

GOA UNIVERSITY
Taleigao Plateau, Goa-403206



PROJECT REPORT ON
"CREDIT SCORING MODELS"
submitted in partial fulfillment of the requirement for the award of
BACHELOR OF ENGINEERING
In
Electronics and Telecommunication Engineering

Submitted by
Shubham Naik (P. R. No. - 201907581)
Sagar Singh (P. R. No. - 201908435)
Arya Karapurkar (P. R. No. - 201907592)
Fouziya Nagarchi (P. R. No. - 201907590)

Under the Guidance of
Mrs. Rohini H. Korti
Assistant Professor, ETC Department



Department of Electronics and Telecommunication Engineering
Padre Conceicao College of Engineering, Goa-403722
2022-23

GOA UNIVERSITY
Taleigao Plateau, Goa-403206



Certificate

Certified that the project work entitled "**CREDIT SCORING MODELS**" is a bona-fide work carried out by **SHUBHAM NAIK , SAGAR SINGH , ARYA KARAPURKAR ,FOUZIYA NAGARCHI** , towards the partial fulfillment for the award of Bachelor of Engineering in Electronics and Telecommunication Engineering at **Padre Conceicao College of Engineering, Verna, Goa** during the year 2022-2023.

The results contained in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Mrs. Rohini H. Korti
(Project Guide)

Dr. Jayalaxmi Devate
(Head of the Department)

Dr. Mahesh Parappagoudar
(Principal)

External Viva

Name of the Examiners:

Signature with date

ACKNOWLEDGMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done is only due to such supervision and assistance and we will not forget to thank them.

We express our gratitude and respect to Dr. Mahesh Parappagoudar, Principal, PCCE, Verna for granting permission to carry out the project.

We express our heartfelt gratitude to Dr. Jayalaxmi Devate, Head of the Department, Department of Electronics and Telecommunication Engineering, PCCE, Verna for being supportive for our work throughout the project.

We would also like to express our heartfelt gratitude to Mrs. Rohini H. Korti, Assistant Professor and Mr. Satish Gangavati, Assistant Professor for guiding us throughout this project and for being supportive for our work throughout the project.

ABSTRACT

With the improving banking sector in recent times and the increasing trend of taking loans, a large population applies for bank loans. But one of the major problem banking sectors face in this ever- changing economy is the increasing rate of loan defaults, and the banking authorities are finding it more difficult to correctly assess loan requests and tackle the risks of people defaulting on loans. In the past, banks used to hire highly professional individuals whose sole purpose was to evaluate applicants and after close review decide and tell whether a candidate was eligible for receiving a loan. The worthiness of a candidate for loan approval or rejection was based on a numerical score called ‘Credit Score’. Generally, the credit score helps the authorities to compute the probability of borrower repaying the loan by the designated time based on their credit history or payment history along with their background. The two most critical questions in the banking industry are (i) How risky is the borrower? and (ii) Given the borrower’s risk, should we lend him/her? In light of the given problems, this paper proposes four machine learning models to predict whether an individual should be given a loan by assessing certain attributes and therefore help the banking authorities by easing their process of selecting suitable people from a given list of candidates who applied for a loan. This paper does a comprehensive and comparative analysis between four algorithms i) Decision tree ii) Random Forest iii)Logistics Regression iv)MLP-NN and tests their accuracy. In order to test their accuracy, we have used the data set from GitHub. Dataset was divided into two parts along with percentage of the training data 80 percent and the testing data 20 percent .

Contents

1	Introduction	1
1.1	Project Statement	1
1.2	Objective of the Project	1
1.3	Organization of the report	2
2	Literature Survey	3
2.1	Research Papers	3
2.2	Summary	7
3	Methodology	8
3.1	Description	8
3.2	Design/Algorithm	16
4	The Dataset	23
4.1	Collected dataset	23
5	Results	26
5.1	Conclusion	35
5.2	Future Scope	35

List of Figures

3.1	Flowchart	9
3.2	Logistic regression	17
4.1	Attributes	23
5.1	(a) Gender Status (b) Dependents Status	27
5.2	(a) Self employed Status (b) Married Status	27
5.3	Education Status	28
5.4	(a) Applicant income status (b)Coapplicant income status	29
5.5	(a)Loan Amount status (b) Credit History status	29
5.6	(a) Applicant income log status (b)Loan amount log status	30
5.7	(a) Loan amount term log status (b) Total income log status	30
5.8	Random Forest : (a) Confusion Matrix	31
5.9	Random Forest: (a) ROC curve	31
5.10	Decision Tree: (a) Confusion Matrix	32
5.11	Decision Tree: (a) ROC Curve	32
5.12	Logistics Regression : (a) Confusion Matrix	33
5.13	Logistics Regression : (a) ROC Curve	33
5.14	MLP NN: (a) Confusion Matrix	34
5.15	MLP NN:(a) ROC Curve	34

Chapter 1

Introduction

1.1 Project Statement

Due to the ever changing economy and ever-increasing competition in the financial world, the activity of taking a loan has become inevitable. Also, small scale to large scale banking firms depend on the activity of lending out loans to earn profits for managing their affairs and to function smoothly at times of financial constraints. In the traditional lending process, banking authorities mainly adopt the '5C principle', i.e., Character, Capital, Capacity, Collateral, and Conditions to evaluate a borrower. This evaluation mainly relied on personal experience and knowledge of customer dealing. This method has great limitations. In order to avoid making biased decisions by loan officers in providing the loans and to overcome the disadvantages of the traditional methods we build a model which will predict the loan lending process.

1.2 Objective of the Project

The goal of our project is :

- 1) To apply different machine learning models in the loan lending process.
- 2) To work out the best approach for a financial institution which accurately identifies whom to lend loan to and help banks identify the loan defaulters for much-reduced credit risk.
- 3) To help the cooperative and public banks to accelerate digital transformations.
- 4) To create a website where in we adopted the Random Forest algorithm as it has the highest accuracy.

1.3 Organization of the report

This report is divided into 6 chapters. In the first chapter we have looked at the problem statement, the objective and organisation of the report. The objective of our project is to build a model that is precise and accurate and will help the co-operative and public banks to accelerate digital transformations.

In the second chapter , we mentioned about research papers we went through. Research papers helped us in understanding the existing technology and its various drawbacks.

In the third chapter we mentioned about the methodology we have followed for the completion of the project. The methodology we followed was data collection, EDA , Data pre-processing and finally setting parameters to the models. The models used were Decision tree, Random Forest, Logistic Regression and NN. The above 4 models were evaluated and compared and the model with the highest accuracy was adopted as the best model for our project.

In the fourth chapter we have wrote about the data set. We collected our data set from GitHub which has 20 attributes out of which we selected the best 12 attributes. There were 6000 tuples(data) which was divided as training and testing data out of which 80 percent was training data and 20 percent was testing.

The fifth chapter includes the Results wherein we adopted the model with the highest accuracy. We have also briefed about the implementation and future scope of this project in this chapter and concluded it.

Chapter 2

Literature Survey

2.1 Research Papers

In this section we explore the previous studies which have been conducted in credit scoring. We briefly look at some of the methods they investigated and the results they got. This will enable us to see the academic progression of the subject and also give context to our study and the comparison of our results thereof.

Reference	Tools and Technologies	Approaches and Algorithms	Application
N H Putri et al Journal of Physics: Conference Series 1836 (2021)012039 doi:10.1088/1742-6596/1836/1/012039 IOP Publishing “Credit Risk analysis using support vector machines algorithm”	Technology: Python. Langrange Multiplier.	Approach: 3 stages I)Data preprocessing- Data cleaning, data type checking and data normalization. II)Model Building Algorithm: SVM SVM uses Kernel trick. Four types of Kernel functions are used: 1) Linear 2) Polynomial 3) Gaussian/Radial Basis Function (RBF) 4) Sigmoid III) Model Evaluation/Testing: Uses a confusion matrix, namely the values of sensitivity, specificity, precision, FPR, FNR, F1-score, ROC and AUC to see the model's performance.	SVM can cope with some problems dealing with financial, medical field and also banking.

