Artificial Neural Network Approach for Credit Risk Management

A report submitted for the degree of Bsc. Computer Science

May 5, 2023

Supervised by Dr Nikolay Nikolaev

Carlos Manuel De Oliveira Alves
Student ID: 33617310

Acknowledgement

I am deeply grateful for the support and guidance I received during this project. My heartfelt thanks go to my supervisor, Dr Nikolay Nikolaev, for his invaluable advice and constant encouragement. I also want to express my appreciation to my friends and family for their unwavering love and support during easy and challenging moments. Last, I thank my partner for being my rock and continuously inspiring me to strive for excellence.

Abstract

The financial industry has long understood the importance of credit risk management, as it plays a critical role in safeguarding the stability and profitability of lending institutions. However, with the advent of advanced technologies, data-driven decision-making has become increasingly vital in the credit risk assessment process. One of the most promising solutions to predict creditworthiness and assist lenders in making informed lending decisions is the use of artificial Neural Networks (ANN). In this project, we propose designing an ANN-based system for credit risk assessment using a random number generator-generated dataset [8] [9][10].

Data analysis in credit risk assessment has become essential for identifying key factors that impact loan approval or rejection. Through a comprehensive evaluation of the dataset, we discovered that credit history, debt-to-income ratio, and total credit utilization are the primary determinants of creditworthiness. These findings provide valuable insights into the relationships between financial variables, which can be leveraged to make better-informed decisions during the credit risk assessment.

To optimize the ANN-based system and ensure its effectiveness, we compared the performance of four classification models – Logistic Regression, Random Forest Classifier, Support Vector Machine (SVM) Classifier, and ANN. By identifying areas for improvement, we aim to enhance the models' performance and accuracy in predicting credit risk. Ultimately, this study serves as a stepping stone towards developing a more efficient credit risk assessment system powered by cutting-edge technology [8] [9][10].

Adopting ANN in credit risk assessment offers numerous advantages over traditional methods. It enables lenders to process vast amounts of data quickly and efficiently, allowing for faster and more accurate credit assessments. Additionally, ANN can adapt and learn from new data, improving the models' predictive capabilities over time. This feature is precious in the ever-changing financial landscape, where the ability to respond quickly to shifts in borrower behaviour and market conditions is crucial.

Moreover, the ANN-based system for credit risk assessment could lead to greater transparency and fairness in the lending process. As the system's decision-making is based on objective, data-driven criteria, it minimizes the potential for human bias and subjectivity. This, in turn, promotes a more equitable lending environment, where individuals are assessed based on their actual creditworthiness rather than any preconceived notions or stereotypes.

However, it is essential to recognize the potential limitations and challenges of implementing an ANN-based credit risk assessment system. The data quality used to train the models is critical, as any inaccuracies or biases in the dataset can negatively impact the predictions. Additionally, the complexity of ANN can make it challenging to interpret and explain the decision-making process, which could lead to issues surrounding transparency and trust in the system.

In conclusion, the development of an ANN-based credit risk assessment can revolutionize the financial industry by providing a more accurate, efficient, and equitable way of evaluating creditworthiness. By leveraging data-driven decision-making and the power of artificial neural networks, lenders can better manage credit risk and make more informed lending decisions. Nevertheless, addressing the challenges associated with implementing such a system is crucial, ensuring that the technology is used responsibly and effectively.

Table of Contents

List of Figures

***Keywords* —** Classification, Machine Learning, Logistic Regression, Random Forest, Support Vector Machines (SVM), Artificial Neural Network (ANN), binary classification task, Approved, Rejected, accuracy, precision, recall, f1-score, class imbalance, hyperplane, high-dimensional data, biological structure, human brain, interconnected layers, artificial neurons, complex patterns, tuning.

Chapter 1 - Introduction

1.1 Aims and Objectives

The primary purpose of this project is to develop a solution to assist financial services in reducing the risk of not coming to the optimal solution when a customer requests a loan. Several distinct algorithms will be implemented into artificial neural networks to accomplish this purpose. These algorithms will then be contrasted and analysed regarding their respective performances and efficiencies.

Consider the possibility of a customer interested in purchasing a car and going to a financial institution to obtain a loan to pay for it. The consumer must fill out an application form, which asks for essential information regarding the customer, such as their salary, the cost of the car, and the amount of money the customer needs to purchase the vehicle. A customer's application for credit is reviewed by the bank, which then decides whether or not to approve the customer for credit based on several factors, including whether or not the customer owns their home, whether or not the customer rents their home, how much money the customer has available in general, and how much money the customer earns. This project aims to develop a model that the bank can use to decide whether or not it is reasonable to lend money to individual customers.

The bank can provide a model that specifies how much money they are willing to lend, and the model will respond with the potential risk that the consumer will not pay back the credit they were given, known as default. The probability, also known as the risk that the customer would default on their payments, will be returned by the model.

This service can help the bank determine whether or not to lend the consumer any money. Therefore, some information about the customers and their applications and a significant amount of background information will be essential to build the model. For each credit application done, the bank knows how much money the consumer requested and whether they can repay the loan. This data from the bank will allow the construction of a model that will analyse all of the account holders and the historical information and processes used to apply for credit.

The model will use binary classification in which the target of our model might be either zero, which indicates that everything is fine, or one, which suggests that something is wrong. With these presumptions in mind, the objective is to train a model that will tell the bank, for each new customer, the probability that the consumer would fail on their loan.

1.2 Section Overview

Chapter 2 - Several studies on using artificial intelligence systems for credit risk management have been conducted, including feature selection and data pre-treatment, the challenges faced by financial institutions in managing credit risk, a method for rating consumer credit based on the long-term memory attention mechanism, and a neural network credit scoring model for classifying credit applications in P2P [7] lending. These studies suggest that machine learning is an effective method for automating credit risk assessment. As a result, banks should consider these methods when evaluating customers' applications or giving them a score.

Chapter 3 – Ethical considerations must be taken into account when developing an artificial neural network to assist financial services in determining whether or not to lend money to customers. This includes avoiding bias, ensuring transparency, clarity, privacy, fairness, explainability, responsibility, and continuous improvement of the ANN's accuracy and fairness over time.

Chapter 4 - The project plan involves defining the problem, reviewing relevant literature, identifying and accessing data sources, preprocessing and cleaning the data, constructing and training the ANN, analyzing its performance, adjusting it if necessary, deploying and monitoring it over time, and considering ethical considerations throughout the process. The goal is to build a reliable and ethical tool that assists financial services in making informed lending decisions.

Chapter 5 - Discusses various machine learning algorithms and techniques for solving classification problems in supervised learning, exploring their mathematical foundations, strengths, weaknesses, and real-world applications. It highlights the importance of considering interpretability and model selection when working with machine learning models. Different models have unique strengths and weaknesses, and performance can be evaluated using accuracy, precision, recall, and F1-score metrics [8] [9].

Chapter 6 - Describes the steps involved in developing an artificial neural network (ANN) based system for credit risk assessment, including feature collection, preparation, model selection and training, prediction, and evaluation. First, the features is generated using random functions in Python [10], while the models are selected and trained using the scikit-learn library and customized ANN model. Finally, accuracy, precision, recall, F1 score, and a confusion matrix evaluate the model's performance [10][11].

Chapter 7 - Analyze the performance of four machine learning models: Logistic Regression, Random Forest, Support Vector Machine (SVM), and Artificial Neural Networks (ANN) for binary classification tasks. The models are evaluated based on accuracy, true positives, true negatives, false positives, and false negatives. The results show that Artificial Neural Networks (ANN) are the most suitable choice for this binary classification task, with better-balanced performance and the ability to handle class imbalances, compared to Logistic Regression, Random Forest, and SVM, which struggle to identify true positives and true negatives.

Chapter 8 - Discuss the performance of four popular machine learning models (Logistic Regression, Random Forest, Support Vector Machines, and Artificial Neural Networks) on a binary classification task. All models exhibit high overall accuracy. However, the Artificial Neural Network model outperforms the others in handling class imbalance, as demonstrated by its improved precision, recall, and f1-score for the "Approved" class.

Chapter 9 - Machine learning, specifically classification, is revolutionizing various industries, with popular algorithms such as logistic regression, decision trees, and support vector machines used to solve real-world problems. Selecting a suitable model for a task is critical. It is crucial to consider factors like interpretability, scalability, and ease of implementation while evaluating performance using accuracy, precision, recall, and F1-score metrics.

Chapter 10 – The Study offers insights into the performance of various machine learning algorithms for binary classification tasks. However, future work should address limitations such as evaluating additional algorithms, using multiple datasets, mitigating class imbalance, exploring alternative metrics, tuning hyperparameters, and enhancing interpretability. Addressing these limitations can lead to more robust and accurate models for real-world applications.

Chapter 2 - Background

2.1 Literature Review

Over the past several years, several studies on the application of artificial intelligence systems for credit risk management have been published in academic journals [1] [2] [3] [4] [5]. Among the research on applying artificial intelligence systems within the classification and discrimination of economic phenomena, with a particular emphasis on managing credit risk, we can highlight Imane, Fatima, Omar [1], Jiang Xiangjian [2], Chongren , Dongmei, Qigang and Suyuan [3] , Ajay, Markku and Jozsef [4] , Ankita, Amit, Aumreesh and Manish [5] .

The paper of Imane, Fatima, and Omar [1] examines several studies and tried approaches on various datasets and concludes that feature selection and data pre-treatment are crucial for producing intriguing findings when using machine learning or deep learning to optimise. They said the best techniques for categorising and predicting loan defaulters use machine learning, especially classification approaches. They said the results could also be impacted by the dataset's size (the number of features and records). Testing the real advancements on larger datasets may be worthwhile for future research. Another important finding is that the application's location may affect the outcomes and timing.

Adding Spatio-temporal data is also a path to development (before or after the Covid crisis). When evaluating customers' applications or giving them a score, banks could consider these methods rather than more conventional ones. Other artificial intelligence apps may be deployed to automate and protect each application procedure.

The objective of the paper Jiang Xiangjian [2] primary focus is to analyse the challenges financial institutions could face while attempting to manage credit risk. The most significant challenge lies in the massive amounts of data about customers' information as well as the unpredictability of their loan behaviour. They put forward an entirely new framework to score the customer's credit and thus predict their future possibility to default seriously or not. The Data Loader can be customised to accommodate the various data formats used by multiple banks. Random Forest is the algorithm most suited to the bank credit case and has an AUC [6] of 0.840. The Output Formatter can make the entire framework appear to users as if it were a black box.

Chongren, Dongmei, Qigang and Suyuan [3] proposed a method for rating consumer credit based on the long-term memory (LSTM [12]) attention mechanism. The user operation behaviour data gathered from the peer-to-peer lending company was the foundation for our developed strategy. To their knowledge, deep learning strategies have never before been implemented in this fashion. This method is validated by applying it to a real-world dataset, and the results of the research indicate that the consumer credit scoring method that is based on the attention mechanism LSTM [12] that is proposed in their paper has a discernible improvement effect when compared with the conventional artificial feature extraction method and the standard LSTM [12] model. This is the conclusion drawn from the findings of the research. In the course of our inquiry, they are only concerned with the chronological sequence in which the activities documented in the data about the activities of online users take place. The amount of time that elapses between each action is not a factor in any way. On the other hand, the length of time that has elapsed since the event in question contains essential information. In their subsequent investigation, they plan to perform thorough research on the subject matter that was just presented.

