Loan defaulter classification system

Mansi Lakhani (40197791), Apekshaba Gohil (40203058), Akshaya Barat Bushan (40220139), Anitha Ramakrishnan (40231724)

I. INTRODUCTION

Accurately assessing the creditworthiness of borrowers is crucial for financial institutions to mitigate the risk of defaults. Identifying potential defaulters with precision empowers lenders to make informed decisions, efficiently manage loan portfolios, and uphold the stability of their financial operations. The problem of predicting loan defaulters is crucial for many reasons including,

Risk Mitigation: Financial institutions can proactively manage and mitigate the risk of financial losses associated with unpaid loans.

Responsible Lending: Predicting defaulters allows lenders to ensure responsible lending practices by evaluating borrowers' creditworthiness and offering loans to those who are more likely to repay them.

A. CHALLENGES

The selected problem application posed several challenges,

1. High Dimension and Poor Data Quality: The dataset comprised over 39,000 records and 111 features, characterized by numerous missing and null values. Addressing this issue required robust data cleaning techniques, such as removing duplicate rows and discarding records with missing values, resulting in a refined dataset.
2. Feature Selection: An essential task was identifying relevant features that accurately captured the borrower's creditworthiness, while minimizing the inclusion of unnecessary or redundant variables.
3. Imbalanced Data: The dataset exhibited a significant class imbalance, with the "Fully paid" class representing 85% of the data and the "Charged off" class comprising only 15%. Balancing the data distribution was crucial to prevent biased model performance. Techniques such as oversampling the minority class or under sampling the majority class were employed to address this challenge.

By effectively addressing these challenges through meticulous data preprocessing, thoughtful feature selection, and appropriate handling of the imbalanced data, we aim to enhance the accuracy and reliability of our models.

II. METHODOLOGIES

A. DATA PREPROCESSING

Loan dataset from Kaggle consists of 111 features and 39,717 records making it high-dimensional data. The following is the overview of the data preprocessing steps:

1. Column Filtering and Selection: Columns with all NULL values were removed, resulting in the elimination of 54 columns. Additionally, irrelevant columns were dropped, reducing the dataset to 45 relevant features.
2. Handling Null Values: Rows with any missing values were removed from the dataset, resulting in the removal of 1,894 rows.
3. Target Variable Selection: The target variable is the "Loan Status," which initially had three classes: Fully Paid, Charged Off, and Current. The "Current" class was deemed irrelevant for the classification task and was therefore discarded. Consequently, the dataset shape was modified to (36,725, 45).
4. Outlier Detection and Removal: Outliers In the "Annual Income" column were identified using the Quantile method. Outliers exceeding the 95th percentile were removed from the dataset. However, no outliers were removed for columns such as Loan Amount, Interest Rate, and Total Payment, as they exhibited a mostly continuous distribution.
5. Feature Selection: Several features, including “id”, “application_type”, “policy_code”, “initial_list_status”, "installment", and "pymnt_plan" were determined to be unimportant for the goal of the problem and were discarded.
6. Data Splitting: The dataset was divided into training and testing sets, with 13,418 records allocated for training and 6,609 records for testing.

These preprocessing steps were conducted to prepare the loan dataset for subsequent analysis and modeling. The resulting dataset is now well-structured, with relevant features, and outliers removed, setting a solid foundation for further analysis and model development.

B. DECISION TREES

The decision tree architecture plays a crucial role in understanding the model's behavior and interpretability. Here is an overview of the decision tree's architecture and its performance metrics:

1. Depth: The decision tree was built with a default depth selection, taking into account the important features identified by the model through the calculation of feature entropy.
2. Splitting Criterion: The entropy was employed to determine the best feature to split the data at each decision node. By utilizing entropy, the decision tree identifies the most informative features that maximize the separation of the target classes. The “recoveries” feature was the most informative with the highest entropy which was selected as the root. Then features like “total_rec_prncp” (Principal received to date) and “funded_amont” and “last_credit_pull” were selected as they have better entropy than other features.
3. Evaluating Metrics: We calculated Accuracy, Recall, Precision and F1 score of the test data. We achieved high accuracy but Recall, Precision and F1 Score of one class (Loan Charged off) was very low. This indicated that the model struggles to classify one class because there is serious imbalance in the dataset.

