

# Bank Loan Prediction Using Machine Learning

by

Parthosarothi Mahottam

19101291

Antara Raida Anika

19101574

Dilshad Jahan

18101453

Tanzina Afrin Lazika

21141004

A thesis submitted to the Department of Computer Science and Engineering  
in partial fulfillment of the requirements for the degree of  
B.Sc. in Computer Science

Department of Computer Science and Engineering  
Brac University  
May 2023

© 2023. Brac University  
All rights reserved.

# Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

## Student's Full Name & Signature:



---

Parthosarothe Mahottam  
19101291



---

Antara Raida Anika  
19101574



---

Dishad Jahan  
18101453



---

Tanzina Afrin Lazika  
21141004

# Approval

The thesis/project titled “Bank Loan Prediction Using Machine Learning” submitted by

1. Parthosarothi Mahottam (19101291)
2. Antara Raida Anika (19101574)
3. Dilshad Jahan (18101453)
4. Tanzina Afrin Lazika (21141004)

Of Spring, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 22, 2023.

## Examining Committee:

Supervisor:  
(Member)



---

Nabuat Zaman Nahim  
Lecturer  
Department of Computer Science and Engineering  
Brac University

Program Coordinator:  
(Member)

---

Md. Golam Rabiul Alam, PhD  
Associate Professor  
Department of Computer Science and Engineering  
Brac University

Head of Department:  
(Chair)

---

Sadia Hamid Kazi, PhD  
Chairperson and Associate Professor  
Department of Computer Science and Engineering  
Brac University

# Abstract

The Banking sector is a core foundation of the global economy as it facilitates the financial transactions while providing fundings in various purposes. One key aspect in the banking industry is the ability to accurately predict loan outcomes, which requires assessing the credit worthiness of the loan applicants. Traditional methods of loan prediction are often time consuming, it lacks transparency and interpretability which makes it challenging for the stakeholders to understand the factors that influence loan decisions. With the new addition of machine learning in technology there is an opening to enhance the loan eligibility prediction models and provide transparent insights into the decision-making process. This thesis aims to explore the application of machine learning models and XAI methods for bank loan prediction with a focus on improving accuracy and to get better experience on bank loan applications for both parties. The primary objective here is to develop a robust machine learning model that is capable of accurately predicting loan eligibility and to leverage XAI techniques to explain the reasoning behind these predictions. The research methodology involves a deep analysis from a dataset collected from the internet. The dataset contains various information such as : credit history, loan amount, self employment, earnings etc. Then initial data pre-processing techniques which includes data cleaning or filtering, feature selection and handling values are applied to ensure the quality and the consistency of the dataset. After that, few machine learning models were applied such as: Decision tree, Random forest, NNC etc to build the predictive model. These models are trained and evaluated using appropriate performance metrics such as accuracy, F1-score and AUC(Area under the ROC Curve) score. The goal is to identify the most effective algorithms by comparing them between each other for loan eligibility prediction based on dataset characteristics. Finally, to enhance the transparency and interpretability of the loan prediction models, XAI techniques are applied. These methods facilitate the comprehension of the factors influencing loan decisions, thereby mitigating issues of bias, discrimination, and unfairness. Interpretability techniques, such as analysing feature importance by employing LIME (Local Interpretable Model-agnostic Explanations), are utilised to offer clear and comprehensible explanations for the predictions made by the model. Furthermore, the thesis investigates the ethical implications and fairness considerations associated with loan prediction models. The experimental results demonstrate the efficacy of the proposed approach accurately predicting outcomes while providing interpretable explanations for these predictions. Finally, by utilising machine learning and XAI approaches, this thesis contributes to the subject of bank loan prediction. It provides a complete framework for constructing loan prediction models that are accurate, interpretable, and fair.[5]

Keywords: Machine Learning, XGBoost, Decision tree, LIME , XAI(Explainable artificial intelligence).

## **Acknowledgment**

We would like to express our sincere gratitude to all those who have supported us throughout the completion of this thesis. First and foremost, praise be to the Almighty God for granting the uninterrupted completion of our thesis. We would like to extend our deepest appreciation to our advisor, Nabuat Zaman Nahim, for his direction, support, and encouragement throughout the entire process. His expertise and invaluable feedback have been instrumental in shaping this research and we are truly grateful for his time and dedication. Finally, we would like to acknowledge the contributions and extend our gratitude to our great friend and mentor, Reshad Karim Navid (19101225), for his unwavering support and guidance throughout the entire process, without which our thesis would not be possible.

# Table of Contents

<b>Declaration</b>	<b>i</b>
<b>Approval</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Acknowledgement</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Research Problems . . . . .	3
1.2 Research Objectives . . . . .	3
<b>2 Background</b>	<b>5</b>
2.1 Related Works . . . . .	5
2.1.1 Bank loan prediction using ML: . . . . .	5
2.2 Machine Learning . . . . .	6
2.2.1 Decision trees: . . . . .	7
2.2.2 Random Forests: . . . . .	7
2.2.3 Support Vector Machines (SVMs): . . . . .	8
2.2.4 Neural Networks: . . . . .	9
2.2.5 K-Nearest Neighbors (KNN): . . . . .	10
2.2.6 XG Boost: . . . . .	11
2.2.7 Logistic Regression: . . . . .	12
2.3 eXplainable Artificial Intelligence (XAI) . . . . .	13
2.3.1 LIME . . . . .	14
2.3.2 SHAP . . . . .	14
2.3.3 Limitations of LIME . . . . .	14
2.3.4 Limitations of SHAP . . . . .	14
2.3.5 Why LIME instead of SHAP . . . . .	14
<b>3 Dataset Description</b>	<b>15</b>
3.1 Loan Prediction Problem Dataset . . . . .	15
3.2 Preprocessing . . . . .	15
3.2.1 Rearranging the dataset . . . . .	16
3.2.2 Feature Selection . . . . .	16
3.3 Feature Inference . . . . .	19

<b>4</b>	<b>Experimental Methodology</b>	<b>20</b>
4.1	Model Implementation . . . . .	20
4.1.1	Tree-Based Models . . . . .	20
4.1.2	Logistic Regression . . . . .	22
4.1.3	K-Nearest Neighbors . . . . .	23
4.1.4	NNC . . . . .	23
4.1.5	SVM . . . . .	24
<b>5</b>	<b>Result analysis</b>	<b>25</b>
5.1	Scoring Matrix . . . . .	25
5.2	Analysis Of Trained Bank Loan Dataset: . . . . .	26
5.3	XAI Obeservations . . . . .	30
5.3.1	LIME (Random Forest) . . . . .	31
<b>6</b>	<b>Conclusion</b>	<b>32</b>
	<b>Bibliography</b>	<b>33</b>

# Chapter 1

## Introduction

The main source of revenue for banks is a loan. The majority of the bank's profits are derived directly from the money made from the loans. Even though the bank authorises the loan after an elaborate verification and testimony process, it is still uncertain whether the selected loan is secured or not. When done manually, this technique requires time. We are able to predict whether a specific client is secure or not, and the entire testimonial process is automated using machine literacy. The client and bank retainers both benefit greatly from loan forecasting, because this process will reduce time for loan processing and also ensure secured loans to eligible clients.

Employees of the bank manually review each applicant's information before awarding loans to those who qualify. To apply for a loan, one must submit the following paperwork: Loan application, Proof of identity, Employer and income verification, Proof of address, Get a co-signer, Work on your credit score, Monthly expenses, Loan purpose, etc. The model of an artificial neural network for predicting a bank's credit risk. To predict credit default, a feed-forward back propagation neural network is used. the process of creating an ensemble model by combining two or more classifiers for improved prediction.[6] Following the usage of random forest approaches, they applied bagging and boosting techniques. The purpose of classifiers is to increase the performance and efficiency of the data. The authors of this article discuss numerous grouping strategies for multiclass classification as well as binary classification. Convolutional Oriented Boundaries (COB), a new technique for grouping that the authors describe, performs classification well but suffers from classification data noise and outliers. Finally, they came to the conclusion that the group-based approach enhances the training data set's results.

We created automatic loan prediction using machine learning approaches to solve the issue. We will use the prior dataset to train the computer. in order for the machine to evaluate and comprehend the procedure. The computer will next determine which applications are qualified and inform us of its findings. Advantages:

- Time period for loan sanctioning will be reduced.
- Human error will be avoided.
- Eligible applicants will be sanctioned loans without any delay.



The following is the order in which the paper is arranged: Chapter 2 focuses on related works and provides the background of our research. Section 3 provides a brief description of the Loan Prediction Problem dataset and details the feature preparation process. Section 4 highlights the experimental methodology setup. Section 5 presents the results and findings of all the ML models and XAI. Lastly, Section 6 provides the conclusion of the research.

Fig 1.1 shows the overall workflow.

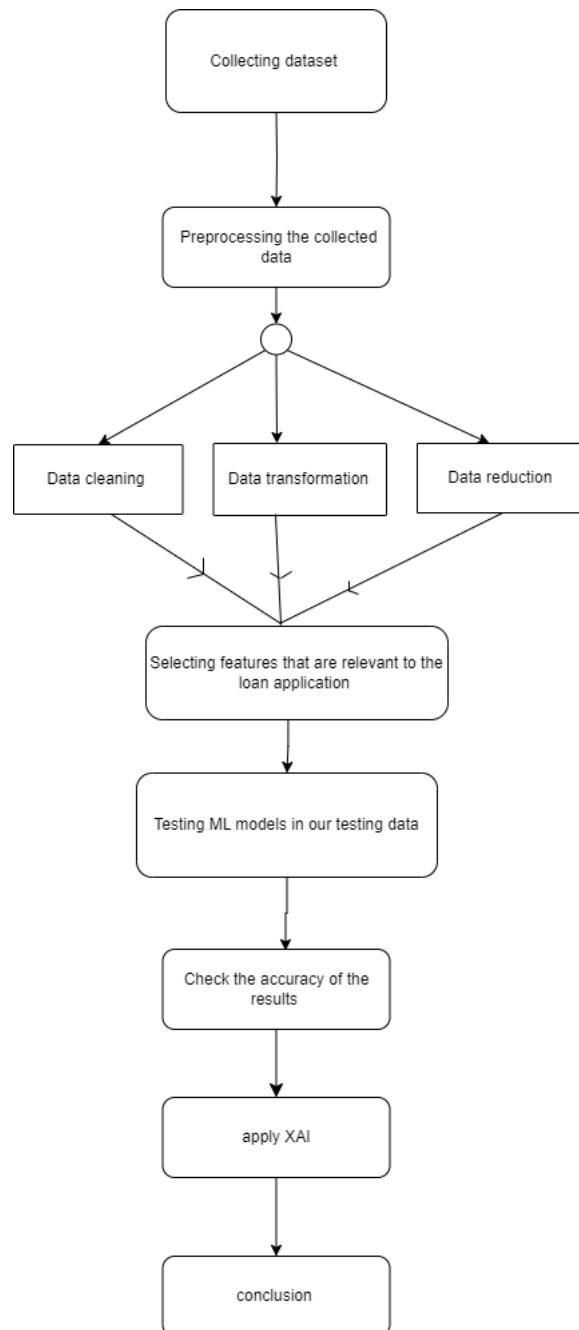


Fig 1.1: Workflow

## 1.1 Research Problems

The topic of loan prediction is one that is frequently discussed in the banking and financial industries. Since technology has developed over time the banking system all over the world is easier than before. Unfortunately, taking loans from banks is still a complex procedure. It takes several days, even months for the banks to approve the loan which can be a big hectic procedure for the clients. Our main goal is to use multiple machine learning approaches to predict the client's loan eligibility. We are planning on using XAI along with machine learning methods. Handful of hampers can be seen during the experiment.

