Credit Risk Modeling Using Machine Learning

Rituj Upadhyay (2020570), Hardik Sachdeva (2022193), Mananya Kohli (2022275), Akshat Tokas (2022056)
IIIT Delhi
Email: {rituj, hardik, mananya, akshat
@iiitd.ac.in

***Abstract*—Credit risk assessment is essential in the financial industry, particularly in today's economic landscape, where institutions are facing increasing non-performing loans (NPLs) and defaults. Traditional credit risk models struggle to capture the complexities of evolving borrower behavior. This project leverages machine learning (ML) techniques to improve credit risk prediction. The goal is to build a flexible model that can better predict defaults, mitigate financial losses, and offer insights to refine lending strategies.**

GitHub Link at: https://github.com/Sachdeva-hardik/Ml_final_project.

I. Introduction: Problem Statement

In today's rapidly evolving financial landscape, credit risk assessment is more crucial than ever. With increasing levels of consumer debt and economic uncertainty due to global events like the COVID-19 pandemic, traditional credit risk models are struggling to adapt to new dynamics. Machine learning (ML) offers a solution by providing more flexible and accurate models that can quickly respond to changes in borrower behavior and economic conditions.

A. Current Situation

The financial sector is facing unprecedented challenges due to economic fluctuations, increasing non-performing loans (NPLs), and a rise in defaults. Financial institutions are under pressure to improve their credit risk management practices to mitigate potential losses and maintain profitability.

B. Impact of Machine Learning

ML models can analyze vast amounts of data in real-time, uncovering patterns and correlations that traditional models might miss. By integrating ML, financial institutions can not only predict defaults more accurately but also offer more personalized lending products, reduce the risk of financial crises, and enhance overall financial stability. Additionally, ML can help in reducing biases in credit risk assessments, leading to fairer lending practices and expanding access to credit for underserved populations.

With the COVID-19 pandemic exacerbating financial uncertainty, the limitations of traditional credit risk models—often relying on historical data and static rules—have become more apparent. This project aims to develop a machine learning-based model that improves the prediction of credit risk, offering insights into lending decisions and improving financial outcomes.

II. Literature Survey

A. Predicting Credit Risk for Unsecured Lending

The paper titled ”Predicting Credit Risk for Unsecured Lending: A Machine Learning Approach” investigates the creation of a machine learning (ML)-based credit scoring model aimed at predicting defaults in unsecured lending, specifically credit card lending. Due to the imbalance in the datasets, where the majority of customers do not default, the authors employ SMOTE (Synthetic Minority Oversampling Technique) to balance the data.

After evaluating seven different ML models, the Light Gradient Boosting Machine (LGBM) classifier was determined to be the most effective for handling large datasets and providing greater accuracy. The research underscores that utilizing such models can enhance the prediction of credit defaults, allowing financial institutions to manage credit risk more efficiently.

The study highlights the increasing demand for credit cards, which has led to a rise in defaults, making it necessary to use modern and robust methods for assessing credit risk.

B. Machine Learning-Driven Credit Risk: A Systemic Review

The paper explores the significant impact that machine learning and artificial intelligence have had across various sectors, with a particular focus on finance and credit risk estimation. Machine learning has become crucial for predicting credit risk, a key factor in determining the probability of a debtor defaulting. The accurate estimation of credit risk is vital for maintaining financial stability, as misjudgments can lead to systemic crises like the 2008 subprime mortgage collapse.

Historically, statistical methods such as Linear Discriminant Analysis and Logistic Regression have been used for credit risk estimation, but they struggle with handling large datasets. Advances in computing power and the availability of big data have paved the way for more efficient AI-driven approaches. Modern machine learning techniques, including k-Nearest Neighbor, Random Forest, Support Vector Machines, and particularly deep learning, have proven to be more flexible and accurate than traditional statistical models.

III. Dataset: Data Preprocessing Techniques

A. Different Attributes and Visualizations

The dataset consists of multiple numerical and categorical attributes. Numerical attributes include loan data such as ”Total_TL”, ”Tot_Closed_TL”, and ”NetMonthlyIncome”. Visualizations like histograms, box plots, and correlation matrices were used for analysis. Categorical attributes, such as ”Marital Status” and ”Education”, were visualized using bar charts.

