

Comparative Analysis and Implementation of Credit Risk Prediction Through Distinct Machine Learning Models

by

Aquib Abtahi Turjo

17101073

S.M. Mynul Karim

17101162

Tausif Hossain Biswas

17101374

Yeaminur Rahman

17101406

Ifroim Dewan

17126016

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

Department of Computer Science and Engineering
Brac University
June 2021

© 2021. Brac University
All rights reserved.

Declaration

It is hereby declared that

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

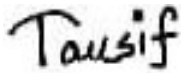
Student's Full Name & Signature:



Aquib Abtahi Turjo
17101073



S.M. Mynul Karim
17101162



Tausif Hossain Biswas
17101374



Yeaminur Rahman
17101406



Ifroim Dewan
17126016

Approval

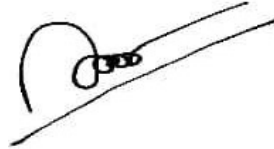
The thesis/project titled “Comparative Analysis and Implementation of Credit Risk Prediction Through Distinct Machine Learning Models ” submitted by

1. Aquib Abtahi Turjo (17101073)
2. S.M. Mynul Karim (17101162)
3. Tausif Hossain Biswas (17101374)
4. Yeaminur Rahman (17101406)
5. Ifroim Dewan (17126016)

Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 02, 2021.

Examining Committee:

Supervisor:
(Member)

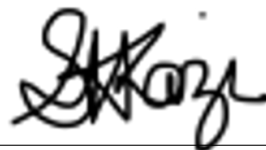


Muhammad Iqbal Hossain, PhD
Assistant Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

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

Head of Department:
(Chair)



Name of Head of Department
Designation
Department of Computer Science and Engineering
Brac University

Ethics Statement

It is hereby proclaimed that all the comparisons made and conclusion drawn in this thesis are based on the findings obtained from the group's individual research on the topic. All relevant resources and the dataset used have been acknowledged through proper citation. Appropriate steps have been taken to ensure the transparency of the analysis. This thesis has previously not been submitted in any form to any other institute.

Abstract

Predicting the risk while lending money has always been a challenge for financial institutions. To make such decisions many banks or financial organizations follow different techniques to analyze a set of data. Manual prediction and analysis of credit risk can not only be very hectic but also quite time-consuming. To solve this issue, what is needed is a system that ensures high predictive accuracy and optimality. Machine Learning algorithms such as various Regression models, Gradient Boosting, Deep Learning, Neural Networks, Support Vector, Random Forest and others can be used to anticipate whether a consumer is eligible for taking a loan with high accuracy. In this thesis, an attempt has been made to find a good ML algorithm that shall help various banks and/or financial institutions to reliably predict the credit risk on an individual by analyzing appropriate datasets. Following that, a highly accurate result for said institutions can be ensured, which they can use to determine whether a consumer requesting credit should be allotted credit or not.

Keywords: Credit Risk, Loan, Machine Learning, Regression Model, Gradient Boosting, Deep Learning, Neural Networks, Support Vector, Random Forest.

Dedication

This thesis has been dedicated to our loving parents and our respectful faculties from our department, who have helped and motivated us throughout the entire journey, and who inspire us to achieve excellence.

Acknowledgement

First and foremost, we would like to show our eternal gratitude to the Almighty Allah (S.W.T) for His blessing and His grace towards us as we moved towards our goal of completing this thesis.

We would also like to state and acknowledge that it has been a great honor for us to have Mr. Iqbal Hossain as our supervisor. His proper guidance, continuous support, constructive criticisms and appropriate suggestions have proved to be invaluable assets in our research.

Lastly, we would like to thank all of our family members and friends who believed in our capabilities and supported us whenever we despaired and helped us move forward towards our objective.

Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iii
Abstract	iv
Dedication	v
Acknowledgment	vi
Table of Contents	vii
List of Figures	ix
List of Tables	x
Nomenclature	xi
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	1
1.3 Objective and Contributions	3
2 Background Analysis	4
2.1 Literature Review	4
2.2 Algorithm Description	7
2.2.1 K-Nearest Neighbor	7
2.2.2 Logistic Regression	8
2.2.3 Linear Regression	10
2.2.4 Random Forest	10
2.2.5 Gradient Boosting	11
2.2.6 XGBoosting	13
2.2.7 AdaBoost	14
2.2.8 Artificial Neural Network	15
3 Proposed Model	17
3.1 Workflow Overview	17
3.2 Dataset Description	18

3.3	Dataset Pre-processing	20
3.4	Feature Selection	23
3.5	Dataset Limitations	24
4	Result Analysis & Comparison	25
4.1	In-depth review of results achieved	25
4.2	Comparison	34
5	Implementation	36
6	Conclusion & Future Works	42
	Bibliography	45

List of Figures

2.1	K-NN Algorithm Example.	8
2.2	Logistic regression curve example.	9
2.3	Linear Regression Example.	11
2.4	Gradient Boosting Example.	13
2.5	Total Error Curve.	15
2.6	Neural Network Layers.	16
3.1	Workflow Model.	18
3.2	Feature Importance for Random Forest.	21
3.3	Feature Importance for AdaBoost.	21
3.4	Feature Importance for Gradient Boosting.	22
3.5	Feature Importance for XGBoost.	22
3.6	Visual representation of feature correlation through a Heatmap.	23
3.7	Feature Correlation Matrix.	24
4.1	Layout of confusion matrix.	25
4.2	ROC for Logistic Regression.	27
4.3	ROC for Random Forest.	28
4.4	ROC for K-Nearest Neighbor.	29
4.5	ROC for Artificial Neural Network.	30
4.6	ROC for AdaBoost.	31
4.7	ROC for Gradient Boosting.	32
4.8	ROC for XGBoost.	33
5.1	Flowchart showing the workings of the web application.	37
5.2	Inputs to be provided.	38
5.3	Scenario 1: The user is eligible for getting a loan.	39
5.4	Scenario 2: The user is not eligible.	40
5.5	Scenario 2: Improvements suggested in order to be eligible.	41

List of Tables

3.1	Feature Description.	19
4.1	Performance evaluation of algorithms used.	34
4.2	Accuracy comparison with relevant reference works.	35

Nomenclature

The next list describes several symbols & abbreviation that have later been used within the body of the document

AdaBoost Adaptive Boosting

ANN Artificial Neural Network

GB Gradient Boosting

KNN K-Nearest Neighbors Algorithm

ML Machine Learning

RF Random Forest Algorithm

ROC Receiver Operating Characteristic Curve

XGB eXtreme Gradient Boosting

Chapter 1

Introduction

1.1 Motivation

Financial institutions usually categorize consumers into two criteria. First are those who can pay back the loan they have received and second are those who cannot. A third might be included which consists of people who can at the very least pay back the collected sum either through a timed extension, but what's important is that creditors belonging to the second category might prove to be disastrous for an institution such as a bank if the numbers are relatively quite high. Moreover, considering the ongoing pandemic around the globe caused due to COVID-19, the current world is already suffering an economic crisis. According to Eusuf M.A, the global FDI has already been decreased by 35% which is quite substantial to say at the very least and definitely something not to scoff at [1]. And to add to that, it is also predicted that in the near future, the world will face yet another economic recession should situations not improve. In that regard, an algorithm that can analyze credit risk with impeccable accuracy can indeed be one of the most useful systems in this hazardous situation. By using different algorithms and datasets, banks can predict the probability of whether the consumer can pay back a received loan or not with incredible accuracy and in optimal time. Now, certainly leaving something this important in the hands of a machine can feel quite daunting, but the accuracy and predictive ability of various ML algorithms have been proven to be quite credible in recent time and therefore it can be said with considerable assurance that Machine Learning is without a shadow of a doubt the way to go for countering the possibility of the world economy possibly falling into a state of collapse.

1.2 Problem Statement

The procedure of calculating a person's credit score in the form of a numerical expression obtained through analyzing their credit files to determine their credit-worthiness, thus predicting the probability of the individual's capability to repay the loan taken, is called Credit scoring.

The method of credit scoring was introduced in the 1950s, and it is used in many spheres of banking in the present times [2]. In order to obtain the credit score, information of the loan applicant is gathered from their loan applications and from historical credit bureaus which may include the applicant's salary, job

status, residence, previous statements, account type, etc. Using regression analysis, the loan performance of the applicant can be determined by using the collected information as variables. The more variables are introduced, the more reliable the data. However, it is seen that only a few of the variables end up being the most prominent ones in influencing the result of the scoreboard of the analysis. Hence according to Fair, Isaac and Company Inc., a leading developer of scoring models, it can be found that even after taking 50-60 variables in the beginning, only 8-12 of them end up in the final scoreboard [3]. Thus, using the necessary variables, the credit score of the applicant can be determined, where a higher score mostly indicates lower risk and vice versa. Following the credit score, the lender sets a cutoff score depending on the amount of risk it is willing to take. Now, all of this begs the question as to why exactly is credit scoring so important and what is its usefulness? To put it in simpler words, credit scoring is the method that allows banks and other such financial institutes to predict a borrower's ability to return what he/she is owed and therefore allows said institutes to determine whether lending money to a person is wise or not. In other words, in the world of economy and finance, it is a rather convenient way to determine the risk associated with a specific individual and in turn is not only useful in terms of reducing any form of losses financial institutions may incur but also to improve upon their previous results, ergo making it a necessity in financial areas [4].

The common statistical methods used to develop such a system are linear regression, linear probability, logistic regression, discriminant analysis, and other models. There are also newer models being applied to determine the risk, such as the options pricing theory model and neural networks. Neural networks do not work based on assumptions, rather they determine the important factors for predicting default by learning through experience. Therefore, this method is better and more flexible than standard statistical scoring methods.

Currently, credit scoring is being used by many banks for loans under \$100,000 in most cases [5]. The use of credit scoring can be seen more widely in credit card loans, organizational loans, digital financing, and renting. However, its adoption has been slow for business loans due to the lack of homogeneity and insufficiency of data to prepare a suitable model.

Here the main focus is on the assessment of the credit risk of loan borrowers. The number of loan defaulters has been on the rise in recent times. This is hampering transactions and causing usable assets to be frozen, thus causing major losses to the financial institutions across the globe. China alone had about 9.2 million loan defaulters in 2018 [6]. In India, the loans defaulted quadrupled within the span of 2013 to 2017 [7]. Around a million American students are defaulted annually [8]. The situation is not very bright in Bangladesh also. The number of defaulted loans has tripled here since 2011 [9]. Economists are hinting at this situation as a major reason for hampering the growth of business and employment opportunities [10].

In order to mitigate the current scenario, a plausible solution would be to filter out the loan applications of those with a higher risk of loan defaulting. To do so, the method applied should have a pattern recognition approach, which should be

able to determine the pattern of a generic defaulter within a vast collection of user data. This very concept is best implemented by applying machine learning. To prepare a model for predicting credit defaulters, several algorithms have already been implemented such as genetic algorithm, k nearest algorithm, gradient boosted regression, random forest regression, etc. [11] [12] [13]. Also, sometimes a hybrid of two algorithms can also be implemented in order to get a more streamlined and personalized result. However, before implementing any certain model, a comparative study should be conducted in order to determine its effectiveness in creating a credit risk analysis system with the best accuracy while maintaining proper simplicity and accessibility, it is also imperative that the structure of the dataset be kept in mind while implementing a model as depending on how the dataset has been constructed and what the type of the target variable is, the results may vary drastically.

1.3 Objective and Contributions

This particular thesis intends on finding and implementing appropriate Machine Learning algorithms that can predict a creditor's capability to properly repay any form of credit taken within the given time frame or at the very least with a timed extension which should help various credit offering institutions to possibly avoid fraudulent creditors and therefore subsequent bankruptcy. For this purpose, a reliable dataset collected from a reputable source has been used to train a select few algorithms and after analyzing the results, a declaration has been made in regards to what is believed to be proper algorithms appropriate for the job at hand.

It is being hoped that this thesis can contribute to making a necessary difference in the struggle to avoid an economic collapse on a worldwide scale and help ease the tension regarding financial stability through these trying times.

Chapter 2

Background Analysis

2.1 Literature Review

In the world of finance and banking, the significance of Credit Risk Analysis is quite substantial, to say the least. Throughout history, the study of various statistics and human evaluation has been at the core of this process. However, due to the exceptional advancements in data science and technology namely machine learning and artificial intelligence not only has credit risk analysis become quite the topic for research but the research itself has also become significantly more dynamic. There are loads of research papers written on the use of machine learning for the estimation of credit risk. Algorithms such as Decision-tree based algorithms, K-Nearest Neighbor, Linear Regression, Gradient Boosting, Random Forest, Neural Network, etc. have all been used with varying results some of which have displayed better accuracy than others. J. Galindo and P. Tamayo have compared the performances between CART (Classification and Regression Trees), Neural Networks and K-Nearest Neighbor model for credit risk assessment in their paper where they used data on home mortgage loans provided to them by Mexico's security exchange and banking commission: Comision Nacional Banacaria y de Valores (CNBV). The dataset had approximately 4,000 records each corresponding to a customer account and contained a total of 24 attributes. It was already being used by CNBV for a regression model and therefore only required a little amount of data pre-processing before they started applying their preferred algorithms on it. After the predictions were made by the three selected algorithms, they recorded the results in both tabular and graphical form, did a side by side comparison, and concluded that a CART with 120 nodes was the most accurate on the largest sample of 2,000 records compared to the test error average from Neural Network (60 hidden nodes trained for 80 iterations) with the same amount of samples and K-Nearest Neighbor (24 neighbors) with a largest sample of 1,000 records. However, they have also aptly noted that in order for CART to perform optimally it would need at least 22,000 records but in its defense, that number is quite acceptable when it comes down to building a production-quality predictive risk model for an institution such as CNBV [14].

