BACHELOR OF SCIENCE IN
COMPUTER SCIENCE AND ENGINEERING

Inspiring Excellence

Application of Machine Learning in Credit Risk Assessment: A Prelude to Smart Banking

Authors
Mir Ishrak Maheer Dhruba
Nawab Haider Ghani
Sazzad Hossain
Syed Zamil Hasan Shoumo

Supervisor
Hossain Arif
Assistant Professor
Department of CSE

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

Department of Computer Science and Engineering
BRAC University, Dhaka - 1212, Bangladesh

December 2018

I would like to dedicate this thesis to my loving parents ...

Declaration

It is hereby declared that this thesis /project report or any part of it has not been submitted elsewhere for the award of any Degree or Diploma.

Authors:

Author: Mir Ishrak Maheer Dhruba
Student ID: 15101007

Author: Nawab Haider Ghani
Student ID: 15101064

Author: Sazzad Hossain
Student ID: 14201003

Author: Syed Zamil Hasan Shoumo
Student ID: 15101009

Supervisor:

Hossain Arif
Assistant Professor, Department of Computer Science and Engineering
BRAC University

December 2018

The thesis titled Application of Machine Learning in Credit Risk Assessment:
A Prelude to Smart Banking
Submitted by:
Author Mir Ishrak Maheer Dhruba Student ID: 15101007
Author Nawab Haider Ghani Student ID: 15101064
Author Sazzad Hossain Student ID: 14201003
Author Syed Zamil Hasan Shoumo Student ID: 15101009
of Academic Year Fall 2018 has been found as satisfactory and accepted as partial fulfillment of the requirement for the Degree of B.Sc. Engineering in CSE

1.	Hossain Arif Assistant Professor BRAC University	Chairman
2.	Md. Abdul Mottalib Professor and Chairperson BRAC University	Member
3.	Name of Internal Member Designation Address	Member
4.	Name of Internal Member Designation Address	Member
5.	Name of External Member Designation Address	Member

Acknowledgements

As this meticulous journey comes to end, we have come across many well-wishers whose invaluable contribution and assistance made this project successful. Our beloved Professor, Mr. Hossain Arif is one such person. His continuous support and guidance has kept us motivated and enabled us to present our best in all cases. We would also like to thank Mr. Samiul Islam and Dr. Iftekharul Mobin for their continuous encouragement and expertise in this field. Last but certainly not the least; we would like to show our appreciation towards our families without whose unconditional support none of this would have been possible.

Abstract

A precise credit risk assessment system is vital to a financial institution for its proper and impeccable functioning. Accurate estimations of credit risk will allow them to continue their operation in a gainful and transparent way. As the rate of loan defaults are gradually increasing, bank authorities are finding it more and more difficult to correctly assess loan requests. Thus the subject of credit risk has become a highly conferred and examined topic throughout the world. Numerous solutions have been given, one being more efficient than the other and several studies are still being made for solving this difficult predicament. Thus keeping the implications of such a problematic matter in mind this paper proposes to build a machine learning model which can precisely assess credit risk and predict possible loan defaulters for any credit lending institution. Taking into account a borrower's financial and social history this paper proposes a way to accurately define whether a customer's loan request should be accepted or not which in turn can steadily save the creditor from incurring further loss. Evaluating data from previous successful borrowers and loan defaulters, a comparative analysis have been made using our supervised learning model and the results obtained can be used to predict the behavior of future borrowers. This model can assist a financial institution in assessing whether it should accept a loan request or not. Different combinations of feature selection algorithm and classifiers have been made and based upon metrics such as accuracy, AUC score, F1 score etc. the best model has been selected. Recursive feature elimination with cross validation (RFECV) and Principal Component Analysis (PCA) have been used to find the optimum number of features needed to make an accurate prediction. This allows us to make more efficient and optimal use of the limited available resources. The assessment will be performed in a supervised environment and so Support Vector Machines (SVM), Random Forest, Extreme Gradient Boosting and Logistic Regression have been used as the classifiers. In order to ensure all possible combinations have been properly tested k folds cross validation has been used to bring out a more balanced result. Furthermore, GridSearchCV has been used to tune the selected hyperparameters for each model in order to obtain the best result possible. And based upon this a comparison in a tabular form has been shown which showcases the most and the least accurate model for precisely assessing loan requests.

Keywords: Credit Risk, Loan Assessment, Machine Learning, SVM, Random Forest, Extreme Gradient Boosting, Logistic Regression.

Table of Contents

List of Figures

List of Tables

Nomenclature

Acronyms / Abbreviations

ALNN Adaptive Linear Neural Network

ANN Artificial Neural Network

AUC Area under the ROC Curve

GBM Gradient Boosting Method

KNN K-Nearest Neighbour Model

LC Lending CLub

LDA Linear Discriminant Analysis

MCAR Missing completely at Random

MV Majority Voting

PCA Principal Component Analysis

RBF Radial Basis Function

RFECV Recursive Feature Elimination with Cross Validation

RF Random Forest

RMSE Root Mean Square Error

SMOTE Synthetic Minority Oversampling Technique

SVM Support Vector Machine

XGB Extreme Gradient Boosting

Chapter 1

Overview

1.1 INTRODUCTION

Machine Learning enables computers to behave and learn like humans do and further improve their learning capability through data, input in the form of real world interactions and observations [14]. Machine learning research is the part of research on artificial intelligence which provides computers knowledge through real world interactions which ultimately allows the computer to adapt to new settings [13]. This field of science and technology has taken the world by storm and as the days pass by it is making more and more contributions in different aspects of the modern world. Finance and banking is one such aspect. Tremendous work is being done in incorporating machine learning techniques with the banking industry in detecting scams, frauds or defaulters. With the help of pattern recognition algorithms complex decisions are being made every day throughout this industry. The ability of machine learning to recognize anomalies and patterns is being heavily used for proper overseeing of financial institutions. Author KY Tam in his paper showed us that neural networks can be used to assess the performance of banks and in turn help them to prepare for bankruptcy. This study helps to evaluate the financial condition of a bank and elaborates the effectiveness of using neural networks over other models [34]. R.H. Davis et al in his paper has discussed the usage of machine learning algorithms in assessing credit cards risk. A comparative analysis of accuracy scores using neural networks and other classifiers were performed. The paper was concluded with the notion that all the algorithms that had been used had shown similar accuracy but time complexity was higher for the neural networks [11].
But for this paper our field of interest is the assessment of credit risk of loan borrowers. The number of loan defaulters and charged off loans are at an all-time high. Assets are being frozen, transactions are being halted and lending institutions such as banks and finance companies are going through huge loss. In the year of 2018, it was reported that around 9

million loan defaulters exist in China alone [28]. The amount of default loans in Bangladesh has increased nearly three times since 2011 [26]. According to the statistics for 2018, each year more than a million student loans go into default in the United States and their education debt has increased to three times the original amount in the last ten years [27]. In case of India the amount of money owed by loan defaulters to the banks have quadrupled from the year of 2013 up until 2017 [35].The same study also tells us that in 2017 default loans have increased at an alarming rate of 27 percent. Experts suggest that the current scenario in Bangladesh will hamper the growth of businesses and put a halt to the implementation of various strategies in creating work opportunities for the general mass [19]. Therefore, we can clearly see that loan defaults not have a major negative impact on the financial institutions but can also weigh down the economy of a country.

A viable solution to this problem is to carefully select the people who deserve a loan. Banks or other loan providers should select only those applicants who have the lowest chance of defaulting. And this is where the power of machine learning and data science comes in. Machine learning can be applied in this scenario to develop a model which has the ability to understand and learn from the behavioral pattern of successful customers and loan defaulters. When there is a new applicant, the model can accurately predict the applicant's chances of defaulting the loan based on the patterns it learned beforehand and using this probability credit institutions such as banks or other loan companies can decide whether or not to accept an applicant's loan request. Many studies have demonstrated the effectiveness of applying machine learning techniques for credit risk assessment. In [29] the authors have used neural networks and genetic algorithm to prepare a model for credit risk assessment. Besides genetic algorithm, they have tested various other feature selection methods such as forward selection, information gain, gain ratio and Gini index and have concluded that for their data set a combination of neural network and genetic algorithm was the most optimum solution. For a more uniform result they have also applied k folds cross validation instead of the traditional train test split. In [21] the authors have also talked about using neural networks in predicting loan defaults. Here they have used three different models of artificial neural networks and tested them in nine different ways. Each neural network will have the same number of nodes in the input layer [3] and the output layer (1) but the number of nodes will differ in the hidden layer [39, 4]. They used nine different learning ratios for nine different ways of testing. The term learning ratio refers to the ratio of train test split. In simpler terms the author had split their data set in nine different ways to find out the optimum train test split ratio. A comparative analysis was made in the three above mentioned neural networks in nine different cases and the ultimate result shows us that the neural network with 23 nodes in its hidden model was the most optimum solution. They also added that the best result was

obtained when the learning ratio was 4:6.
Besides neural networks many other supervised classifiers have been used in this field. In the paper [17] the authors have opted to use Bayes net, Naïve Bayes and j48 algorithm for this purpose. Based on attributes such as gender, history of previous credit, occupation of the applicant, the purpose of loan, age, type of housing and the amount of credit, the authors have predicted whether or not a new applicant will be a loan defaulter or not. According to their study the j48 algorithm was their preferred choice with an accuracy of 78.3784 percent. Support vector machines (SVM) are also a very popular choice for classification problems. In the paper [40] the author has performed a comparative analysis between multi agent learning models and single agent models for credit risk assessment of new credit card applicants. As single agent models linear discriminant analysis, quadratic discriminant analysis, feed-forward neural network, logistic regression and support vector machine were used. And for the multi agent model three different SVM models were created and their individual score were later aggregated into one output using different ensemble strategies. Support vector machines with an rbf kernel, a sigmoid kernel and a polynomial kernel has been used for the initial predictions of the multi agent model. Later adaptive linear neural network (ALNN), TA-based (TA) weight averaging and majority voting based (MV) multi agent ensemble learning models were used as ensemble strategies to combine the SVM scores. Two folds cross validation technique was used to get a more complete result and grid search was used to find the optimum parameters for the classifiers. It was seen the multi agent ensemble technique outperformed the single agent based models in all cases. Additionally it was seen that the ALNN ensemble technique was the best ensemble strategy to use among the three. Therefore, we see that through machine learning this problem can be addressed and a solution can be provided. Accordingly this paper will discuss the application of different supervised algorithms along with several feature selection methods in working out the best investment for a loan providing institution by accurately predicting whether or not a new loan applicant can be a loan defaulter or not. Our data set includes personal history along with credit history of an applicant. Classifiers such as extreme gradient boosting, support vector machine, random forest and logistic regression will be used to identify the complex patterns that exist in the previous borrowers be it successful or defaulter and upon that knowledge we can classify a new applicant into a defaulter or a non-defaulter category. PCA and RFECV will be used to extract the optimum number of features and the exact features to use for the classification process to minimize computational cost and time. Later on 5 folds cross validation and grid search will be used to select the best solution for each model. And last a comparative analysis of each model will be made to select the most optimum model for credit risk assessment. The rest of the paper includes brief discussion about some relevant work that

has been done in this field. This will be followed by a detailed description of our proposed model and our data set. The later sections will discuss the steps of data pre-processing, the results and experimental analysis and finally the paper will be concluded with future works and concluding remarks.