2007 International Conference on Computational Intelligence and Security Bo Wang ¹ , Yongkui Liu ¹ , Yanyou Hao ² , Shuang Liu ¹ ¹ College of Computer Science and Engineering, Dalian Nationalities University, China ² Dalian Branch of China Construction Bank, China “Defaults Assessment of Mortgage Loan with Rough Set and SVM” []]	Python	Approach: I)Attribute reduction algorithm Pretreatment Data Reduction Get the knowledge rules II) SMO algorithm III)Search best parameters-Grid search method use cross validation. IV)Algorithm: SVM V) Experiment and analysis	Banking
---	--------	--	---------

I.J Modern Education and Computer Science,2018,5,9,9-16. Published Online May 2018 in MECS DOI:10.5815/IJMECS.2018.05.02 “Credit Risk Prediction using ANN Algorithm” Deepak Kumar and Shruti Goyal.	Python	Approach 1)About data source-Data has been collected from kaggle.com and it will consist of dependent and independent variables. 2)Model: Feed Forward NN is used. Network is trained using supervised learning algorithm. Data Normalization is performed using Min and Max linear transformation function and follows the black box approach. 3)Testing Algorithm	Bond rating, rating short term investments that can last up to 1-year, long term and short-term ratings of local and foreign currencies, sovereign or country ratings.
--	--------	--	--

2017 International Conference on Computational Science and Computational Intelligence. “An Improved Credit Scoring Model A Naïve Bayesian Approach”	GUI-Graphical User Interface MATLAB	Approach: I)The Variables II)The Construction of the improved credit scoring model. Process Flow 1)Data retrieval 2)Data preparation, data transformation, noise removal, normalization. 3)Fit the Bayesian model. 4)validate the model 5)examine and update fit until satisfied. 6)fitted model for prediction. III)Model Validation and Updates. Algorithm: Naïve Bayesian Algorithm works on the Bayesian theorem.	Banking
--	--	--	---------

IOP Conf. Series: Materials Science and Engineering 1022 (2021) 012042 IOP Publishing doi:10.1088/1757-899X/1022/1/012042 “Loan default prediction using decision trees and random forest: A comparative study”	Python	Approach: -To import the necessary libraries and data files required for the model. And the second step was to do an exploratory data analysis (EDA) of the given data to examine its features. -Data Cleaning -EDA -Modelling a) Decision tree b) Random Forest	Banks
--	--------	--	-------

2017 International Conference on Soft Computing, Intelligent System and Information Technology. “Credit Scoring Refinement Using Optimized Logistic Regression” Hendri Sutrisno, Siana Halim Industrial Engineering Department Petra Christian University Surabaya, Indonesia.	Python	Algorithm: Logistic Regression AUC Optimisation- Nelder Mead Algorithm.	Banking
--	--------	--	---------

“PREDICTIVE AND PROBABILISTIC APPROACH USING LOGISTIC REGRESSION: APPLICATION TO PREDICTION OF LOAN APPROVAL” IEEE – 40222 8th ICCCNT 2017 July 3-5, 2017, IIT Delhi, Delhi, India.	Python	-Describing the data - Mathematical representation of Logistic Regression. - Distributional and Categorical Variable Analysis. - Data Munging a) Handling the missing values. b) Handling the extreme values. - Fitting data into model	-Logistic regression is widely used in data analytics were analyzing of the preexisting data within all kinds of organization is required. This helps in economic growth of the organization as predictions can be made about policies to be made in the future based on preexisting policies. -Banks
---	--------	---	--

<p>INTELLIGENCE FOR CREDIT RISK ASSESSMENT: ARTIFICIAL NEURAL NETWORK AND SUPPORT VECTOR MACHINES</p> <p>Khemakhem, Sihem and Boujelbene, Younes “Artificial Intelligence for Credit Risk Assessment: Artificial Neural Network and Support Vector Machines” ACRN Oxford Journal of Finance and Risk Perspectives 6.2 (2017): 1-17.</p>	<p>MATLAB</p>	<p>database: sample and variables.</p> <ul style="list-style-type: none"> -Study of linear dependencies between variables using correlation matrix. -Data Preparation -Training and testing samples -The data mining techniques in credit scoring <ul style="list-style-type: none"> a) Logistic Regression b) ANN c)SVM -Comparing several performance evaluation metrics. <ul style="list-style-type: none"> a) The confusion matrix b) ROC 	
---	---------------	---	--

2.2 Summary

In conclusion, the first two chapters successfully presented the problem statement, objectives and relevant research papers, setting the foundation for this project. The problem statement highlighted the issues that this project aims to address. The inclusion of research papers demonstrated a comprehensive understanding of the existing literature on the subject matter. These papers served to provide a basis for this project with its various methodologies and showcasing the gap between the current knowledge and need for further advancement. Overall, the introduction chapter laid a groundwork for further progress in the project, establishing clear objectives and incorporating relevant sources.

Chapter 3

Methodology

3.1 Description

Here we will give a brief description about our project. The steps are as follows:

- 1)Data Collection and Preparation.**
- 2) Exploratory Data Analysis.**
- 3)Pre-processing.**
- 4)Model Building.**
- 5)Evaluation and Comparison of models.**

Description of the steps: - 1)Data Collection and preparation.

We have collected the dataset from GitHub and prepared it accordingly to meet our goals. So, the training dataset is of 80 percent and testing dataset is of 20 percent. The dataset consists of 43 attributes out of which one is a target attribute. We did the formatting of the raw CSV dataset using both Microsoft Excel and SQL combined. First, we did CSV formatting using Microsoft Excel, replaced the blank values with Null to avoid truncated data warning in SQL, removed thousand separator and then saved it. We decided to limit data rows for 6000 rows due to efficiency reason.

2)EDA-Exploratory Data Analysis

I)Importing new formatted CSV.

II)Descriptive Analysis- is the process of using current and historical data to identify trends and relationships. It is sometimes called the simplest form of data analysis because it describes trends and relationships.

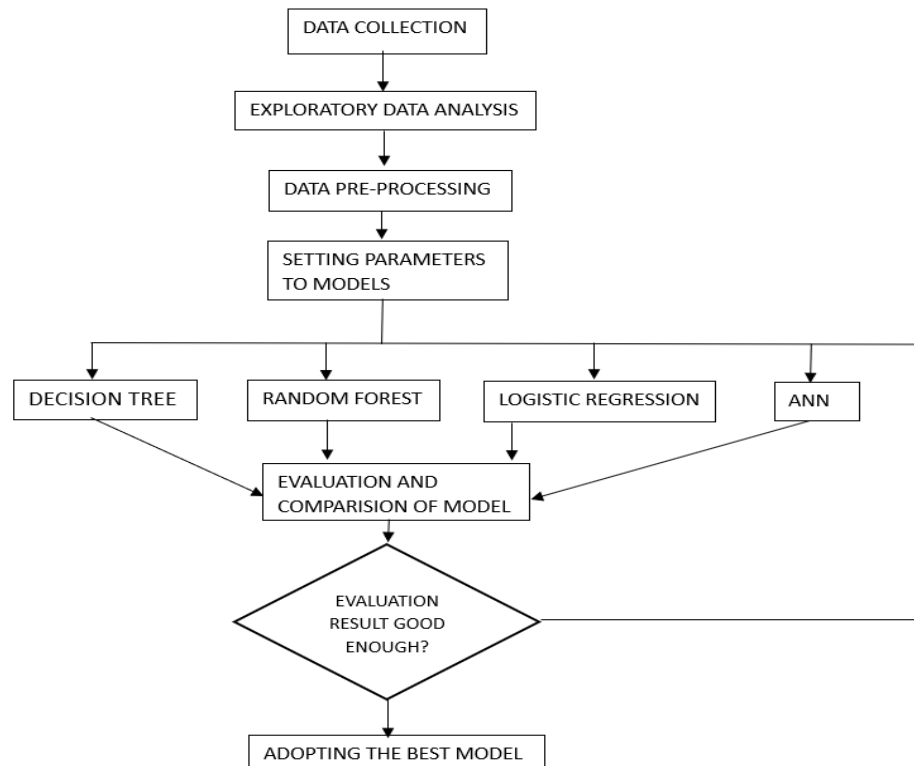


Figure 3.1: Flowchart

III)Customer/ Client Profiling

- 1)Who are our clients (by gender)?
- 2) How many Children that usually our clients have?
- 3) What is the income type that usually our clients have?

IV)Client/Customer Behaviour

- 1) On what days do usually our clients apply the loan?
- 2) On what days did our clients apply for the previous application?
- 3) What income type that usually our clients have who have late payment than X days?

3)Pre-processing

It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific data mining task.

I) Value Encoding-This step is for preparing the dataset to be ready feature selected and modelled.

- a) Income type
- b) Gender
- c) Education background
- d) Family Status
- e) Num of children
- f) Number of dependents
- g) Weekdays applied
- h) Hour applied
- i) Organization type
- j) Contract type in previous application.
- k) Weekdays applied in previous application
- l) Term of Payment
- m) Grouped IR into small, medium and high of the previous application.
- n) Approximately at what hour the client applied for the previous loan -hour applied in previous application.

II) Feature Selection

1. A random forest is made from multiple decision trees (as given by estimators). Each tree individually predicts for the new data and random forest spits out the mean prediction from those trees. The idea for confidence level of predictions is just to see how much predictions coming from different trees are varying for the new observations. Then to analyze further, we can seek some pattern for observations which have highest variability of predictions.
2. In most of the cases, random forests can beat linear models for prediction. An objection frequently raised for random forests is interpretation of results as compared to linear models. But one can address the misconceived objection using the discussed methodologies of interpretation.
3. Feature selection is also known as attribute selection is a process of extracting the most relevant features from the dataset and then applying machine learning algorithms for the better performance of the model. A large number of irrelevant features increases the training time exponentially and increase the risk of overfitting.
4. From the feature selection, we understand that; Credit History is the most important feature in this credit risk modelling as followed by Loan amount term and Applicant income.

