A Machine Learning Approach to Credit Default Prediction and Individual Credit Scoring

Supervisor: Dr. Mahbub Alam Majumdar

Co-Spervisor: Mr. Moin Mostakim

Authors:

MD. JABER RAHMAN 14301041

HASIB AHMED 14301095

A N M SAJEDUL ALAM 12201027

Department of Computer Science and Engineering,
BRAC University.

Declaration

This is to certify that the research work titled “A Machine Learning Approach to Credit Default Prediction and Individual Credit Scoring” is submitted by Md. Jaber Rahman, Hasib Ahmed and A N M Sajedul Alam to the Department of
Computer Science & Engineering, BRAC University in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

We hereby declare that this thesis is based on results obtained from our own work. The materials of work found by other researchers and sources are properly acknowledged and mentioned references.
This thesis, neither in whole nor in part, has been previously submitted to any other University or Institute for the award of any degree or diploma. We carried our research under the supervision of Dr. Mahbub Alam Majumdar and co-supervision of Mr. Moin Mostakim.

Signature of Supervisor:

Dr. Mahbub Alam Majumdar

Signature of Co-Supervisor:

Mr. Moin Mostakim

Signature of Authors:

Md. Jaber Rahman

Hasib Ahmed

A N M Sajedul Alam

Acknowledgement

We might want to offer our appreciation to our supervisor Dr. Mahbub Alam Majumdar for being sympathetically understanding with us all through this one year and directing us by what method should we way to deal with the issue. His imperative proposals and information sources had been of a colossal incentive to our work.

We are likewise Thankful to our theory co-supervisor, Mr. Moin Mostakim for his help.

Ultimately, we might want to give our authentic thankfulness to our folks who constantly upheld us in our tough circumstances and our friend Mr. Ridwan Chowdhury who regularly specifically or by implication helped us or urged us to finish our Research work.

Abstract

In our country the credit scoring system is not in practice yet so as for our undergrad thesis, we have taken upon the challenge of delivering a model well equipped with machine learning techniques to predict loan defaults. Here our main goal is to forecast credit defaults using machine-learning techniques and so we developed a model to output a target score, known as "credit score" which will describe the trustworthiness of an individual for getting a loan. We trained and tested this model based on 'German credit data', which was modified later on. We have Figured out 37 features based on which the data were taken and then after feature selection, we narrowed the number to 23 only by means of feature selection. Then again after thorough observations we analyzed the dataset with different models like Logistic Regression, FLDA, Naïve Bayes, Decision tree, Gradient Boosting tree, Random Forest etc. After that we made a scoring format using weights derived from information gains and also depending on their correlations, which will ensure the assigning of credit score to an individual. Later on we predicted who should receive loan on basis of the scores generated and this prediction was done using a decision tree.

Keywords: *machine learning, credit risk, credit score, model.*

Table of Contents

Chapter 7 Conclusion and Future works

List of Figures

List of Tables

List of Equations

Chapter 01
Introduction

1.1 Introduction

Over the last few decades, the Consumer Spending has emerged as one of the Key drivers of macro-economic conditions throughout the world. Here the gradual increase in Consumer Spending is directly linked with the evolving monetary policies of different organizations which help the consumers with granting loan to them. Again, after the infamous economic recession 2007-08 which is called the worst financial crisis since 1930, it has become quite transparent that consumer behavior has played pivotal roles in every stage. Starting from sowing the seeds of the crisis, consumer role has borne the legacy all along till the end date.

Therefore, to improve consumer behavior and predict their attitude 'credit scoring' had been bestowed with paramount importance. With the passage of time, 'credit scoring' has emerged as a reliable method to ensure the individuals/organizations that can be trusted to grant a loan for the sake of security of the Loan Granting organizations. So basically, 'credit scoring' is a technique that helps organizations to decide whether or not to grant loans to consumers who applies for them. In other words, it can be stated as a statistical number by which lenders will evaluate the probability that an individual will repay his/her debt in time and this credit score is determined based on each individual's credit history.

A person's credit score usually ranges in between 300-850. The higher the score, the more is the person considered financially trustworthy. The Fair Isaac Corporation, known as FICO, created the standard credit score model used by financial institutions. FICO score is by far the most commonly used method in the industry.

As we know that, this is an era of social networking, we can assume people's behavior and characteristics by following their social media activities. By following their social media activities, we are also thinking about generating a credit score to know how credit worthy they are and for verifying false positives we can compare it with some other attributes like a person's job specifications, income, expenses assets, debt, delinquency rate, number of dependents, number of open credit lines and loans, residence history, credit history etc.
In the recent times, facilitating 'credit scoring' with machine learning techniques has added more

feathers to the wings. Currently there are numerous classification algorithms for example, Logistic Regression, Gradient Boosted Trees, Support Vector Trees, and Random Forest etc. for assessment of borrower's trustworthiness. Now, a question arises, which algorithm is best for Credit scoring model? There have been some comparative studies of the classification algorithms by Bae-sens et al. (2003) [1] and Lessmann et al. (2015) where the ranking of classifiers is provided. However, we are concerned about relevance of these research results in application to the real life, because these studies were done on the basis of sample data which were not taken from any commercial bank. Therefore, the prediction of the model too will be influenced since some part of the data was also used to train the model.

Here our work seeks to give an overview of the techniques that we are going to use to develop a model that will forecast the credit default risks and assess the credit score of individuals. We have tried to inject the use of machine learning techniques in credit scoring model that can yield the best results and therefore be considered as a pragmatic approach to the forecasting of credit risk defaults. We are also focusing on retrieving the data from social networks that can be calculated as a futuristic option. We embarked on this study with a view to determine the efficacy and fervently hope that the aggregation of machine learning forecasts of individuals may have much to contribute to the credit lending system and will be much more sophisticated in terms of use. Since machine learning forecasts are substantially more adaptive and are able to pick up dynamics of changings it will be able to deliver more accurate results.

1.2 Motivation

In banking sector of Bangladesh, there is credit rating risk grading (CRG) for companies or businesses but not for an individual implying that there is no Credit scoring system for individuals in practice within our country. With the Implementation of this model, we are trying to shed some light at the fact that personal credit scoring by machine learning is a far better and faster method to assess the scoring and loan system. Our goal is to guide the Bangladeshi industries so that they can be motivated to change their old system of personal banking for customers and make the system faster, more accurate, efficient and secure. Now, the industries are doing it manually and this is only applied for organizations and not on personal levels. For this, a huge amount of time is being wasted for personal loaning process. If the personal credit scoring system gets initiated in Bangladesh, individual loaning system will take minimal time and complexities and can be maintained easily. Customers would not have to wait for longer times for getting loans then, for which they are suffering now.

Chapter 02
Literature Review

2.1 Literature Review

This part of research paper is devoted to the study of different research papers comparing the classification techniques for credit scoring. Abdou & Pointon (2011) orchestrated an in-depth analysis on credit scoring in various fields and finally concluded their thorough research with saying that, there is no single overall best classification technique for credit scoring models [2]. However, they were in accordance with Hand and Henley (1997), who said that the performance of classification technique precisely depends on the convenient variables on the data sets, data structure or just the objective classification [3].

In this study we covered a good number of papers and most of them tried to showcase some new classification techniques alongside the classic classification methods. They compared these classifiers with each other to find out which classifier works as the best. Among this classification methods Logistic regression is assumed to be the industry standard for credit scoring models (Ala'Raj and Abbod (2015)). [4]

Going In-depth to a research work done by Khandani, kim and Lo (2010) [5], we found that they applied machine learning techniques to construct nonlinear nonparametric forecasting models of consumer credit risk. Their Model forecasted delinquencies up to 85% using Linear regression. They divided whole data with the range of six months' period for better output, evaluated the cs scores for those 6 months and calculated the default frequency in the subsequent 6 months for each level of scores. They used Classification and regression tree(CART) in this model. They also specified that Random forest will perform well for discrete outputs but will fail miserably in continuous outputs. They also used Bagging and Boosting to control the risk of failure of model when the variance between the data seemed higher. One of the critical issue of this work is the model will work only for the records of the account holders whose accounts are valid for at least 90 days.

Adding to that, Bellotti and Crook (2007) in their research created a general framework in which the performance of Support Vector Machine is compared with Logistic Regression, LDA and K-nearest neighbors (kNN) [6]. They only restricted their choice to only those accounts which were

opened with in a same range of particular 3 months just to avoid the “population drift” which is explained by Hand (2006). From their research we get, Nonlinear kernels do not outperform the simple linear models. To be precise, polynomial kernels perform poorly due to its overfitting of the data. Here the best results are achieved when a large number of SV’s are extracted. KNN stands out as the poorest in terms of performance in their research.

Bastos (2008) in his research has proposed a credit scoring model using boosted decision trees. He evaluated the performance of the Boosted decision trees using two publicly available credit card applications dataset. His research showed The prediction accuracy of Boosted decision tree being benchmarked against two alternative datamining approach: Multilayer perceptron and Support Vector machines and Boosted decision tree is found to be a competitive method for credit scoring after out-performing both SVM and Multilayer perceptron on two real world credit datasets. In his studies he also showed that Boosted decision trees can also be used to rank the attributes which expresses the likelihood of default [7].

Anne Kraus and Helmut kuchenhoff (2014) in their study have tried to benchmark different techniques for building the scoring models in order to maximize the Area under the ROC curve(AUC). They have introduced AUC as a direct optimization Criterion. Later on they briefly described the classical scoring system. They also found out that by adding non-linearity in the algorithm the logistic additive model gives higher predictive accuracy than the classical logit model. [8]

Masyutin A.A. (2015) in his research has focused on credit scoring based on social network data. He described the schema of social data retrieval in banking data sphere and developed two scorecards: Fraudulent case and Ordinary default purely derived from social data. Furthermore, he used ROC analysis for evaluating the performance of scorecards. This data retrieval from social network also adds a new dimension for credit scoring [9].

The Intention of this paper is to find an optimum machine learning technique for prediction and then to build a scoring model which will assess a core and prediction will be made based on that.

Chapter 03
Models and Methodologies

3.1 Machine Learning

Machine learning is a kind of algorithms that allow software applications to become more accurate in predictability without being explicitly programmed. It is a subset of artificial intelligence based on the idea that system can learn from data, identify the pattern and make decision to get optimal solutions with minimum human intervention. There are two kinds of ML algorithms, supervised machine learning algorithms and unsupervised machine learning algorithms.

3.2 Supervised Learning:

In supervised learning, we have an input variable and an output variable; Algorithms are used to learn the mapping function, from input to output. Supervised machine learning techniques mainly classified in two sub- groups, classification and regression. Regression deals with continuous outputs and Classification deals with discrete outputs. Support vector machine (SVM), Random Forest (RF), Logistic Regression, Decision Tree are the most popular and widely used supervised algorithms.

3.3 Models:

The data was analyzed using several machine learning algorithms like Linear regression, Logistic Regression, Decision tree, Random Forest, Support Vector Machine, LDA, Multilayer Perceptron etc.

3.3.1 Naïve Bayes Classifier:

Naïve Bayes Classifier is one of the most popular and powerful algorithm for classification task. If a dataset has millions of instance with many attributes, then the Naïve Bayes classifier is the suggested one. The foundation of Naïve Bayes classifier is Bayes theorem. Bayes Theorem works on conditional probability. A conditional probability is something like, probability of an event (A), given that another (B) has already occurred [12]. The assumption of Naïve Bayes classification on Bayesian probability called class condition independence.

$$P(A|B) = \frac{(P(B|A) \times P(B))}{P(B)} \quad \text{.....(1)}$$

Here, P (A) is prior probability of the class

$P(A|B)$ is the posterior probability of class given predictor

P (B) is the probability of the predictor

P (B|A) is the likelihood of the probability to normalize the result.

Naïve Bayes classifier predicts the probability of each instances of a class, and the class with highest probability is counted as most likely class, the process of determining the class with highest probability is called Maximum A posteriori (MAP). [10][11]

MAP (A):

= max ($P(A|B)$)

= max (($P(B|A)$ *P (B)) / P (B)))

= max (($P(B|A)$ *P (B))

Predicting credit score is a classification problem and its need Gaussian Naïve Bayes Classification, Gaussian Naïve Bayes gave powerful output in classification.

$$P(x_i|y) = \frac{1}{\sqrt{(2\pi\sigma_y^2)}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right) \quad \text{.....(2)}$$

The parameters σ_y and μ_y are estimated using maximum likelihood.