In the study conducted by Xiaomeng Ma and Shuliang Lv they have used a combination of logistic regression model and an improved machine learning model which they are calling MLIA algorithm. It is a decision tree-based algorithm

which adopts the basis function otherwise known as a weak classifier of the lifting algorithm. According to them, a decision tree algorithm is quite appropriate for analyzing credit risk because it has over the years seen widespread use, has low computational complexity and the output provided is easy to understand. An Internet finance company had provided them their sample data. However, they have decided to keep the name of the company private and have only referred to said company as Company A. They then took the data provided by Company A, split them into modeling samples and cross-time window verification samples and matched these sample with the account data of an Internet search company which they have also not mentioned the name of and have only referred to as Company B. After matching the data, the “good guy” users are listed as users who have no overdue repayments and the “bad guy” users are listed as users who have overdue repayment for 60+ days. Now, for the machine learning algorithm MLIA itself, in order to showcase its performance what was used in this paper are three common test functions known as Schwefel function, Rastrigin function and Greiwank function to test the algorithm and compare the result with the logistic prediction model. Each test function is iterated 1,000 times after which the average fitness value and the optimal fitness values are used as the basis of comparison. After the results were found what they concluded was that MLIA is quite appropriate for the underlying original search words and is quite capable of effectively capturing the credit risk of the Internet users listed in the dataset based on their search terms. On the other hand, the variable coefficient of the logistic regression model can reveal the contribution of the variable to the prediction result with relatively more intuition. So, by combining these two models a rather robust model stability, proper scalability and competitive computational efficiency can be achieved at a comparatively low time cost [15].

According to Pandey T.N , the Bayesian Classifier method can be used in finding the probability of returning the loan [16]. In this method, some strategies from DAG (directed acyclic graph) are used. Where the nodes represent some random variable (for the datasets) and edges represent the dependencies of those nodes. And the accuracy depends on how well the dependencies are joined in the network with the datasets. Another modified version of the Bayesian classifier is called Naïve Bayesian classifier. This method represents the attributes of the datasets as independent variables. Thus, it requires less amount of dataset. Again, there is another method called KNN (K-nearest neighbor). In this method, it is used in a nonparametric way. It works with training sets in which it is divided into two cases (Positive and negative). And the functionality is divided into two phases: the training phase and the testing phase. In the training phase, this method does not generalize the training points and in the testing phase, it calculates the Euclidean Distance among the training points. By using regression and making instances, the highest similar instance is taken as the output. There are also some other methods used in this research work which are K-Means, multilayer perceptron, support vector machine, and extreme learning machine.

Mentioning Elastic Net algorithm, Addo P.M has come up with another path to find the defaulters in credit risk analysis [17]. In this method, some extensions of linear regression are used. This method possesses logical and multinomial functions and a

good error checking mechanism which reduces the error and increases the accuracy. Working with an estimation procedure and a choice of model this method gives out a good accuracy regarding returning the loan. By creating a graph where x is denoted as predictors and y as response variables, elastic net penalty is detected. 2 equations of elastic net algorithm are used in this whole process. Again, with the help of other algorithms like gradient boosting machine, Random forest modeling; credit risk analysis has been done in this paper.

Banks are looking for more effective ways for evaluating and analyzing credit risk in order to comply with the Base II standardized approach for credit risk. Hence in order to efficiently predict risk while complying with the necessary standards, it is necessary to select the proper models for risk analysis and also to segregate the relevant data and impacting factors behind the risk calculation. As described in a paper by Danenas, P., Garsva, G., and Gudas, S., an effective technique for such a case would be Support Vector Machine or SVM [18]. The key advantage of SVM over other options involving artificial intelligence would be that the solution derived through SVM would not be trapped in the local minima. In order to implement such a method, it is necessary to identify the special data points which are to be used as the support vectors in the solution.

According to Baesens B., LS-SVM and Neural Network algorithms have better performance in terms of building a credit scoring system [19]. In their study they used three strategies to implement SVM for credit scoring. In addition to that, they tested the accuracy of the SVM by using two UCI datasets. The accuracy of this method is almost similar to decision tree and neural network methods and the advantage is SVM uses less input features. Moreover, by using genetic algorithms with SVM (GA-SVM) the model parameters can be optimized. In their paper of Oreski G. they have used a hybrid HGA-NN algorithm, that is a combination of genetic algorithm and neural network [20]. They have used fast filter technique for the initial stage of feature ranking. Furthermore, improvements are made to establish the initial population including an incremental stage in the genetic algorithm. This hybrid algorithm can increase the expandability and accuracy of the credit risk assessment. Moreover they evaluated the algorithm by using real life data of Croatian Bank. They have used a fast filter technique for the initial stage of feature ranking.

The applicable SVM classifiers for determining credit risk are near LIBLINEAR, stochastic gradient based Pegasos and SGD, LibSVM C-SVC, mySVM, SMO, SVMLight, Core Vector Machines (CVM) and Ball Vector Machines (BVM). Using these classifiers, a desirable output can be generated by following the widely applied discriminant Altman technique. The efficiency of this methodology depends on the selection of the necessary parameters. Moreover, the effectiveness of this method can be further improved by combining it with other algorithmic techniques like genetic algorithm or particle swarm optimization. However, it is important to keep track of the proper balance of accuracy and complexity while developing the system.

A paper by Islam, S. A., Aziz, R. S., Ahmed, A., and Abida, F. deals with the analysis of the risk of credits [21]. It mainly concerns with giving each individual

a credit score which would represent the creditworthiness of a said individual. The Credit Score mainly plays a vital role in the paper to determine the risk of the credits being provided by banks and other financial institutions for an individual's personal or business needs. The purpose of this paper was to have a robust software mechanism that ensures the credit scores were calculated correctly being built up with a trustworthy algorithm that can run effortlessly. The results were conclusive and imitated the process of evaluating an individual precisely and accurately. The accuracy scores of the algorithm were found to be in the lower ninety percent. The proposed model for the solution was based on CART (Classification and Regression Trees) using the Gradient Boosting Method (GBM); they also included another machine learning algorithm, namely XGBoost which solves a wide range of problems such as problems of regression ranking and user-defined problems. Furthermore, the paper itself proposed a hybrid model that is a two-step architecture. The workflow diagram that they proposed could be implemented in production to provide a concrete base for the evaluation and prediction of defaulters. Simultaneously provide a detailed overview of the results obtained. This could help financial institutions immensely and help them save millions lost by default loans.

Lastly, after looking into a paper by Lawi, A., Aziz, F., and Syarif, S., it has been found that they have used the Generalized Linear model algorithm which is a modification of the logistic regression model [22]. Logistic regression is a classification algorithm that generates a binary response when given a set of independent variables. An improvement of this method is GLM which provides confidence bound with the possibility of a positive outcome. A relatively high predictive confidence was achieved by their model, while their overall and average accuracy was found to be respectably high as well. Further improvements were proposed by Nalic, J., and Svraka, A., who used Ensemble Logistic Regression boosted by Gradient Boost on German and Australian datasets found in UCI machine learning repository and they managed to acquire decently high accuracies as well [23][24].

2.2 Algorithm Description

2.2.1 K-Nearest Neighbor

For classification and regression, K-Nearest Neighbor is one of the fastest algorithms. It is a non-parametric algorithm that works with train-test sets and does not make any assumptions. Considering both negative and positive cases of training sets, it provides output as a regression or a classification. This algorithm is known as a lazy algorithm because it does not require any training data points for generalization. Hence this makes the algorithm fast at the training phase while considering all the data points for the testing phase. In this case, k-instances are to be found conducting training datasets. And to find the real valued input variable, Euclidean Distance calculation is used.

$$Euclidean\ distance : d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

In this case, when a particular point is being classified, the distance between that point and the randomly selected K-neighbor points are measured with the Euclidean Distance formula. Then, value to the k is assigned (usually an odd number), and all the measured distances are sorted. After that, the class having a greater number of votes as per in the sort queue will possess the random point. The optimal value of k can easily be calculated by rooting the total number of neighbors and taking the integer only. And for the regression task, the mean or median values of k-instances takes the decision.

Let categories 1 and 2 be two clusters of points in the data, and k be the randomly selected datapoint. Figure 2.1 shows how the datapoint k is represented while being possessed by the cluster with more neighbors, which in this case is category 2

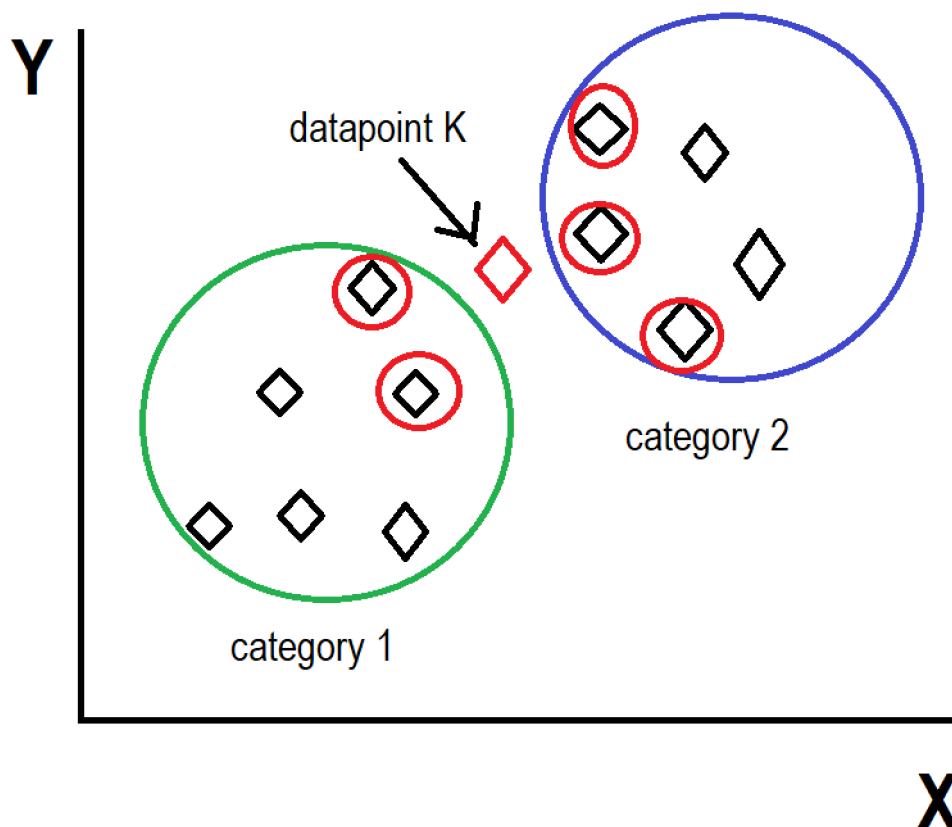


Figure 2.1: K-NN Algorithm Example.

2.2.2 Logistic Regression

This algorithm falls under supervised learning that deals with nonlinear functions. It can work with both discrete and continuous values but always provides output

as a discrete result. Binary logistic regression, multinomial logistic regression, and ordinal logistic regression are the three forms of logistic regression. Tasks based on true/ false decision making are done with binary logistic regression and if there are more than 2 variables, it falls to multinomial logistic. And the ordinal logistic regression deals with order or review-based tasks.

For binary logistic regression, it uses a sigmoid function. By this, a curve is derived which outputs a probability. And it determines in which region (0 or 1) the elected point is closer to. Although it is named regression, its main task is to classify a point.

$$\text{Sigmoid function : } S(x) = \frac{1}{1 + e^{-x}}$$

$$\text{Probability function : } \frac{1}{1 + e^{-(\beta_0 + x\beta_i)}}$$

From the equations, it can be found whether the predicted output is in the upper or lower threshold. In addition, the function deals with conditional probability to find the predicted output. The curve of the function has been shown below in figure 2.2. Here the position of the threshold value on the curve determines its probability of occurrence in either of the regions.

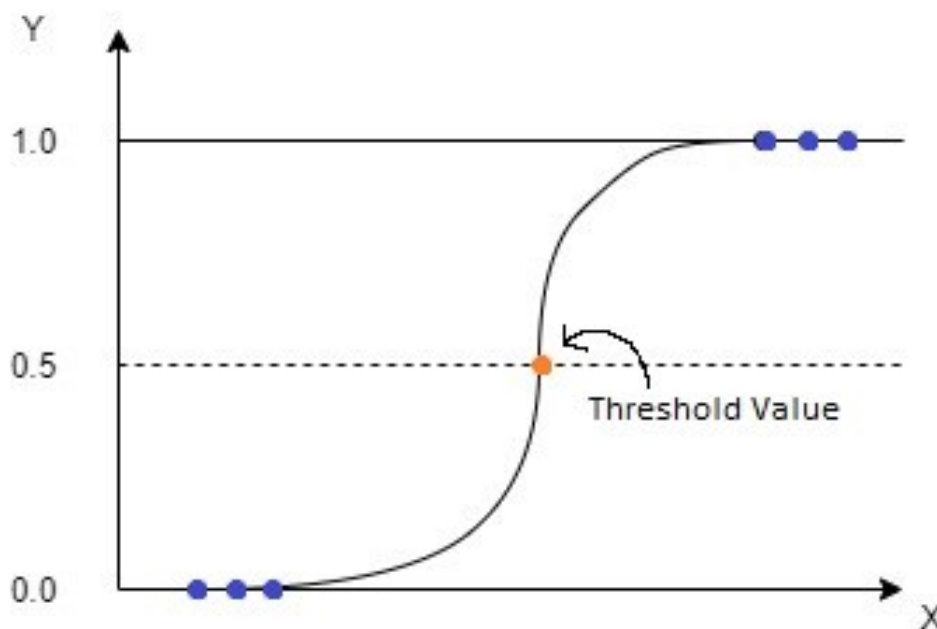


Figure 2.2: Logistic regression curve example.

