Loan Default Prediction

Project Report

Neha Patil

patil.neha1@northeastern.edu

Problem Setting:

Should a customer be given a loan? This critical question for the financial industry serves as the motivation for the project. It aims to mitigate losses from loan defaults. The focus of this project is on risk management and loan issuance. The defaulters not only impact the lending institutions but also the economy if the defaults occur in significantly higher numbers leading to liquidity crises. A machine learning model can help detect the default customer early, helping avoid major losses. Challenges include finding the right datasets, dealing with inconsistent data, and addressing imbalanced data.

Problem Definition:

The problem is to predict the loan default risk by analysing borrower and loan characteristics. To achieve this, the following questions should be answered:

- What features will help us predict loan default?
- Before issuing a loan, can we correctly classify borrowers into 'likely to default' and 'unlikely to default' categories?

Data Sources:

The dataset of the Loan Default Prediction is from Kaggle. The data source provides details on loan characteristics and borrowers.

Data Description:

The Lending Club dataset has 396,030 entries and 27 features, including both numerical and categorical data points essential for evaluating loan default risks.

Key numerical variables are:

loan_amnt: Loan amount requested by the borrower.

annual_inc: Borrower's self-reported yearly income.

dti: Debt-to-income ratio calculated from monthly debt payments on total debt (excluding mortgage and requested loan) divided by monthly income.

mort_acc: Number of mortgage accounts.

total_acc: Total credit lines in the borrower's credit file.

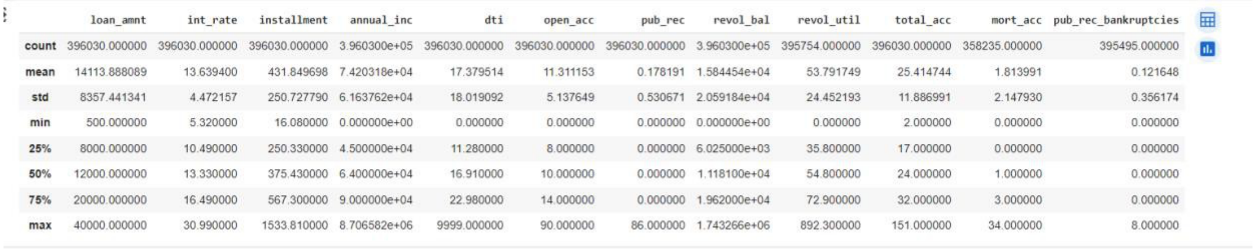
Important categorical features include:

sub_grade: Where A1 is the best grade and G5 is the worst.

home_ownership: Borrower's housing status (rent, own, mortgage).
purpose: Purpose of the loan.
application_type: Specifies if the loan is an individual or joint application.

Data Exploration:

While exploring the dataset, first we applied descriptive statistics. This has helped us summarize the central tendency, dispersion, and shape of the dataset's distribution.



	loan_amnt	int_rate	installment	annual_inc	dti	open_acc	pub_rec	revol_bal	revol_util	total_acc	mort_acc	pub_rec_bankruptcies
count	396030.000000	396030.000000	396030.000000	3.960300e+05	396030.000000	396030.000000	396030.000000	3.960300e+05	395754.000000	396030.000000	358235.000000	395495.000000
mean	14113.888089	13.639400	431.849698	7.420318e+04	17.379514	11.311153	0.178191	1.584454e+04	53.791749	25.414744	1.813991	0.121648
std	8357.441341	4.472157	250.727790	6.163762e+04	18.019092	5.137649	0.530671	2.059184e+04	24.452193	11.886991	2.147930	0.356174
min	500.000000	5.320000	16.080000	0.000000e+00	0.000000	0.000000	0.000000	0.000000e+00	0.000000	2.000000	0.000000	0.000000
25%	8000.000000	10.490000	250.330000	4.500000e+04	11.280000	8.000000	0.000000	6.025000e+03	35.800000	17.000000	0.000000	0.000000
50%	12000.000000	13.330000	375.430000	6.400000e+04	16.910000	10.000000	0.000000	1.118100e+04	54.800000	24.000000	1.000000	0.000000
75%	20000.000000	16.490000	567.300000	9.000000e+04	22.980000	14.000000	0.000000	1.962000e+04	72.900000	32.000000	3.000000	0.000000
max	40000.000000	30.990000	1533.810000	8.706582e+06	9999.000000	90.000000	86.000000	1.743266e+06	892.300000	151.000000	34.000000	8.000000

Fig.1 Statistics (numerical variables)