3.3.2 Logistic Regression

Apart from having good interpretability of the results, simple explanations and robust models, Logistic Regression gives another advantage: Logistic Regression is less sensitive against outliers. However, there are a number of assumptions underlying logistic regression model.

$$p = \frac{\exp(\beta_0+\beta_1.\chi_1 \ldots\ldots+\beta_n\chi_n)}{1+\exp(\beta_0+\beta_1.\chi_1 \ldots\ldots+\beta_n\chi_n)} \quad \text{.....(3)}$$

Most important of these are:

1. Linearity in the explanatory Variables.

2. Absence of interactions among explanatory variables.

Logistic Regression model is simplest in terms of nature. For its simplicity we can consider it has one explanatory variable then we can rewrite this as:

$$\ln\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 . x_1 \quad \text{.....(4)}$$

Now suppose x is a Boolean variable, then we get two equations:

$$\ln\left(\frac{p(1)}{1-p(1)}\right) = \beta_0 + \beta_1 . \quad \text{.....(5)}$$

$$\ln\left(\frac{p(0)}{1-p(0)}\right) = \beta_0 \quad \text{.....(6)}$$

This two together lead to neat result:

$$\frac{p(0)}{1-p(0)} = e^{\beta_1} \frac{p(1)}{1-p(1)} \quad \text{.....(7)}$$

This interprets the co-efficient β_1. It describes the change in probability of default if the change in variables is exactly 1 unit. For example: it can change the concern about 'How the probability of default changes if any of the important feature value is changed. If the savings 1000USD changes to 10,000USD then the change in probability of default can be explained by this.

So finally the simple logistic equation can be represented by:

$$\mathrm{P}\left(\text{Loan Status = default or } 1\right) = \frac{1}{1+e^{-(\beta_0+\beta_1 x+\cdots+\beta_k x_k)}} \quad \text{.....(8)}$$

Where k is the number of independent variables.

3.3.3 Decision Tree:

Let us assume that the dataset of loan applicants is described by n attributes or characteristics: x_1, x_2, x_3….etc. The applicants fall under two different classes: 'good credit' and 'bad credit'. The basic goal of this decision tree model is to find a classifier which can separate good credit samples and bad credit samples. The tree comprises of a set of sequential binary splits of data. At the beginning of the tree there is a root node containing both the samples of good or bad credit data. In order to find the attribute x and corresponding cut-off value C whose objective is giving the best separation keeping the good samples mostly on one side and bad on the other , the algorithm starts to loop over each and every binary split. Fig 1: exhibits a tree which is optimized when the data in the root node is split between the instances with attributes $x_i \geq C_i$ and $x_i < C_i$. The procedure is continuously repeated for the child nodes until it satisfies a stopping criterion. [13]

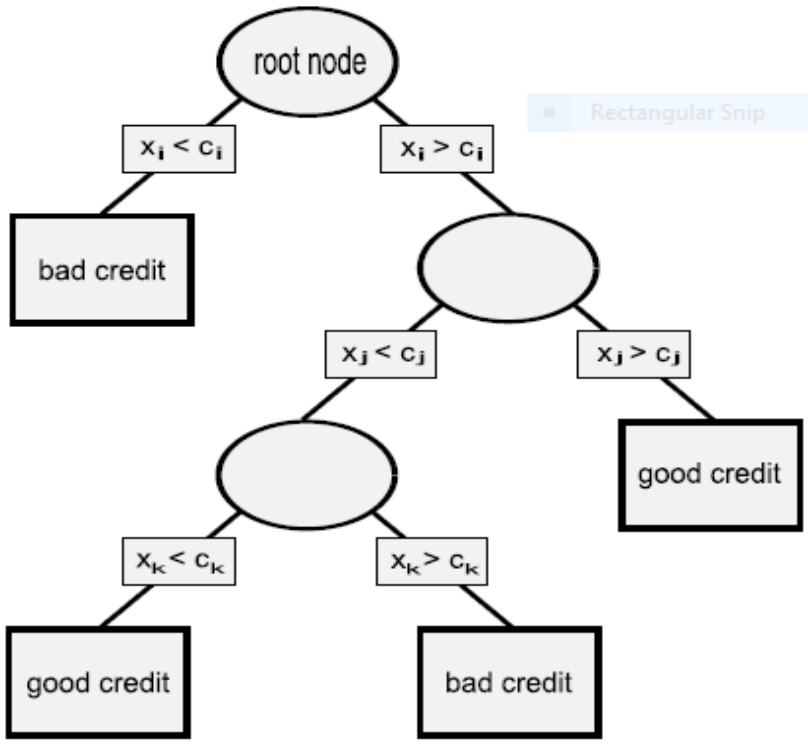

Fig 1: Illustration of a decision tree

3.3.4 Random Forest:

Random Forest uses Decision trees but holds another approach. Rather than growing a single tree with greater depth which has greater probability to be overseen by the analyst, Random forest depends on aggregating the output from numerous “shallow” trees (in other words “stumps”) which are tuned and pruned without much oversights from the analyst. A good number of trees may have been developed from the samples which suggested *age* to be more important feature (as opposed to *income*) whereas some other trees may suggest completely different features to be relevant. The splitting of the trees into child nodes takes place being influenced from Gini Criterion.

$$\text{Gini} = N_L \sum_{k=1\ldots k} p_{kl}(1 - p_{kl}) + N_R \sum_{k=1\ldots k} p_{kR}(1 - p_{kR}) \quad \ldots\ldots\ldots\ldots\ldots\ldots (9)$$

Where
p_{kl} = proportion of class k in left node
p_{kR} = proportion of class k in right node.

When it comes to generate a prediction, polling takes place among all the underlying trees and out of them the prediction value with majority number of votes emerge as victorious.

With a good number of input variables and a potentially sparse dataset, there is a greater chance for a predictive model to find out spurious relationships between those inputs and the chosen target variable which will lead to overfitting of data. We can represent the model by this equation:

$$space = \{F(X, a_i); i = 1,2,3 \ldots . nos\ of\ trees\} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

a_i represents the number of independent and identically distributed random vector in such a way that every tree has a vote for most popular class. For building the algorithm for this model we choose Random K data points from the training set and thus develop a decision Tree using the data which belonged to these data points. A concept called pruning can be leveraged to reduce complexity of the model by replacing sub-trees, that only provide little predictive power with leaves. Extremely randomized trees can be obtained by checking the split random parameter and disabling pruning. Important parameters to tune for this method are the minimal leaf size and split ratio, which can be changed after disabling guess split ratio. Good default choices for the

minimal leaf size are 2 for classification and 5 for regression problems. After multiple times tuning the random forest model has performed this way:

3.3.5 Gradient Boosted Trees (XGBoost):

The Gradient Boosted Trees Operator trains a model by iteratively improving a single tree model. After each iteration step the Examples are reweighted based on their previous prediction. The final model is a weighted sum of all created models. Training parameters are optimized based on the gradient of the function described by the errors made.

Boosting:

Boosting is a procedure which aggregates many weak classifiers only for it to achieve a high classification performance. Again, Boosting does the work of stabilizing the responses of classifiers w.r.t changes in the training sample. As a representative of boosting algorithms 'ADABOOST' is applied here. After building K^{th} decision tree total misclassification error, ε_k is calculated which is stated to be the sum of weights of misclassified credits over the sum of weights of all the credits.

$$\varepsilon_k = \frac{\sum_{i\, mis}\, \omega_i^{(k)}}{\sum_i\ \omega_i^{(k)}} \quad \text{..... (11)}$$

Here I iterates over all the sample data and then the weights of misclassified applicants are boosted up.

$$\omega_i^{(k+1)} = \frac{1-\varepsilon_k}{\varepsilon_k}\, \omega_i^{(k)} \quad \text{..... (12)}$$

After that the latest weights are re-normalized:

$$\omega_i^{(k+1)} \rightarrow \omega_i^{(k+1)} / \sum_i \omega_i^{(k+1)} \quad \text{..... (13)}$$

And the tree (k+1) is developed. Here the final classification or score of an applicant i is sum of the weights of classification over the individual trees.

$$F_i = \sum_{K=1}^{N} \left(\log \left(\frac{1-\varepsilon_k}{\varepsilon_k} \right) f_i^{(k)} \right) \quad \text{..... (14)}$$

Here N= number of grown trees; $f_i^{(k)} = 1(-1)$ if the Kth tree can place the instance on a good (bad) credit leaf. So as it stands, good credits will tend to have a higher positive score and bad credits will tend to be a negative score. [14]

3.3.6 Multi-layer perceptron:

Multi-Layer perceptron classifier is a classifier based on Artificial Neural Network; MLPC consists of multiple layers of nodes, where nodes are fully connected between themselves. They are composed of an input layer to receive input signals and an output layers to make predictions on given inputs. Between these input and output layer there are many arbitrary layers and that's arbitrary layers are the computational engine of the multilayer perceptron. MLP is also used for supervised classification problem.

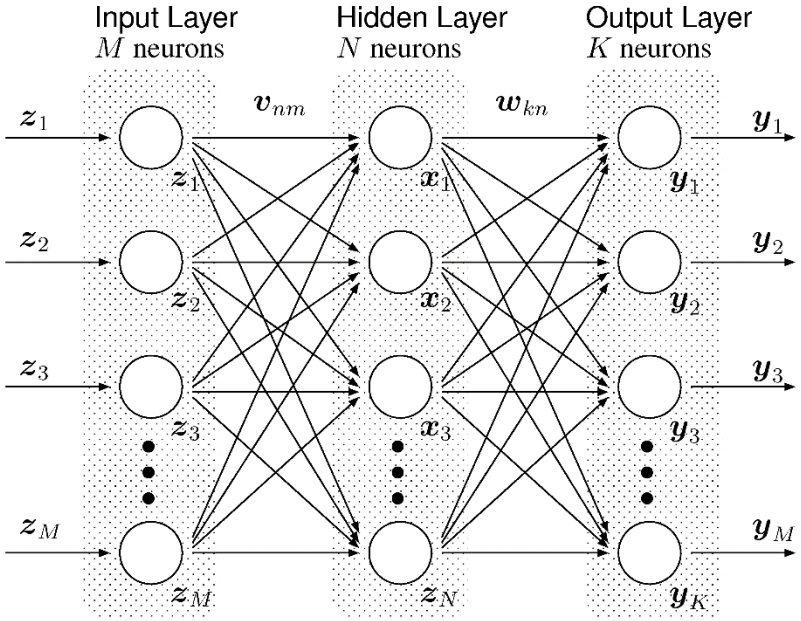

Fig 2: Multi-layer Perceptron

All nodes in intermediate layer use sigmoid function, sigmoid function takes real valued input and convert it into 0 and 1.

3.3.7 Linear Discriminant Analysis

For classification problem, we traditionally use Gaussian-based linear discriminant analysis if the dataset is balanced. Linear discriminant analysis uses covariance matrices for classifying data into two classes, when the covariance matrices of two classes are not the same then the sample of dataset is called imbalanced. An imbalanced dataset has a negative impact in the performance of LDA and result might be misleading. In LDA a balanced dataset gave more accurate performance than the unbalanced one.

Model Co-efficient: $\beta = C^{-1}(\mu_1 - \mu_2)$ ..... (15)

Pooled Covariance Matrix: $C = \frac{1}{n_1+n_2}(n_1C_1 + n_2C_2)$ ..... (16)

Where: β= Linear model coefficient, C_1, C_2= Covariance matrices, μ_1, μ_2= Mean vectors

As our Dataset is imbalanced, we need to use the confusion matrices to calculate the F-measure for determining the accuracy of the dataset.

Fisher Linear Discriminant:

Fisher's linear discriminant is a classification method that projects high-dimensional data onto a line and performs classification in this one-dimensional space. The projection maximizes the distance between the means of the two classes while minimizing the variance within each class. This defines the Fisher criterion, which is maximized over all linear projections, *w*:

$$J(w) = \frac{|m_1 - m_2|^2}{s_1^2 + s_2^2}$$ ..... (17)

Where m represents a mean, s^2 represents a variance, and the subscripts denote the two classes.

3.3.8 Support Vector Machine (SVM):

Support vector machine is one of the most powerful and widely used supervised machine learning technique. Generally, SVM offers higher accuracy than other classification algorithms such as Logistic regression, Decision tree etc. In general, SVM used for classification problem but it also has a potentiality to give a very optimistic output for regression problem. SVM has the capability to handle categorical and continuous data smoothly. SVM generate a hyperplane in multidimensional space to differentiate multiple classes. SVM construct optimal hyperplane in iterative manner to minimize the error. [15] SVM separates binary classified data by a hyperplane such that the marginal width between hyperplane. By maximizing the marginal width, the complexity of the model has been reduced. For, imbalanced dataset SVM detects accuracy by this equation.