Ajay, Markku and Jozsef [4] concluded that the neural network credit scoring model had demonstrated promising results in classifying credit applications in P2P [7] lending. This allows the lenders to make an informed decision when picking a loan application. In addition, the comparison between the model and the logistic regression model demonstrated that the neural network performs more effectively when screening potential default loans. Therefore, recognising default loans in advance allows lenders to decrease the money they lose by not investing in less-than-desirable applicants. Their research offers some valuable insights into the use of neural networks in screening loan applications in peer-to-peer lending. According to the findings, neural networks appear to be an excellent method for discarding unwanted applications. However, the research only looks at one particular instance of P2P lending. As a result, in further study, the parallels and variations in the outcomes of other peer-to-peer lending situations that took place in different economic environments will be investigated.

The last paper from Ankita, Amit, Aumreesh and Manish [5] said it is necessary to have an accessible tool for practical judgement to determine whether a specific client is a defaulter. In their study, various machine learning tools are dissected, each of which provides evidence that demonstrates why machine learning is an effective method for automating the process of credit risk assessment. Features such as sex, age, occupation, number of dependant individuals, amount of loan collected, level of education, number of years with current employment, etc., can be used to access any consumer, regardless of whether they are a good or bad risk. These characteristics are essential to look at while determining a customer's trustworthiness. Therefore, the suggested work will be done using a machine-learning technique to improve efficiency.

Chapter 3 - Ethical Considerations

Ethical considerations that should be taken into account while developing a solution based on an artificial neural network (ANN) to assist financial services in determining whether or not it is permissible to lend money to particular customers are as follows:

It is essential to make sure that the predictions made by the ANN do not contain any form of bias.

This may require selecting the training data carefully to ensure that it is representative of the population and employing techniques like data augmentation and balancing to eliminate bias.

Additionally, this may involve ensuring that the data is balanced.

Transparency: The financial service should be transparent about using the ANN in the lending process and provide clear explanations for why loans are accepted or denied.

Clarity: The financial assistance should explain why loans are authorised or declined.

Privacy: The financial service should preserve the customer's data and ensure that it is not shared with third parties or used for any reason other than evaluating the customer's loan application.

For the sake of fairness, the ANN should not show favouritism toward any one particular category of customers while making lending choices.

To accomplish this goal, the ANN can use fairness metrics to make impartial decisions.

Explainability: The financial service should be able to provide consumers with concise explanations for the decisions that the ANN makes regarding lending to guarantee that customers comprehend the rationale behind such choices.

Responsibility: The financial service should be responsible for the accuracy and reliability of the ANN's predictions and should have appropriate safeguards to ensure that the ANN is making fair and unbiased decisions. In addition, the ANN should be responsible for the accuracy and reliability of its own decisions.

Continuous advancement: The financial service should continually assess the performance of the ANN and seek to improve the system's accuracy and fairness over time.

To accomplish this goal, the ANN could need to be retrained on the new data, or its design might be modified as necessary.

Chapter 4 - Project Plan

Define the issue at hand. We first need to describe the problem we are attempting to resolve precisely. In this particular scenario, the challenge consists of designing a solution that can assist financial services in assessing whether or not it is permissible to lend money to specific consumers. Review the relevant literature. After that, we review the relevant literature about utilising ANNs in credit scoring is necessary. We will better understand the area's present status as a result of this, and we will be better able to recognise any potential obstacles or restrictions we may face when building this solution. Find our data sources: To construct and train an ANN, we will need access to an extensive dataset that contains information on previous loans and the results of those loans (e.g., default, repayment). It is essential to locate relevant data sources and ensure that access to such authorities is protected. It will first need to be pre-processed and cleaned to ensure that the data is in a format that can be used for training the ANN. Then, once we have access to them, we will need to perform these steps. This may involve detecting outliers, imputation of missing values, and feature selection. Construct the ANN: Now that the data have been pre-processed and cleaned, we can begin constructing the ANN. This may involve selecting a proper architecture and hyperparameters for the network, implementing the web in a programming language such as Python and Matlab, and training the grid on the data. Analyse the ANN: After the ANN has been constructed, it is essential to analyse how well it works. This may comprise activities such as dividing the data into training, validation, and test sets and using measures such as accuracy, precision and recall to evaluate the performance of the ANN on the test set. Adjust the ANN as necessary: Depending on the evaluation's findings, we might need to make changes to the ANN to improve its overall performance. This may require modifying the network design or the hyperparameters to achieve the desired results. Deploy the ANN once the ANN has proven reliable, it can assist financial services in assessing whether or not it is appropriate to lend money to particular consumers. Again, this can be done when the ANN is working well. This may involve integrating the ANN into existing systems or developing a standalone application that financial services may utilise. Monitor and maintain the ANN: It is essential to do routine monitoring and maintenance on the ANN to ensure that it will continue to function effectively over time. This may involve duties including retraining the ANN on new data, updating the ANN to reflect changes in business requirements or regulations, and addressing any problems or issues that may develop [8] [9] [10] [11].

Chapter 5 – Algorithms and Techniques

Introduction

Classification is a critical task in machine learning that involves mapping input features (x) to output classes (y) by learning a function f (x) = y. This report will discuss various algorithms and techniques for solving classification problems in supervised learning, where models are trained on labelled data. We will explore the mathematical foundations of these algorithms, their strengths and weaknesses, and their applications in real-world scenarios using resources such as Tom Mitchell's "Machine Learning" [8] and Chris Bishop's "Pattern Recognition and Machine Learning" [9].

The Task: Classification

Classification aims to learn a mapping function f (x) = y that can accurately predict the class labels of unseen data based on its features. In binary classification, there are only two possible classes, such as "yes" and "no," "positive" and "negative," or "fraudulent" and "legitimate." In multi-class classification, there are more than two classes, such as "cat," "dog," and "bird." Classification models can be evaluated based on various metrics, such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic (ROC) curve.

The Algorithms:

1. Logistic Regression

Logistic regression is a simple and efficient linear model that estimates the probability of a binary outcome based on a weighted combination of input features [8]. Mathematically, it can be represented as:

$P\ (y = 1 \mid x) = 1\ /\ (1 + \exp(\ -z\ )\ )$

where z = w0 + w1*x1 + w2*x2 + ... + wn*xn

w0, w1, w2, ..., wn are the model's parameters, and x1, x2, ..., xn are the input features. Logistic regression can handle continuous and categorical features and can be regularized to avoid overfitting. However, it assumes a linear relationship between the features and the output, which may not be true in some real-world problems [9].

2. Ensemble Methods: Random Forests and Gradient Boosting

Ensemble methods combine multiple decision trees with improving the model's performance and robustness [8]. For example, random forests use bagging and feature sampling to reduce variance and improve generalization [9]. Gradient boosting, on the other hand, uses boosting and gradient descent to minimize the loss function and optimize the model [8]. Both methods can handle high-dimensional data and noisy features but can be computationally expensive and difficult to tune.

3. Support Vector Machines (SVMs)

SVMs are a powerful classification method that aims to find the optimal hyperplane that separates the data into different classes with the maximum margin [8]. The mathematical formulation of an SVM can be represented as a convex optimization problem:

minimize 0.5 * ||w|| ^ 2 + C * Σ ξ_i, subject to y_i (w·x_i + b) ≥ 1 - ξ_i, ξ_i ≥ 0

SVMs can handle linear and nonlinear data, work with high-dimensional data, and be regularized to avoid overfitting [8]. They can also use kernel functions to transform the input features into a higher-dimensional space where they are more separable [9]. However, SVMs can be sensitive to the choice of kernel function and hyperparameters and can be computationally expensive for large datasets [8].

4. Neural Networks and Deep Learning

We will use Multilayer Perceptrons (MLPs) and backpropagation with neural networks

(NN) for our task. Neural networks, particularly deep neural networks, have lately garnered significant attention and achieved cutting-edge results in various classification tasks [8]. These networks are composed of interconnected layers of neurons that learn intricate, hierarchical representations of the input data. Specifically, MLPs are a feedforward neural network consisting of an input layer, one or more hidden layers, and an output layer. The forward pass involves computing the output of each layer using activation functions, and the backpropagation algorithm is used to optimize the network's weights and biases through gradient descent. The training process of MLPs involves adjusting the weights and biases during each epoch to minimize a loss function, such as mean squared error (MSE), which measures the discrepancy between the predicted output and the true output.

The output of a neuron can be mathematically expressed as:

1. **Forward pass** [8] [9]

$h_1 = \text{sigmoid}(X \cdot W_1 + b_1)$

$\text{y\_pred} = \text{sigmoid}(h_1 \cdot W_2 + b_2)$

Where:

X is the input data or features,
W_1 and W_2 are the weight matrices,
b_1 and b_2 are the bias terms,
h_1 is the output of the hidden layer,
y_pred is the predicted output of the MLP,
sigmoid is the activation function.

2. **Backpropagation and weight updates** [8] [9]

$\Delta y = y - \text{y\_pred}$

$\Delta h_1 = \Delta y \cdot W_2\text{\^{}}T \odot \text{sigmoid\_derivative}(\text{y\_pred})$

$\Delta W_2 = h_1$^T · Δy

$\Delta W_1 = X$^T · Δh_1

$\Delta b_2 = \Sigma(\Delta y)$

$\Delta b_1 = \Sigma(\Delta h_1)$

W_2_new = $W_2 + \eta * \Delta W_2$

W_1_new = $W_1 + \eta * \Delta W_1$

b_2_new = $b_2 + \eta * \Delta b_2$

b_1_new = $b_1 + \eta * \Delta b_1$

Where:

y is the true output,
η is the learning rate,
⊙ denotes element-wise multiplication,
sigmoid_derivative is the derivative of the sigmoid activation function,
Σ denotes the sum over all samples in the batch.

This set of formulas describes a 2-layer MLP (1 hidden layer and 1 output layer) trained using gradient descent optimization. The training function takes input data (X), true output (y), and the number of epochs as input parameters, and it updates the weights and biases during each epoch. The mean squared error (MSE) is used as the loss function [9].

By stacking multiple layers of neurons and optimizing the weights and biases using techniques like backpropagation and gradient descent, neural networks can effectively learn to map input features to output classes [8].

Deep learning architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have demonstrated remarkable performance in image and sequence classification tasks, respectively [8]. CNNs employ convolutional layers to learn local patterns in the input data, while RNNs are designed to capture temporal dependencies in sequences [9].

Interpretability and Model Selection

When working with machine learning models, it is essential to consider the interpretability of the results. Some models, such as logistic regression and decision trees, are highly interpretable, making it easy to understand how the model makes its predictions [8]. However, other models, such as neural networks, can be more challenging to interpret. While they may have higher accuracy rates, it can be difficult to understand how the model makes its decisions, making it harder to explain the results to stakeholders. When selecting a machine learning model, this tradeoff between interpretability and accuracy should be carefully considered [9].