C. Unsupervised decision trees

The preprocessed and balanced dataset after sampling was divided into labeled and unlabeled subsets. Specifically, 20%, was set aside as the labeled dataset, which was used for supervised learning. The remaining 80% of the data was considered as the unlabeled dataset. The labeled dataset was used to train a Decision Tree with a depth of 3 and entropy as the splitting criteria. The trained model was then applied to the unlabeled dataset to assign pseudo-labels to the unlabeled observations.

D. Deep Neural Networks

During the preprocessing stage, numerical features undergo scaling using methods like min-max scaling or standardization. This standardizes their values and prevents any single feature from dominating the learning process. Categorical features are transformed through techniques like one-hot encoding or entity embeddings, which represent them as binary vectors or continuous representations, respectively.

By following these preprocessing steps, the data is standardized in a format so that the neural network can process them properly and assign the correct weightage. The deep neural network architecture incorporates three hidden layers with ReLU activation functions, introducing non-linearity to the input and hidden layers. To address overfitting and enhance the model's ability to generalize, we introduced dropout regularization with appropriate dropout rates. For the output layer, which performs binary classification, we used a sigmoid activation function. To accommodate this classification task, we applied categorical cross-entropy loss function after transforming the true labels using one-hot encoding. Additionally, the Adam optimizer was used as it improves both accuracy and training speed in deep learning models.

Total 761,751 trainable parameters were there in the neural network architecture.

III. Attempts at solving problems

A. Over Sampling and under sampling

The initial class distribution of the target variable, Loan Status, revealed a significant imbalance, with 85.66% of instances classified as Fully Paid and only 14.34% as Charged Off. To address this issue, a combination of random under sampling and Synthetic Minority Over-sampling Technique for Nominal and Continuous (SMOTENC) was employed.

Random under sampling was applied to the majority class (Fully Paid), reducing its sample size. Subsequently, the dataset was divided into training and testing sets, with 13,418 records allocated for training and 6,609 records for testing.

To address the minority class imbalance, Synthetic Minority Over-sampling Technique for Nominal and Continuous (SMOTENC) was utilized, resulting in an augmented training dataset. This technique successfully increased the representation of the Charged Off class, achieving a more balanced class distribution in the training data, with Fully Paid accounting for 59.88% and Charged Off for 40.12%.

These sampling techniques and class balancing methods ensured a more equitable representation of both classes in the training data, laying the foundation for improved model performance and more reliable predictions.

B. Improved Decision Trees

Following the sampling technique, we constructed a decision tree with a depth of 3 and utilized entropy as the splitting criteria. Evaluating the model's performance yielded remarkable results. The accuracy, precision, and recall metrics all exceeded 97% on the train and test data.

	Before Sampling		After Sampling	
Classes/ Evaluation metric	Fully Paid	Charged off	Fully Paid	Charged off
Precision	94 %	92 %	97 %	100 %
Recall	99 %	69 %	100 %	89 %
F1 score	96 %	79 %	98 %	94 %
Accuracy	94 %		97 %	

Table 1: The table shows various evaluation metric on test set before and after sampling the dataset.

This outcome signifies the decision tree's proficiency in accurately classifying instances from both classes, including the minority class. The high precision indicates minimal false positive predictions, while the high recall highlights the model's capacity to identify a substantial number of instances from the minority class. These exceptional performance metrics underscore the effectiveness of the decision tree in effectively addressing the classification task at hand.