<u>Key insights from our statistical exploration include:</u>

- Loan Amount: The average loan amount requested by borrowers is approximately $14,113, with a standard deviation of $8,357, indicating variability in the loan amounts. Loans range from $500 to $40,000.
- Interest Rate: Interest rate is around 13.64%, but it varies significantly, as shown by the standard deviation of 4.47%. The rates span from 5.32% to 30.99%.
- Annual Income: There is considerable diversity in borrowers' annual incomes, with an average of about $74,230 but a wide range — the maximum reported income is over $8 million, suggesting the presence of outliers that could be influential in predictive modelling.
- Debt-to-Income Ratio: The average DTI is 17.38, with the 50th percentile at 16.91. This indicates most borrowers have a manageable level of debt compared to their income.
- Total Accounts: On average, borrowers have around 25 credit lines in their credit file, but this can go up to 151, suggesting the presence of outliers.
- Mortgage Accounts: There is an average of roughly 1.8 mortgage accounts per borrower, with some having up to 34, pointing to a segment of borrowers with extensive real estate holdings.

Key insights from our visualization exploration include:

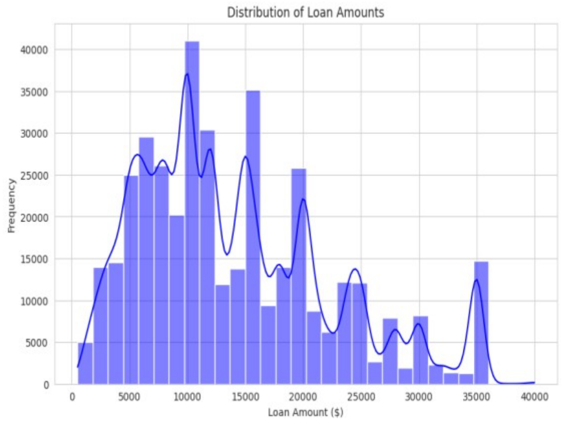


Fig. 2 Histogram - Distribution of Loan Amounts

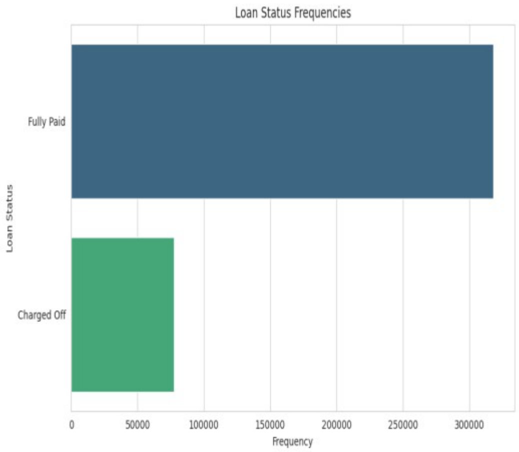


Fig. 3 Count Plot - Loan Status Frequencies

The histogram reveals insights into the distribution of loan amounts::

- Mode and Distribution: The most common loan amount appears to be around $10,000.
- Range and Spread: Loan amounts vary widely, from small amounts up to approximately $40,000, with a right-skewed distribution.
- Tendency towards Round Numbers: Borrowers tend to prefer round figures like $5,000, $10,000, and $15,000 for their loan amounts.

The Count Plot reveals insights into the loan status frequencies, below are the key findings: "Fully Paid" and "Charged Off" are the predominant states of loans in this dataset, highlighting two key statuses. Most loans are 'Fully Paid', indicating successful repayments. Charged Off' loans, while less common, are crucial indicators of defaults. The notable difference between

'Fully Paid' and 'Charged Off' frequencies raises class imbalance concerns

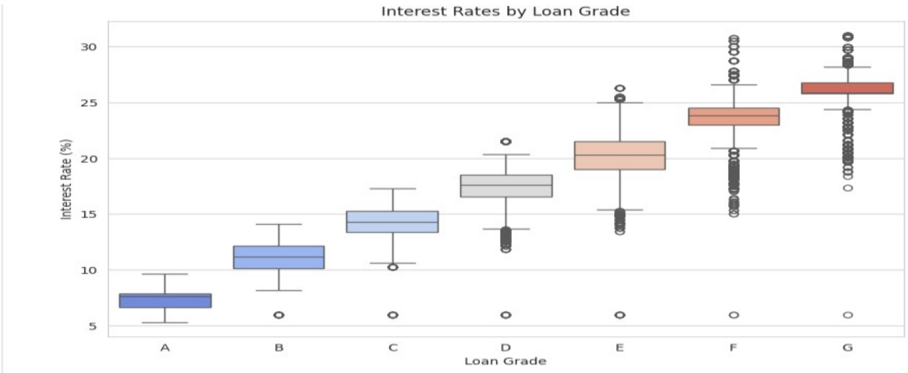


Fig. 4 Box Plot – Interest Rate by Loan Grade

The boxplot shows how interest rates change with different grades:

- Distribution and Variation: Higher grade loans (like A and B) have more consistent interest rates, while lower grade loans (like F and G) have more varied rates, reflecting higher risk.
- Outliers: Certain grades (like C to G) have outliers, indicating significant interest rate differences within the same grade. This suggests varying borrower risks within a grade.