### **Intricacy of using XAI:**

Deep neural networks are one example of an AI model that is extremely sophisticated and challenging to comprehend. As XAI is quite complicated, it might seem hard to define these models' decision-making processes in simple terms.

### **Difficulties in finding right dataset:**

As the information for the bank loan is mostly private, it is quite strenuous to find the fitting dataset with accurate documents. Moreover, finding a dataset that has been pre-processed and has models run on it to support the claims made in a certain research paper is an even more difficult undertaking.

### **Black box nature:**

Black-box machine learning describes machine learning models that provide results or make choices without explicitly stating or substantiating how they performed. The internal procedures and numerous weighting factors used are unknown. For this particular reason, it can be difficult to find the faults and its corresponding results in the decision-making process.[8]

## 1.2 Research Objectives

The main purpose of this research is to find the eligibility before applying for a loan based on an explainable AI (XAI) model. To detect the features and procedures that are indeed essential for identifying anomalies, we are aiming to use Machine learning and XAI. the objectives for this research are following-

1. Preprocessing the dataset to find the clients' eligibility of taking the loan that can be later interpreted by XAI.
2. Understanding the abstruse nature of machine learning and it's multi-parameters and hyper-parameters
3. Improving and building an efficient banking system using machine learning methods.
4. Observing the suitable models which might be the best fit for the loan prediction system.

5. Understanding the working mechanism of machine learning.
6. Interpreting the suitable parameters for the loan prediction system.

# Chapter 2

## Background

In the current era, everything is digitised and banks as well. There was a time when people had to stand in a queue to get their things done. The procedure of loan was also tough for both people and banks. People needed to bring so many papers and banks check those manually which generally costs a lot of time. To give loans banks must check their client's financial status and some other things as well. If they consume too much time by doing these individually it will cause both time and hard work.[2] For this Machine Learning Model can be a saviour. It can accurately determine the prediction of loan for banks which will lessen the time. If all the tasks are done manually it will increase risk factors. By using a Machine Learning model, we can reduce all the risks that the bank faces and, we can help the people do loan related work more conveniently.

### 2.1 Related Works

#### 2.1.1 Bank loan prediction using ML:

Banks usually use analytical and manual techniques to demonstrate certain factors and determine the loan status of an applicant. Loans that have been approved typically have a solid credit history, proper applicant income, and dependability in other areas. The purpose of this is to describe the likelihood that this loan will be approved. The ML Models require datasets to be trained. We can work on the various datasets we will see. We will learn the results after the model has been trained, and then we can choose the algorithm that performs the best. Various machine learning algorithms exist. The algorithms will display their accuracy rate once the models have been trained. The more accurate the result, the higher the accuracy rate will be.[4]. To know about the best fit dataset, there are three terms which are calculated. These are F1 score, accuracy and AUC. A machine learning evaluation metric that assesses a model's accuracy is the F1 score. It combines the precision and recall ratings of a model. AUC denotes the probability that a random positive example will be placed to the right of an arbitrarily negative example. And then comes the accuracy rate which mainly means the total number of predictions the model made. These three categories tell us about the fitness of a model. Several papers have previously been published.

If we go through those, we will have a better understanding of the Machine Learning Algorithms and which algorithm produces the most accurate results. In an article of International Research Journal of Modernization in Engineering Technology and Science we learnt about the functions they used and the implementation processing they went through. The dataset they worked with, gave the best accuracy in Naïve-Bayes Model where Decision tree came second to it . So, they worked on some different datasets for this model. After testing and training those models the Naive Bayes Classifier was again the best performer thus it was chosen[2].

In another article, they mainly worked on the Logistic Regression Model. The process is the same but what differs is the accuracy of Algorithms depending on data sets and how they are trained. In this process they achieved a good accuracy using it, as it fulfils all the purposes, the dataset is useful for further use[5]. An IEEE paper of prediction of loan eligibility shows the dataset and the accuracy rates of different Machine Learning Methods. They used a licensed dataset which is found on Kaggle named “Loan Eligible Dataset”. In this article, six Machine Learning Algorithms were trained, in which, Random Forest Algorithm showed the highest accuracy rate whereas Logistic Regression showed the least accuracy rate among all the used models[3]. In another Journal of Computer Science and Engineering, they trained a model and got some accuracy rates for different models. From that model, Extra Trees Classifier achieved the highest accuracy rate[6]. From the above articles, we learnt that there are many works that have been done before and each work defines different results about different models. Our project is to determine which model works best for our dataset so that we can suggest a new and effective solution to this problem. [9]

## 2.2 Machine Learning

A significant area of artificial intelligence (AI) called machine learning is devoted to creating software programs that, despite not being explicitly intended to do so, try to increase precision by learning data over time. It is a method based on data, where the accuracy of the program as a whole is fundamentally based on the data. Machine learning mostly comes in two flavours. They have both supervised and unsupervised machine learning.

### Supervised Machine Learning

In this method, the output variable is specified, meaning that labelled data is provided to the models. The model is capable of linking mapping functions. And a mapping between the variables in the result. Regression and classification issues are two instances of supervised ML.

To predict the loan eligibility, a variety of supervised machine learning models can be utilised, including:

## 2.2.1 Decision trees:

The most basic tree-based machine learning algorithm is a decision tree. This model enables us to partition the dataset indefinitely based on certain parameters until a final choice is reached. Decision trees are divided into nodes until an outcome is found.

Based on a parameter known as entropy, the decision tree will select a variable to split on first. When a "pure split" is established, that is, when all of the data points belong to a single class, it will stop splitting. A decision tree can be constructed in a variety of ways. The tree must discover a feature to divide on first, second, third, and so on. This structure is based on a statistic known as information gain. The best decision tree is the one with the most information gain. One of the most significant advantages of decision trees is their ease of interpretation. Working backward, it is simple to comprehend how a decision tree achieved its final output based on the training dataset. However, if allowed to grow entirely, decision trees are particularly prone to overfitting. They are unable to generalize successfully to external data since they are designed to split properly on all samples of the training dataset. This disadvantage of decision trees can be overcome by employing the random forest method.

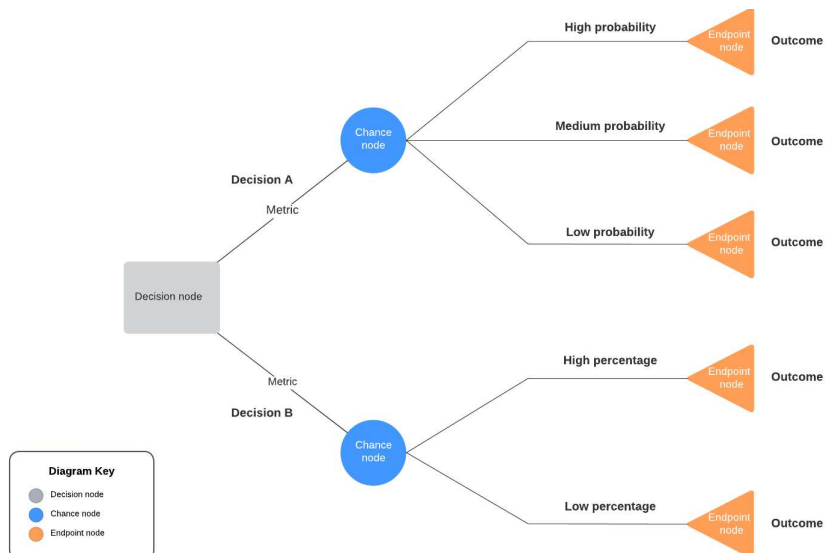


Fig 2.1: Plotting Decision Tree

## 2.2.2 Random Forests:

These models are a collection of decision trees and are called random forests. By integrating the predictions of various trees, random forests can be used to increase the accuracy of decision trees. Random forests are a type of supervised machine learning model that is an ensemble of decision trees. They are commonly used for classification and regression tasks, and are known for their accuracy and robustness. The random forest model is a tree-based approach that assists us in mitigating some of the issues that come when utilizing decision trees, such as overfitting. Random

forests are formed by merging the predictions of numerous decision tree models and producing a single output.

It accomplishes this in two steps:

- Step 1: The dataset's rows and variables are randomly sampled with replacement. Then, for each data sample, several decision trees are generated and trained.
- Step 2: The predictions provided by all of these decision trees are then integrated to produce a single output. The outcome of a regression problem is the average prediction of all decision trees.

Because it mixes the output of numerous decision trees that have been trained on a subset of features, the random forest approach generalizes effectively.

Furthermore, whereas the output of a single decision tree can vary drastically based on a tiny change in the training dataset, the random forest approach does not have this problem because the training dataset is sampled multiple times.

