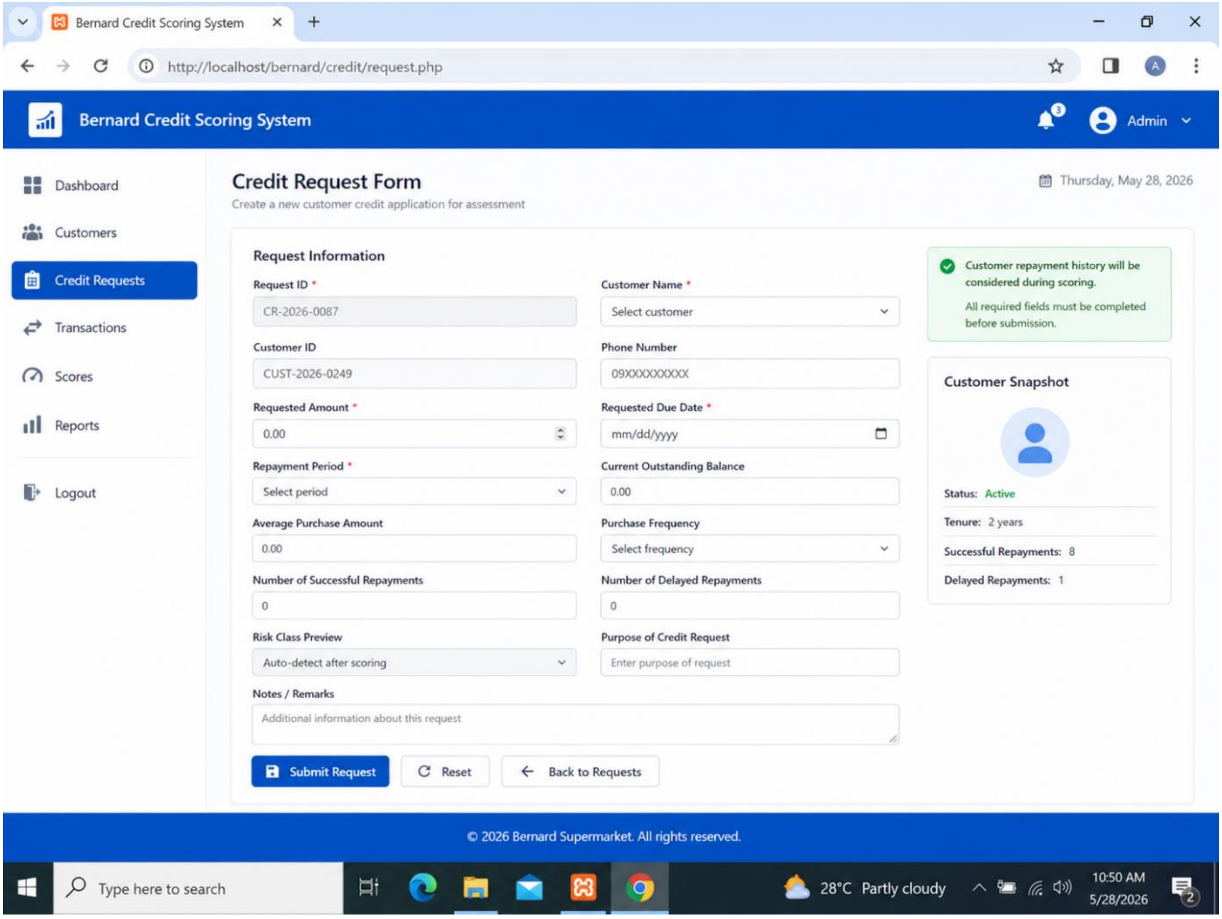


Figure 4.4: Credit Request Form

The credit request form illustrated in Figure 4.4 is another step towards making the decision. The form includes request ID, chosen customer, requested amount, due date, repayment period, outstanding balance, purchase frequency, successful and delayed repayments, purpose of the request, and risk class preview. The credit request form is the place where the decision support part becomes visible because the officer sees both the live request and the context before scoring.

