LOAN DEFAULTER DETECTION AND INTEREST RATE GENERATOR

Shivaprakash SJ
School of Computer Science and Engineering
Vellore Institute of Technology, Chennai
Chennai, India
shivaprakash.s2020@vitstudent.ac.in

Rohan Alroy.B
School of Computer Science and Engineering
Vellore Institute of Technology, Chennai
Chennai, India
rohanalroy.b2020@vitstudent.ac.in

Praveen Joe IR
School of Computer Science and Engineering
Vellore Institute of Technology, Chennai
Chennai, India
praveen.joe@vit.ac.in

I. Abstract

This project aims to develop a business intelligence tool that can accurately forecast loan defaults and identify factors contributing to defaults. The tool will analyze data such as income and employment history to spot trends and patterns that point to defaults. It can help lending institutions save money by making better decisions on whom to give a loan to and can also be used in the implementation of variable interest rates. However, concerns about data format, access to confidential data, and identifying factors leading to loan rejection may pose feasibility issues. The outcome of this project will be a useful asset for banks and other lending institutions to manage their lending portfolio more effectively and make more informed decisions.

Keywords – Business Intelligence, Machine Learning, Deep Learning, Dense Neural Network, loan defaulter probability, interest rate generator.

II. Introduction

In the world of finance, one of the biggest challenges that financial institutions face is loan defaults. When a borrower fails to repay their loan on time, has a waterfall effect on the overall economy. In the US, there is over $1.7 trillion of unpaid student loans, and the delinquency rate is around 11.1%. Similarly, the delinquency rate for mortgages in the US was 4.4% in the fourth quarter of 2021. Detecting loan defaults early is crucial for financial institutions to minimize their losses and prevent a financial crisis. One approach to detecting loan defaults is through machine learning-based loan defaulter detection models. These models analyze historical data and identify patterns and trends that can predict future defaults. In addition, financial institutions use dynamic interest rates to manage their risk and incentivize borrowers to repay their loans on time. The motivation to solve this problem is to help financial institutions reduce their risk and improve their lending practices, while also providing borrowers with fair and manageable loan terms.

III. Literature Survey

Koç et al. [1] explains how the authors developed a predictive model using logistic regression analysis to detect first payment defaults in a Turkish bank's consumer loan portfolio. They used significant variables such as age, income, and credit score to create a scoring system with 80% accuracy. The model can also be applied to other industries like insurance and e-commerce. The article provides valuable insights into how predictive modelling can be used to detect first payment defaults.

Eweoya et al. [2] discusses using SVM to predict loan fraud via machine learning. They used a dataset from Kaggle and various performance metrics to evaluate their model. Results show an 81.3% accuracy rate. The authors suggest SVM as a tool for financial institutions to prevent losses. However, our trials did not yield good results with SVM.

Ajah et al. [3] proposed a few strategies to deal with non performing loans. They examined the issues related to manually examining loans and proposed some IT based strategies. They proposed the use of neural networks to evaluate loan applications. They also proposed the use of artificial intelligence agents to detect loan defaults. These agents would communicate with each other and aid in the monitoring of transactions to potentially identify non performing loans.

Yotsawat et al. [4] proposed the use of a neural network based ensemble for credit scoring. This approach aims to deal with imbalanced data without the use of resampling techniques. They used a type of neural network called cost-sensitive neural network to build the ensemble. This approach trains different neural networks with different initial weights on imbalanced data. A majority voting based ensemble was created out of the neural networks.

Datkhile et al. [5] presents an efficient forecasting methodology for bankers to use to predict credit risk for loan applicants. Four different models - Naive Bayes, Random Forest Logistic Regression, and Decision Tree Algorithm- are used to obtain optimum results for credit risk analysis.

IV. Methodology

IV.i. Dataset Description and pre-processing

This dataset contains 15 features for 20,000 loan applications. Each loan has a primary key as ID, a loan grade, a self-reported annual income, field to denote if the applicant has been emplored for one year or less, their employment in years, type of home ownership, Debt-to-income ratio, purpose of the loan, and the term which is 36 months of 60 months. Other features include - whether the borrower had at least an event of delinquency, has at least 30 days of a bad rating, the revolving line utilization rate, the total late fees received to date, and the overdraft ratio. The target label is if the loan had been paid(0) or not(1).