B. Preprocessing Details

Several preprocessing steps were essential before proceeding with the modeling phase:

- **Loading and Cleaning Datasets:** Two datasets were loaded from Excel files. Invalid values represented by -99999 were identified and removed. Columns with a high number of invalid entries (greater than 10,000) were dropped to avoid introducing noise into the analysis.
- **Merging Datasets:** The datasets were merged using an inner join on the *PROSPECTID* column. This ensured that only matching records were retained, effectively eliminating any null values in the final merged dataset.
- **Categorical Feature Analysis:** Chi-Square tests were performed on categorical features such as *MARITALSTATUS*, *EDUCATION*, *GENDER*, *last_prod_enq2*, and *first_prod_enq2*. Only categorical features with a significant relationship to the target variable (*Approved_Flag*) were retained (p-value 0.05).
- **Multicollinearity Check on Numerical Features:** Variance Inflation Factor (VIF) was calculated to detect multicollinearity in numerical features. Features with a VIF ¿ 6 were sequentially removed to avoid redundancy and improve model performance. After filtering, the remaining numerical features were validated using ANOVA to ensure their relevance to the target variable.
- **Encoding Categorical Variables:** Ordinal encoding was applied to the *EDUCATION* feature, following the specified hierarchy:
  - *SSC: 1, 12TH: 2, GRADUATE: 3, UNDERGRADUATE: 3, PROFESSIONAL: 3, POST-GRADUATE: 4, OTHERS: 1.*
  - Business input may be needed to verify the grouping of “OTHERS.”

  One-hot encoding was applied to other categorical features such as *MARITALSTATUS*, *GENDER*, *last_prod_enq2*, and *first_prod_enq2*.
- **Feature Selection:** Statistical techniques such as Chi-Square tests were used to select the most relevant categorical features, and VIF was used for numerical features.
- **Outlier Detection and Removal:** Outliers were handled using the IQR method or other statistical techniques to prevent skewed predictions.
- **Final Dataset Preparation:** The cleaned and transformed dataset included both encoded categorical and validated numerical features. This version of the dataset ensures optimal feature representation for future model experimentation and training.

This structured process guarantees a robust feature set with minimal multicollinearity and maximum relevance to the target variable, setting a solid foundation for model building.

C. Exploratory Data Analysis (EDA)

Exploratory data analysis was performed to understand the structure of the data, detect missing values, and identify relationships between features.It refers to the process of analyzing and visualizing data to grasp its main features, uncover trends, detect outliers, and recognize relationships between variables. This method involves exploring datasets to understand their key characteristics and identify patterns.

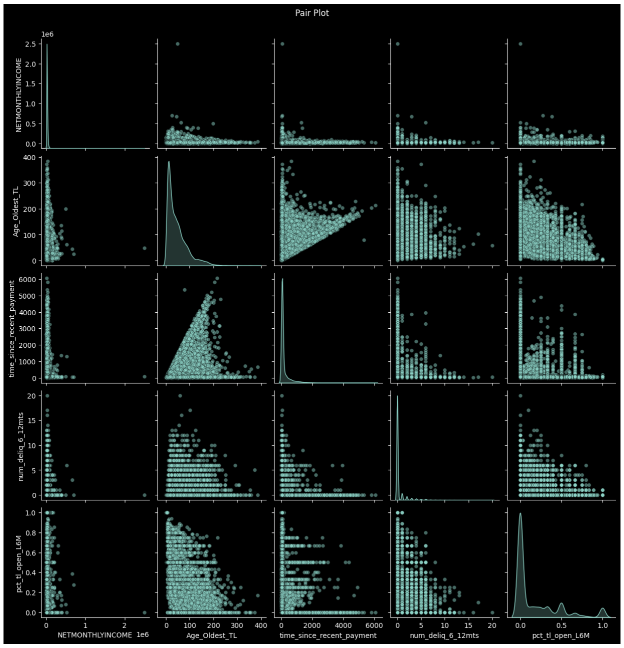

Fig. 1. Pair Plot

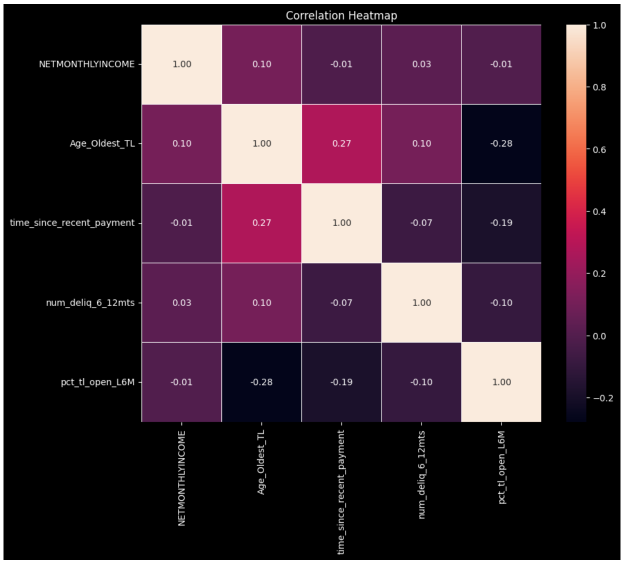

Fig. 2. Correlation Heatmap

IV. Methodology and Model Details

The goal of this phase is to evaluate the feasibility of various machine learning models for predicting the *Approved_Flag*.

