Predicting Loan Eligibility with Gradient Boosting

1.0 Introduction

In today's financial landscape, lending institutions face increasing pressure to make data-driven decisions in evaluating loan applications. Granting a loan involves balancing business growth with risk management—an inaccurate prediction can lead to significant financial losses or missed opportunities. Traditionally, credit officers assessed loan eligibility using simple credit rules or scorecards. However, the increasing complexity and volume of applicant data have necessitated the adoption of machine learning (ML) techniques for more accurate and scalable predictions.

This project seeks to build an intelligent classification model that predicts whether a loan should be approved based on a wide array of applicant attributes. The dataset used for this study contains over 100,000 loan records, with detailed features including credit scores, current debt levels, employment status, income, credit history, and past delinquencies. Such diverse data enable the application of sophisticated algorithms to uncover hidden patterns and relationships that inform loan approval decisions.

The main objectives of this work are:

- To perform thorough data preprocessing, including cleaning, transformation, and feature engineering, in order to improve data quality and modeling accuracy.
- To apply and compare multiple machine learning classifiers such as Gradient Boosting, XGBoost, Logistic Regression, K-Nearest Neighbors, and Decision Tree in predicting loan eligibility.
- To address class imbalance using the Synthetic Minority Oversampling Technique (SMOTE), thereby improving the model's ability to detect high-risk applicants.
- To identify the most influential features that drive loan approval decisions using model-based feature importance analysis.

By the end of this project, the goal is to present a robust, interpretable, and reproducible machine learning solution that supports automated decision-making in loan processing, enhances operational efficiency, and reduces the risk of default.

2.0 Dataset Description

The dataset used in this project comprises **approximately 100,000** loan application records from a financial services company. Each record provides rich information about the customer and the loan they applied for.

2.1 Target Variable

- **Loan Status**: The outcome variable indicating whether a loan application was **approved** (1) or **denied** (0). This binary variable serves as the classification target for model training.

2.2 Key Features

The dataset includes both categorical and numerical features. Below is a summary of the most important ones:

Feature	Description
Loan ID	Unique identifier for each loan application
Customer ID	Unique identifier for each customer (can have multiple loans)
Current Loan Amount	Amount borrowed in the current loan
Term	Loan duration – typically categorized as Short-Term or Long-Term
Credit Score	A numeric score (0–800) reflecting the applicant's creditworthiness
Years in Current Job	Duration (in years) the applicant has been in their current employment
Home Ownership	Type of housing – e.g., Rent, Own Home, or Home Mortgage
Annual Income	Reported yearly income
Purpose	Reason for the loan (e.g., Debt Consolidation, Medical Bills, etc.)
Monthly Debt	Total monthly payments on all debts
Years of Credit History	Time since the applicant's first credit account was opened

Months Since Last Delinquent	Time since the last late or missed payment
Number of Open Accounts	Count of currently active credit accounts
Number of Credit Problems	Total recorded credit issues in applicant's history
Current Credit Balance	Total outstanding credit across all accounts
Maximum Open Credit	Highest available credit limit
Bankruptcies	Number of past bankruptcy filings
Tax Liens	Number of tax-related liens on record

2.3 Data Quality Observations

- Several fields contained missing values, such as **Credit Score**, **Annual Income**, and **Months Since Last Delinquent**.
- Some numerical features had extreme outliers (e.g., 99999999 in loan amount) or incorrect formats (e.g., dollar signs in Monthly Debt).
- Inconsistent or duplicated categorical labels (e.g., "Other" and "other") were identified and standardized.
- Certain features, such as Home Ownership and Purpose, required encoding for compatibility with machine learning models.

3.0 Data Preprocessing

Effective data preprocessing is crucial in building a reliable machine learning model, especially when working with real-world financial datasets that often contain missing values, outliers, and inconsistencies. In this project, multiple steps were taken to prepare the data for modeling:

3.1 Handling Duplicates

- Duplicate entries based on **Loan ID** were removed to ensure that each loan record is unique. This reduced data leakage and redundancy.