Conclusion:

The field of machine learning is constantly evolving, with new algorithms and techniques always emerging. While the models discussed in this report represent some of the most popular and well-established options, other models may be more appropriate for specific situations. Selecting a suitable machine learning model for a particular task is a critical step in the data science process. Different models have different strengths and weaknesses, and it is essential to carefully consider the specific requirements of the problem when selecting [8]. The performance of a model can be evaluated using metrics such as accuracy, precision, recall, and F1-score, and it is essential to consider factors such as interpretability, scalability, and ease of implementation [9]. By carefully selecting and tuning a machine-learning model, data scientists can unlock valuable insights and predictions that drive significant business value.

Chapter 6 - Application to Credit Risk Management

6.1 Get the Features

The purpose of this project is to develop an artificial neural network (ANN) based system for credit risk assessment. In order to train the ANN, an extensive dataset of information on potential borrowers is required. However, due to ethical considerations, the use of real private or public datasets is not feasible. To overcome this challenge, the project has decided to simulate the features using the random function in Python [10].

The numerical values in the features are generated by defining a range, while the categorical variables are created by compiling a list of possible options. The random number generator is then used to select a random element from the list to create the simulated fetures. This simulated features will be used to train the models, allowing it to make accurate predictions about a borrower's creditworthiness based on distributions representative of the population of potential borrowers.

Dataset Description:

In our machine learning process, we strongly emphasise the features that make up our dataset, as they represent observed feature values from the real world. The dataset includes various features that comprehensively understand an individual's financial situation, creditworthiness, and loan requirements. These features are crucial in making informed decisions on loan approval.

The dataset consists of the following features:

X_1 = Credit history: A record of an individual's past borrowing and repayment activity, including information about late payments and defaults.

X_2 = Employment status: The individual's current employment situation (e.g., employed, unemployed, self-employed, etc.).

X_3 = Collateral: Assets pledged by the borrower to secure a loan, which can be seized by the lender in case of default.

X_4 = Payment history: A record of the borrower's past payments on existing debts and loans.

X_5 = Type of credit accounts: The various types of credit accounts an individual holds (e.g., mortgage, credit card, auto loan, etc.).

X_6 = Public records and collections: Any legal actions or collections related to the borrower's debts or financial obligations.

X_7 = Purpose of loan: The reason for which the borrower is seeking a loan (e.g., home purchase, car financing, debt consolidation, etc.).

X_8 = Income: The borrower's total earnings from various sources.

X_9 = Assets value: The total value of the borrower's assets, including real estate, investments, and personal property.

X_{10} = Debt to income ratio: A ratio that compares the borrower's total monthly debt payments to their gross monthly income.

X_{11} = Length of credit history: The duration of the borrower's credit history, typically measured from the opening date of their oldest account.

X_{12} = Number of credit inquiries: The number of times a lender has requested the borrower's credit report within a specific time frame.

X_{13} = Number of credit accounts: The total number of active and closed credit accounts the borrower has.

X_{14} = Number of credit accounts opened in the last 12 months: The number of new credit accounts the borrower has opened within the past year.

X_{15} = The current balance of credit accounts: The total outstanding balance on all of the borrower's credit accounts.

X_{16} = Total credit limit: The combined credit limit across all of the borrower's credit accounts.

X_{17} = Total credit utilization: The percentage of the borrower's available credit that is currently being used.

X_{18} = Loan amount: The total amount of money the borrower is requesting to borrow.

X_{19} = Saving account balance: The current balance in the borrower's savings account.

Y_1 = Approval status: Whether the loan application has been approved or denied by the lender.

For a logistic regression model [8] [9] with 19 input features and one binary output, the mathematical formula can be represented as follows:

y_pred = sigmoid($w_0 + w_1x_1 + w_2x_2 + \ldots + w_{19}x_{19}$)

Or, in matrix notation:

y_pred = sigmoid($w \cdot X + b$)

Where:

X is the input vector with 19 features ($x_1, x_2, \ldots, x_{19}$),

y_pred is the predicted output (probability of the positive class),

w is the weight vector ($w_1, w_2, \ldots, w_{19}$),

b is the bias term (scalar),

$w \cdot X$ is the dot product of the weight vector and the input vector,

sigmoid is the activation function, which maps the input to a probability

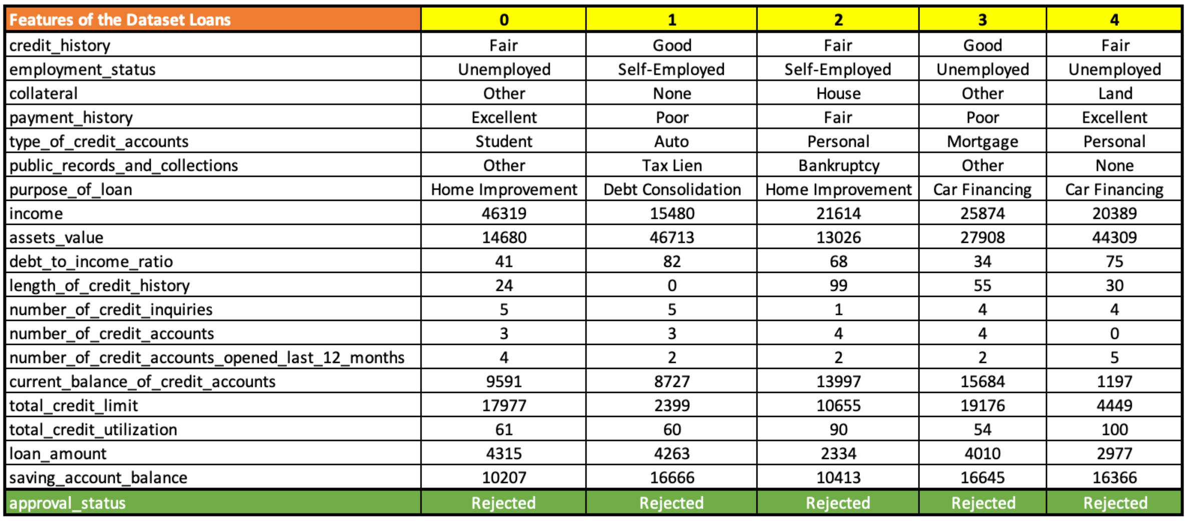
in the range (0, 1):

sigmoid(z) = 1 / (1 + exp(-z))

The logistic regression model estimates the probability of the positive class (Y=1) given the input features X. To make a binary classification decision, and a threshold is typically applied to the predicted probability (e.g., 0.5). If y_pred is greater than or equal to the threshold, the predicted class is 1; otherwise, it is 0.

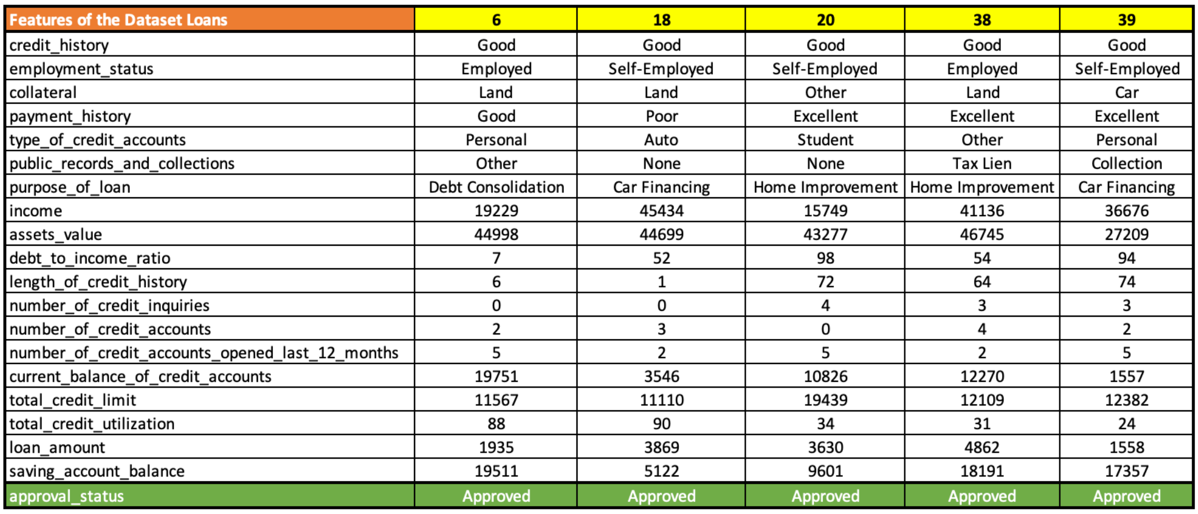
Dataset Analysis:

The dataset analysis reveals several factors that may contribute to the approval or rejection of credit loan applications. For example, credit history, debt-to-income ratio, length of credit history, number of credit inquiries, and total credit utilisation play a significant role in determining loan approval. By understanding these factors, borrowers can work to improve their financial profile and increase their chances of obtaining a loan. At the same time, lenders can better assess the risks associated with each applicant and make informed decisions.



Features of the Dataset Loans	0	1	2	3	4
credit_history	Fair	Good	Fair	Good	Fair
employment_status	Unemployed	Self-Employed	Self-Employed	Unemployed	Unemployed
collateral	Other	None	House	Other	Land
payment_history	Excellent	Poor	Fair	Poor	Excellent
type_of_credit_accounts	Student	Auto	Personal	Mortgage	Personal
public_records_and_collections	Other	Tax Lien	Bankruptcy	Other	None
purpose_of_loan	Home Improvement	Debt Consolidation	Home Improvement	Car Financing	Car Financing
income	46319	15480	21614	25874	20389
assets_value	14680	46713	13026	27908	44309
debt_to_income_ratio	41	82	68	34	75
length_of_credit_history	24	0	99	55	30
number_of_credit_inquiries	5	5	1	4	4
number_of_credit_accounts	3	3	4	4	0
number_of_credit_accounts_opened_last_12_months	4	2	2	2	5
current_balance_of_credit_accounts	9591	8727	13997	15684	1197
total_credit_limit	17977	2399	10655	19176	4449
total_credit_utilization	61	60	90	54	100
loan_amount	4315	4263	2334	4010	2977
saving_account_balance	10207	16666	10413	16645	16366
approval_status	Rejected	Rejected	Rejected	Rejected	Rejected