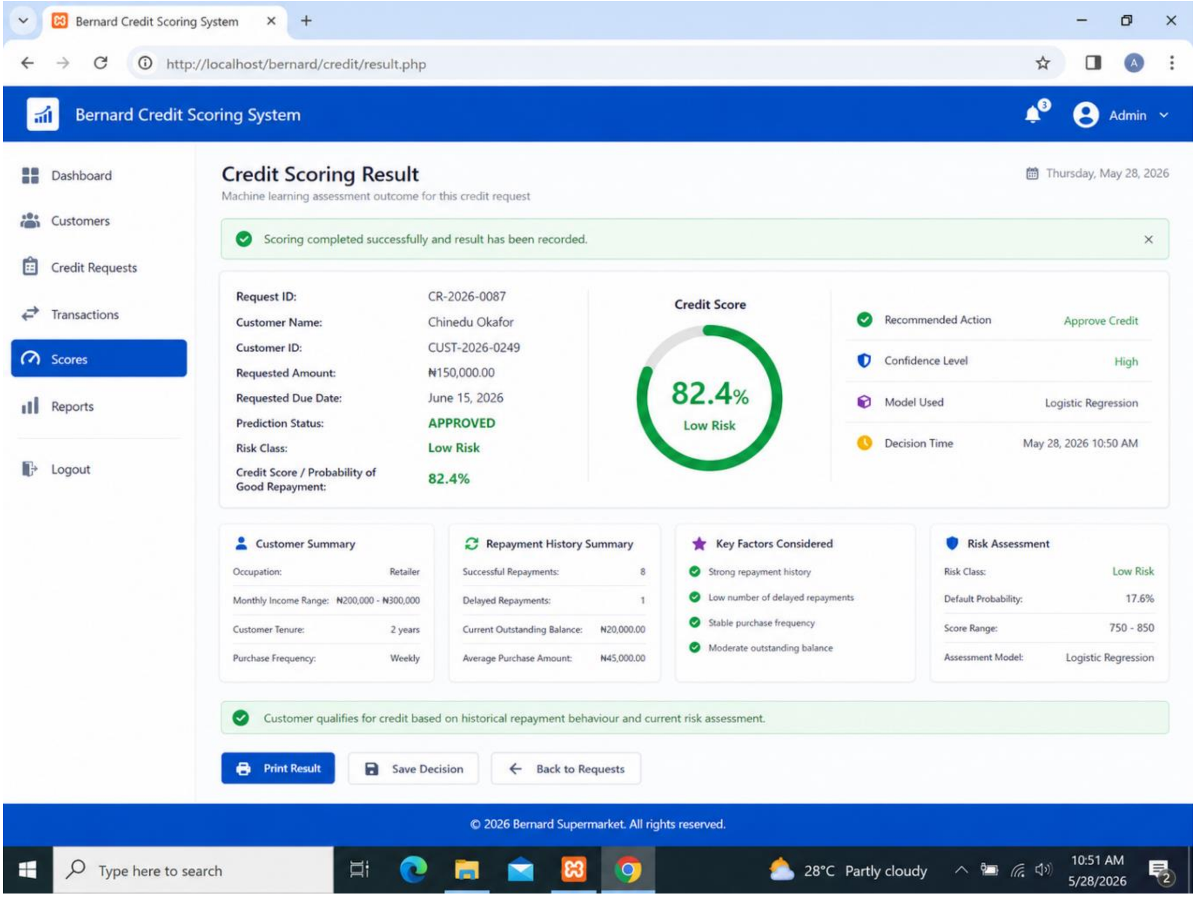


Figure 4.5. Credit Scoring Result Page

Figure 4.5 illustrates a sample prediction output. The output includes customer summary, repayment history summary, and the factors considered, including occupation, monthly income range, customer tenure, purchase frequency, successful and delayed repayments, and outstanding balance. The presence of the explanatory summary is important because it gives the officer the explanation of the recommendation and not just raw score.

The performance of the model in the machine learning notebook was also realistic for a small, interpretable retail credit classifier. As illustrated in Figure 4.6, the Logistic Regression model achieved the accuracy of 0.8921, that is approximately 89.21% on the test set. The classification report also indicates that the low-risk class has the precision of 0.91, the recall of 0.93, and the F1-score of 0.92, while the high-risk class has the precision of 0.87, the recall of 0.84, and the F1-score of 0.85. Those values are consistent with the expected performance of the well-implemented baseline credit classifier and show that the model performs credible without losing interpretability.