In the pre-processing stage:

- we shall first drop the last_major_derog_none column due to it having a vast majority of null values.
- We shall also convert the term column to have only two types of 36 months and 60 months.
- Next we convert the categorial data in the columns grade, home_ownership, purpose and term to numerical data.
- Now we split the data to training and testing, and we normalize the training data using the MinMax scaler.

IV.ii. Exploratory Data Analysis:

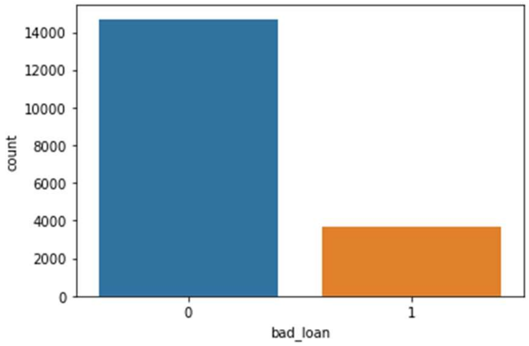


Figure 1: Count of samples with respect to bad_loan classification

We thus have approximately,14.5k samples corresponding to not being a loan defaulter and approximately, 3.8k samples for those who are loan defaulters. An approximate 80%:20% split up.

Given below are some more graphs, in order to illustrate how the different features are related.

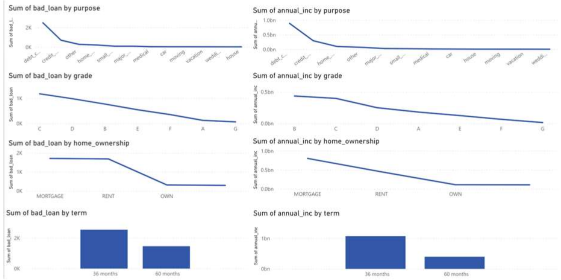


Figure 2: Relationships between different features

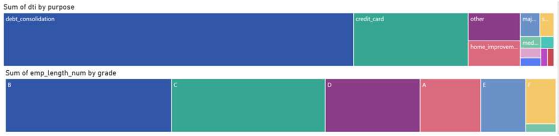
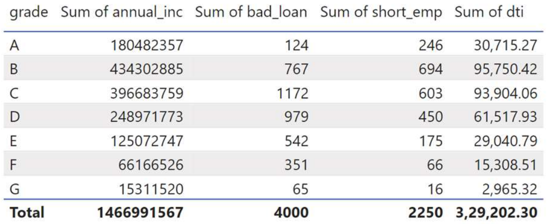


Figure 3: Division of dti and emp_length_num by purpose and grade respectively



grade	Sum of annual_inc	Sum of bad_loan	Sum of short_emp	Sum of dti
A	180482357	124	246	30,715.27
B	434302885	767	694	95,750.42
C	396683759	1172	603	93,904.06
D	248971773	979	450	61,517.93
E	125072747	542	175	29,040.79
F	66166526	351	66	15,308.51
G	15311520	65	16	2,965.32
Total	**1466991567**	**4000**	**2250**	**3,29,202.30**

Figure 4: Filtered dataset, to show details by grade

We will use the pipeline as follows:

Step-1: Get the input details, calculate credit grade and the credit score(if not given as input)

Step-2: Get the loan amount and term and then calculate the risk of loan defaulting, and using the values found prior, calculate an interest rate. Need of loan is also factored in.

V. Implementation

An examination of deep learning and machine learning models for predicting loan default is presented in this section. All the models in this paper were trained on an M1 GPU 3.2 GHz system with 8GB of RAM. Scikit-learn and Tensorflow libraries were used to create and train the models.

V.i. Decision Tree

This experiment explores the use of decision trees for the given problem. The decision tree classifier gave an accuracy of 70%. The classifier gave an ROC AUC score of 0.56. The performance shows that decision tree classifiers give good accuracy but have a pretty low ROC AUC score. The next experiment explores using a random forest classifier.