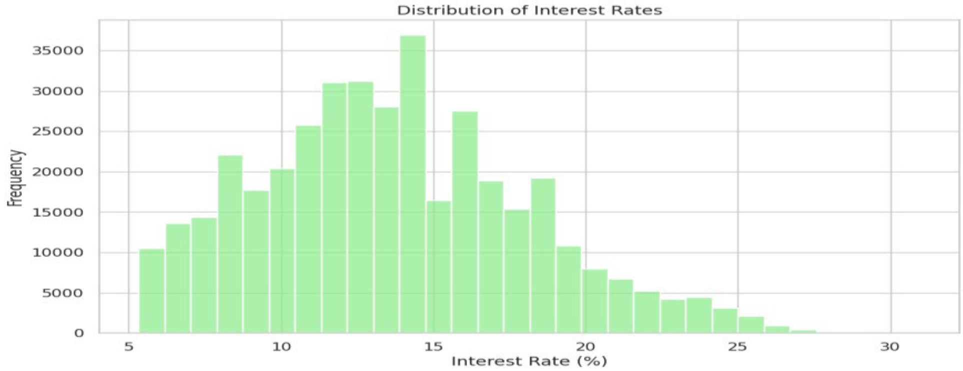
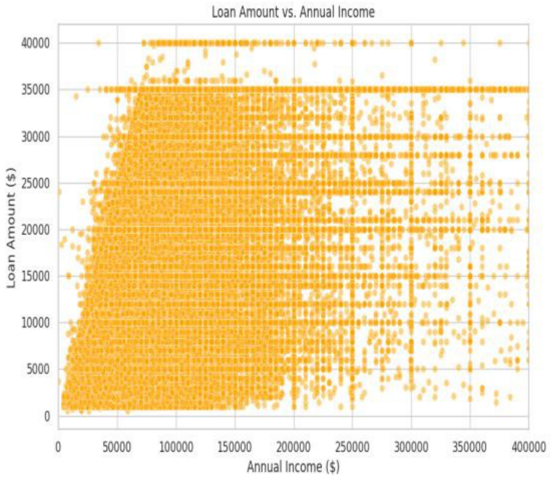


Fig. 5 Histogram – Distribution of Interest Rate

The histogram shows how interest rates are distributed, below are the key findings:

- Central Tendency and Modality: Interest rates cluster around a central value, showing a common rate for many loans.
- Skewness: The distribution skews right, meaning most loans have moderate rates, while fewer have very high rates, often for higher-risk borrowers.



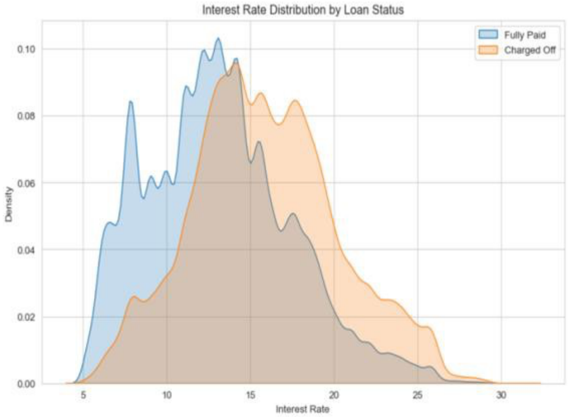


Fig. 6 Scatter Plot – Loan Amount vs Annual Income. Fig.7 KDE Plot Interest Rate Distribution

The scatter plot shows loan amount versus annual income, below are the key findings:

- Positive Correlation: Generally, higher annual incomes correspond to larger approved loans, showing lenders prefer granting bigger loans to higher earners.
- Income Distribution and Loan Caps: Most data points cluster at lower incomes, below $100,000. Loans for these incomes are usually under $40,000, suggesting a cautious lending approach or capped loan amounts in these brackets.
- Variability at Higher Incomes: Above $100,000 income, there's more variability in loan amounts.

The Kernel Density Estimates plot shows how interest rates are distributed:

- 1. "Charged Off" loans typically feature higher interest rates, hinting at a higher risk of non-repayment.
- 2. "Fully Paid" loans mostly cluster at lower interest rates, while "Charged Off" loans are more common at higher rates.
- 3. Overlapping interest rates for both statuses suggest factors beyond interest rate influence repayment.