1.2 LITERATURE REVIEW

The subject of credit risk assessment is a very weighty and talked about topic in the field of banking and financing. Furthermore, after the recent flourish in data science and several influential advancements in the field machine learning, this topic has gained even more significance. Many noteworthy research findings have been in this regard which act as a stepping stone for ongoing and future studies. Artificial Neural Network is a widely used technique in this field.

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information [1]. In [8] the authors used an RBF multilayer feed forward neural network and compared it with a logistic regression model. Their data set contained 492 test cases collected from Jordanian Commercial Banks. It was seen that the regression model was more accurate in properly classifying accepted applications and the neural network has the edge in classifying rejected cases. A similar comparison between these two models was also made in [6] where the authors used the chi square test to evaluate the bad customers. A thousand instances were used for the data set where the logistic regression outperformed the neural network.

Support Vector Machine is also an immensely popular method for classifying in the field of machine learning. In [37] the authors have used an SVM based on fuzzy logic to assess credit risk on three different data sets. The authors present a new bilateral weighted support vector machine where all the instances are treated as both good and bad to achieve better generalization capability. They performed a comparative analysis between their svm model with a logistic regression model, a linear regression model, a neural network model and other svm models. It was evident that their bilateral weighted svm model had a better overall accuracy but the authors further mentioned that the proposed model also had to bear huge computational cost. On another note, ensemble models are also a very interesting approach to the problem of loan defaulters. The authors in [39] have used neural networks to build an ensemble agent where de-correlation maximization technique was used to select the required neural net models for the ensemble agent. The data set was collected from the UCI Machine Learning Repository which consists of 653 instances of approved and denial credit card applications. The outputs of the selected neural networks were integrated using five different

reliability based ensemble strategy – maximum, minimum, median, mean and product. To make a comparative analysis the ensemble models were compared with single based agents (LogR, SVM, and ANN), hybrid agents (Neuro-fuzzy, Fuzzy SVM) and a voting based reliability ensemble model. It was seen that the reliability based neural network ensemble agent outperformed the other models marginally.

Research has shown various other methods can be implemented to achieve better results based on the data set and our desired computational time. In [24] we see the application of Gradient Boosting method (GBM) on a Brazilian bank data set. Additionally generalized linear modeling and distributed random forest were also used in this paper. Over 20 thousand instances were used for this research where 70 percent of the data set was used to train the model. It was shown that the GBM edged the aforementioned methods significantly with an AUC score of nearly 99 percent. Naïve Bayes is another prominent classifying algorithm which works on the basis of Bayes' theorem. In [5] the authors have used a Naïve Bayes model against a Linear Discriminant Analysis (LDA) model, a k-nearest neighbor model (knn), a Logistic Regression model, classification trees and a model based on neural networks. It was observed that the knn model performed the best and performance of the Naïve Bayes model was the poorest. Though the difference in scores are slight and comparatively insignificant, the authors suggest that the size of the data set is a major factor here regarding the poor performance of the Naïve Bayes algorithm. Furthermore the dependencies of the categories also contributed in the negative performance of the Naïve Bayes model.

Classification and regression trees (CART) or more popularly known as Decision trees are also a prevalent form of supervised learning algorithm which work on the basis of a tree structure. In the tree structure the internal nodes represent the features of the data set and the corresponding leaf nodes represent the target label. The authors in [20] have used such a model to assess credit risk of loan borrowers. In order to overcome the challenge of an imbalanced data set the authors have used adaptive boosting to weight the less prominent labels more heavily. The data set combines attributes related to account-balance, transactions data and data from the credit bureau. The data set had been divided into several train and test sets for better understanding of the trends and underlying developments. Each training set consisted of successful and delinquent instances for a 90 day period (3 months) and the results were evaluated on the instances of the next 90 days (3 months). Precision, recall, accuracy and AUC score were used as the performance metric and commendable scores were achieved in this model. 10 folds cross validation were also performed to get a more reliable and even score. In [30] the authors propose a decision tree model to evaluate the eligibility of a loan borrower where information gain has been used as the feature selection method.

After data pre-processing the data set was cut down to 3271 instances from an initial count of 4520. Ranker in Weka was used as the default search algorithm for feature selection and information gain method was used as the attribute evaluator. Different levels of accuracies were achieved based on the size test set ranging from 85 to 90 percent. The paper [4] also describes a case were the tree based models overcome the models based on neural network. In this proposal comparisons have been made between logistic regression, gradient boosting, random forest and neural network models. Modifications were made in the algorithms for better accuracy and lesser computational cost. For the logistic regression model the alpha and lambda hyper parameters were tuned by using elastic net approach. This helps to not "over regularize" the LogR model and achieve better accuracy [3]. The number of trees for the random forest and gradient boosting method where set to 120. Lastly four different neural network models were created with different number of hidden layers and different values of the regularization functions. A grid search was used to find the optimum values for the drop out ratio, activation functions, hidden layers and regularization functions and were set to one of the neural networks. For evaluating the results AUC score and root mean square error (RMSE) were calculated. The results have shown that the tree based models, that is the random forest and gradient boosting models outperformed the neural networks and LogR model by a significant margin. Furthermore, in the paper [36] the authors choose to make a comparative analysis between models formed using decision trees, artificial neural networks, naïve bayes classifier, k-nearest neighbor classifier and a model based on linear discriminant analysis. Moreover, ensemble models were also created using these classifiers and in all cases it was seen that decision tree and the model based on naïve bayes classifier gave the highest prediction scores. The authors also presented that it was difficult to find the best network topology for the neural network model and it was difficult to find the optimum value of k for the knn based model. Overall the best result was obtained by the model based on naïve bayes classifier which was presumed due to the fact that the attribute independence assumption of the classifier was not violated.

Therefore, it is evident from the past related works that much progress has been done in the assessment of credit risk and much more can be done. The aforementioned papers suggest that in order to develop a good model a proper feature selection procedure or dimension reduction procedure is essential depending on the data set being used. Also the use of different performance metrics are vital to properly assess the performance of a machine learning model. Parameter tuning, noise handling, correcting imbalanced data sets are fundamentals in solving a problem in the field of credit risk assessment. Much heed should also be paid to computational cost and time complexity. This paper proposes to create such a model which addresses all these issues. A comparative analysis between different supervised

classification algorithms will be made using different feature selection methods to choose the optimum combination for credit risk assessment. Parameter tuning and noise handling will be performed using grid search and the problem of an imbalanced data set will also be dealt with. Due to the complexity and computational cost of using neural networks, it will not be showcased in this paper. Overall our aim is to showcase a model which addresses all the above affairs and create an accurate model for assessing credit risk in the banking sector.

Chapter 2

Credit Risk Assessment Model

2.1 DATASET AND PREPROCESSING

In our proposed model the dataset from Lending Club will be used to perform credit risk assessment using supervised learning algorithms. The data set pre-processing stage will begin from here. Firstly dummy variables are instantiated to handle categorical values and the output label has been binarized. The desired output is now either a "1" which represents "Fully Paid" or a "0" which represents "Charged Off". Dimensionality reduction will be performed using RFECV as a candidate for feature selection methods and PCA as a member of the feature extraction family. Feature scaling will be performed before PCA and after RFECV. A train test split and a cross validation procedure will be used to evaluate the model performance. But before that SMOTE will be applied only on the training set to handle class imbalance. Finally grid search with cross validation will be used to tune the hyperparameters and will be fed into the classifiers for prediction. Lastly the predicted outcomes will be evaluated using different evaluation metrics and the results will be illustrated in tabular and graphical fashions.

The flow chart below illustrates the workflow that has been followed in formulating the proposed model.