Figure 1 - Sample of the dataset with Approval Status Rejected

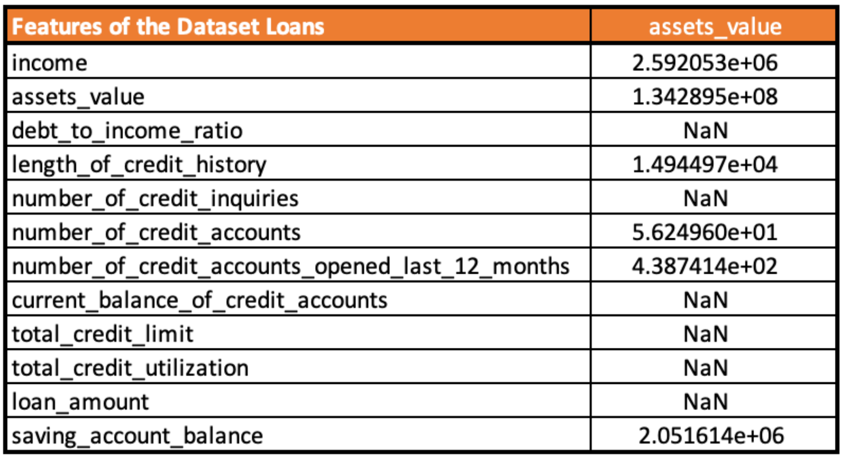


Features of the Dataset Loans	6	18	20	38	39
credit_history	Good	Good	Good	Good	Good
employment_status	Employed	Self-Employed	Self-Employed	Employed	Self-Employed
collateral	Land	Land	Other	Land	Car
payment_history	Good	Poor	Excellent	Excellent	Excellent
type_of_credit_accounts	Personal	Auto	Student	Other	Personal
public_records_and_collections	Other	None	None	Tax Lien	Collection
purpose_of_loan	Debt Consolidation	Car Financing	Home Improvement	Home Improvement	Car Financing
income	19229	45434	15749	41136	36676
assets_value	44998	44699	43277	46745	27209
debt_to_income_ratio	7	52	98	54	94
length_of_credit_history	6	1	72	64	74
number_of_credit_inquiries	0	0	4	3	3
number_of_credit_accounts	2	3	0	4	2
number_of_credit_accounts_opened_last_12_months	5	2	5	2	5
current_balance_of_credit_accounts	19751	3546	10826	12270	1557
total_credit_limit	11567	11110	19439	12109	12382
total_credit_utilization	88	90	34	31	24
loan_amount	1935	3869	3630	4862	1558
saving_account_balance	19511	5122	9601	18191	17357
approval_status	Approved	Approved	Approved	Approved	Approved

Figure 2 - Sample of the dataset with Appoval Status Approved

The dataset consists of 1000 records, with the first five representing rejected loan applications (Figure 1) and the latter five representing approved loan applications (Figure 2). Each record represents an individual borrower's profile, detailing their financial history, current financial status, and the purpose of the loan.

6.2 Discover and Visualise the Features to Gain Insights



Features of the Dataset Loans	assets_value
income	2.592053e+06
assets_value	1.342895e+08
debt_to_income_ratio	NaN
length_of_credit_history	1.494497e+04
number_of_credit_inquiries	NaN
number_of_credit_accounts	5.624960e+01
number_of_credit_accounts_opened_last_12_months	4.387414e+02
current_balance_of_credit_accounts	NaN
total_credit_limit	NaN
total_credit_utilization	NaN
loan_amount	NaN
saving_account_balance	2.051614e+06

Figure 3 - Insights from the Covariance Matrix of Selected Financial Variables

Covariance matrices are powerful statistical tools [11] that help to quantify the relationship between pairs of variables. By analysing the covariance matrix of financial variables, we can better understand how they interact and draw meaningful conclusions about their relationships. This report on figure 3 will examine the covariance matrix of selected financial variables, focusing on income, assets_value, length_of_credit_history, number_of_credit_accounts, and number_of_credit_accounts_opened_last_12_months, and discuss the implications of the observed relationships.

Key Observations:

1. Income and assets_value: The covariance between income and assets_value is 2.592, a positive value. This suggests that individuals with higher incomes typically have higher asset values. This relationship is expected, as higher-income earners are generally more capable of accumulating wealth over time.

2. Length_of_credit_history and assets_value: The covariance between these variables is 1.494, indicating a positive relationship. This suggests that individuals with longer credit histories tend to have higher asset values. The longer a person has been managing credit, the more likely they are to accumulate wealth through investments or other means.

3. Number_of_credit_accounts and assets_value: The covariance between these variables is 5.624, which is also positive. This indicates that individuals with more credit accounts generally have higher asset values. Therefore, those with more credit accounts may have better access to credit, enabling them to invest and grow their wealth.

4. Number_of_credit_accounts_opened_last_12_months and assets_value: The covariance between these variables is 4.387, suggesting a positive relationship. This finding implies that individuals who have opened more credit accounts in the last 12 months tend to have higher asset values. This relationship might be attributed to individuals with more credit accounts having more significant financial resources and opportunities to grow their wealth.

Limitations:

It is important to note that the covariance matrix does not provide information about the strength or causality of the relationships between variables [11]. Additionally, this analysis does not consider several other variables, such as debt_to_income_ratio, current_balance_of_credit_accounts, total_credit_limit, total_credit_utilization, loan_amount, and saving_account_balance. These variables may also play a significant role in understanding the relationships between financial variables and should be considered in future analyses.

Conclusion:

Analysing the covariance matrix of selected financial variables offers valuable insights into the relationships between these variables [11]. For example, the positive covariances between income, assets_value, length_of_credit_history, number_of_credit_accounts, and number_of_credit_accounts_opened_last_12_months suggest that these variables are related and may influence each other. However, it is essential to recognise the limitations of this analysis, as it does not establish the strength or causality of the observed relationships. Further research incorporating additional financial variables and employing more robust statistical techniques is needed to validate and build upon these findings.

6.3 Prepare the Features for the Algorithms

In the field of data science and machine learning [8] [9], the quality and accuracy of a predictive model largely depend on the quality of the data it is trained on. Therefore, proper data preparation is crucial to ensure the model can generalise well to new and unseen data. This report will discuss the steps involved in preparing data for building a predictive model, using a loan approval status prediction as an example. We will cover the following steps: creating a new dataframe, selecting features and targets, standardising the features, and splitting the dataset into training and test sets [8] [9] [10] [11].

Creating a New Dataframe:

The first step in data preparation is to create a new dataframe containing only the highly correlated features with the target variable. In our case, we will focus on predicting loan approval status based on various financial indicators. We can create a new dataframe by selecting the columns highly correlated with the income, such as assets_value, debt_to_income_ratio, length_of_credit_history, number_of_credit_accounts, number_of_credit_accounts_opened_last_12_months, saving_account_balance, and approval_status.

Selecting Features and Target:

Once we have created a new dataframe with the relevant columns, we must separate the features from the target variable. In our example, the features will be stored in a variable called 'X', and the target variable, 'approval_status', will be stored in a variable called 'y'. We can do this by dropping the 'approval_status' column from the new dataframe for the 'X' variable and extracting the 'approval_status' column for the 'y' variable.

Standardising the Features:

Before using the features to train a predictive model, it is essential to standardise them. Standardisation involves scaling the features to have a mean of 0 and a standard deviation of 1. This is important because some machine learning algorithms are sensitive to feature scales, and having features with different scales can lead to suboptimal performance. We can use the StandardScaler class from the scikit-learn library to standardise the features. We first create an instance of the StandardScaler, then fit it to the features, and finally transform it using the scaler [10] [11].

Splitting the Dataset into Train and Test Sets:

The final step in data preparation is to split the dataset into training and test sets. This is crucial because it allows us to evaluate the model's performance on unseen data, ensuring it generalises well. In our example, we will use the train_test_split function from scikit-learn to create X_train, X_test, y_train, and y_test variables, with 80% of the data used for training and 20% for testing. We also set a random_state for reproducibility [10] [11].

Conclusion:

Data preparation is critical in building an accurate and reliable predictive model. By creating a new dataframe with relevant columns, selecting the appropriate features and target, standardising the features, and splitting the dataset into train and test sets, we can ensure that our model has a solid foundation. This process will help us build a predictive model capable of making accurate predictions on new, unseen data, ultimately enhancing its usefulness and applicability in real-world scenarios.

6.4 Select and Train the Models

In machine learning, selecting an appropriate model is crucial in model-building. It requires considering several factors, such as the problem domain, data characteristics, model complexity, and performance metrics. Furthermore, once the model is selected, it must be trained on the training data and evaluated on the test data to assess its performance. This report will discuss the selection and training of four classification models: Logistic Regression, Random Forest Classifier, Support Vector Machine (SVM) Classifier and Artificial Neural Networks (ANN) [8] [9] [10] [11].

We have selected Logistic Regression as the primary classification model for this project.

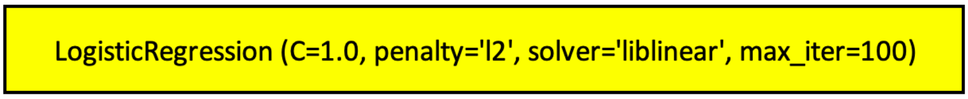
Logistic Regression is a popular model used for binary classification problems. It works by fitting a logistic function to the input features, which maps the input to a probability of a particular class. The logistic regression classifier has several advantages: interpretability, simplicity, and scalability. It is also well-suited for large datasets with a high number of features [8] [9] [10] [11].

We also compared the performance of Logistic Regression with three other classification models, Random Forest Classifier, Support Vector Machine (SVM) Classifier and Artificial Neural Networks (ANN). Random Forest is an ensemble-based method that combines multiple decision trees to improve the accuracy and generalisation of the model. It is known for handling noisy and unbalanced datasets and capturing complex interactions between features. SVM, on the other hand, is a powerful model that uses a kernel trick to map the input data to a high-dimensional space where the classes are linearly separable. Hence, SVM is beneficial for datasets with high overlap between classes [8] [9] [10] [11].

Artificial Neural Networks (ANN) represent another approach to classification problems, offering a highly flexible and adaptable modelling technique. ANNs are inspired by the structure and functionality of the human brain, comprising interconnected nodes or neurons organized into layers. These networks can learn from data through a process known as backpropagation, allowing them to adjust their weights and biases in response to the input-output relationship [8] [9] [10] [11].

6.5 Models Parameters Configuration

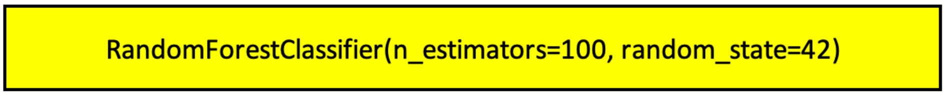
In this project, the Logistic Regression, Random Forest Classifier, and Support Vector Machine (SVM) Classifier models were created using the scikit-learn library [10]. These models were easily created due to the vast array of pre-built functions and modules available in scikit-learn [10]. However, a more challenging approach was taken for the Artificial Neural Network (ANN) model. The model was meticulously developed from the ground up, employing binary classification tasks to optimize its performance, which required a deeper understanding of the underlying concepts. In addition, it involves training a model to predict a categorical label based on a set of input features that are not time-based. This level of customization went beyond the typical coursework and required a solid understanding of Neural Networks. The goal was to have a deeper understanding of the inner workings of artificial neural networks and to demonstrate the ability to create a model without relying on pre-built functions [8] [9] [10].



```
LogisticRegression (C=1.0, penalty='l2', solver='liblinear', max_iter=100)
```

Figure 4 - Snippet of code in Python with Logistic Regression function

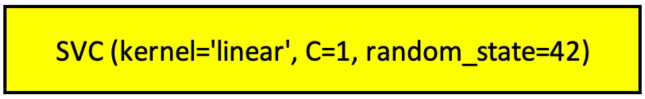
The logistic regression model instantiated with the provided code (figure 4) is defined by several parameters that affect its performance. The 'C' parameter controls the regularization strength, the 'penalty' parameter determines the type of regularization penalty, the 'solver' parameter selects the optimization algorithm, and the 'max_iter' parameter limits the maximum number of iterations allowed for the algorithm to converge. By tuning these parameters appropriately, one can create a logistic regression model that provides optimal performance for a given dataset and problem [10].



```
RandomForestClassifier(n_estimators=100, random_state=42)
```

