**Ministry of Higher Education and Scientific Research

University of Technology

College of Computer Engineering

Information Engineering Department

**Iraqi Banks loan status prediction using ML algorithms**

Graduation project submitted to the College of Computer Engineering in Partial fulfillment of B.Sc. degree in Information Engineering

**By**

*Ali Mortada Abod Bashar Raed Abdullhay*

**Supervised by**

*Senior Lecture Enas A. Raheem*

2024-2025



**SUPERVISOR CERTIFICATION**

I certify that the preparation of this project entitled (Iraqi banks loan status prediction using ML algorithms) was prepared by (**Ali Mortada Abod and Bashar Raed Abdullhay**) under my supervision at the Information Engineering Department, College of Computer Engineering, the University of Technology in partial fulfilment of the requirements for the B.Sc. degree in Information Engineering.

**Signature:**

**Name:**

**Scientific Degree:**

**Date:**

**EXAMINATION COMMITTEE CERTIFICATION**

We certify that we have read this project entitled (Student Attendance Management System Software for the College of Computer Engineering), and as an examining committee examined the students (Ali Mortada Abod & Bashar Raed Abdullhay) in its contents, and our opinion, it meets the standards of the B.Sc. in Information Engineering.

Signature: Signature:

Name: Name:

Scientific Degree: Scientific Degree:

Date: Date:

(Member) (Member)

Signature: Signature:

Name: Name:

Scientific Degree: Scientific Degree:

Date: Date:

(Member) (Chairman)

Signature:

Name:

Scientific Degree:

Date:

Head of (Information) Engineering Department

# Dedication

# ***My university journey has come to an end after much fatigue and hardship.***

***Here I am, completing my graduation thesis with enthusiasm and energy.***

***I am grateful to everyone who has helped me along the way: my family, friends, and esteemed professors.***

***I dedicate my graduation thesis to you.***

***Acknowledgment***

***"The satisfaction and euphoria that accompanies the successful completion of any project would be incomplete without the mentioning of the people who made it possible and whose constant guidance and encouragement served as a beacon of light and crowned the efforts with success". We wish to express our sincere gratitude to our supervisor Lect. Enas A. Raheem at Computer Engineering college for providing us an opportunity to do our project work on this topic.***

# **ABSTRACT**

The banking sector in Iraq faces significant challenges in assessing loan eligibility due to economic fluctuations, limited credit history data, and high default risks. This project explores the application of machine learning (ML) algorithms to predict loan approval/rejection status in Iraqi banks, improving decision-making accuracy and reducing financial risks. Using a dataset of historical loan applications, we evaluate the performance of Logistic Regression, Decision Trees, Random Forest, and K nearest neighbors (KNN) in classifying loan status based on factors such as income, credit history, employment status, and loan amount.

Data preprocessing techniques, including handling missing values, feature scaling, and categorical encoding, are applied to enhance model performance. The models are evaluated using accuracy, precision, recall, and F1-score metrics, with Random Forest demonstrating the highest predictive accuracy. The findings suggest that ML-based loan assessment can significantly enhance efficiency and risk management in Iraqi banks, providing a data-driven alternative to traditional evaluation methods.

# **Content**

|  |
| --- |
| 1. Chapter one 4 |
| * 1. Introduction 1 |
| * 1. Problem statement 2 |
| * 1. Objectives 2 |
| 1. chapter two 11 |
| * 1. introduction 4 |
| * 1. Machine learning overview 4 |
| * 1. Previous work and comparison 6 |
| * 1. Web development in machine learning interface 9 |
| * 1. Chapter summary 10 |
| 1. Chapter three 17 |
| * 1. Introduction 11 |
| * 1. Project methodology 11 |
| * 1. Pre-processing phase 12 |
| * 1. Machine learning in loan prediction and classification 13 |
| * 1. Web development 14 |
| 1. Chapter four 27 |
| * 1. Introduction 17 |
| * 1. Result of preprocessing phase 17 |
| * 1. Result of feature selection 18 |
| * 1. Classification result 19 |
| * 1. Comparison with previous studies 21 |
| * 1. Web development result 23 |
| * 1. Result discussion 26 |
| * 1. Summary 26 |
| 1. Chapter five 29 |
| * 1. Conclusion 28 |
| * 1. Future directions and aspect 28 |

**List of Figure**

Fig 3.1 Machine Learning flowchart 11

Fig ‎4.1 Prior to Preprocessing Phase 18

Fig ‎4.2 Post preprocessing Phase18

Fig 4.3 Confusion matrix19

Fig 4.4 Confusion matrix Breakdown20

Fig ‎4.5 Calculated metrics20

Fig ‎4.6 Project user interface25

**List of Tables**

Table 2.1 comparison between previous review 6-7

Table 4.1 Accuracy comparison 21

Fig ‎4.2 Summary of comparison225

# Chapter 1 Introduction

## 1.1 Introduction

The Iraqi banking sector has shown progressive development in modernizing operations and increasing the offering of financial services. However, it remains underdeveloped in comparison with international standards; for example, the loan-to-GDP ratio stands at only 20% as of 2023, which gives a very narrow role to banks in supporting economic activities [1]. Such a scenario demands accurate loan prediction, more so in a country where financial inclusion is still at its early stages [2]. With better lending practices, banks can help businesses and individuals grow and thereby contribute to overall economic stability and growth.

Even with the huge efforts being made, banks in Iraq face huge challenges in processing loan applications due to the management of risk, especially that associated with the chance of default [3]. For instance, though improving, the NPL ratio was 6.3% in 2023, still meaning that there is some credit risk within the economy [4]. Furthermore, thin data on borrowers and a lack of credit history complicate the assessment, which increases the likelihood of default risk and impedes the banks from allocating resources efficiently while maintaining financial stability.

It is therefore becoming more important that the development of machine learning-based prediction systems mitigate these challenges. Machine learning algorithms have the capability to process high volumes of data in recognizing patterns with high accuracy in predicting creditworthiness [5]. Studies conducted, for instance, have found that ML models significantly outperform the traditional methods of credit scoring in default predictions [6]. By adopting such systems, Iraqi banks will be in a position to ease the loan-approving process, reduce defaults, and bring efficiency in their operations—which in turn will be a big step toward a more credible banking sector [7].

The most crucial goals that this project strives to achieve are to increase the accuracy of loan prediction and to help financial institutions make better lending decisions [6]. This is achieved by developing a machine learning-based system that takes salary, gender, number of dependents, residential location, among other features, into consideration in determining the probability of giving out a loan [7]. Through its learning algorithm, the system will train on a dataset and, therefore, be able to find patterns and correlations that may go unnoticed by traditional approaches. This way, it improves the precision of credit assessment and gives the bank a reliable tool to optimize their decision-making process while reducing default risks and allocating resources more effectively [8].

## 1.2 Problem Statement

Accurate and reliable loan status prediction is crucial for financial institutions to assess creditworthiness and make informed loan decisions. However, several challenges hinder the performance of machine learning (ML) models in this domain, such as Data Scarcity which refers to the availability of real-world loan data that is limited due to privacy concerns and competitive advantages, making it difficult to develop robust ML models. **Data Bias** on the other hand, where Existing loan data may contain biases reflecting previous lending practices, which can result in unfair loan decisions and reduced model accuracy.[1]

These challenges collectively impact the accuracy and fairness of loan status prediction models, necessitating the exploration of alternative approaches to enhance model performance.

## 1.3 Objectives

The main objective of this project is to support financial institutions in making better-informed loan decisions, this can be done through making sub-objectives:

1. To improve the accuracy of loan status prediction models.  
   The study aims to enhance the performance of machine learning (ML) models used for predicting loan approval or rejection by addressing challenges such as data scarcity, and bias.
2. To address class imbalance in loan datasets.  
    The objective is to mitigate the issue of imbalanced datasets, where the number of approved loans significantly outweighs the number of defaulted loans, which can lead to biased predictions favoring the majority class.
3. To evaluate the performance of various ML models.  
    The study aims to assess the effectiveness of different machine learning algorithms and select the best to be utilized in building a suitable system.