4) Model Building

Handling Imbalance Target

- Oversampling using SMOTE
- Undersampling using NearMiss

Handling class imbalance is a common challenge in machine learning when the number of instances in one class significantly outweighs the other class(es). Two popular techniques to address this issue are oversampling and undersampling. Let's discuss two specific methods: oversampling using SMOTE (Synthetic Minority Over-sampling Technique) and undersampling using NearMiss.

Oversampling using SMOTE:

SMOTE is a widely used technique for oversampling the minority class by creating synthetic examples. It works by interpolating new instances between existing minority class instances.

Here's how you can use SMOTE:

- a. Identify the minority class: Determine the class that has fewer instances compared to the other class(es).
- b. Split the dataset: Divide the dataset into features (X) and the target variable (y).
- c. Apply SMOTE: Use the SMOTE algorithm to generate synthetic instances of the minority class. The synthetic instances are created by selecting a minority class instance and finding its k nearest neighbors. New instances are created by interpolating features between the selected instance and its neighbors.
- d. Combine the oversampled data: Combine the original minority class instances with the newly generated synthetic instances. This results in a balanced dataset.
- e. Train the model: Use the balanced dataset to train your machine learning model.

Undersampling using NearMiss:

NearMiss is an undersampling technique that aims to reduce the majority class instances by selecting examples that are most similar to the minority class instances. Here's how you can use NearMiss:

- a. Identify the minority class: Determine the class that has fewer instances compared to the other class(es).
- b. Split the dataset: Divide the dataset into features (X) and the target variable (y).
- c. Apply NearMiss: Use the NearMiss algorithm to select a subset of instances from the majority class. NearMiss selects the majority class instances that are closest to the minority class instances based on distance metrics such as Euclidean distance.
- d. Combine the undersampled data: Combine the selected majority class instances with the original minority class instances. This results in a balanced dataset.
- e. Train the model: Use the balanced dataset to train your machine learning model.

Both SMOTE and NearMiss have their advantages and limitations. SMOTE can introduce synthetic instances that might be outliers or unrealistic, while NearMiss may discard

potentially useful information from the majority class. Additionally, it's worth mentioning that there are other methods available for handling class imbalance, such as ensemble techniques like Random Forest, boosting algorithms like AdaBoost or XGBoost, and cost-sensitive learning. The choice of technique depends on the specific characteristics of your dataset and the goals of your machine learning task.

Logistic Regression before and after Tuning.

Logistic Regression is a popular and widely used classification algorithm that models the relationship between the dependent variable and one or more independent variables using a logistic function. Tuning the logistic regression model involves optimizing its hyperparameters to improve its performance. Let's discuss the process of tuning logistic regression, both before and after tuning.

Before Tuning:

Data Preprocessing: Perform necessary data preprocessing steps, such as handling missing values, encoding categorical variables, and scaling numerical features.

Split the Data: Split the dataset into training and testing sets. Typically, a common split is 70-30 or 80-20, with the majority of the data used for training.

Train the Initial Logistic Regression Model: Fit the logistic regression model to the training data using default hyperparameter values. The model will learn the relationship between the features and the target variable.

Evaluate Performance: Evaluate the performance of the initial logistic regression model using appropriate evaluation metrics such as accuracy, precision, recall, F1 score, or area under the ROC curve (AUC-ROC).

After Tuning:

Identify Hyperparameters: Determine the hyperparameters of the logistic regression model that can be tuned. Some commonly tuned hyperparameters include the regularization parameter (C or lambda), penalty type (L1 or L2), solver algorithm, and class weight balancing.

Define Hyperparameter Search Space: Define a search space for each hyperparameter. For example, specify a range of values or a list of options to explore during the tuning process.

Hyperparameter Tuning Techniques: There are various techniques to tune hyperparameters, such as Grid Search, Random Search, or Bayesian Optimization. These methods involve systematically searching the hyperparameter space and evaluating the model's performance using cross-validation.

Cross-Validation: Perform cross-validation to assess the model's performance with different hyperparameter combinations. This helps in avoiding overfitting and obtaining a more reliable estimate of the model's performance.

Select Best Hyperparameters: Select the hyperparameter combination that yields the best performance based on the chosen evaluation metric.

Retrain the Model: Retrain the logistic regression model using the best hyperparameters on the entire training dataset.

Evaluate Performance: Evaluate the performance of the tuned logistic regression model on the testing set or using cross-validation. Compare the performance metrics with the initial model to assess the improvement.

Iterative Process: If necessary, iterate through steps 3 to 7, adjusting the hyperparameter search space or trying different techniques to further optimize the model's performance.

Tuning the hyperparameters of a logistic regression model can significantly improve its performance by finding the optimal configuration for your specific dataset and problem. The process involves a combination of experimentation, evaluation, and iteration to achieve the best results.

Random Forrest Classifier before and after Tuning.

Random Forest Classifier is a popular ensemble learning algorithm that combines multiple decision trees to make predictions. Tuning the Random Forest model involves optimizing its hyperparameters to improve its performance. Let's discuss the process of tuning the Random Forest Classifier, both before and after tuning.

Before Tuning:

Data Preprocessing: Perform necessary data preprocessing steps, such as handling missing values, encoding categorical variables, and scaling numerical features.

Split the Data: Divide the dataset into training and testing sets. Typically, a common split is 70-30 or 80-20, with the majority of the data used for training.

Train the Initial Random Forest Classifier: Fit the Random Forest Classifier to the training data using default hyperparameter values. The model will create an ensemble of decision trees and learn from the features to make predictions.

Evaluate Performance: Evaluate the performance of the initial Random Forest Classifier using appropriate evaluation metrics such as accuracy, precision, recall, F1 score, or area under the ROC curve (AUC-ROC).

After Tuning:

Identify Hyperparameters: Determine the hyperparameters of the Random Forest Classifier that can be tuned. Some commonly tuned hyperparameters include the number of trees in the forest (n estimators), the maximum depth of each tree (max depth), the minimum number of samples required to split an internal node (min samples split), and the number of features to consider when looking for the best split (max features).

Define Hyperparameter Search Space: Define a search space for each hyperparameter. Specify a range of values or a list of options to explore during the tuning process.

Hyperparameter Tuning Techniques: There are various techniques to tune hyperparameters, such as Grid Search, Random Search, or Bayesian Optimization. These methods involve systematically searching the hyperparameter space and evaluating the model's performance using cross-validation.

Cross-Validation: Perform cross-validation to assess the model's performance with different hyperparameter combinations. This helps in avoiding overfitting and obtaining a more reliable estimate of the model's performance.

Select Best Hyperparameters: Select the hyperparameter combination that yields the best performance based on the chosen evaluation metric.

Retrain the Model: Retrain the Random Forest Classifier using the best hyperparameters on the entire training dataset.

Evaluate Performance: Evaluate the performance of the tuned Random Forest Classifier on

the testing set or using cross-validation. Compare the performance metrics with the initial model to assess the improvement.

Iterative Process: If necessary, iterate through steps 3 to 7, adjusting the hyperparameter search space or trying different techniques to further optimize the model's performance.

Decision Tree Classifier before and after Tuning.

Decision Tree Classifier is a versatile and widely used classification algorithm that builds a binary tree structure to make predictions based on a set of decision rules. Tuning the Decision Tree model involves optimizing its hyperparameters to improve its performance. Let's discuss the process of tuning the Decision Tree Classifier, both before and after tuning.

Before Tuning:

Data Preprocessing: Perform necessary data preprocessing steps, such as handling missing values, encoding categorical variables, and scaling numerical features.

Split the Data: Divide the dataset into training and testing sets. Typically, a common split is 70-30 or 80-20, with the majority of the data used for training.

Train the Initial Decision Tree Classifier: Fit the Decision Tree Classifier to the training data using default hyperparameter values. The model will create a tree structure and learn from the features to make predictions.

Evaluate Performance: Evaluate the performance of the initial Decision Tree Classifier using appropriate evaluation metrics such as accuracy, precision, recall, F1 score, or area under the ROC curve (AUC-ROC).

After Tuning:

Identify Hyperparameters: Determine the hyperparameters of the Decision Tree Classifier that can be tuned. Some commonly tuned hyperparameters include the maximum depth of the tree (maxdepth), the minimum number of samples required to split an internal node (minsamples split), the minimum number of samples required to be at a leaf node (min samples leaf), and the criterion for measuring the quality of a split (criterion).

Define Hyperparameter Search Space: Define a search space for each hyperparameter. Specify a range of values or a list of options to explore during the tuning process.