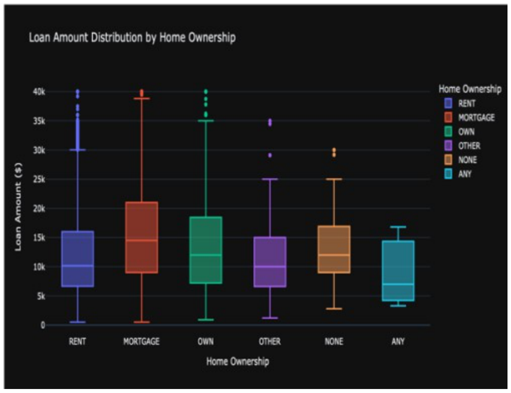


Fig. 8 Box Plot – Loan Amount Distribution by Home Ownership

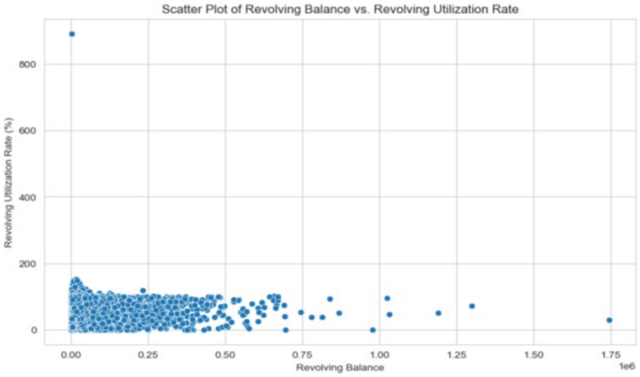


Fig. 9 Scatter Plot – Revolving Balance vs. Revolving Utilization Rate

The Box plot shows how interest rates are distributed, below are the key findings:

- Borrowers with mortgages have the highest median loan amounts, while those without any home ownership have the lowest.
- Loan amount outliers are mostly seen in the "MORTGAGE" and "OWN" categories, indicating occasional much higher loans.

The Scatter plot shows how interest rates are distributed, below are the key findings:

- Concentration at Lower Balances: A dense cluster of points at lower revolving balances suggests that most borrowers have lower amounts of revolving credit.
- Utilization Rates Variability: For lower revolving balances, there's a wide range of utilization rates, indicating diverse usage of credit relative to the credit limit.
- Outliers in Utilization: There are a few outliers with extremely high revolving utilization rates, some exceeding 100%, which could indicate overutilization or reporting anomalies.
- Decrease in Utilization with Higher Balances: Higher revolving balances are generally associated with lower utilization rates, suggesting larger credit limits or more credit use.

Data Mining Tasks:

The following have been performed as part of data mining tasks:

1. Data Transformation:

Changing Data Types: To enhance model compatibility and performance:

- Converted 'term' from object type to numeric to facilitate numerical operations.
- Transformed 'issue_d' and 'earliest_cr_line' from object type to datetime to accurately handle temporal calculations.
- Modified 'earliest_cr_line' from a date to the number of days since the earliest credit line, providing a more quantifiable measure of credit history.
- Changed data types of categorical columns such as 'grade', 'sub_grade', 'home_ownership', 'verification_status', 'purpose', 'application_type', 'initial_list_status', 'term', and 'emp_length' to category to optimize memory usage and improve processing speed.
- Normalization of 'emp_length': Transformed 'emp_length' from a categorical description (e.g., '9 years') to a numeric value, standardizing this variable for analytical purposes.

2. Checking for Duplicates:

Duplicate Records: Ensured data integrity by verifying and confirming that no duplicate records are present in the dataset.

3. Missing Data Imputation:

Utilized various techniques to handle missing values, crucial for dataset's robustness:

- Imputed 'revol_util' and 'pub_rec_bankruptcies' with the mode to preserve common trends.
- Replaced missing values in 'emp_title' and 'title' with 'Unknown', maintaining consistency in categorical data.
- For 'mort_acc', missing values were imputed based on the mode of 'total_acc', leveraging correlations to make informed substitutions.
- Set missing 'emp_length' entries to '< 1 year', simplifying the representation of short-term employment.

4. Data Reduction:

Dropping Irrelevant Columns: Removed non-contributory columns such as 'emp_title', 'title', 'address', and 'issue_d' to focus the model on relevant predictors, enhancing prediction accuracy.

5. Removing Outliers:

Identified and removed outliers in variables such as 'dti', 'annual_inc', 'open_acc', 'total_acc', 'revol_util', 'revol_bal', and 'mort_acc'. These values, representing a minor fraction of the dataset, are deemed unlikely to enhance prediction accuracy and could distort the predictive model due to their anomalous nature.

Data Mining Models/Methods:

Below are the Models Selected based on the dataset and model’s characteristics:

Naïve Bayes Classifier

The Naïve Bayes classifier was chosen initially due to its simplicity and effectiveness in binary classification tasks. This probabilistic model applies Bayes' theorem, assuming independence between predictors, which simplifies computation without sacrificing performance.