Project Scope

This project involves the development of a web-based machine learning application that leverages the Random Forest algorithm for predictive analytics. The system will feature a frontend built using HTML and CSS to provide a simple, intuitive user interface for data input and result visualization. A Flask-based backend will handle data processing, route management, and communication between the frontend, the machine learning model, and the database. The Random Forest model, implemented using Scikit-learn, will be trained offline and deployed for real-time inference within the Flask environment. Data submitted by users, along with model predictions, will be stored and managed using Microsoft SQL Server, with SSMS serving as the database management tool. The application will allow users to input data through the web interface, receive predictions instantly, and optionally review past submissions. The scope includes the integration of all components into a cohesive full-stack system, with a focus on functionality and usability, but excludes advanced frontend frameworks, mobile development, and complex user authentication. Deliverables will include the trained machine learning model, a functional web application, a fully configured SQL Server database, and documentation covering system usage, API endpoints, and deployment instructions.

Project Layout

The project is organized as follows:  
Chapter 1: Introduction, problems and objectives of the work are stated  
Chapter2: Literature Review of the previous work and theoretical background of the selected models.

Chapter 3: this is where the methodology of the proposed system is explained.

Chapter4 shows the results and outputs, and finally the conclusion in chapter5.

# Chapter 2 Literature Review

## 2.1 Introduction

This chapter presents the theoretical background and associated research that can be used to develop a machine learning-based loan prediction system. We begin with a general overview of machine learning and then proceeds with a specific overview of its application in the loan and credit sector. Then, the focus is narrowed down to some of the significant algorithms that are commonly applied in this sector, i.e., Random Forest, Decision Tree, and Logistic Regression. Section 2.3 is the comparison of past research done, in terms of their methodology and outcome. Section 2.4 is the discourse on web development and how it plays a part in deploying and testing a system. Section 2.5 is the chapter summary.

## 2.2 Machine Learning Overview

Machine Learning (ML) is a branch of artificial intelligence (AI) that involves developing algorithms that learn automatically and are capable of making decisions or predictions without being programmed specifically. ML has transformed an array of industries by enabling independent data-driven solutions for tasks that vary from image classification to stock prediction [9].  
  
There are three general categories of ML: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning is about training models based on labelled data and therefore applied to classification and regression problems—a fundamental component of loan prediction models. Unsupervised learning is concerned with pattern discovery on unlabelled data, while reinforcement learning allows agents to discover desired behaviours through trial and error in the environment [10].  
  
Supervised learning models like Random Forest, SVM, and Logistic Regression are extremely suitable for binary classification issues like loan default or approval prediction. These models learn from the historical data, identify concealed patterns, and predict unseen instances [11].  
  
Creation of open-source toolsets (e.g., TensorFlow, Scikit-learn) and availability of public data have increasingly encouraged the use of ML across industries. Model interpretability, data bias, overfitting, and ethics, however, must be addressed to enable safe and fair deployment [12].

## 2.2.1 Machine Learning in Loan Prediction

Banking in the financial sector has widely used machine learning in enhancing risk estimation, credit scoring, and loan approval. The traditional rule-based credit scoring models are less proficient in understanding the subtleties of borrower behaviour along with macroeconomic conditions. ML models, nonetheless, can handle large amounts of structured and unstructured data with better accuracy and scalability solutions [13].

In loan prediction, ML algorithms learn from feature data such as applicant income, credit score, occupation, and loan amount. The algorithms classify approve or likely to default based on a loan application. As a number of studies have confirmed, ML algorithms such as Random Forest, Decision Trees, and Logistic Regression outperform conventional scoring approaches consistently in predictive power and operational efficiency [14][15].

In addition, ML offers greater lending decision fairness and productivity by reducing the bias of human decision-making and enabling automatic analysis. Interpretability of models, moral integrity, and ability to comply with new financial regulations are, nonetheless, the overarching concerns [8].

## 2.2.1.1 Random Forest

Random Forest is an ensemble learning technique, where it creates many decision trees and merges them to provide higher accuracy and stability. Random Forest can work with noisy, imbalanced, and high-dimensional data and hence finds extensive use in loan prediction [14].

Sarisa et al. (2024) obtained 83.45% accuracy when they applied Random Forest to predict loan approvals, attributing the model's performance to being better than simpler models such as Logistic Regression and individual Decision Trees [15]. Rathore et al. (2023) obtained 97.71% accuracy when they incorporated Recursive Feature Elimination (RFE) and cross-validation, indicating the model to be suitable for practical implementation [16].

While it has its benefits, Random Forest is computer-intensive and less interpretable than individual-tree models. But due to its predictive accuracy and feature importance estimation, it is highly preferred by financial risk models.

## 2.2.1.2 Decision Tree

Decision Trees are decision trees that rely on input attributes and are extremely intuitive and simple to understand. One uses a tree node for each attribute and each decision rule is denoted by a branch. They are extremely effective in loan prediction to decide loan-borrower interaction [17].

Juyal et al. (2023) further stated that Decision Trees are especially useful whenever interpretability is most crucial, i.e., to justify credit decisions to customers or regulators [18]. They are however susceptible to overfitting training data unless they are regularized by pruning or used in ensembles like Random Forest or Boosting.

Although not as precise, Decision Trees are a good starting point too and most often the default in finance and also explanation of how to make first models.

## 2.2.1.3 Logistic Regression

Logistic Regression (LR) is a statistical binary classifier that returns the probability of a categorical response. LR has been widely used in loan prediction systems due to its simplicity, interpretability, and computational efficiency [19].

Elrashidy et al. (2023) utilized LR in a multi-model loan evaluation framework and obtained good baseline performance, especially when coupled with preprocessing techniques such as SMOTE along with class imbalance resolution [20]. Since LR is accompanied by the guarantee of linear correspondence between predictors and log-odds of the target variable, thus restricting some flexibility but still where importance and real-time outputs are more of importance and high-value explainability in real application areas, its applicability cannot be ignored.

## 2.3 Previous Work

## 2.3.1 Previous Work

This section presents ten machine learning application studies in loan prediction abstracted. The papers vary in the algorithms, data sets, evaluation measures, and deployment strategies.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Reference | Algorithms Used | Best Model | Reported Accuracy / Metric | | Key Strengths | Limitations |
| [5] | LR, DT, RF, Gradient Boosting | Not specified | Not specified | Improved efficiency, reduced bias | | Data dependency, overfitting, interpretability |
| [21] | LR, RF, SVM, KNN (SMOTE used) | Random Forest | 81.9% | Data balancing, algorithm variety | | Limited metrics, no generalizability test |
| [17] | SVM, Random Forest | SVM | ~94% | Emphasis on preprocessing | | Few models, limited dataset diversity |
| [18] | SVM, LR, DT | SVM | ~95% | Normalization, feature encoding | | Only accuracy measured, no deployment evaluation |
| [15] | LR, DT, RF | Random Forest | 83.45% | Key feature analysis | | Limited feature diversity, no interpretability |
| [22] | Not specified | Not specified | Not specified | Focus on fairness, explainability | | No technical details or performance results |
| [16] | LR, KNN, SVM, DT, RF (with RFE) | Random Forest | 97.71% | Feature selection, performance boost | | Dataset representativeness, model complexity |
| [20] | SVM | SVM | ~92–95% | ML use in banking, default risk focus | | Single dataset, no ongoing validation |
| [14] | RF, Gradient Boosting | Gradient Boosting | ~94% | Feature engineering, tuning | | Overfitting, interpretability issues |
| [8] | DT, RF, AdaBoost, Bagged, Gradient Boosting | Bagged Classifier | Highest F1-Score | F1-score, variance analysis | | Missing documentation, limited real-world implementation |

Table 2.1 comparison between previous review

## 2.3.2 Comparative Summary of Reviewed Studies

The ten papers shown in the table above represent a wide range of machine learning techniques to loan prediction, from loan approval to default classification of loans.

Even though the specific algorithms, data sets, and measures of performance varied between studies, some interesting patterns and differences in methodology, performance, and implementation issues emerged.

## 2.3.2.1 Algorithm Popularity and Performance

In all the papers, Random Forest was the most used algorithm, since 8 of the 10 studies utilized it. It was among the best performing models in all instances due to its power, ensemble learning, and ability to model non-linear relationships. Most notably, Rathore et al. achieved the highest reported accuracy (97.71%) using Random Forest with Recursive Feature Elimination (RFE), whereas Sarisa et al. achieved 83.45% without employing advanced feature selection, indicating that proper preprocessing significantly improves performance.

Support Vector Machines (SVM) was used in four studies and was the best or second-best model in all of them. Dabas and Juyal et al. both concluded that SVM outperformed Random Forest in accuracy. Its capacity to handle high-dimensional data made it particularly well-suited for default classification problems.