2.2.3 Linear Regression

Linear regression is the next step after correlation; being a machine learning algorithm it predicts the value of a variable depending on the value of another variable. These variables might be dependent or independent. The dependent variable in a regression is known by a variety of names, including outcome variable, endogenous variable, regressands, criterion variable, and so on. Exogenous variables, predictor variables, or regressors are all terms for independent variables. The function for the linear regression is given below:

$$y = ax\theta_1 + \theta_2$$

The variables x and y are given while the model is being trained.

- x : input training data (univariate – one input variable(parameter/s) or multivariate)
- y : labels to data (supervised learning)
- θ_1 : intercept
- θ_2 : coefficient of x

Figure 2.3 shows how the function for linear regression is represented among the scattered data points plotted.

2.2.4 Random Forest

Random forest consists of numerous individual decision trees, and it functions as an ensemble for the trees. A class prediction is obtained from every individual tree in the random forest and the highest chosen class becomes the model's prediction. The low correlation between models is essential, where it is found that the accuracy of the prediction of the ensemble is higher than the collection of the individual predictions. Higher accuracy is obtained through cross-validation. A random forest classifier can maintain the accuracy of a large proportion of data while handling the missing values. It has the capability of handling a large data set with higher dimensionality.

Therefore, the two features that are required for random forest to operate are as follows:

- Features having at least some predictive abilities are required
- The trees of the forest and more importantly their predictions need to be uncorrelated

After training, averaging the predictions from all the various regression trees on x' can be used to make predictions for unseen samples x :

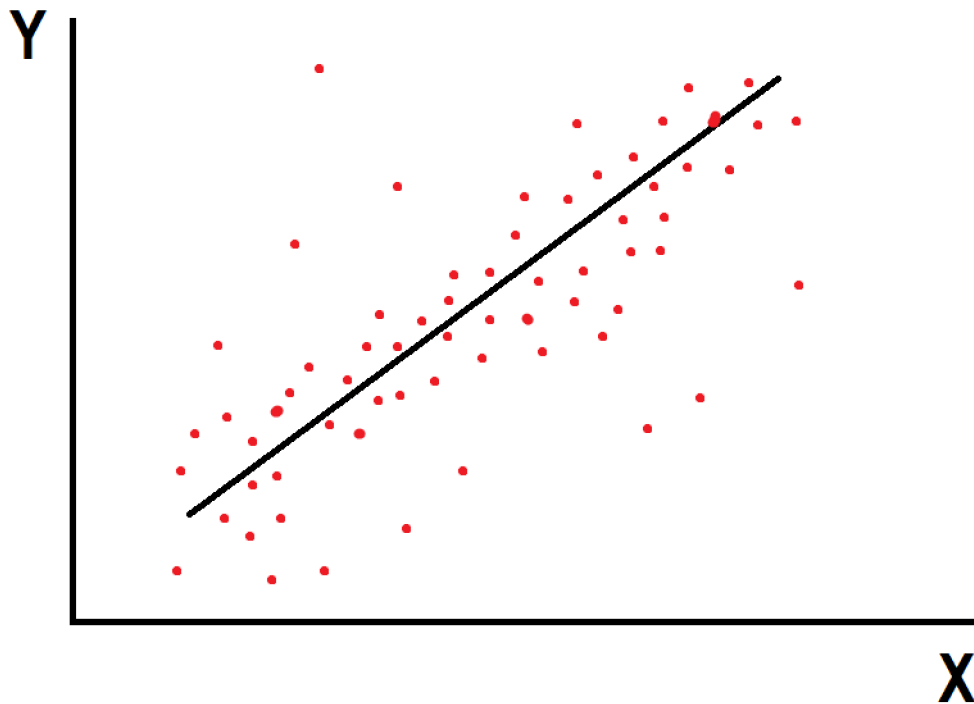


Figure 2.3: Linear Regression Example.

- Sample, with replacement, n training examples from X, Y ; call these X_r, Y_r .
- Train a classification or regression tree fr on X_r, Y_r .

$$f^n = \frac{1}{R} \sum_{r=1}^R f_r(x')$$

The standard deviation of the predictions from all of the separate regression trees on x' can be used to estimate the prediction's uncertainty.

$$\sigma = \sqrt{\frac{\sum_{r=1}^R (f_r(x') - \hat{f})^2}{R - 1}}$$

2.2.5 Gradient Boosting

Gradient Boosting is a sequential and gradual training method. In Gradient Boosting the original data is randomly sampled and is sent to the first model. After the predictions from the first model, the residuals (errors) are collected. Then the loss between the targeted value and the predicted value is compared to try and

learn if there are any patterns in the errors found. These errors are used to improve upon the models, being focused solely on the areas where the performance is not up to par. Every residual collected is modelled until a threshold is reached where the residuals are very close to zero thus resulting in a very low error rate between predicted and target variables. While this is being done it has to be ensured that the model is not being over-fitted, which is one of the issues of GB. This initial prediction is taken and modified (Summed up) based on the learnings from the errors (Models Based on the Errors).

f_0 is the initial guess, $\phi_m(x)$ is the base estimator at iteration m and θ_m is the weight for the m^{th} estimator in the gradient boosting algorithm.

$$f(x) = \sum_{m=0}^M f_m(x) = f_0(x) + \sum_{m=1}^M \theta_m \phi_m(x)$$

In the m^{th} iteration, the direction of the steepest descent is given by the negative gradient of the loss function:

$$-g_m(x) = - \left[\frac{\partial L(y, f(x))}{\partial f(x)} \right]_{f(x)=f^{(m-1)}(x)}$$

Typically, as a surrogate loss, the squared error is used, $\varphi_m(x)$.

$$\varphi_m = \arg \min_{\varphi} \sum_{i=1}^n [(-g_m(x_i)) - \phi(x_i)]^2$$

Further effort is necessary to determine the step length ρ_m .

$$\rho_m = \arg \min_{\rho} \sum_{i=1}^n L(y_i, f^{(m-1)}(x_i) + \rho \varphi_m(x_i))$$

The step taken at each iteration m is given by

$$f_m(x) = \eta \rho_m \varphi_m(x)$$

The resulting model can be written as

$$f(x) = f^{(M)}(x) = \sum_{m=0}^M f_m(x) = f_0(x) + \sum_{m=1}^M \eta \rho_m \varphi_m(x)$$

Where f_0 is typically initialized using a constant.

$$f_0(x) = \theta = \arg \min_{\theta} \sum_{i=1}^n L(y_i, \theta)$$

Figure 2.4 illustrates how data is sampled and processed in Gradient Boosting by a workflow diagram.

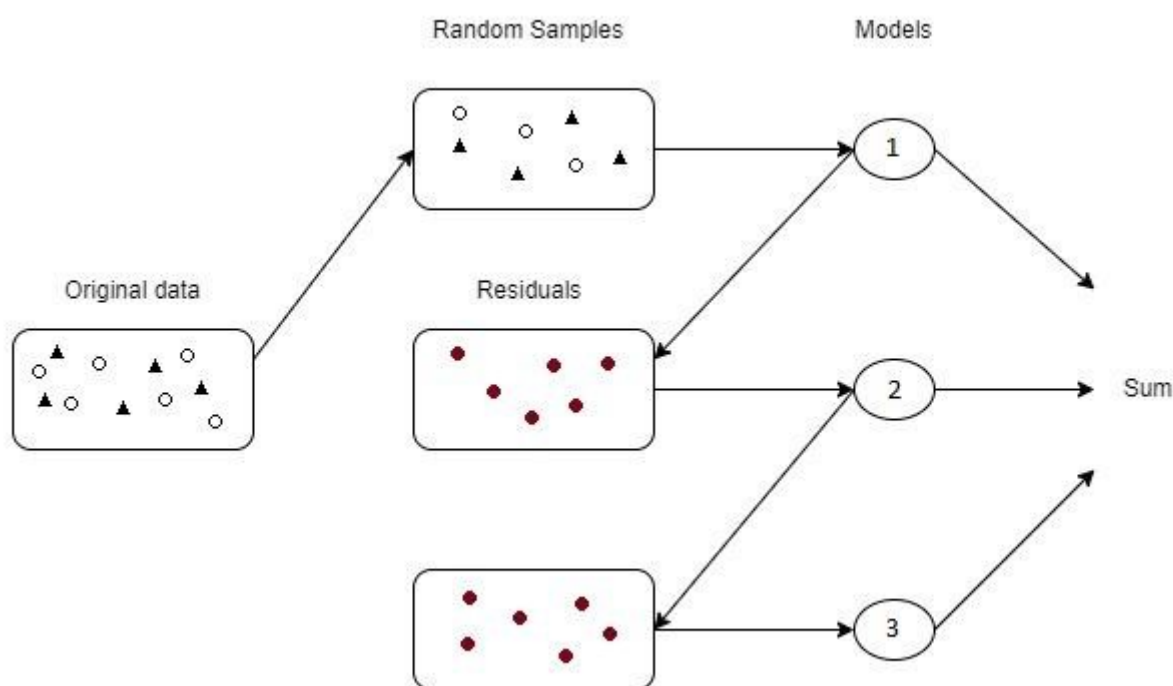


Figure 2.4: Gradient Boosting Example.

2.2.6 XGBoosting

XGBoosting, stands for eXtreme Gradient Boosting, being one of the most powerful Machine Learning algorithms to date. Its ease of use whilst being reliable, fast, and computationally efficient makes it one of the key algorithms for credit risk analysis. Being another implementation of gradient boosting algorithm just being extreme is what sets it apart from others is its execution speed and depending on the model the algorithm works on it dominates structured and tabular datasets having classification and regression predictive behaviours. Although XGBoosting is very much the same as gradient boosting it improves over it through systems optimization and algorithmic enhancements.

The system optimization includes:

- Parallelization
- Tree Pruning
- Hardware Optimization

The Algorithmic Enhancements includes:

- Regularization
- Sparsity Awareness
- Weighted Quantile Sketch
- Cross-validation

2.2.7 AdaBoost

Adaboost is a boosting technique of machine learning to exploit the dependencies among the models. This usually works with decision trees. Unlike random forest algorithm, it handles multiple stumps altogether called stump forest. The basic difference between random forest and AdaBoost is that in random forest the depth of a tree cannot be determined, but in the case of Adaboost, the depth is only 1 with 2 leaf nodes and a parent attribute node. Firstly, a data set is considered as the main input to determine a particular result. And the training data set is filled with instances or tuples. Each of the instances has a corresponding weight. Initially, all the instance weights are equally distributed. The weight is given by the following:

$$(1/\textit{number of total instances})$$

Then, the stump with the least amount of Gini coefficient will be selected as the first stump for conducting the first phase. After that, the error will be counted for the first stump; which is basically the sum of misclassified instance weights. Then after calculating the total error; calculation of the stump performance takes place using the following formula:

$$\frac{1}{2} \log((1 - \textit{Total error})/(\textit{Total error}))$$

Then the sample weights need to be updated according to the performance value of the previous stump. And only the misclassified instances will go further in the next stumps as train sets. To update the weight for incorrectly classified instances, this formula is used:

$$\textit{New instance weight} = \textit{old instance weight} \cdot e^{(\textit{stump performance})}$$

And for updating the correctly classified instances the formula is:

$$\textit{New instance weight} = \textit{old instance weight} \cdot e^{-(\textit{stump performance})}$$

After getting the new instance weights, the normalized valued weights will be counted. The normalized values will make a new data set which will pass for the

2nd phase. This process will loop around as the number of total instances. The greater number of instances will determine less error. After all the stumps are done working, the most voted classification will be the result.

Figure 2.5 shows how the total error curve can assess the value range of stump performance.

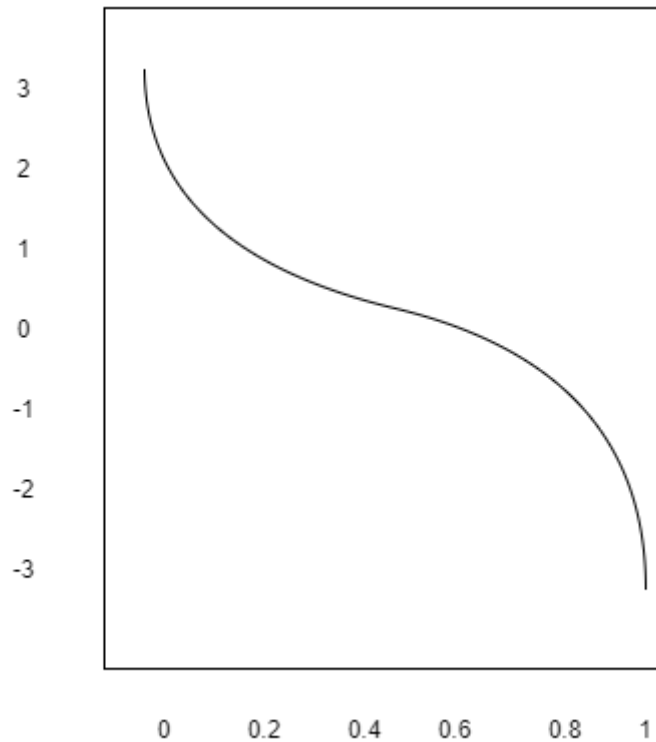


Figure 2.5: Total Error Curve.

2.2.8 Artificial Neural Network

Artificial Neural Network algorithm is a subfield of machine learning highly influenced by human brain structure. The basic mechanism of ANN is, take a set of input data, recognize the pattern inside the data and give a set of outputs. This process is classified into three core neural layers- Input layer, Hidden layer, and Output layer. ANN works as an iterative process. Each of the given inputs are divided into the smallest parts and conveyed through an infinite number of neurons (as nodes). In neural network, the neurons are connected through channels called Synapse which carry weights. Also, a neuron can collect the other neuron's output as its input. Each of the Neurons has two workings- Summation and Function. The summation part does the weighted sum portion. For example- if there are 'N' number of input signals $A_1, A_2, A_3, \dots, A_N$; and they are associated with their channel weights $W_1, W_2, W_3, \dots, W_N$. Then the summation will be $S(N) = A_1.W_1 + A_2.W_2 + A_3.W_3 + \dots + A_N.W_N$. After this summation is done or the input signals are collected in each node, they are forwarded to the Activation

function part. The activation function generates a certain output of a given neuron. The activation function works on Bias factor and Threshold. There are three types of activation function; Linear, Heaviside step function, and Sigmoid Function.