Figure 5 - Snippet of code in Python with Random Forest function

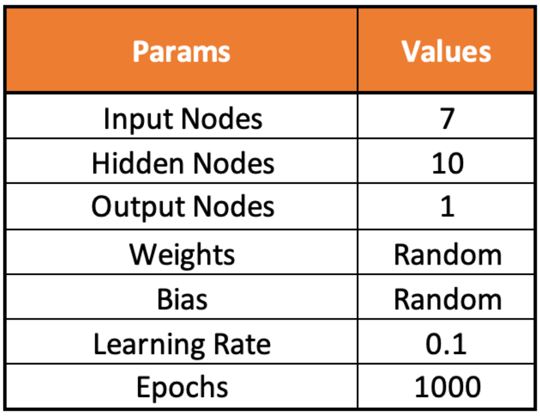
The parameters of the Random Forest Classifier (figure 5) are used to control the behaviour and performance of the random forest classifier. The n_estimators parameter specifies the number of decision trees to use, while the random_state parameter is used to seed the random number generator. Choosing the correct values for these parameters can significantly impact the accuracy and reproducibility of the classifier [10].



```
SVC (kernel='linear', C=1, random_state=42)
```

Figure 6 - Snippet of code in Python with Support Vector Machine function

The code (figure 6) specifies three critical parameters for a Support Vector Machine (SVM) classifier in Python. The kernel parameter specifies the type of kernel function used, the C parameter controls the trade-off between training and testing errors, and the random_state parameter ensures the algorithm's results are reproducible. Understanding these parameters is crucial for fine-tuning the performance of the SVM algorithm and achieving the best possible results.



Params	Values
Input Nodes	7
Hidden Nodes	10
Output Nodes	1
Weights	Random
Bias	Random
Learning Rate	0.1
Epochs	1000

Figure 7 - Table with Artificial Neural Networks parameters

The values used for the parameters in an Artificial Neural Network (ANN) play a crucial role in determining its performance and accuracy. Figure 7 presents the number of Input Nodes, Hidden Nodes, and Output Nodes determined by the input data size and the complexity of the solved problem. Second, the Weights and Bias values are initialized randomly to avoid biases in the network and allow for a more diverse range of solutions to be explored. Third, the Learning Rate is set to a value small enough to avoid overshooting the optimal solution but large enough to allow for rapid convergence. Finally, the number of Epochs is chosen to ensure that the ANN has enough time to converge to the optimal solution, but not so many that it over-fits the data [8] [9].

6.6 Make Predictions and Evaluate the Models

Once the model has been selected and trained, the next step is to use it to make predictions on new data and evaluate its performance. This report will discuss making predictions and evaluating the performance of four classification models: Logistic Regression, Random Forest Classifier, Support Vector Machine (SVM) Classifier and Artificial Neural Networks (ANN).

To make predictions on new data, we use the predict() function of the model and pass the input features as arguments. For example, in Logistic Regression, we would use the predict() function to predict the target variable based on the test data: y_pred =

log_reg.predict(X_test). Similarly, we would use the predict() function to predict the Random Forest Classifier, SVM Classifier and Artificial Neural Networks test data [10].

Once we have made predictions on the test data, we need to evaluate the model's performance. One standard metric for assessing classification models is accuracy, which measures the percentage of correctly classified instances. We can calculate accuracy using the accuracy_score() function, which takes the actual target variable (y_test) and predicted target variable (y_pred) as arguments. Additionally, we can use other performance metrics such as precision, recall, and F1 score to evaluate the model [10].

To better understand the model's performance, we can also use a confusion matrix, which shows the number of true positives, true negatives, false positives, and false negatives. A confusion matrix can be calculated using the confusion_matrix() function, which takes the actual target variable (y_test) and predicts the target variable (y_pred) as arguments [10].

Finally, we can generate a classification report summarising the target variable's precision, recall, F1 score, and support for each class. This can be done using the classification_report() function, which takes the actual target variable (y_test) and predicted target variable (y_pred) as arguments [10].

Chapter 7 - Results

7.1 Evaluate all Models

This report will analyze and compare the performance of four popular machine learning models: Logistic Regression, Random Forest, Support Vector Machine (SVM), and Artificial Neural Networks (ANN) in binary classification tasks. We will base our discussion on the accuracy, true positives, true negatives, false positives, and false negatives obtained from these models.

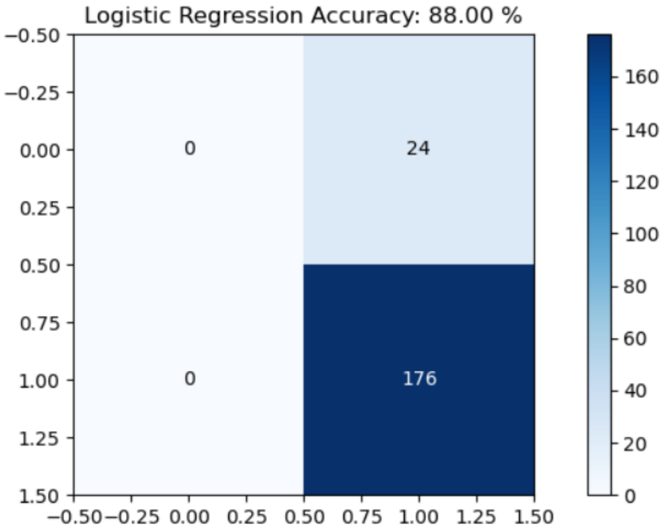


Figure 8 - Confusion Matrix of Logistic Regression Classifier

Logistic Regression is a widely used statistical model for predicting the probability of a binary outcome. In our given case, it has an accuracy of 88.00% (figure 8), with TP = 0, FP = 24, TN = 0, and FN = 176. While the overall accuracy is relatively high, the model fails to correctly identify any true positive or true negative instances. This indicates a severe class imbalance issue, suggesting that the Logistic Regression model may not be suitable for this classification task.

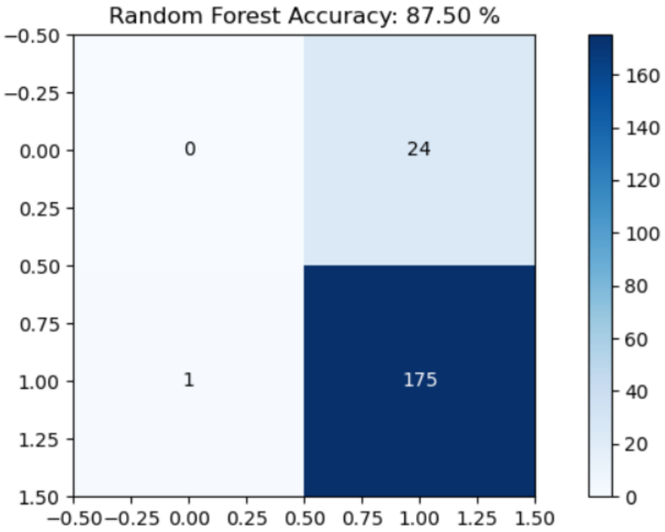


Figure 9 - Confusion Matrix of Random Forest Classifier

Random Forest is an ensemble learning method that constructs multiple decision trees and combines their predictions. With an accuracy of 87.50% (figure 9), it performs similarly to Logistic Regression. The Random Forest model results in TP = 0, FP = 24, TN = 1, and FN = 175. Like Logistic Regression, this model also struggles to identify true positive instances, and its success in identifying true negatives is negligible. This suggests that Random Forest is not ideal for this classification problem.

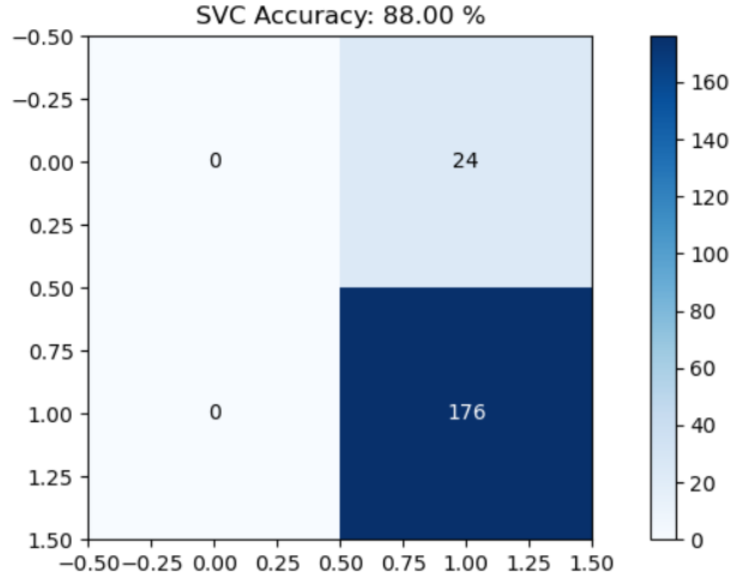


Figure 10 - Confusion Matrix of Support Vector Machine (SVM) Classifier

Support Vector Machine (SVM) is another popular machine learning model that aims to find the best hyperplane that separates the data into two classes. In this scenario, SVM shares the same accuracy as Logistic Regression 88.00% (figure 10). The TP, FP, TN, and FN values are identical to those of Logistic Regression, with TP = 0, FP = 24, TN = 0, and FN = 176. This demonstrates that SVM also needs help in accurately identifying true positives and true negatives, making it a less suitable option for this classification task.

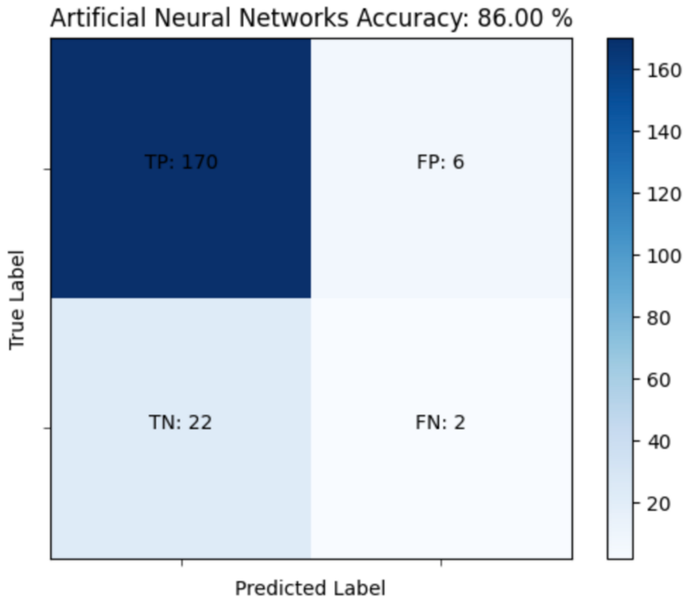


Figure 11 - Artificial Neural Networks (ANN) Classifier

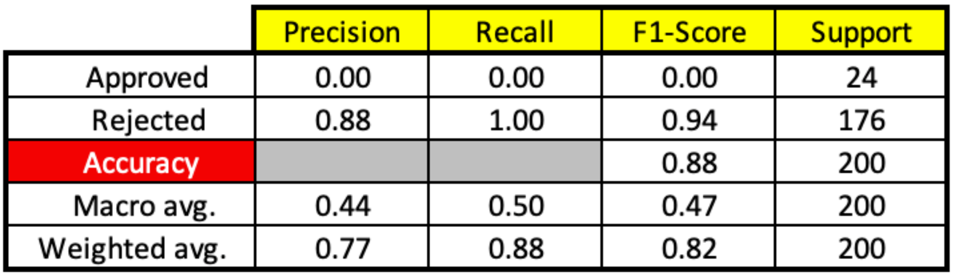
Artificial Neural Networks (ANN): ANNs are computational models inspired by the structure and function of biological neural networks. In this case, ANN exhibits an accuracy of 86.00% (figure 11). However, it performs substantially better in terms of true positives (TP = 170) and true negatives (TN = 22), while having a low number of false positives (FP = 6) and false negatives (FN = 2). Despite slightly lower overall accuracy, ANN outperforms the other models in correctly identifying instances from both classes.