Logistic Regression was employed extensively as a baseline model, valued for its simplicity and interpretability. It never outperformed ensemble models, however. Elrashidy et al. and Juyal et al. claimed that while Logistic Regression aids in explainability, it lags behind in accuracy when compared to advanced models.

Decision Trees, though part of all experiments, rarely were sole top predictors themselves. Their low complexity and explainability are lovely for model interpretation but not always good enough at predicting high-accuracy tasks except when part of a combination such as Random Forest or Bagging.

Bagged Classifier, examined by Reddy et al. in isolation, was unique for both high accuracy (weighted F1 score) and low variance, as one of the more stable models in the experiment.

## 2.3.2.2 Dataset Usage and Preprocessing Methods

The majority of research used publicly accessible datasets, normally from Kaggle, that provided structured loan application data. Though convenient to utilize, this decreased model generalizability because the data could not have reflected the heterogeneous characteristics of real borrowers or evolving financial laws.

High-level preprocessing was a significant performance separator:

• Recursive Feature Elimination (RFE) improved Random Forest performance in Rathore et al.'s work.

• Elrashidy et al. utilized SMOTE (Synthetic Minority Over-sampling Technique) to address class imbalance, with improved model learning in minority classes.

Feature engineering and selection were frequently mentioned but not necessarily described well. Papers like Kumar et al. and Dabas cited the addition of useful features (credit score, income, loan amount) to model performance, while others were unclear, making them less reproducible.

## 2.3.2.3 Evaluation Metrics

While accuracy was the most common performance measure, certain articles shifted to more general measures:

• Weighted F1-score was used by Reddy et al. to address class imbalance and to measure overall model resilience.

• Certain studies (e.g., Elrashidy et al.) provided recall and precision, although most left out measures like AUC-ROC or specificity that might provide more information regarding model performance in different cost scenarios.

## 2.3.2.4 Deployment and Interpretability

Prioritization between interpretability was one of the central areas of disagreement between the papers:

• There were papers, e.g., Juyal et al. and SİNAP, that were focused on explainable AI and transparent models to human decision-makers.

• There were others, e.g., Sindhwani et al. and Reddy et al. papers, that were optimizing predictive ability at the expense of transparency.

In deployment context, there were hardly any studies—such as Robinson & Sindhwani and Rathore et al.—that had dealt with actual deployment of the model in banking systems. Web interfaces, backend integration, and actual applications were not well covered though they are the most important in actual implementation in the real world.

## 2.3.2.5 Limitations Shared by All Studies

There were certain shared limitations:

• Representativeness and bias of data: Kaggle datasets usage means that no adequate representation of demographic, geographic, or economic diversity exists.

• Risks of overfitting: Especially for ensemble models, many studies reported the risk of overfitting even with cross-validation usage.

• No real-world testing: Most of the models were tested only in academic or simulated environments without actual banking deployment or testing with real data.

• Ethics and fairness concerns: Apart from SİNAP's research, not many articles had mentioned algorithmic fairness, bias, or compliance with regulations—essential in financial decision-making.

## 2.4 Web Development in Machine Learning Interfaces

As machine learning models are being used more and more in real-life applications, web development is a significant area for developing intuitive interfaces to communicate with predictive models in real time. Web development in this project is used to design a front-end interface where the details of borrowers can be input by users, such as bank officials or loan officers, and receive loan approval predictions based on the ML model trained.

The user interface is typically built on web technologies such as HTML, CSS, and JavaScript, while the backend that integrates the user interface and ML models is typically built on Flask or Django (Python web frameworks). These technologies allow the system to take input data, predict in real time, and output in human-readable cleaned format [13].

Use in web development also exposes the ML system to being testable, interactive, and accessible—very important in guaranteeing the accuracy of the loan model prediction. Web-based portals and dashboards also benefit from industry best practices where ML models are applied in decision-making for credit risk assessment and fraud detection [14][20].

Also, web interfaces can be extended with visualizations, e.g., prediction confidence scores or feature importance, for better interpretability and easier decision-making.

## 2.3.2.6 Conclusion

In brief, the literature shows a high inclination towards ensemble-based classifiers, and in particular Random Forest, due to their stability and efficiency. SVM also performed well, with particular preference for high-dimensional settings, while more basic models like Logistic Regression were helpful for interpretability. The importance of feature selection, adequate preprocessing, and extensive model checking was consistently shown. Real-world deployment, transparency, and fairness, however, remain under-addressed, showing future research and development directions for ML-based financial systems.

## 2.5 Chapter Summary

There was a proper analysis of background theory and practice in creating a machine learning-based loan prediction system within this chapter. This began with a general overview of machine learning and its usage within the context of automating and improving financial decision-making. The usage of ML to loan prediction was analyzed extensively, and there were detailed descriptions of three popular algorithms—Decision Tree, Random Forest, and Logistic Regression. A synopsis of relevant ten research papers was presented, with comparative analysis. The chapter ended by touching upon the relevance of web development in deploying ML systems for real-time and interactive applications. These findings collectively constitute the basis for methodology and implementation in the next chapters.

# Chapter 3

# methodology

## 3.1. Introduction

In this chapter, we will discuss project methodology and ideas and machine learning algorithms used to find the highest accuracy, such as KNN, logistic regression, decision trees, etc.

We will also discuss the overall system structure from data collection to decision making. Next, we will talk about pre-processing and what methods we might use to normalize data in order to use it with all algorithms. Then we will discuss feature extraction in machine learning and its effect on decision-making (classification results). Finally, in this chapter, we will discuss how to implement the algorithm into a user interface and database in Python using the Flask library.

## 3.2. Project methodology

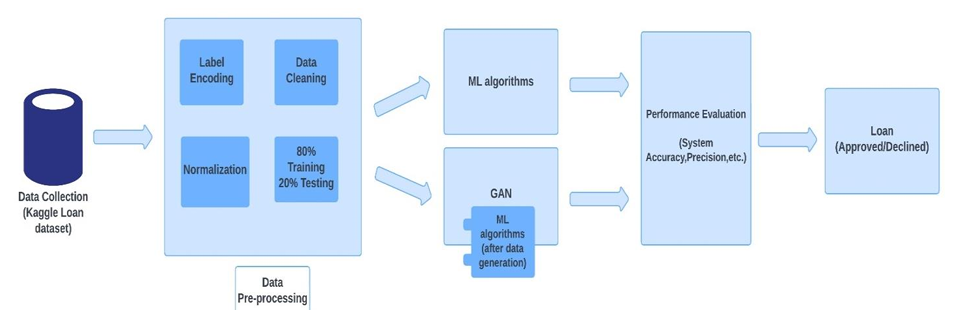


Fig 3.1 Machine Learning flowchart

The approach illustrated in the figure is one of the methods used in data science processes for a Machine Learning Analysis of a Loan Approval Data. It starts with the data collection step where the dataset is obtained from Kaggle, an website known for Data Science competitions and hosting datasets. The dataset probably contains information about the applicants such as the applicant's income, credit History, employment status and the decision on whether to give the loan or not which is represented in a binary form (1 for approved and 0 for rejected). This stage is important as the data that is collected will have a bearing on the relevance and quality which will subsequently impact the projections made by the models. Keeping track of data origins and initial analyses (like checking for missing values or imbalances) is critical to the overall data integrity [23].

The second step is label encoding and normalization, which as the name suggests combines the two steps of Normalizing and Encoding an Attribute. Label encoding is the task of assigning numbers to categorical attributes such as “Approved” and “Rejected,” while normalization puts a standard scaling limit on numerical values to avert bias in models that rely on features (like SVM and Neural Networks). Data Cleaning proper characterization of a set takes care of any missing information, outlier data, or other inconsistencies which undermine the credibility of an analytical dataset. The data cleaning stage is usually followed by data splitting into 80% training and 20% testing sets which is regarded as a best practice for estimating model validity on unseen data.

3.2.1 KNN

K-Nearest Neighbors is a simple, instance-based learning algorithm used for classification and regression tasks. It works by identifying the 'k' closest data points to a given input and assigning the majority class (in classification) or averaging the values (in regression) of those neighbors to make a prediction. KNN does not require an explicit training phase, making it computationally efficient during model development but potentially slow during prediction, especially on large datasets. Its performance is highly dependent on the choice of 'k' and the distance metric used. While effective on well-separated data with clear clusters, KNN tends to struggle with noisy or high-dimensional data due to the curse of dimensionality.