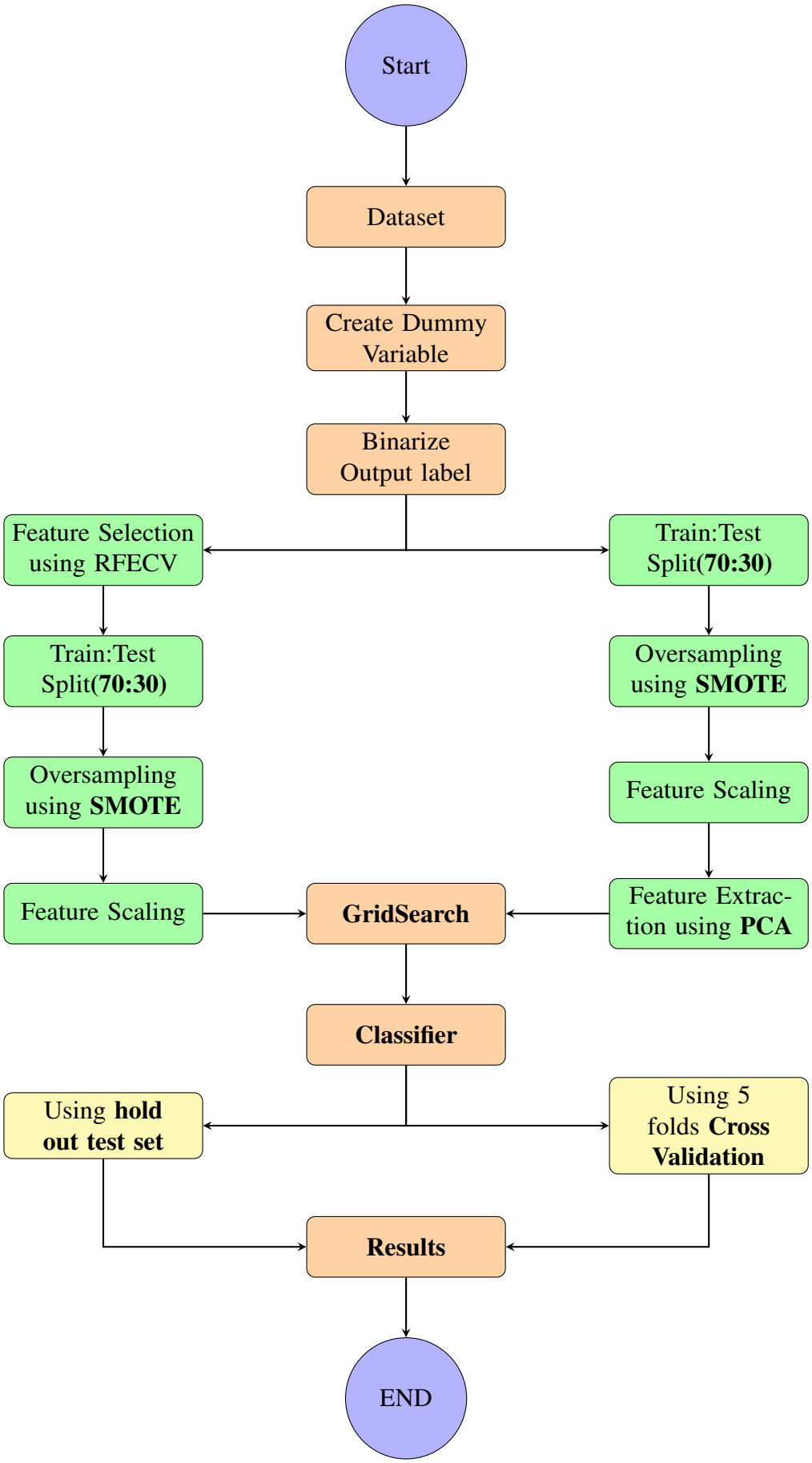


Fig. 2.1 Flowchart of proposed model

2.1.1 DATASET

The dataset for the proposed model is obtained from an online free repository controlled by Lending Club [2]. Lending club (LC) is an established and recognized peer to peer credit lending institution located in the United States. It is currently the largest peer to peer lending platform in the world [38]. Now, the fact that LC is a massively impactful and influential organization throughout the United States has already been established and so like all sophisticated financial institutions it performs a thorough and in depth analysis of all of its borrowers before they accept their loan request. Therefore, the dataset obtained from the above mentioned source contains records of all the loans that were given out by LC from 2007 to 2017. The features present in the dataset exhibit background history of the borrowers, previous records of any transactions, detailed information about any ongoing transactions with LC, information about current accounts in LC, information about previous successful or default loans and finally the borrower's current loan status. The current loan status of the borrower will be our target variable. The loan status indicates whether a borrower has defaulted or successfully repaid his previous loans. In a nutshell, each instance or row in the dataset represents a specific borrower's personal records (investigated by LC before accepting loan request), previous and current credit history and whether the borrower has defaulted the loan or not. Based on the trends that has been exhibited by these features we will try to predict whether the loan request for a potential borrower should be accepted or not.

Now proper data pre-processing is a very important step in developing any machine learning and it has been showed that efficiently use of data can make massive differences in results. As we are dealing with a dataset of high dimensionality, it is indispensable for us to implement a proper data pre-processing program which perfectly handles issues such as missing values, unwanted features, data imbalance or duplicate instances. A detailed description of the data pre-processing performed and a brief description of the final dataset after all the necessary pre-processing techniques is described in the next section.

2.1.2 PREPROCESSING

Starting from the year of 2007 up until the first quarter of 2017, records for more than 2 million borrowers have been instantiated in this dataset. Due to the obvious huge computational cost in handling a dataset of this magnitude, firstly the number of instances were reduced. In this case all the instances were sorted based upon the year of loan issuance and the loans for 2007 up until 2011 were selected. This drastically reduced the number of instances. Again loan_status, the target variable is a multi class variable which showed the borrowers who successfully repaid the loan, the borrowers whose accounts were charged off due to loan

default, the borrowers who paid the loan within the grace period, the borrowers who repaid the loans but with a fine and lastly the borrowers whose loan terms are currently ongoing. As our model will focus on the defaulters and the successful borrowers so we have kept the borrowers who loan status are either fully paid or charged off and have omitted the instances with fines or current ongoing loans. The next steps of pre-processing includes reducing the unwanted features and instances.

2.1.2.1 CLEANING THE DATA

Some features were initially omitted due to the presence of missing values or for having either zero or minimum variance. Mean or mode imputation are frequently used for imputing the missing values.Classifying algorithms can also be used to impute missing values. MissForest is such a technique which uses trees for handling mixed type variables [33]. The paper [7] discusses the use of k nearest neighbours in imputing missing values. But all these methods can be implemented if there is sufficient amount of data upon which the imputation will be based on. Otherwise the imputed values will be inaccurate and will diminish the quality and accuracy of the model. In light of the above notion, we were unable to use any imputation method on any feature as many features contained a huge number of missing values and as a result they were removed. Furthermore all the missing data are of the MCAR (Missing completely at Random) category and for this a sufficient relationship among other features for imputation could not be established. And so as per the teachings from [31] missing value imputation is not recommended at this stage. Also some features have shown zero or very minimum variance. Features with very less variance are not favorable for classification and work as a hindrance in properly separating the output variables. Therefore, in light of the above mentioned events the following actions have been taken for reducing unwanted features:

- The “ID” and “member” features were not used as they did not have relevant information regarding our analysis.
- “Grade” and “Sub_grade” columns were removed as the grades of the loans provided are not coherent in the assessment.
- The “Emp_title” feature which represents the occupation of the borrower was not selected because as the clients entered their designations in their own unique way, the sheer divergence created made the feature not favorable for our research.
- The “pymnt_plan” feature was deleted because it contained absolutely no variance in data.

- The “url” column contained the web address of specific clients from the official website of LC. And is it was deemed unfit for our model.
- The feature “Desc” was meant to carry all the description of the loan. But it was omitted as it had a massive number of missing values.
- The feature “Title” of the dataset contained the borrower’s reason behind the loan request. To keep the dataset coherent with their standards, the spaces between the words were removed by Lending Club and stored in a compact structure in a different column named “Purpose”. As a result the “Title” column was deleted and the “Purpose” feature was kept instead.
- “Zip_code” column was deemed unnecessary because the exact zip code was not mentioned in any case and a feature titled “addr_state” was already present to identify the location of the borrower.
- “mths_since_last_delinq” and “mths_since_last_record” were removed as more than 50 percent of the data was missing.
- The features “total_rec_late_fee”, “Out_prncp_inv”, and “out_prncp” were deleted due to very low variance.
- The feature “last_pymnt_d” could be used to assess the loan statues of the borrower but as we already have a feature which can assess that and so the feature “last_pymnt_d” was removed.
- The feature “next_pymnt_d” contained the next payment date of the borrower and was omitted from the model because it did not correspond to either defaulters or the borrowers who successfully paid off their debt.
- The Following features were also removed as more than 80 percentage of each features consisted of empty cells:
  - Collections_12_mths_ex_med
  - mths_since_last_major_derog
  - policy_code
  - application_type
  - annual_inc_joint
  - dti_joint

- verification_status_joint
- acc_now_delinq
- tot_coll_amt
- tot_cur_bal
- open_acc_6m
- open_act_il
- open_il_12m
- open_il_24m
- mths_since_rcnt_il
- total_bal_il
- il_util
- open_rv_12m
- open_rv_24m
- max_bal_bc
- all_util
- total_rev_hi_lim
- inq_fi
- total_cu_tl
- inq_last_12m
- acc_open_past_24mths
- avg_cur_bal
- bc_open_to_buy
- bc_util
- chargeoff_within_12_mths
- delinq_amnt
- mo_sin_old_il_acct
- mo_sin_old_rev_tl_op
- mo_sin_rcnt_rev_tl_op
- mo_sin_rcnt_tl

- mort_acc
- mths_since_recent_bc
- mths_since_recent_bc_dlq
- mths_since_recent_inq
- revol_bal_joint
- sec_app_fico_range_low
- sec_app_fico_range_high
- sec_app_earliest_cr_line
- sec_app_inq_last_6mths
- sec_app_mort_acc
- sec_app_open_acc
- sec_app_revol_util
- sec_app_open_act_il
- sec_app_num_rev_accts
- sec_app_chargeoff_within_12_mths
- sec_app_collections_12_mths_ex_med
- sec_app_mths_since_last_major_derog
- hardship_flag
- hardship_type
- hardship_reason
- hardship_status
- deferral_term
- hardship_amount
- hardship_start_date
- hardship_end_date
- payment_plan_start_datem
- hardship_length
- hardship_dpd
- hardship_loan_status

  - orig_projected_additional_accrued_interest
  - hardship_payoff_balance_amount
  - hardship_last_payment_amount
  - disbursement_method
  - debt_settlement_flag
  - debt_settlement_flag_datem
  - settlement_status
  - settlement_date
  - settlement_amount
  - settlement_percentage
  - settlement_term

- The features “tot_hi_cred_lim”, “total_bal_ex_mort”, “total_bc_limit” and “total_il_high_credit_limit” were removed as they were deemed unnecessary for our analysis.

After the initial steps of data preprocessing some extra measures were also taken for better and easier classification of the model. These steps include:

- The feature “Term” describes the duration in which the loan should be repaid by the borrower. The data contained the string “months” added to each value (for e.g 36 months or 60 months). The string “months” was removed from each cell keeping only the integer value. And thus the feature was converted from a string type to an integer type.

- The feature “emp_length” contained the duration of the borrower’s employment. The string “year” and “years” were removed from the feature (similar to the above measure). The employment duration of “<1 year” and “10+ years” were reduced to “1” and “10” respectively to reduce complications.

- The feature “Verification_status” contained the verification status of the client. This column contained strings such as “Source Verified”, “Verified” and “Not Verified”. The term “Source Verified” was changed to “Verified” in each cell as both the terms referred to the exact same thing.

- “Issue_d” and “earliest_cr_line” features were present in a dd/mm/year format. It was changed to an integer value of year only.