For linear function, Bias factor is simply added with the weighted sum, and in the case of Heaviside Step function; it generates two outputs- 1 and 0. If the weighted sum is greater or Equals to the threshold, then it will give 1 and otherwise 0. Whereas, for sigmoid function, the range is 0 to 1. However, in neural network the output accuracy depends on the hidden layers. More hidden layers mean more accuracy to the output. When the activation function passes its result to another layer the process is called forward propagation. In the output layer, the node which has the highest value defines the output. Mostly, in the output layer, the Heaviside function is used to determine result and in other 2 layers, sigmoid and linear function is used. The determined output is later compared with the actual output to define the error. In the error determination process the values define how wrong the prediction was and with that consideration, backpropagation is used to readjust the synapse weight. This process goes iteratively until the correct output is predicted. Figure 2.6 shows the 3 core layers of Artificial Neural Network and how the neurons are connected with each other.

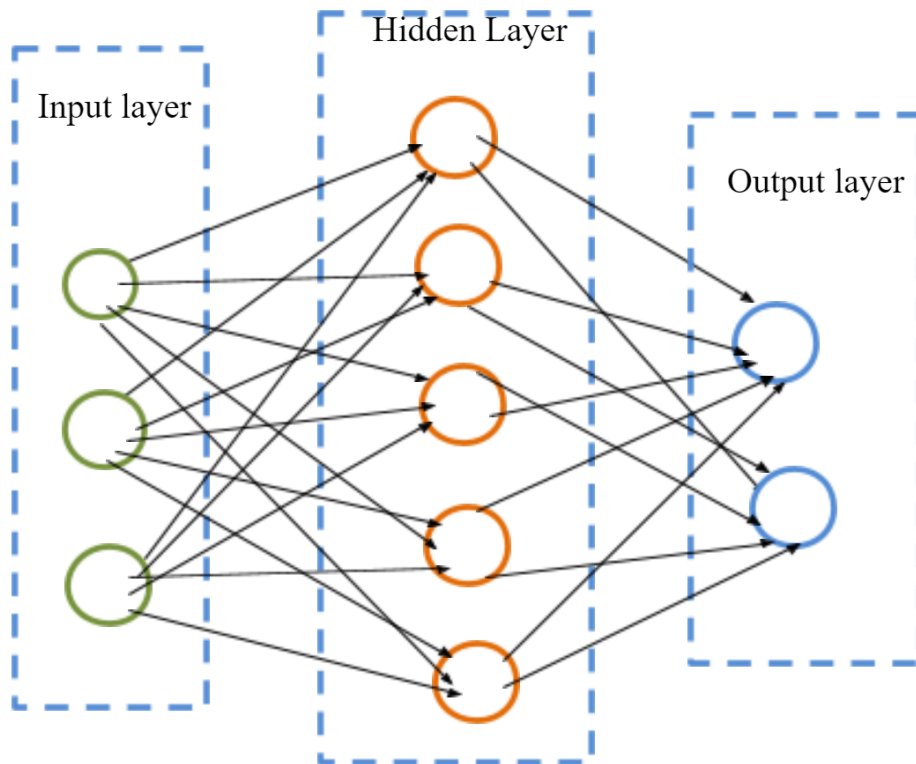


Figure 2.6: Neural Network Layers.

Chapter 3

Proposed Model

3.1 Workflow Overview

Before any ambitious undertaking, such as the one done under this thesis, a proper plan or idea of the workflow is imperative because the tiniest of missed details can lead to results that might prove to be unsatisfactory. For a Machine Learning based thesis such as the one being presented, selecting a dataset with the appropriate amount and type of data is crucial and should be the first step, second comes the selection of various ML algorithms that would be in charge of predicting the target variable, followed by appropriately pre-processing the dataset, splitting the dataset into training and testing sets (in this case the train-test split ratio was 80:20), training the algorithms and comparing the predicted values with the ones in the test set. After all of that has been completed, to ensure that the algorithms work as efficiently as they possibly can it is necessary to see whether or not further pre-processing makes any positive changes to the predicted values. Lastly, the accuracies of the ML algorithms need to be found out and the credit risk of a newly added data should be displayed in a user-interactive way to serve as a practical example which will further strengthen the thesis.

All of the aforementioned steps are displayed in the figure 3.1 for an easier visualization.

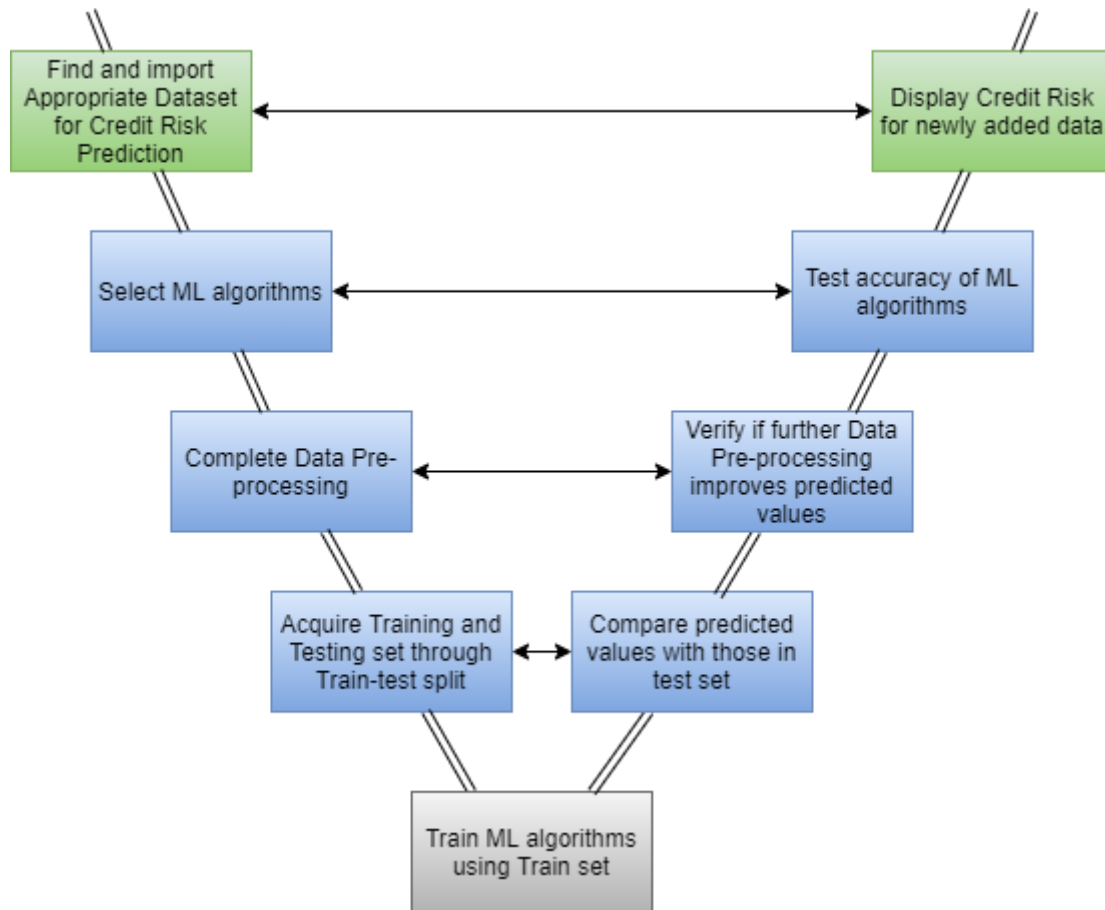


Figure 3.1: Workflow Model.

3.2 Dataset Description

For the proposed model of this thesis, the dataset used was collected from Kaggle, which contains the bank loan status of 82,000 bank account owners [25]. The provided credit_train dataset was used by splitting it into 80% training and 20% testing sets. The dataset in question here has a total of 19 columns or in other words “features” each of which contains certain information about every single account owner. Every account has features/columns called “Customer ID” and “Loan ID” which help specify the account number and the specific loan taken under that account respectively. Then comes the “Loan Status” of an account, which states whether or not a loan taken was paid back in full or was charged off, there are two specific object type of variables under this column, “Fully Paid” specifies that the amount taken was paid back in full and “Charged off” specifies that the creditor has given up on being repaid. Another set of features would be “Current loan amount”, “Term” and “Credit Score” which respectively state the currently remaining amount of a loan, whether the said loan was taken on a long term or short-term basis, and the credit score (higher is better) of an account. Alongside these features, the dataset also contains other self-explanatory features such as “Annual income”, “Years in current job”, “Home Ownership”, “Purpose”, “Monthly Debt”, “Years of Credit History”, “Months since last delinquent”, “Number of Open Accounts”, “Number of Credit Problems”, “Current Credit Balance”, “Maximum Open Credit”, “Bankruptcies” and “Tax Liens”. From the

aforementioned features, “Loan Status” was taken as the target variable where an account owner who was able to fully pay back the loan will be considered suitable for loan and those whose loans were charged off will be considered unsuitable, in other words if the value under “Loan Status” is 1 then the account in question will be eligible to take a loan but if the value is 0 then the account will be denied a loan.

A table describing necessary descriptions of the columns/features present in the dataset is provided below.

Table 3.1: Feature Description.

Feature Name	Description	Variable Type
Loan Status	Current status of loan taken under the account.	object
Current Loan Amount	Amount of money left to be repaid.	float64
Term	Whether the loan taken was for short or long term.	object
Credit Score	Credit score given to an account. Higher is better.	float64
Annual Income	Annual Income of account owner.	float64
Years in current job	Years spent working in current job.	object
Home Ownership	Whether account owner owns a home or not.	object
Purpose	Purpose behind loan taken.	object
Monthly Debt	Amount of debt to be repaid every month.	float64
Years of Credit History	Years since loan was 1st taken.	float64
Months since last delinquent	Months since owner last missed repayments.	float64
Number of Open Accounts	Number of open accounts account owner has.	float64
Number of Credit Problems	The amount of times owner has run into credit problems.	float64
Current Credit Balance	Current credit balance owner has under this account.	float64
Maximum Open Credit	Maximum amount of open credit under this account.	float64

Bankruptcies	Whether account owner has faced bankruptcy.	float64
Tax Liens	Whether tax lien was imposed on account owner.	float64
Loan ID	Specific Loan Identifier taken under said account.	object
Customer ID	Customer's unique ID associated with said account.	object

3.3 Dataset Pre-processing

At first glance over the dataset, it is evident that while the dataset has enough information for credit risk prediction, the data inside the dataset however was not necessarily ready to train a machine learning algorithm and needed some refining. Therefore, basic data pre-processing, as described below, was necessary to make the dataset ready for training.

- Feature Encoding:** Some of the columns inside the dataset have object-type variables. These are basically string variables that certain machine learning algorithms cannot be trained with. To train the ML algorithms numerical values were required and as such, the values of the columns “Loan Status”, “Term”, “Years in current job”, “Home Ownership” and “Purpose” had to be encoded. Now, since all of these columns had unique values, specific integer numbers were mapped to every unique string value. For example, in the “Loan Status” column, “Fully Paid” was mapped to “1” and “Charged Off” to “0”, in the “Term” column, “Short Term” was mapped to “1” and “Long Term” to “0”, in the “Home Ownership” column, values of “Home Mortgage”, “Own Home”, “Rent” and “HaveMortgage” were mapped respectively to “0”, “1”, “2” and “3” and lastly in the “Purpose” column the values of “Home Improvements”, “Debt Consolidation”, “Buy House”, “Other”, “Business Loan”, “Buy a Car”, “major_purchase”, “Take a Trip”, “small_business”, “wedding”, “vacation”, “Educational Expenses”, “moving” and “renewable_energy” were mapped to “0”, “1”, “2”, “3”, “4”, “5”, “6”, “7”, “8”, “9”, “10”, “11”, “12” and “13” respectively.
- Null value Imputing:** After finding out the total number of null values present in each column, it was found that every single column had at least 514 null values or more. ML algorithms cannot be trained with null values, hence the null values present in almost all of the columns were imputed, except for a select few (the reason behind that will be explained in the feature selection part), with the mean value of all the data of the respective columns.
- Feature Scaling:** After encoding and imputing the data it was found that some columns had very large numerical values compared to the other ones.

Therefore all the values were scaled to be within the range of 0 to 1 so that no single column got more priority than the others when training the algorithms. Additionally, feature importance of some algorithms like Random Forest, AdaBoost, XGBoost and Gradient Boosting were generated. The feature importance charts of these algorithms have been provided as follows:

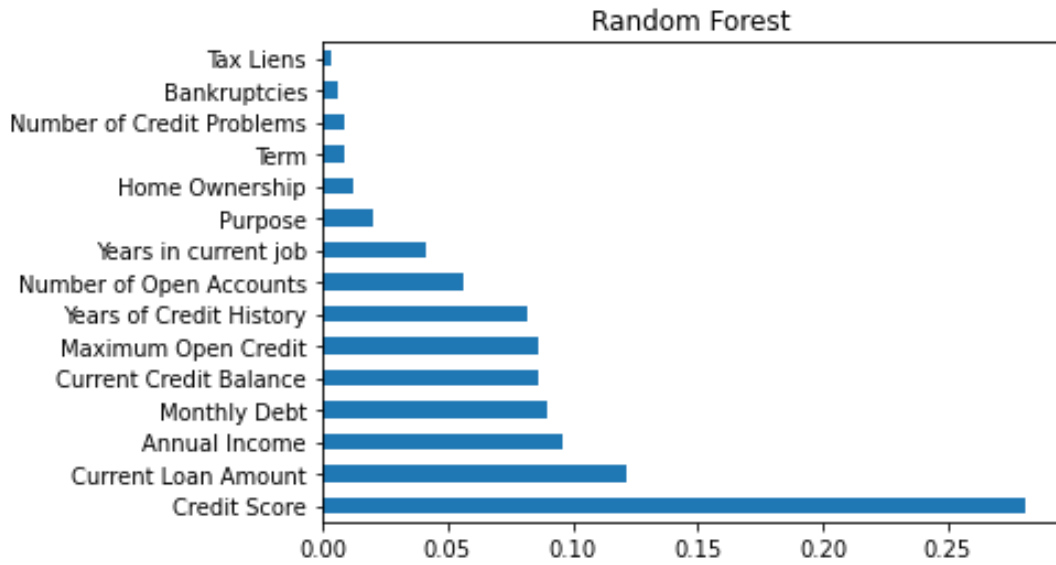


Figure 3.2: Feature Importance for Random Forest.