The models considered in this analysis are:

A. Random Forest

The Random Forest algorithm is an ensemble machine learning technique used for both classification and regression tasks. It builds multiple decision trees during training, where each tree is trained on random subsets of the data and features. The randomness introduced by using different subsets reduces overfitting and increases model robustness. For classification, it aggregates the results by voting across trees, and for regression, it averages the predictions. This collective decision-making from multiple trees makes Random Forest highly accurate, capable of handling complex data, and resilient to noise in the dataset. Random Forest is considered because it can handle imbalanced datasets, reduces overfitting through bagging, and provides feature importance.

B. XGBoost

XGBoost, short for "Extreme Gradient Boosting," is a highly efficient and scalable machine learning algorithm designed for both classification and regression tasks. It is an ensemble method that combines multiple weak models to produce a more accurate and robust prediction. XGBoost stands out for its ability to handle large datasets and missing values efficiently, without requiring extensive pre-processing. With built-in support for parallel processing, XGBoost accelerates training on large datasets, making it a popular choice for achieving state-of-the-art performance in various machine learning tasks. XGBoost is explored due to its high performance with tabular data and ability to handle multi-class classification with fine-tuning options.

C. Decision Tree

A decision tree is a supervised learning algorithm used in machine learning for both classification and regression tasks. It models decisions through a tree-like structure, where each internal node represents a test on an attribute, each branch corresponds to the outcome of that test, and each leaf node signifies a final decision or prediction. Decision trees split data into subsets based on feature values, making decisions at each step until a final outcome is reached. Known for being intuitive and interpretable, decision trees are widely applied in various domains to model complex decision-making processes.

D. Scaling

Scaling is the process of adjusting the values of features in your data so that they all fall within a similar range. This is important because some machine learning models can be affected by differences in the scale of features. For example, if one feature ranges from 1 to 1000 and another from 0 to 1, the model might give more importance to the feature with larger values, even though both features might be equally important. Scaling makes sure all features contribute fairly to the model's predictions.

StandardScaler is a popular way to scale data. It works by subtracting the average (mean) value of a feature and then dividing by how spread out the values are (standard deviation). This ensures that the data for each feature has a mean of 0 and a standard deviation of 1.

E. Hyperparameter Tuning and Grid Search

Hyperparameter tuning means changing the settings of a model to make it work better. Instead of using random or default settings, you try out different values to see which ones give the best results. There are different ways to tune a model, like using Grid Search (all possible combinations), Random Search (random combinations), or even smarter methods like Bayesian Optimization that learn from past tests to find better settings faster.

Grid Search is a way to find the best settings for a machine learning model. Models have hyperparameters that control how they learn, like how fast they learn or how complex the model should be. Grid search tries all possible combinations of these settings to see which one works best.

V. Results and Analysis

A. Dataset

The EDA indicated strong predictive power in certain features but revealed imbalances in the dataset. The analysis reveals several imbalances and patterns in the dataset. The *Approved_Flag* is skewed toward P2, suggesting a higher likelihood of this category being approved compared to P1, P3, and P4. In terms of demographics, *MARITALSTATUS_Married* and *GENDER_F* show significant imbalances, with 73.4% of individuals being married and only 11.2% being female, indicating the dataset is predominantly male and married. Loan ownership data, especially for home and personal loans, show that most individuals don't have these loans, as reflected in *HL_Flag* and *PL_Flag*. The *NETMONTHLYINCOME* distribution is heavily skewed, pointing to a middle-income segment. The *Age_Oldest_TL* suggests most individuals have a relatively young credit history, while delinquency-related variables such as *recent_level_of_deliq* indicate low recent delinquency. Overall, the dataset shows notable imbalances across different features, which could impact predictive modeling.

B. Model Comparison

Model	**Accuracy**
Decision Trees	0.71
XGBoost	0.78
Random Forest	0.76

TABLE I
Accuracy of Different Models

Based on the metrics provided, XGBoost stands out as the best model overall with highest accuracy (0.78). It delivers strong performance in key classes like p1 and p2, with high Precision, Recall, and F1 Scores. In particular, for p2, it achieves the highest Recall (0.9136) and F1 Score (0.8673) which indicates that it can effectively capture important patterns. While p3 remains a challenging class across all models, XGBoost marginally outperforms the others with a Recall of 0.3094, showing some ability to identify this class. Random Forest is a close competitor but falls slightly short in the critical classes p1 and p2. Decision Tree, on the other hand, struggles with lower F1 Scores across these crucial categories. Overall, XGBoost offers a more balanced and reliable performance, making it the most suitable choice for this task.