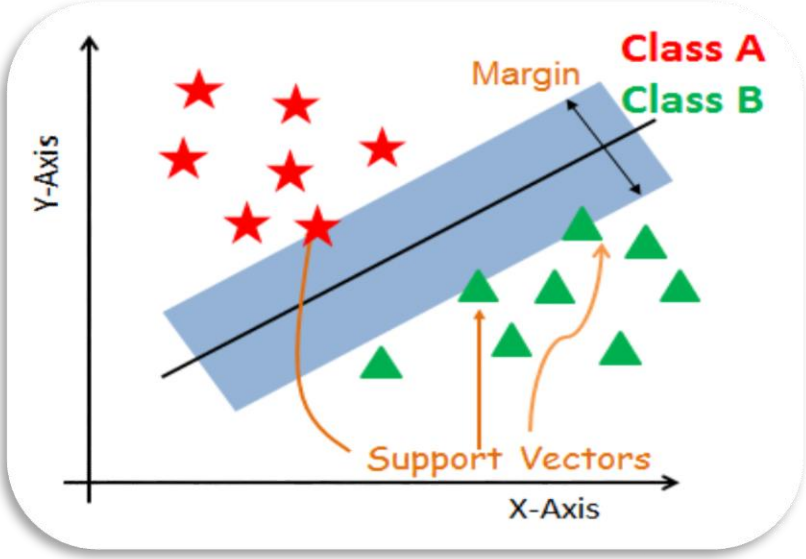

Fig 3: Support Vector Machine.

$$Accuracy = \frac{(TP+TN)}{(FP+FN+TP+TN)} \quad \text{..... (18)}$$

To get optimal output from data set we need to monitor Sensitivity (TP) and Specificity (TN) separately for different class. Based on TP rate and TN rate, A G-Mean has been proposed for a balanced data set, Here, G-mean is the geometric mean of the sensitivity and specificity [16][17].

TN = TN/ (FN+TN)

TP = TP/TP+TN

G-mean = $\sqrt{(Sensitivity * Specificity)}$ ..... (19)

Now, as we measured that out data set is imbalanced, for this reason using G-mean for getting optimal accuracy won't work here. We proposed F-measure, recall, precision for calculating accuracy in imbalanced dataset.

Precision = True Positive / (True Positive + False Positive)

Recall = True Positive / (True positive + False Negative)

F-measure = 2*((Precision*Recall) / (Precision +Recall))

For, SVM we found F-measure of 0.8536 and AUC is 0.6874. Our dataset built on categorical output, in this point SVM can generate a good output curve but the problem is we have a large data set and SVM has high training time (0.01s) for large dataset compare to other algorithms such as Naïve Bayes Classifier (74ms).

3.3.9 Logit Boosting

Logit boost is a boosting algorithm. Considering Adaboost as a general additive model if we apply the cost function of logistic regression then we can drive Logit boost algorithm. It optimizes log-loss instead of exponential loss which is usually optimized by adaboost, that's why logitboost is considered to be less sensitive to the outliers (mislabeled data) than adaboost. Boosting uses a simple working approach: it applies the classification algorithm sequentially over the reweighted versions of training data and then takes the weighted majority of votes of sequence of classifiers thus produced. Actually this process significantly improves the accuracy of the prediction. Logistic regression models the posterior class probability Pr(G=k|X=x) where K is bi-valued since it has only 2 classes A and B. Linear regression uses the linear functions in x to model these probabilities while ensuring that they sum to one and also remain within (0,1). Here the logit function is:

$$f(z) = \frac{e^z}{e^z+1} = \frac{1}{1+e^{-z}} \quad \text{(20)}$$

Here the logit of a number p between 0 and 1 is given by formula:

$$\log(p) = \log\left(\frac{p}{1-p}\right) = \log(p) - \log(1-p) \quad \text{(21)}$$

The logits of the odds of the unknown probability of this model are modeled as the linear function of X_i.

$$\text{logit}(p_i) = \text{logit}\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{1,i} \ldots\ldots + \beta_k x_{k,i} \quad \text{(22)}$$

3.4 Methodology

We have organized the total work procedures in to some key divisions. For instance, after starting our research at the very first we went for some background studies and figured out all the necessary features important for our model by analyzing all the available features used in 'Lending club loan data' dataset and 'German Credit Data'. We also consulted with some officials of 'Bangladesh Bank Training Academy' and following this we finalized the short list of features. We obtained the data from 'German Credit Data' and removed unnecessary features from it. Then the dataset was pre-processed before the final feature selection where the initial 37 features were cut down to 23 only using several methods like Chi-squared, Information Gain etc. After that the data was trained and tested with multiple number of Supervised machine learning models. After that we tuned the parameters of algorithms to get the best results. The performance was evaluated using several performance measurements. Then A scoring pattern was formulated and thus we were able to generate a credit score and give verdict on the individual which was followed by the Loan default prediction using that credit score. The workflow is given in Fig-4:

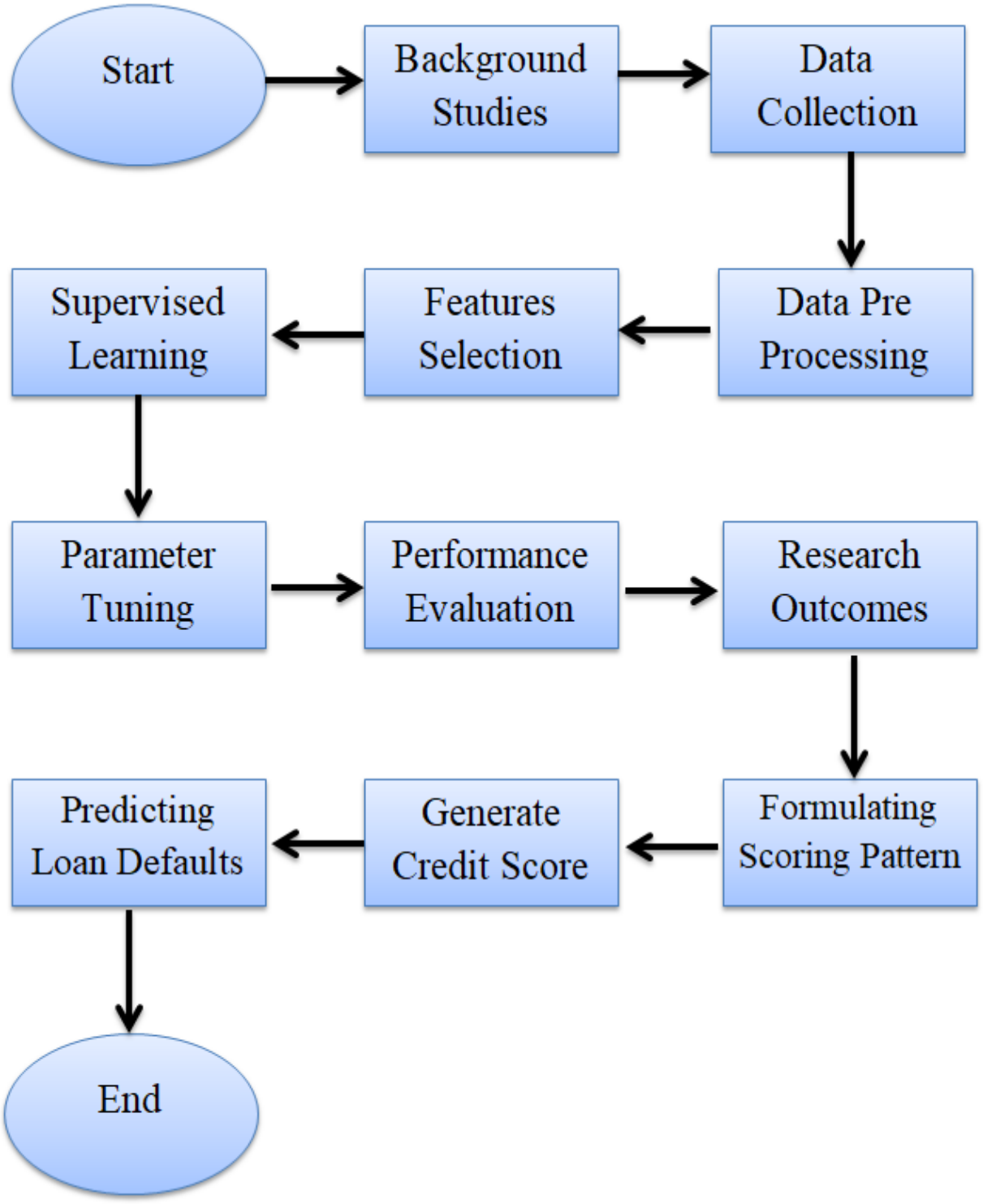

Fig 4: Workflow

Chapter 04

Data collection, Analysis and Processing

4.1 Features

For Listing down the Features that can work as a baseline for our credit scoring model we studied about the prevalent banking systems in our country. During the study we consulted with some of the officials of ‘Bangladesh Bank Training Academy’ and sorted out a wide array of important features. From the economic perspective of our country These features are important for credit scoring and loan default predictions but many of these features may act differently in other countries in predicting loan defaults due to the difference in economic conditions. Here the features are described below:

1. Status (Credit status): Current credit status of a consumer.
2. Seniority (Job seniority in years): Experience of job in years.
3. Home (Type of Home Ownership): Home ownership determines the expenses of the client of accommodation. If the client has own home, then the expense from home rent will not be added to expense.
4. Time (Time of requested loan): Here this feature indicates that the duration of paying the loan.
5. Age (Client's age): Client’s age is basic feature because of government job’s age limitation and age determines ability of doing job.
6. Job (Type of Job): In job seniority we will decide how consistent job type is. For example, Govt. job, multinational company, Bank job are the more secure job. Inconsistent jobs are proprietorship, small firms, pharmacy.
7. Expenses (Amount of expenses): Monthly expense of the loaner.
8. Income (Amount of income): Monthly income of the loaner.
9. Assets (Amount of assets): Amount of all properties, all the assets the client has.
10. Debt (Amount of debts): It checks is there any existing load or pressure of loans of the client.
11. Amount (Amount of requested loans): How much money the client requested for the loan.
12. installmentCommitment (Installment Commitment): Committed monthly installment for paying the loan in months.
13. openAcc (Open Account): Number of open credit lines in the borrower's credit file.

14. Savings: The amount rate which is getting saved monthly from income after deducting the monthly installment and expenses. The more amount the client has in their savings account the more reliability will be counted for giving loan.
15. seniorityR (Seniority Ratio): Seniority Ratio basically the ratio of job seniority of the loner.
16. timeR (Time Ratio): The ratio of the time of paying the full amount of loan which was taken.
17. ageR (Age Ratio): Age ratio of the loaner.
18. expensesR (Expenses Ratio): Expense ratio of the loaner.
19. incomeR (Income Ratio): Income ratio of the loaner.
20. assetsR (Assets Ratio): Estimated amount ratio of the assets of the loaner.
21. debtR (Debt Ratio): Debt ratio of the loaner.
22. amountR (Amount Ratio): Amount ratio of requested loan.
23. residenceSince (Residence Since): From how much time the loaner is resident in this local area.
24. Totalpymnt (Total payment): The total how much amount is paid by the loaner.
25. savingsR (Savings Ratio): Ratio of the savings of the loaner.
26. delinq2yrs (Delinquency of two years): In previous two years the amount how much delinquency the loaner had.
27. installment (Installment): The monthly payment owed by the borrower if the loan originates from the bank.
28. numDependents (Number of Dependents): Number of financially dependent people on the loaner.
29. inqlast6mths (Inquiries of Last Six month): Number of credit inquiries in past six months of the loaner.
30. foreignWorker (Foreign Worker): The loaner is permanent citizen here or not.
31. creditHistory (Credit History): The credit history of the loaner from the local credit information bureau.
32. totalacc (Total Account): Total number of accounts of the loaner.
33. revolBal (Revolving Balance): Total credit revolving balance of the loaner.
34. revolUtil (Revolving Utilization): Revolving line utilization rate, or the amount of

credit the borrower is using relative to all available revolving credit.

35. annualinc (Annual Income): Annual income of the loaner.
36. Marital: Marital Status of the loaner.
37. Records: Checking of the loaner's credit record existence.

This Feature list is presented in a tabular form here:

Table 1: Total features

#	Designation	#	Designation
1.	Status	20.	Records
2.	Seniority	21.	Expenses
3.	Home	22.	Income
4.	Time	23.	Assets
5.	Age	24.	Debt
6.	Marital	25.	Amount
7.	installmentCommitment	26.	openacc
8.	Savings	27.	seniorityR
9.	Timer	28.	ageR
10.	ExpensesR	29.	incomeR
11.	assetsR	30.	debtR
12.	Amount	31.	residenceSince
13.	Totalpymnt	32.	savingsR
14.	delinq2yrs	33.	Installment
15.	numDependents	34.	inqlast6mths
16.	foreignWorker	35.	creditHistory
17.	Totalacc	36.	revolBal
18.	Job	37.	revolUtil
19.	Annualinc		

4.2 Data Pre-processing

We have taken a dataset containing 4447 entries of customers who have solicited a personal credit to the bank, with loan terms ranging from 12 to 60 months. This dataset contained both categorical attributes and continuous ones. We made sure that there are no missing values as it could affect the training of the model. We added the mean values of that particular column which contained missing values. This is an approximation which can add variance to the particular dataset. For calculating mean a simple function was called in the python code:

data [‘particular feature with missing value/values’].mean( ) .