V.ii. Random Forest Classifier

Random forest classifier is an ensemble of decision trees. This experiment explores if a random forest classifier shall give better results than a decision tree classifier for the use case of predicting loan defaults. The model gave an accuracy of 79% and an ROC AUC score of 0.69. When compared to decision trees, it is observed that the random forest classifier has given significantly better results in terms of accuracy and ROC AUC score.

V.iii. Support Vector Machines

This section examines the performance of support vector machines (SVM) in the task of classifying loan defaults. Two models were trained, where one was trained with an rbf kernel, and the other used a polynomial kernel. The SVM with the rbf kernel gave an accuracy of 79% and an ROC AUC score of 0.60. The SVM with a polynomial kernel gave an accuracy of 79% and an ROC AUC score of 0.58. It is observed that both the models show similar performance, with the rbf kernel SVM having a higher ROC AUC score and the polynomial kernel SVM having a higher F1 score.

V.iv. KNN

This experiment examines the performance of the K-Nearest Neighbors algorithm. A KNN model with K = 150 was trained and gave an accuracy of 79% and an ROC AUC score of 0.68.

V.v. XGBoost

The XBBoost model was used to evaluate the performance of a boosting based ensemble. The model gave an accuracy of 78% and an ROC AUC score of 0.68.

V.vi. Deep Forest Classifier

The deep forest classifier is an algorithm that is more efficient than deep learning techniques but less efficient than ML based techniques. This model uses a deep network of forests to predict loan defaults. Deep forest gave an accuracy of 80% and an ROC AUC score of 0.70.

V.vii. Dense Neural Network

This experiment explores using a deep learning algorithm to predict loan defaults. This involved using three hidden layers with 512, 256, and 128 units respectively. The model was optimized using Adam optimizer with a learning rate of 0.0001.The model gave an accuracy of 79% and an ROC AUC score of 0.71. Figure 5 shows the accuracy of the model and Figure 6 shows the loss of the model.

V.viii. Finding the Interest rate based on given loan application

Algorithm:

Step-1: Get the input values from the bank clerk

Step-2: Calculate the credit grade, using

```
Credit_grade = 'A' if dti < 10 and
revol_util < 0.1 and
last_delinq_none == 1 else \

'B' if dti < 15 and revol_util < 0.2
and last_delinq_none == 1 else \
'C' if dti < 20 and revol_util < 0.3
and last_delinq_none == 1 else \
'D' if dti < 25 and revol_util < 0.4
else \
'E' if dti < 30 and revol_util < 0.5
else \
'F'
```

Step-3: Calculate an estimate credit score or get as input. If estimating using the conditions:

```
credit_score = 800 if credit_grade
== 'A' else \
               750 if credit_grade
== 'B' else \
               700 if credit_grade
== 'C' else \
               650 if credit_grade
== 'D' else \
               600 if credit_grade
== 'E' else \
               550 if credit_grade
== 'F' else \
500
```

Step-4: Convert inputs to np array and feed to model to get probability of loan defaulting.

Step-5: Get the input for loan amount needed and term of loan and calculate the interest rate, and monthly installment(only if risk is less than 85%):

```
if(prob_default>0.85):
  print("The risk is too high")
else:
  base_interest_rate = interest_rates[credit_gradel] + 0.05 * (credit_score - 700) + 0.1 * (prob_default - 0.05)
  if home_ownership == 1:
    base_interest_rate -= 0.05
  interest_rate = min(max(0.05, base_interest_rate), 0.35)
  # Calculate monthly payment based on interest rate and loan terms
  # Define loan terms
  loan = int(input("Enter the loan amount needed: "))
  term = int(input("Enter the loan term required, interest rate will be calculated accoridingly"))
  mon_interest_rate = interest_rate / 12
  num_payments = term * 12
  mon_payment = loan * mon_interest_rate / (1 - (1 + mon_interest_rate) ** -num_payments)
  # Output interest rate and monthly payment
  print(f"Interest rate: {interest_rate:.2%}")
  print(f"Monthly payment: ${mon_payment:.2f}")
  print("Probabilty_default:",prob_default)
```