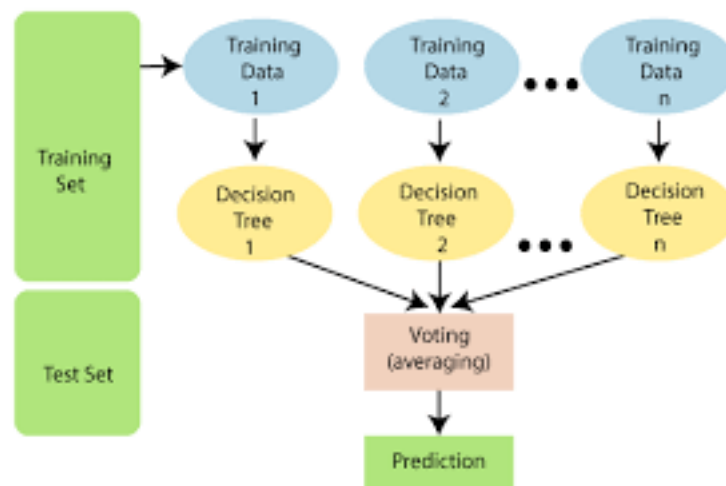


Fig 2.2: Random Forest Algorithm

### 2.2.3 Support Vector Machines (SVMs):

SVM is a strong machine learning method that is used for classification and regression applications. It works especially well with complex, high-dimensional datasets. The SVM method seeks an ideal hyperplane that divides the various classes in the data. The hyperplane is chosen to maximize the margin, or the distance between the hyperplane and the nearest data points of each class (known as support vectors). Through the use of kernel functions, SVM can handle both linearly and non-linearly separable data. The algorithm can use kernel functions to turn the original feature space into a higher-dimensional space where the classes can become linearly separable.

SVM has several major advantages, including the capacity to handle high-dimensional data, resistance to overfitting, and effectiveness even with limited datasets. It is also less affected by local optima because its goal is to maximize the margin. However,

the performance of SVM can be affected by the kernel function and its parameters. Furthermore, for large datasets, SVM training time might be computationally expensive.

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which is defined as data that can be divided into two classes using just one straight line. The classifier used for such data is known as a Linear SVM classifier.
- **Non-linear SVM:** This is used when a dataset cannot be classified using a straight line, it is said to have been non-linearly separated, and the classifier employed is known as a non-linear SVM classifier.

SVMs are considered to be one of the most accurate and robust machine learning algorithms, particularly when the data is not linearly separable. They have been successfully used in many real-world applications, such as text classification, image classification, and bioinformatics. However, they can be sensitive to the choice of kernel and the regularisation parameter, and they can also be relatively sensitive to the scale of the data.

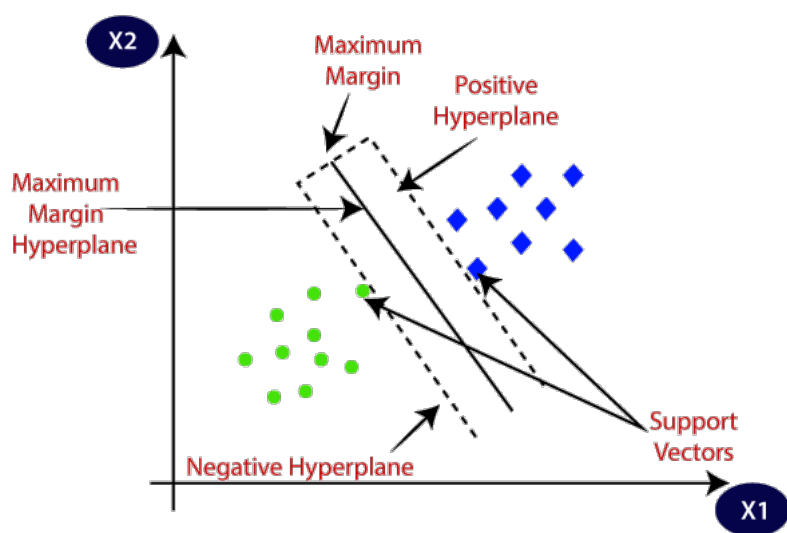


Fig 2.3: SVM Diagram

## 2.2.4 Neural Networks:

Neural networks are a type of supervised machine learning model that are inspired by the structure and function of the human brain. They are commonly used for a wide range of tasks, such as image and speech recognition, natural language processing, and control systems. A neural network is composed of layers of interconnected nodes, called neurons, which are organised in layers. Each neuron receives inputs, performs a computation, and produces an output. The computation performed by each neuron is determined by a set of weights, which are adjusted during training to minimise the error between the predicted outputs and the true outputs. The accuracy of a neural network depends on several factors, including the number of layers, the number of neurons per layer, the choice of activation function, and the



quality of the data. Neural networks can be used to model both linear and non-linear relationships, and they are particularly useful when the data is high-dimensional or when the relationship between the inputs and the outputs is complex. One of the main advantages of neural networks is that they can learn from large amounts of data and can automatically extract features from the data. Additionally, they are relatively robust to noise and can handle missing data. Furthermore, they can be used for a wide range of tasks and can be used in combination with other machine learning models to improve performance.

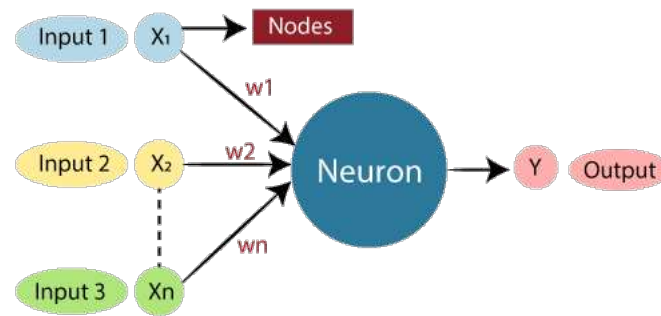


Fig 2.4: Neural Networks Algorithm

In general, neural networks are considered to be one of the most powerful machine learning models and have been successfully used in many real-world applications. They can perform well on a wide range of tasks, but they can be relatively difficult to interpret and they can be sensitive to the choice of architecture and the quality of the data.

### 2.2.5 K-Nearest Neighbors (KNN):

By comparing the features of a new observation to the features of previously witnessed observations, the KNearest Neighbors (KNN) algorithm can be used to categorise network traffic as normal or anomalous. K-Nearest Neighbors (KNN) is a type of supervised machine learning model that is commonly used for classification and regression tasks. It is a simple and intuitive algorithm that can be used for both small and large datasets. The KNN algorithm works by finding the k-nearest training examples to a new observation, and then it classifies the new observation based on the majority class among its k-nearest neighbors. The value of k is a user-defined parameter that controls the number of neighbors to consider. Higher values of k will result in decision boundary of much smoothing, while smaller values of k will result in a more of a complex boundary. The accuracy of the KNN algorithm depends on several factors, including the value of k, the choice of distance metric, and the quality of the data. The algorithm can be sensitive to the choice of k and the distance metric, and it can also be sensitive to the scale of the data and the presence of noise or outliers. One of the main advantages of KNN is its simplicity and interpretability. The algorithm is easy to understand and implement, and it can be used for both small and large datasets. Additionally, it does not make any assumptions about the underlying distribution of the data and it can handle both categorical and numerical data.

- Step 1: Determine the number K of neighbors.

- Step 2: Determine the Euclidean distance between K neighbors.
- Step 3: Determine the K closest neighbors based on the estimated Euclidean distance.
- Step 4: Count the number of data points in each category among these k neighbors.
- Step 5: Assign the new data points to the category with the greatest number of neighbors.

In general, KNN is regarded as a simple yet effective algorithm that has been successfully applied in a variety of real-world applications such as image and speech recognition, natural language processing, and bioinformatics. However, when employed with large datasets, it can be computationally expensive, and it is sensitive to the choice of k and the distance metric.

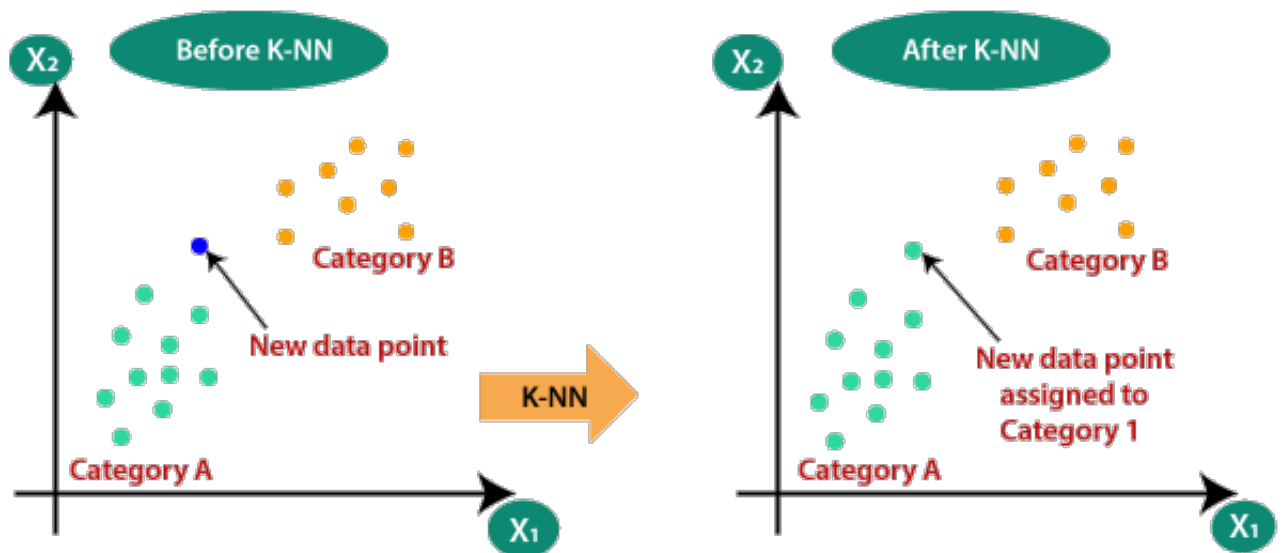


Fig 2.5:KNN Algorithm

### 2.2.6 XG Boost:

One of the most popular model of machine learning as it's known for its efficiency and effectiveness. The model is from the family of gradient boosting algorithms and is very handy to solve problems with classification and regression. The model at first assembles all the weak learners and combines them to form a strong predictive model by forming decision trees, then it trains them sequentially with a goal of each learners correcting the previous mistake it made. This boosting assigns higher weights to each example that are tougher to predict. After that, the XGBoost specifies an objective function that needs to be optimised during training. The objective function is divided into two parts, where one part quantifies the differences between loss functions between predicted and actual values, and after that it regulates terms that

penalises complex models in order to avoid overfitting. Xg Boost then defines an objective function which needing to optimise. Then the Regularization techniques are included in XGBoost to control model complexity and prevent overfitting. Regularisation parameters such as max depth, min child weight, and gamma are used to limit tree development and prune branches that do not significantly contribute to the target function. It then gets to shrinking which works to contribute to control each of the weak learners to the final prediction.