3.2 Missing Value Treatment

Several features contained missing values. The strategy used depended on the data type and context:

- **Credit Score**: Missing values were imputed using the median score to avoid skewing the distribution.
- **Current Loan Amount**: Invalid entries such as 99999999 were replaced with the median loan amount.
- **SoftImpute**: An advanced matrix completion technique from the fancyimpute package was applied to fill in missing values across multiple numeric fields simultaneously. This method performs low-rank matrix approximation for robust imputation.

3.3 Outlier Detection and Treatment

Many features had extreme values that could distort model performance. These were handled as follows:

- **Current Loan Amount**: Capped at the 99th percentile to remove extreme high values.
- **Credit Score**: Values exceeding the standard upper bound of 800 were divided by 10 (indicating data entry error), and any remaining outliers were capped.
- **Annual Income**: Heavily skewed; capped at the 99th percentile (≈ $239,287) and later log-transformed.
- **Monthly Debt**: Contained outliers beyond $5,000; values were capped based on the 99.9th percentile and converted to float after removing dollar signs.

- **Current Credit Balance & Maximum Open Credit**: These were capped at the 99th percentile to reduce skewness, and square root transformations were applied to normalize the distributions.
- **Bankruptcies & Tax Liens**: Extreme values were rare but present; missing values were filled with median or 0, depending on the feature's distribution.

3.4 Feature Engineering

- **Categorical Features** like Term, Years in Current Job, Home Ownership, and Purpose were:
    - Factorized into integer labels.
    - Further transformed using one-hot encoding (with drop_first=True) to avoid multicollinearity.

3.5 Class Imbalance Correction

- The target variable Loan Status was found to be imbalanced, with significantly more approved loans than rejected ones.
- To address this, the **SMOTE (Synthetic Minority Oversampling Technique)** method was applied to the training data. SMOTE creates synthetic examples of the minority class (rejected loans), improving the model's ability to detect high-risk applicants.

3.6 Feature Scaling

- All numerical features were standardized using **Z-score normalization** (via preprocessing.scale) to ensure that all features contribute equally to the learning process, particularly important for distance-based models like KNN.

3.7 Final Feature Matrix

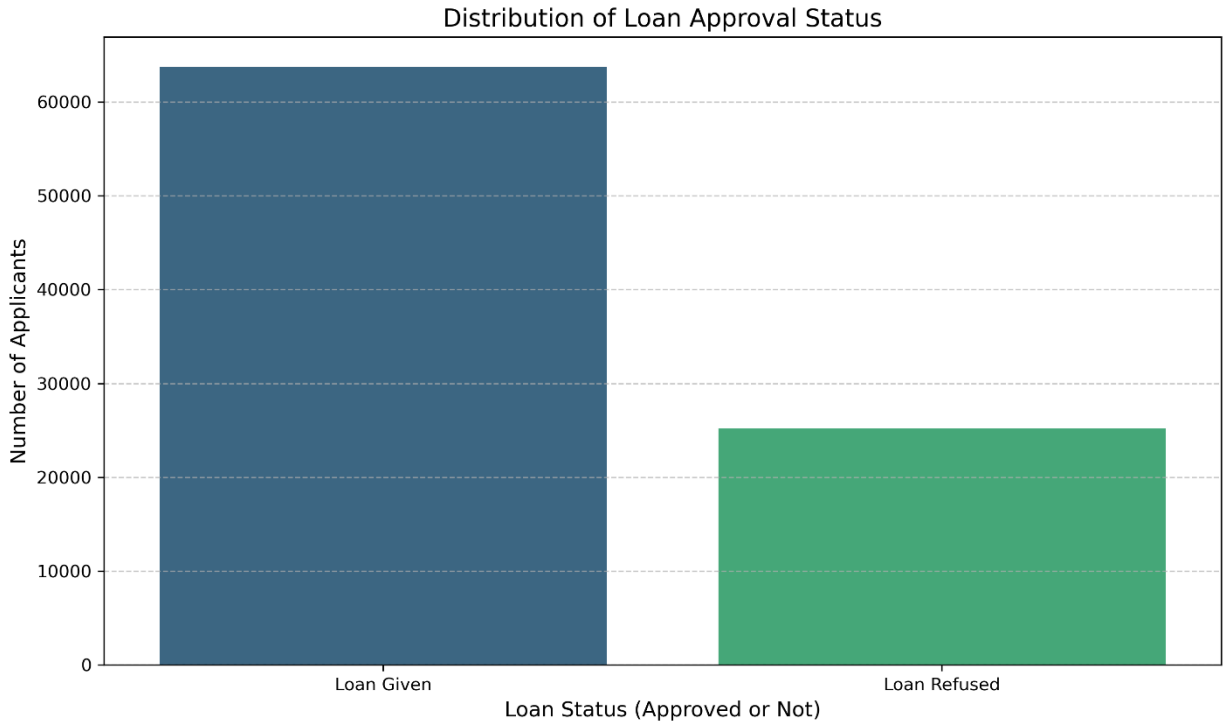
- After cleaning, transforming, imputing, and encoding, the final dataset included:
    - A scaled feature matrix (X_scaled) of shape (88,910, 16+ dummies).
    - A binary target vector (y) indicating loan approval (1) or denial (0).