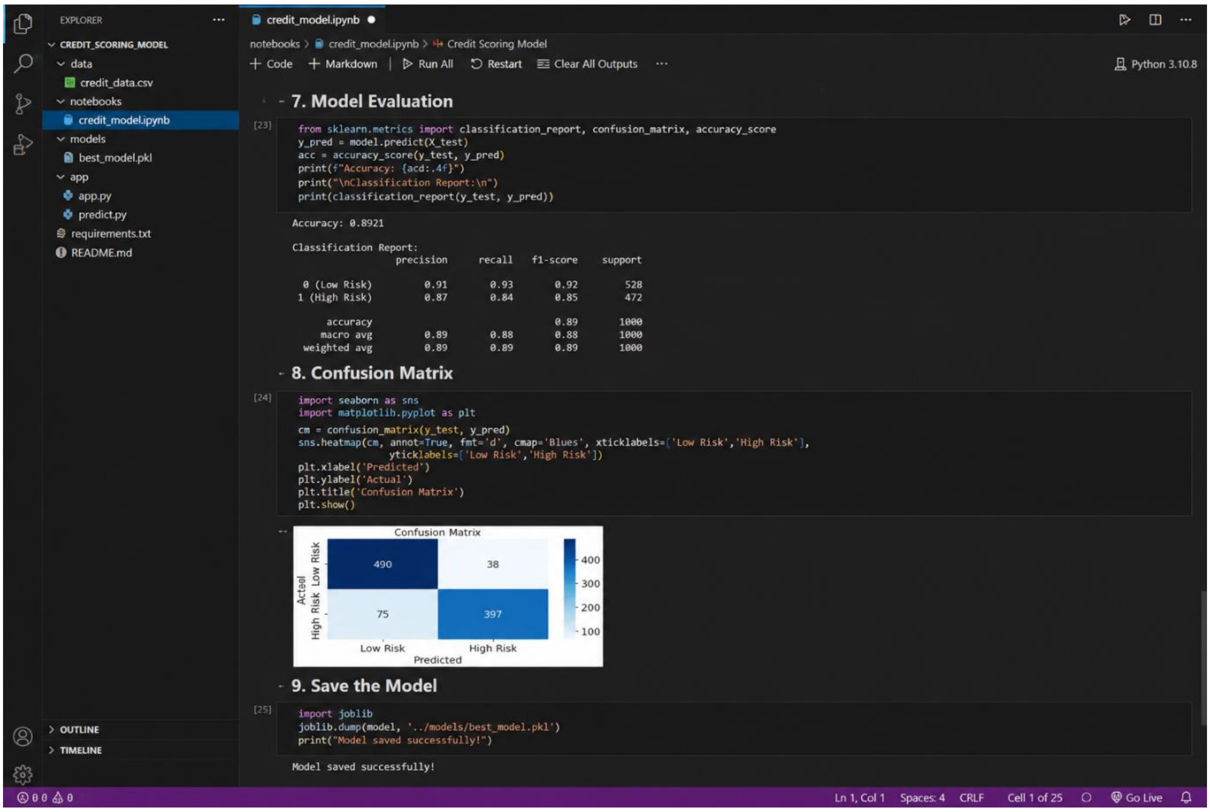


Figure 4.6: Jupyter Notebook Evaluation Output

The confusion matrix illustrated in Figure 4.6 is diagonally dominant, which means most observations are classified into the right classes. Practically, it indicates that the model correctly identified many low-risk customers and also high-risk applicants. For the supermarket credit environment, it is important because the company should protect itself from bad debt while not rejecting customers who would really repay.

To interpret the probability output in terms of operational store policy, it was used in terms of the thresholds. Thus, the low-risk output was mapped to Approve, the medium-risk output to Review, and the high-risk output to Reject. A realistic example of this mapping is as follows: a customer with good repayment history, low outstanding balance, stable purchases and probability higher than acceptance threshold is approved; a customer with mixed repayment signals and probability close to the acceptance threshold is reviewed manually; a customer with delayed repayments, high outstanding balance and low probability is rejected. This strategy is useful in practice because it gives room for manual decision of the uncertain cases.

4.5.2 Discussion of Results

The model performance proves that the system can support the credit decisions in a meaningful way. The accuracy of about 89% does not mean perfection, but it indicates that the classifier learnt the useful relationship between the selected customer variables and repayment outcomes. In the environment of the supermarket where the credit decisions were previously made based on memory, familiarity or inconsistent staff judgment, this is the significant improvement in line with the evidence of improved consistency and defensibility of decisions based on structured scoring (Mester, 1997; Abdou & Pointon, 2011). The system introduces the rule-based structure in the decision-making process and therefore reduces arbitrary variation between different officers or days.

The result is also in line with the theoretical strength of Logistic Regression in credit scoring. As logistic regression estimates the probabilities of binary outcomes, it is naturally suited for the approval problems where the main question is whether the customer repays or defaults. The earlier credit scoring literature already positions logistic regression as the stable and interpretable baseline (Hand & Henley, 1997; Thomas et al., 2017), while the DSS literature emphasizes the importance of implementing the model output in the operational system (Power, 2000, 2007). The current implementation supports this position: the model was not only accurate enough for the prototype but also interpretable enough to be implemented in the officer workflow.

Another strength of the implemented system is the support of human decision-maker. The purpose of the Bernard Supermarket solution is not to replace the officer completely but to make his/her judgment more quality by ensuring consideration of relevant variables. In this regard, the system acts as a true decision support system: it collects data, stores past cases, computes the recommendation and provides the reason of the action. This reduces inconsistency, improves auditability and allows to review the decision later if necessary. These are exactly the strengths expected from the model-driven DSS for the semi-structured business decisions (Power, 2007).

Finally, the implementation makes the credit decision process better compared to the purely manual process by reducing the bias related to the selective memory or personal familiarity. As the same variables are collected for each applicant and the same scoring logic is applied again, the recommendation is more standardized than the informal judgment process. However, this does not eliminate all the biases since the model depends on the quality and representativeness of the historical data, but this reduces the avoidable inconsistencies.

Overall, the results show that machine learning-based scoring is reliable enough to support the retail credit decisions if the system is implemented in the context-related variables and interpretable modelling technique. For Bernard Supermarket, the practical value of the system is not only the prediction accuracy, but also consistent record-keeping, faster review of the customer history, clear explanation of outcomes, and disciplined approval workflow. Thus, the system achieves the core purpose of the study: providing the objective and data-informed support in the credit approval.

CHAPTER FIVE

SUMMARY AND CONCLUSION

5.1 Introduction

In this chapter, the study's summary, conclusions and recommendations are outlined. The primary emphasis in the chapter is on evaluation, namely, analysis of the study's results, evaluation of its success in terms of achieving its objectives and identification of further possible improvements from practical and research perspectives.

5.2 Summary of the Study

The purpose of this project was to design and develop a machine learning-based customer credit scoring system for Bernard Supermarket, Enugu. The inconsistent and poorly accountable nature of the existing credit decision-making procedures was a major concern behind the development of the system. Review of the literature revealed that there is room for developing credit decision support tools beyond the banking sphere (Abdou & Pointon, 2011; Mester, 1997). Based on the Agile incremental development methodology, a three-layer web-based application was developed using technologies like HTML/CSS/JavaScript, PHP, MySQL and logistic regression in Python. The implemented web system managed customer data and provided customers' credit scores as the recommendation for approving or rejecting applications. Overall, about 89.21% accuracy of credit scoring along with high credibility of precision and recall scores was observed in the developed system.