Overall XGBoost plays a vital role to indicate the relative importance of each feature in the prediction process. This information can help identify the most influential features for the task at hand.

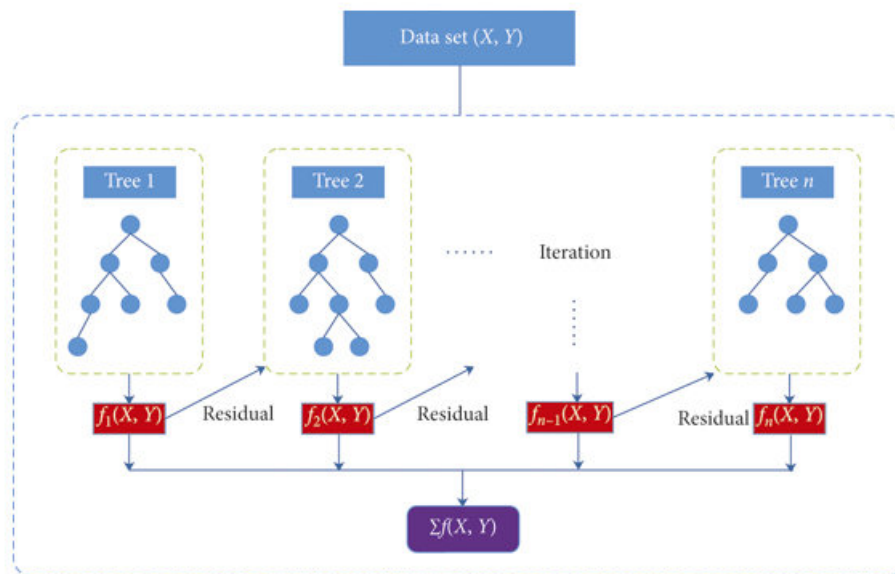


Fig 2.6: XGBoost Algorithm

### 2.2.7 Logistic Regression:

Logistic regression is a statistical model used to predict the likelihood of an event occurring in binary classification situations. It is a linear model in which the logistic function (also known as the sigmoid function) is used to convert the linear output into a probability.

At first it begins by collecting and preparing the dataset. Each data point consists of a set of features or independent variables ( $x$ ) and a binary outcome or dependent variable ( $y$ ) representing the class labels (0 or 1). After that the model assumes a linear relationship between the features and the log-odds of the outcome variable. The log-odds are also known as the logit and are calculated using the equation:

$$\text{logit}(p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$$

Here,  $p$  represents the probability of the positive class  $\beta_0$  is the intercept,  $\beta_1, \beta_2, \dots, \beta_p$  are the coefficients or weights associated with the respective features  $x_1, x_2, \dots, x_p$ . The linear equation is then transformed using the sigmoid function to obtain the predicted probability of the positive class.

The sigmoid function is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Over here the value  $z$  in the sigmoid function is the output of the linear equation (logit) from the previous data. To train the logistic regression model, a suitable loss function is used to measure the discrepancy between the predicted probabilities and the actual class labels. The most commonly used loss function is the binary cross-entropy (log loss) given by:  $L(y, \hat{y}) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})]$ . Then it sets the goal is to minimize the loss function by finding the optimal values for the coefficients (weights)  $\beta_0, \beta_1, \beta_2, \dots, \beta_p$ . This is typically done using optimization techniques such as gradient descent or numerical solvers. The optimization process involves iteratively updating the weights to minimise the loss function. Finally, when the model is trained and the optimal weights are obtained, new data points can be fed into the model. This equation is calculated using the learned weights, and the sigmoid function is applied to obtain the predicted probability. A common threshold of 0.5 is used to classify the data points into their respective classes where: if the predicted probability is above the threshold, the data point is assigned to class 1; otherwise, it is assigned to class 0.

Finally, it can be said that this model is excellent for interpretable model for binary classification problems.

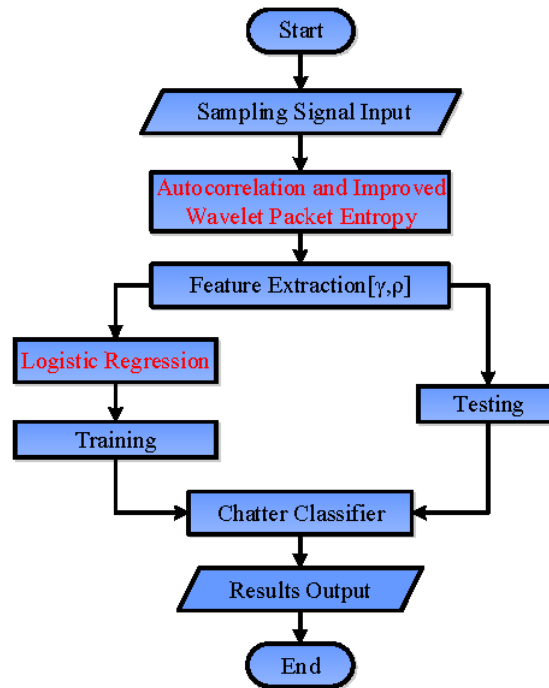


Fig 2.7: Logistic Regression Flowchart

## 2.3 eXplainable Artificial Intelligence (XAI)

An explainable AI (XAI) system's goal is to make its behaviour more understandable to humans by providing explanations [bhattacharya2022appliedoverle](#)

### 2.3.1 LIME

LIME stands for local interpretable model-agnostic explanations. We can use this model while training our data and getting the output.[7] LIME investigates what transpires to the predictions when various iterations of your data are fed into the machine learning model. By combining perturbed samples with the corresponding black box model predictions, LIME generates a completely new dataset.

LIME and SHAP both are useful in Machine Learning Model but both have their own strengths and weaknesses.

### 2.3.2 SHAP

SHAP stands for SHapley Additive exPlanations. It is also XAI like LIME. SHAP, a mathematical approach, is used to explicate machine learning model projection. It can be used to describe the prognosis of an ML model and is primarily subject to game theory. It determines the subsidies for each feature.

### 2.3.3 Limitations of LIME

in general LIME generates data randomly. Due to this problem, it sometimes gives inaccurate results. It is possible to fit a linear model inaccurately. LIME works on unmethodical data so it is possible that it is not stable. To get a more accurate result, we can accumulate SHAP along with LIME to get maximum accuracy[12]. LIME is expensive when working with large and complex datasets.

In conclusion, LIME is a powerful tool, but it should be used with care and in combination with other interpretability techniques to provide a more complete understanding of a model's behavior.[1]

### 2.3.4 Limitations of SHAP

it is used to determine the actual purpose. The SHAP values specify the contribution of each model feature to a prediction. They do not explain how the features contribute to the target variable[13]. SHAP basically shows the importance of a feature in the model. It does not show how it is going to be useful in real life. [13]. SHAP can be less useful for linear models. The goal of the model should always be considered when analysing SHAP output[8].

In conclusion, SHAP is a powerful tool, but it should be used with care and in combination with other interpretability techniques to provide a more complete understanding of a model's behavior.

### 2.3.5 Why LIME instead of SHAP

LIME produces a concerned dataset to fit an explainable model. Whereas SHAP values must be calculated using the complete sample. This indicates that SHAP necessitates multiple observations while LIME necessitates one. Using LIME is more pliable than using SHAP. LIME can determine accuracy of a model individually while on the contrary SHAP needs to calculate the whole thing first to know the individual accuracy. SHAP is not that useful for already interpretable datasets and linear models like LIME.Hence, LIME is much preferable than SHAP[4]

# Chapter 3

## Dataset Description

### 3.1 Loan Prediction Problem Dataset

The data set gathered is divided into a training set and a testing set for the purpose of predicting loan failure. 615 loan data are used for Training Purpose. The 70:30 ratio is typically used to divide the training set from the testing set. The training set is used to apply the decision tree-created data model, which depends on the accuracy of the test set forecast.

This dataset had important features. In table 3.1 **Bank loan Dataset**, we tried to showcase a few of them:

Table 3.1: Bank Loan Dataset

ID	Feature	Description
1	Loan_ID	Unique Loan Identifier Code
2	Gender	Male/Female
3	Marital_Status	Marital Status of Applicant
4	Dependents	Number of individuals dependent on applicant
5	Education	Status of education
6	Self_Employed	Earning source
7	Applicant_Income	Amount of Income
8	Coapplicant_Income	Amount of income of the guarantor
9	Loan_Amount	Amount of loan
10	Loan_Amount_Term	Duration of repayment of loan
11	Credit_History	History of loan/credit payment
12	Property_Area	Urban, Semi Urban or Rural

### 3.2 Preprocessing

The dataset we have found was not gargantuan, yet it was quite disorganised. It had several null values in some necessary sections which made few of the columns of the dataset concoction of string and numeric inputs. Before we executed the

decision tree, random forest, XGBoost algorithms etc. we had to preprocess the dataset which made the procedure easier.

### **3.2.1 Rearranging the dataset**

Firstly, we observed the errors and the issues in the dataset. The unnecessary null inputs were removed from the collection. As the machine does not understand attributes, we convert them into binary numbers(1 or 0) ; e.g: to simplify whether the applicant is married or not, it simply searches for 1 if the applicant is married, if not then the result will be 0. Then ,the integrated string and numerical values were fixed by separating the string values using the apply map' function( using the lambda function.) .