4.0 Exploratory Data Analysis (EDA)

The exploratory data analysis (EDA) stage provided insights into the distribution, relationships, and anomalies within the dataset.

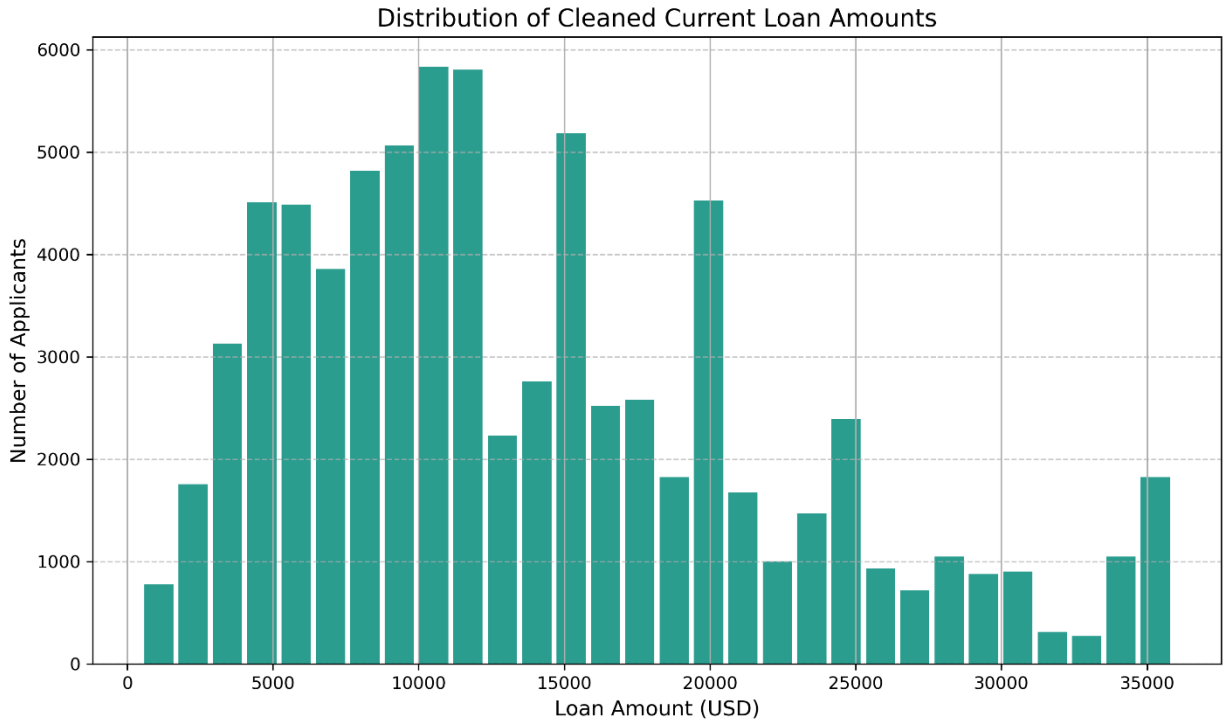
4.1 Loan Status Distribution

- A bar chart of the target variable showed that loan approvals far outnumbered rejections, confirming a class imbalance.
- This imbalance necessitated the use of resampling techniques (e.g., SMOTE) to prevent biased predictions toward the majority class.