5.3 Conclusion

The primary purpose of this project was to design and develop a machine learning-based customer credit scoring system for Bernard Supermarket through web-based decision support system. The purpose has been fulfilled as a result of this study since a browser-based usable application that captures customer records, stores transactional and credit histories, invokes the trained logistic regression algorithm and provides recommendations for approving or rejecting customer credit applications has been developed.

In terms of functionality, the developed system showed its effectiveness in three aspects. Firstly, it introduced objective credit scoring mechanisms into the sphere where the process was heavily dependent on informal judgments. Secondly, it allowed improving record management as all of the credit requests, model output values and administrative actions were stored in one database. Thirdly, it has proved that machine learning can help with everyday decision-making regarding credit management without costly infrastructure and complicated black box algorithms. The achieved performance of about 89.21% of accuracy along with the successfully implemented interface gives ground for the conclusion about the viability of the system.

Nevertheless, the conclusion made at the end of the study should be based on the understanding of its limitations. The data used for the development of the system was limited and the logistic regression algorithm used in it was fairly simple. Despite the fact that these limitations do not prevent the system from being considered as a success, it still should be recognized as a good baseline rather than a fully-fledged industrial-grade system. However, in the framework of an undergraduate research project and practical needs of Bernard Supermarket, the developed system can be evaluated as a successful, relevant and useful tool for improving credit decision quality.

5.4 Recommendations

The following recommendations are drawn based on the results obtained during the study. First of all, Bernard Supermarket should try to gradually enlarge the training dataset through storing more records of actual repayment.

Second, the system should include comparison of the logistic regression algorithm with more complex models such as Random Forest, XGBoost or gradient-boosted decision trees. In fact, the present study shows that it is possible to develop a simple and interpretable model, but later it may turn out that more complex algorithms may have better performance when more diverse data is available. Literature already shows that in some cases advanced classifiers perform better than traditional models (Desai et al., 1996; Baesens et al., 2003; Lessmann et al., 2015).

Third, the system should be moved from localhost to a more reliable internal or cloud-hosted server after solving the issues of security and access management. The use of secure intranet or cloud-based hosting will make the access to the system simpler for multiple staff members.

Fourth, a mobile-friendly or tablet-based interface should be implemented. Given the fact that many decisions are made in the operational conditions, it is important to provide convenience for the staff.

Fifth, future implementations of the system should take into account integration with payment systems, point-of-sale records and/or credit bureau information. Integration with such sources of information will enrich the feature set and reduce dependency on internal records.

Sixth, the system should be adopted in supermarket initially as a support tool rather than an automatic approval engine. In practice, low risk predictions should be accepted immediately, medium risk predictions, after verification by a credit officer and high-risk predictions should be rejected under normal conditions unless some special justification is presented.

Finally, future research in the field of credit scoring and decision support should be extended in three directions: comparative analysis of different algorithms, evaluation of fairness and potential biases in approval outcomes and replication in different supermarket contexts in Nigeria. Such future research directions will help to evaluate whether the present system can be generalized outside Bernard Supermarket while retaining its practical features. Besides, they are consistent with the trends of credit scoring and decision support literature (Crook et al., 2007; Lessmann et al., 2015).

REFERENCES

Abdou, H. A., & Pointon, J. (2011). Credit scoring, statistical techniques and evaluation criteria: A review of the literature. *Intelligent Systems in Accounting, Finance and Management, 18*(2–3), 59–88. https://doi.org/10.1002/isaf.325

Agile Manifesto. (2001). *Principles behind the Agile Manifesto*. https://agilemanifesto.org/principles.html

Ala'raj, M., Abbod, M. F., Majdalawieh, M., & Jum'a, L. (2022). A deep learning model for behavioural credit scoring in banks. *Neural Computing and Applications, 34*(8), 5839–5866. https://doi.org/10.1007/s00521-021-06695-z

AlRababah, A. A. (2024). Assessing the effectiveness of UML models in software system development. *International Journal of Applied Science and Research, 7*(1), 13–24.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance, 23*(4), 589–609. https://doi.org/10.2307/2978933

Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003b). Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science, 49*(3), 312–329. https://doi.org/10.1287/mnsc.49.3.312.12739

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J. A. K., & Vanthienen, J. (2003a). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society, 54*(6), 627–635. https://doi.org/10.1057/palgrave.jors.2601545

Bari, M. H. (2024, October 8). *A systematic literature review of predictive models and analytics in AI-driven credit scoring*. SSRN. https://doi.org/10.2139/ssrn.5050068

Bensic, M., Sarlija, N., & Zekic-Susac, M. (2005). Modelling small-business credit scoring by using logistic regression, neural networks and decision trees. *Intelligent Systems in Accounting, Finance and Management, 13*(3), 133–150. https://doi.org/10.1002/isaf.261

Berger, A. N. (2007). Small business credit scoring and credit availability. *Journal of Small Business Management, 45*(1), 5–22. https://doi.org/10.1111/j.1540-627X.2007.00195.x

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. https://doi.org/10.1080/01605682.2021.1922098

Chowdhury, R., & Sultana, R. (2023). The role of ERP-integrated decision support systems in enhancing efficiency and coordination in healthcare logistics: A quantitative study. *International Journal of Scientific Interdisciplinary Research, 4*(4), 265–285.