Therefore after handling missing values for both rows and columns these are the columns which have been deemed fit and favourable for our model :

1. **Loan_amnt :** The amount of loan requested by the borrower
2. **Funded_amnt :** The total amount committed by the borrower to that loan at that time
3. **Funded_amnt_inv :** The total amount committed by the investors to that loan at that time
4. **Term :** The duration of the loan
5. **Int_rate :** Interest rate on the loan
6. **Installment :** The payment the borrower has to make to repay the loan
7. **Emp_Length :** The duration of em
8. **Home_ownership :** A borrower's household status. Possible values are rent, mortgage, own, other
9. **Annual_inc :** The annual report that is submitted by the borrower during registration
10. **Verification_status :** Shows whether the income source of the borrower was verified or not by the Lending club
11. **Issue_d :** The specific year on which the loan was issued
12. **Loan_status :** Current Loan Status
13. **Purpose :** The borrower purpose for the loan request
14. **Addr_state :** Specifying the address (state) of the borrower
15. **Dti :** Debt to income ratio of the borrower
16. **Delinq_2yrs :** The number of times a borrower has exceeded the initial 30 days limit of delinquency in the past 2 years
17. **Earliest_cr_line :** The year the borrower's first credit line was open
18. **Fico_range_low :** The lower boundary range the borrower's FICO at loan origination belongs to.

19. **Fico_range_high :** The upper boundary range the borrower's FICO at loan origination belongs to.
20. **Inq_last_6mths :** Inquiries made about the borrower in the last 6 months without auto and mortgage inquirings
21. **Open_acc :** The number of open credit lines in the borrower's credit file.
22. **Pub_rec :** Number of critical public records
23. **Revol_bal :** Total credit revolving balance
24. **Revol_util :** Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
25. **Total_acc :** The total number of credit lines currently in the borrower's credit file
26. **Total_pymnt :** Installments received so far for total amount financed
27. **Total_pymnt_inv :** Installment received so far for the part of the total amount given by investors
28. **Total_rec_prncp :** Total principle received so far
29. **Total_rec_int :** Total interest received so far
30. **Total_rec_late_fee :** Total amount of late fees received so far
31. **Recoveries :** Post bad debt gross recovery
32. **Collection_recovery_fee :** Post bad debt collection fee
33. **Last_fico_range_high :** The upper boundary range the borrower's last FICO pulled belongs to.
34. **Last_fico_range_low :** The lower boundary range the borrower's last FICO pulled belongs to.

2.1.2.2 NUMERICAL FEATURES

“Loan_amnt”, “Funded_amnt”, “Funded_amnt_inv”, “Term”, “Int_rate”, “Installment”, “Emp_length”, , “Annual_inc”, “Issue_d”, “Dti”, “Delinq_2yrs”, “Earliest_cr_line”, “Fico_range_low”, “Fico_range_high”, “Inq_last_6mths”, “Open_acc”, “Pub_rec”, “Revol_bal”, “Revol_util”, “Total_acc”, “Total_pymnt”, “Total_pymnt_inv”, “Total_rec_prncp”, “Total_rec_int”, “Total_rec_late_fee”, “Recoveries”, “Collection_recovery_fee”, “Last_fico_range_high”, “Last_fico_range_low” - These are the numerical features that have been selected favourable for our model. Numerical features are needed for any classifier and are essential in separating the class variables.

2.1.2.3 CATEGORICAL FEATURES

The categorical features present in this dataset are namely : “Home_ownership”, “Verification_status”, “Loan_status”, “Purpose” and “Addr_state”. In this model we have used dummy variables to handle categorical values. The features “Home_ownership”, “Verification_status”, “Purpose” and “Addr_state” had been converted into binary dummy variables using the Pandas library. Here each categorical feature has been divided into multiple binary features depending on the different number of classes present in that feature. For example the feature “Verification_status” has 2 different classes or categories such as, Verified or Not Verified. In this case after applying the dummy variable operation on “Verification_status”, two new binary features have been created namely “Verified” and “Not Verified” and the feature “Verification status” has been removed automatically. Thus two new binary features, “Verified” and “Not Verified” have replaced the old categorical feature “Verification status”. This has been repeated for “Home_ownership”, “Purpose” and “Addr_state”. Now as “Loan_status” is the output / target variable, instead of dividing it into dummy variables we have used the LabelEncoder function from Scikit learn to convert it into a single binary feature where each defaulter (“charged off”) is represented by a “0” (zero) and each borrower who had successfully repaid their loan (“fully paid”) is represented by a “1” (one).

2.1.2.4 IMBALANCED DATASET

An imbalance dataset is such a case where there is major difference in the number of classification categories. In our domain our classification categories consist of “Fully paid” and “Charged off” where the number of “Fully paid” cases outnumber the number of “Charged off” cases. Nearly 86 percent of the cases observed are non defaulters. In such a situation a model becomes more inclined to the majority class and cannot properly identify the minority class. To solve this issue we can over sample the minority class or under sample the majority

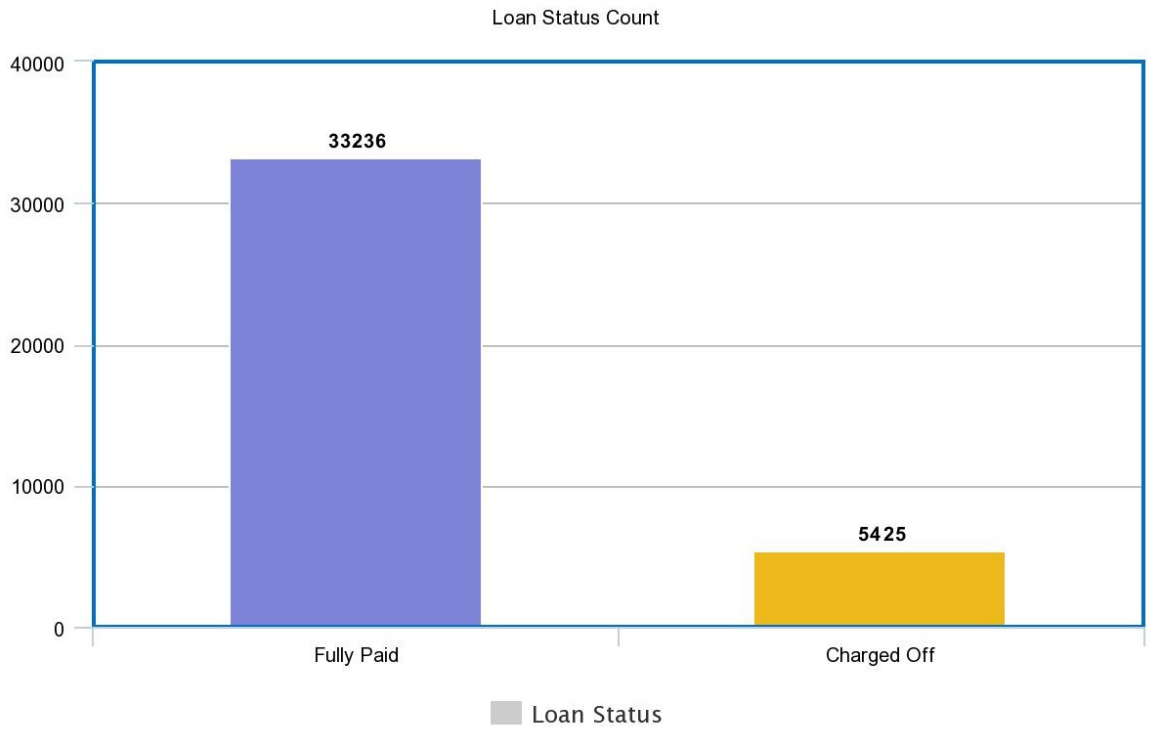


Fig. 2.2 Count Plot of Dataset

class. But under sampling the majority class will act as a hindrance in properly understanding the trends in our independent features. Furthermore, only over sampling the minority class will also not solve this as the techniques lying behind the over sampling will also matter greatly. Thus in such a scenario we have used Synthetic Minority Oversampling Technique (SMOTE). This technique which has been first described in [9] uses both oversampling and undersampling. Synthetic instances of the minority class are created to reduce the margin between the majority and minority class. For our model we have used SMOTE to increase the minority class and keep equal number of defaulters and non defaulters. But we have to mention that SMOTE will only be applied on the training set keeping the test set pure and untouched. And therefore, this will help us to properly classify the borrowers keeping the model aware of both the output classes.

2.1.2.5 FEATURE SCALING

The classifiers that will be used in this model work on the basis of euclidean distance. The euclidean method is used to calculate the distance between two points. It is very common for a feature to have a wide range of values. Now if a feature has a broader range of values in comparison with other features, then in this case the euclidean distance calculated will be

dictated by this particular feature. This will hamper the classification process and will result in a biased model. To solve this issue all the features are scaled so that the contribution of each feature is maximized proportionately.

Normalization (min-max scaling) and standardization (z-score) are two popular methods for scaling the features. For this model we have chosen to standardize our dataset rather than normalize. Classifiers such as support vector machines, logistic regression or neural networks prefer standardization over normalization. Also our model proposes to use such feature extraction methodologies where standardization is preferred over normalization as standardization helps to illustrate the features which maximize variance.

$$\textbf{Standardization} = \frac{x - \mu}{\sigma} \tag{2.1}$$

Here z is the standardized value, xis the original value of an instance of a feature, is the average or mean of the values of the feature and is the standard deviation. Standard deviation has been applied using in our model using the StandardScaler method from the scikit learn library. And therefore, this is how feature scaling has been implemented in our model.

2.1.2.6 SPLITTING THE DATASET

For the purpose of checking the performance of any machine learning model in an effective manner, splitting the dataset is an invaluable task. Splitting the dataset helps to prevent overfitting by evaluating the performance of the model on a portion of the dataset upon which the model has not been trained. We have used a 70:30 train test split ratio for our supervised model. This has been done using the "train_test_split" function from scikit learn library. This means that 70 percent of the entire dataset will be used to train the model and the remaining 30 percent will be used to evaluate the performance of the model.

But splitting the dataset using the train test split does have a drawback. As we are using a very specific portion of the dataset for evaluation we cannot be sure that the same accuracy will be achieved if a different portion of the same dataset was used for testing. This scenario can be handled using the cross validation technique [22]. Here firstly the entire dataset is divided into different folds and each fold is divided into train and test set respectively. Then the accuracy of each fold is taken and the mean of all the accuracies will be the final outcome. For our model we have used both these techniques for evaluation because though cross validation will give us a much more balanced overall performance, the train test split is necessary to understand the generalizability of the model.

Finally the pre-processing stage of our model is concluded here. Our dataset has been

cleaned and is now ready to be fit into the model for optimization and evaluation. The next section of the paper discusses the different classifiers and feature selection / extraction methods used in implementation of the model.