For the preprocessing works we mostly used Python library scikit-learn (Sklearn). The dataset contained values in form of strings and numbers. Since the machine learning models cannot run datasets with text attributes, we converted those features into numerical ones. We set a range from 0 to 1 for every categorical features only. Then their values were assigned with any of 0/0.25/0.5/0.75/1 according to their weights and importance. We used the python library ‘numpy’ to label the dataset after converting everything (except that column which is used for ‘status’) into numerical values.

4.2.1 Feature Scaling

In order to analyze the data by the machine learning algorithms, we needed to scale the data down to such a range which will be easily evaluated by the algorithms and will be clearly understandable. The continuous attributes don’t mean any effect on calculations, so we only converted the categorical data into numerical ones by plotting a certain range and putting the corresponding categorical data into that range to get a numerical value.

We set the range for categorical values from 0 to 1 and all of them were divided according to this fixed range.

ageR (Age Ratio): Age ratio of the loan applicant was scaled with four segments. In first segment, from 0 years to 25 years old people that means in our dataset ‘age (0, 25]’ was exchanged with ‘0’ value. Similarly, the section ‘age (25, 30]’ was replaced with value ‘0.25’,

‘age (30, 40]’ was replaced by value ‘0.5’. Then, ‘age (40, 50]’ was exchanged with value ‘0.75’and ‘age (50, 99]’ group was exchanged by value ‘1’.

timeR (Time Ratio): In time ratio feature, we have five segments which are ‘time (0, 12]’, ‘time (12, 24]’, ‘time (24, 36]’, ‘time (36, 48]’ and ‘time (48, 99)’. All of these range are in months and was exchanged by 1, 0.75, 0.5, 025, 0 sequentially.

Home (Resident Type): This feature includes five groups of data which are ‘own’, ‘rent’, ‘parent’, ‘others’ and ‘ignore’. All of these are categorized by these values sequentially 1, 0.75, 0.5, 0.25, 0.

Records: Data of existence of records of the borrower is basically two types ‘no_rec’ and ‘yes_rec’ in our dataset which are exchanged by -1 and 3 values, where ‘no_rec’ is changed by 0 and ‘yes_rec’ is changed by 1.

Job (Occupation Type): ‘freelance’, ‘fixed’, ‘partime’ and ‘other’ these four groups of data were exchanged by 0.75, 1, 0.5 and 0.

seniorityR (Seniority Ratio): Seniority ratio data is a group of five categorical data which are ‘sen (-1, 1]’, ‘sen (1, 3]’, ‘sen(3, 8]’, ‘sen (8, 14]’ and ‘sen (14, 99]’ are exchanged by 0, 0.25, 0.5, 0.75, 1 etc. values sequentially.

incomeR (Income Ratio): Similarly, ‘inc (0, 80]’, ‘inc (80, 110]’, ‘inc (110, 140]’, ‘inc (140, 190]’ and ‘inc (190, 1e+04]’ are the group of data those were exchanged by values 0, 0.25,0.5, 0.75, 1 etc. sequentially.

expensesR (Expense Ratio): In the same way, ‘exp (0, 40]’, ‘exp (40, 50]’, ‘exp (50, 60]’, ‘exp (60, 80]’ and exp (80, 1e+04] these group of data of expense ratio were exchanged by 1, 0.75, 0.5 ,0.25, 0 etc. values sequentially.

expensesR (Expense Ratio): In the same way, ‘exp (0, 40]’, ‘exp (40, 50]’, ‘exp (50, 60]’, ‘exp (60, 80]’ and exp (80, 1e+04] these group of data of expense ratio were exchanged by 1, 0.75, 0.5 ,0.25, 0 etc. values sequentially.

assetsR (Asset Ratio): Asset ratios of loaners are five groups of data which are ‘asset (-1, 0]’, ‘asset (0, 3e+03]’, ‘asset (3e+03, 5e+03]’, ‘asset (5e+03,8e+03]’ and ‘asset (8e+03, 1e+06]’ are exchanged by values 0, 0.25,0.5, 0.75, 1 etc. sequentially.

debtR (Debt Ratio): ‘debt (-1, 0]’, ‘debt (0, 500]’, ‘debt (500, 1.5e+03]’, ‘debt (1.5e+03, 2.5e+03]’ and ‘debt (2.5e+03, 1e+06] these groups of data represent debt ratios of loaners which are exchanged by values 1, 0.75, 0.5 ,0.25, 0 etc. Sequentially.

amountR (Amount Ratio): ‘am (0,600]’, ‘am (600,900]’, ‘am (900, 1.1e+03)’, ‘am (1.1e+03, 1.4e+03]’ and ‘am (1.4e+03, 1e+05]’ these group of values represent amount ratios of requested loan which are exchanged by the values 1, 0.75, 0.5 ,0.25, 0 etc.

savingsR (Savings Ratio): ‘sav (-99, 0]’, ‘sav (0, 2]’, ‘sav (2, 4]’, ‘sav (4,6]’ and ‘sav (6, 99]’ these groups of data represent amount of savings ratio of loaners which are exchanged by values 0, 0.25, 0.5, 0.75, 1 respectively.

Status: In this feature, for value ‘good’, we put the value ‘A’ which is the best possible rating and we exchange ‘bad’ value with ‘B’ value.

4.3 Training and testing dataset

We split our data into two subsets: 1) Training Data 2) Testing data. This is done only to fit our model into training data, in order to make predictions on the test data.
After going through all these we would want to avoid both over-fitting and under-fitting since they both will affect the predictability. Due to overfitting the model will be very accurate on training data, but will perform poorly on new entries or the data which were not trained. This happens because the model is not generalized anywhere which means anyone can now generalize the outcomes and from here can’t make any inferences on other data.

On the contrary, when the model being under-fitted means the model doesn’t fit into the training data and thus misses out on the pattern of the data. So it can’t be generalized to new entries as well. Therefore, to avoid both the scenarios we can do either cross validation or Train/test-split. In this particular case since the dataset has around 4447 entries, hence there is a lesser opportunity that any particular feature’s values will be similar in nature, that means they shouldn’t look like they are representing the whole dataset being in certain smaller group. The above mentioned criteria is where cross validation works efficiently, so we are going to skip cross validation and adopt Train/test-split method.

Train/test-split:

It is already said that the data is divided into training data and test data. The technical term associated with this division is “splitting”. In the training data, there is a known output on which our model has to learn the trend for being generalized on the test data later on. Using scikit-learn library and specifically the train_test_split method we can split our dataset into desired ratios. *test_size=0.2* inside the train_test_split function describes that the split ratio is 80-20, which means 80% of data is assigned for training the model and the rest 20% is assigned for testing the model.

4.4 Feature selection

Machine Learning basically follows a simple rule- if you input garbage, you will get garbage (in this case noisy data) as output. Here we don’t always need a large array of attributes from our disposal to have a better output and that implies sometimes it is better when its lesser in number because it reduces training time, testing time and also features less things to get our concern.

Selection of attributes is a critical and important point because it can necessarily highlight the difference between successfully modeling the problem and not. We need to be careful enough in case of including redundant attributes, for example: K-nearest neighbor uses small neighborhoods in the attribute space to find out classifications and regression predictions. Redundant features can skew the predictions as well.

Irrelevant attributes might cause overfitting of data. Here in this particular case Decision tree algorithms like C4.5 creates optimal spits in the feature’s values. So before evaluating algorithms we need to remove redundant and irrelevant features from our dataset.

In this Feature selection part, the best subset of attributes in the dataset is automatically sorted out. A well preformed way of analyzing the problem of selecting attributes is a “state-space search”, where the search space is discrete and contains the most possible combinations of features one can choose from the dataset.

The principal benefits of doing feature selection on the data are:

1. Reduces Over-fitting: It implies that the less the redundant data are, the less opportunity to make decisions clearly based on bad noises
2. Improve accuracy: The less misleading data, the higher the modeling accuracy.
3. Reduces training time: Low amount of data ensures faster training of the model.

Due to the presence of many categorical attributes, feature selection process turns into a difficult one. We used a few methods like Chisquared, GainRatio and SVM attribute evaluation etc. to select the attributes, but each time the set of selected attributes was slightly different from the previous set of selected attributes. That's why, we selected 23 features with the highest consensus among all the rankers.

Finally, we could determine the selected set of features:

Selected Features = {SeniorityR, Records, Seniority, SavingsR, Job, AssetsR, Savings, IncomeR, Home, Amount, Income, amountR, Time, Marital, timeR, Assets, Age, ageR, Expenses, inqlast6mths, expensesR, ResidenceSince, Debt}

These histograms represent all instances for some particular values of our selected features. For instance, we have a feature named "Records" and its range in histogram is 0, .5, and 1. We plotted all of our instances according to these assign values. Here, in 0 we have 3677 instances and for 1 we have 769 instances. In these histograms X axis represent the range of features and Y axis represents number of instances of our dataset.

Histogram explaining the selected features:

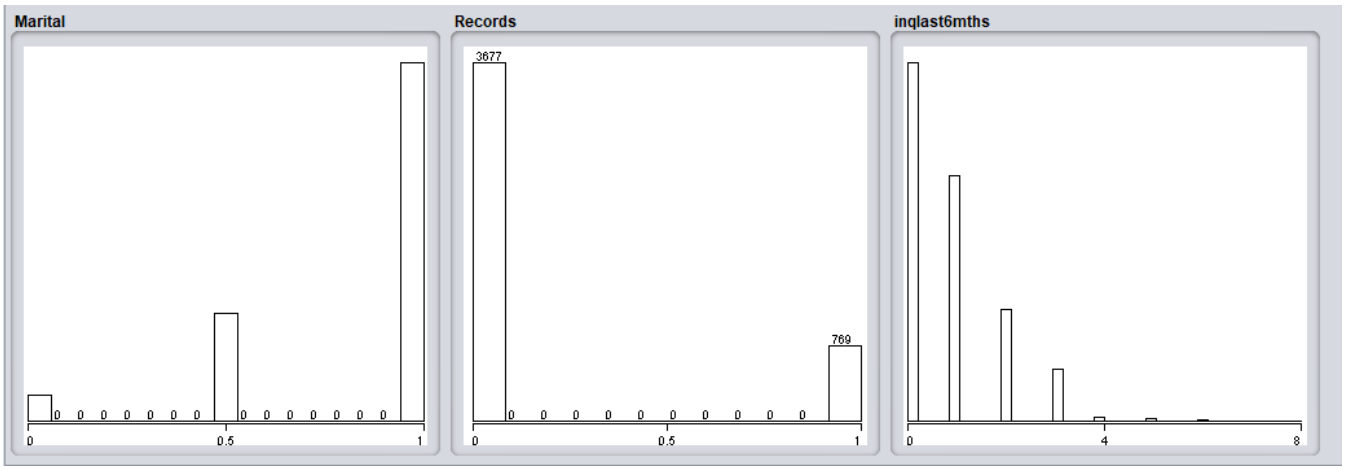

Fig 5: Features (Marital, Records, inqlast6mths)

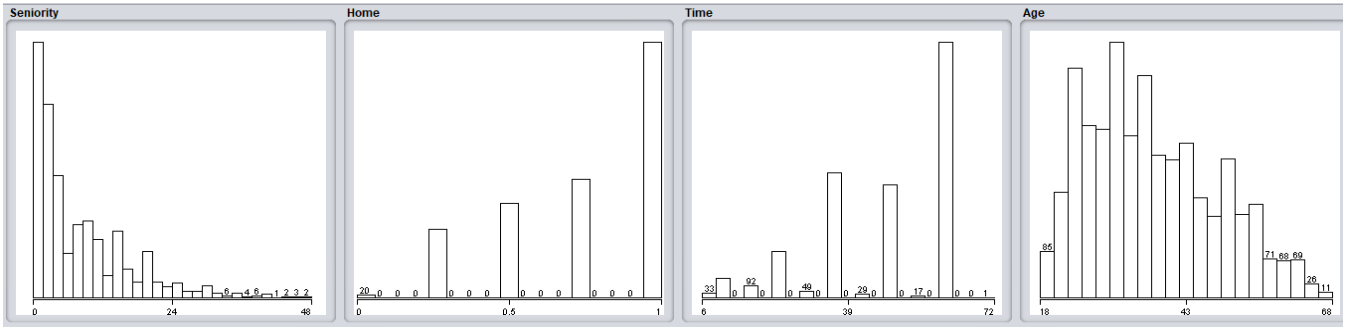

Fig 6: Features (Seniority, Home, Time, Age)