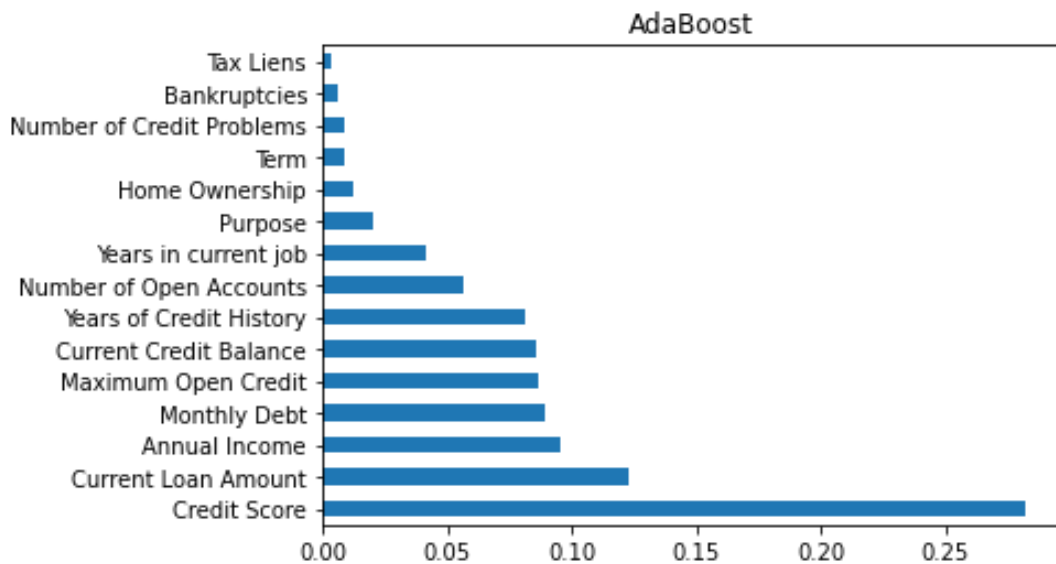


Figure 3.3: Feature Importance for AdaBoost.

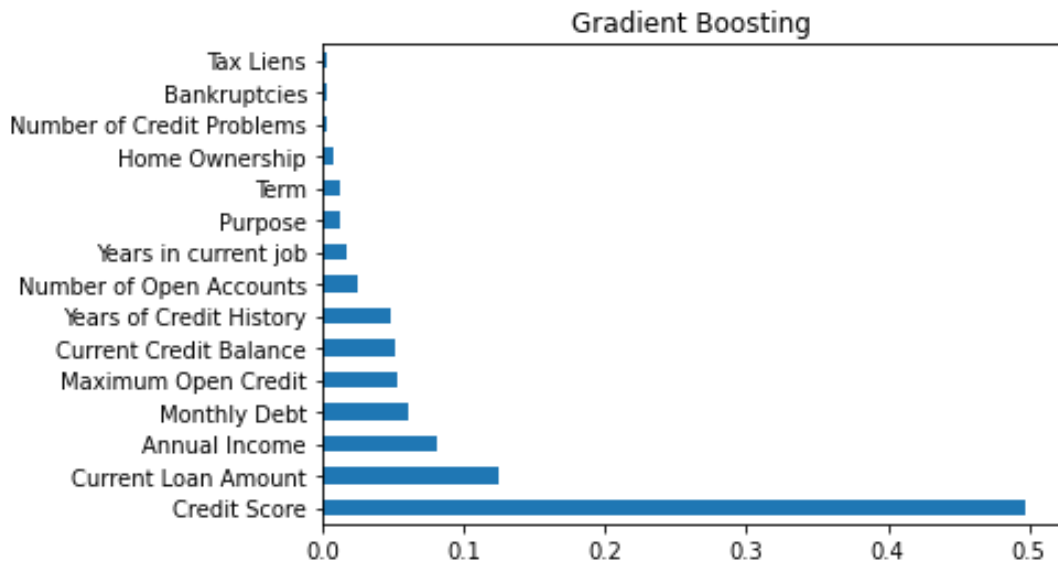


Figure 3.4: Feature Importance for Gradient Boosting.

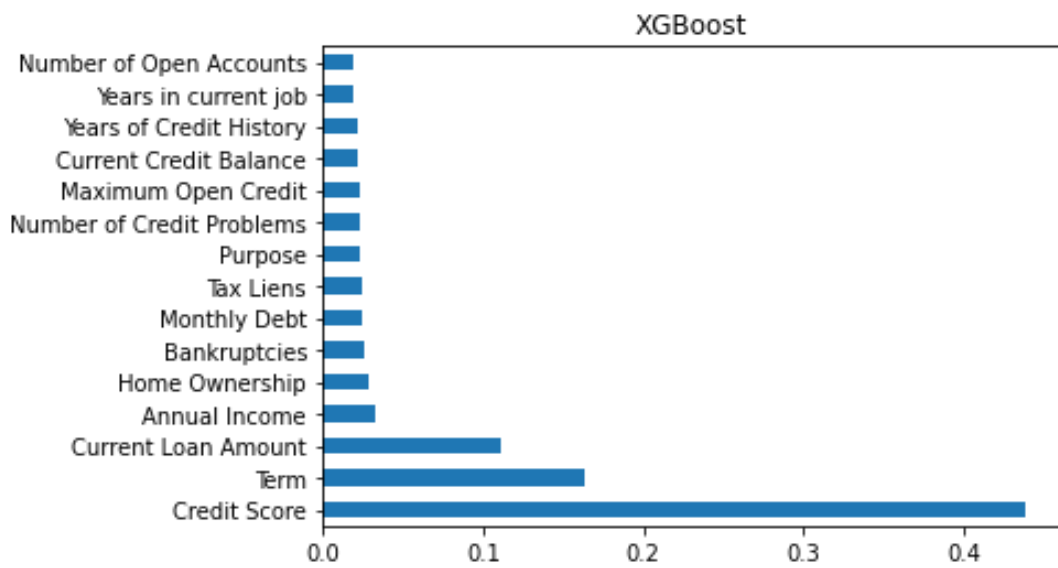


Figure 3.5: Feature Importance for XGBoost.

According to these algorithms, 40% - 60% of the predictions of the algorithms here depend on the features: Credit Score, and Annual Income. As a result, these features have a significant impact on the prediction result. Adding to that, some null inputs were found on these two features that significantly hurt the algorithm's performance, even if those values are imputed with mean, median, etc. These values were found to be decreasing performance across the board by almost 3% to 4% depending on different imputation methods. That is why those columns from the chosen dataset were removed to improve the prediction's solidity.

3.4 Feature Selection

The dataset contained columns or features that were thought to have either not contributed much when training a Machine Learning algorithm or had other issues. For example, there were columns in the dataset such as "Customer ID" and "Loan ID" that were believed to not hold much contribution in terms of training the selected algorithms. Other than that, referring to the correlation matrix and heatmap, the decision to drop the column "Months since last delinquent" was made, as it had relatively low correlation to the target column "Loan Status" and "Months since last delinquent" had more than 50% null values. Figures 3.6 and 3.7 show the heatmap and the matrix of feature correlation respectively.



Figure 3.6: Visual representation of feature correlation through a Heatmap.

	Loan Status	Current Loan Amount	Term	Credit Score	Annual Income	Years in current job	Home Ownership	Purpose	Monthly Debt	Years of Credit History	Months since last delinquent	Number of Open Accounts	Number of Credit Problems	Current Credit Balance	Maximum Open Credit	Bankruptcies	Tax Liens
Loan Status	1.000000	0.194633	0.110637	-0.467328	0.053135	0.006782	-0.051524	-0.008721	-0.007916	0.023705	0.013655	-0.011949	-0.002358	0.009663	0.008404	0.006584	-0.010217
Current Loan Amount	0.194633	1.000000	0.059006	-0.096652	0.013112	-0.001925	-0.010787	-0.005055	-0.006643	0.019282	0.011248	0.001478	-0.002795	0.003880	-0.001271	-0.000608	-0.002048
Term	0.110637	0.059006	1.000000	-0.034624	-0.086169	-0.068793	0.119785	0.057892	-0.158641	-0.041505	0.014905	-0.082605	0.026160	-0.104719	-0.008348	0.028958	0.003430
Credit Score	-0.467328	-0.096652	-0.034624	1.000000	-0.017078	-0.005214	0.020291	-0.003289	-0.001674	-0.009720	-0.003739	0.006435	-0.003022	-0.000104	-0.002827	-0.006835	0.005146
Annual Income	0.053135	0.013112	-0.086169	-0.017078	1.000000	0.078364	-0.172684	-0.021587	0.485230	0.161669	-0.077577	0.146175	-0.017006	0.312340	0.053064	-0.047672	0.040167
Years in current job	0.006782	-0.001925	-0.068793	-0.005214	0.078364	1.000000	-0.208745	-0.065126	0.130790	0.224946	-0.002782	0.053797	0.041553	0.098045	0.001734	0.040045	0.016163
Home Ownership	-0.051524	-0.010787	0.119785	0.020291	-0.172684	-0.208745	1.000000	0.135140	-0.205397	-0.197200	0.057198	-0.138149	-0.002839	-0.167332	-0.024351	0.003327	-0.005129
Purpose	-0.008721	-0.005055	0.057892	-0.003289	-0.021587	-0.065126	0.135140	1.000000	-0.082081	-0.034199	-0.006885	-0.084837	-0.014158	-0.059519	0.001464	-0.024782	0.006418
Monthly Debt	-0.007916	-0.006643	-0.158641	-0.001674	0.485230	0.130790	-0.205397	-0.082081	1.000000	0.199289	-0.056818	0.411353	-0.055383	0.481348	0.039268	-0.078979	0.020119
Years of Credit History	0.023705	0.019282	-0.041505	-0.009720	0.161669	0.224946	-0.197200	-0.034199	0.199289	1.000000	-0.044292	0.132349	0.061588	0.208470	0.031124	0.066247	0.017245
Months since last delinquent	0.013655	0.011248	0.014905	-0.003739	-0.077577	-0.002782	0.057198	-0.006885	-0.056818	-0.044292	1.000000	-0.032569	0.104642	-0.028662	-0.008785	0.123951	0.012624
Number of Open Accounts	-0.011949	0.001478	-0.082605	0.006435	0.146175	0.053797	-0.138149	-0.084837	0.411353	0.132349	-0.032569	1.000000	-0.013995	0.228136	0.031341	-0.024575	0.006545
Number of Credit Problems	-0.002358	-0.002795	0.026160	-0.003022	-0.017006	0.041553	-0.002839	-0.014158	-0.055383	0.061588	0.104642	-0.013995	1.000000	-0.112516	-0.012072	0.752942	0.581290
Current Credit Balance	0.009663	0.003880	-0.104719	-0.000104	0.312340	0.098045	-0.167332	-0.059519	0.481348	0.208470	-0.028662	0.228136	-0.112516	1.000000	0.139204	-0.122603	-0.015645
Maximum Open Credit	0.008404	-0.001271	-0.008348	-0.002827	0.053064	0.001734	-0.024351	0.001464	0.039268	0.031124	-0.008785	0.031341	-0.012072	0.139204	1.000000	-0.014574	-0.001029
Bankruptcies	0.006584	-0.000608	0.028958	-0.006835	-0.047672	0.040045	0.003327	-0.024782	-0.078979	0.066247	0.123951	-0.024575	0.752942	-0.122603	-0.014574	1.000000	0.046110

Figure 3.7: Feature Correlation Matrix..

3.5 Dataset Limitations

After going through a good number of datasets, it was unanimously decided to use the one cited in this thesis for performing the necessary research due to it having more practical attributes and a large quantity of user data. However, despite making many efforts to find the most suitable dataset for the stated purpose, the dataset still feels a bit lacking in some aspects which may have not been in anybody’s control.

Firstly, the dataset was not up-to-date with the time of when the thesis was being conducted. That was mainly due to the lack of availability of properly tallied datasets for the topic of this thesis from online sources. At the initial stages of the thesis, the plan was to gather a dataset physically from the established economic institutions in the locality. Alas, that did not come to fruition due to quarantining because of the pandemic. Thus, the resources available online had become the only viable options for gathering data. When scouring through the online dataset repositories, only a handful of relevant datasets were found. Among them there were some newer but very short datasets and some more detailed but ancient datasets, leading to choosing this very dataset in order to strike a proper balance.

Moreover, since the dataset chosen was collected from an online source, it was not plausible to determine or guarantee the authenticity of the data provided by the users in the dataset. The online source chosen is a reputable one, but it is not even completely feasible on their part to ensure that each and every portion of user-provided data is accurate. In this scenario, it could only be assumed that people have provided their credentials in good faith, believing that even if there are some arbitrary portions of data provided, the quantity would be negligible and would not offset the desired outcome of the thesis.