C. Scaling

Applying feature scaling using StandardScaler did not enhance the performance of the XGBoost model, which is not surprising given the nature of tree-based algorithms. Unlike models like Logistic Regression, SVMs, or KNNs, which rely on the magnitude and distribution of features, XGBoost splits data based on feature thresholds. This means the model can handle features with different ranges by itself, so you don't need to change them most of the time. It shows that tree-based models are good at dealing with different kinds of data. So, it might be better to focus on things like picking the right features, improving them, or changing the model settings to make it better.

D. Fine Tuning

The best parameters for the XGBoost model were found to be 'learningrate': 0.2, 'maxdepth': 3, 'nestimators': 200. A learning rate of 0.2 helps the model learn faster by making bigger adjustments during training. The maximum depth of 3 keeps the trees simple, which helps the model avoid overfitting and focus on general patterns in the data. Using 200 estimators provides enough trees to improve accuracy without making the model overly complex. This combination resulted in a test accuracy of 0.78, showing a good balance between learning speed, simplicity, and performance.

A grid search is also utilized to find the best hyperparameter values for an XGBoost model. It tests different combinations of hyperparameters, such as colsamplebytree, learningrate, maxdepth, alpha, and nestimators, and evaluates the model's performance on both the training and test datasets. It stores the results for each combination, including the train and test accuracies. After completing the grid search, the code identifies and prints the best hyperparameter combination that achieved the highest test accuracy, helping to fine-tune the model for optimal performance.

For the hyperparameters colsamplebytree: 0.9, learningrate: 0.1, maxdepth: 3, alpha: 10, nestimators: 100, The hyperparameter tuning results show that the model has a training accuracy of 81 and a test accuracy of 78. The small difference between the two suggests that the model is doing well but might be slightly overfitting. This means the model is performing well on the training data

VI. Conclusion

A. Learnings from the Project

- XGBoost was the best model for predicting the Approved Flag because it had the highest accuracy and worked well for important classes like p1 and p2.
- Tree-based models, like Random Forest and XGBoost, use thresholds to make decisions instead of the size of the features, so scaling the data is not always needed.
- Preprocessing steps, like fixing missing values and balancing the data, helped improve how well the models worked.
- Feature selection methods, like ANOVA and Chi-square tests, helped pick the most useful features, which made the predictions better. This shows how important it is to preprocess data by fixing imbalances and choosing the right features.
- Different models, like Random Forest, XGBoost, and Decision Trees, were tested to see how well they work with complicated data. The goal was to make better lending decisions and improve financial plans in a changing economy.
- Random Forest and Decision Trees are simpler and easier to understand, but XGBoost gave better results overall, which shows simpler models are easier to explain but may not work as well for hard problems, while more complex models like XGBoost can give better results.
- The study also highlighted using different metrics, like accuracy, precision, recall, and F1 score, to check how good the models are, especially when the data is imbalanced.

B. Challenges Faced

- **Data Imbalance:** The dataset was unbalanced, meaning some categories, like the Approved Flag and certain demographic groups, had way more examples than others. This made it hard for the model to pay equal attention to all groups, especially the smaller ones, and affected how well it worked for those underrepresented categories.
- **Overfitting:** A big challenge was stopping the model from overfitting. Overfitting happens when the model learns the training data too perfectly, including tiny details or random patterns, which makes it bad at handling new data. Finding the right settings during hyperparameter tuning was important to make sure the model worked well on both training and test data.
- **Hyperparameter Tuning:** Choosing the best settings for the model, like learning rate and maximum tree depth, took a lot of time. There were so many combinations to try during grid search, and testing them all required a lot of computer power to find the best ones.

C. Contribution of Each Team Member

- **Mananya Kohli & Akshat Tokas:** Handled data collection and preprocessing.
- **Hardik Sachdeva & Rituj Upadhyay:** Led the exploratory data analysis and feature engineering.
- **Akshat Tokas & Rituj Upadhyay:** Focused on model development and evaluation.
- **Mananya Kohli & Hardik Sachdeva:** Will handle model interpretation and deployment.

References

1) Predicting Credit Risk for Unsecured Lending: A Machine Learning Approach
2) Machine Learning-Driven Credit Risk: A Systematic Review