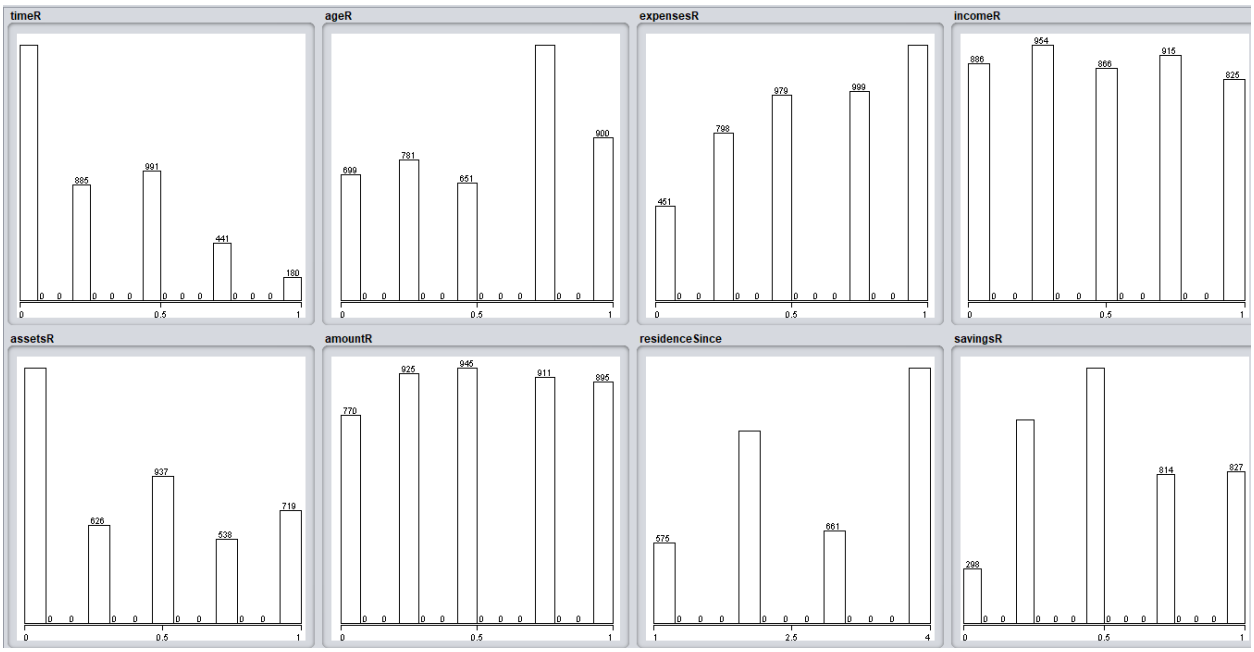

Fig 7: Features (timeR, ageR, expensesR, incomeR, assetsR, amountR, residence since, savingsR)

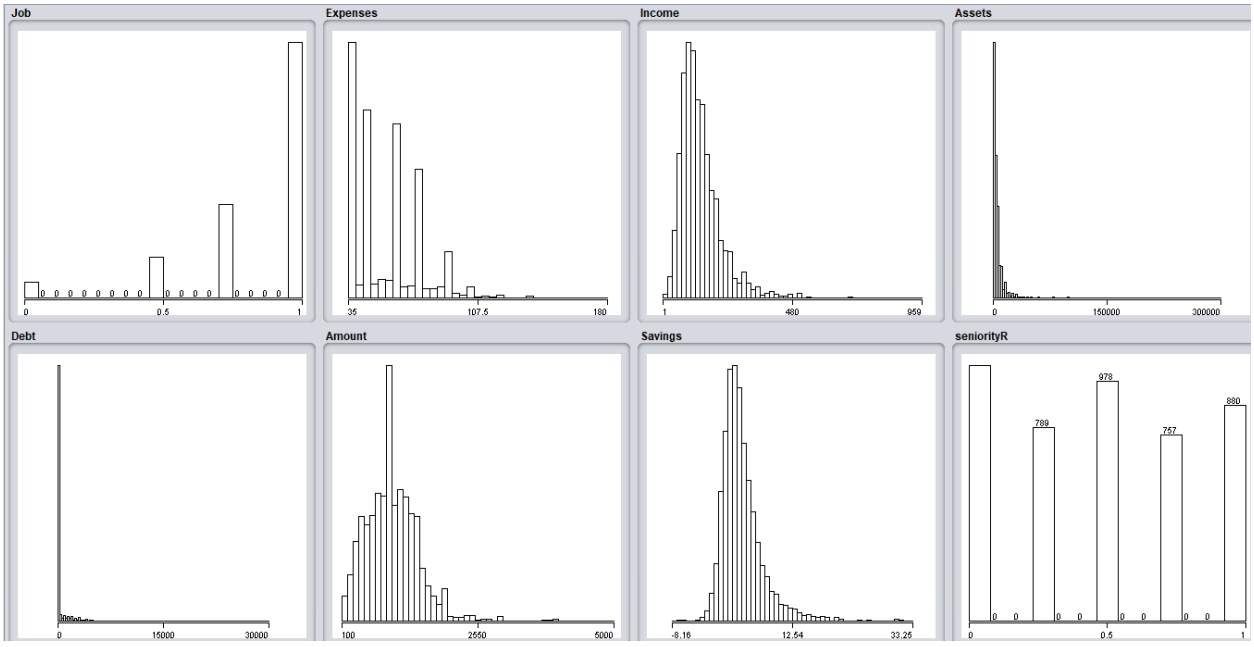

Fig 8: Features (Job, Expenses, income, assets, amount, debt, savings, seniority)

4.5 Correlation

Before building any complex model we need to have an overview of the data relation to see how our different variables interact together. Correlation finds the trends shared between two variables. When with the increase in value of one variable is associated with the increase in values of another variable, then these two variables are said to be positively correlated. On the other hand, when with the increase in value of one variable is associated with decrease in value of another variable then these two variables are said to be negatively correlated. If the correlation is Zero, then that means there is no pattern visible between two variables.

The total selected features are shown in Table-2 with their correlation percentage in a tabular form:

Table 2: Correlation

Selected Features	**Correlation**	**Selected Features**	**Correlation**	**Selected Features**	**Correlation**
seniority	8.82%	Home	2.81%	Age	0.91%
Records	7.72%	Amount	2.38%	ageR	0.20%
Seniority	6.75%	Income	2.08%	Expenses	0.09%
savingsR	4.82%	amountR	2.07%	inqlast6mths	0.06%
Job	4.66%	Time	1.02%	expensesR	0.03%
assetsR	4.41%	Marital	1.01%	residenceSince	0.02%
Savings	3.44%	timeR	0.97%	Debt	0.01%
IncomeR	3.23%	Assets	0.95%		

Chapter 05
Results and Performance Evaluation

5.1 Parameter Tuning:

Supervised classification and regression models are applied on our prepared and modified dataset of German credit data to predict if the applicant should be given loan or not. In this following part performance of different classification models are analyzed. Before going straight to performance of all the models at first some parameter tuning is needed to find out the optimum parameters for each model. Parameters tuning includes changing the split ratios of the data set for each tests to gain a better output. Here in this case we tested each and every model for the split ratios of 50:50, 60:40. 70:30 and 80:20 for obtaining the best output from each model. Apart from the split ratios, for both decision tree and Random forest the 'Maximum depth' were changed repeatedly from 2 to 40 to get the best performance. Decision tree performed best at a split ratio of 60:40 and at 'maximum depth>15'. Again for Random forest the 'Number of trees' were also tuned from 2 to 40 and it gave the best prediction rate at the split ratio of 80:20, at maximum depth of 4 and at 'number of trees=20'. The rest of the models were tuned with only split ratios.

5.2 Performance Evaluation

Considering our dataset, where 3197 instances are classified as class A and 1249 as Class B (Class A represents Good credits and Class B represents Bad credits). So this is an imbalance dataset and since the variation between two classes is very high so subsampling (down sampling in this particular case) may lead to information loss, so the dataset does not require any further modification. Now for an imbalance dataset to measure the performance we can use a wide range of performance measurements.

Performance metrics should be selected keeping problem domain, project goal and business objectives into considerations.

5.2.1 Confusion Matrix

Confusion Matrix is an additional measure of predictability performance. A confusion matrix is a table which is often used to describe the performance of a classification model on set of test data,

whose true value is known. We can easily calculate algorithm's accuracy for a particular test data through confusion matrix. Here:

1. *True Positive*: If a default is correctly classified and also labelled as a default then it is said to be '*True positive'*.

2. *False Positive*: if a non-default is wrongly classified and is labelled as a default then it is termed as '*False Positive'*.

3. *True Negative*: If a non-default is correctly classified and also labelled as a non-default then it is said to be '*True negative'*.

4. *False Negative*: if a default is wrongly classified and is labelled as a non-default then it is termed as '*False negative'*.

Table 3: The confusion matrix:

	Positive	Negative
Positive	True Positive	False Positive
Negative	False Negative	True Negative

Sometimes accuracy can be decisive, to ensure the accuracy of a model we use confusion matrix. Here basically we test the model's classification results against the actual observed classification.

Precision:

Precision is the number of true positives divided by total positive number. Precision can be declared as the exactness of a model. Therefore,
Precision = True Positives / (False positives + True Positives)

Recall:

Recall is another number in confusion matrix, it represented by true positive is divided by true positives and false negatives.
Recall = Trues positive/ (True Positive + False Negative)
We can call recall as sensitivity or true positive rate of a data

For example, we generated a confusion matrix for Naïve Bayes classifier:

Naïve Bayes Classifier:

	A	B
A	453	82
B	176	178

Here A represents as True and B represents as False. Now Precision for the Naïve Bayes is:

Precision = AA/ (AA+AB)

= 453/ (453+82)

= 0.8467

Here, for Naïve Bayes algorithms we represented recall as:

Recall = AA/ (AA+BA)

= 453 / (453+176)

= 0.72

F-measure conveys the balance between the precision and the recall. We can represent F-Measure as

F-measure = 2*((Precision*Recall) / (Precision + Recall))

=2 * ((0.72*0.8467) / (0.72+0.8467))

= 0.775

=77.5%

Support Vector Machine (SVM):

	A	B
A	586	43
B	164	96

Precision = 0.8045
Recall = 0.9093;

F-measure = 85.36%

Fisher-Linear Discriminant Analysis (F-LDA):

	A	B
A	462	72
B	167	188

Precision = 0.8651

Recall = 0.7344

F-measure = 79.44%

Logistic Regression:

	A	B
A	575	143
B	54	117

Precision = 0.8008

Recall = 0.9141

F-measure = 85.37%

Random Forest:

	A	B
A	560	127
B	69	133

Precision = 0.8151

Recall = 0.8903

F-measure =85.10%

Multi-Layer Perception:

	A	B
A	566	126
B	63	134

Precision = 0.8179

Recall = 0.8998

F-measure = 85.68%

Gradient Boosted Trees (XGBoost):

	A	B
A	577	126
B	52	134

Precision = 0.821

Recall = 0.890

F-measure = 85.61%

Decision Tree:

	A	B
A	803	145
B	189	197

Precision = 0.742

Recall = 0.750

F-measure = 74.5%

Logit Boost:

	A	B
A	592	37
B	136	124

Precision = 0.801

Recall = 0.805

F-measure = 79.0%

5. 2.2 Classification Accuracy:

Accuracy is a metric for evaluating the classification models. Formally, Accuracy can be expressed as:

$$\text{Accuracy} = (1\text{- error}) = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} = \frac{TP+TN}{TP+TN+FP+FN} \quad \ldots\ldots (23)$$

Probably this is the most common evaluation metric for classification problems. It performs better for equal number of observations in each class.