In conclusion, this comparative analysis demonstrates that Artificial Neural Networks are the most suitable choice among the four machine learning models for this binary classification task. While Logistic Regression, Random Forest, and Support Vector Machine all exhibit similar accuracies, they struggle to identify true positives and true negatives, indicating an inability to handle class imbalances. On the other hand, Artificial Neural Networks demonstrate a more balanced performance and are better equipped to tackle this classification problem.

Chapter 8 - Discussion

In machine learning, classification is an essential technique for analyzing and predicting categorical outcomes based on a given set of input features. Several algorithms have been developed to address classification problems, each with strengths and weaknesses. In this report, we will compare the performance of four popular classification models Logistic Regression, Random Forest, Support Vector Machines (SVM), and Artificial Neural Network on a binary classification task to classify cases as either "Approved" or "Rejected."

8.1 Logistic Regression

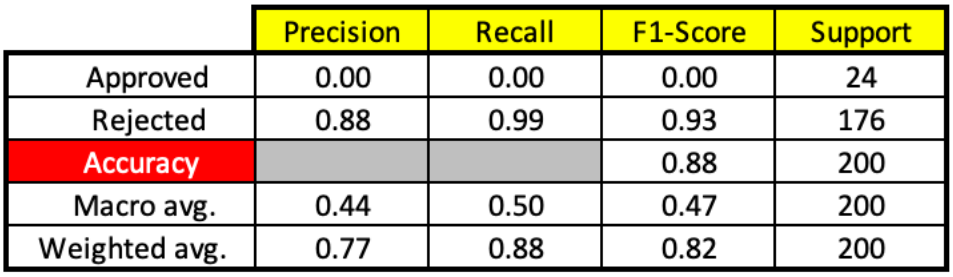


	Precision	Recall	F1-Score	Support
Approved	0.00	0.00	0.00	24
Rejected	0.88	1.00	0.94	176
Accuracy			0.88	200
Macro avg.	0.44	0.50	0.47	200
Weighted avg.	0.77	0.88	0.82	200

Figure 12 - Logistic Regression Classification Report

Logistic Regression is a linear model commonly used for binary classification problems. The model utilizes a logistic function to generate probability values between 0 and 1, which can then be thresholded to obtain the final class predictions. In this case, the Logistic Regression model exhibits an overall accuracy of 0.88 (figure 12), which is relatively high. However, the precision, recall, and f1-score for the "Approved" class are all 0, indicating that the model fails to correctly identify any instances of the "Approved" class. This is a significant shortcoming of the model in this scenario, as it needs to demonstrate better performance in handling class imbalance.

8.2 Random Forest

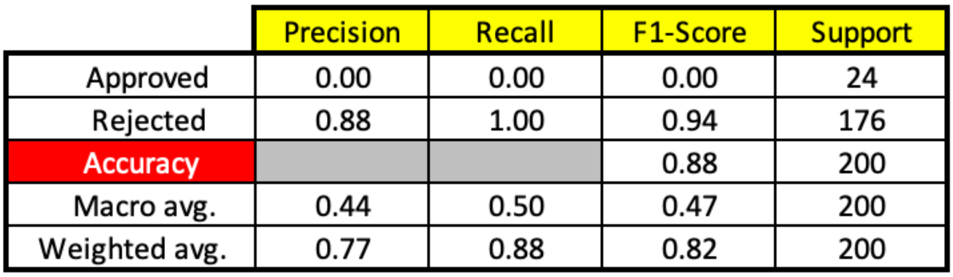


	Precision	Recall	F1-Score	Support
Approved	0.00	0.00	0.00	24
Rejected	0.88	0.99	0.93	176
Accuracy			0.88	200
Macro avg.	0.44	0.50	0.47	200
Weighted avg.	0.77	0.88	0.82	200

Figure 13 - Random Forest Classification Report

Random Forest The Random Forest classifier is an ensemble method that uses multiple decision trees to make predictions. It is known for its robustness against overfitting and its ability to handle large datasets. In this case, the Random Forest model achieves an overall accuracy of 0.88 (figure 13). However, similar to Logistic Regression, it fails to identify instances of the "Approved" class, resulting in precision, recall, and f1-score values of 0 for the "Approved" class. This suggests that the model struggles with class imbalance, just as the Logistic Regression model does.

8.3 Support Vector Machine (SVM)

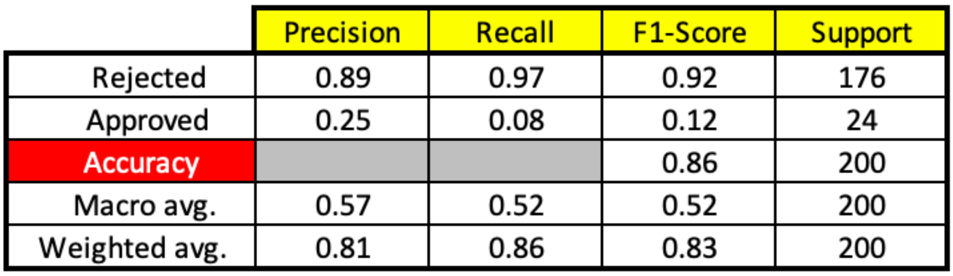


	Precision	Recall	F1-Score	Support
Approved	0.00	0.00	0.00	24
Rejected	0.88	1.00	0.94	176
Accuracy			0.88	200
Macro avg.	0.44	0.50	0.47	200
Weighted avg.	0.77	0.88	0.82	200

Figure 14 - Support Vector Machine Classification Report

Support Vector Machines (SVM) Support Vector Machines is a classification algorithm that seeks to find the optimal hyperplane separating the classes. It is particularly effective for high-dimensional data and has a solid theoretical foundation. The SVM model, in this case, also achieves an overall accuracy of 0.88 (figure 14), but like the previous two models, it fails to identify any instances of the "Approved" class. This indicates that the SVM model cannot handle the class imbalance in this particular problem.

8.4 Artificial Neural Networks (ANN)



	Precision	Recall	F1-Score	Support
Rejected	0.89	0.97	0.92	176
Approved	0.25	0.08	0.12	24
Accuracy			0.86	200
Macro avg.	0.57	0.52	0.52	200
Weighted avg.	0.81	0.86	0.83	200

Figure 15 - Artificial Neural Networks Classification Report

Artificial Neural Networks Artificial Neural Networks are a family of machine learning models inspired by the biological structure and function of the human brain. They consist of interconnected layers of artificial neurons that can learn complex patterns in data. In this case, the Artificial Neural Network model achieves an overall accuracy of 0.86 (figure 15), slightly lower than the other models. However, it outperforms the other models regarding precision, recall, and f1-score for the "Approved" class. In addition, although the scores are still relatively low (0.25, 0.08, and 0.12, respectively), the Artificial Neural Network model can handle class imbalance better than the other models.

In summary, while all four classification models achieve relatively high overall accuracy, they exhibit significant differences in their ability to handle class imbalance. For example, the Logistic Regression, Random Forest, and SVM models fail to identify instances of the "Approved" class, whereas the Artificial Neural Network model demonstrates improved performance. This suggests that Artificial Neural Networks may be a more appropriate choice among these models when dealing with class imbalance. However, further analysis and tuning of the models may still be required to improve their performance on imbalanced datasets.

Chapter 9 – Conclusion

Artificial intelligence (AI) and machine learning (ML) are revolutionizing various industries, from healthcare and finance to manufacturing and transportation. In recent years, the growing availability of data, powerful algorithms, and affordable computing resources has fueled the widespread adoption of ML in business, research, and government. One of the most common applications of ML is classification, which involves assigning input data to predefined categories based on their features. Classification is critical to many real-world problems, such as fraud detection, spam filtering, disease diagnosis, and sentiment analysis. This report explores the concepts, methods, and challenges of classification in machine learning and discuss some of the most popular algorithms and techniques used to solve classification problems [8] [9] [10] [11].

Classification is a type of supervised learning, which means that the machine learning model is trained on labelled data, where the inputs are paired with their corresponding outputs or classes. Classification aims to learn a mapping function that can accurately predict the class labels of unseen data based on its features. For example, in binary classification, there are only two possible classes, such as "yes" and "no," "positive" and "negative," or "fraudulent" and "legitimate." In multi-class classification, there are more than two classes, such as "cat," "dog," and "bird." Classification models can be evaluated based on various metrics, such as accuracy, precision, recall, F1-score, and area under the receiver operating characteristic (ROC) curve. Accuracy measures the proportion of correct predictions, while precision measures the fraction of true positives among the predicted positives. Recall measures the fraction of true positives among the actual positives, while F1-score is the harmonic mean of precision and recall. The ROC curve plots the true positive rate against the false positive rate for different classification thresholds, and the area under the curve (AUC) indicates the model's overall performance [8] [9] [10] [11].

There are many classification algorithms and machine learning techniques, each with strengths and weaknesses. Logistic regression is a simple and efficient linear model that estimates the probability of a binary outcome based on a weighted combination of input features. Logistic regression can handle continuous and categorical features and can be regularized to avoid overfitting. However, logistic regression assumes a linear relationship between the features and the output, which may only be true in some real-world problems [8] [9] [10] [11].

Decision trees are another popular classification method that uses a tree structure to recursively partition the feature space into smaller regions and assign labels to them. Decision trees are easy to interpret and visualize, handle both numerical and categorical data, and capture nonlinear relationships between the features and the output. However, decision trees can be unstable and prone to overfitting, especially when the tree is deep and complex. Ensemble methods such as random forests and gradient boosting combine multiple decision trees, improving the model's performance and robustness. Random forests use bagging and feature sampling to reduce variance and improve generalization, while gradient boosting uses boosting and gradient descent to minimize the loss function and optimize the model. Ensemble methods can handle high-dimensional data and noisy features but can be computationally expensive and difficult to tune [8] [9] [10] [11].