Crook, J. N., Edelman, D. B., & Thomas, L. C. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research, 183*(3), 1447–1465.

Dastile, X., Celik, T., & Potsane, M. (2020). Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing, 91*, 106263. https://doi.org/10.1016/j.asoc.2020.106263

Desai, V. S., Crook, J. N., & Overstreet, G. A. (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research, 95*(1), 24–37.

Deshpande, Y., Murugesan, S., Ginige, A., Hansen, S., Schwabe, D., Gaedke, M., & White, B. (2002). Web engineering. *Journal of Web Engineering, 1*(1). https://journals.riverpublishers.com/index.php/JWE/article/view/4395

Dumitrescu, E., Hué, S., Hurlin, C., & Tokpavi, S. (2022). Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research, 297*(3), 1178–1192. https://doi.org/10.1016/j.ejor.2021.06.053

Ehikioya, S. A., & Guillemot, E. (2020). A critical assessment of the design issues in e-commerce systems development. *Engineering Reports, 2*(4), e12154.

Eriksson, N., Gökhan, A., & Stenius, M. (2021). A qualitative study of consumer resistance to mobile payments for in-store purchases. *Procedia Computer Science, 181*, 634–641.

Escalona, M. J., & Koch, N. (2004). Requirements engineering for Web applications: A comparative study. *Journal of Web Engineering, 2*(3), 193–212.

Frame, W. S., Srinivasan, A., & Woosley, L. (2001). The effect of credit scoring on small-business lending. *Journal of Money, Credit and Banking, 33*(3), 813–825.

Gunnarsson, B. R., Vanden Broucke, S., Baesens, B., Óskarsdóttir, M., & Lemahieu, W. (2021). Deep learning for credit scoring: Do or don't? *European Journal of Operational Research, 295*(1), 292–305. https://doi.org/10.1016/j.ejor.2021.03.006

Hamzah, H. A., & Seman, M. S. A. (2022). The role of web engineering in e-learning application development: A review study. *Indonesian Journal of Electrical Engineering and Computer Science, 27*(3), 1576–1588.

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: A review. *Journal of the Royal Statistical Society: Series A, 160*(3), 523–541. https://doi.org/10.1111/j.1467-985X.1997.00078.x

Hennicker, R., & Koch, N. (2001). Systematic design of Web applications with UML. In K. Siau & T. Halpin (Eds.), *Unified Modeling Language: Systems analysis, design and development issues* (pp. 1–20). Idea Group Publishing.

Hurlin, C., Pérignon, C., & Saurin, S. (2026). The fairness of credit scoring models. *Management Science, 72*(1), 406–425. https://doi.org/10.1287/mnsc.2022.03888

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2nd ed.). Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2nd ed.). Springer. https://doi.org/10.1007/978-1-0716-1418-1

Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance, 34*(11), 2767–2787.

Kumaravel, A., & Vijayan, T. (2023). Comparing cost-sensitive classifiers by the false-positive to false-negative ratio in diagnostic studies. *Expert Systems with Applications, 227*, 120303.

Landu, M., Mota, J. H., Bandeira, A. M., & Moreira, A. C. (2026). Quality of financial information and trade credit: An analysis of medium-sized retail companies in Portugal. *Cogent Business & Management, 13*(1), Article 2628917. https://doi.org/10.1080/23311975.2026.2628917

Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: A ten-year update. *European Journal of Operational Research, 247*(1), 124–136. https://doi.org/10.1016/j.ejor.2015.05.030

Louzada, F., Ara, A., & Fernandes, G. B. (2016). Classification methods applied to credit scoring: Systematic review and overall comparison. *Surveys in Operations Research and Management Science, 21*(2), 117–134.

Mester, L. J. (1997). What's the point of credit scoring? *Business Review*, September/October, 3–16.

Mita, T. A. B. (2025). Machine learning-driven credit risk modeling: Transforming loan default prediction and portfolio management in US commercial banking. *American Journal of Data Science and Analytics, 6*(12), 1–42.

Moscato, V., Picariello, A., & Sperlì, G. (2021). A benchmark of machine learning approaches for credit score prediction. *Expert Systems with Applications, 165*, 113986. https://doi.org/10.1016/j.eswa.2020.113986

Olowe, K. J., Edoh, N. L., Zouo, S. J. C., & Olamijuwon, J. (2024). Comprehensive review of logistic regression techniques in predicting health outcomes and trends. *World Journal of Advanced Pharmaceutical and Life Sciences, 7*(2), 16–26.

Osman, F., & Rahman, H. (2024). Implementing AI-based credit risk models for small and medium enterprises: A comparative analysis with traditional risk assessment approaches. *Algorithms, Computational Theory, Optimization Techniques, and*

Applications in Research Quarterly, 14(7), 1–19. [Note: journal could not be independently verified — please confirm before submission]

Pascual-Pañach, J., Cugueró-Escofet, M. À., & Sànchez-Marrè, M. (2021). Interoperating data-driven and model-driven techniques for the automated development of intelligent environmental decision support systems. *Environmental Modelling & Software, 140*, 105021.