### **3.2.2 Feature Selection**

There are total 12 columns in the dataset where each represents a different feature. We tested different samples for each feature and omitted out one feature to make it more efficient for our models. The below figure depicts feature importance plot:

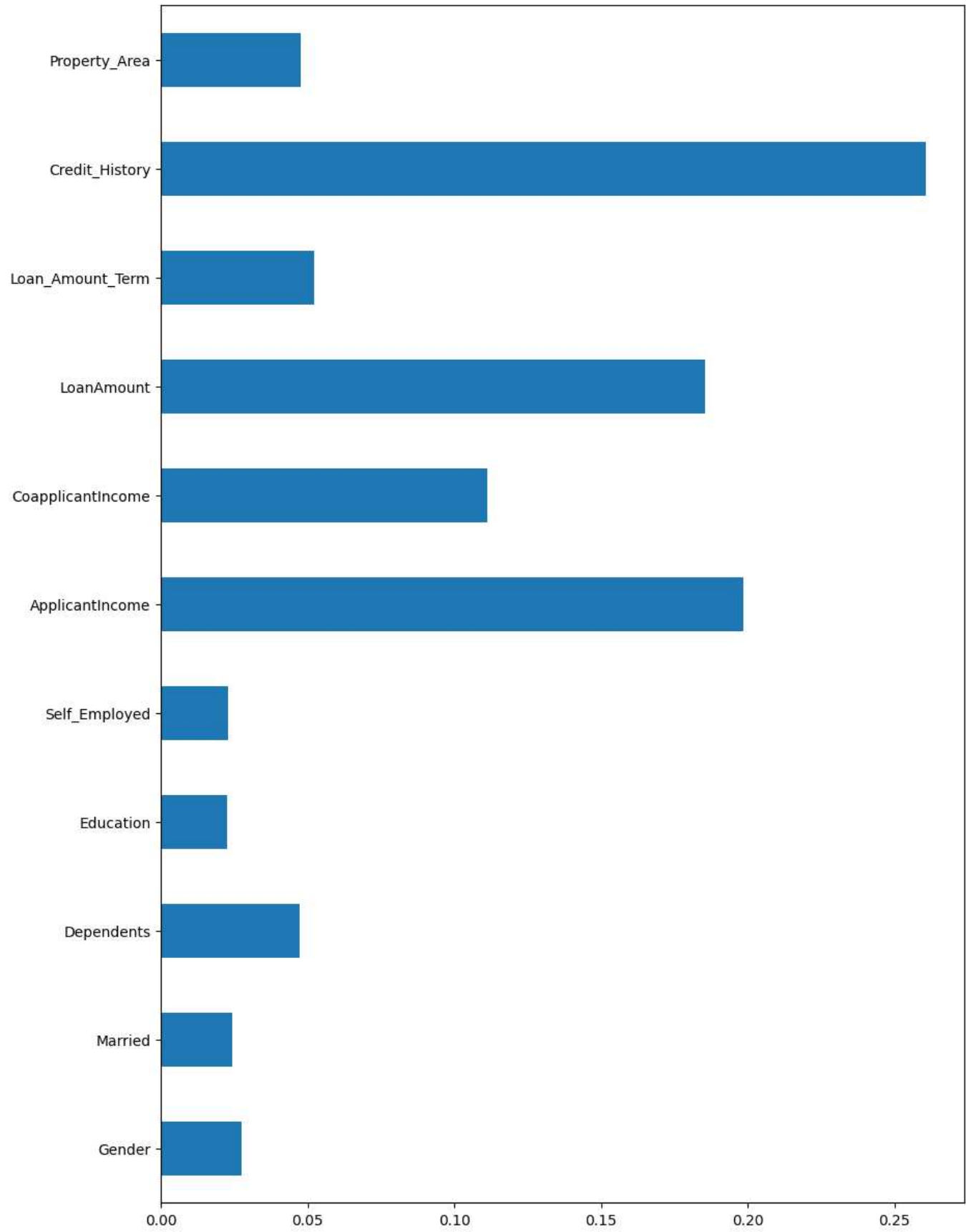


Fig 3.1: feature importance rate



As the dataset was not too big we decided to go for all of them except Loan status as it used for the result of the dataset. After that we plotted a heatmap to cross reference as shown in the below figure:

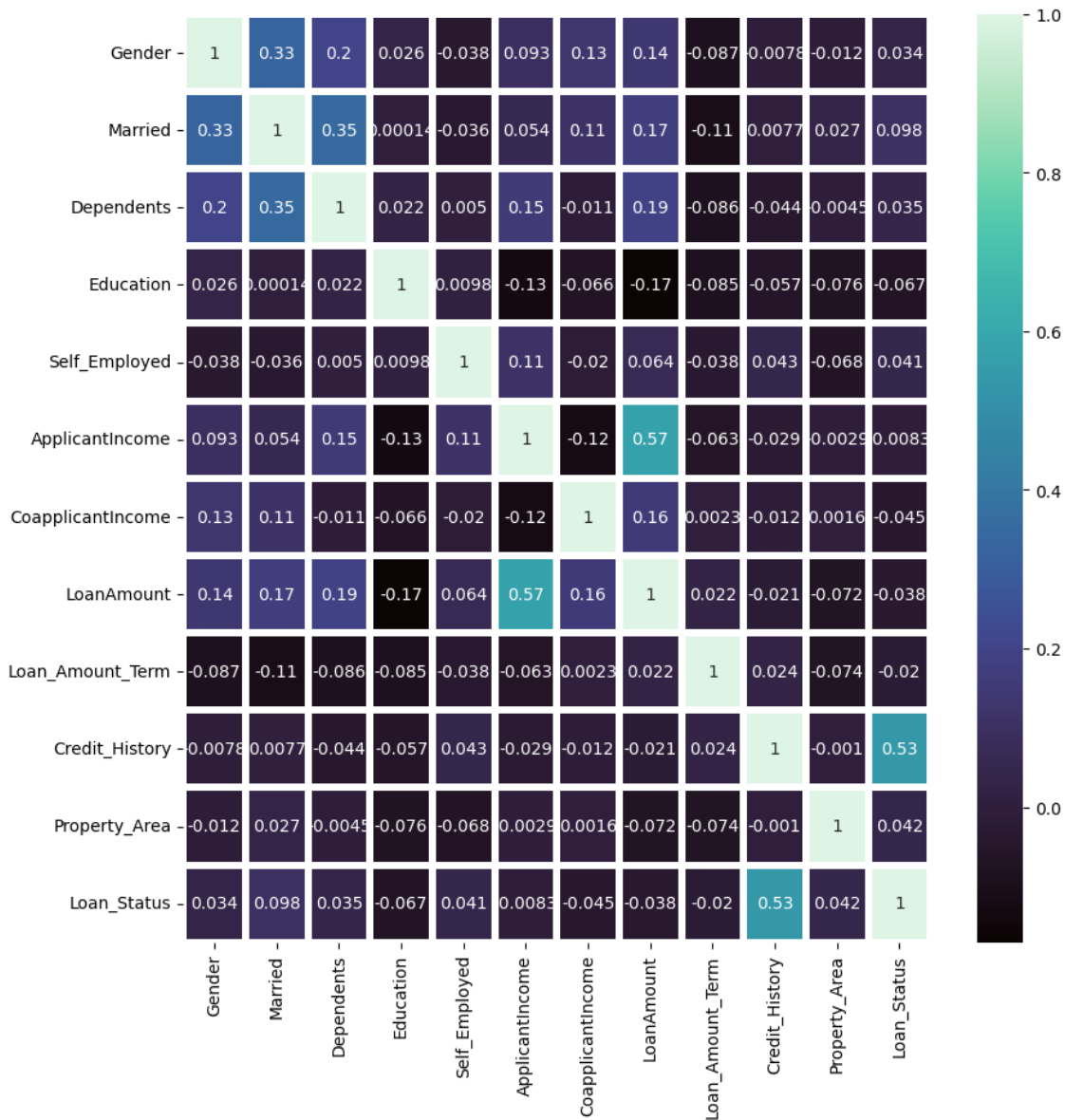


Fig 3.2: Heatmap with all the features

After pre-processing was done, we moved on to train our models. We used Stratified Kfold for our models across the board as it was relatively easier for us.

### 3.3 Feature Inference

Based on the provided dataset "Loan Prediction Problem Dataset," we can infer some insights about the "Loan\_Status" feature. The "Loan\_Status" is likely to be the target variable in this dataset, representing the status of the loan application (whether it was approved or not). Here are some possible inferences about the "Loan\_Status" feature: The "Loan\_Status" feature is likely to be a binary categorical variable with two possible values, such as "Approved" and "Not Approved" or "Yes" and "No." It indicates the final decision on the loan application. After that, The "Loan\_Status" feature is of significant importance in predicting the outcome of loan applications. It plays a central role in understanding the loan approval process and identifying factors that influence the decision. Other features in the dataset, such as "ApplicantIncome," "Credit\_History," "LoanAmount," "Education," "Marital\_Status," "Property\_Area," etc., might be considered as potential predictors that contribute to determining the loan status. Moreover, It is possible that the distribution of loan status in the dataset might be imbalanced, i.e., one class (e.g., "Approved" or "Yes") may dominate over the other class (e.g., "Not Approved" or "No"). This imbalance can have implications on the model training and evaluation process. Finally, Analysing the relationship between the "Loan\_Status" feature and other variables in the dataset, such as through exploratory data analysis or statistical tests, can provide insights into the factors influencing loan approval decisions.

# Chapter 4

## Experimental Methodology

### 4.1 Model Implementation

#### 4.1.1 Tree-Based Models

Some of the tree-based supervised machine learning algorithms are Decision Tree, Random Forest and XGBoost. A group of nonparametric algorithms, also referred to as "tree-based models", break the feature space into a number of few (non-overlapping) sections with comparable response values.

Decision trees are simpler to understand and explain. In this model, both continuous and categorical features/ variables can be handled. In a random forest, the process of bootstrapping refers to the random selection of data instances from the data collection. While decision trees are easy to implement, random forest generates better performance. Then again, even superior to random forest trees is XG Boost. XGBoost stands for "Extreme Gradient Boosting" trees. Each succeeding tree in this process attempts to learn from the mistakes of its predecessor and strives to minimise those mistakes. Weak decision trees are assembled in series to build a powerful learner or estimator.

The parameters for decision trees, random forest trees and XGBoost vary depending on the dataset.

Models	Parameters
Decision tree	<code>DecisionTreeClassifier(criterion='gini', max_depth=None, max_features=1, min_samples_leaf=4)</code>
Random forest	<code>RandomForestClassifier(n_jobs=-1, bootstrap=False, criterion='gini', max_depth=10, max_features=2, min_samples_leaf=2, n_estimators=200)</code>
XGBoost	<code>XGBClassifier(use_label_encoder=False, eval_metric='mlogloss')</code>

With the aforementioned inputs as well as the default parameters, we executed our tree-based models with and without Standard Scaling. Feature scaling is often not necessary for tree-based models.

For decision tree algorithms, The ultimate objective is to construct subsets or groups that share the same desired values.