Hyperparameter Tuning Techniques: There are various techniques to tune hyperparameters, such as Grid Search, Random Search, or Bayesian Optimization. These methods involve

systematically searching the hyperparameter space and evaluating the model's performance using cross-validation.

Cross-Validation: Perform cross-validation to assess the model's performance with different hyperparameter combinations. This helps in avoiding overfitting and obtaining a more reliable estimate of the model's performance.

Select Best Hyperparameters: Select the hyperparameter combination that yields the best performance based on the chosen evaluation metric.

Retrain the Model: Retrain the Decision Tree Classifier using the best hyperparameters on the entire training dataset.

Evaluate Performance: Evaluate the performance of the tuned Decision Tree Classifier on the testing set or using cross-validation. Compare the performance metrics with the initial model to assess the improvement.

Iterative Process: If necessary, iterate through steps 3 to 7, adjusting the hyperparameter search space or trying different techniques to further optimize the model's performance.

5)Evaluation and comparison of the models.

Here we will evaluate the Accuracy, Recall, Precision, ROC AUC Score and F1 score and compare it with all the 4 models.

3.2 Design/Algorithm

1)Logistic Regression:

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. In Logistic regression, instead of fitting a

regression line, we fit an S shaped logistic function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc. Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

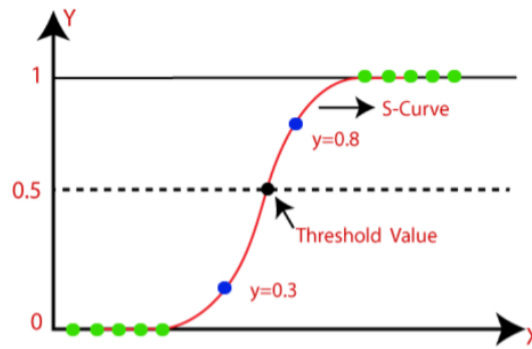


Figure 3.2: Logistic regression

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below: We know the equation of the straight line can be written as:

$$y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n \quad (3.1)$$

In this equation, y represents the dependent variable, b_0 is the intercept term, and b_1, b_2, \dots, b_n are the coefficients corresponding to the independent variables x_1, x_2, \dots respectively. In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by $(1-y)$:

$$\frac{y}{1-y} = \begin{cases} 0 & \text{if } y = 0 \\ \infty & \text{if } y = 1 \\ \frac{y}{1-y} & \text{otherwise} \end{cases} \quad (3.2)$$

In this equation, the fraction $y/1-y$ is evaluated differently based on the values of y . The cases environment is used to specify the different conditions and their corresponding re-

sults. The 0 is shown when y equals 0, the infinity symbol (∞) is shown when y equals 1, and the fraction $y/(1-y)$ is shown for any other value of y .

But we need range between $-\infty$ to $+\infty$, then take logarithm of the equation it will become:

$$\log \left(\frac{y}{1-y} \right) = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \dots + b_nx_n \quad (3.3)$$

2)ANN-Artificial Neural Network. An artificial neural network is a nonlinear approach that provides a new alternative to linear methods, especially in the situations where the dataset possesses complex relationships between the independence of the nonlinear variables. Artificial neural network is a learning system that models a relationship between inputs and outputs, considering the relationship is nonlinear. They are also known as black box systems, in which extraction of information from internal system is impossible. Artificial Neural networks are machine learning system that simulates structure and function like a biological neuron. ANNs (Artificial Neural Networks) perform a task by changing its parameters, the same way a neuron changes its states to perform a cognitive task. A network is composed of a set of neurons structured in a specified topology. Neurons are connected by links with associated weights which determines information flow intensity; weights are the functions that represent behavior of the neural network.

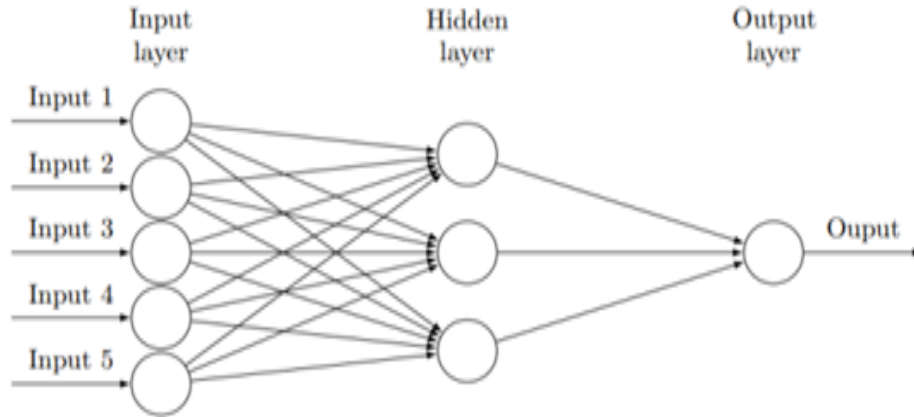


Fig.1. Basic Structure of Artificial Neural Network

Fig. 1, ANN has three layers, input layer, a hidden layer and output layer, input layer rep-

resents neurons receiving input stimulus. Then the information is transferred to next level of layer known as a hidden layer. Information is weighed before sending to next level of layers depending upon the size of the connections amongst neurons. Information is sized as per the processing unit or a transfer function represented in Fig. 2.

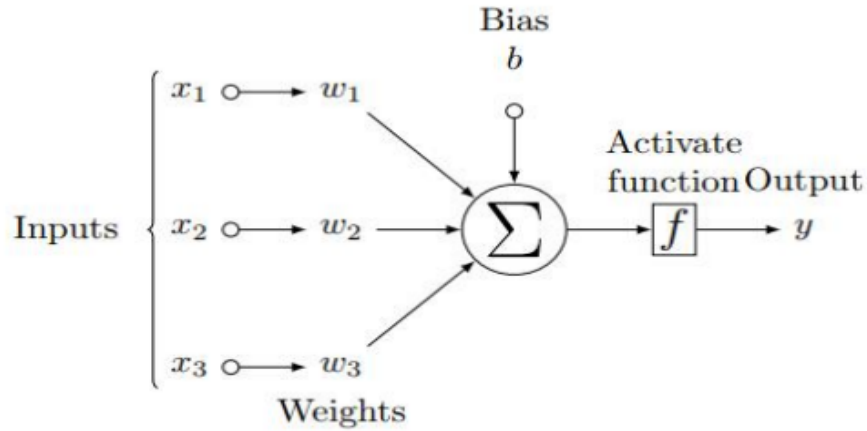


Fig.2. Mathematical Equation of Artificial Neural Network

In fig 2 each neuron is characterized by a minimum value that activates a neuron (threshold value) and a transfer function. A Hidden layer can consist of several layers and performs the summation of input neurons and multiplies the weights with the summation to generate output neurons. Output generation is a two-step process: first, each input is multiplied by the weight on corresponding connection and then all valued are summed together; second, activation function is applied to summation of the inputs.

$$\hat{y} = f \left(\sum_{i=1}^n w_i \cdot x_i + b \right) \quad (3.4)$$

In this equation, y represents the predicted output, f is the activation function, w_i are the weights associated with the input variables x_i , b is the bias term, and n is the number of input variables.

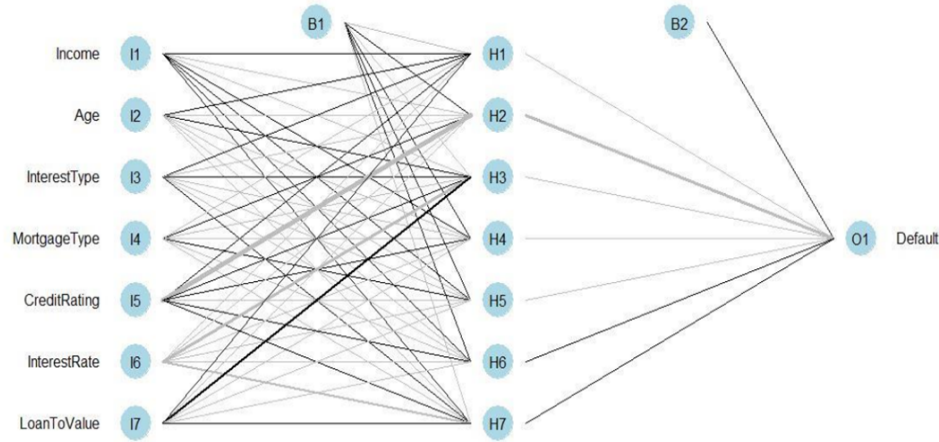
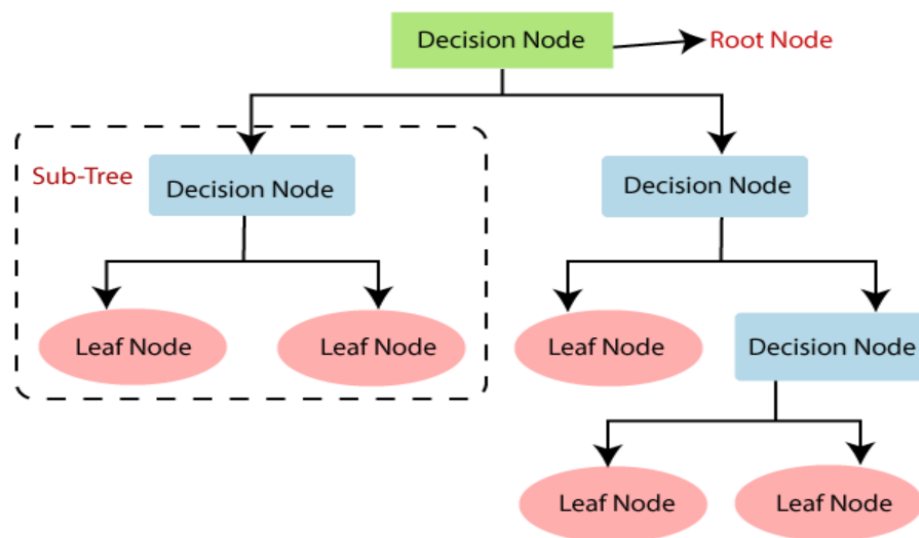