IV. Future Improvements

1. In decision trees, we aim to optimize performance by experimenting with various hyperparameters such as max depth, max features and min samples leaf.
2. Our objective is to improve the evaluation metrics scores of the neural networks. To achieve this, we will experiment with various optimization techniques and hyperparameters, ensuring a balance that enhances accuracy without encountering overfitting or underfitting issues. Additionally, to address the underfitting problem observed in the model, we will enhance the neural network architecture by adding more hidden layers with a fully connected architecture.
3. In unsupervised decision trees, we will iteratively add the top 10% of high-confidence pseudo labels to the training data, repeating the process until all training data has been utilized. Additionally, we will investigate different confidence thresholds for selecting high-confidence pseudo-labels.
4. Finally, we want to compare and evaluate different models. By exploring and comparing multiple modeling techniques, such as decision trees, neural networks, and unsupervised decision trees, we can gain valuable insights into their strengths and weaknesses. This comparative analysis will allow us to select the most suitable model for the given problem, further improving the accuracy, precision, and recall of our predictions. Additionally, by benchmarking different models against each other, we can identify areas for refinement and develop an optimized model that maximizes performance on our dataset.

V. References

[1] Dataset: “Loan Classification dataset” from Kaggle. https://www.kaggle.com/datasets/abhishek14398/loan-dataset

[2] Gupta, Kanishk and Chakrabarti, Binayak and Ansari, Aseer Ahmad and Rautaray, Siddharth Swarup and Pandey, Manjusha, Loanification - Loan Approval Classification using Machine Learning Algorithms (April 24, 2021). Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2021

[3] Rokach, L., Maimon, O. (2005). Decision Trees. In: Maimon, O., Rokach, L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA.

[4] Github: https://github.com/Apekshaba/COMP_6721/

[5] Decision tree classifier: https://scikitlearn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[6] Neural network architecture: https://keras.io/api/layers/

[7] Feature importance: https://machinelearningmastery.com/calculate-feature-importance-with-python/

Supplementary:

A. Feature Importance Graph

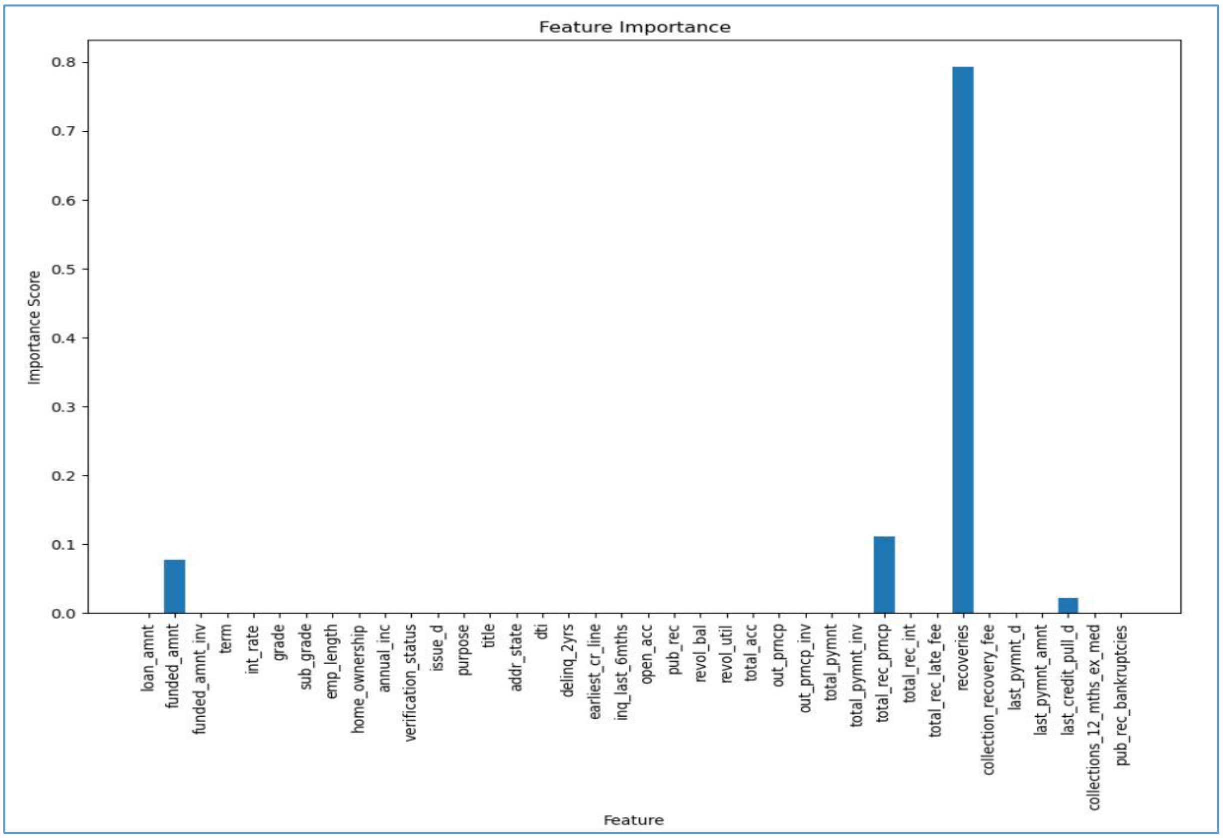

Figure 1: Feature Importance graph of the dataset

B. Decision Tree

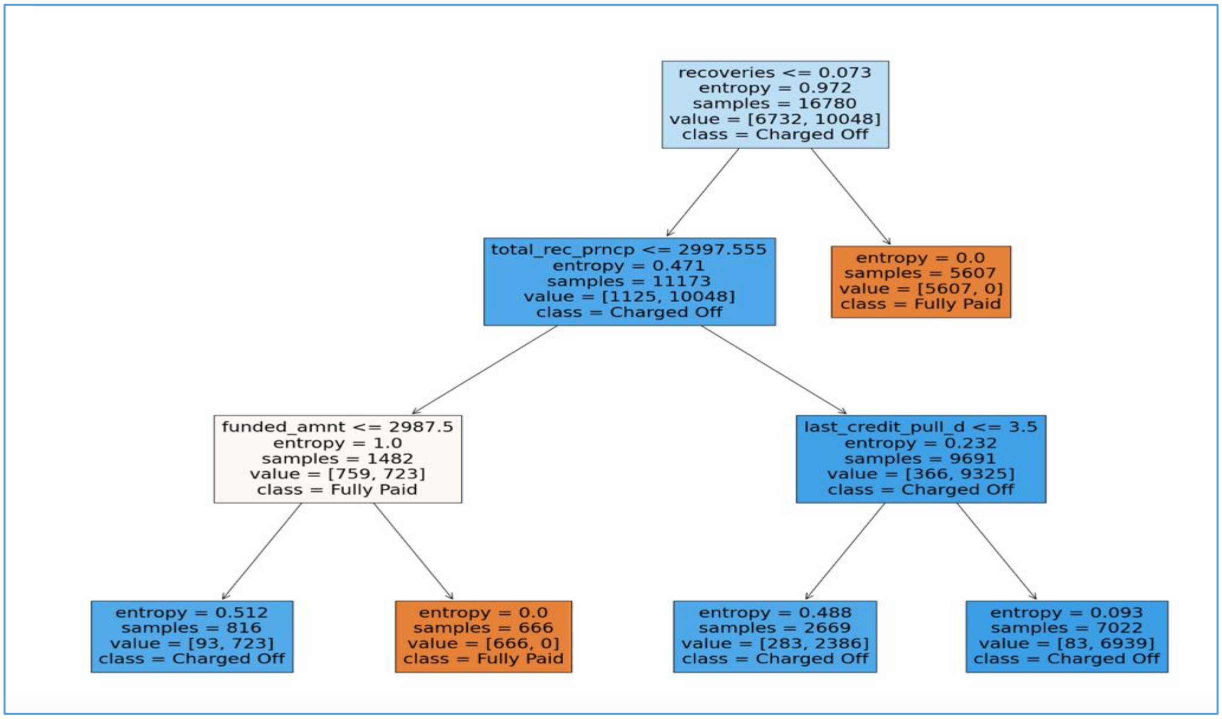

Figure 2: Final Decision Tree after Sampling