Support vector machines (SVMs) are another powerful and widely used classification method that aims to find the optimal hyperplane that separates the data into different classes with the maximum margin. SVMs can handle linear and nonlinear data, work with high-dimensional data, and be regularized to avoid overfitting. SVMs can also use kernel functions to transform the input features into a higher-dimensional space where they are more separable. However, SVMs can be sensitive to the choice of kernel function and hyperparameters and can be computationally expensive and memory-intensive for large datasets [8] [9] [10] [11].

Neural networks, especially deep neural networks, have recently gained popularity and achieved state-of-the-art results. Another factor to consider when working with machine learning models is the interpretability of the results. Some models, such as Logistic Regression and Decision Trees, are highly interpretable, meaning it is easy to understand how the model makes its predictions. However, other models, such as Neural Networks, can be more challenging to interpret. While they may have higher accuracy rates, it can be challenging to understand how the model makes its decisions, making it harder to explain the results to stakeholders. When selecting a machine learning model, this tradeoff between interpretability and accuracy should be carefully considered [8] [9] [10] [11].

Overall, the field of machine learning is constantly evolving, and new algorithms and techniques are emerging all the time. While the models discussed in this report represent some of the most popular and well-established options, other models may be more appropriate for specific situations. For example, Gradient Boosting Machines (GBMs) are another popular ensemble method that can effectively handle imbalanced datasets. Similarly, Deep Learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are powerful tools for image and language processing, respectively [8] [9] [10] [11].

In conclusion, selecting the suitable machine learning model for a particular task is a critical step in the data science process. Different models have different strengths and weaknesses, and it is essential to carefully consider the specific requirements of the problem when selecting. The performance of a model can be evaluated using metrics such as accuracy, precision, recall, and F1-score, and it is essential to consider factors such as interpretability, scalability, and ease of implementation. While the models discussed in this report represent some of the most popular and well-established options, it is worth exploring other models and techniques to find the best fit for your particular use case. By carefully selecting and tuning a machine-learning model, data scientists can unlock valuable insights and predictions that drive significant business value.

Chapter 10 - Limitations and Future works

Although this study has provided valuable insights into the performance of various machine learning algorithms for binary classification tasks, there are several limitations that need to be addressed in future work.

One of the limitations of this study is that it only evaluates four machine learning algorithms for binary classification tasks. However, numerous other algorithms can be used for binary classification, and it would be beneficial to explore their performance. For instance, decision trees, k-nearest neighbors, and gradient boosting are popular algorithms that have shown exemplary performance in various classification tasks.

Another limitation of this study is that it only uses one dataset. Although the dataset used in this study is widely used in the machine learning community, it would be interesting to explore the performance of the algorithms on other datasets as well. In addition, datasets with different characteristics and properties could lead to different algorithm performances, and therefore, a more comprehensive analysis would be required.

Furthermore, the dataset used in this study suffers from class imbalance, where one class's instances are significantly lower than the other. Class imbalance is a common issue in many real-world problems, and the performance of machine learning algorithms can be severely affected by it. In this study, the Artificial Neural Network algorithm performed better in handling class imbalance than the other algorithms. However, further research is needed to explore techniques to mitigate class imbalance issues, such as resampling, cost-sensitive learning, and threshold-moving techniques.

Moreover, in this study, we only focused on the accuracy, precision, recall, and f1-score metrics to evaluate the performance of the algorithms. While these metrics provide valuable insights into the performance of algorithms, they do not always capture the true performance of algorithms in real-world applications. Therefore, other metrics, such as the area under the ROC curve (AUC-ROC) and the area under the precision-recall curve

(AUC-PR), could be used to evaluate the performance of the algorithms.

In addition, this study only uses default hyperparameters for the algorithms. However, the performance of machine learning algorithms can be significantly improved by tuning their hyperparameters. Therefore, it would be interesting to explore the effect of hyperparameter tuning on the performance of the algorithms.

Finally, the interpretability of machine learning algorithms is a critical issue in many applications. While some algorithms, such as logistic regression and decision trees, provide interpretable models, others, such as artificial neural networks and support vector machines, are black-box models. In this study, we did not explore the algorithms' interpretability. Therefore, future research could focus on developing more interpretable models without sacrificing performance.

In conclusion, while this study provides valuable insights into the performance of various machine learning algorithms for binary classification tasks, there are several limitations that need to be addressed in future work. These limitations include exploring other machine learning algorithms, evaluating performance on different datasets, mitigating class imbalance issues, using alternative evaluation metrics, hyperparameter tuning, and developing more interpretable models. Addressing these limitations could lead to more robust and accurate machine-learning models that can be applied in real-world applications.

Bibliography

[1] Imane Rhzioual Berrada, Fatima Zohra Barramou, and Omar Bachir Alami, ‘A Review of Artificial Intelligence Approach for Credit Risk Assessment’, in 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP) (2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP), Vijayawada, India: IEEE, 2022), 1–5, https://doi.org/10.1109/AISP53593.2022.9760655.

[2] Jiang Xiangjian, ‘Research on Computer Intelligent Risk Prediction Model and Identification Algorithm with Machine Learning’, in 2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT), 2021, 642–48, https://doi.org/10.1109/ICESIT53460.2021.9696813.

[3] Chongren Wang et al., ‘A Deep Learning Approach for Credit Scoring of Peer-to-Peer Lending Using Attention Mechanism LSTM’, IEEE Access 7 (2019): 2161–68, https://doi.org/10.1109/ACCESS.2018.2887138.

[4] Ajay Byanjankar, Markku Heikkilä, and Jozsef Mezei, ‘Predicting Credit Risk in Peer-to-Peer Lending: A Neural Network Approach’, in 2015 IEEE Symposium Series on Computational Intelligence, 2015, 719–25, https://doi.org/10.1109/SSCI.2015.109.

[5] Ankita Mittal et al., ‘A Study on Credit Risk Assessment in Banking Sector Using Data Mining Techniques’, in 2018 International Conference on Advanced Computation and Telecommunication (ICACAT), 2018, 1–5, https://doi.org/10.1109/ICACAT.2018.8933604.

[6] ‘Classification: ROC Curve and AUC | Machine Learning’, Google Developers, accessed 10 November 2022, https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc.

[7] ‘P2P | Definition, Examples, & Facts | Britannica’, accessed 10 November 2022, https://www.britannica.com/technology/P2P.

[8] Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.

[9] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[10] Aurelian Geron, Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow, 2nd Edition, O’Reilly, 2019

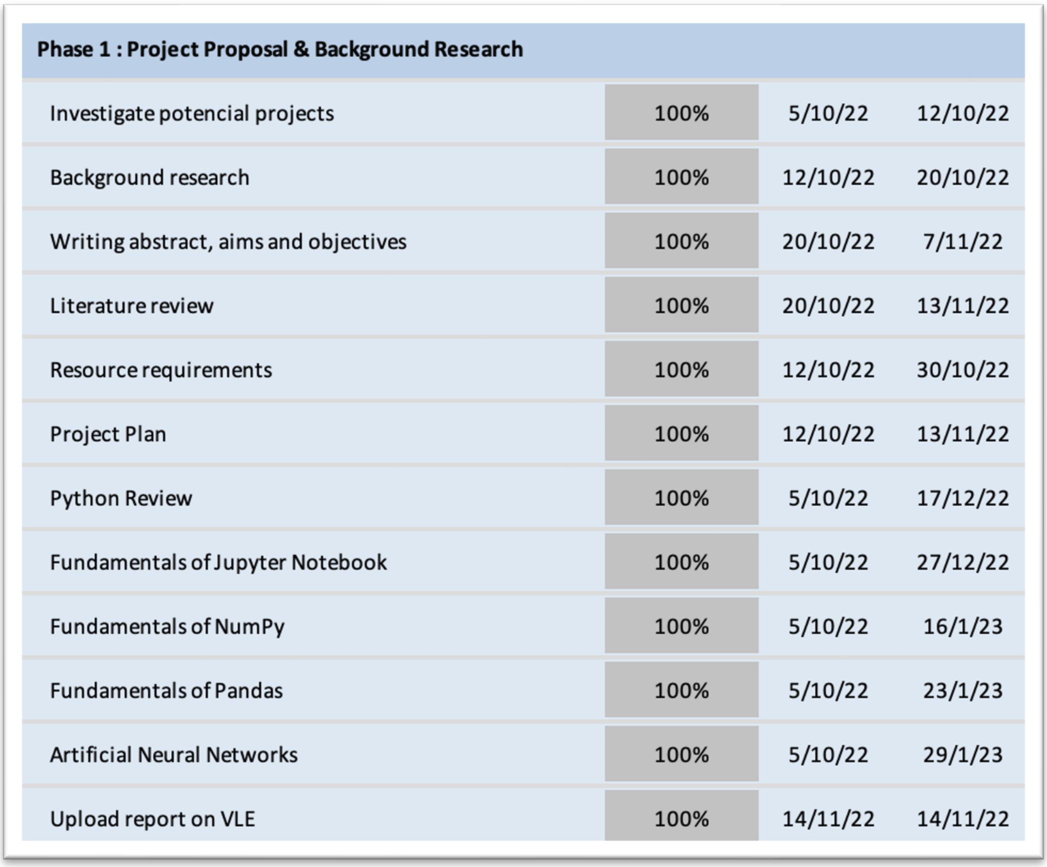
[11] Introduction to Data Mining, 2nd Edition, Pearson, 2018, by Pang-Ning Tan, Michael Steinbach, et al.

[12] LSTM stands for Long Short-Term Memory, a type of recurrent neural network (RNN) architecture commonly used in machine learning (ML) for processing sequential data, such as speech, text, and time series data [8] [9].