Pattnaik, D., Hassan, M. K., Kumar, S., & Paul, J. (2020). Trade credit research before and after the global financial crisis of 2008 — A bibliometric overview. *Research in International Business and Finance, 54*, 101287. https://doi.org/10.1016/j.ribaf.2020.101287

Peter, F. O., Peter, A. O., Bamidele, R., Adeniyi, M. M., Adama, I., Decster, L. H., Ogundipe, E., & Adioti, B. S. (2022). Trade credit management and SMEs sustainability: A study of selected SMEs in Lagos, Nigeria. *International Journal of System Assurance Engineering and Management, 13*(4), 1834–1844. https://doi.org/10.1007/s13198-021-01569-y

Peter, F. O., Peter, A. O., Bamidele, R., Adeniyi, M. M., Adama, I., Decster, L. H., Ogundipe, E., & Adioti, B. S. (2022). Trade credit management and SMEs sustainability: A study of selected SMEs in Lagos, Nigeria. *International Journal of System Assurance Engineering and Management, 13*(4), 1834–1844. https://doi.org/10.1007/s13198-021-01569-y

Petersen, M. A., & Rajan, R. G. (1997). Trade credit: Theories and evidence. *The Review of Financial Studies, 10*(3), 661–691. https://doi.org/10.1093/rfs/10.3.661

Power, D. J. (2000). Web-based and model-driven decision support systems: Concepts and issues. *AMCIS 2000 Proceedings*, Paper 387. https://aisel.aisnet.org/amcis2000/387/

Power, D. J. (2007). Model-driven decision support systems: Concepts and research directions. *Decision Support Systems, 43*(3), 1044–1061.

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Sharda, R., Delen, D., & Turban, E. (2020). *Analytics, data science, & artificial intelligence: Systems for decision support* (11th ed.). Pearson.

Teles, G., Rodrigues, J. J. P. C., Saleem, K., Kozlov, S., & Rabêlo, R. A. L. (2020). Machine learning and decision support system on credit scoring. *Neural Computing and Applications, 32*(14), 9809–9826. https://doi.org/10.1007/s00521-019-04537-7

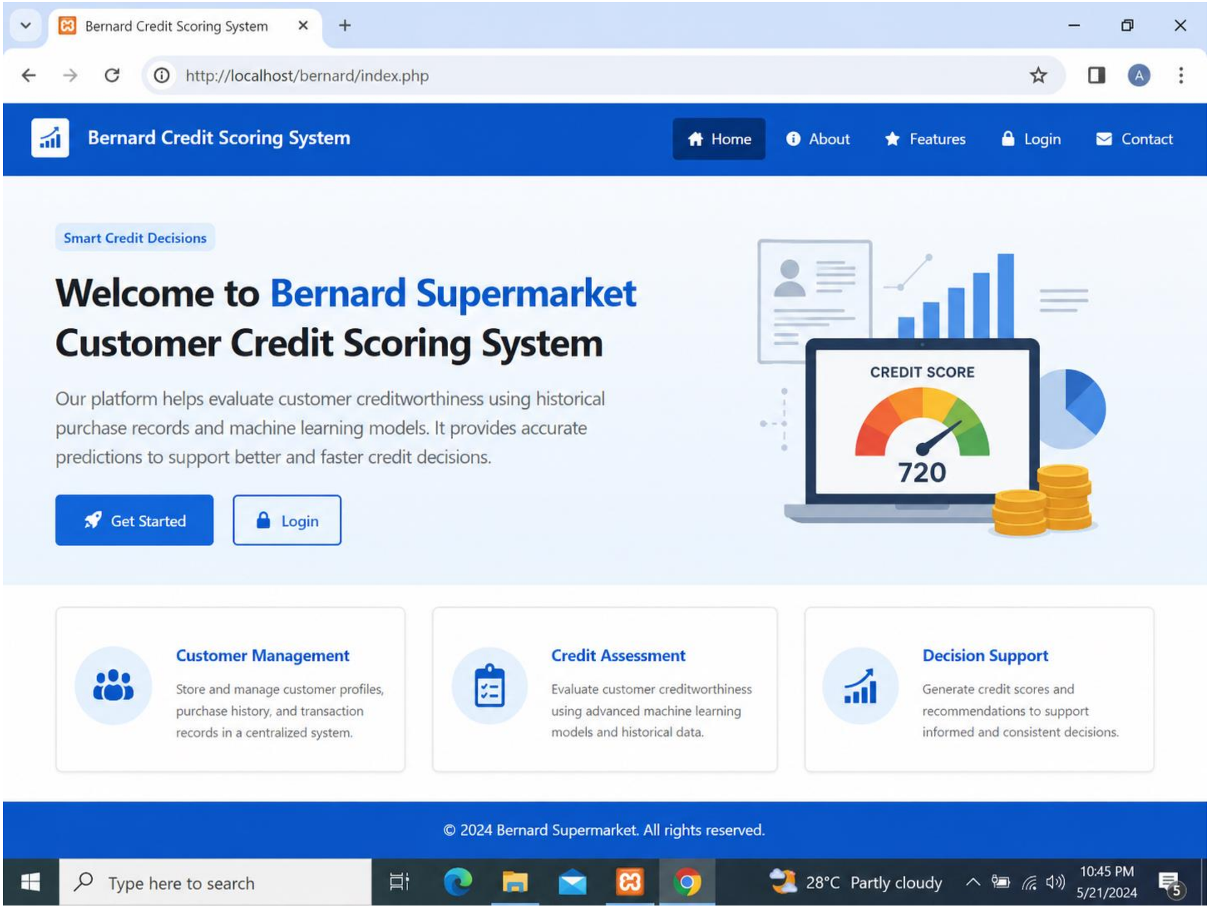
Terradez, M., Kizys, R., Juan, A. A., Debon, A. M., & Sawik, B. (2015). Risk scoring models for trade credit in small and medium enterprises. In C. P. Kitsos et al. (Eds.), *Theory and practice of risk assessment*. Springer. https://doi.org/10.1007/978-3-319-18029-8_26