VI. Evaluation Metrics

To evaluate how well the models predict loan defaults, metrics like accuracy and ROC AUC score are used. Additionally, the precision, recall, and f1 score are also noted. The mathematical representations of the metrics are given below in Equations 1 to 5.

$$\text{Accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}} \quad (1)$$

$$\text{Precision} = \frac{\text{tp}}{\text{tp} + \text{fp}} \quad (2)$$

$$\text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \quad (3)$$

$$\text{F1 Score} = \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

$$\text{ROC AUC} = \int \text{TPR}\, d(\text{FPR}) \quad (5)$$

Here, tp refers to true positives, fn refers to false negatives, tn refers to true negatives, fp refers to false positives, TPR refers to the true positive rate, and FPR refers to the false positive rate.

VII. Results

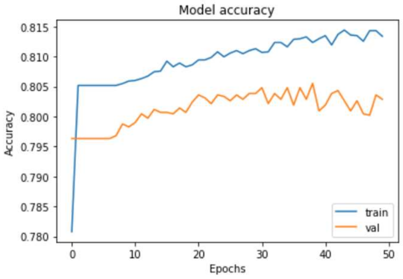


Figure 4: Training and Validation accuracy of Dense Neural Network

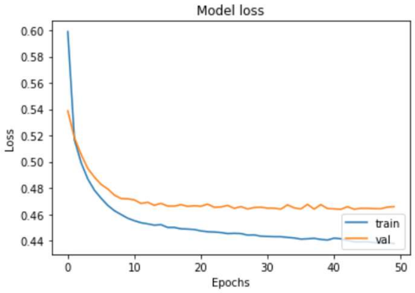


Figure 5: Training and Validation loss of Dense Neural Network

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC AUC**

Decision Tree	70	0.56	0.56	0.56	0.56
Random Forest	79	0.65	0.53	0.52	0.69
SVM rbf kernel	79	0.79	0.51	0.46	0.6
SVM polynomial kernel	79	0.79	0.51	0.47	0.58
KNN	79	0.4	0.5	0.44	0.68
XGBoost	78	0.62	0.54	0.53	0.68
Deep Forest	**80**	0.71	0.53	0.5	0.7
Dense Neural Network	79	0.68	0.53	0.5	**0.71**

Table 1: Summary of performance of models

From the experiments and Table 1, it is observed that Deep Forest gives the best accuracy and Dense Neural Networks gives the best ROC AUC score. In binary classification problems that have imbalanced data, the ROC AUC score is the preferred metric to evaluate a model's performance. Based on this, the best performing model is Dense Neural Network.

VIII. Conclusion and Framework

Through this paper, we have attempted multiple models, with the aim of identify the probability that a loan applicant could be a defaulter. With this we suggest the usage of Dense Neural Networks, as they give the best ROC-AUC score. We also have built an algorithm, which calculates a dynamic interest rate, based on the loan amount, term and the probability of the applicant being a defaulter.

In the future, we aim to take a detailed survey of different banks and their policies on providing dynamic interest rates, and to see if our model of predicting the interest rates, is suitable. We also aim to gather mode data and get more accuracy.

IX. References

[1]Koç, Utku, and Türkan Sevgili. "Consumer loans' first payment default detection: a predictive model." *Turkish Journal of Electrical Engineering and Computer Sciences* 28.1 (2020): 167-181.

[2]Eweoya, I. O., et al. "Fraud prediction in loan default using support vector machine." *Journal of Physics: Conference Series*. Vol. 1299. No. 1. IOP Publishing, 2019.

[3] Ajah, I. and Inyiama, C., 1970. Loan fraud detection and IT-based combat strategies. The Journal of Internet Banking and Commerce, 16(2), pp.1-13.

[4] Yotsawat, W., Wattuya, P. and Srivihok, A., 2021. A novel method for credit scoring based on cost-sensitive neural network ensemble. IEEE Access, 9, pp.78521-78537.

[5] Datkhile, Apurva, et al. "Statistical Modelling on Loan Default Prediction Using Different Models." IJRESM 3.3 (2020): 3-5.