Below are the steps performed to work on Naïve Bayes model:

- Base Model Implementation: The Naïve Bayes classifier was chosen as a base model.

- Initial Model on Imbalanced Data: Initially, the model was built using the original, imbalanced dataset. This step was crucial to establish a baseline performance of the model.
- Handling Imbalanced Data: Employed Synthetic Minority Over-Sampling Technique (SMOTE) to address imbalance between the 'Fully Paid' and 'Charged Off' classes.
- Rebuilding Model on Balanced Data: After balancing the dataset, the Naïve Bayes model was retrained to evaluate performance when class distribution is more balanced.

Advantages of Naïve Bayes:

- Efficient and Good Performance: Effective in binary classification tasks, particularly when the assumption of independent predictors holds true.
- Ease of Implementation: The model has simple implementation and interpretation.

Disadvantages of Naïve Bayes:

- Assumption of Independence: Limitation due to assumption of feature independence, potentially leading to biased estimates in real-world scenarios.
- Data Scarcity: May perform poorly with data scarcity in certain classes, affecting probability estimation.

Performance Evaluation:

The Naïve Bayes model's performance was assessed using key metrics: Accuracy, Precision, Recall, and F1-Score on both training and testing datasets.

Model Performance on Imbalanced Data:

- Accuracy: Achieved around 79.68% on training and 79.72% on testing data, indicating good generalization. However, this high accuracy is skewed due to class imbalance.
- Precision and Recall: Precision for the Charged Off class was about 45%, indicating low accuracy in identifying defaults.
- Recall was approximately 16%, showing poor identification of actual defaults.
- F1-Score was around 23%, indicating inefficiency in balancing precision and recall for defaults.
- For the Fully Paid class, precision and recall were higher, resulting in an F1-Score of approximately 88%.

Performance on Balanced Data using SMOTE:

- Accuracy: Decreased to 68.57% on training and 67.35% on testing, reflecting the model's true predictive power across balanced classes.
- Precision and Recall for Charged Off Class: Precision dropped to 31%, indicating more frequent incorrect predictions of defaults.
- Recall improved to 55%, suggesting better identification of actual defaults.
- F1-Score was approximately 40%, reflecting moderate effectiveness.

Summary: The Naïve Bayes model effectively identifies non-default cases but struggles with accurately predicting defaults. The use of SMOTE improved recall for the Charged Off class but reduced precision, leading to false positives.

<u>Random Forest</u>

Because of its dependability and expertise in handling a variety of classification problems, including binary classification, the Random Forest classifier is chosen.

Below are the steps performed to work on Random Forest model:

- Strategic Choice: An ensemble approach, specifically Random Forest, was utilized not as the base model but as an advanced technique to improve upon initial simpler models considered earlier in the process.
- Objective: The model was first applied to the original dataset, which was characterized by significant class imbalance.
- SMOTE Application: To counteract the imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was employed.
- Rebuilding Model on Balanced Data: Following the application of SMOTE, the model was retrained. This was essential for assessing the model's effectiveness in scenarios where the class distribution is balanced.

Advantages of Random Forest:

- High Accuracy and resistance to Overfitting: Random Forest applies the principle of ensemble learning, which is the process of combining multiple classifiers to solve a complex problem and improve the performance of the model. It often produces a
highly accurate model.

- Handling Large Data Sets: It can handle thousands of input variables and identify most significant variables, so it is considered as one of the dimensionality reduction methods. It is also effective in high dimensional spaces as well as large data sets.

Disadvantages of Random Forest:

- Complexity and Resource Intensive: Random Forest generates multiple trees to form a robust model, which significantly increases the computational complexity and resource consumption compared to simpler models like decision trees.
- Low Interpretability: Unlike decision trees, which clearly show decision paths, Random Forest combines the outputs of many trees, making it difficult to interpret and understand the model's decision-making process.
- Long Training Time and High Memory Usage: It requires a longer training period and more memory compared to simpler models because it constructs numerous trees and stores their individual outputs.
- Suboptimal for Linear Relationships: Random Forest may underperform on datasets with linear relationships, where linear models like logistic regression could be more effective and efficient. Additionally, prediction times are slower due to the need to aggregate outputs from multiple trees.

Performance Evaluation:

The Random Forest model's performance was assessed using key metrics: Accuracy, Precision, Recall, and F1-Score on both training and testing datasets.

Model Performance on Imbalanced Data:

- Accuracy: Initially high at approximately 80.61%, indicative of the model's tendency to predict the majority class well due to the imbalance.
- Precision and Recall: Demonstrated low precision (52% for class 0) and even lower recall (8% for class 0), highlighting difficulties in classifying the minority class effectively.
- F1-Score: At 13% for class 0, the model struggled to balance precision and recall, typical in imbalanced datasets.