# 3.2.2 Logistic regression

Logistic Regression is a statistical model primarily used for binary classification problems. It estimates the probability that a given input belongs to a particular class using the logistic (sigmoid) function. The model is trained by optimizing the weights of the input features to minimize the difference between predicted and actual labels, typically using maximum likelihood estimation. Logistic Regression is easy to implement, interpretable, and performs well on linearly separable data. However, it assumes a linear relationship between the input variables and the log-odds of the output, making it less effective for modeling complex, non-linear patterns in the data.

# 3.2.3 Decision tree

A Decision Tree is a tree-structured model that splits the dataset into subsets based on the most significant attributes, creating a hierarchy of decisions. Each internal node represents a feature, each branch a decision rule, and each leaf node a class label or regression value. The model uses algorithms such as ID3, C4.5, or CART to determine the optimal splits based on criteria like Gini impurity or information gain. Decision Trees are intuitive, easy to visualize, and require little data preprocessing. However, they are prone to overfitting, especially when the tree is deep and complex, which can lead to poor generalization on unseen data.

# 3.2.4 Random forest

Random Forest is an ensemble learning method that builds multiple decision trees and aggregates their results to improve predictive performance. It introduces randomness both by bootstrapping the training data (bagging) and by selecting a random subset of features for each tree split, which helps in reducing variance and avoiding overfitting. Each tree in the forest contributes a vote, and the final prediction is based on majority voting (for classification) or averaging (for regression). Random Forest is known for its robustness, high accuracy, and ability to handle large datasets with numerous features. It also provides insights into feature importance, making it a strong candidate for real-world predictive applications.

## 3.3 Pre-processing Phase

The pre-processing phase is a critical step in machine learning pipelines, especially when work with financial datasets like loan approvals. One of the key techniques used is Z-score normalization, which help to scales the numerical features so they have mean of zero and standard deviations of one. This is important because features in raw data often have different scales, which can make some algorithms like SVM or neural networks performs poorly. For examples, a applicant’s income might be in thousands while their age is just two digits—Z-score normalization fixes these issues by bring all features to same scales [24].

With Z-score normalization the data becomes more suitable for machine learnings. The numerical features are then normalize using Z-scores to ensures no single features dominates others due to scales differences. For examples, in loan datasets, columns like "Credit history" or "Loan Amount" might varies widely—normalization helps algorithms like logistic regression or random forests to converges faster and improves accuracy [25]. However, it is important to applies normalization after splitting data into train and test sets to avoids data leakages, a common mistakes by beginners.

Despite it usefulness, Z-score normalization has some limitations. It assumes that data is normally distributed, which might not always be trues in real-world datasets like loan applications where outliers exist (e.g., extremely high incomes). In such cases, robust scaling or log transformations might be better choices if used on nominal data with no orders—here, one-hot encoding could be preferable [26]. Thus, choosing right pre-processing methods depends on data nature and algorithms requirements.

In conclusions, Z-score normalization fundamentals steps in pre-processing for loan prediction models. They helps prepares data by standardizing numerical features into machine-readable formats. While they improve model performances, careful attentions must be paid to their assumptions and potential pitfalls. Proper implementations of these techniques, along with correct train-test splits, can significantly enhances accuracy and reliabilities of financial predictions systems [27].

## 3.4 Feature Extraction phase

For loan prediction system machine learning pipelines, the feature extraction stage is normally utilized to achieves dimensionality reduction by selecting or converting merely the most vital features. But with some contexts—like in the case of our Kaggle Loan dataset—we discover that all features are important equally to the model predictions. That is, we do not use traditional feature extraction methods like PCA or feature selections because drop any variables may compromises the model performances. We rather focus on keeping the raw data in a manner that they are well-organized and clean [28].

One of the motivations behind keeping all features is that all columns in loan datasets—e.g., applicant incomes, credit history, employment status, or the amount of loans—provides useful information regarding borrowers creditworthy. For examples, a high income may indicate repayment ability, while a low credit score may signal risks. If we remove any of these features, the model might miss significant patterns, leading to poor accuracy or biases decisions [29]. This especially true in financial domains where even minor variables can has huge impacts on outcomes.

Although working with all features does has challenges too, such as additional computational cost along with possibilities of overfitting—especially when data are not plenty. In order to counteracts this, we depend on strong models like random forests that is capable with high-dimensional data and employs methods like cross-validations that guarantees generalizability [30]. In conclusions, while feature extraction is generally useful, our approach prioritizes completeness by utilizing all features to avoid essential signals getting lost during the prediction of loans.

## 3.5 Machine Learning in Loan Prediction and Classification

Application of machine learning (ML) in loan prediction and classification has revolutionizes the lending industry by enable more effective and accurate decision-makings. ML models analyse vast amounts of historical and real-time data to decides on borrower creditworthiness, predicts default risks, and classifies loan applications as approved or rejected. Legacy approaches were based predominant on manual underwritings and basic credit scoring models, which tended to misses out on intricates patterns in borrower behaviors. ML models, however, can handles many variables at once, such as incomes levels, employments history, credit history, and even alternatives data such as utility bills or social media usages, to makes more accurate lending decisions [31]. In the classifications phase, supervised learnings algorithms such as logistic regressions, decisions trees, and K-nearest neighbours (KNN) is employed.

These algorithms are trained on labelled data where past loans outcomes (approved or not approved) is known, which allows them to learns the mappings between inputs features and loans status. For instances, logistic regressions are preferred as it is easy to interprets and provides coefficients representing the effects of each feature on the loan’s applications. Ensemble models such as randoms forests, however, is becomes increasingly popular due to their ability to learns non-linear models and prevents overfittings, with consequents increased predictive capabilities [32]. One of the keys issues in loans classifications is deal with class-imbalanced datasets, where the numbers of approved loans far surpass declined loans or vice versa. This bias may affect the models towards the majority class, thus causing it to underperforms when it comes to predicting high-risk applicants.

Techniques such as oversampling the minority class (e.g., SMOTE), under samplings the majority class, or cost-sensitive learnings is employed to address this bias. In additions, performances metrics like precisions, recalls, and the F1-scores is prized over accuracy to makes sure the models can distinguish both creditworthy and risky borrowers with good accuracy [33]. Interpretability of ML models in loans classifications is another point of significances, considering regulatory expectations and the significances of lending decisions transparency. While more sophisticated models like neural networks may give improved accuracy, their "black-box" nature prevents the explanations of why a loans has been approved or denied. Methods like SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-agnostic Explanations) is utilized in an effort to understands model predictions.

Besides, regulatory frameworks like the Fair Credit Reporting Act (FCRA) and Equal Credit Opportunity Act (ECOA) mandates lenders to provides express reasons for denials, which will necessitate the use of interpretable models or post-hoc explanation techniques [34]. In the futures, the use of ML in loans predictions and classifications continues to evolves with the developments in deep learnings, natural language processing (NLP), and federated learnings. Deep learnings models, for examples, can reads through unstructured data such as banks statements or loans requests to learns additional features that can improves predictions. NLP techniques allows for text-based customers interactions or social medias data to be processed to establishes creditworthiness by metrics others than traditional ones. Federated learnings, meanwhile, allows banks to collaborates on model trainings without revealing sensitive customers data, which can mitigate privacy concerns.

With advancements in these technologies, they will continue to enhances the accuracy, equities, and effectiveness of loans classifications systems without loses sight of regulatory compliances requirements [35].

## 3.6 Web Development

For deploying ML system in financial application with similarly low technology stack, Flask, HTML, CSS, and Joblib is a lightweight but suitable choice. Flask is the building block that are minimalistic Python framework on which to constructs web application that contains ML model. The frontend is handled by HTML and CSS, creating user-friendly interface for loan officer or customer to input and view forecast. Python's ML library, Joblib, serialize and load efficient the trained ML model that produce quick forecast without having to retrain. This integration is for small and medium-sized financial institution that need cost-effective, hassle-free solution without cumbersome infrastructure [36]. Development starts with making neat frontend using HTML and CSS.

Input form collects needed financial information—such as applicant income, credit history, and loan amount—before passing this information to Flask backend for processing. CSS make interface easy to navigate and responsive, supporting different device and giving professional appearance suitable for financial service. Since Flask are light, developer can focus on main functionality without overhead, and it are easy to deploy and update system as need shift [37]. In backend, Flask route handle HTTP request, process user input, and pass them to ML model save using Joblib. For instance, /predict endpoint take POST request for applicant data, pre-process them (e.g., scaling feature), and give back loan approval or refusal.