Appendices

Appendix A - Progress Logs

Preliminary Report No. 1

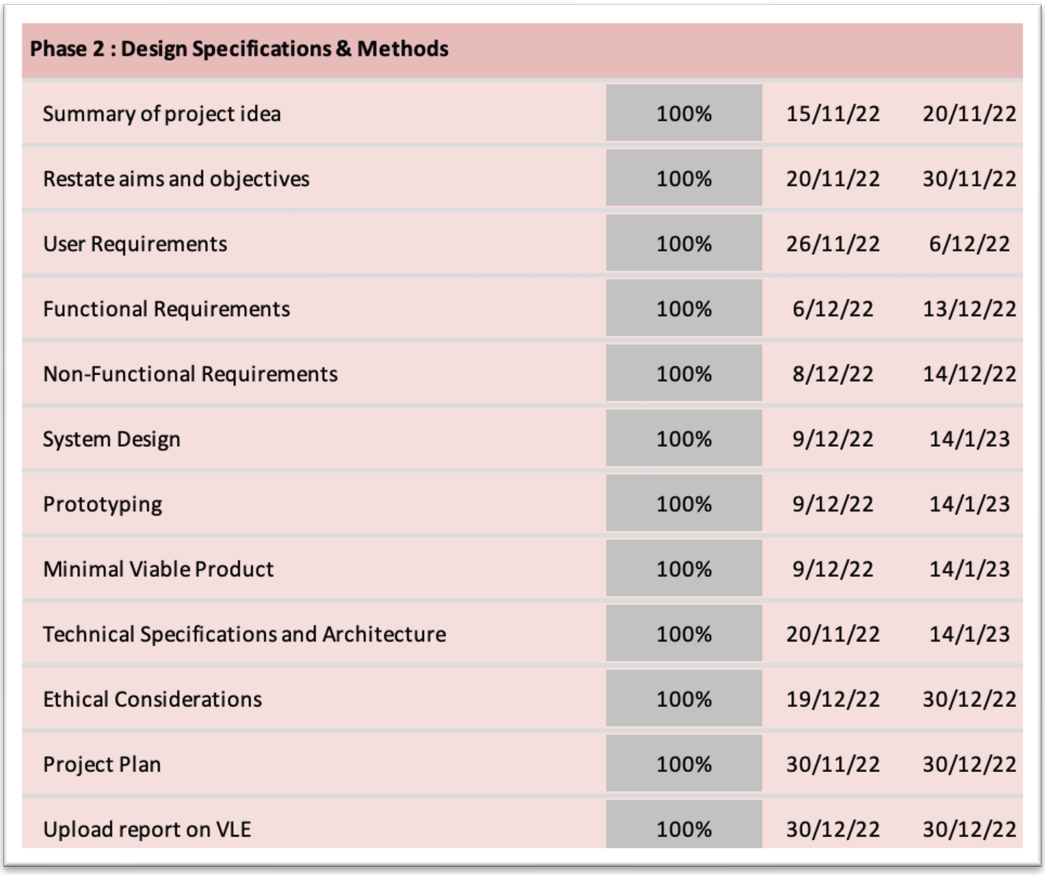


Phase 1 : Project Proposal & Background Research			
Investigate potencial projects	100%	5/10/22	12/10/22
Background research	100%	12/10/22	20/10/22
Writing abstract, aims and objectives	100%	20/10/22	7/11/22
Literature review	100%	20/10/22	13/11/22
Resource requirements	100%	12/10/22	30/10/22
Project Plan	100%	12/10/22	13/11/22
Python Review	100%	5/10/22	17/12/22
Fundamentals of Jupyter Notebook	100%	5/10/22	27/12/22
Fundamentals of NumPy	100%	5/10/22	16/1/23
Fundamentals of Pandas	100%	5/10/22	23/1/23
Artificial Neural Networks	100%	5/10/22	29/1/23
Upload report on VLE	100%	14/11/22	14/11/22

Figure 16 - Progress Log of Preliminry Report No. 1

The milestones accomplished when this report was submitted clearly shown in the Progress Log in Figure 16. The Python review and fundamentals of Jupyter Notebook, NumPy, Pandas, and Artificial Neural Networks are ongoing tasks. They must, however, be completed by the end of the following phase, according to the project plan.

Preliminary Report No. 2

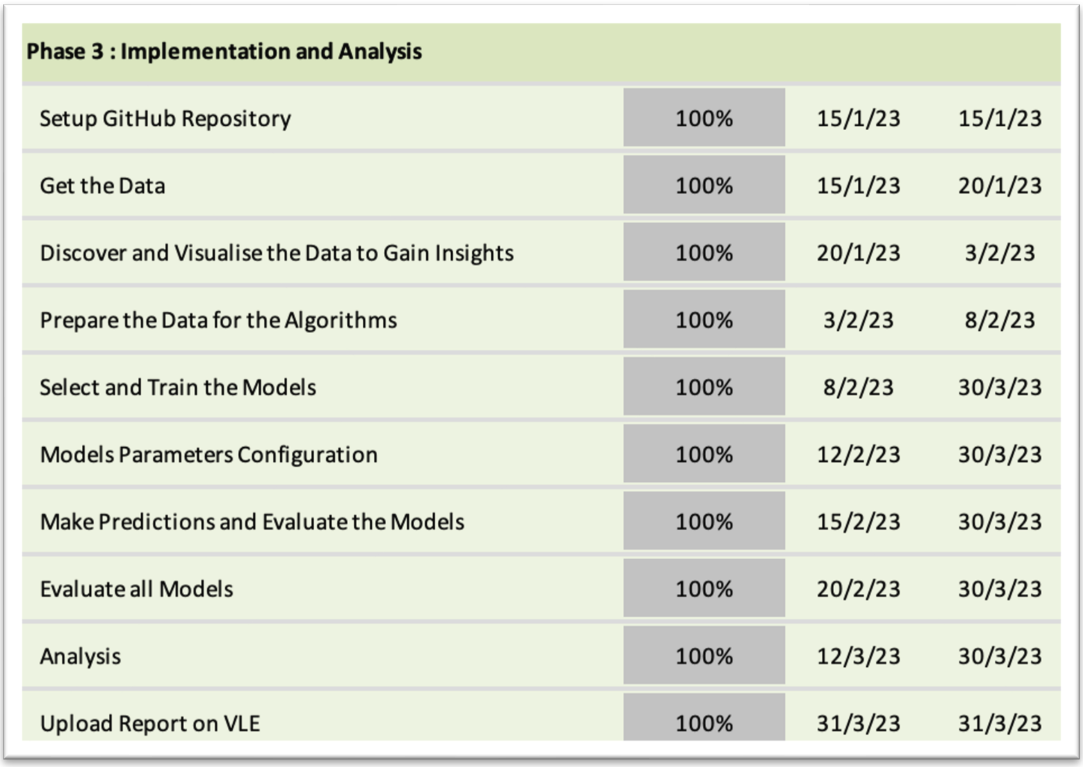


Phase 2 : Design Specifications & Methods			
Summary of project idea	100%	15/11/22	20/11/22
Restate aims and objectives	100%	20/11/22	30/11/22
User Requirements	100%	26/11/22	6/12/22
Functional Requirements	100%	6/12/22	13/12/22
Non-Functional Requirements	100%	8/12/22	14/12/22
System Design	100%	9/12/22	14/1/23
Prototyping	100%	9/12/22	14/1/23
Minimal Viable Product	100%	9/12/22	14/1/23
Technical Specifications and Architecture	100%	20/11/22	14/1/23
Ethical Considerations	100%	19/12/22	30/12/22
Project Plan	100%	30/11/22	30/12/22
Upload report on VLE	100%	30/12/22	30/12/22

Figure 17 - Progress Log of Preliminry Report No. 2

The milestones accomplished when this report was submitted are clearly shown in the Progress Log in Figure 17.

Preliminary Report No. 3



Phase 3 : Implementation and Analysis			
Setup GitHub Repository	100%	15/1/23	15/1/23
Get the Data	100%	15/1/23	20/1/23
Discover and Visualise the Data to Gain Insights	100%	20/1/23	3/2/23
Prepare the Data for the Algorithms	100%	3/2/23	8/2/23
Select and Train the Models	100%	8/2/23	30/3/23
Models Parameters Configuration	100%	12/2/23	30/3/23
Make Predictions and Evaluate the Models	100%	15/2/23	30/3/23
Evaluate all Models	100%	20/2/23	30/3/23
Analysis	100%	12/3/23	30/3/23
Upload Report on VLE	100%	31/3/23	31/3/23

Figure 18 - Progress Log of Preliminry Report No. 3

The milestones accomplished when this report was submitted clearly shown in the Progress Log in Figure 18.

Appendix B - Code Repository

Generate Random Dataset Code

https://github.com/carlos-alves-one/NeuroCredit/blob/master/random_data.ipynb

Logistic Regression Code

https://github.com/carlos-alves-one/NeuroCredit/blob/master/logistic_regression.ipynb

Artificial Neural Networks Code

https://github.com/carlos-alves-one/NeuroCredit/blob/master/artificial_neural_networks.ipynb

Appendix C – Poster A3 Presentation of the Project

Artificial Neural Network Approach for Credit Risk Management

Carlos Manuel de Oliveira Alves, Dr Nikolay Nikolaev

INTRODUCTION

- ➢ Financial solution to minimize loan decision risks
- ➢ Implement various algorithms in artificial neural networks
- ➢ Compare algorithm performances and efficiencies
- ➢ Create a model assess reasonableness lending to customers
- ➢ Estimate probability customer defaulting on loans
- ➢ Use customer information, application data, and historical information
- ➢ Employ binary classification to predict loan repayment success or failure

Image 1: Credit Application Process with Artificial Neural Networks

METHODS

- ➢ Get the Data
  - ✓ Simulate dataset using Python's random function
  - ✓ Various financial features for loan analysis
- ➢ Discover and Visualize the Data
  - ✓ Analyse the covariance matrix
  - ✓ Identify key relationships and limitations
  - ✓ Conclude the need for further research
- ➢ Prepare the Data for Algorithms
  - ✓ Create dataframe with highly correlated features
  - ✓ Select features and targets, standardize features
  - ✓ Split dataset into training and test sets
- ➢ Select and Train Models
  - ✓ Compare Logistic Regression, Random Forest Classifier, Support Vector Machine (SVM), and Artificial Neural Networks (ANN)
  - ✓ Choose Logistic Regression as the primary model
- ➢ Models Parameters Configuration
  - ✓ Configure parameters with scikit-learn library
  - ✓ Develop a customized ANN model from scratch
- ➢ Make Predictions and Evaluate Models
  - ✓ Use performance metrics to assess models
  - ✓ Employ confusion matrix and classification report for further insights

RESULTS

- ➢ Logistic Regression, Random Forest, SVM, ANN
- ➢ Assess accuracy, TP, TN, FP, and false negatives
- ➢ Logistic Regression, Random Forest, and SVM exhibit similar accuracies but struggle with class imbalance
- ➢ ANN has slightly lower overall accuracy but performs better in identifying true positives and true negatives
- ➢ ANN is the most suitable model for this binary classification task, as it handles class imbalances better

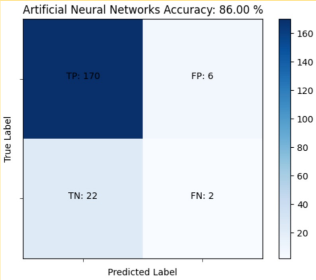
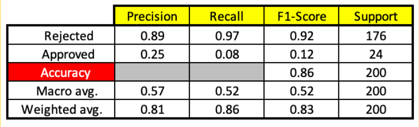


Figure 1 – Confusion Matrix Results



	Precision	Recall	F1-Score	Support
Rejected	0.89	0.97	0.92	176
Approved	0.25	0.08	0.12	24
Accuracy			0.86	200
Macro avg.	0.57	0.52	0.52	200
Weighted avg.	0.81	0.86	0.83	200

Figure 2 – Artificial Neural Networks Reports

LIMITATIONS

- ➢ Only evaluates four algorithms for binary classification
- ➢ It uses only one dataset
- ➢ The dataset suffers from class imbalance
- ➢ Focuses on the accuracy, precision, recall, and f1-score metrics only
- ➢ Uses default hyperparameters
- ➢ Interpretability not explored

FUTURE RESEARCH

- ➢ Explore more algorithms
- ➢ Evaluate performance on various datasets
- ➢ Mitigate class imbalance issues
- ➢ Use alternative evaluation metrics
- ➢ Tune hyperparameters
- ➢ Develop interpretable models

Goldsmiths
UNIVERSITY OF LONDON

References: https://github.com/carlos-alves-one/NeuroCredit/CS_Thesis_FinalProject_2023.pdf
Image 1 created with Microsoft Designer. Figure 1 highlights the model's accuracy and precision, and Figure 2 explores the performance metrics from the artificial neural networks algorithm.
Contact: cdeol003@gold.ac.uk