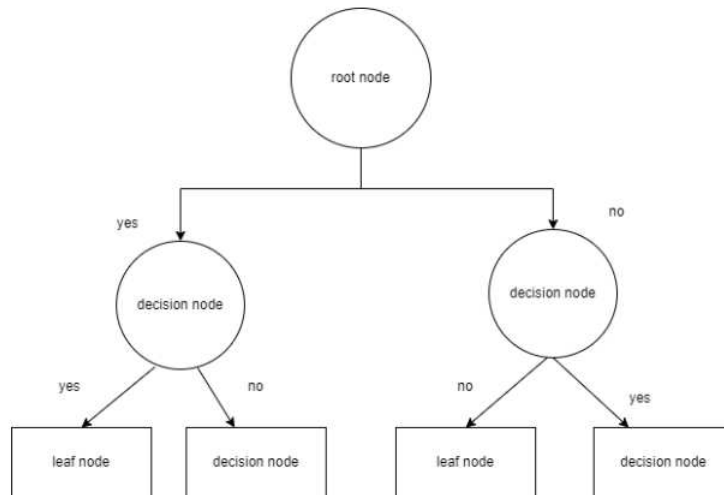


Fig 2.6: Decision Tree

After executing the following model, the results given by the decision tree model are:

- Accuracy of Decision tree: 82.78
- F1 score of Decision tree: 86.87
- AUC Score of Decision Tree: 73.83

The number of decision trees in the forest (`n_estimators`), the maximum depth to which each tree can grow, the minimum number of samples needed for a split to occur at internal nodes (`min_samples_split`), the minimum number of samples needed for a leaf node (`min_samples_leaf`), and the maximum number of features to be taken into account when determining the best split (`max_features`) are the key factors that affect how a random forest behaves. Gini impurity and entropy are both ways to measure the impurity of a set of data. Entropy is typically used for multi-class classification problems, while Gini impurity is typically used for binary classification problems. However, in practice, the number of trees, maximum depth, and maximum features frequently have a greater impact on the overall performance of the random forest than the choice of Gini impurity or entropy as a criterion for splitting in a decision tree.

Our random forest tree model is basically constructed with several decision trees and amalgamated them to generate the final result. This model does not require scaling the features. In some cases, scaling the features before using them can be preferable in this model. The results we get after executing the random forest tree are:

- Accuracy of Random forest: 83.33
- F1 score of Random forest: 88.89
- AUC Score of Random forest: 75.27

For the XGBoost model,

Decision trees are trained as part of XGBoost's operation. The predictions from each tree are integrated to get the final forecast once each tree has been trained on a portion of the data. Here, we used some general parameters and learning task parameters in our project. Our model combines new improvements like regularization with elements from scikit-learn and R implementations.

After running the model, it generates the following results:

- Accuracy of XG Boost: 81.25
- F1 score of XG Boost: 88.0
- AUC Score of XG Boost: 69.94

### 4.1.2 Logistic Regression

While looking at numerous factors is typically more helpful since it highlights the distinct impact of each variable after controlling for the others, logistic regression may just involve one or several independent variables. Logistic regression's parameters include "C," which is the inverse of regularisation strength, "penalty," which designates the regularisation's kind, and "solver," which specifies the algorithm for optimization. [7]The equation for logistic regression: The sigmoid function is referred to as an activation function for logistic regression and is defined as:

$$\text{sigmoid function} = \frac{1}{1 + e^{-\text{value}}}$$

where,

$e$  = base of natural logarithms value = numerical value one wishes to transform

The following equation represents logistic regression:

$$y = \frac{1}{1 + e^{-(b_0 + b_1 x)}}$$

here,

$x$  = input value  $y$  = predicted output  $b_0$  = bias or intercept term  $b_1$  = coefficient for input ( $x$ )

Instead of denoting an estimated continuous outcome in logistic regression,  $\hat{Y}_i$  denotes the estimated likelihood of falling into one of two binary outcome categories ( $i$ ) vs the other.

We chose Stratified K-Fold cross-validation for model training and splitting since it is suitable for the target variable's distribution.

Using the feature importance map created for the random forest classifier, the dataset's features were chosen. In our project, min-max scaler was implemented to standardise the independent variables. Again, to scale the features, we monitored the scatter plot and histogram. We tested our model both ways—with and without the min-max Scaler—and discovered that the Scaler produces the best outcomes.

After training and testing our model results:

- Accuracy of LR: 81.81
- f1\_score of LR: 88.89
- AUC Score of LR: 72.73

### 4.1.3 K-Nearest Neighbors

The k-nearest neighbors, which is a non parametric classification model, has a high cost of classifying new instances.

The K-Nearest Neighbors (KNN) algorithm's performance was assessed using stratified k-fold cross-validation, which divided the data into k partitions or "folds" and trained the algorithm on k-1 of them while testing the remaining folds k times in total. All features of equal importance but not on the same scale should be normalised. Otherwise, the feature with the largest magnitude will dominate the overall Euclidean distance unless Manhattan distance is used. Feature scaling is essential because ANNs are used to measure distance. I used a MixMaxScaler to scale the values between 0 and 1.

For this dataset, MinMaxScaler and StandardScaler gave comparable results, so both are applicable. The value of k is often determined through experimentation, and common distance include Euclidean distance, Manhattan distance, and Minkowski distance. In KNN, we have to select any similarity measure which can be euclidean distance or cosine similarity, the have to create a similarity matrix to train the dataset. After training and testing the model, the accuracy rate is:

- Accuracy of KNN: 77.78
- f1\_score of KNN: 85.71
- AUC Score of KNN: 66.88

### 4.1.4 NNC

Neural network can generate an estimation of the probability of a pattern to belong to a class. The 'supervisor' system that allocates the probability estimation generated by individual classifiers grants the final decision.

$$H_j = f \left( \sum_{i=1}^n (w_{ji}x_i + b_i) \right) \quad (4.1)$$

Here,  $w_{ji}$  represents weights,  $b_i$  represents biases, and  $f()$  represents the nonlinear activation function for the output of neuron  $j$  in the hidden layer.

The equation for the network output is given by:

$$y = f \left( \sum_{j=1}^m (w_{kj}H + b_o) \right) \quad (4.2)$$

Here,  $f$  represents the output layer neuron activation function,  $w_{kj}$  represents weights, and  $b_o$  represents the bias. Here, we used MLPClassifier, where Stratified K-fold cross-validation technique. Here, we used MLPClassifier, where the Stratified K-fold cross-validation technique was used to divide the features. With the help of `x_train` (a subset of the input features) and `y_train` (a subset of the target variable), the MLP classifier (`nnc`) is trained on the training set using the `fit` technique. In order to reduce the discrepancy between expected and actual outputs, this stage entails optimizing the neural network's weights and biases.

After executing the code, we get the prediction rate:

- Accuracy of NNC: 81.48
- f1\_score of NNC: 86.58
- AUC Score of NNC: 72.57

#### 4.1.5 SVM

Support vector machines divide input values into hyperplanes. Hyperplanes can have different shapes depending on how the data is distributed, but only the points that help categorize the classes are taken into account. SVM's primary goal while categorizing the datasets is to find the most extreme minor hyperplane.

A Stratified K-fold cross-validation method was used to divide the input features ( $x$ ) and marked variable ( $y$ ) into multiple train-test sets. The model is then initialized.

Step	Description
1	The desired parameter ( $y_{\text{pred\_svm}}$ ) for the test set ( $x_{\text{test}}$ ) is predicted using the trained SVM classifier.
2	The <code>predict</code> method is used to obtain the predicted labels.
3	The <code>accuracy_score</code> from the <code>sklearn.metrics</code> module is used to calculate the accuracy.
4	After training and testing through the F1Score, the results are obtained.

- Accuracy of SVM: 69.44
- F1-score of SVM: 81.67
- AUC Score of SVM: 50.12

# Chapter 5

## Result analysis

### 5.1 Scoring Matrix

In this section our main focus is to investigate the outcomes we get while training our machine learning models. While training the models we verified our outcomes using three specific methods which are as following:

- F1 Score.
- Accuracy.
- AUC (Area Under the ROC Curve) score.

At first the F1 score is a way to measure the accuracy that takes into account both recall and precision in a confusion matrix. The fraction of true positives accurately recognized by our model can be quantified using a statistic known as recall or sensitivity. Precision, on the other hand, is a quantitative measure of how near the actual value is to the expected value. These two parameters combine to form the function that determines the F1 Score.

Here,

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$
$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$
$$\text{F1 Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}).$$

On the other hand, Classification models can be compared to one another using an accuracy metric. It computes a model's accuracy rate, or the percentage of successful predictions compared to the total number of forecasts. The formula used to calculate this is depicted below:

$$\text{Accuracy} = (\text{True Positives} + \text{True Negatives}) / \text{Total predictions}.$$

A Receiver Operating Characteristics (ROC) curve, on the other hand, is a graph that displays the performance of a binary classifier as the threshold for classifying input points is changed. The graph compares the true positive rate (TPR) to the false positive rate (FPR) at various threshold values. Another method for evaluating the performance of a binary classifier is the AUC (Area Under the Curve) score.



This score is calculated by calculating the area under the ROC curve at various threshold settings. It is computed by plotting the TPR (sensitivity) versus the FPR (1-specificity) for various threshold values and then computing the definite integral of the TPR versus the FPR.