Chapter 4

Result Analysis & Comparison

4.1 In-depth review of results achieved

After careful feature selection and appropriate dataset preprocessing, multiple Machine Learning algorithms were tested on the preprocessed dataset. The main objective of which was to discover the algorithm that gives the most precise results at detecting credit risk accurately in comparison to its competitors. And the way this was achieved was by finding out the Accuracy, Precision, Recall and F1 Scores of the various algorithms using reliable built-in functions. Now, since the algorithms are used to predict whether or not an individual is under credit risk, the problem at hand can be classified as a binary classification problem. Below is the layout of a confusion matrix related to the predicted binary values achieved through the dataset where the parameters are as follows.

TP = True Positive.

TN = True Negative.

FP = False Positive.

FN = False Negative.

		Actual Values	
		Credit Risk	No Credit Risk
Predicted Values	Credit Risk	<i>TP</i>	<i>FP</i>
	No Credit Risk	<i>FN</i>	<i>TN</i>

Figure 4.1: Layout of confusion matrix.

The true positive (TP) and true negatives (TN) indicate the correct prediction made by the model whereas the false positive (FP) and false negatives (FN) are indicators of the wrong predictions made by the model.

Now, the accuracy is calculated by finding out the ratio between the number of correctly classified predictions and the total number of predictions. Provided below is an equation to help compute the accuracy of the model.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision is used to figure out how many of the positive predictions are actually correct and is calculated by computing the ratio of the correctly predicted positive outcomes and the total number of positive predictions made. Provided below is an equation to help compute the precision of the model.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall is a metric which is used to determine what proportion of the actual positives was classified correctly. Provided below is an equation to help compute the recall of the model.

$$\text{Recall} = \frac{TP}{TP + FN}$$

The last metric being used is the F1 score. It is calculated using the precision and recall values and is considered to be a better form of evaluation. Provided below is an equation to help compute the F1 score of the model.

$$\text{F 1 Score} = \frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

With all of the metrics for evaluation described above, the maximum accuracy achieved was 84.6% by using Gradient Boost. For better understanding and a proper visual representation of the results achieved, the ROC curves of all of the models have been included below.

Figure 4.2 represents the ROC curve for the selected dataset when using Logistic Regression.

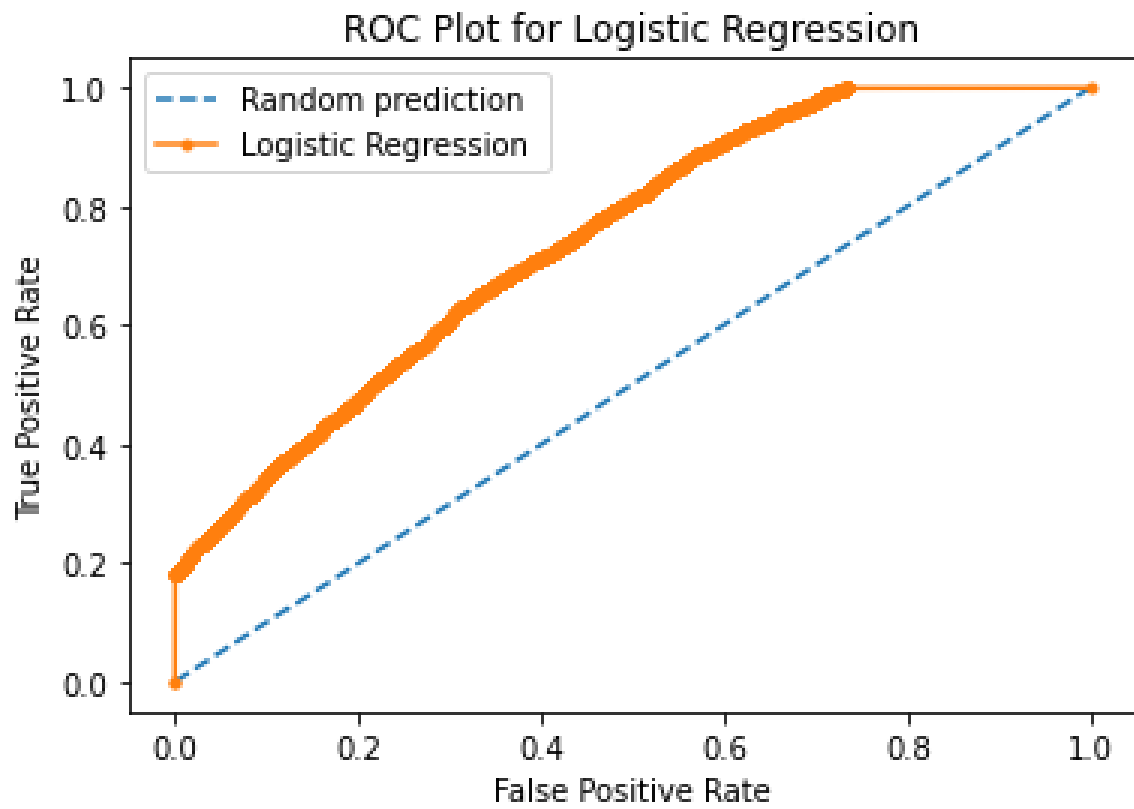


Figure 4.2: ROC for Logistic Regression.

Figure 4.3 represents the ROC curve for the selected dataset when using Random Forest.

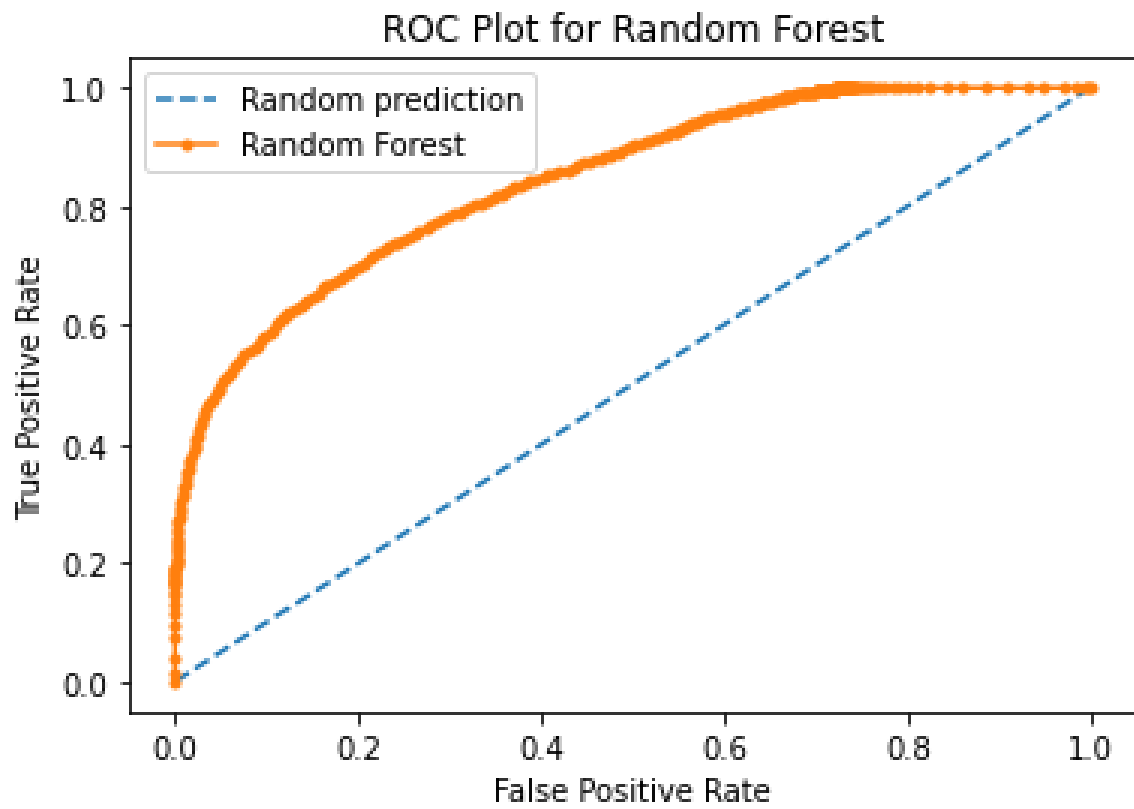


Figure 4.3: ROC for Random Forest.

Figure 4.4 represents the ROC curve for the selected dataset when using KNN.

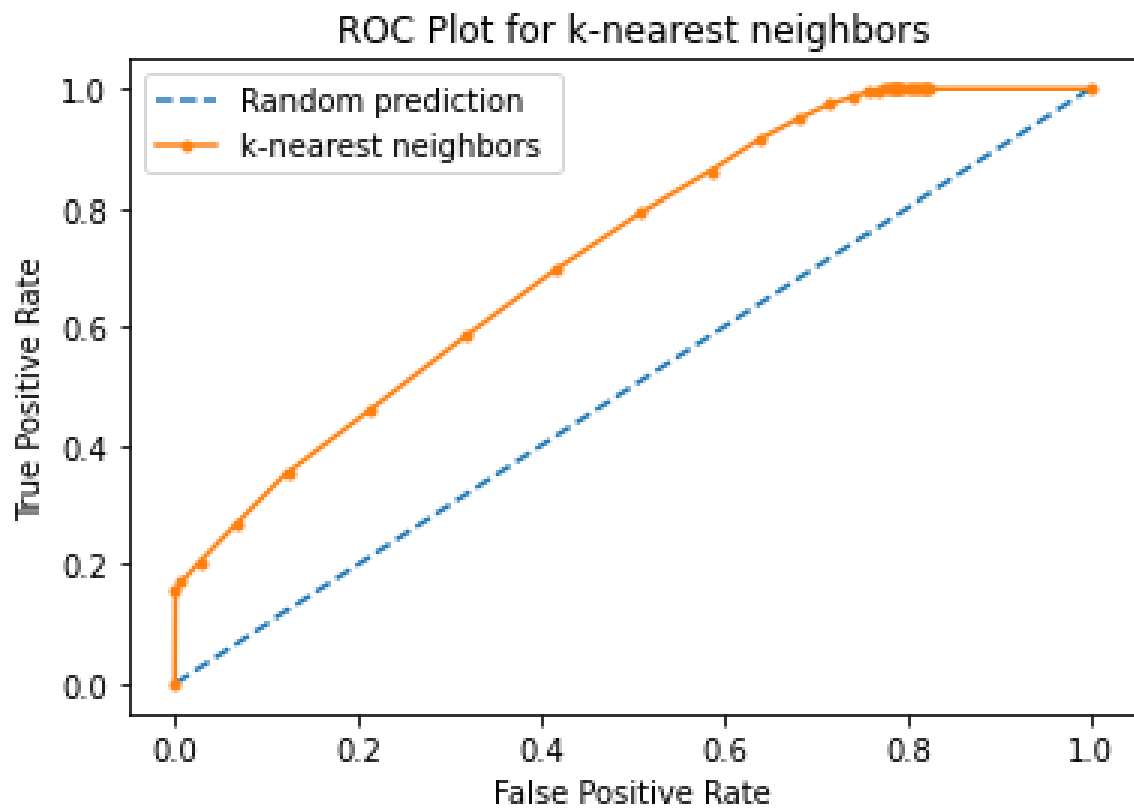


Figure 4.4: ROC for K-Nearest Neighbor.

Figure 4.5 represents the ROC curve for the selected dataset when using ANN

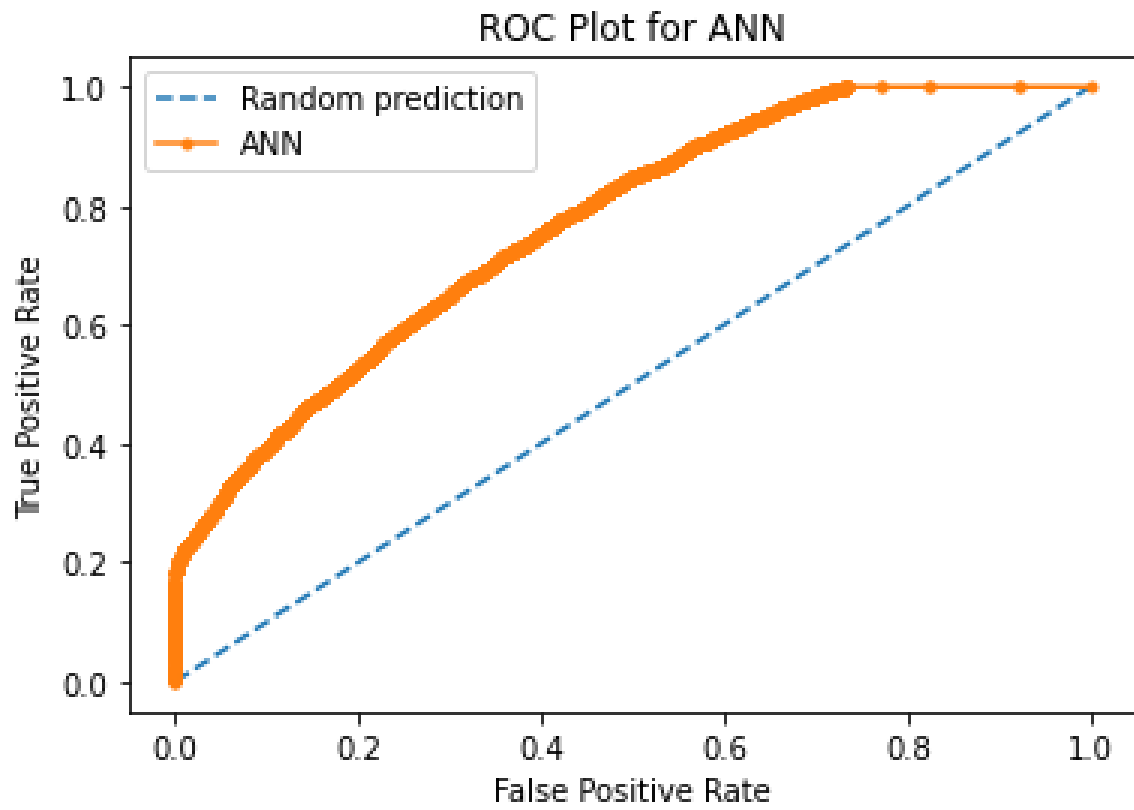


Figure 4.5: ROC for Artificial Neural Network.