Performance on Balanced Data using SMOTE:

- Balanced Metrics: Slight reduction in overall accuracy to 80.00%, a more genuine reflection of the model's balanced class performance.
- Enhanced Recall for Minority Class: Notable improvement in recall for class 0 to 13%, showing better identification of minority class instances, albeit at the expense of precision.
- Technique Used: Threshold Moving applied to a Random Forest classifier.
- Purpose: To optimize recall by adjusting the decision threshold.
- Standard Threshold: Originally set at 0.5.
- Adjusted Threshold: Increased to 0.74 based on precision-recall curve analysis.
- Target Recall: Aimed to achieve at least 0.70 for class 0.
- Recall Outcome: Recall for class 0 improved to 59%, indicating a more effective identification of positive cases.
- Precision Impact: Precision for class 0 decreased, highlighting the trade-off with improved recall.
- Overall Accuracy: Achieved 68%, showing the model's effectiveness with the adjusted threshold.
- Trade-off Demonstration: Illustrates the typical precision-recall compromise to meet specific performance criteria.

Summary: While the ensemble model excels in identifying non-default cases, it faces challenges with accurate default predictions. The use of SMOTE has improved the recall for the minority class significantly, suggesting better capability in predicting actual defaults. However, this improvement in recall comes with decreased precision, a common trade-off in such scenarios.

XG Booster-

We chose the XGBoost model as our second choice due to its proven effectiveness in handling both balanced and unbalanced datasets. Building upon the foundational insights provided by the Naïve Bayes classifier, XGBoost offers a more sophisticated and powerful approach to binary classification tasks. Its robust handling of complex data and superior performance metrics make it an ideal option for enhancing the predictive accuracy and reliability required in our analysis.

Below are the steps performed to work on XG Booster model:

Steps Performed with the XGBoost Model:

- Base Model Implementation: XGBoost was utilized to enhance the baseline performance established by the other models.
- Initial Model on Imbalanced Data: The model was first applied to the original imbalanced dataset to establish a performance baseline.
- Handling Imbalanced Data: Techniques like SMOTE, Class Weights, Sampling were employed to correct imbalances between the 'Fully Paid' and 'Charged Off' classes.
- Rebuilding Model on Balanced Data: After balancing the dataset, the XGBoost model was retrained to evaluate its performance with a more equal class distribution.

Advantages of XGBoost:

- Performance Efficiency and Model Robustness: Fast processing, high efficiency and provides strong predictive performance with advanced regularization to prevent overfitting.
- Flexibility: Supports various customization options for tuning and optimization.

Disadvantages of XGBoost:

- Complexity: More complex to tune and interpret compared to simpler models like Naïve Bayes.
- Computational Intensity: While efficient, the training process can be resource-intensive, particularly with large datasets and deep trees.

Performance Evaluation:

- Performance metrics such as Accuracy, Precision, Recall, and F1-Score were utilized to evaluate the model's effectiveness.

Model Performance on Imbalanced Data:

- Accuracy: The XGBoost model achieved a training accuracy of approximately 81.74% and testing accuracy of 80.78%, demonstrating good generalization. However, this high accuracy is somewhat skewed due to class imbalance.
- Precision and Recall:

☐ For the Charged Off class:

Precision: Approximately 68% during training and 54% during testing, indicating challenges in reliably predicting defaults on unseen data.

Recall: Notably low at 13% for training and 10% for testing, suggesting significant difficulty in identifying actual defaults.

F1-Score: 22% for training and 17% for testing, showing poor balance between precision and recall.

☐ For the Fully Paid class:

Precision: Consistently high at around 82%.

Recall: Extremely high at 99% during training and 98% during testing.

F1-Score: 90% for training and 89% for testing, indicating strong performance.

ROC-AUC Scores:

Training: 0.7693, suggesting a decent capability to distinguish between the classes.

Testing: 0.7209, indicating a slight decrease in discriminative power on unseen data.

Performance on Balanced Data using SMOTE, SAMPLING AND CLASS WEIGHT:

Method	Accuracy (Training)	Accuracy (Testing)	Precision (Charged Off)	Recall (Charged Off)	F1-Score (Charged Off)	Precision (Fully Paid)	Recall (Fully Paid)	F1-Score (Fully Paid)	ROC-AUC (Training)	ROC-AUC (Testing)
XGBoost Using SMOTE	88.22%	80.69 %	0.98 (Training) / 0.52 (Testing)	0.78 (Training) / 0.10 (Testing)	0.87 (Training) / 0.17 (Testing)	0.82 (Training and Testing)	0.98 (Training and Testing)	0.89 (Training and Testing)	0.7693	0.7209
XGBoost Using Sampling Method	66.67% (Average)	N/A	0.634 (Average)	0.551 (Average)	0.590 (Average)	0.684 (Average)	0.753 (Average)	0.717 (Average)	N/A	N/A