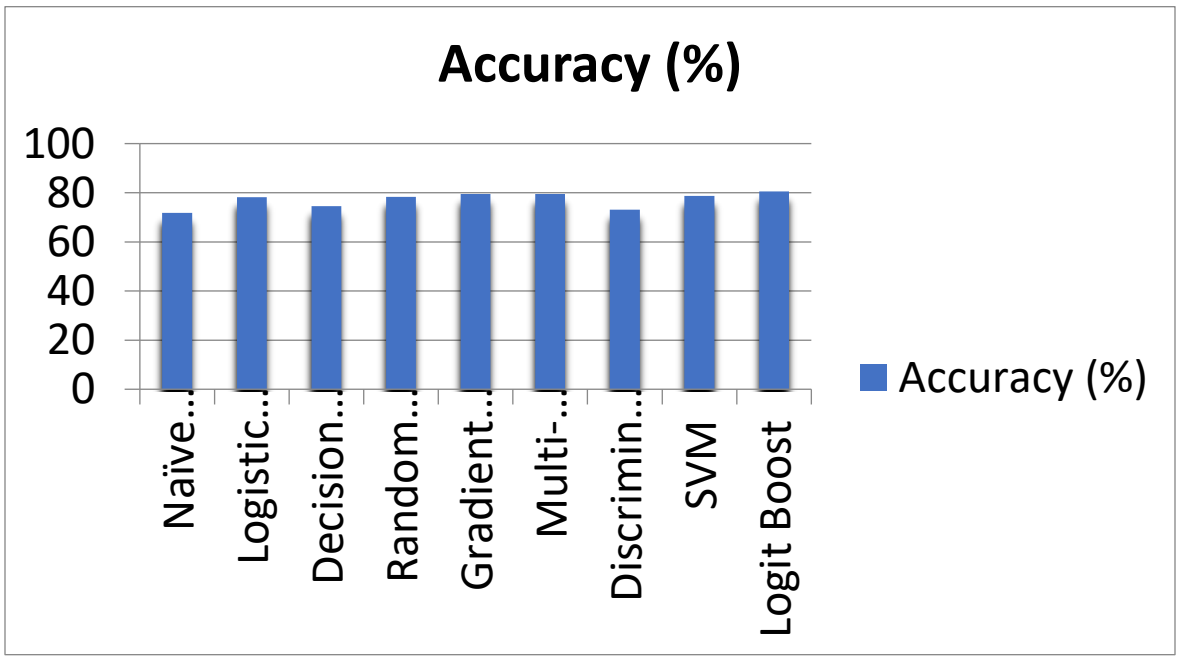

Fig 9: Classification accuracy

Table 4: Accuracy and runtime of models

Model	Split Ratio	Accuracy	Runtime
Naïve Bayes	80:20	71.76%	0.015s
Logistic Regression	50:50	78.23%	0.17s
Decision tree	60:40	74.58%	0.17s
Random Forest	80:20	78.29%	1.52s
Gradient Boosted Trees (XGBoost)	50:50	79.42%	45s
Multi-layer perception	80:20	79.42%	9.62s
Discriminant Analysis (FLDA)	50:50	73.11%	0.03s
SVM	50:50	78.68%	0.01s
Logit Boost	80:20	80.54%	0.02s

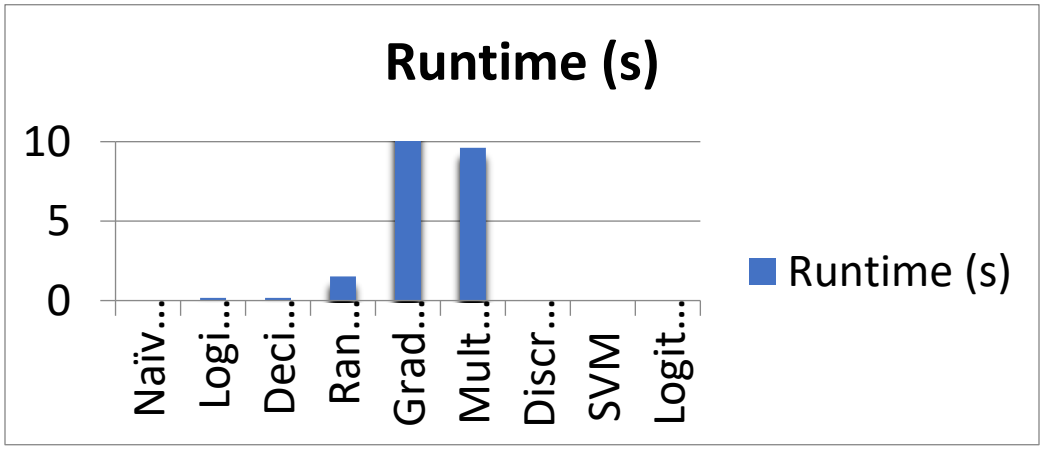

Fig 10: Runtime of the models

5.2.3 AUC:

One of the prominent one is Receiver Operating Characteristic (ROC) and the related Area under the Curve (AUC). In the case of Credit scoring a binary problem is examined; whether the customer will do the default or non-default? Due to this two-class problem, two classes are required for describing the ROC graphs. A classification model, which is supposed to give a discrete output for prediction, can face 2 different cases: True positives and false positives.

So can derive the following parameters from this concept:

$$tp\ rate(\ true\ positive\ rate) = \frac{Defaults\ correctly\ classified(tp)}{Total\ number\ of\ defaults(p)}$$

$$fp\ rate(\ false\ positive\ rate) = \frac{Non-Defaults\ wrongly\ classified(fp)}{Total\ number\ of\ non-defaults(n)}$$

Plotting the above mentioned both the fractions of TPR (true positive rate) and FPR (False positive rate) initiates ROC curve. The diagonal in the graph means no predictive ability and the perfect model will implement a curve which will be tending to the particular point (0, 1). The goal of the ROC curves is to find and tune a model that can maximize true positive rate. The AUC is a measure of the predictive accuracy for different kind of machine learning algorithms and methods, which directly contributes as optimization criterion. The area under the ROC curve (AUC) ranges from 0 to 1. A perfect model has an AUC value of 1. Whereas an uninformative scoring classifier would give a value of 0.5 [1]. An AUC of 0.5 can essentially be compared with random guessing.

Table 5: AUC values

Models	AUC	Models	AUC
Naïve Bayes Classifier	0.776	Gradient Boosted Trees	0.824
Logistic Regression	0.817	SVM	0.682
Decision Tree	0.682	Logit boost	0.814
Random Forest	0.821	F-LDA	0.797
MLP	0.790		

Now we plot the ROC curves for the algorithms using:

X axis False Positive Rate.

Y axis True Positive Rate.

Naïve Bayes Classifier:

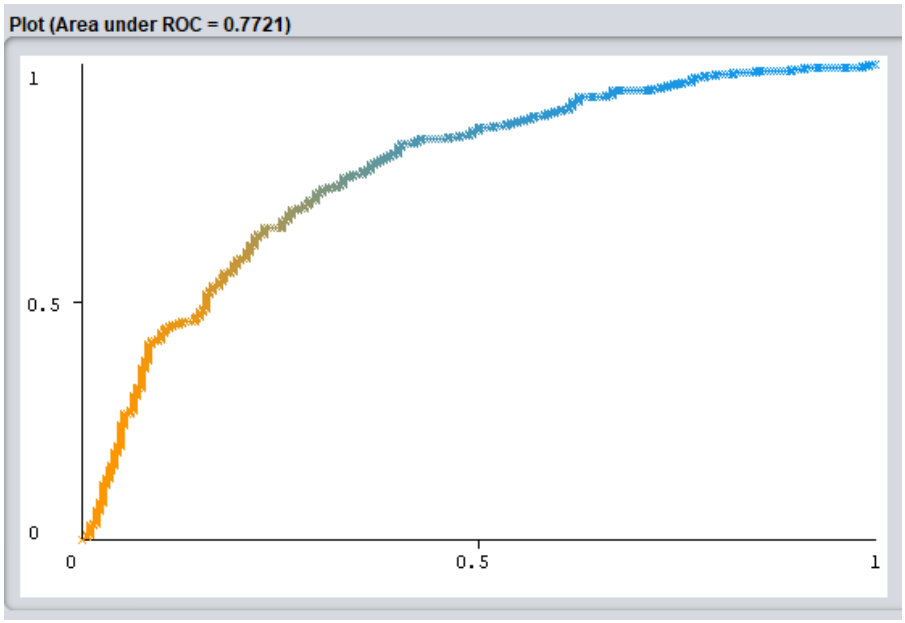

Fig 11: ROC curve of Naïve Bayes Classifier.

Logistic Regression:

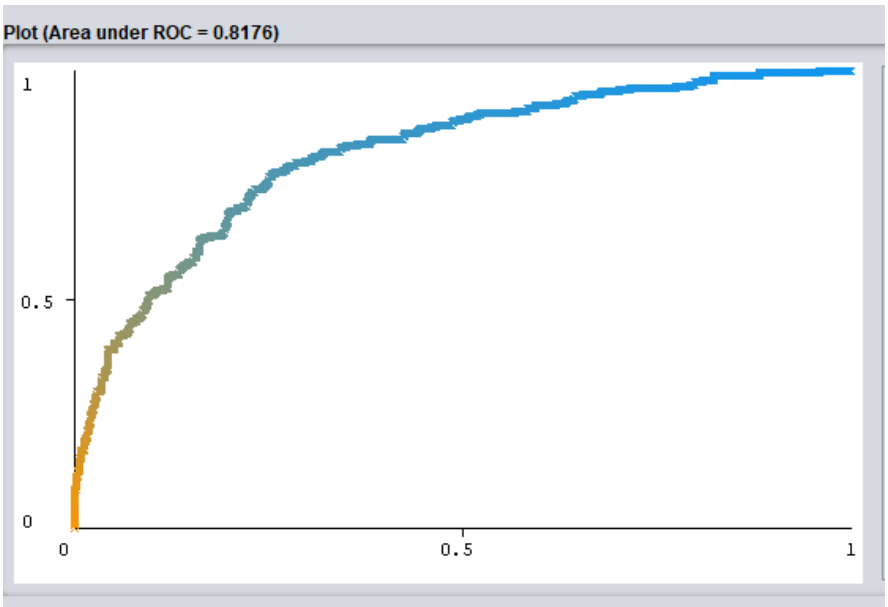

Fig 12: ROC curve of Logistic Regression.

Multi-Layer perceptron(MLP):

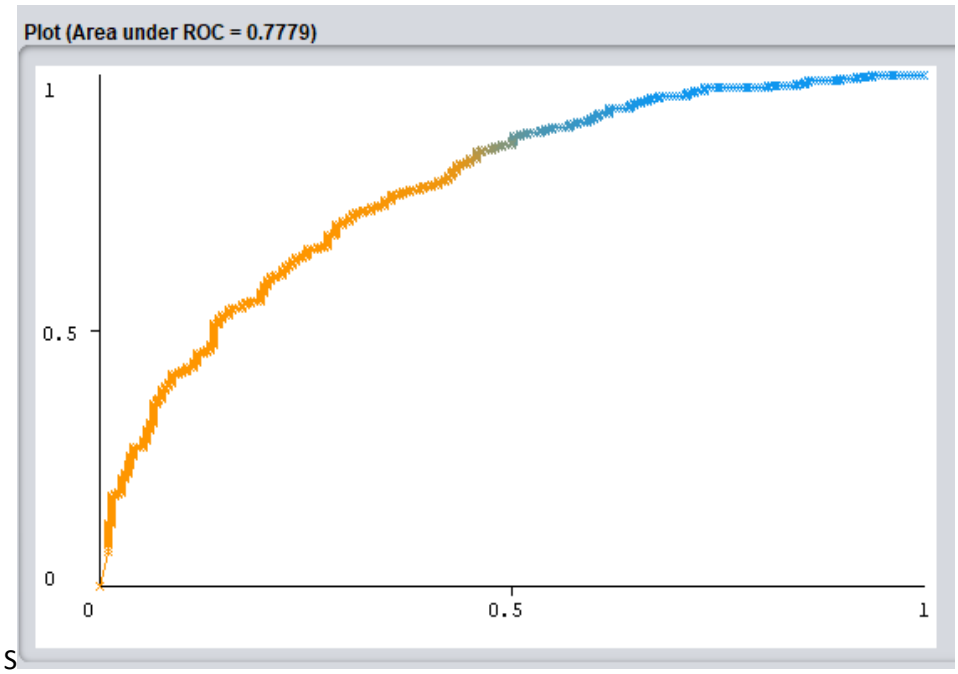

Fig 13: ROC curve of MLP

Support Vector Machine:

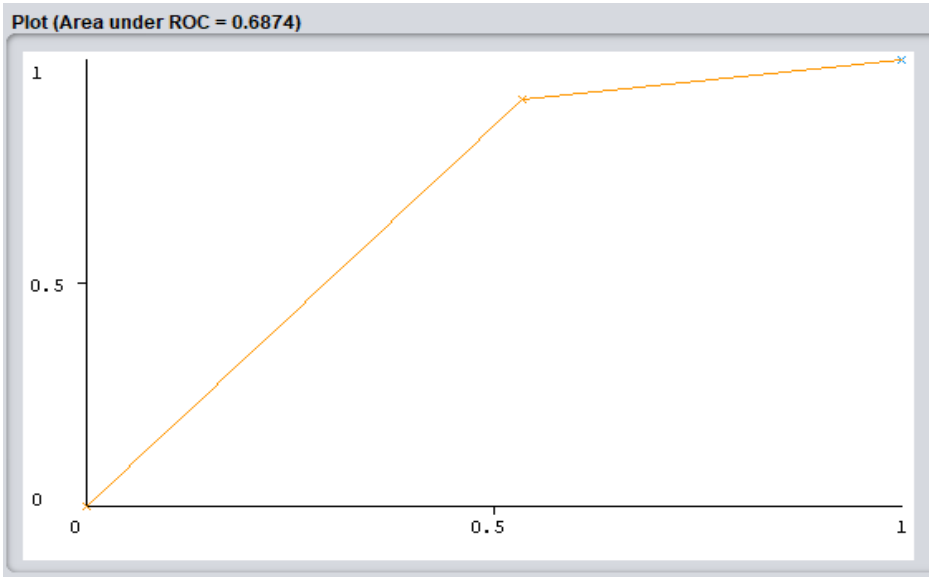

Fig 14: ROC curve of SVM.