Figure 4.6 represents the ROC curve for the selected dataset when using AdaBoost.

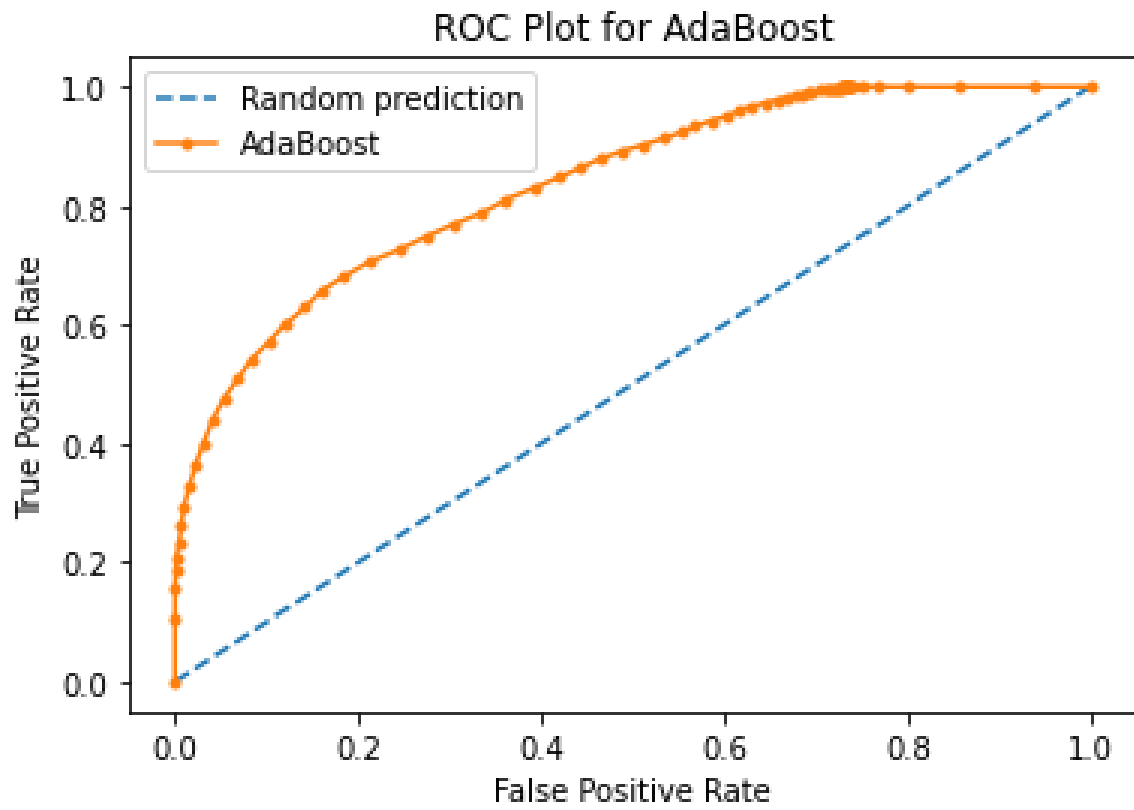


Figure 4.6: ROC for AdaBoost.

Figure 4.7 represents the ROC curve for the selected dataset when using Gradient Boosting.

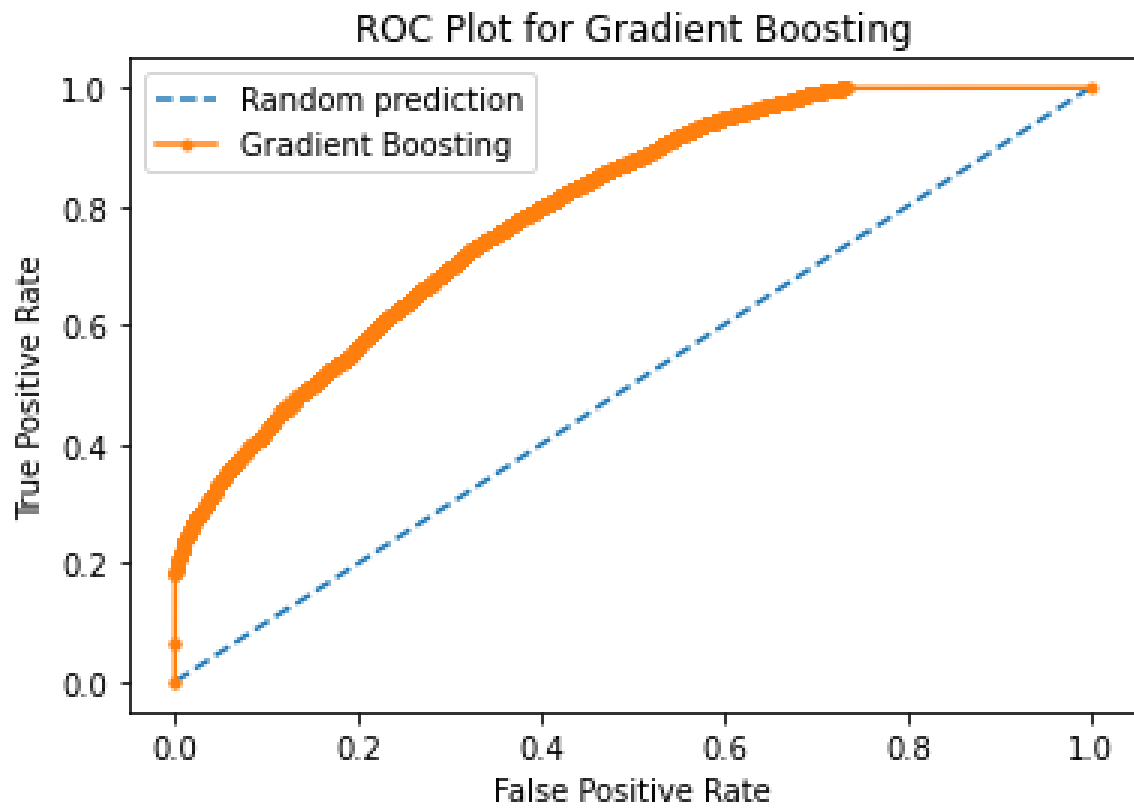


Figure 4.7: ROC for Gradient Boosting.

Figure 4.8 represents the ROC curve for the selected dataset when using XGBoost.

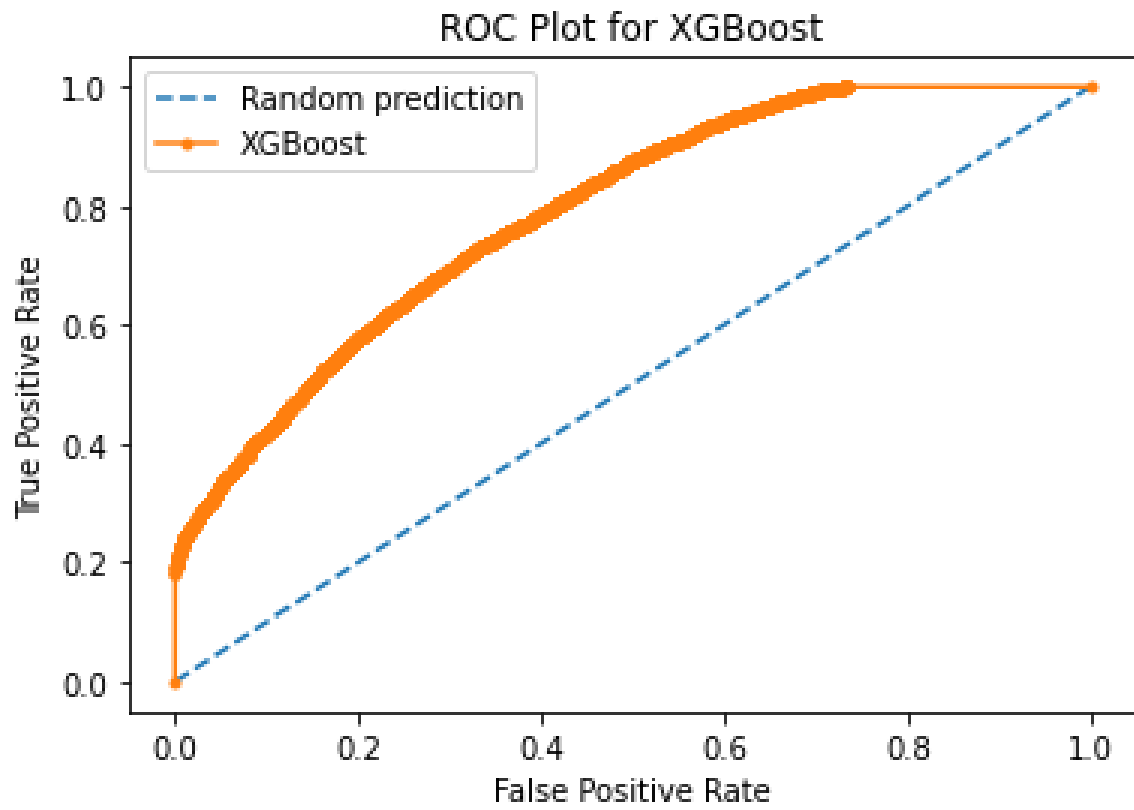


Figure 4.8: ROC for XGBoost.

To show the comparison between the results achieved through the different algorithms that were used, a table below has been attached which contains the accuracy, precision, recall, and F1-score of said algorithms.

Table 4.1: Performance evaluation of algorithms used.

Algorithm	Accuracy	Precision	Recall	F1 Score
Linear regression	84.42%	84%	100%	91%
Random Forest	84.55%	84%	99%	91%
Logistic Regression	84.42%	84%	100%	91%
K-Nearest Neighbor	83.54%	83%	100%	91%
Gradient Boosting	84.6%	84%	99%	91%
XGBoost Classifier	84.45%	84%	100%	91%
Artificial Neural Network	84%	84%	100%	91%
AdaBoost	84.58%	84%	100%	91%

4.2 Comparison

Some notebooks on Kaggle were found that also used the same dataset. One of the notebooks got Random Forest as their best performing algorithm and achieved an accuracy of 74.67% which, compared to the best performing algorithm in this thesis is approximately 9.93% less accurate [26]. Another paper got the highest 77.17% accuracy on the XGBoost algorithm, which is outperformed by the obtained result as well [27].

The accuracy obtained was found to be higher for a few reasons. It is seen that the “Months since last delinquent” feature of the dataset contains almost 50% Null values. That harms the prediction and drags the algorithm’s performance down. In both of the notebooks, the comparison had been made by imputing those 50% null values with mean values and it was found that it leads to a major disadvantage for their accuracy. One of the notebooks divided most of the numerical features with the categorical encoding that also drags down the authenticity of the dataset. In addition to that, it was found min-max scaling to be helpful for the algorithm for learning better. As a result, a better accuracy was obtained after scaling the dataset compared to both of the notebooks, since they did not properly scale the dataset.

A comparison between the findings of the thesis and the mentioned notebooks has been shown in a tabular format below:

Table 4.2: Accuracy comparison with relevant reference works.

Algorithm	Thesis Findings	Ref work 1	Ref work 2
K-Nearest Neighbor	83.53%	74.95%	70.2%
ANN	84.4%	-	-
Linear Regression	84.42%	-	-
Logistic Regression	84.42%	77.17%	74.65%
XGBoost	84.48%	77.17%	74.30%
Random Forest	84.55%	-	74.67%
AdaBoost	84.58%	-	-
Gradient Boosting	84.60%	-	-

Chapter 5

Implementation

The purpose behind this thesis is to present the topic at hand (the topic being the ability to predict credit risk through the use of ML algorithms) to the readers of this paper, discuss the importance of the said topic and give an in-depth review of the various results achieved. And while this thesis is theoretical in nature, the topic itself dwells within the realms of practicality and wishes to tackle a problem prevalent in the practical world. Therefore, an endeavor was taken to provide a glimpse of the usefulness of the topic discussed in this thesis through a possible practical usage.

A web application was developed that would allow people to at the very least get an approximate idea of whether or not they are under credit risk or to put it in more user-friendly words, to see whether or not they are eligible to get a loan. In light of the current situation of the world, it is hoped by all to be able to survive the pandemic known as Covid-19, and the best way to do that unfortunately is for people to isolate themselves in the confines of their homes as much as possible and minimize the risk of exposure by restricting ourselves to the best of their abilities from going to public and crowded places. Now, since banks are an important part of everybody's lives, they usually tend to be crowded especially during trying times like these when businesses are going bankrupt and people are losing jobs. So, it seems rather convenient to have websites and web applications such as this prototype which would allow the general populace to determine their chances of getting a loan instead of running around and asking about it from bank to bank. Below are pictures of the prototype web application which uses ML algorithms to predict a user's chances to get a loan based on their credit risk in exchange for some information (which would obviously be completely secure) and even offering to advise on what field(s) said user needs to make improvements on provided that they are not eligible for a loan but wish to get one in the future.