2.2 DIMENSIONALITY REDUCTION

In a supervised machine learning model the output label is predicted and evaluated using the feature vectors i.e. the independent variables. The classifier learns from the features, understands their trends and uses that learning in predicting and evaluating the future outcomes. In this regard selecting the optimum features and removing redundancy will not only make the model more precise and cost effective but will also help to generalize the model and help it to differentiate between the output labels properly. Dimensionality reduction will reduce the number of feature vectors based upon correlation between the feature vectors and their contribution in predicting the overall outcome. As a result using dimensionality reduction techniques we will be weeding out the correlated features to avoid redundancy, remove the features who can deemed unnecessary for output prediction to avoid great computational cost and make the model more adaptable to new data by omitting irrelevant constraints.

Feature selection and feature extraction are two ways of dimensionality reduction. In this model he have proposed to use both as we will not only be focusing on the optimum classifier but also the optimum feature reduction method.

2.2.1 FEATURE SELECTION

Feature selection is the process by which we reduce the dimensionality of our data set by selecting a subset of the original features for future prediction and evaluation. The elimination of features is based upon their performance and contribution in the overall prediction process. In this paper we have used “Recursive Feature Elimination with Cross Validation” (RFECV) as the feature selection method. In this method the weakest or the least contributing features are removed recursively until the best features are obtained.

In most feature selection methods the feature selection algorithm does not know the optimum number of features needed for the classifier and this number is later obtained by repeated trials of the algorithms. Now this drawback can be perfectly covered using recursive elimination with cross validation method. We use a five fold cross validation for RFECV which means that the feature selection process will be repeated five times and in each fold the importance of the features will be duly noted. The final ranking will be made using the mean of the feature importances of the feature vectors of all the five folds. And so not only will the

algorithm give us the weakest features but also the optimum number of features required for this specific model. The features will be reduced iteratively based on their importance until the optimum number of features are met. The algorithm is implemented using the feature selection library from scikit learn.

2.2.2 FEATURE EXTRACTION

Feature extraction, unlike feature selection does not necessarily omit the irrelevant features rather it extracts necessary data from the given features and forms new more compact features leaving the leaving behind the unnecessary redundant data. But the process is done in such a way that even though some information is disregarded, it will not hamper the performance of the model.

A very popular method for feature extraction is Principal Component Analysis (PCA) [25, 32]. In this method the original features are reduced to new feature vectors (commonly known as principal components) which exhibit the derived necessary information from the original features. PCA always tends to extract the maximum amount of variance from the features. PCA can be tuned to our need using the eigenvalues generated or variance selection. We have used the later here. Here we have selected to extract 85 percent variance from our original set features. This is implemented in our model using scikit learn library.

2.3 CLASSIFIER

A precise classifier is the backbone of any machine learning model. As we will conduct our experiments in a supervised environment and so we have chosen four supervised algorithms : Support vector machine (SVM), Logistic Regression (LR), Extreme Gradient Boosting (XGB) and Random Forest (RF). Furthermore, the hyperparameters of these algorithms were tuned using GridSearchCV to select the best set of values for our purpose. These will be discussed in details in the next sections of this paper.

2.3.1 PARAMETER TUNING

Tuning the hyperparameters of any classification algorithm is an essential task to formulate an efficient and optimized model. The hyperparameters can take in a wide range of values and it is essential to select the best value for each classifier so that the model can be the possible best for any given scenario. There are many popular methods for parameter tuning GridSearchCV is one such method.

GridSearchCV is a hyperparameter tuning procedure where an exhaustive cross validation search is performed to find the best values for our desired hyperparameters. In [18] the author has shown how the penalty and gamma parameters can be perfectly tuned using grid search. Studies done in [12] perfectly illustrate how a five folds cross validation grid search can be effectively used to tune the parameters of a linear classifier like svm. And so in our model we have also tuned the parameters of our classifiers using a five fold cross validation grid search. This has been applied using model selection class from scikit learn. Below we have shown the parameters which have been tuned using grid search for each classifier :
SVM : “kernel” , “C”
LR : “penalty parameter”, “C”
RF : “max_depth", "max_features", "min_samples_leaf", "bootstrap", "min_samples_split" , "n_estimators"
XGB : “n_estimators”, “gamma”, “subsample”, “colsample_bytree”, “max_depth”

2.3.2 SUPPORT VECTOR MACHINE

Support vector machines, first introduced by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1963 is a supervised learning model which can be used for both classification and regression problems. When training data is fed into an svm model, the algorithm separates the data into two categories using a hyperplane. Then when new data is fed for testing, the svm model allocated the new data to either one of the categories. The algorithm makes sure that the classes are separated by the hyperplane are as distance as possible for clear classification.

Linear separation as well as non linear separation can be done successfully using svm models. By simply using the kernel trick non linear classification can be done using support vector machines. In this paper using grid search we have pointed out that for our data set a linear kernel gives better classification accuracy than non linear kernels. Furthermore svm works extensively well with large data sets which gives us more reason of using such a classifier for our model.

2.3.3 LOGISTIC REGRESSION

In regression analysis logistic and linear regression are two very commonly used algorithms but unlike linear regression and logistic regression model deals with a much limited number of labels. The labels of a linear regression model can be of continuous forms but that of a logistic regression cannot. In case of linear regression the exact outcomes of the labels can be predicted but with logistic regression we can calculate the probability of a specific outcome.

The probabilities are calculated using a logistic function and as we will be using a binary LR so the output labels will be represented in binary form (either 0 or 1). The formula used in the process can be defined as :

$$\mathbf{log}(\frac{\mathbf{p(X)}}{\mathbf{1\text{-}p(X)}}) = \mathbf{a + bX} \tag{2.2}$$

Here the above function can be described as the logistic function used in estimating the probabilities of the label. The term "X" in the above formula represents the input or the independent variable and "p(X)" represents the dependent variable and "b" represents the regression coefficient. Also (p(X) / 1 - p(X)) is known as the odd ratio and the log of the odd ratio is used in finding the probabilities of the outcomes. We have tuned the penalty and C hyperparameters of this LR model and have found that L1 regularization which uses lasso regression is favoured over the traditional L2 regularization which favours ridge regression.

2.3.4 RANDOM FOREST

Random forest is a tree based supervised learning model. It is an ensemble learning method which can be used for both regression problems and classification problems. It is an advanced version of the traditional Decision tree classifier. In this case multiple decision trees are formed and merged together which ultimately results in a more precise correction of the overfitting issue and enables the model to give out a more accurate prediction [23]. The hyperparameters max_depth, max_features, min_samples_leaf, bootstrap, min_samples_split, n_estimators were tuned using grid search and different values were obtained for both of the dimensionality reduction methods used here.

2.3.5 EXTREME GRADIENT BOOSTING

Extreme gradient boosting shows an unique way of using the gradient boosting method and ultimately giving a much more efficient classifying algorithm than traditional tree based methods. It was first introduced in [10] and allows the implementation of both a tree based model and a linear model. It is based on the works done in [15, 16] and paves a path for better performance and speed in comparison with the traditional decision tree classifiers. To further enhance the performance of this algorithm the parameters namely: n_estimators, gamma, subsample, colsample_bytree, max_depth were tuned. It can be used for both classification and linear model predictions. It was implemented using the xgboost library.

Chapter 3

EXPERIMENTAL RESULTS ANALYSIS

3.1 CONFUSION MATRIX

A confusion matrix is a great evaluator of the performance of a model. It perfectly illustrates the total number of true positive, true negative, false positive and false negative values. Further metrics such as f1 score or precision or recall can be derived from a confusion matrix

Table 3.1 Confusion Matrix

	Predicted Negative	**Predicted Positive**
Actually Negative	TRUE NEGATIVE	FALSE POSITIVE
Actually Positive	FALSE NEGATIVE	TRUE POSITIVE

3.2 ACCURACY

This performance metric is the most commonly used evaluator. The accuracy of a model shows the ratio between the number of correctly predicted labels and the total number of predictions done by the model. Accuracy can be formulated using the given formula :

$$\textbf{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{3.1}$$

Here TP stands for True Positive, TN for True Negative, FP for False Positive and FN for False Negative.

3.3 PRECISION

Precision shows the ratio between the true positive values and the sum of the true positive and false positive values. In simpler terms it shows the ratio of the correctly predicted borrowers who successfully repaid their loans with the total number of successful borrowers predicted by the model.

$$\textbf{Precision} = \frac{TP}{TP+FP} \tag{3.2}$$

3.4 RECALL

Precision shows the ratio between the true positive values and the sum of the true positive and false positive values. In simpler terms it shows the ratio of the correctly predicted borrowers who successfully repaid their loans with the total number of actual successful borrowers.

$$\textbf{Recall} = \frac{TP}{TP+FN} \tag{3.3}$$

3.5 F1 SCORE

F1 score helps to regulate the balance between precision and recall. It is the harmonic average of precision and recall. It can be formulated as :

$$\textbf{F1 score } = \frac{Precision * Recall}{Precision + Recall} * 2 \tag{3.4}$$

3.6 ROC CURVE

The Receiver Operating Characteristic curve or ROC curve in short illustrates the diagrammatic comparison between the true positive rate and the false positive rate at all thresholds. The figure given below illustrates an ROC curve.

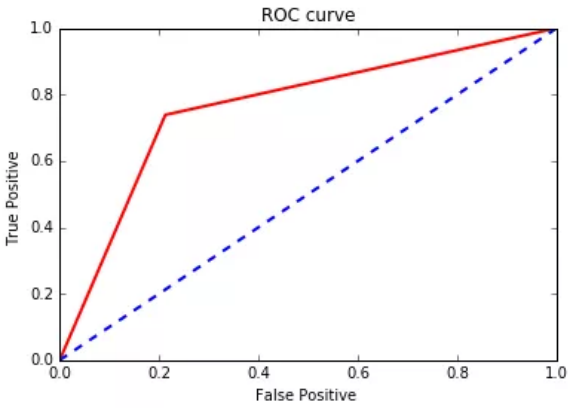


Fig. 3.1 ROC CURVE

3.7 AUC SCORE

This is also an important metric which helps to analyze the performance of a model. The Area under the ROC curve or AUC in short described the total area covered under the ROC curve generated by the model. The highest value for this metric is 1 and lowest is 0. The below figure shows the AUC generated from a ROC curve.