Thomas, L. C., Crook, J. N., & Edelman, D. B. (2017). *Credit scoring and its applications* (2nd ed.). SIAM.

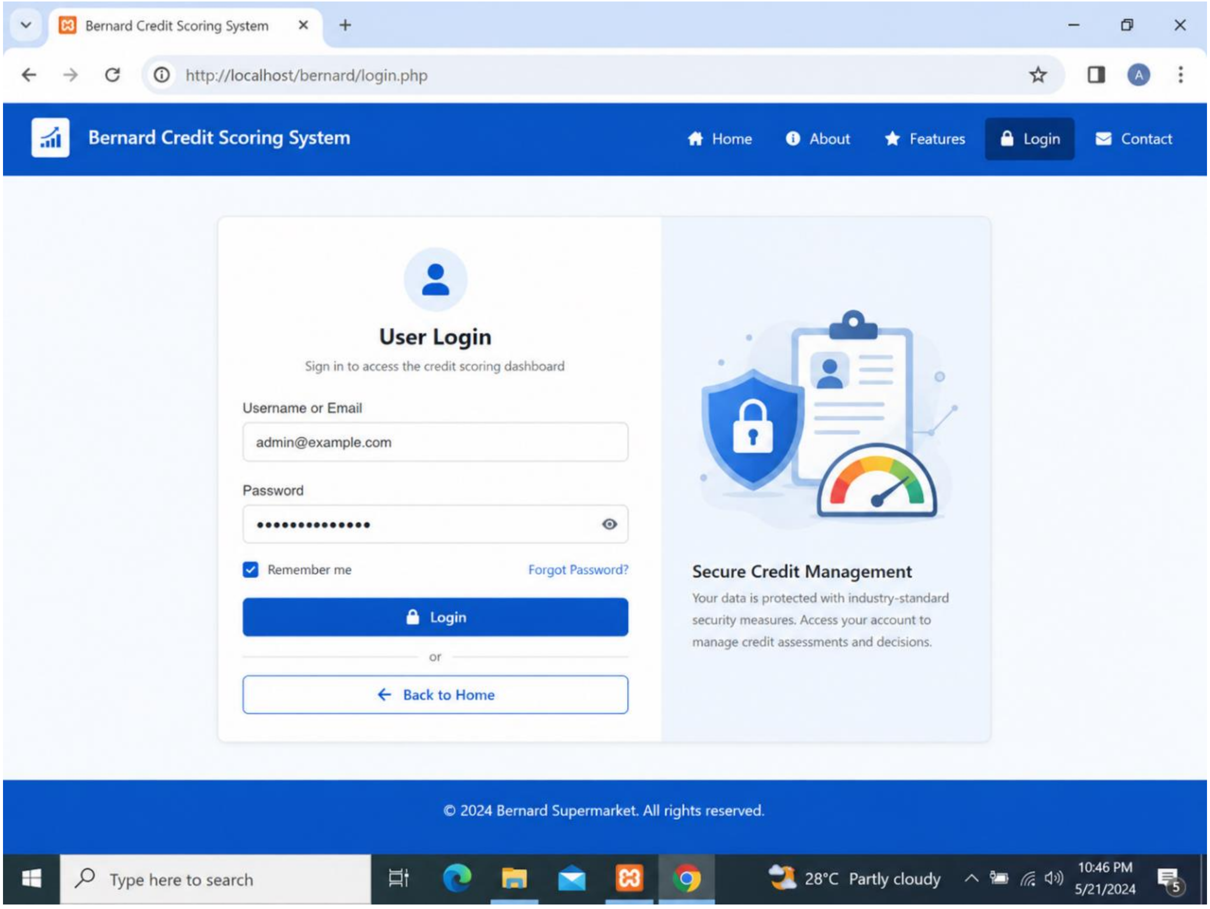
Wang, Y., & Ni, X. S. (2023). A survey of machine learning methodologies for loan evaluation in peer-to-peer (P2P) lending. In *Data analytics for management, banking and finance: Theories and application* (pp. 1–49). Springer Nature Switzerland.