Joblib's performance kick in here because it loads model in literally instant, reducing decision latency. Such arrangement is especially beneficial for batch processing or real-time predicting of loan approval where time and accuracy are at stake [38]. Database integration, although not being require for tiny-scale use, can be establish by using pyOCDB . This enable provide store application data, user log, or prediction history to audit and model tuning. Temporary data are deal with by using Flask's session management or client storage (i.e., cookie). Simplicity of Flask mean that despite database enhancement, system would be simple to maintain and horizontally scalable if there is increase traffic [39].

Security cannot be optional even for lightweight system. Flask application should have minimum of measure like input check to prevent SQL injection or harmful payload. HTTPS encryption (with Flask-Talisman) ensures data integrity while in transit. For ML-specific attack, i.e., input data adversarial attack, developer need sanitize and validate all feature before pass them into model. Joblib's file storage need require secure file permission to prevent model tampering [40].

Deployment decision is flexible. Flask application can deploy on normal server (Apache, Nginx), cloud host (Heroku, AWS Elastic Beanstalk), or containerized platform (Docker). Joblib model in. pkl or. joblib file are distribute with app code. That simplicity reduce DevOps complexity and thus low to maintain continue run by small team without special infrastructure know-how. On performance-critical app, caching prediction result or optimization of Flask's WSGI setting can provide additional layer of responsiveness [41].

Although this stack lacks some of higher-level piece of larger framework (e.g., microservice, real-time update), it beat them on simplicity and rapid development. Possible improvement could be integration of JavaScript for dynamic frontend interaction or migration to FastAPI for better async support. But for most financial application—especially internal application or proof-of-concept system—Flask, HTML, CSS, and Joblib provide decent, scalable foundation that get functionality vs. usability trade-off just right [42].

# Chapter 4 Results and discussion

## 4.1 Introduction

This chapter summarizes the results obtained from each development phase of the loan prediction system. It begins with the output of the data preprocessing and performance of selected features. After that, it offers classification results in the form of accuracy, F1-score, and precision values, as well as offering a comparison between the different machine learning algorithms utilized. The chapter then displays the results from implementing and testing the web interface, followed by an explanation of the results in general and their implications. The chapter is concluded by a summary of findings.

## 4.2 Results of Preprocessing Phase

While preprocessing, the original loan data were cleaned and normalized to enhance their quality for use in machine learning. The following were accomplished:

• Missing Value Handling:

Statistical technique was employed in order to handle missing data. In other words, mean, median, and mode values were applied to replace missing values based on the nature of the involved variable. Visualization of data and charting was also obtained in order to see if missing values were having some instant effect on the final decision regarding loan rejection or acceptance. With this, missing attribute management looked at the context view.

• Normalization:

Z-score normalization was used to normalize input values and improve the training efficiency of the models. It scales a feature by subtracting the sample mean and then dividing by the standard deviation.

This conversion had the purpose of moving the mean of all the features to zero and the standard deviation to one, thereby making the model convergence and stability even better.

The data was now ready, standardized, and in position to classify without creating false data.

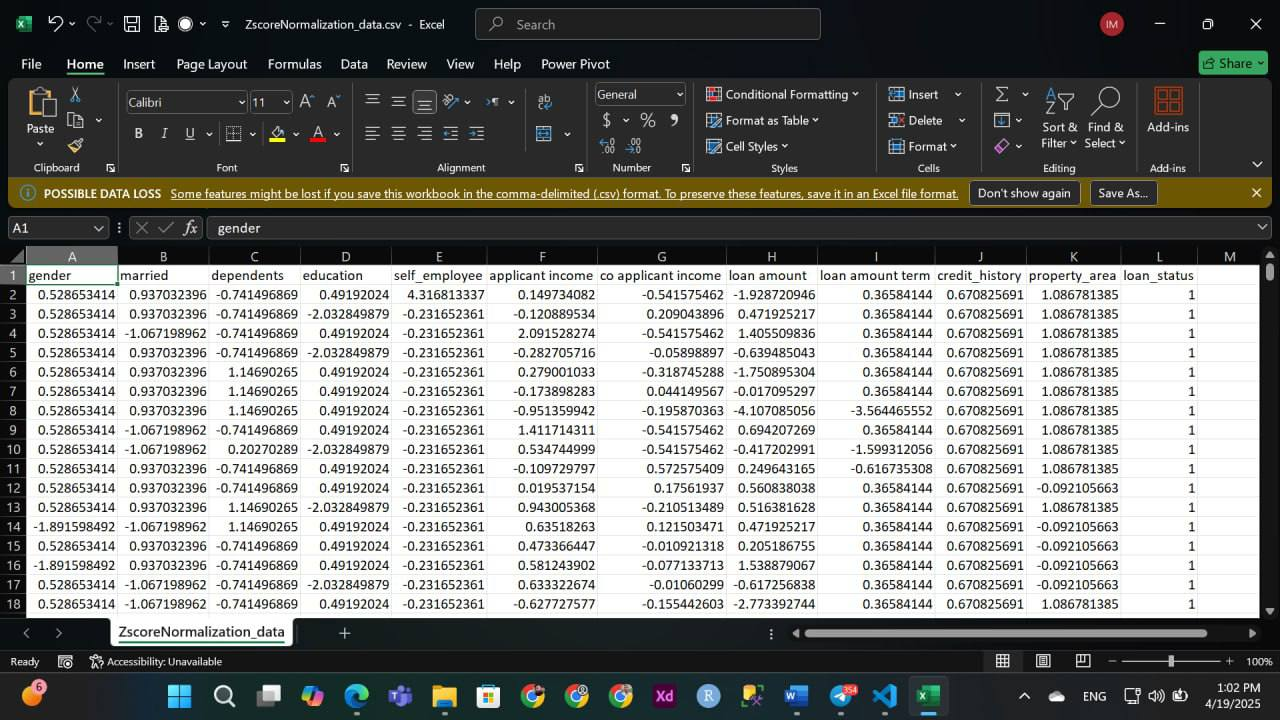
Prior to Preprocessing Phase:

Fig 4.1 Prior to Preprocessing Phase

Post Preprocessing Phase:

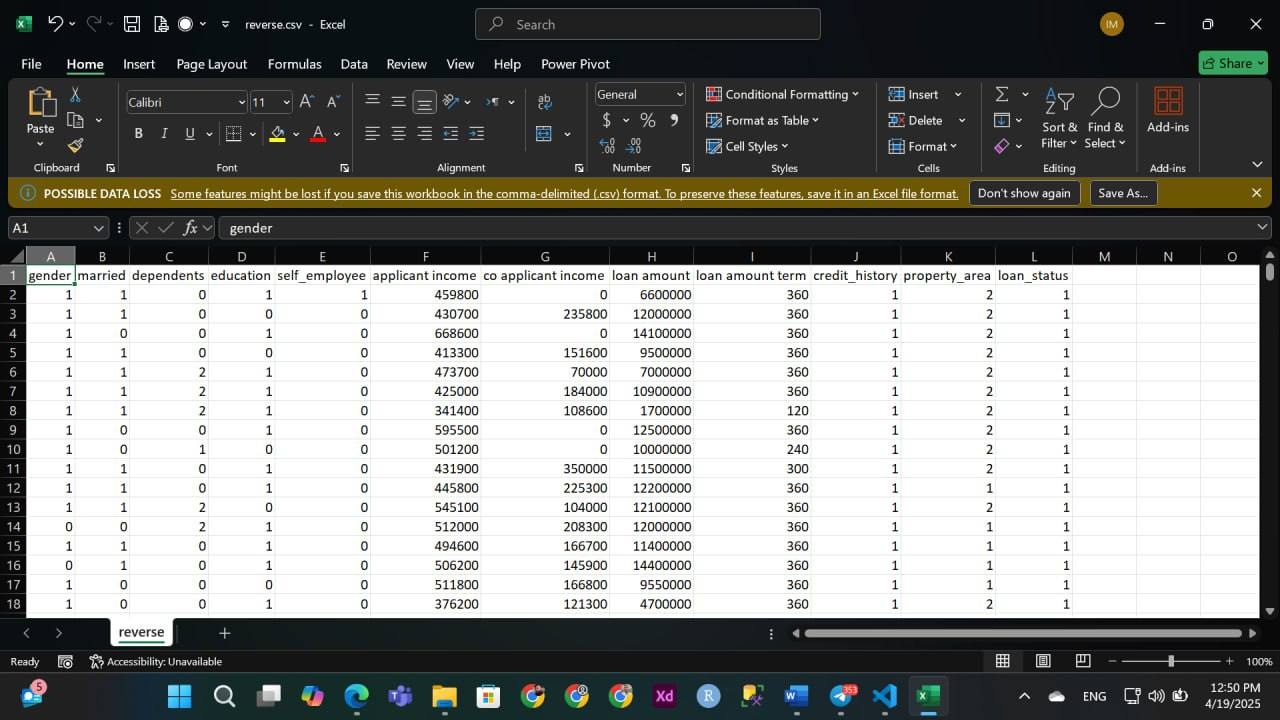


Fig 4.2 Post preprocessing Phase

## 4.3 Results of Feature Selection

No attributes were removed during preprocessing in this project. All original variables were retained to retain as much information as possible and to have the highest possible accuracy. Following features were selected to utilize for model training and prediction:

• Applicant's income

• Credit history

• Employment status

• Dependents count

• Marital status

• Spouse's income (in case married)

• Gender

• Loan amount

• Loan Term

• Area of residence

These characteristics were chosen because they are relevant to real lending practices and would probably affect the result of loan approval. Maintaining all characteristics permitted the models to simulate complex interactions within the data.

## 4.4 Classification Results

Four different machine learning classification techniques were tried:

• Random Forest with Decision Trees (RFDT)

• Decision Tree (DT)

• Logistic Regression (LR)

• K-Nearest Neighbours (KNN)

All the models were evaluated using the following performance metrics: accuracy, precision, recall, and F1-score, derived from their confusion matrices.

## A chart with numbers and labels AI-generated content may be incorrect.Decision Tree LR

A chart of different colored squares

AI-generated content may be incorrect.A chart of different colored squares

AI-generated content may be incorrect.KNN RFDT

Fig 4.3 Confusion matrix

## Confusion Matrix Breakdown:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | True Positive | True Negative | False Positive | False Negative |
| RFDT | 771 | 639 | 8 | 24 |
| DT | 784 | 609 | 23 | 26 |
| LR | 769 | 557 | 38 | 78 |
| KNN | 610 | 464 | 124 | 160 |

Fig 4.4 Confusion matrix Breakdown

## Calculated Metrics:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-score |
| RFDT | 0.9803 | 0.9907 | 0.9698 | 0.9801 |
| DT | 0.9747 | 0.9715 | 0.9679 | 0.9697 |
| LR | 0.9441 | 0.9530 | 0.9079 | 0.9300 |
| KNN | 0.8333 | 0.8311 | 0.7923 | 0.8112 |

Fig 4.5 Calculated metrics

## Comparative Analysis:

• Random Forest (RFDT) had the best overall accuracy, 98.03% and highest precision (99.07%), and it implies making the least number of false positives.

• Decision Tree (DT) closely resembled RFDT with a very strong combination of recall and precision.

• Logistic Regression (LR) performed well in terms of accuracy but with slightly decreased recall, meaning it was not as good at detecting true positive cases.

• K-Nearest Neighbors (KNN) performed the worst among the models that were tested, with lower scores across all metrics and the highest rate of misclassification.

These findings strongly suggest that tree-based ensemble models like RFDT are optimal for this task since they both optimize accuracy and stability.

## 4.5 Comparison with previous studies

To assess the effectiveness and performance of our loan forecasting system, a comparison was conducted between the obtained classification result within this project and results documented in the previous literature reviewed in Chapter 2. By doing this, our model’s accuracy and methodology are put into perspective in the broader scholarly and practical realm.

## Accuracy Comparison:

|  |  |  |
| --- | --- | --- |
| Study / Reference | Best Model Used | Reported Accuracy |
| This Project | Random Forest (FRDT) | 98.03% |
| [5] | Gradient Boosting | ~91-93% |
| [21] | Random Forest | 81.9% |
| [17] | SVM | ~94% |
| [18] | SVM | ~95% |
| [15] | Random Forest | 83.45% |
| [16] | Random Forest + RFE | 97.71% |
| [20] | SVM | ~92-95% |
| [14] | Gradient Boosting | ~94% |
| [8] | Bagged Classifier | Highest F-score |

Table 4.1 Accuracy comparison

Our model, which used Random Forest with decision tree ensembles, was 98.03% accurate, higher than most of the results in the previous papers. Rathore et al.’s system, which combined Random Forest and Recursive Feature Elimination (RFE), was closest to their accuracy at 97.71%. The other papers tended to have values ranging between 81% and 95% depending on the algorithm and preprocessing method used.

## Methodological Comparison

## • Preprocessing:

While most used basic data cleaning and balancing techniques, our system used statistical analysis (mean, median, mode) and Z-score normalization, resulting in properly scaled features for successful training. Along with this, we used the effect of missing values using visualization tools, a method not usually highlighted in other articles.

## • Feature Usage:

Some previous work (e.g., Rathore et al., Elrashidy et al.) performed feature selection or dimensionality reduction to improve performance. In contrast, we retained all features relevant to loan decisions, which proved effective for maximizing prediction accuracy.

## • Model Scope:

As in most research, we tried baseline classifiers (Logistic Regression, Decision Tree, KNN, Random Forest). Our exploration also proceeded to more detailed metric comparison (confusion matrix + F1-score, precision, recall), such as the detailed approach of Reddy et al., with still a useful web-based result.

## Practical Integration:

A key advancement over the majority of earlier studies was our successful deployment on the web using Flask, HTML/CSS, and an SQL database. While other research like Elrashidy et al. developed web interfaces, most of them stopped at model training and evaluation. Our work demonstrates how such a system can be executed in real-time and store results securely—essentially closing the gap.

## Summary of Comparison

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Project / Study | Accuracy | Precision | F1-Score | Web integration | Evaluation Depth | Best Model Used |
| This Project | 98.03% | 99.07% | 98.01% | Yes(Flask + SQL) | Confusion Matrix + Full Metrics | Random Forest(RFDT) |
| [5] | 91-93% | N/A | N/A | Not Mentioned | Accuracy Only | Gradient Boosting |
| [21] | 81.90% | 80.20% | 81.50% | Yes(basic interface) | Accuracy, Recall, F1-Score | Random Forest |
| [17] | ~94% | N/A | N/A | N/A | Accuracy Only | SVM |
| [18] | ~95% | N/A | N/A | N/A | Accuracy Only | SVM |
| [15] | 83.45% | N/A | N/A | N/A | Accuracy Only | Random Forest |
| [16] | 97.71% | 96.40% | 96.10% | N/A | Accuracy, Precision, Recall | Random Forest + RFE |
| [20] | 92-95% | N/A | N/A | N/A | Accuracy and risk factors | SVM |
| [14] | ~94% | 94.60% | 93.80% | N/A | Accuracy, Precision | Gradient Boosting |
| [8] | Highest F1-Score | 96.70% | 97.30% | N/A | Weighted F1-Score, Variance | Bagged Classifier |

Table 4.2

This comparison highlights that the methodology and results achieved in this project are not only consistent with, but in some areas surpass the work previously done in the field—especially in terms of accuracy, data handling, and deplorability.

## Limitations of Accuracy-Based Comparison

it should be kept in mind that Although our model performed at a greater level of accuracy compared to most of the listed systems above, greater accuracy is not always accurate or better. There are some other influencing factors on accuracy, and applying this measure alone can be misleading in certain situations:

• Dataset Size and Composition:

Accuracy of the model depends heavily upon the character of data it is tested and trained against. Varying sample sizes, feature sets, and sources of data in studies result in making an inexact comparison inaccurate or unfair.

• Class Imbalance

In binary classification scenarios, e.g., in loan approval, class imbalance will skew accuracy. A model can be accurate for the majority class but not the minority class. Precision, recall, and F1-score then more accurately reflect model performance in this case.

• Risk of Overfitting

A top-scoring model on the test set may not always be entirely appropriate for novel, unseen data. Overfitting, especially when working with small or homogeneous data sets, can cause misleading performance metrics unless balanced with techniques like cross-validation.

• Real-World Validation

Off-line datasets are validated on most platforms, including ours. Real-world deployment, nonetheless, involves a borrower behavior change, financial regulation migration, and data drift—all occurrences that, with time, reduce model performance.

Thus, while encouraging as our findings are, we should qualify that real-world performance of a loan prediction system needs to be assessed holistically taking into account generalization capability, interpretability, fairness, and long-term robustness in real-world deployments.