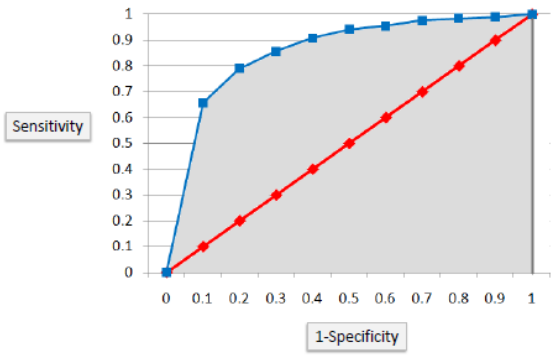


Fig. 3.2 AUC Score

As the performance metrics have been discussed, now the overall performance of all the classifiers will be evaluated in a comparative analysis in the next section.

3.8 RESULT ANALYSIS

In this section a comparative analysis will be made between the supervised learning models based on the performance metric discussed above. The results will be discussed in two categories. Firstly a comparative analysis will be made between the classifiers using both

PCA and RFECV using train test split (pure test set). Then another comparative analysis will be made using the same classifiers and same dimensionality reduction procedure but using a 5 folds cross validation to get the accuracy. All the hyperparameters were tuned using 5 folds cross validation.

3.8.1 USING PURE TEST SET

In this case as we have discussed above the performance metrics will be measured after the train test split where SMOTE has been applied on only on the training set keeping the test set pure. Furthermore GridSearchCV has not been applied here.

Table 3.2 PCA with Pure Test Set

Details	PCA			
	Logistic Regression	**Support Vector Machine**	**Random Forest**	**XGB**
Precision	0.987	0.982	0.945	0.971
Recall	0.969	0.961	0.902	0.898
F1 Score	0.978	0.971	0.923	0.980
AUC Score	0.984	0.974	0.888	0.945
Accuracy	0.963	0.923	0.871	0.889

Table 3.3 RFECV with Pure Test Set

Details	RFECV			
	Logistic Regression	**Support Vector Machine**	**Random Forest**	**XGB**
Precision	0.995	0.998	0.998	0.998
Recall	1.0	1.0	1.0	1.0
F1 Score	0.997	0.999	0.999	0.999
AUC Score	0.998	0.999	0.999	0.999
Accuracy	0.996	0.998	0.999	0.999

The above tables (3.2 and 3.3) show a comparison between the scores generated using classifiers after applying PCA and the same classifiers after applying RFECV using the train test split. Here we see that the tree based model using RFECV as feature selection shows the most promise in accurately identifying and classifying the loan cases. They edge the linear models by a very small margin.

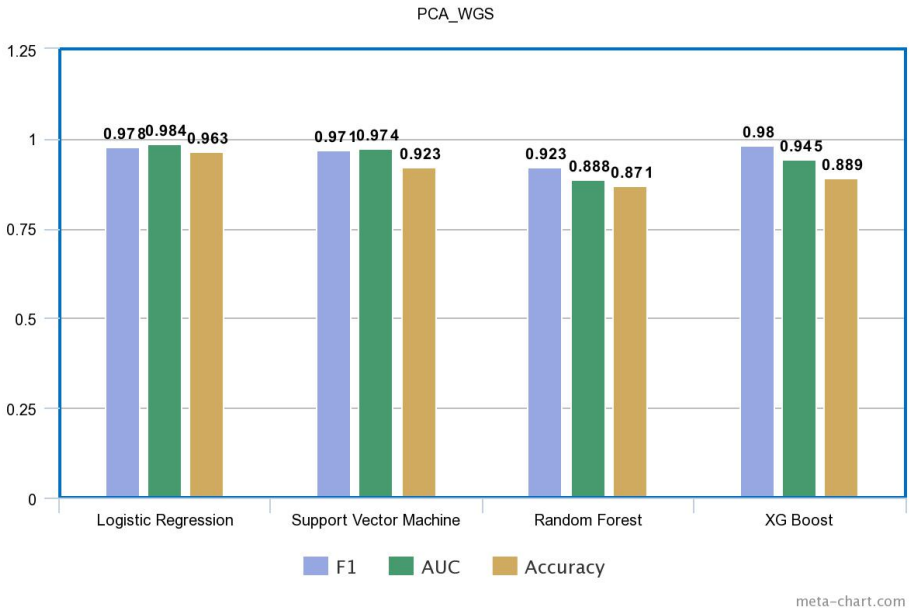


Fig. 3.3 Model Comparison using PCA and train test split

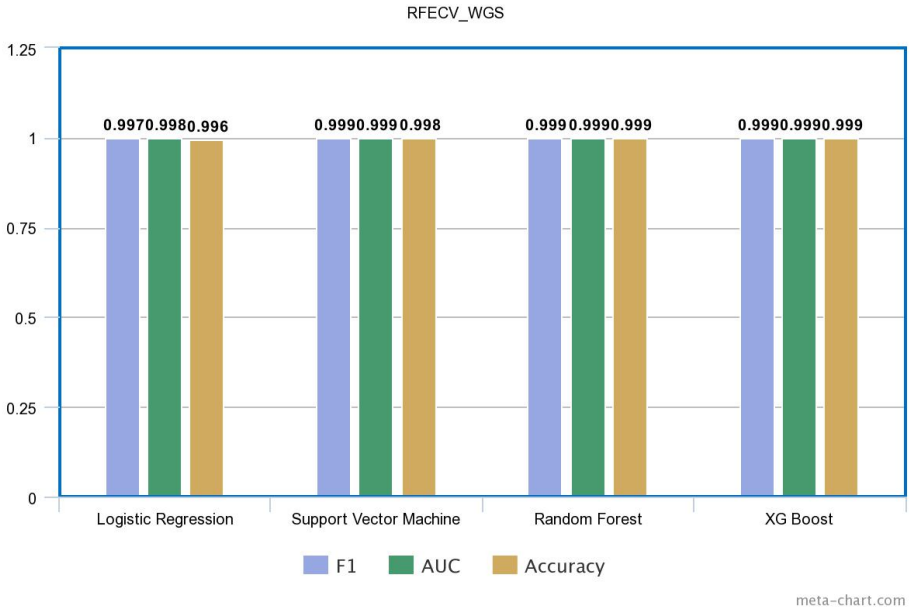


Fig. 3.4 Model Comparison using RFECV and train test split

The above graphs help to illustrate the effectiveness of the tree based models using RFECV over all other combinations used in this model. The train test split helps to understand the generalizability of the model and measures the adaptability of the model on new data. In this regard we see that the tree based models followed by the svm model outscore the logistic regression model.

3.8.2 USING CROSS VALIDATION

In this case accuracy will be measured using a 5 folds cross validation and the hyperparameters will be tuned using GridSearchCV. This is the overall accuracy that we will be,looking at mostly.

Table 3.4 PCA with Cross Validation

Details	PCA			
	Logistic Regression	**Support Vector Machine**	**Random Forest**	**XGB**
Precision	0.984	0.976	0.921	0.965
Recall	0.953	0.954	0.965	0.966
F1 Score	0.968	0.965	0.943	0.966
AUC Score	0.978	0.971	0.910	0.969
Accuracy	0.951	0.945	0.954	0.971

Table 3.5 RFECV with Cross Validation

Details	RFECV			
	Logistic Regression	**Support Vector Machine**	**Random Forest**	**XGB**
Precision	0.998	1.0	0.999	0.999
Recall	1.0	1.0	1.0	1.0
F1 Score	0.999	1.0	0.999	0.999
AUC Score	0.999	1.0	0.999	0.999
Accuracy	0.999	0.999	0.999	0.999

Here the above tables (3.4 and 3.5) illustrate the evaluation of the model after applying GridSearchCV and a 5 folds cross validation. We see that the SVM model based on RFECV outperforms all the other models. The SVM models is followed by the tree based model and then by the logistic regression model.

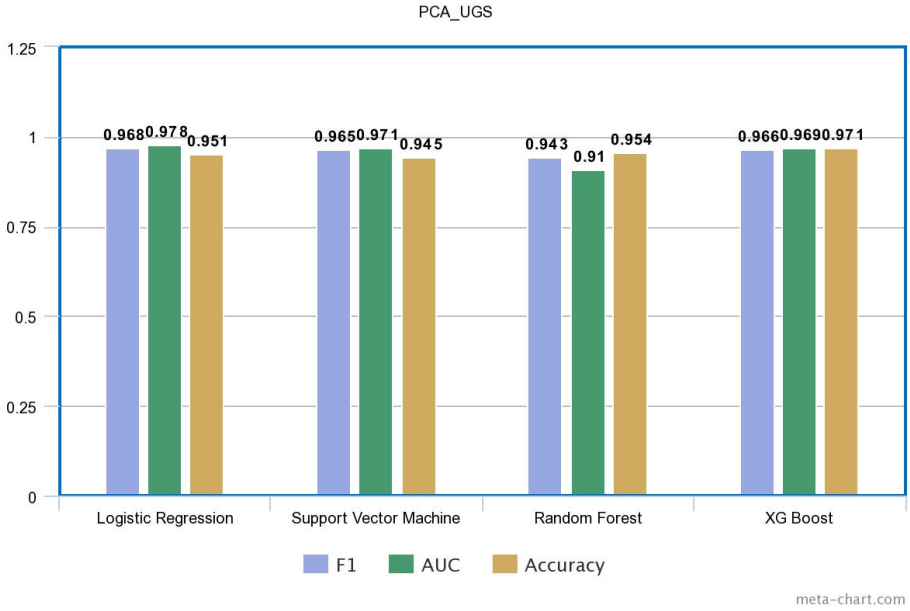


Fig. 3.5 Model Comparison using PCA and Cross Validation

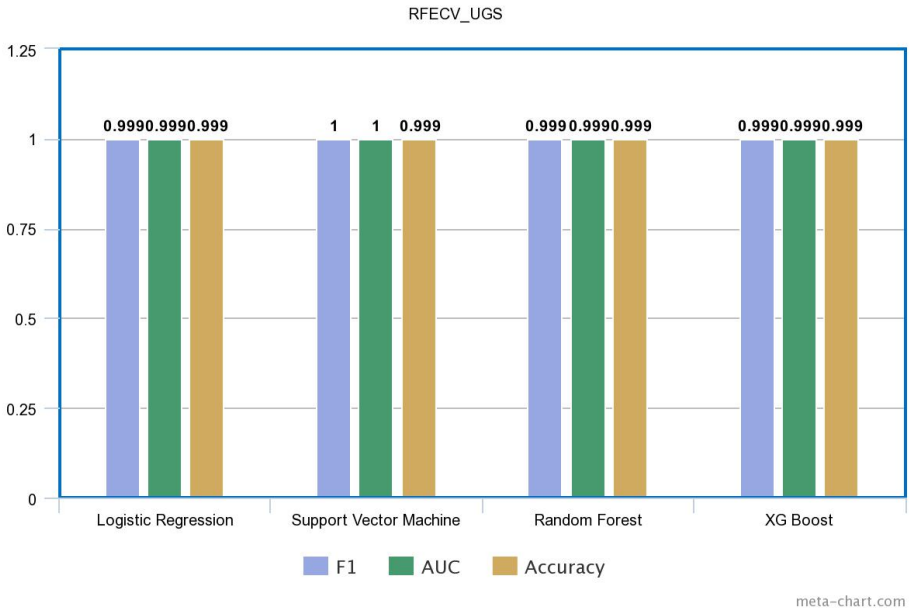


Fig. 3.6 Model Comparison using RFECV and Cross Validation

The above graphs illustrate how the svm model overcame the tree based models after applying GridSearchCV to tuned the hyperparameters. Both the AUC score and F1 score of the SVM model are maximized. But it is to be mentioned that even here the RFECV based models outperform the PCA based models.