The flowchart of the implemented web application has been illustrated below in figure 5.1:

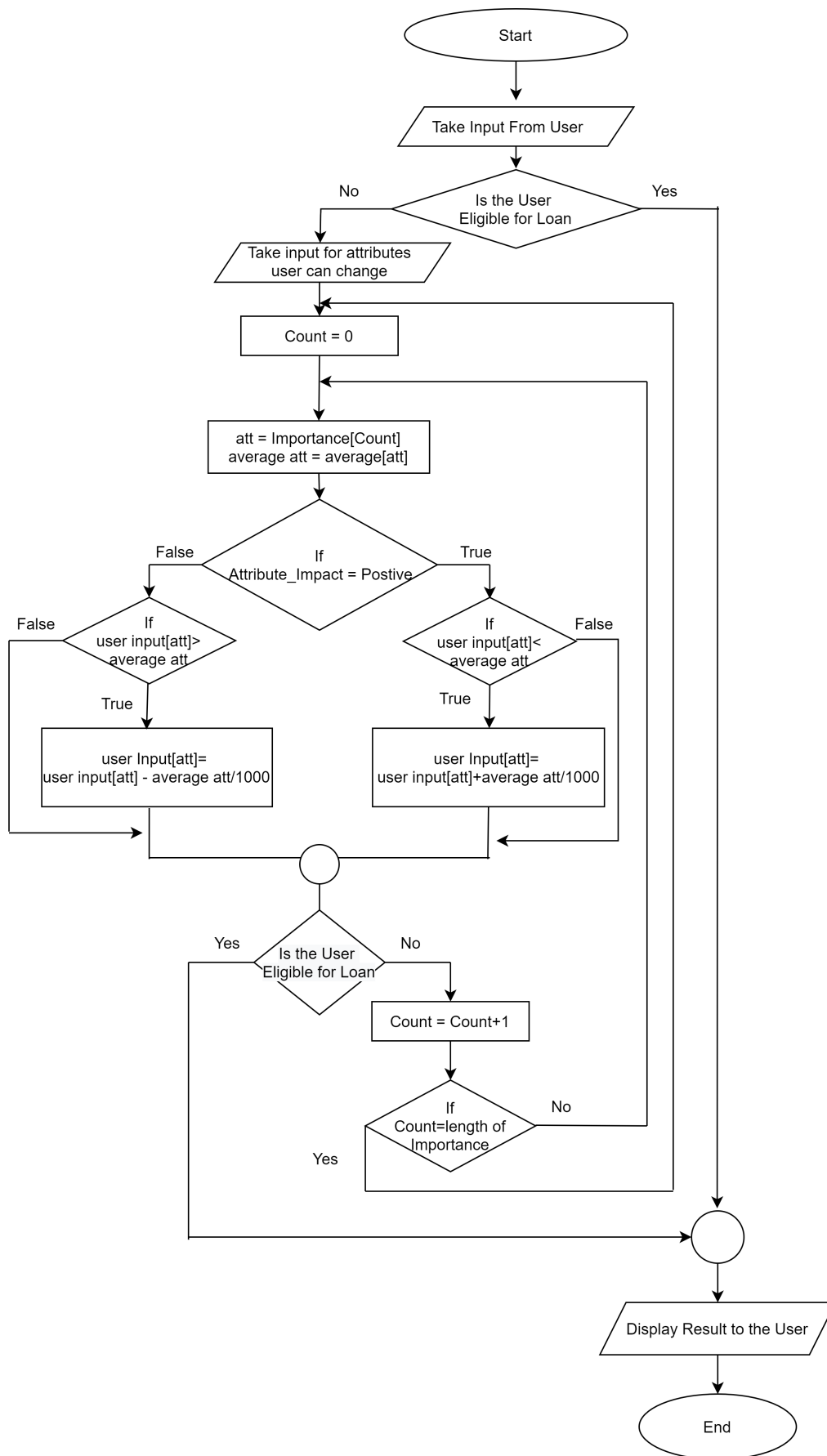


Figure 5.1: Flowchart showing the workings of the web application.

Below are the snapshots of the graphical user interface of the web application. The first one would be the input page where the users are to enter their credentials to check for their eligibility.

Loan Recommender Home Sign Up

Please Enter Your Credentials

Credit Score

Your Annual Income \$

Current Loan Amount \$

Years in current job Years

How Many Home You Own

Purpose of Loan

Loan Term

Monthly Debt \$

Years of Credit History Years

Number of Open Accounts

Current Credit Balance \$

Maximum Open Credit \$

[Confirm identity](#)

Figure 5.2: Inputs to be provided.

Scenario 1 shows the outcome where the user is not in a Credit risk.



Figure 5.3: Scenario 1: The user is eligible for getting a loan.

Scenario 2 shows the outcome where the user is in a Credit risk. Here the user is given the option to choose the aspects that may be changeable on the user's part in order to gain eligibility.

Loan Recommender Home Sign Up

Sorry You are Not Eligible for Getting a Loan

To get recommendation Click the button below

Proceed

Please Select The Attributes You are Flexible with Changing

<input checked="" type="checkbox"/> Credit Score	<input checked="" type="checkbox"/> Your Annual Income	<input checked="" type="checkbox"/> Home Ownership	<input checked="" type="checkbox"/> Monthly Debt
<input type="checkbox"/> Current Loan Amount	<input type="checkbox"/> Number of Open Accounts	<input type="checkbox"/> Current Credit Balance	<input type="checkbox"/> Maximum Open Credit

Confirm

Figure 5.4: Scenario 2: The user is not eligible.

After choosing the attributes, the user is provided with the necessary changes needed to be made in order to be eligible.

Loan Recommender Home Sign Up		
Your Result is Ready. Here is What We Got For You		
Attributes	Before	After
Current Loan Amount	486046.0	486046.0
Term	0.0	0.0
Credit Score	695.0	730
Annual Income	1440827.0	1440827.0
Years in current job	8.0	8.0
Home Ownership	0.0	0
Monthly Debt	33979.22	29636.2
Number of Open Accounts	18.0	18.0
Current Credit Balance	633631.0	633631.0
Maximum Open Credit	937024.0	937024.0
Suggestive Improvement		
Credit Score	Should be improved by 35.0	
Annual Income	No Improvement Required	
Home Ownership	No Improvement Required	
Monthly Debt	Should be decreased by 4343.02	

Figure 5.5: Scenario 2: Improvements suggested in order to be eligible.

Chapter 6

Conclusion & Future Works

In this thesis, the goal was to predict whether an organization or individual is a competent creditor or not through analyzing datasets correlated to credit risk and training appropriate Machine Learning algorithms with said datasets and selecting the best one. Datasets containing the required information have been collected through an already available and reliable online source. In recent times, various Machine Learning algorithms have shown competence when it comes down to credit risk analysis. Analyzing the chosen dataset and testing it through implementing AI-based algorithms, the most efficient algorithm to use has been sorted out, which can provide the highest accuracy . Moreover, using the data and the results of the thesis, an application has been developed to be used by any relevant institution or individual in order to check their eligibility of accepting a loan and the factors influencing their shortcomings in the cases their loan requests are not being accepted. Through these perilous times analyzing credit risk with high enough accuracy and speed has become of the utmost importance and in such circumstances Machine Learning can prove to be a notable solution to said issue.

As mentioned before, the pitfall of the world crisis during the development of this thesis had hampered the process of providing the best outcome possible, mainly in conducting work on the most up-to date user data. Therefore, it can certainly be said that the obtained outcome could be improved provided a newer and more robust dataset could be acquired. In the future, a dataset could be collected from a globally established banking firm to ensure that the data obtained is the latest and does not contain any irrelevant or false user information.

Additionally, when developing the application for testing the practical implementation of the thesis, the main focus of the application was on its core functionality. Needless to say, a more user approachable and detailed application ought to be developed in the future which shall focus on user privacy and security of user provided data.

It is being ardently hoped that the thesis and its obtained results can contribute to saving the world economy which is spiraling downwards at an appalling rate and close to collapsing due to the ongoing pandemic.

Bibliography

- [1] M. A. Yusuf and J. Rahman, *Covid-19 impact and the world economy in eight charts by rapid*, The Financials Express, Ed., <https://thefinancialexpress.com.bd/economy/covid-19-impact-and-the-world-economy-in-eight-charts-by-rapid-1591098739>, 2020.
- [2] Forbes Magazine, *Credit scoring made simple*, Forbes, Ed., <https://www.forbes.com/credit-center/credit-scoringmade-simple.html>.
- [3] T. Yiu, *Understanding random forest - towards data science*, <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>, 2019.
- [4] L. Weston, *Why your credit score is important*, <https://tiny.cc/nerdwallet>, 2019.
- [5] L. Mester, “What is the point of credit scoring?” *Business Review*, vol. Sep/Oct, pp. 3–16, Feb. 1997.
- [6] Press Trust of India, *9 million loan defaulters blacklisted in china; \$27 billion frozen*, Hindustan Times, Ed., <https://www.hindustantimes.com/world-news/9-million-loan-defaulters-blacklisted-in-china-27-billion-frozen/story-Z9KBvxffasUwsD8LRLjpPJ.html>, 2018.
- [7] BBC News, *Nirav modi fraud case and india’s bank debt crisis*, BBC, Ed., <https://www.bbc.com/news/world-asia-india-43123337>, 2018.
- [8] A. Nova, *More than 1 million people default on their student loans each year*, CNBC, Ed., <https://www.cnbc.com/2018/08/13/twenty-two-percent-of-student-loan-borrowers-fall-into-default.html>, 2018.
- [9] G. Mowla, *Default loans plague banking sector*, Dhakatribune, Ed., <https://www.cnbc.com/2018/08/13/twenty-two-percent-of-student-loan-borrowers-fall-into-default.html>, 2018.
- [10] S. Islam, *Bad loans cripple the banking sector*, Dhakatribune, Ed., <https://www.dhakatribune.com/business/banks/2017/10/06/defaulted-loans-amount-12-gdp>, 2017.
- [11] S. Oreski, D. Oreski, and G. Oreski, “Hybrid system with genetic algorithm and artificial neural networks and its application to retail credit risk assessment,” *Expert Systems with Applications*, vol. 39, no. 16, pp. 12 605–12 617, 2012. DOI: 10.1016/j.eswa.2012.05.023. [Online]. Available: <https://doi.org/10.1016/j.eswa.2012.05.023>.
- [12] L. Yu, S. Wang, and K. K. Lai, “Credit risk assessment with a multistage neural network ensemble learning approach,” *Expert Systems with Applications*, vol. 34, no. 2, pp. 1434–1444, 2008. DOI: 10.1016/j.eswa.2007.01.009. [Online]. Available: <https://doi.org/10.1016/j.eswa.2007.01.009>.

- [13] A. Khashman, “Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes,” *Expert Systems with Applications*, vol. 37, no. 9, pp. 6233–6239, 2010. DOI: 10.1016/j.eswa.2010.02.101. [Online]. Available: <https://doi.org/10.1016/j.eswa.2010.02.101>.
- [14] J. Galindo and P. Tamayo, “Credit risk assessment using statistical and machine learning: Basic methodology and risk modeling applications,” *Computational Economics*, vol. 15, pp. 107–143, 2000. DOI: 10.1023/A:1008699112516. [Online]. Available: <https://doi.org/10.1023/A:1008699112516>.
- [15] X. Ma and S. Lv, “Financial credit risk prediction in internet finance driven by machine learning,” *Neural Computing and Applications*, vol. 31, no. 12, pp. 8359–8367, Dec. 2019. DOI: 10.1007/s00521-018-3963-6. [Online]. Available: <https://doi.org/10.1007/s00521-018-3963-6>.
- [16] T. Pandey, A. Jagadev, S. Mohapatra, and S. Dehuri, “Credit risk analysis using machine learning classifiers,” Aug. 2017, pp. 1850–1854. DOI: 10.1109/ICECDS.2017.8389769.
- [17] P. Addo, D. Guegan, and B. Hassani, “Credit risk analysis using machine and deep learning models,” *Risks*, vol. 6, p. 38, Apr. 2018. DOI: 10.3390/risks6020038.
- [18] P. Danėnas, G. Garšva, and S. Gudas, “Credit risk evaluation model development using support vector based classifiers,” *Procedia Computer Science*, vol. 4, pp. 1699–1707, Jun. 2011. DOI: 10.1016/j.procs.2011.04.184.
- [19] B. Baesens, T. Van Gestel, S. Viaene, M. STEPANOVA, J. Suykens, and J. Vanthienen, “Benchmarking state-of-the-art classification algorithms for credit scoring,” *The Journal of the Operational Research Society*, vol. 54, no. 6, pp. 627–635, Jun. 2003. DOI: 10.1057/palgrave.jors.2601545.
- [20] S. Oreski and G. Oreški, “Genetic algorithm-based heuristic for feature selection in credit risk assessment,” *Expert Systems with Applications: An International Journal*, vol. 41, pp. 2052–2064, Mar. 2014. DOI: 10.1016/j.eswa.2013.09.004.
- [21] A. Ahmed, F. Abida, S. Islam, and R. Aziz, “Building a credit scoring model to assign a reference score based on credit transaction and relevant profile data,” Ph.D. dissertation, Sep. 2019.
- [22] A. Lawi, F. Aziz, and S. Syarif, “Ensemble gradientboost for increasing classification accuracy of credit scoring,” Aug. 2017, pp. 1–4. DOI: 10.1109/CAIPT.2017.8320700.
- [23] J. Nalic and A. Svraka, “Using data mining approaches to build credit scoring model: Case study — implementation of credit scoring model in microfinance institution,” Mar. 2018, pp. 1–5. DOI: 10.1109/INFOTEH.2018.8345543.
- [24] J. Nalic and A. Svraka, “Importance of data pre-processing in credit scoring models based on data mining approaches,” May 2018, pp. 1046–1051. DOI: 10.23919/MIPRO.2018.8400191.
- [25] Z. Begiev, *Bank loan status dataset*, https://www.kaggle.com/zaurbegiev/my-dataset?fbclid=IwAR0siBjqf15APjhHRO7ia8nhGpmnvrHVWHnJwUnKH6JPbEpCJsnV_LhL8nM&select=credit_train.csv, Jun. 2017.

- [26] S. Upadhyaya, *Bank loan status classification*, <https://www.kaggle.com/sazack/loan-status-classification>, May 2019.
- [27] Megr, *Bank loan cleasing and more 5 mach learning*, <https://www.kaggle.com/megr25/bank-loan-cleasing-and-more-5-mach-learning>, Oct. 2020.