## 5.2 Analysis Of Trained Bank Loan Dataset:

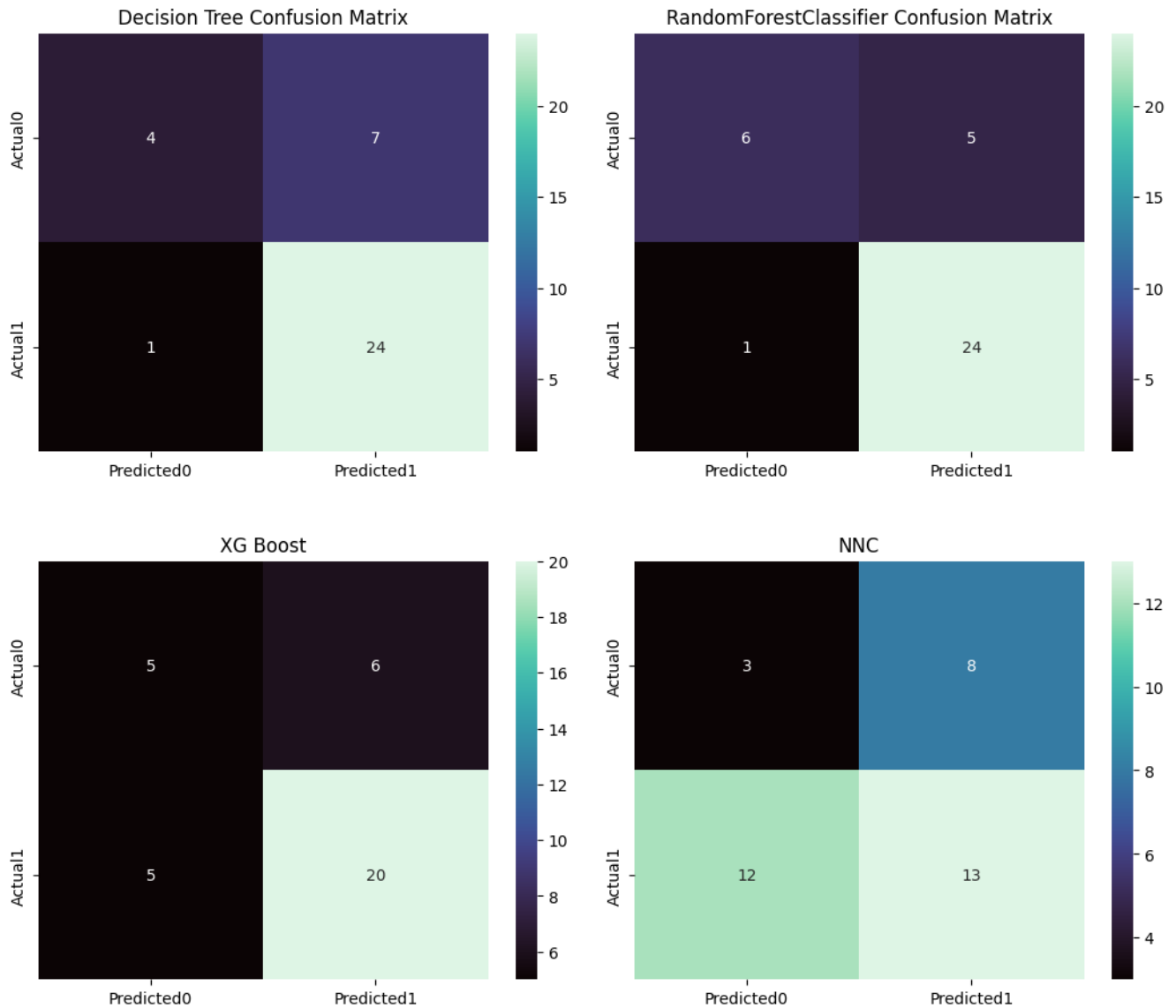


Fig 5.1: Confusion Matrix result

Here, we are showing 4 Confusion matrix results between Random Forest, Decision Tree, NNC, XG Boost.

In the Confusion matrix we can denote when 4 of these models have predicted wrong when the actual answer was 0 and when the models have predicted the right answer for actual answer was 0. Similarly, it has the tendency of higher ratio of predicting correct answer when the actual answer was 1 and vice-versa. In the confusion matrix for the random forest, we can observe that out of a total of 7 loan denial records,

5 were incorrectly predicted as accepted (FP). While, 25 actual acceptance records only 1 was predicted wrong. Moreover, out of a total of 11 denial records, 7 were incorrectly predicted as denials (FN). However, out of 25 acceptance records only 1 was predicted wrong. Similarly for NNC the correct prediction rate is less for both types of records, whereas, XG Boost could predict more accurately whenever loan was accepted.

Thus in comparison we discovered that Random Forest Classifier had the highest accuracy rate thus we decided to implement this in our custom made website.

Below is a table of all seven models prediction score for everyone's convenience:

Model Performance

<b>Model Name</b>	<b>Accuracy</b>	<b>F1-Score</b>	<b>AUC</b>
Random Forest	83.33	88.89	75.27
Decision Tree	82.78	86.87	73.83
KNN	77.78	85.71	66.88
Logistic Regression	81.81	88.89	72.73
XG Boost	81.25	88.0	69.94
NNC	81.48	86.52	72.57
SVM	69.44	81.67	50.12

It's vital to keep in mind that having a low false negative rate is not the only element to take into account; the model's overall accuracy, precision, recall, and other metrics like F1-score, and AUC, should also be taken into account to evaluate the model's performance.

The resulting Accuracy, F1, and AUC scores from each method are shown in figure 5.2 in percentage form in the aforementioned table of model performance, allowing for easy comparison between them. It appears that all seven models have high accuracy and F1-score, which indicates that all models are doing a good job of classifying the data. However, looking at the numbers in the table, it's clear that the Random Forest Classifier algorithm outperformed the other classifiers. The ml model has shown tremendous potential by exceeding all other methods on almost every measures of the matrix. It provides the highest accuracy and F1 scores among the models given. The Random forest classifier algorithm also has the highest AUC score among the models.

Since F1 score is a measure of a test's correctness that takes into account both the precision and the recall of the test, we will place more emphasis on F1 score than accuracy. It is precise and recalls harmonic mean. As a result, the lower precision or recall rating is given greater weight. This is helpful when precision and recall need to be balanced, such as when the costs of false positives and false negatives are considerably different. Contrarily, accuracy only takes into account the percentage of real outcomes (both true positives and true negatives) relative to the overall

number of cases.

In a bank loan prediction dataset, high accuracy and F1-score are critical, but having a high AUC is also significant since it suggests that the model is doing a good job distinguishing between intrusion and non-intrusion situations. The AUC measures a model's ability to differentiate between positive and negative classes. Because it is unaffected by the threshold used to transform predicted probability into class labels, it provides a trustworthy metric of model performance. Another metric for summarizing a classifier's performance over all possible categorization levels is AUC. As a result, based on the data supplied, Random Forest and random forest models may be a better solution for bank loan prediction challenges.

The ROC curve graphically depicts the performance of a binary classifier system as the discrimination threshold is modified. The ROC curve plots the true positive rate (TPR) vs the false positive rate (FPR) at various threshold values.

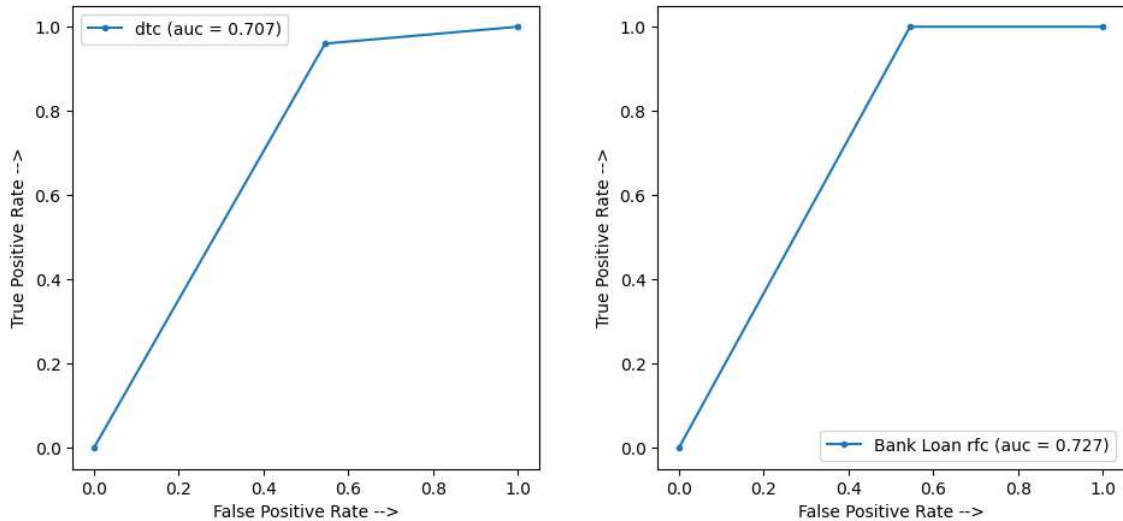


Fig 5.3: ROC curve of Decision tree and Random Forest

From Figure 5.3 5.4, we can see that the ROC curve of the Neural Code Comprehension indicates that the classifier has a high ability to distinguish between positive and negative classes. In this case, the AUC of 0.76 for Logistic Regression falls short to 0.68, whereas for XG Boost 0.69 where both these lr and xg boost performs almost same, which means that all three of the classifier are almost able to correctly classify a high proportion of positive and negative cases. The ROC curve is likely to be close to the top-left corner of the plot, which represents a high true positive rate (TPR) and a low false positive rate (FPR) at various threshold settings. It means that the Random Forest is able to identify most of the positive cases as positive and most of the negative cases as negative. With the exception of Random forest which is the closest to it. Finally, all of the models have produced a nearly ideal ROC curve. Again, the best ROC curve was produced by Random Forest.

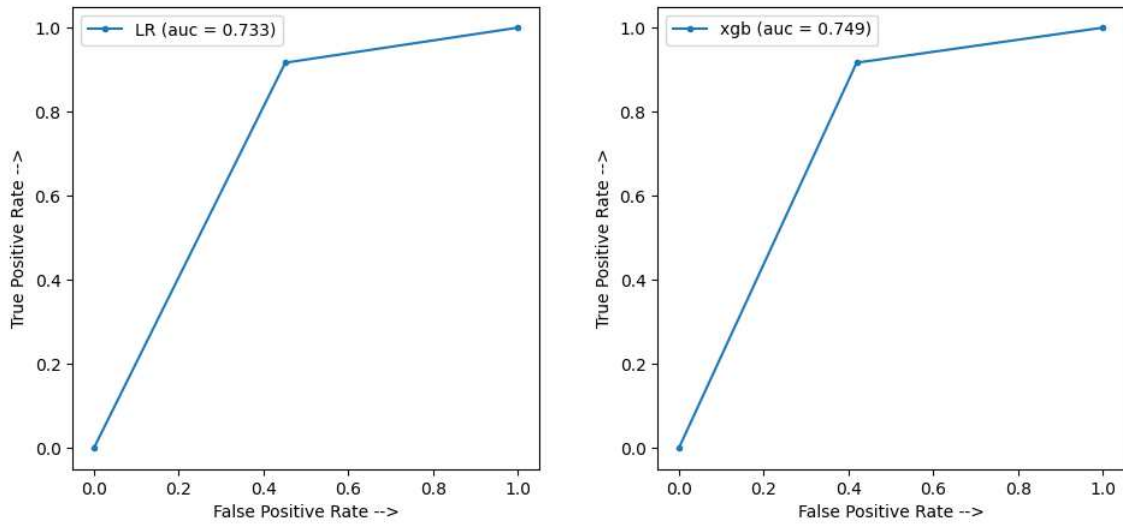


Fig 5.4: ROC curve of Logistic Regression and XGBoost

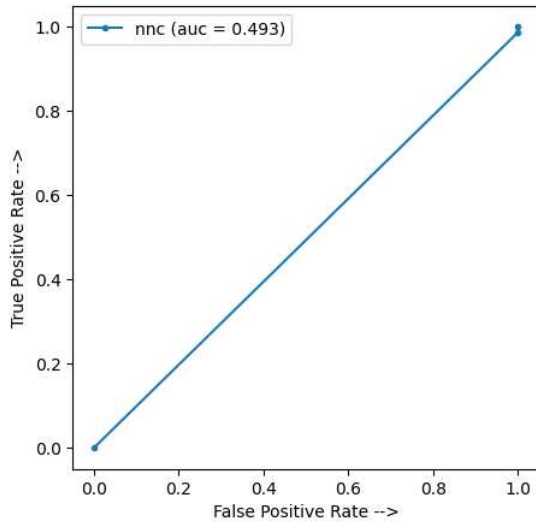


Fig 5.4: ROC curve of NNC

If we now check the comparison among all these models we can be sure of which one was the best classifier for the problem of this particular data set. In the below figure 5.5 the random classifier achieves the highest score in all three departments of scoring deviation. It tops all the divisions with highest possible score which is non-rival to rest of the six models. It depicts if the Random Forest Classifier is applied in a website while using the model it will achieve correct prediction almost every time. Though the scores aren't above 90 yet but we hope to achieve it soon with more careful investigations and better dataset. For now we proceed to work with the Random forest and finally, implement it in a website for practical use.