Table 3.6 Variance Comparison for Different Algorithms using PCA

PCA	
Algorithm	**Variance**
Logistic Regression	0.00659
Support Vector Machine	0.00104
Random Forest	0.04076
XGB	0.01832

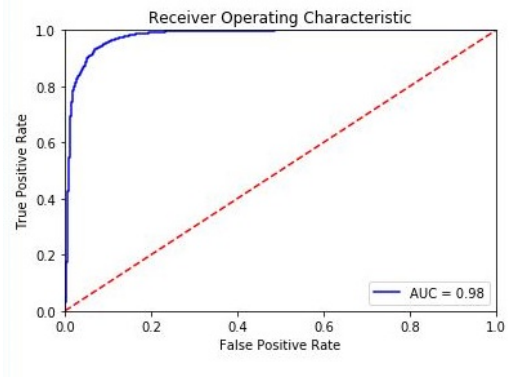
Table 3.7 Variance Comparison for Different Algorithms using RFECV

RFECV	
Algorithm	**Variance**
Logistic Regression	0.00078
Support Vector Machine	0.00012
Random Forest	0.00104
XGB	0.00074

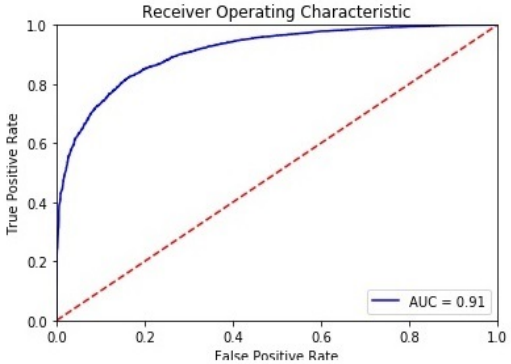
The above two tables show the variance of each of the 5 folds that were created during the evaluation of every model. The less the variance the more precise and stable our model will be. Here we also see that the SVM model excels in both cases whereas there is slight difference between LR and XGB followed by the RF model.

3.8.3 ROC CURVE ANALYSIS

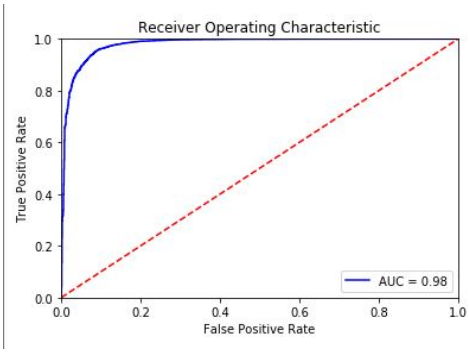
Below we have shown a comparative analysis of all the models using the ROC curves. Here all the ROC curves have been generated after the cross validation procedure and hyperparameter optimization.



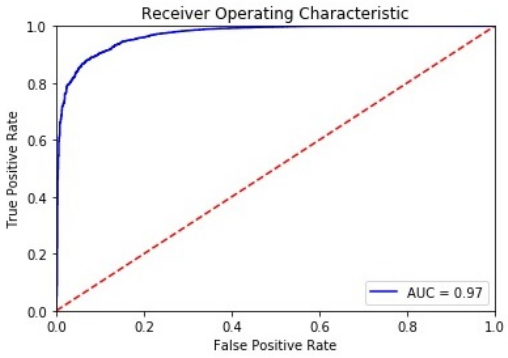
(a) ROC curve for Logistic Regression using PCA



(b) ROC curve for Random Forest using PCA

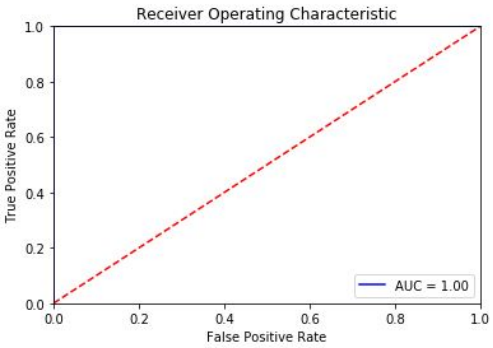


(c) ROC curve for SVM using PCA

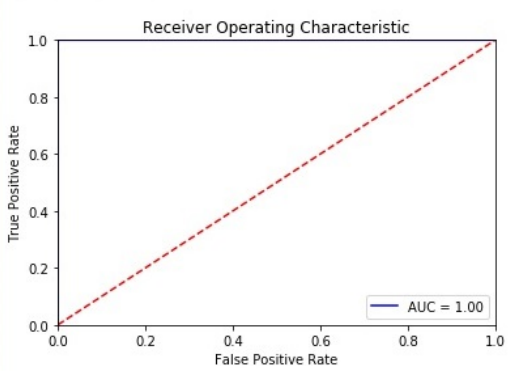


(d) ROC curve for XGB using PCA

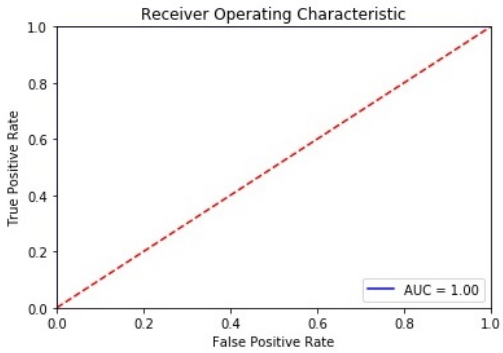
Fig. 3.7 ROC Curves for PCA



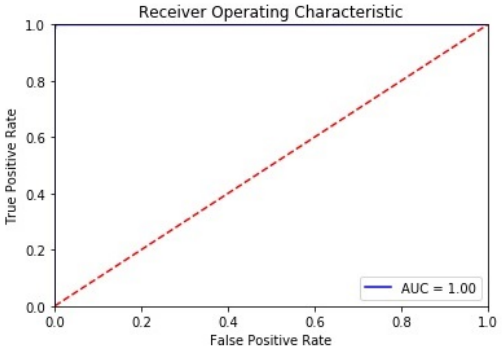
(a) ROC curve for Logistic Regression using RFECV



(b) ROC curve for Random Forest using RFECV



(c) ROC curve for SVM using RFECV

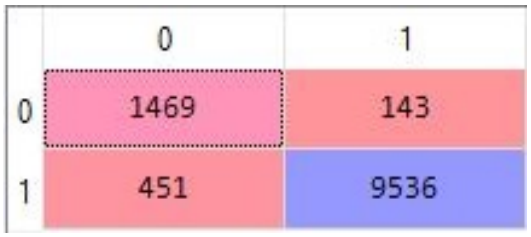


(d) ROC curve for XGB using RFECV

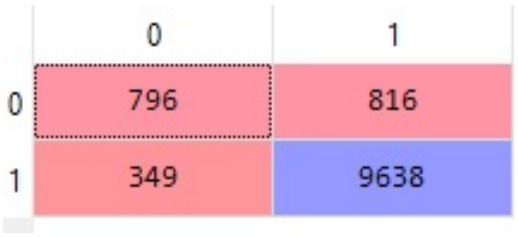
Fig. 3.8 ROC Curves for RFECV

3.8.4 CONFUSION MATRIX ANALYSIS

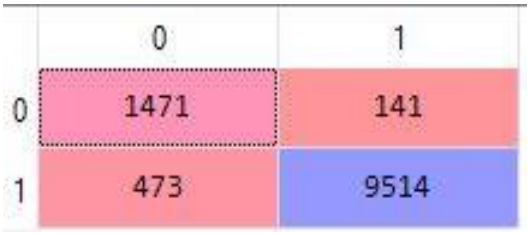
Below we have illustrated the confusion matrices for all the models used in this paper.



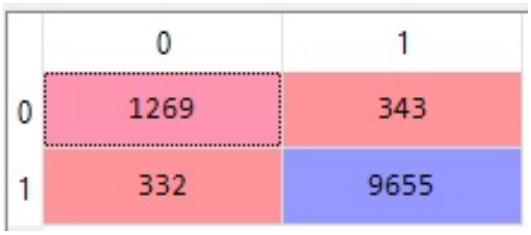
(a) Confusion Matrix for Logistic Regression using PCA



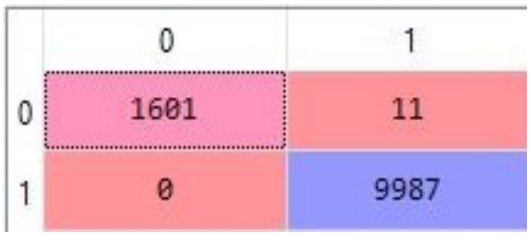
(b) Confusion Matrix for Random Forest using PCA



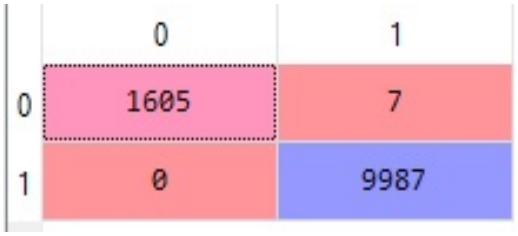
(c) Confusion Matrix for SVM using PCA



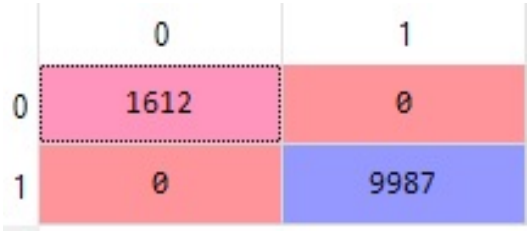
(d) Confusion Matrix for XGB using PCA



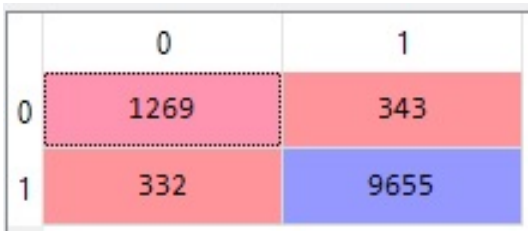
(e) Confusion Matrix for Logistic Regression using RFECV



(f) Confusion Matrix for Random Forest using RFECV



(g) Confusion Matrix for SVM using RFECV



(h) Confusion Matrix for XGB using RFECV

Fig. 3.9 Confusion Matrix

3.9 SUMMARY

We have used SMOTE to handle the problem of an imbalanced data set. Now when using SMOTE we have to be cautious in keeping the test set untouched and pure. And so we have applied SMOTE only on the training set and have used a hold out test set to evaluate the models. This type of evaluation helps to assess the adaptability of the formed models.