Fig.3. Neural Network Plot of the Credit Default Model.

As (1) represents evaluation of input and (2) represents activation function; where W_{ji} represents Weights on the connection between j and i and a_j is activation function of neuron j . For a neural network to work efficiently, weights should be tuned accurately. This task can be achieved by using a learning algorithm, which trains the network and modifies weights until verified. Mostly, these algorithms stop when there occurs an error between output generated by network falls under threshold and expected output. There are three types of learning algorithms for artificial neural networks i.e., supervised Learning, unsupervised Learning and reinforced Learning. In supervised learning, a training set of correct examples is being used to train the network model. It consists of pairs of several inputs and expected outputs. Weights will be tuned based on the errors generated in the network. Most common example of supervised learning is classification, where the network has to learn to generalize relations between corresponding input and output variables.

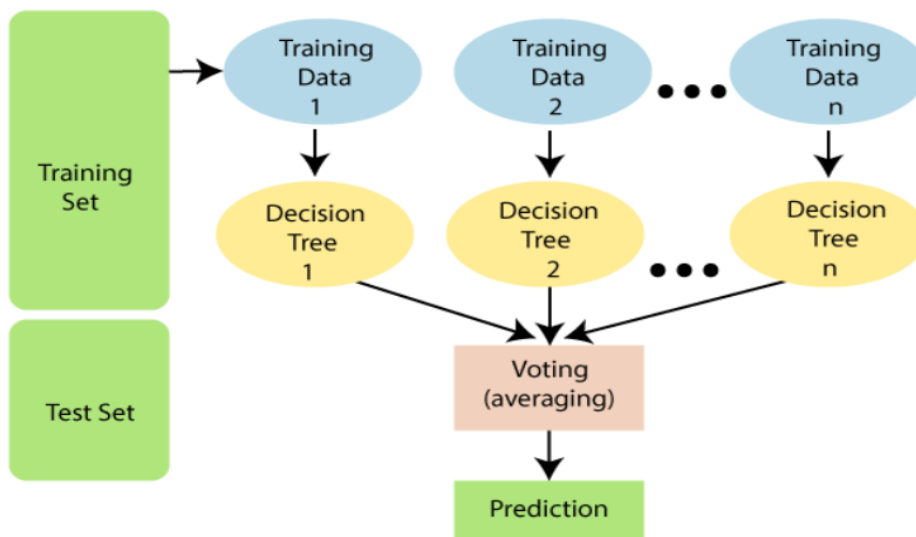
3)Decision tree

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees. Below diagram explains the general structure of a decision tree:



4)Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. As the name suggests, Random Forest is a classifier that contains a number of decisions trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.; Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. The below diagram explains the working of the Random Forest algorithm:

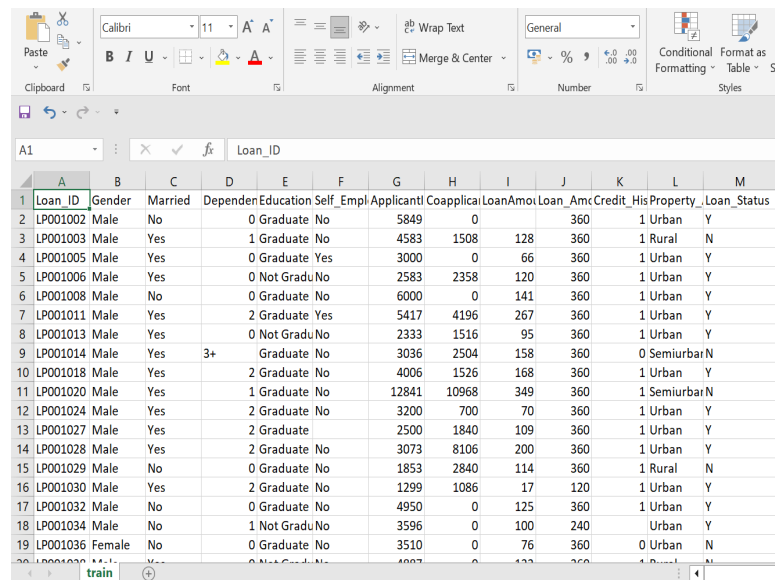


Chapter 4

The Dataset

4.1 Collected dataset

For this project we have used the dataset from GitHub. There were total of 6000 tuples then we divided he as 80 percent training and 20 percent testing.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Loan_ID	Gender	Married	Dependen	Education	Self_Empl	ApplicantI	Coapplica	LoanAmo	Loan_Amc	Credit_His	Property_	Loan_Status
2	LP001002	Male	No	0	Graduate	No	5849	0		360	1	Urban	Y
3	LP001003	Male	Yes	1	Graduate	No	4583	1508	128	360	1	Rural	N
4	LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	360	1	Urban	Y
5	LP001006	Male	Yes	0	Not Gradu	No	2583	2358	120	360	1	Urban	Y
6	LP001008	Male	No	0	Graduate	No	6000	0	141	360	1	Urban	Y
7	LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	360	1	Urban	Y
8	LP001013	Male	Yes	0	Not Gradu	No	2333	1516	95	360	1	Urban	Y
9	LP001014	Male	Yes	3+	Graduate	No	3036	2504	158	360	0	Semiurban	N
10	LP001018	Male	Yes	2	Graduate	No	4006	1526	168	360	1	Urban	Y
11	LP001020	Male	Yes	1	Graduate	No	12841	10968	349	360	1	Semiurban	N
12	LP001024	Male	Yes	2	Graduate	No	3200	700	70	360	1	Urban	Y
13	LP001027	Male	Yes	2	Graduate		2500	1840	109	360	1	Urban	Y
14	LP001028	Male	Yes	2	Graduate	No	3073	8106	200	360	1	Urban	Y
15	LP001029	Male	No	0	Graduate	No	1853	2840	114	360	1	Rural	N
16	LP001030	Male	Yes	2	Graduate	No	1299	1086	17	120	1	Urban	Y
17	LP001032	Male	No	0	Graduate	No	4950	0	125	360	1	Urban	Y
18	LP001034	Male	No	1	Not Gradu	No	3596	0	100	240		Urban	Y
19	LP001036	Female	No	0	Graduate	No	3510	0	76	360	0	Urban	N
20	LP001037	Male	No	0	Not Gradu	No	4887	0	133	360		Urban	N

Figure 4.1: Attributes

Loan ID refers to a unique identifier assigned to a loan application or loan account. It is a unique identification number or code that helps in tracking and managing individual loans within a lending institution or financial organization.

In a dataset, the "gender" variable typically represents the biological or perceived sex of individuals. It is a categorical variable that classifies individuals into distinct groups based on their gender identity or biological sex characteristics.

In a dataset, the variable "married" typically represents a person's marital status. It is a categorical variable that indicates whether an individual is married or not.

"Dependents" in a dataset typically refers to a variable that represents the number of dependents or individuals who rely on a particular person for financial support or care. It is a numerical or categorical variable that provides information about the number or presence of dependents associated with each observation in the dataset.

"Education status" in a dataset typically refers to a variable that indicates an individual's level of education or their current educational attainment. It is a categorical variable that provides information about an individual's educational background or status.

"Self Employed status" in a dataset typically refers to a variable that indicates whether an individual is self-employed or not. It is a categorical variable that provides information about an individual's employment status, specifically whether they work for themselves or are employed by someone else.

"Applicant income" in a dataset typically refers to the financial earnings or income of an individual who is the primary applicant for a loan, financial application, or other relevant context. It represents the income earned by the individual who is applying for a specific financial product or service.