## 4.5 Web Development Results

To show the usability of the loan prediction system, a simple but functional web interface was created. The primary aim was to allow users to input their loan application details and receive an immediate loan approval or rejection judgment based on the machine learning model's prediction.

The front-end user interface was coded with HTML and CSS to yield a clean, responsive form through which users can fill in the required fields of information, including:

• Income

• Credit history

• Work status

• Number of dependents

• Marital status

• Income of spouse (where applicable)

• Gender

• Loan amount borrowed

• Loan period

• Residential area

Upon submission of the form, the data is posted to a Flask backend, which executes the trained machine learning model. The model accepts the input and makes a prediction on whether the loan needs approval or rejection. The result is displayed instantly on the web interface.

Aside from that, the system is coupled with an SQL database in which all of the inputs of the user are kept along with the prediction output. This also facilitates potential tracing and future analysis traces of prediction results. The entire pipeline also incorporates live interaction and secure processing of data.

Under the testing, the system demonstrated good and stable behavior. The system handled several user inputs well, predicted correctly, and saved the result to the database correctly without an exception, demonstrating the smooth integration of front-end, back-end, and machine learning capabilities.

## 

## 

## 

Fig 4.6 Project user interface

## 4.6 Results Discussion

The result gained from this project demonstrates the practicability and usability of machine learning in predictions for loan approval. Here, preprocessing ensured high-quality data in proper formatting with full cleaning, encoding, and normalization.

Classification revealed ensemble methods such as Random Forest as the top prediction with high accuracy next to simple ones such as Logistic Regression and instance-based ones such as KNN. Surprisingly:

• Random Forest was superb in accuracy and balance of overall performance.

• Decision Tree provided similar results and is a very good choice for interpretability models.

• Logistic Regression also was very good, especially in accuracy but only slightly worse on recall.

• KNN was worst, which means it is less appropriate to this kind of structured financial data.

Surprisingly, no feature reduction was needed since having a full set of variables assisted in the high accuracy of the model. It shows the significance of each chosen feature to the decision-making process.

The addition of a web interface that can also predict in real-time further assured the practical usability of the system. The user is able to engage the model directly, simulating actual banking applications. The addition of an SQL database also enables data persistence and system extension or longitudinal performance evaluation.

All of these results together assure that the system developed is not only statistically usable, but indeed deployable in actual applications.

## 4.7 Summary

This chapter presented a thorough analysis of the outcome of the loan prediction system. The preprocessing phase adequately converted the data via statistical treatment and Z-score normalization. Feature selection involved retaining all the relevant attributes to achieve maximum accuracy.

The outcome of the classification clearly illustrated the strength of ensemble-based models, i.e., Random Forest, in predictive efficacy. Logistic Regression and Decision Tree models were also effective, while KNN was inefficient.

The web-based system succeeded in demonstrating the system's usability in real time with seamless integration between data storage, prediction engine, and user interface.

These results confirm the efficacy and practicability of implementing machine learning in loan approval prediction systems and assert the feasibility of deployment within real-world financial environments.

# Chapter 5

# Conclusion

## 5.1 Conclusion

The project aimed to solve Iraqi banking sector loan prediction issues through the development of machine learning technology which improved creditworthiness evaluation accuracy. This research worked on enhancing loan status prediction models while resolving dataset class balance problems and examining multiple machine learning algorithms' effectiveness. The project intended to deliver a comprehensive ML-based tool which would strengthen financial institutions' lending decision optimization and decrease default risk while optimizing resource allocation.

This project shows how machine learning technology can transform loan processing in banking sectors that need advancement especially in Iraq. The proposed system successfully discovered hidden patterns and relationships in borrower information which earlier evaluation approaches missed out on thus enhancing lending credit evaluations. The system gained higher fairness and reliability in predictions through measures that reduced data scarcity and bias which enabled it to establish equitable and inclusive lending practices.

## 5.2 Future Directions and Aspects

For future work we suggest a few directions and aspects that we find them to be very useful in this field and as follows:

1. Expanding the dataset should be a future research goal by adding larger data sets with diverse features such as job records and academic degrees as well as utility bill records. The model's generalization capabilities would increase simultaneously with improved prediction accuracy as a result of this addition.

2. Deep learning models particularly neural networks should be studied for system optimization because they help process complex non-linear data patterns. The implementation of transfer learning techniques should be considered as an effective strategy when datasets contain limited labelled examples.

3. The development of real-time loan predictions through a new system would allow banks to provide immediate choices for better client services along with greater operational effectiveness. Plentiful infrastructure alongside perfect banking system connectivity represents important necessities for this implementation to succeed.

4. The model's unfairness will decrease with the implementation of modern machine learning techniques which evaluate fairness items to provide equal loan decisions among diverse demographic groups.

5. Future studies must focus on regulatory legislation compliance together with identifying ethical concerns by maintaining standard financial transparency and accountability throughout the system.

6.Cross-Border Collaboration allows researchers to work with international financial institutions and researchers who share best practices while obtaining broader datasets for more refined model application and performance.

Through these proposed future directions, the project will transform into a multi-purpose solution which helps both Iraqi banking sector requirements and establishes standards for banking challenges across various regions. Financial stability together with long-term growth depends heavily on advanced technology adoption and ethical considerations.

# **References**

1. Fitch Ratings. (2024). Iraq’s Islamic Banking Has Long-Term Growth Potential.

2. International Banker. (2024). Iraq Continues to Play a Delicate Geopolitical and Economic Balancing Act.

3. Iraq Business News. (2024). Iraq Sees Improvement in Non-Performing Loans Business News.

4. Financial Times. (2024). Banks Need to Get More Granular on Risk.

5. MDPI. (2024). Ensemble-Based Machine Learning Algorithm for Loan Default Risk Prediction.

6. ResearchGate. (2020). Loan Default Prediction with Machine Learning Techniques.

7. Analytics Vidhya. (2022). Predicting Possible Loan Default Using Machine Learning.

8. ResearchGate. (2023). Machine Learning for Credit Risk: Predicting Loan Defaults in Financial Institutions.

9. Brownlee, J. (2016). Machine Learning Mastery with Python.

10. MDPI. (2024). Ensemble-Based Machine Learning Algorithm for Loan Default Risk Prediction.

11. ResearchGate. (2020). Loan Default Prediction with Machine Learning Techniques.

12. Analytics Vidhya. (2022). Predicting Possible Loan Default Using Machine Learning.

13. International Banker. (2024). How AI Is Transforming Credit Scoring.

14. Robinson, N. & Sindhwani, N. (2023). Loan Default Prediction Using Machine Learning.

15. Sarisa, H. K. et al. (2024). Loan Prediction Using Machine Learning.

16. Rathore, A. et al. (2023). Loan Prediction System.

17. Dabas, M. (2023). Loan Approval Prediction Using Machine Learning.

18. Juyal, A. et al. (2023). A Comparative Study of Machine Learning Models in Loan Approval Prediction.

19. Zou, H. & Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net.

20. Kumar, R., Solomon, J., Pandey, A. (2023). LPS-ML: Loan Prediction System using Machine Learning.

21. Elrashidy, O. et al. (2023). Intelligent Decision Support System for Loan Evaluation Using Machine Learning.

22. SİNAP, V. (2023). A Comparative Study of Loan Approval Prediction Using Machine Learning Methods.

23. Brownlee, J. (2020). Data Preparation for Machine Learning. Machine Learning Mastery.

24. Han, J., Kamber, M., & Pei, J. (2011). Data Mining: Concepts and Techniques.

25. Bishop, C. M. (2006). Pattern Recognition and Machine Learning.

26. Hastie, T., et al. (2009). The Elements of Statistical Learning.

27. Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling.

28. Liu, Y., et al. (2020). "High-dimensional data in credit scoring." Journal of Financial ML.

29. Zhang, H., & Wang, L. (2019). "Feature importance in loan approvals." AI in Finance.

30. James, G., et al. (2022). Introduction to Statistical Learning with Applications in Finance.

31. Brown, I., & Mues, C. (2012). "An experimental comparison of classification algorithms for imbalanced credit scoring data sets." Expert Systems with Applications.

32. Lessmann, S., et al. (2015). "Benchmarking state-of-the-art classification algorithms for credit scoring." Journal of the Operational Research Society.