Again we have also used a 5 fold cross validation to evaluate our models. A train test split though may help us to understand a model's generalizability, it cannot give us a good overall performance of our model. And so as we have first used a grid search to tune the hyperparameters of our model and then applied a 5 folds cross validation to get a more stable and overall score which will give us a broader view of our model.

In these two situations we have used a feature selection method against a feature extraction method and made a comparative analysis using four different supervised learning models. This setup will tell us the best combination dimensionality reduction technique and classifier that will go with our data set.

From the test results we see that in all cases the classifiers using RFECV outperform the classification models which use PCA. Furthermore, in case of the train test split scenario we observe that the tree based models (RF and XGB) outperform the linear models SVM and LR be a small margin. But after using GridSearchCV to tune the parameters and cross validation to find the mean accuracy we see that the SVM model outclass the tree based models in all cases.

We see that all the models presented in this paper has brought out acceptable results but among them support vector machines have brought the most promising conclusion. The appropriate choice of kernels and penalty parameter is a major factor in the SVM's performance. Furthermore we also see the rise in the performance of the LR model when using PCA. LR models tend to perform well when there is less correlation between features which is achieved using PCA. The performance tree based models have come in between the SVM and the LR model but it is to be duly noted that the computational cost the tree based model have been recorded as the least.

Therefore it is our finding that for our data set and for fulfilling the objective that we have set out at the beginning of this paper, the SVM model and the XGB model are the best performers followed by the RF model and lastly the LR model.

Chapter 4

CONCLUSION AND FUTURE WORKS

4.1 CONCLUSION AND FUTURE WORKS

In the debate between which supervised learning model to use, we have come to the conclusion that support vector machines (SVM) or extreme gradient boosting models can outperform other tree based or linear models if the setup of the experiment is similar to that of ours. Furthermore in the debate of which dimensionality reduction technique to use, our model has shown us that recursive feature elimination with a five fold cross validation can outperform models based on principle component analysis.

Now as we have discussed before, computational cost is also an issue here. It is to be noted that the SVM model had taken the most amount of time to train. Now in the future to reduce computational time we would like to the use Apache Hadoop in building further supervised models. Furthermore, we have used the data ranging from the year of 2007 to 2011. For future improvements we would like to use all the data for illustrating a better understanding of the trends present in this field.

We have mentioned that in order to reduce computational cost we have omitted the idea of using neural networks. But if we get the chance to work with larger amounts of data in this field we would like to make a comparative analysis using neural networks as well. It is a known fact that neural networks tend to perform better with massive amounts of data and we would like to implement this hypothesis in our future works.

Again as we are also discussing the contributions of feature selection / extractions techniques, we would like to implements other dimensionality reduction techniques such as genetic algorithm, univariate feature selection methods, tree based feature selections etc to gauge their performances as well.

Besides, we would like to mention that besides GridSearchCV there are other ways we can tune the hyperparameters. Though GridSearchCV performs an exhaustive search and

checks for all possible combinations, we would like to use methods such as RandomizedCV, which has been said to take less computational time and finally compare the results.

Therefore we would like to conclude with the statement that this paper illustrates an interesting approach in identifying loan defaulters in the current ever changing economy. Using the dataset from Lending Club our model has brought about remarkable results which in turn can play a major role in assessing credit risk of borrowers and enable all the worldwide financial institutions to keep operating in a transparent and profitable way.

References

[1] Neural network. https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html.

[2] Peer to peer lending and alternative investing. https://www.lendingclub.com/.

[3] (2017). Variable selection with elastic net. https://www.r-bloggers.com/variable-selection-with-elastic-net/.

[4] Addo, P. M., Guegan, D., and Hassani, B. (2018). Credit risk analysis using machine and deep learning models. *Risks*, 6(2):38.

[5] Antonakis, A. and Sfakianakis, M. (2009). Assessing naive bayes as a method for screening credit applicants. *Journal of applied Statistics*, 36(5):537–545.

[6] Attigeri, G. V., Pai, M., and Pai, R. M. (2017). Credit risk assessment using machine learning algorithms. *Advanced Science Letters*, 23(4):3649–3653.

[7] Batista, G. E. and Monard, M. C. (2003). An analysis of four missing data treatment methods for supervised learning. *Applied artificial intelligence*, 17(5-6):519–533.

[8] Bekhet, H. A. and Eletter, S. F. K. (2014). Credit risk assessment model for jordanian commercial banks: neural scoring approach. *Review of Development Finance, Universiti Tenaga Nasional (UNITEN), 43000 Kajang, Selangor, Malaysia*, 4(1):20–28.

[9] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

[10] Chen, T., He, T., Benesty, M., et al. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, pages 1–4.

[11] DAVIS, R. H., Edelman, D., and Gammerman, A. (1992). Machine-learning algorithms for credit-card applications. *IMA Journal of Management Mathematics*, 4(1):43–51.

[12] Duan, K.-B. and Keerthi, S. S. (2005). Which is the best multiclass svm method? an empirical study. In *International workshop on multiple classifier systems, Nanyang Technological University, Nanyang Avenue, Singapore*, pages 278–285. Springer.

[13] Faggella, D. (2017). The rise of neural networks and deep learning in our everyday lives - a conversation with yoshua bengio -.

[14] Faggella, D. (2018). What is machine learning? - an informed definition. https://www.techemergence.com/what-is-machine-learning/.

[15] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics, Stanford University, USA*, pages 1189–1232.

[16] Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis, Stanford University, Stanford, CA 94305, USA*, 38(4):367–378.

[17] Hamid, A. J. and Ahmed, T. M. (2016). Developing prediction model of loan risk in banks using data mining. *Machine Learning and Applications: An International Journal,University Khartoum, Sudan*, 3(1):1–9.

[18] Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). A practical guide to support vector classification. *National Taiwan University, Taipei 106, Taiwan*.

[19] Islam, S. (2017). Bad loans cripple the banking sector.

[20] Khandani, A. E., Kim, A. J., and Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34(11):2767–2787.

[21] Khashman, A. (2010). Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes. *Expert Systems with Applications, Lefkosa, Mersin 10, Turkey*, 37(9):6233–6239.

[22] Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai, Stanford University Stanford, CA.*, volume 14, pages 1137–1145. Montreal, Canada.

[23] Liaw, A., Wiener, M., et al. (2002). Classification and regression by randomforest. *R news*, 2(3):18–22.

[24] Lopes, R. G., Carvalho, R. N., Ladeira, M., and Carvalho, R. S. (2016). Predicting recovery of credit operations on a brazilian bank. In *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*, pages 780–784. IEEE.

[25] Mackiewicz, A. and Ratajczak, W. (1993). Principal components analysis (pca). *Computers and Geosciences, Department of Mathematics, Technical University of Poznań, Piotrowo 3a, Poznań Poland*, 19:303–342.

[26] Mowla, G. (2018). Default loans plague banking sector.

[27] Nova, A. (2018). More than 1 million people default on their student loans each year.

[28] of India, P. T. (2018). 9 million loan defaulters blacklisted in china; 27 billion dollar frozen.

[29] Oreski, S., Oreski, D., and Oreski, G. (2012). Hybrid system with genetic algorithm and artificial neural networks and its application to retail credit risk assessment. *Expert systems with applications, Bank of Karlovac, I.G.Kovacica 1, 47000 Karlovac, Croatia*, 39(16):12605–12617.

[30] S, S. M. and T, R. S. (2015). Loan credibility prediction system based on decision tree algorithm. *International Journal of Engineering Research and*, V4(09).

[31] Saar-Tsechansky, M. and Provost, F. (2007). Handling missing values when applying classification models. *Journal of machine learning research, New York, NY 10012, USA*, 8(Jul):1623–1657.

[32] Shlens, J. (2014). A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*.

[33] Stekhoven, D. J. and Bühlmann, P. (2011). Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics, Journal of Applied Artificial Intelligence*, 28(1):112–118.

[34] Tam, K. (1991). Neural network models and the prediction of bank bankruptcy. *Omega, University of Texas at Austin, USA*, 19(5):429 – 445.

[35] Thakur, A. (2018). India's wilful defaulters owe more than rs 1 lakh crore to banks - times of india .

[36] Twala, B. (2010). Multiple classifier application to credit risk assessment. *Expert Systems with Applications*, 37(4):3326–3336.

[37] Wang, Y., Wang, S., and Lai, K. K. (2005). A new fuzzy support vector machine to evaluate credit risk. *IEEE Transactions on Fuzzy Systems*, 13(6):820–831.

[38] Wikipedia contributors (2018). Lending club — Wikipedia, the free encyclopedia. [Online; accessed 26-November-2018].

[39] Yu, L., Wang, S., and Lai, K. K. (2008). Credit risk assessment with a multistage neural network ensemble learning approach. *Expert systems with applications*, 34(2):1434–1444.

[40] Yu, L., Yue, W., Wang, S., and Lai, K. K. (2010). Support vector machine based multiagent ensemble learning for credit risk evaluation. *Expert Systems with Applications, Chinese Academy of Sciences, Beijing 100190, China*, 37(2):1351–1360.