"Co-applicant income" in a dataset typically refers to the financial earnings or income of an individual who is the secondary or joint applicant along with the primary applicant for a loan, financial application, or other relevant context. It represents the income earned by the co-applicant, who is applying together with the primary applicant.

"Loan amount" in a dataset typically refers to the total amount of money requested or granted as a loan to an individual or entity. It represents the principal amount that is borrowed or lent for a specific financial transaction.

"Loan amount term" in a dataset typically refers to the duration or period over which a loan is scheduled to be repaid. It represents the length of time that the borrower has agreed to repay the loan in regular installments.

"Credit history" in a dataset typically refers to a variable that captures an individual's or entity's past record of borrowing and repaying debts. It represents the historical information about an individual's creditworthiness and their previous repayment behavior.

The "credit history" variable can have different representations depending on the dataset and its specific context. It is often categorical or binary, indicating whether an individual has a positive or negative credit history. Common categories include:

1. Positive Credit History: Indicates that the individual has a good or positive credit history, meaning they have a record of making timely payments, meeting financial obligations, and managing their debts responsibly.
2. Negative Credit History: Represents that the individual has a poor or negative credit history, indicating instances of late payments, defaults, bankruptcy, or other adverse credit events.
3. No Credit History: Indicates that the individual does not have a credit history or has limited credit information available, such as individuals who have not borrowed or used credit facilities in the past.

"Property status" in a dataset typically refers to a variable that indicates the ownership or status of a property associated with an individual or entity. It provides information about the current state or legal ownership of a property.

Chapter 5

Results

This section represents the output graphs of all the four Models used. A confusion matrix is a performance measurement tool used in machine learning and statistics to evaluate the performance of a classification model. It provides a summary of the predicted and actual values for a set of data points.

In the confusion matrix:

True Positive (TP): The number of instances correctly predicted as positive.

False Positive (FP): The number of instances incorrectly predicted as positive.

False Negative (FN): The number of instances incorrectly predicted as negative.

True Negative (TN): The number of instances correctly predicted as negative.

The confusion matrix allows us to calculate various evaluation metrics, such as accuracy, precision, recall, and F1-score, which provide insights into the performance of the classification model.

Accuracy: $(TP + TN) / (TP + TN + FP + FN)$

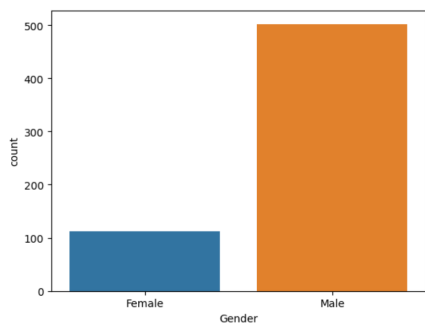
Precision: $TP / (TP + FP)$

Recall (Sensitivity or True Positive Rate): $TP / (TP + FN)$

Specificity (True Negative Rate): $TN / (TN + FP)$

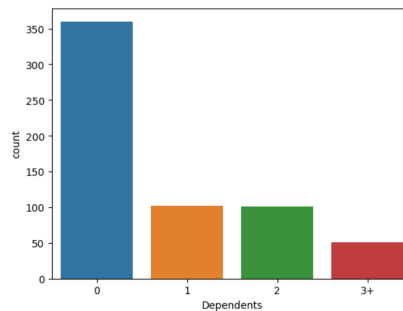
F1-score: $2 * (Precision * Recall) / (Precision + Recall)$

```
Out[18]: <Axes: xlabel='Gender', ylabel='count'>
```



(a)

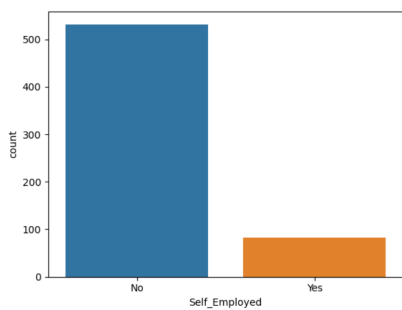
```
Out[20]: <Axes: xlabel='Dependents', ylabel='count'>
```



(b)

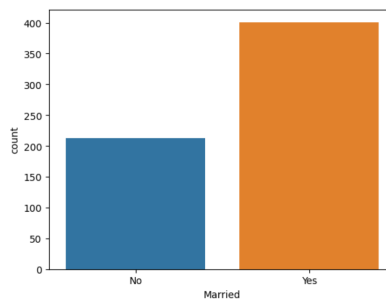
Figure 5.1: (a) Gender Status (b) Dependents Status

```
Out[23]: <Axes: xlabel='Self_Employed', ylabel='count'>
```



(a)

```
Out[21]: <Axes: xlabel='Married', ylabel='count'>
```



(b)

Figure 5.2: (a) Self employed Status (b) Married Status

```
Out[19]: <Axes: xlabel='Education', ylabel='count'>
```