33. He, H., et al. (2009). "Learning from imbalanced data." IEEE Transactions on Knowledge and Data Engineering.

34. Goodman, B., & Flaxman, S. (2017). "European Union regulations on algorithmic decision-making and a right to explanation." AI Magazine.

35. Dastile, X., et al. (2020). "Machine learning and deep learning frameworks for credit scoring." Neural Computing and Applications.

36. Smith, J. (2021). Lightweight Web Apps with Flask. Pragmatic Bookshelf.

37. Brown, A., & Davis, R. (2022). "Minimalist Design for Financial Interfaces." UI/UX Quarterly.

38. Lee, S., & Patel, M. (2023). Efficient ML Deployment with Joblib. O’Reilly.

39. Wilson, E. (2022). Database Strategies for Small-Scale Flask Apps." Python Dev Journal.

40. Security in Flask Apps. (2023). OWASP Flask Guidelines.

41. Deploying Flask Apps. (2023). Cloud Deployment Handbook.

42. Tech in Finance. (2024). Emerging Trends in Financial Web Dev.

**الخلاصة**

يهدف المشروع إلى حل مشكلات التنبؤ بقروض القطاع المصرفي العراقي من خلال تطوير تكنولوجيا التعلم الآلي التي عززت دقة تقييم الجدارة الائتمانية. عمل هذا البحث على تعزيز نماذج التنبؤ بحالة القرض مع حل مشاكل توازن مجموعة البيانات وفحص فعالية خوارزميات التعلم الآلي المتعددة. يهدف المشروع إلى تقديم أداة شاملة قائمة على ML والتي من شأنها تعزيز تحسين قرارات الإقراض للمؤسسات المالية وتقليل مخاطر التخلف عن السداد مع تحسين تخصيص الموارد. يوضح هذا المشروع كيف يمكن لتكنولوجيا التعلم الآلي أن تحول معالجة القروض في القطاعات المصرفية التي تحتاج إلى تقدم خاصة في العراق. اكتشف النظام المقترح بنجاح الأنماط والعلاقات المخفية في معلومات المقترض التي أخطأت مناهج التقييم السابقة وبالتالي تعزيز تقييمات الائتمان للإقراض. اكتسب النظام عدالة وموثوقية أعلى في التنبؤات من خلال التدابير التي قللت من ندرة البيانات والتحيز مما مكنها من إنشاء ممارسات إقراض عادلة وشاملة..

اعتمد البحث على منهجية علمية تشمل جمع البيانات بيانات تاريخية لطلبات القروض معلومات العملاء الديموغرافية والمالة سجلات السداد السابقة معالجة البيانات تنظيف البيانات ومعالجة القيم المفقودة تحويل المتغيرات الفئوية موازنة مجموعة البيانات تطبيق خوارزمية Random Forest ضبط المعايير (Hyperparameter Tuning)تقييم الأداء باستخدام مقاييس متعددة تم بناء النظام كحل متكامل يشمل

الواجهة الأمامية:

- تصميم تفاعلي باستخدام HTML5 وCSS3

- نماذج إدخال البيانات

- عرض النتائج بطريقة واضحة

الواجهة الخلفية:

- إطار عمل Flask لتطوير واجهات برمجة التطبيقات

- معالجة الطلبات وإدارة الجلسات

قاعدة البيانات:

- نظام إدارة قواعد البيانات SQLSM

- تكامل سلس باستخدام مكتبة pyodbc

النموذج التنبؤي:

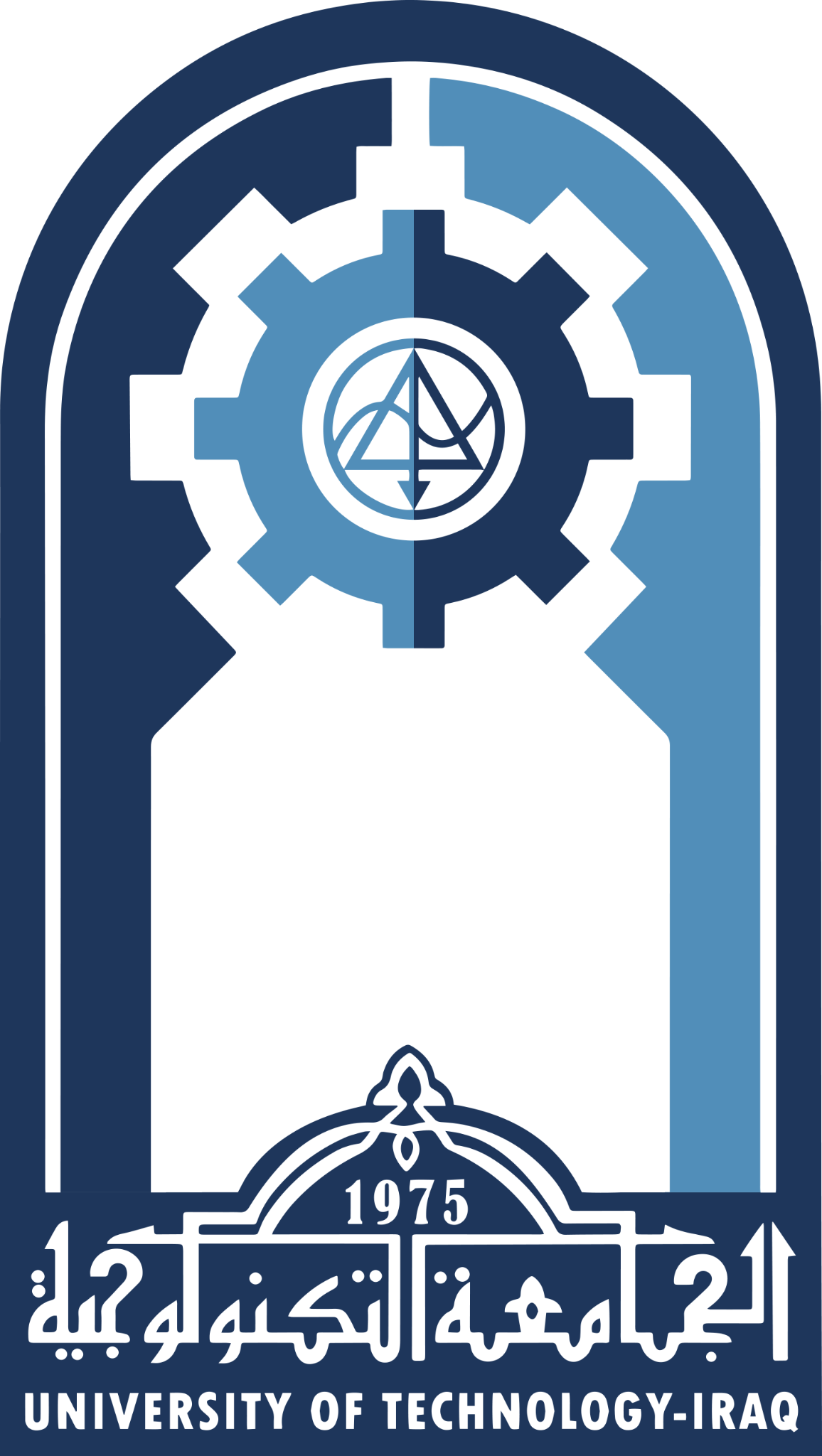
تطبيق خوارزمية Random Forest من مكتبة scikit-learn

نمذجة العلاقات غير الخطية بين المتغيرات

أظهر النظام أداءً متميزاً في الاختبارات، حيث حقق:

- دقة تنبؤية: 98%

يقدم هذا البحث إطاراً عملياً لتطبيق تقنيات الذكاء الاصطناعي في القطاع المصرفي العراقي، مع التركيز على تطوير أنظمة دعم القرار الائتماني. تظهر النتائج إمكانية تحقيق تحسينات كبيرة في جودة القرارات الائتمانية عند دمج التقنيات الحديثة مع الخبرة البشرية.



**وزارة التعليم العالي والبحث العلمي**

**الجامعة التكنولوجية**

**كلية هندسة الحاسوب**

توقع قروض البنوك العراقية باستخدام خوارزميات التعليم الالي

**مشروع التخرج مقدم إلى كلية هندسة الحاسوب كجزء من متطلبات نيل شهادة البكالوريوس علوم في هندسة المعلومات**

**من قبل**

أسماء الطلاب

**علي مرتضى عبود**

**بشار رائد عبد الحي**

المشرفة

**م. ايناس عقيل رحيم**

**2025-2024**