XGBoost Using Class Weights	70.67%	65.90 %	0.65 (Training) / 0.60 (Testing)	0.71 (Training) / 0.66 (Testing)	0.68 (Training) / 0.63 (Testing)	0.76 (Training) / 0.72 (Testing)	0.70 (Training) / 0.66 (Testing)	0.73 (Training) / 0.69 (Testing)	0.78	0.72

The three methods show varied strengths and weaknesses:

1. SMOTE: Offers the best training performance with high precision and recall but shows a significant drop in precision and recall for 'Charged Off' during testing, indicating potential overfitting to the training data.
2. Sampling Method: Presents more balanced results for both classes but with moderate accuracy, suggesting a better generalization compared to SMOTE but still room for improvement.
3. Class Weights: Provides a balanced approach with relatively good recall and precision across both classes and a modest ROC-AUC score, indicating consistent performance between training and testing, which may be preferable in practical applications where both classes are equally important.

Logistic Regression-

After exploring models like Naïve Bayes, XGBoost, and Random Forest for binary classification in loan default prediction, Logistic Regression is introduced as a strategic choice to complement these advanced models. This decision is driven by Logistic Regression’s capability to provide a clear, interpretable baseline for performance comparison. Known for its straightforward interpretability, Logistic Regression offers transparent insights into how predictors influence the probability of default. Moreover, Logistic Regression outputs probabilities, facilitating detailed risk assessments crucial in financial decision-making.

Steps Performed with the Logistic Regression Model:

- Base Model Implementation: Logistic Regression was implemented to complement the baseline performance established by previous models like Naïve Bayes and XGBoost.
- Initial Model on Imbalanced Data: Initially, the model was applied to the original, imbalanced dataset to assess its performance and establish a baseline.

- Handling Imbalanced Data: Techniques such as weight assignment and SMOTE were utilized to address the imbalances between the 'Fully Paid' and 'Charged Off' classes.
- Integration of PCA with SMOTE: In addition to standard SMOTE, Principal Component Analysis (PCA) was combined with SMOTE to enhance data preprocessing, aiming to improve model performance by reducing dimensionality and focusing on significant features before balancing.
- Rebuilding Model on Balanced Data: After applying these techniques to balance the dataset, the Logistic Regression model was retrained to evaluate its performance under more equitable class distribution conditions.

Advantages of Logistic Regression:

- Interpretability and Simplicity: Offers straightforward interpretability, which is valuable for explaining outcomes directly to stakeholders and for use in regulatory environments.
- Probabilistic Outputs: Provides probabilities for outcomes, which can be crucial for decision-making processes that require risk assessment.

Disadvantages of Logistic Regression:

- Assumption of Linearity: Assumes a linear relationship between the independent variables and the logit of the dependent variable, which might not hold in complex scenarios.
- Performance Under High Dimensionality: Can perform poorly if there are highly correlated inputs or the dimensionality is very high, without prior dimensionality reduction.

Performance Evaluation:

Performance Metrics: Accuracy, Precision, Recall, and F1-Score were utilized to comprehensively evaluate the effectiveness of the Logistic Regression model across different setups and data balancing techniques.

Model Performance on Imbalanced Data:

- Accuracy: The Logistic Regression model demonstrated a training accuracy of approximately 80.27% and testing accuracy of 80.36%, showing good generalization. However, this high level of accuracy is influenced by the class imbalance present in the dataset.

- Precision and Recall:

□ For the Charged Off class:

Precision: Recorded at 45% during training and 42% during testing, indicating challenges in accurately predicting defaults on unseen data.
Recall: Notably low at 2% for both training and testing, highlighting significant difficulty in identifying actual default cases.
F1-Score: 5% for training and 4% for testing, which shows a poor balance between precision and recall, reflecting the model's inefficacy in predicting defaults accurately.

□ For the Fully Paid class:

Precision: Consistently high at approximately 81%.
Recall: Extremely high at 99% during training and testing, showcasing the model's effectiveness in identifying non-default cases.
F1-Score: 89% for both training and testing, indicating strong performance and reliable predictions for non-default loans.

- ROC-AUC Scores:

Training: 0.7693, suggesting a decent ability to distinguish between classes despite the imbalanced data.
Testing: 0.7209, indicating a slight decrease in discriminative power on unseen data, which may reflect the model's limitations under real-world conditions.

This evaluation highlights the Logistic Regression model's tendency to perform well in terms of overall accuracy and identifying 'Fully Paid' loans but struggles significantly with the
'Charged Off' class due to the imbalanced nature of the dataset. The low recall rate for defaults
is a critical area for improvement, necessitating methods to enhance the model’s sensitivity to the minority class without compromising its high performance on the majority class.