Random forest:

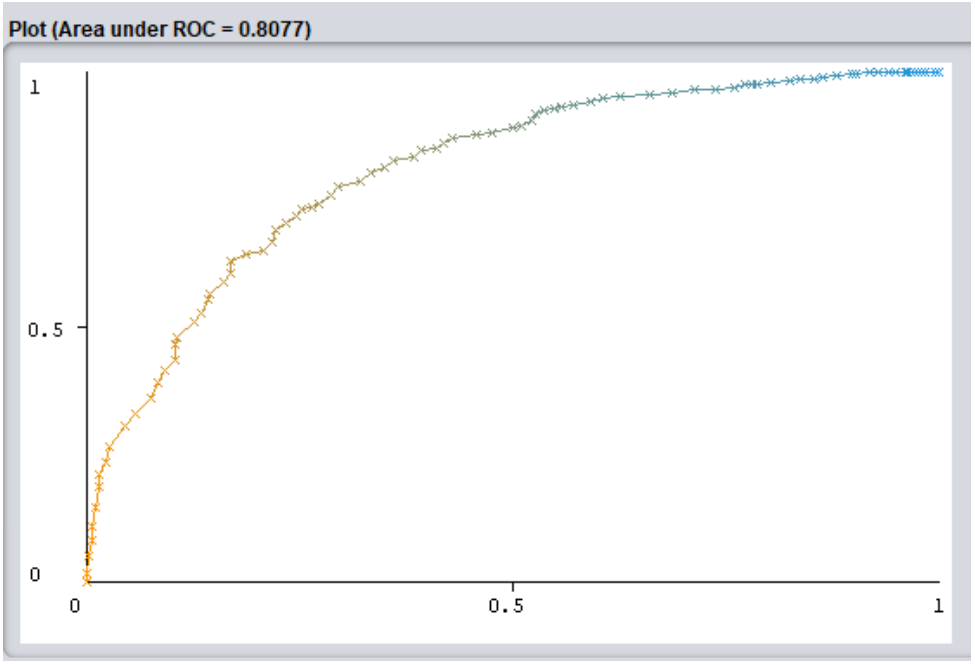

Fig 15: ROC curve of Random forest.

F-LDA:

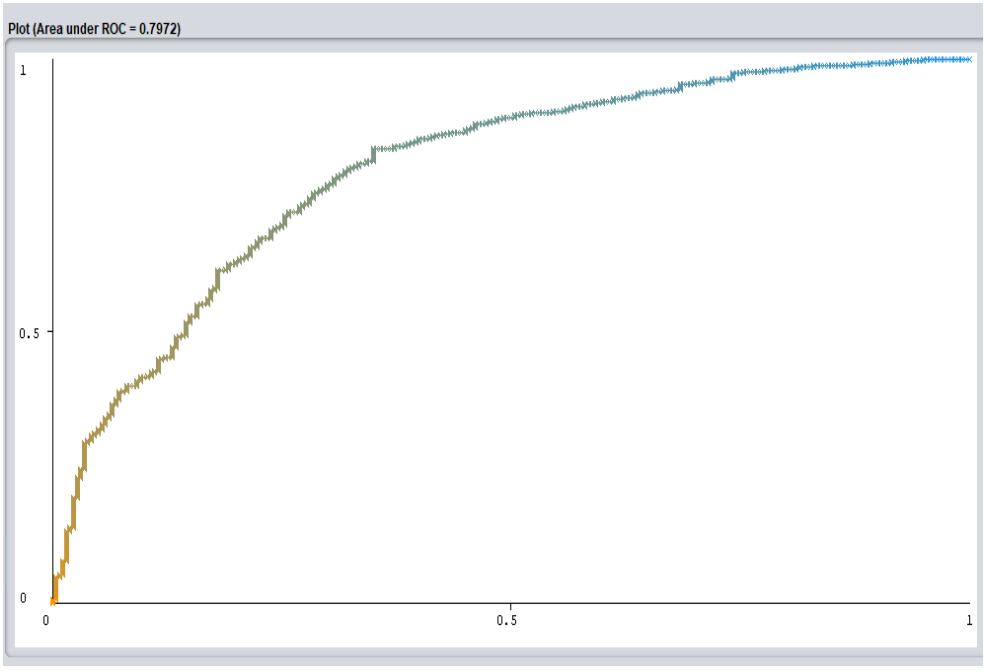

Fig 16: ROC curve of F-LDA.

Any equal transformation to AUC like these ROC curves would imply a Gini co-efficient with definition *Gini=2 and AUC= -1*

We can get a clear concept about the predictive power of the models basing on this table:

Table 6: Standards of AUC

Predictive Power	Area Under ROC
Very Good	>85%
Good	>80%
Acceptable	>70%

Here two samples of defaults and non-defaults are expressed as x_n andx_{nd}.

$S(n_{nd}.n_d)$= 1 if$n_d < n_{nd}$; 0.5 if$n_d = n_{nd}$; 0 if$n_d > n_{nd}$.

AUC can be written as follows:

AUC= $\frac{1}{n_{nd}.n_d}\sum_1^{n_{nd}}\sum_1^{n_d} S(n_{nd}.n_d)$ ..... (24)

5.2.4 F-Measure:

The F-Measure or in other words F-score or F1 score is another measure of test's accuracy. It is defined as the weighted harmonic mean of precision and recall. F-measure is just a combined metric which works best for an unbalanced class where one class is important than the other (e.g.: scenarios like loan default or fraud predictions). For these cases it is better to pick a classifier which have a better F1-score.

$F\text{-}measure = 2\times\frac{Precision\times Recall}{Precision+Recall}$ ..... (25)

F-measure is used here for obtaining the exactness and positive rate of algorithms. F-score varies from algorithm to algorithm because of the differences in exactness and true positive rate of data, the higher exactness and true positive rate indicates better F-score and the accuracy of the algorithms.

Table 7: F-measures of models.

Model	**F-Measure**	**Models**	**F-measure**
Naïve Bayes	77.5%	Multi-layer perception	85.68%
Logistic Regression	85.37%	Discriminant Analysis (FLDA)	79.44%
Decision tree	81.3%	Logit Boost	79.0%
Random Forest	85.10%	SVM	85.36%
Gradient Boosted Trees	86.6%		

5.2.5 Matthews Correlation Coefficient

According to Matthews correlation coefficient formulas [18]:

$$N = TN + TP + FN + FP$$

$$S = (TP + FN)/N$$

$$P = (TP + FP)/N$$

Where, N= Total Observations & P= Prevalence. So,

$$Correlation = \frac{\mathrm{TP/N}\sqrt{b^2-4ac}}{2a} \quad \text{..... (26)}$$

Mathews co-relation co-efficient is more informative than other confusion matrix measures, for example F-measure & accuracy. As we never consider all four component of confusion matrix for calculating F-measure of an accuracy, thus there remains a possibility of misleading

predictions and it's the biggest advantages of Mathews correlation co-efficient over F-measure and accuracy.

Table 8: Matthews correlation coefficient of models

Models	MCC	Models	MCC
Naïve Bayes Classifier	0.396	Gradient Boosted Trees(XGBOOST)	0.480
Logistic Regression	0.421	Multilayer Perceptron	0.473
Decision Tree	0.371	SVM	0.430
Random Forest	0.440	FLDA	0.412
Logit Boost	0.494		

Chapter 06

Credit Scoring and Loan Default Prediction

6.1 Formulating Scoring pattern:

After selecting 23 features from our dataset, we ranked the feature by Info gain attribute. Then we followed Credit Risk Grading (CRG) Model, Bangladesh bank for scaling the features and classified consumer according to their credit score. Our consumers are classified in 9 categories.

Table 9: Credit scores and grading

Grading	Short name	Score
Superior	SUP	801-900
Good	GD	701-800
Acceptable	ACCPT	601-700
Marginal	MG	501-600
Special-Mention	SM	401-500
Sub-standard	SS	301-400
Doubtful	DF	201-300
Bad & Loss	BL	1-200

We assigned values to our 23 features according to their importance and Ranking by Info gain attribute1. Our highest ranked feature is ‘SeniorityR’, there we assigned score 10 out of 100. In ‘seniorityR’ there are some subsets when the range is between (0-.25) we assign the score 3, when it is (.25-.5) score will be 5, for (.5-.75) it will be 7, and for (.75-1) it will be 10.

Range	Score out of 100	Score out of 900
(0-25)	3	27
(.25-.5)	5	45
(.5-.75)	7	63
(.75-1)	10	90

2. For ‘Records’ we assigned a total score of 9 and there will be many subsets of records, when the interval is (0-.25) score will be 2, for (.25-.5) score will be 4, for (.5-.75) score will be 6, for (.75-1) score will be 9.

Range	Score out of 100	Score out of 900
(0-.25)	2	18
(.25-.5)	4	36
(.5-.75)	6	54
(.75-1)	9	81

3. For ‘Seniority’, we assigned a total score of 8 and its subsets are for (0-8) score will be 2, for (8-16) score will be 4, for (16-24) score will be 6, for (24-32) score will be 7, for (32-48) score will be 8.

Range	Score out of 100	Score out of 900
(0-8)	2	18
(8-16)	4	36
(16-24)	6	54
(24-32)	7	63
(32-48)	8	72

4. For ‘SavingsR’s the assigned score is 7: for interval (0-.25) score will be 1, for (.25-.5) score will be 3, for (.5-.75) score will be 5, for (.75-1) score will be 7.

Range	Score out of 100	Score out of 900
(0-.25)	1	9
(.25-.5)	3	27
(.5-.75)	5	45
(.75-1)	7	63

5. For ‘Job’ the score is assigned as 7, and for its subset value will be assigned as for interval (0-.25) score will be 1, for (.25-.5) score will be 3, for (.5-.75) score will be 5, for (.75-1) score will be 7.

Range	Score out of 100	Score out of 900
(0-.25)	1	9
(.25-.5)	3	27
(.5-.75)	5	45
(.75-1)	7	63

6. For ‘AssetsR’ the value is assigned as 7 and its subsets are for interval (0-.25) score will be 1, for (.25-.5) score will be 3, for (.5-.75) score will be 5, for (.75-1) score will be 7.

Range	Score out of 100	Score out of 900
(0-.25)	1	9
(.25-.5)	3	27
(.5-.75)	5	45
(.75-1)	7	63

7. ‘Savings’ has total score of 6 and its subsets are, for interval (-8.16-0) score will be 0, for (0-13) score will be 3, for (13-33.25) score will be 6.

Range	Score out of 100	Score out of 900
(-8.16-0)	0	0
(0-13)	3	27
(13-33.25)	6	54

8. Here our assigned value for the feature ‘IncomeR’ is 6. For interval (0-.25) score will be 1, for (.25-.5) score will be 2, for (.5-.75) score will be 4, for (.75-1) score will be 6.

Range	Score out of 100	Score out of 900
(0-.25)	1	9
(.25-.5)	2	18
(.5-.75)	4	36
(.75-1)	6	54

9. The assigned score for ‘Home’ is 5, for interval (0-.25) score will be 1, for (.25-.5) score will be 2, for (.5-.75) score will be 3, for (.75-1) score will be 5.

Range	Score out of 100	Score out of 900
(0-.25)	1	9
(.25-.5)	2	18
(.5-.75)	3	27
(.75-1)	5	45

10. The assigned score for ‘Amount’ is 5: for interval (100-1500) score will be 2, for (1500-3000) score will be 3, for (3000-4000) score will be 4, and for (4000-5000) score will be 5.

Range	Score out of 100	Score out of 900
(100-1500)	2	18
(1500-3000)	3	27
(3000-4000)	4	36
(4000-5000)	5	45

11. The assigned score for ‘Income’ is 4, for interval (100-300) score will be 2, for (300-600) score will be 3, for (600-959) score will be 4.

Range	Score out of 100	Score out of 900
(100-300)	2	18
(300-600)	3	27
(600-959)	4	36

12. The assigned score for ‘AmountR’ is 4, for interval (0-.25) score will be 1, for (.25-.5) score will be 2, for (.5-.75) score will be 3, for (.75-1) score will be 4.

Range	Score out of 100	Score out of 900
(0-.25)	1	9
(.25-.5)	2	18
(.5-.75)	3	27
(.75-1)	4	36

13. The assigned score for ‘Time’ is 3 for interval (1-30) score will be 3, for (30-50) score will be 2, and for (50-72) score will be 1.

Range	Score out of 100	Score out of 900
(1-30)	3	27

(30-50)	2	18
(50-72)	1	9

14. The assigned score for ‘Marital’ is 3. For (0-.25) score will be 3, for (.25-.5) score will be 1, for (.5-.75) score will be 2, for (.75-1) score will be 3.

Range	Score out of 100	Score out of 900
(0-.25)	3	27
(.25-.5)	1	9
(.5-.75)	2	18
(.75-1)	3	27

15. The assigned score for ‘TimeR’ is 3. For (0-.5) score will be 3, for (.5-.75) score will be 2, for (.75-1) score will be 1.

Range	Score out of 100	Score out of 900
(0-.5)	3	27
(.5-.75)	2	18
(.75-1)	1	9

16. The assigned score for ‘Assets’ is 3. For (0-10k) score will be 1, for (10k-20k) score will be 2, and for (20k-30k) score will be 3.