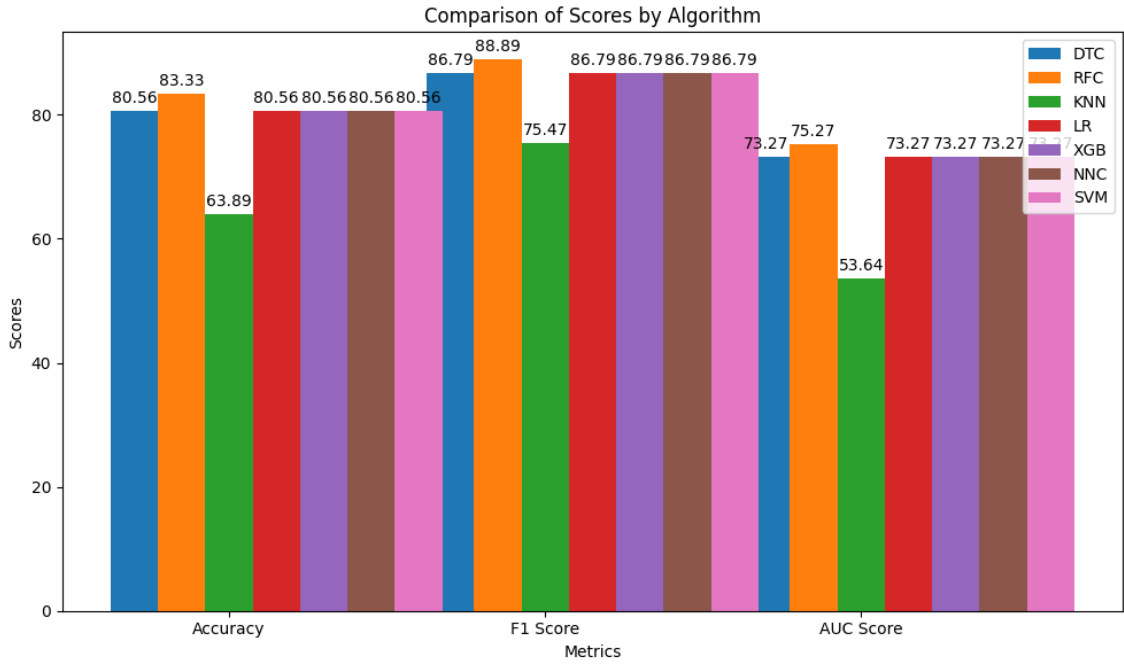


Fig 5.5: All Scores

### 5.3 XAI Observations

We ran LIME on all of our machine-learning models before creating correlation bar charts for 50 observations. It is vital to remember that LIME only displays the ten most correlated attributes each observation by default. There were three observations: the actual value, the anticipated value, and the residue. The residue represented the discrepancy between the actual and expected values. So, if the actual value is 1 and the anticipated value is 1, the residual is 0, and vice versa. The features on the left (colored blue) dispute the forecast, while the features on the right (colored orange) approve the prognosis.

We selected a subset of observations and then examined many cases when the model correctly predicted 1 and 0. Then, for those situations, we looked at the ten most connected feature correlations. We looked at several instances of correct observations and looked for biased features (features whose correlation changed little to none even when the output changed). Some of the lime observations for each model are listed below.

### 5.3.1 LIME (Random Forest)

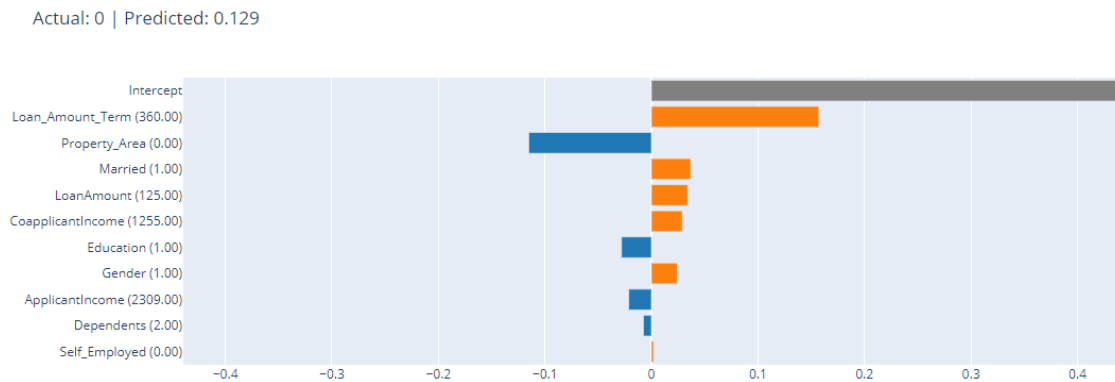


Fig 5.6: Random Forest 0 observation

Actual: 1 | Predicted: 0.774

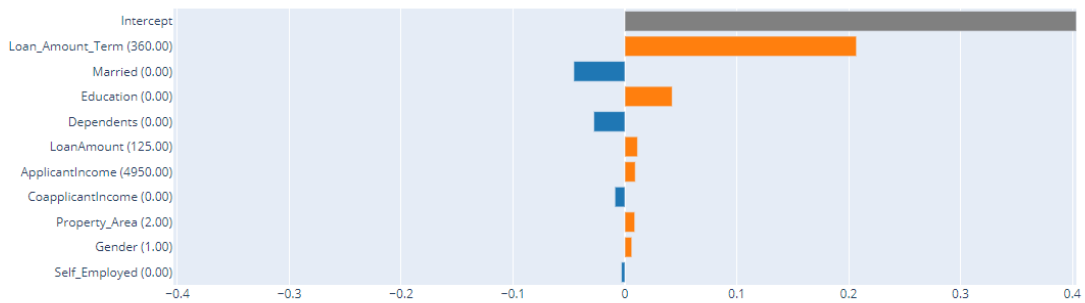


Fig 5.7: Random Forest 1 observation

In figure 5.6 for the RF model, we can see all the features except the feature Loan\_amount\_term, gender and married are all on the right side of the prediction, indicating a positive correlation.

But in figure 5.7, when the prediction is 1, it is important to note that some of the features such as 'Education, Self-employed, Loan\_amount, Dependents' are now on the left side, further confirming that it is negatively correlated with the predicted output 1.

To summarize, a XAI such as LIME cannot provide a complete picture of how each feature interacts with each model because it can only interpret local data. The graphs above, however, clearly show some patterns among some traits. It's worth noting that LIME only shows the top ten most linked attributes for each observation. As a result, if some features were present while the prediction was 1, the same features will not always be present when observing the features when the prediction is 0.

# Chapter 6

## Conclusion

Banks are the most important financial component of any country. Machine Learning can be incredibly useful in predicting loans at banks. People take loans to build their future and banks make their profits through these loans. AI has been proven beneficial in terms of loan prediction.. It can be done easily by building a ML Model. The methods and techniques of Machine Learning to build AI models are favourable enough to predict bank loans. Much research has been done into this. Machine Learning methods and algorithms can help the corporate and financial world.[3]

In our paper, we have implemented seven Machine Learning Models such as Decision tree, Random forest, XGBoost, SVM, NNC etc. Our goal was to make sure that the users get the maximum benefit from our website. To ensure this we trained all the seven models and came out with the output about which model give the maximum output. After executing all the models, we learned that Random Forest Classifier shows the highest accuracy rate. Hence, we implemented this in our website to get the maximum accurate result in this loan prediction process. As our main goal for our project is to obtain the most accurate result to find eligibility for taking loans. Hence, using random forest model in our website we found the most accurate result.

In general, the methods and algorithms used in research might be a path to a successful financial condition for both banks and borrowers in order to upgrade the whole loan prediction system[2].

# Bibliography

- [1] M. T. Ribeiro, S. Singh, and C. Guestrin, *Introduction to local interpretable model-agnostic explanations (lime) a technique to explain the predictions of any machine learning classifier*, 2016.
- [2] M. A. Sheikh, A. K. Goel, and T. Kumar, “An approach for prediction of loan approval using machine learning algorithm,” in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2020, pp. 490–494. DOI: 10.1109/ICESC48915.2020.9155614.
- [3] S. M. Fati, “Machine learning-based prediction model for loan status approval,” *Journal of Hunan University Natural Sciences*, vol. 48, no. 10, 2021.
- [4] A. Gramegna and P. Giudici, “Shap and lime: An evaluation of discriminative power in credit risk,” *Frontiers in Artificial Intelligence*, vol. 4, p. 752 558, 2021.
- [5] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, “Explainable artificial intelligence (xai) to enhance trust management in intrusion detection systems using decision tree model,” *Complexity*, vol. 2021, 2021.
- [6] G. B. Rath, D. Das, and B. Acharya, “Modern approach for loan sanctioning in banks using machine learning,” in *Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI 2019*, Springer, 2021, pp. 179–188.
- [7] A. Bhattacharya, *Applied Machine Learning Explainability Techniques: Make ML models explainable and trustworthy for practical applications using LIME, SHAP, and more*. Packt Publishing Ltd, 2022.
- [8] J. Petch, S. Di, and W. Nelson, “Opening the black box: The promise and limitations of explainable machine learning in cardiology,” *Canadian Journal of Cardiology*, vol. 38, no. 2, pp. 204–213, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0828282X21007030>.
- [9] T. Rahul, Y. S. Deepika, T. V. Vivek, and M. Shereesha, “Bank loan prediction using machine learning,”