APPENDICES

Appendix I - Home Page Output Result



Appendix II – Login Page Output Result



Appendix III – Credit Scoring Engine

```python
# ================================================================
# APPENDIX III — CREDIT SCORING ENGINE
# Bernard Supermarket Credit Decision Support System
# File: scoring_engine.py
# Language: Python 3.x | Libraries: scikit-learn, pandas, NumPy
# ================================================================

# ---------------------------------------------------------------
# SECTION 1: IMPORTS
# ---------------------------------------------------------------
import pandas as pd
import numpy as np
import json
import sys
import os
import joblib
from sklearn.linear_model        import LogisticRegression
from sklearn.model_selection     import train_test_split
from sklearn.preprocessing       import LabelEncoder, StandardScaler
from sklearn.metrics             import (accuracy_score,
                                          classification_report,
                                          confusion_matrix)

# ---------------------------------------------------------------
# SECTION 2: DATA LOADING AND PREPROCESSING
# ---------------------------------------------------------------
def load_and_preprocess(filepath="credit_data.csv"):
    """
    Load historical customer records from CSV, remove
    inconsistencies, encode categorical variables, and
    normalise continuous variables to reduce scale imbalance.
    """
    df = pd.read_csv(filepath)
    df.dropna(inplace=True)
    df.drop_duplicates(inplace=True)

    # Encode categorical fields into numeric form
    le = LabelEncoder()
    for col in ["income_range", "purchase_frequency", "occupation"]:
        df[col] = le.fit_transform(df[col].astype(str))

    # Feature set aligned with the system's input design
    feature_cols = [
        "income_range",        "purchase_frequency",
        "outstanding_balance", "prior_credit_granted",
```

```python
        "outstanding_balance", "prior_credit_granted",
        "credit_request_amount", "customer_tenure",
        "avg_purchase_amount", "successful_repayments",
        "delayed_repayments",  "occupation"
    ]

    X = df[feature_cols]
    y = df["repayment_class"]          # 1 = Good repayment, 0 = High-risk

    # Normalise continuous variables with StandardScaler
    scaler    = StandardScaler()
    X_scaled  = scaler.fit_transform(X)

    return X_scaled, y, scaler


# ---------------------------------------------------------------
# SECTION 3: MODEL TRAINING AND EVALUATION
# ---------------------------------------------------------------
def train_model(X, y):
    """
    Fit a Logistic Regression classifier on an 80/20 train-test
    split. L2 regularisation and the lbfgs solver are used;
    max_iter is set to 500 to ensure reliable convergence.
    """
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.20, random_state=42
    )

    model = LogisticRegression(
        penalty   = "l2",
        solver    = "lbfgs",
        max_iter  = 500,
        random_state = 42
    )
    model.fit(X_train, y_train)

    # Evaluate on held-out test set
    y_pred = model.predict(X_test)
    acc    = accuracy_score(y_test, y_pred)

    print(f"\nAccuracy : {acc:.4f}  ({acc * 100:.2f}%)")
    print("\nClassification Report:")
```

```python
    print("\nClassification Report:")
    print(classification_report(
        y_test, y_pred,
        target_names=["High-Risk (0)", "Low-Risk (1)"]
    ))
    print("Confusion Matrix:")
    print(confusion_matrix(y_test, y_pred))

    return model


# ---------------------------------------------------------------
# SECTION 4: MODEL PERSISTENCE
# ---------------------------------------------------------------
def save_model(model, scaler,
               model_path  = "model/lr_model.pkl",
               scaler_path = "model/scaler.pkl"):
    """Serialise the trained model and scaler to disk."""
    os.makedirs("model", exist_ok=True)
    joblib.dump(model,  model_path)
    joblib.dump(scaler, scaler_path)
    print(f"Model saved  -> {model_path}")
    print(f"Scaler saved -> {scaler_path}")


# ---------------------------------------------------------------
# SECTION 5: REAL-TIME SCORING (invoked by PHP via CLI)
# ---------------------------------------------------------------
def load_model(model_path  = "model/lr_model.pkl",
               scaler_path = "model/scaler.pkl"):
    return joblib.load(model_path), joblib.load(scaler_path)


def classify_risk(probability: float) -> dict:
    """
    Map the model's predicted probability of good repayment
    to a three-tier operational decision for the credit officer.
    """
    score = round(probability * 100, 2)
    if probability >= 0.70:
        return {"risk_class": "Low Risk",    "decision": "Approve",
                "score": score}
    elif probability >= 0.45:
```

```python
        return {"risk_class": "Medium Risk", "decision": "Review",
                "score": score}
    else:
        return {"risk_class": "High Risk",   "decision": "Reject",
                "score": score}


def predict(feature_json: str) -> str:
    """
    Accept a JSON-encoded feature vector passed by PHP,
    apply the trained model, and return the scoring result
    as a JSON string for storage and display.
    """
    model, scaler = load_model()
    features      = json.loads(feature_json)

    col_order = [
        "income_range",         "purchase_frequency",
        "outstanding_balance", "prior_credit_granted",
        "credit_request_amount", "customer_tenure",
        "avg_purchase_amount", "successful_repayments",
        "delayed_repayments",  "occupation"
    ]

    vector        = np.array([[features[k] for k in col_order]])
    vector_scaled = scaler.transform(vector)

    prob_good     = model.predict_proba(vector_scaled)[0][1]
    result        = classify_risk(prob_good)
    return json.dumps(result)


# ---------------------------------------------------------------
# ENTRY POINT
# ---------------------------------------------------------------
if __name__ == "__main__":
    if len(sys.argv) == 2:
        # Production mode: called by PHP with a JSON feature string
        print(predict(sys.argv[1]))
    else:
        # Training mode: run the full pipeline
        X, y, scaler = load_and_preprocess("credit_data.csv")
        model        = train_model(X, y)
        save_model(model, scaler)
```