Range	Score out of 100	Score out of 100
(0-10k)	1	9
(10k-20k)	2	18
(20k-30k)	3	27

17. The assigned score for ‘Age’ is 3, For (18-25) score will be 1, for (25-35) score will be 2, for (35-50) score will be 3, for (50-68) score will be 2.

Range	Score out of 100	Score out of 100
(18-25)	1	9
(25-35)	2	18
(35-50)	3	27
(50-68)	2	18

18. The assigned score for ‘AgeR’ is 2. For (0-.25) score will be 0, for (.25-.5) score will be 1, for (.5-.75) score will be 2, for (.75-1) score will be 1.

Range	Score out of 100	Score out of 900
(0-.25)	0	0
(.25-.5)	1	9
(.5-.75)	2	18
(.75-1)	1	9

19. The assigned score for ‘Expenses’ is 1. For (35-107.5) score will be 1, for (107.5-180) score will be 0.

Range	Score out of 100	Score out of 900
(35-107.5)	1	9
(107.5-180)	0	0

20. The score for ‘InqLast6months’ is 1. For (0-4) score will be 0, for (4-8) score will be 1.

Range	Score out of 100	Score out of 900
(0-4)	0	0
(4-8)	1	9

21. The assigned score for ‘ExpenseR’ is 1. For (0-.5) score will be 1, for (.5-1). Score will be 0.

Range	Score out of 100	Score out of 900
(0-.5)	1	9
(.5-1)	0	0

22. The assigned score for ‘ResidenceSince′ is 1. For (1-2.5) score will be 0, for (2.5-4) score will be 1.

Range	Score out of 100	Score out of 900
(1-2.5)	0	0
(2.5-4)	1	9

23. The total assigned score for ‘Debt’ is 1. For interval (0-10k) score will be 1, for (10k-30k) score will be 0.

Range	Score out of 100	Score out of 900
(0-10k)	1	9
(10k-30k)	0	0

After assigning all values, we will sum all values and multiplied that with 9. The final value will be the credit score and it will determine a consumer’s creditworthiness.

We will generate our Credit score by calculating the value of our selected feature. It’s not guaranteed that a good credit score always ensures your loan request, it will improve the probability of getting loan with low interest. There will be some other condition to fulfill the loan requirements. Here, at first we will generate a person’s credit score, if its low then it will be a

bad credit, else we will then focus on the amount of loan, if the value of income is higher than the value of loan amount, then we could declare it's a good credit. Otherwise we will check the installment rate, is it higher than the cumulative sum of asset and income. If it is higher than that, we will consider it as a bad credit request. If the condition is false then we will check the time consumer will take to repay the loan is lower than T or not, if it's higher than T then it's a bad loan request, else its good credit request and the loan will be accepted. Here, T is the Standard deviation of time in our dataset.

Output credit score will be like this figure below one and it will be associated with a verdict too.

Q	R	S	T	U	V	W	X	Y	Z
expenses	incomeR	assetsR	amountR	residence	savingsR	inqlast6m	redit Scor	Verdict	
0.25	0.5	0	0.75	4	0.75	1	459	SM	
0.75	0.5	0	0.5	2	0.75	5	504	MG	
0	1	0.25	0	3	0.25	2	567	MG	
0.25	0.75	0.25	0.75	4	1	1	441	SM	
0.75	0.25	0	1	4	1	0	423	SM	
0.25	1	0.5	0.75	4	1	3	540	MG	

Fig 17: Credit score and verdicts

6.2 Predicting Loan Defaults using a decision tree

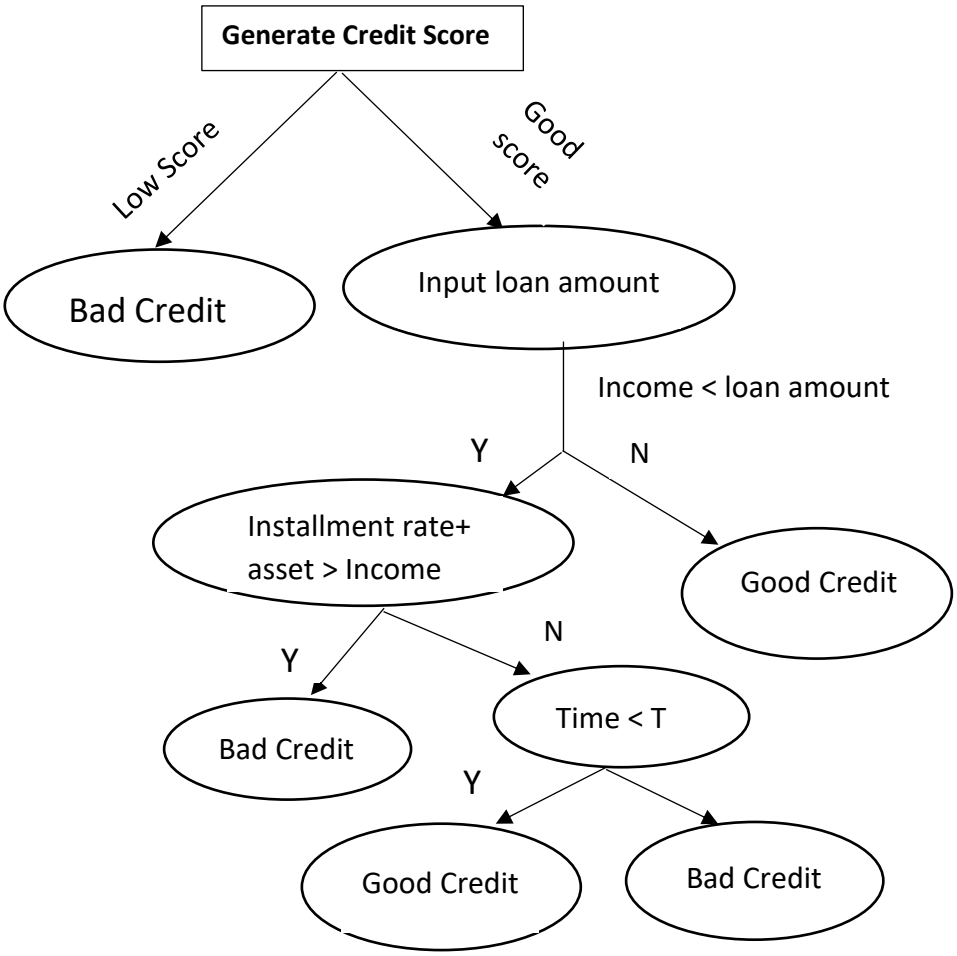

Fig18: Loan default prediction using the credit score.

We will generate our Credit score by calculating the value of our selected feature. It's not guaranteed that a good credit score always ensures your loan request, it will improve the probability of getting loan with low interest. There will be some other factors to fulfill the loan requirements. By a decision tree we will explain how to identify a good credit request and a bad credit request. Here, at first we will generate a person's credit score, if its low then it will be a bad credit, else we will then focus on the amount of loan, if the value of income is higher than the value of loan amount, then we could declare it's a good credit. Otherwise we will check the

installment rate, is it higher than the cumulative sum of asset and income. If it is higher than that, we will consider it as a bad credit request. If the condition is false then we will check the time consumer will take to repay the loan is lower than T or not, if it's higher than T then it's a bad loan request, else its good credit request and the loan will be accepted. Here, T is the mean of time in our dataset. For example, in our model one person has a credit score of 567, that's indicate marginal, now if we focused on our tree we found loan amount=2000, income=200, Assets=3000, Time=60days, T.mean =46.5, installment rate =1000. Now the measured credit score is marginal, now we will check (income >Amount) or not, higher income indicates more trustworthiness in our country, here (income 200 < amount 2000), now we will check installment rate is lesser than the sum of income and assets. Here, installment is lesser (1000<3200). Now we will check time for repaying the loan is lesser than T.mean or not. Her time is higher, so that it's a bad credit request. Requesting for more time to repay the loan is increasing the more risk in loan request. Every model is not hundred percent accurate, Here, for model we got an accuracy of 79.42%, to improve this accuracy and better loan prediction we used above decision tree.

Chapter 07

Conclusion and Future Works

7.1 Conclusions:

Through our research we have tried to accomplish the proposed model to predict Credit Defaults. A description of the ‘German credit Data’ has been given earlier as a basis for presented evaluation. Even though the Binary structure and clarity of the classification rules imply an advantage of our dataset in correspondence to the research yet some of the well-known machine learning technique didn’t quite perform up to the mark.

So here is some of our research Findings: Here we have observed that Logit Boost can outperform any other machine learning models for a balanced data set containing a medium range of data with a significant prediction accuracy of almost 81%.

Besides this, Gradient Boosted Trees(XGBoost), Multilayer perceptron and SVM also performed well. Although some significantly popular Machine learning models could not provide a satisfactory outcome; for example, Decision tree and Naïve Bayes.

After finding these, our sole attempts were on to building a scoring Format using which we can output Credit scores. Credit scores were generating by multiplying a weights derived from Information gains to all attributes of an individual maintaining a particular range which would output score both in a range of 100 and in a range of 900.

After that we predicted the credit default risk after developing a decision tree which used the credit score along with some other important factors to determine whether an individual should receive the loan or not.

The Main challenges were to collect the data and process it. After investing a long time in trying to collect the real time data from any commercial bank, we realized we would fall short of time to complete our research work in time. That is why we opted for the “German Credit Data’. This dataset was processed multiple times to obtain a balanced dataset.

We believe that our proposed model will be of great help in credit default prediction and Individual credit scoring in this country since no other such system right now exists in our country and conclude our research work hereby.

7.2 Future works:

There are several further lines of investigation in our research. The priority of our work will be to represent a Ranking of classifiers. we are planning to test the hypotheses based on our classifier ranking and the hypotheses will also be proposed by us based on our literature review. As a part of futuristic approach and for creating a more reliable model, we will also try to evolve our model in a way that it will retrieve necessary data from social networks like twitter, Facebook (if possible) etc. Furthermore, we will try to test our model considering different ranges of time of consideration of data. Since most of nowadays studies are based on data from online repositories like UCI, Kaggle etc. at the preliminary phase we too will train our model with data collected from online repositories, later on we will collect the real time data. Since the datasets from online repositories are often from unclear origins, complex and rather found in small sizes so it will be better if we can collect required data first hand from commercial banks.

References

[1] Baesens, B., T. V. Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society. 54 (6), 627–635.

[2] Abdou, HAH and Pointon, J. (2011). Credit scoring, statistical techniques and evaluation criteria: A review of the literature, Intelligent Systems in Accounting, Finance & Management. 18 (2-3), pp. 59-88.

[3] Hand, D. J. and W. E. Henley. (1997). Statistical classification methods in consumer credit scoring: A review. Journal of the Royal Statistical Society. Series A 160 (3), 523–541.

[4] Ala'raj ,M. ,Abbod, M. (2015). International Symposium on Innovations in Intelligent SysTems and Applications (INISTA).

[5] Khandaniy, A E. Kimz, AJ. and Lox, A W. (2010). This Consumer Credit Risk Models via Machine-Learning Algorithms.

[6] Bellotti, T. and Crook, J. (2007). Credit Scoring with Macroeconomic Variables Using Survival Analysis.

[7] Joao A, Bastos Cemapre. (2008). “Credit scoring with boosted decision trees”. School of Economics and Management (ISEG) Technical University of Lisbon. Portugal.

[8] Anne Kraus. (2014). Recent Methods from Statistics and Machine Learning for Credit Scoring. Munich.

[9] Masyutin A.A. (2015). Credit scoring based on social network data. Business Informatics. (No. 3 (33). P. 15–23.

[10] Saxena, Rahul. (February 6, 2017). How the Naive Bayes Classifier works in Machine Learning. Retrieved from http://dataaspirant.com/2017/02/06/naive-bayes-classifier-machine-learning/

[11] H. Zhang (2004). The optimality of Naive Bayes. Proc. FLAIRS. Retrieved from http://scikit-learn.org/stable/modules/naive_bayes.html

[12] Wikipedia, the free encyclopedia. (18 July 2018). Conditional probability.Retrieved from https://en.wikipedia.org/wiki/Conditional_probability

[13] Anne Kraus and Helmut Küchenhoff , vol 8,Number 1; pages 31-67; March 2014.

[14] Bastos. J.A. (2008). Credit scoring with boosted decision trees. Page 4.

[15] Navlani, Avinash. (2018). Support Vector Machines with Scikit-learn. Datacamp Community. Retrieved from https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python.

[16] M. Kubat and S. Matwin. (1997). “Addressing the curse of imbalanced training sets: one-side selection,” in Proc. of the 14th International Conference on Machine Learning k2(ICML1997), pp. 179–186.

[17] Yuchun Tang, Yan-Qing Zhang, Nitesh V. Chawla, Sven Krasser. (2002). SVMs Modeling for Highly Imbalanced Classification. JOURNAL OF LATEX CLASS FILES, VOL. 1, NO. 11.

[18] Matthews, B. W. (1975). "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". *Biochimica et Biophysica Acta (BBA) - Protein Structure*. 405 (2): 442–451.