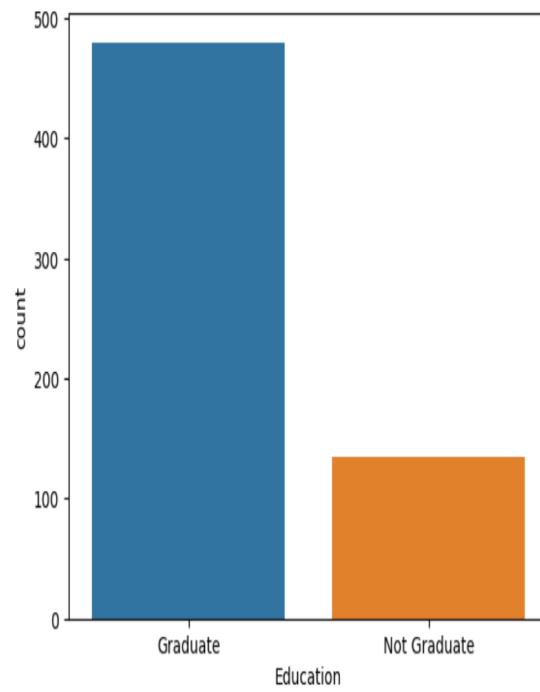


Figure 5.3: Education Status

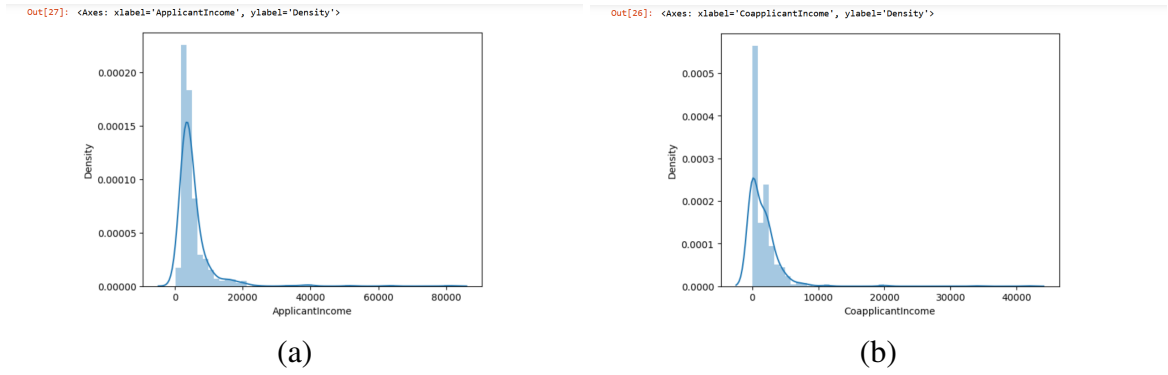


Figure 5.4: (a) Applicant income status (b)Coapplicant income status

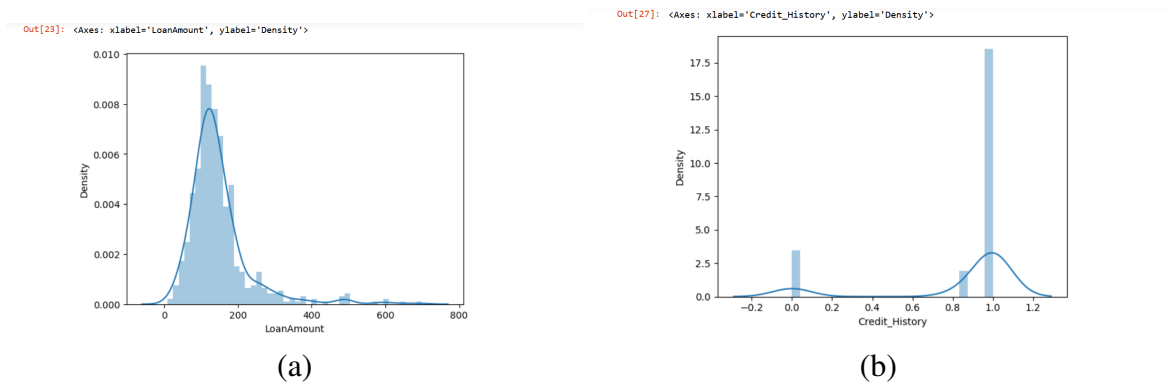


Figure 5.5: (a)Loan Amount status (b) Credit History status

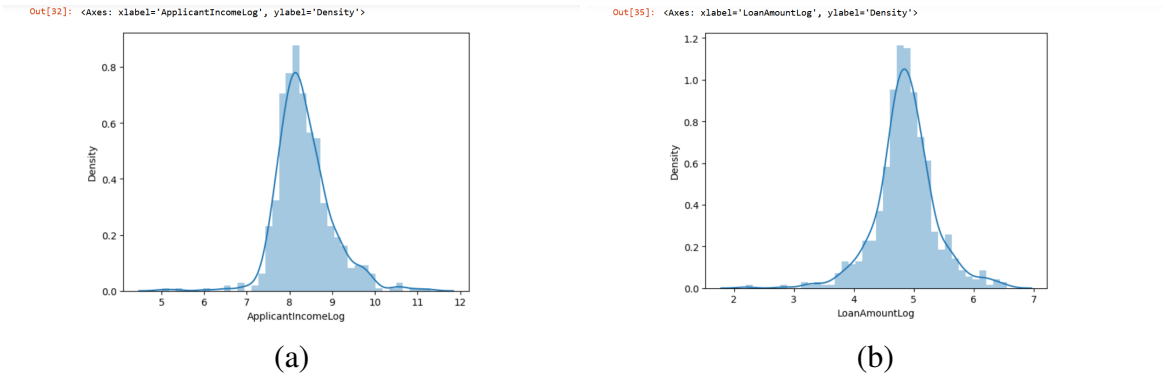


Figure 5.6: (a) Applicant income log status (b) Loan amount log status

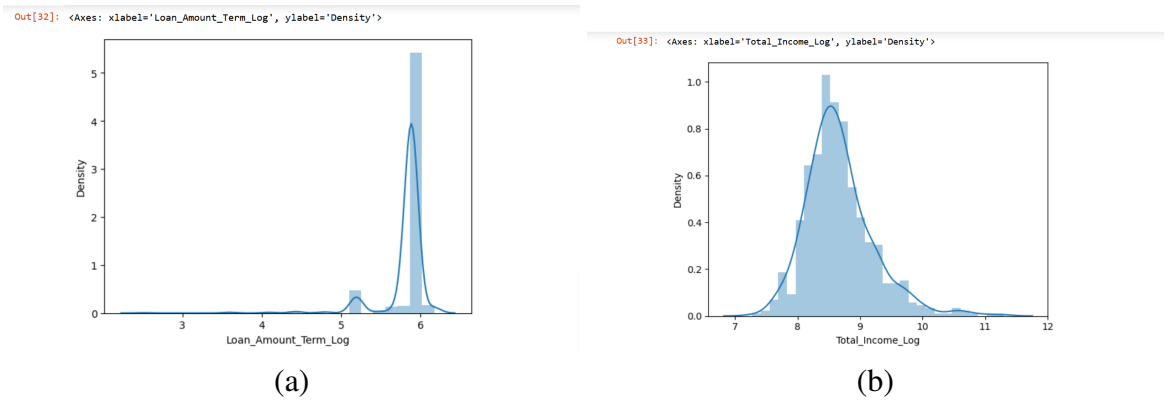


Figure 5.7: (a) Loan amount term log status (b) Total income log status

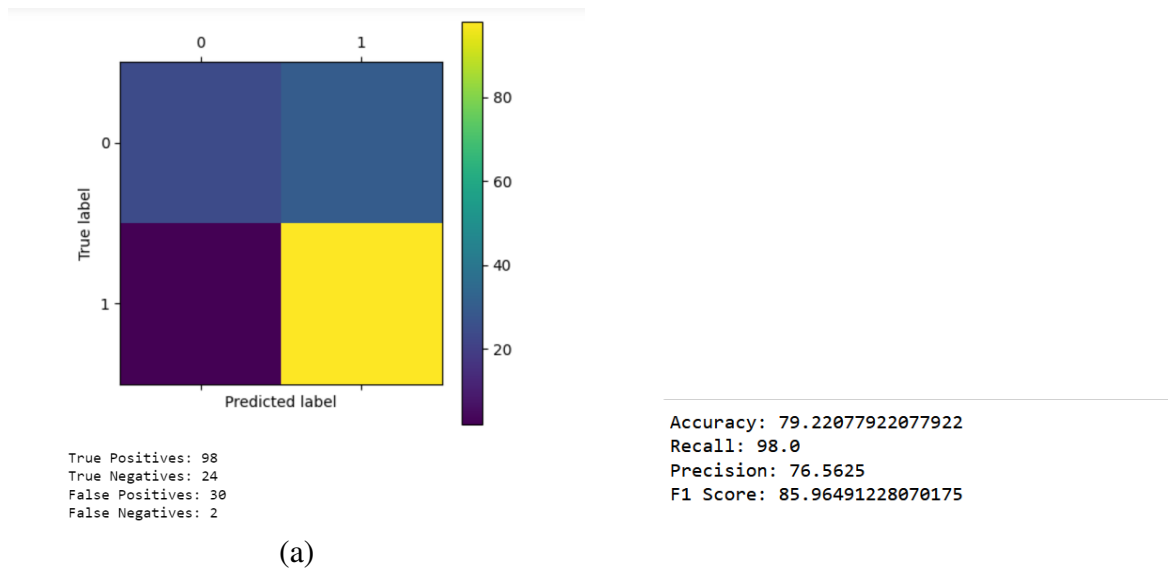
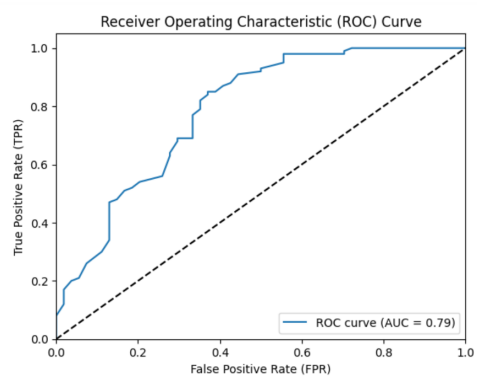


Figure 5.8: Random Forest : (a) Confusion Matrix



(a)