Model Performance on balanced Data using WEIGHT ASSIGN, SMOTE, Using PCA and SMOTE

Method	**Accuracy (Training)**	**Accuracy (Testing)**	**Precision (Charged Off - Training / Testing)**	**Recall (Charged Off - Training / Testing)**	**Precision (Fully Paid - Training / Testing)**	**Recall (Fully Paid - Training / Testing)**	**F1-Score (Charged Off - Training / Testing)**	**F1-Score (Fully Paid - Training / Testing)**	**ROC-AUC (Training)**	**ROC-AUC (Testing)**
Weight Assignment	60.59 %	60.59 %	0.29 / 0.28	0.67 / 0.67	0.88 / 0.88	0.59 / 0.59	0.40 / 0.40	0.71 / 0.71	0.6791	0.6773
SMOTE	66.35 %	61.73 %	0.65 / 0.27	0.70 / 0.57	0.68 / 0.86	0.63 / 0.63	0.67 / 0.37	0.65 / 0.73	0.7254	0.6414
PCA and SMOTE	72.55 %	72.48 %	0.33 / 0.32	0.39 / 0.38	0.84 / 0.84	0.81 / 0.81	0.36 / 0.35	0.83 / 0.83	0.6616	0.6582

When comparing the three logistic regression models adjusted for balanced data:

- SMOTE stands out for significantly improving recall without drastically sacrificing precision, especially in the training phase.
- PCA with SMOTE provides the best overall accuracy and generalizes well to unseen data, making it the most robust choice among the three. It maintains a good balance between precision and recall, especially for the "Fully Paid" class, which makes it effective in practical scenarios where both classes are important.

The choice between these models would depend on the specific requirements and objective:

If maximizing overall accuracy and generalizability is crucial, PCA with SMOTE is recommended.

If focusing on identifying more defaults is critical, even at the cost of precision, SMOTE could be preferable.

	Model Performance on Imbalanced Data		
Model	Accuracy	Recall	F1 Score
Naïve Bayes	80%	Charged off - 15% Fully Paid – 95%	Charged off - 23% Fully Paid – 88%
Logistic Regression	80%	Charged off - 3% Fully Paid – 99%	Charged off - 5% Fully Paid – 89%
Random Forest	81%	Charged off - 8% Fully Paid – 98%	Charged off - 13% Fully Paid – 89%
XG Boost	81%	Charged off - 13% Fully Paid – 99%	Charged off - 22% Fully Paid – 90%

Model	Accuracy	Recall	F1 Score
Naïve Bayes	67%	Charged off - 15% Fully Paid – 95%	Charged off - 55% Fully Paid – 70%
Logistic Regression	72%	Charged off - 38% Fully Paid – 81%	Charged off - 35% Fully Paid – 83%
Random Forest	68%	Charged off - 59% Fully Paid – 70%	Charged off - 42% Fully Paid – 78%
XG Boost	65.81%	Charged off - 66% Fully Paid – 66%	Charged off - 17% Fully Paid – 89%

Model Performance on Balanced Data(All Balance data Metrices are mentioned in separate tables in respective sections)

Conclusion:

1. Model Best for Predicting "Fully Paid" Loans: XGBoost with SMOTE

- Accuracy and Precision: XGBoost with SMOTE showcases a high accuracy of 80.69% on the testing set with a precision of 0.82 for the "Fully Paid" class. This indicates that it reliably identifies loans that will be fully repaid.
- Recall for "Fully Paid": The model achieves a very high recall of 0.98 for "Fully Paid" loans during testing, ensuring that nearly all loans that are not going to default are correctly identified.
- Balance and Generalization: It maintains a balanced performance with a decent ROC-AUC score of 0.7209, indicating a good capability to distinguish between "Fully Paid" and "Charged Off" cases under varied conditions.
- Application Suitability: This model is particularly valuable in financial environments where securing income through reliable loan repayments is crucial—minimizing false negatives is vital to maintain cash flow and reduce credit risk.

2. Model Best for Detecting "Charged Off" Loans: Logistic Regression with Class Weights

- Focus on Recall: With an adjusted recall of 0.67 for "Charged Off" loans in the testing phase, this model is specifically effective at identifying potential defaults, which is critical for risk management.
- Precision and Accuracy Trade-offs: While the overall testing accuracy is lower at about 60.59%, the model emphasizes detecting as many defaults as possible—even at the cost of more false positives (lower precision).
- Strategic Application: This approach is most beneficial in scenarios where avoiding financial losses from defaults is more critical than the occasional misclassification of a potential good loan. The high recall ensures that fewer defaults slip through, thus protecting the financial institution from high-risk loans.
- Operational Implications: Given the high recall, this model is suitable for environments where the consequences of a missed default are severe, such as in high-value loans or where defaults could trigger significant financial repercussions.