Figure 5.9: Random Forest: (a) ROC curve

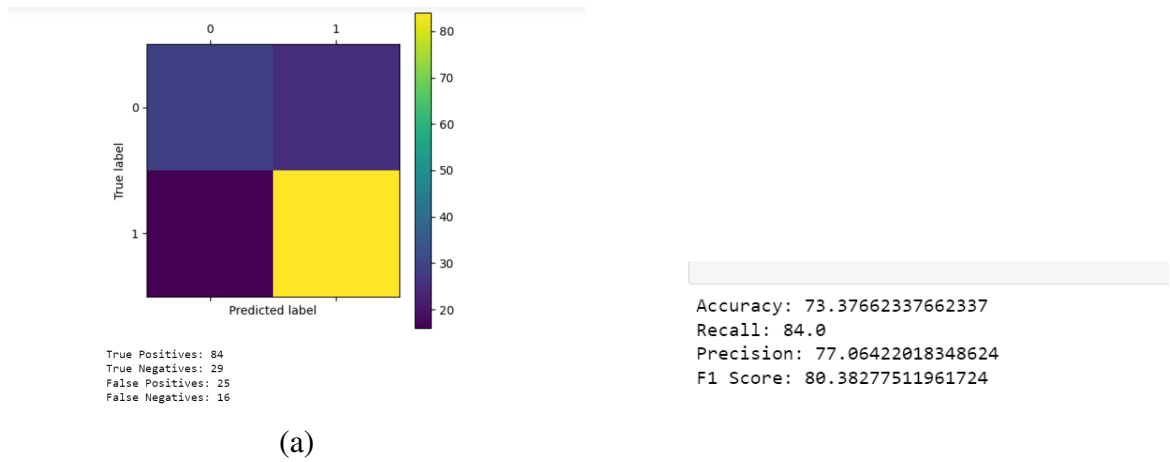


Figure 5.10: Decision Tree: (a) Confusion Matrix

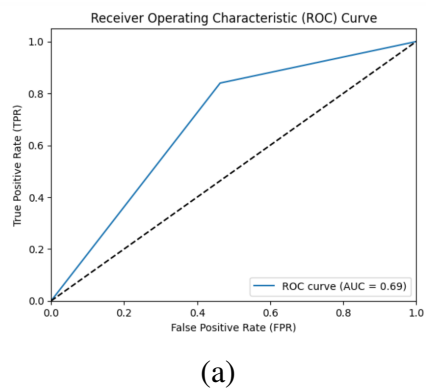


Figure 5.11: Decision Tree: (a) ROC Curve

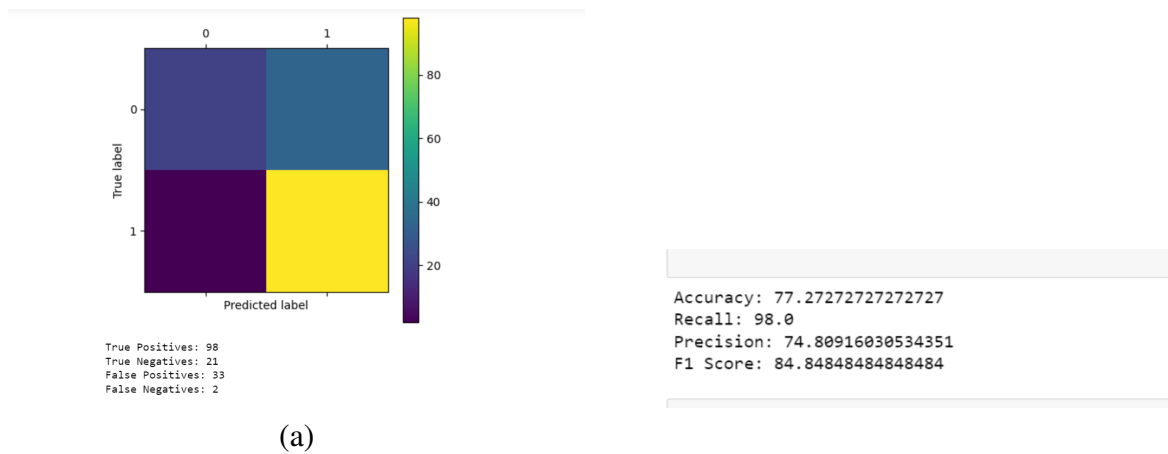


Figure 5.12: Logistics Regression : (a) Confusion Matrix

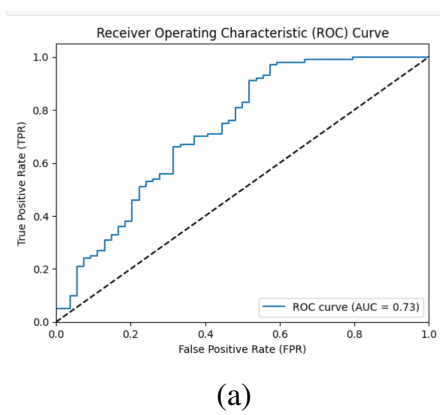


Figure 5.13: Logistics Regression : (a) ROC Curve

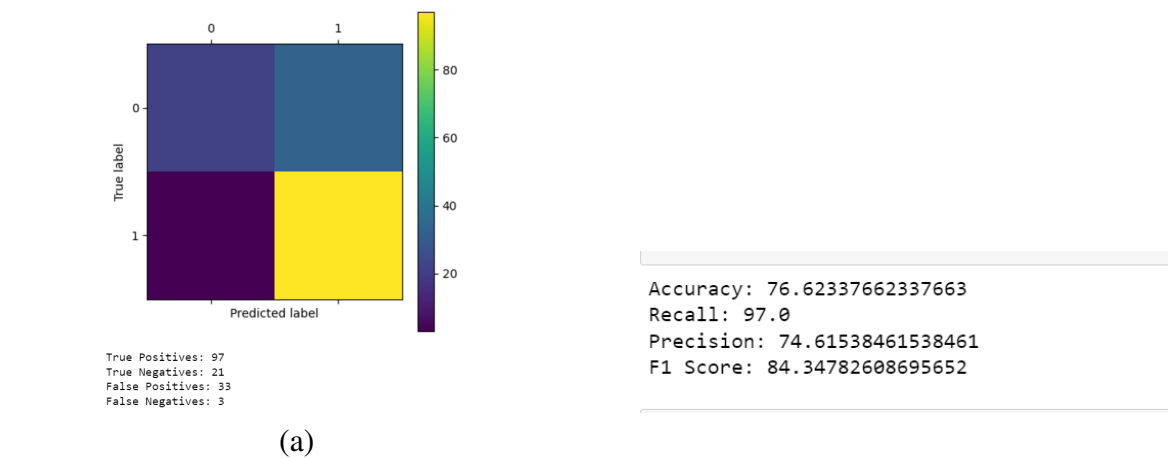


Figure 5.14: MLP NN: (a) Confusion Matrix

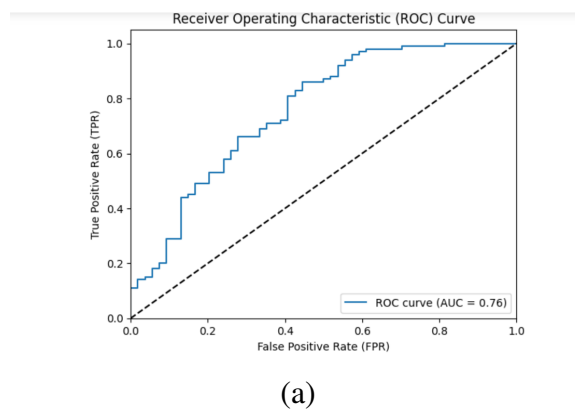


Figure 5.15: MLP NN:(a) ROC Curve

5.1 Conclusion

The aim of this project is to develop a proposed model that identifies different models as an enabling tool for evaluating credit loan application. In the traditional lending process banking authorities adopted '5C' principle to evaluate a borrower. This evaluation relied on personal experience and knowledge of customer delay. The model we created will be precise and accurate with minimum false error. The credit scoring method does not require too much information and reduces bias by inspecting rejected applicants. The models used are Random Forest, Logistic Regression, Decision Tree and NN. The above given models were compared and the model with highest accuracy was adopted. The project was completed wherein we observed that **Random Forest** had the highest accuracy. The model achieved an accuracy of 79.22 percent whereas logistic regression, decision tree and NN (MLP Classifier) had an accuracy of 77.27 percent, 73.37 percent and 76.623 percent.

5.2 Future Scope

In financial firms as illustrated by the problem of loan sanctioning the manual work required for processing of large number of petitions, automation with a great accuracy has proved to be a blessing for these firms. From a Govt. standpoint this model can be applied to predict whether or not the policies they wish to introduce will yield desired results or not. A multi-dimensional credit scoring approach can empower business to gain digital trust in the long haul. Robust creditworthiness empowers a thin file business to get access to formal credit whereas it protects banks from making bad decisions.

Bibliography

- [1] N H Putri et al Journal of Physics: Conference Series 1836 (2021)012039 doi:10.1088/1742-6596/1836/1/012039 IOP Publishing “Credit Risk analysis using support vector machines algorithm”.
- [2] 2007 International Conference on Computational Intelligence and Security Bo Wang¹, Yongkui Liu¹, Yanyou Hao², Shuang Liu¹ ¹ College of Computer Science and Engineering, Dalian Nationalities University, China ² Dalian Branch of China Construction Bank, China “Defaults Assessment of Mortgage Loan with Rough Set and SVM”.
- [3] Muhammad Saad Rahman, *Buck Converter Design Issues*, Master Thesis performed in division of electronics Devices, Thesis No:LiTH-ISKY-EX-06/3854-SE, Linkoping Date:2007-07-17
- [4] I.J Modern Education and Computer Science,2018,5,9,9-16. Published Online May 2018 in MECS DOI:10.5815/IJMECS.2018.05.02 “Credit Risk Prediction using ANN Algorithm” Deepak Kumar and Shruti Goyal.
- [5] 2017 International Conference on Computational Science and Computational Intelligence. “An Improved Credit Scoring Model a Naïve Bayesian Approach”
- [6] IOP Conf. Series: Materials Science and Engineering 1022 (2021) 012042 IOP Publishing doi:10.1088/1757-899X/1022/1/012042 “Loan default prediction using decision trees and random forest: A comparative study”
- [7] 2017 International Conference on Soft Computing, Intelligent System and Information Technology. “Credit Scoring Refinement Using Optimized Logistic Regression” Hendri Sutrisno, Siana Halim Industrial Engineering Department Petra Christian University Surabaya, Indonesia.

- [8] “PREDICTIVE AND PROBABILISTIC APPROACH USING LOGISTIC REGRESSION: APPLICATION TO PREDICTION OF LOAN APPROVAL” IEEE – 40222 8th ICCCNT 2017 July 3-5, 2017, IIT Delhi, Delhi, India.
- [9] ARTIFICIAL INTELLIGENCE FOR CREDIT RISK ASSESSMENT: ARTIFICIAL NEURAL NETWORK AND SUPPORT VECTOR MACHINES Khemakhem, Sihem and Boujelbene, Younes “Artificial Intelligence for Credit Risk Assessment: Artificial Neural Network and Support Vector Machines” ACRN Oxford Journal of Finance and Risk Perspectives 6.2 (2017): 1- 17.