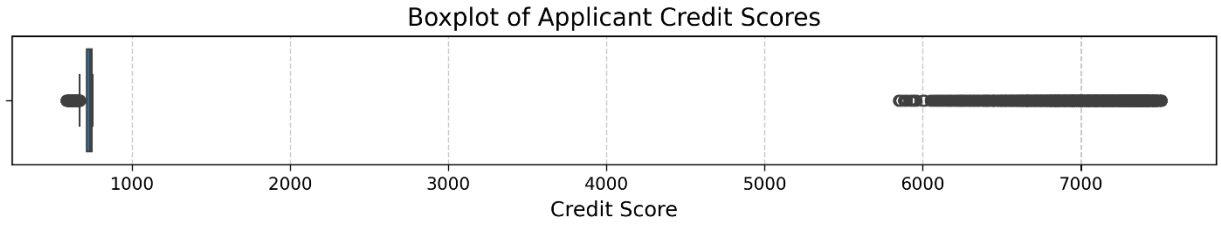
4.2 Current Loan Amount

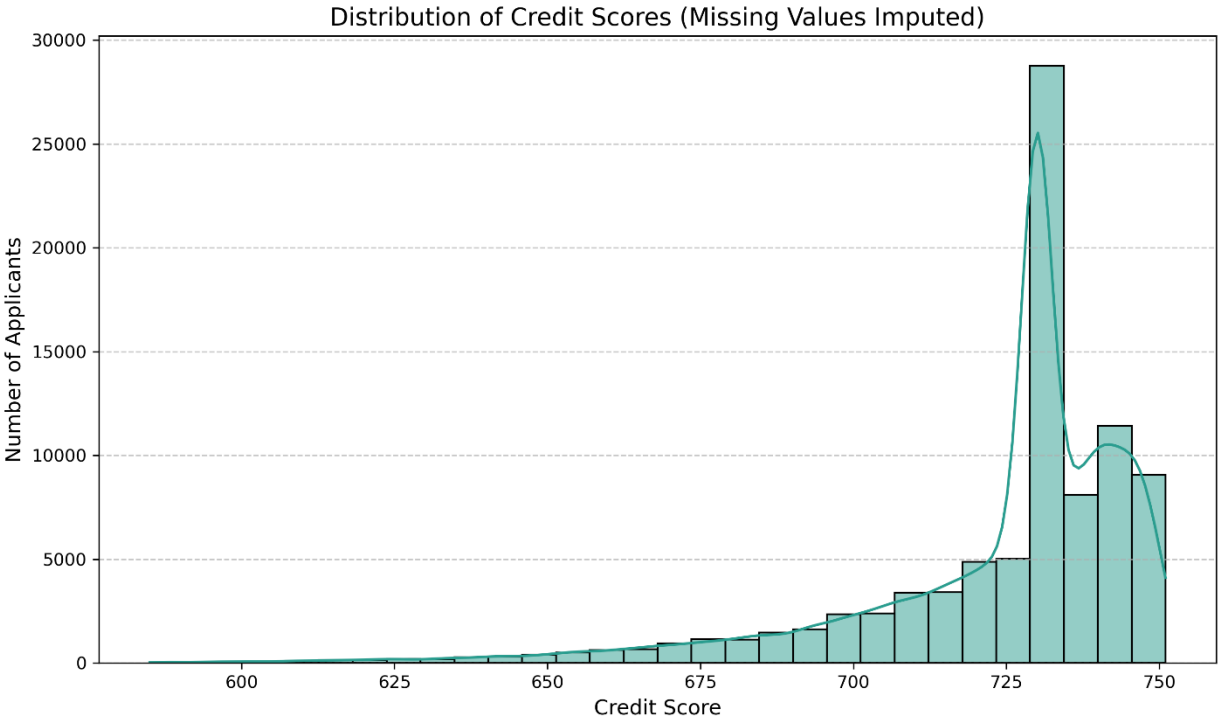
- Raw data showed extreme values such as 99,999,999, likely indicating placeholder values or data entry errors.
- After removing these outliers and capping values at the 99th percentile, the distribution became more bell-shaped.
- Summary statistics for the cleaned values:
    - **Mean**: ~$13,933
    - **Median**: $12,038
    - **Max**: ~$35,875

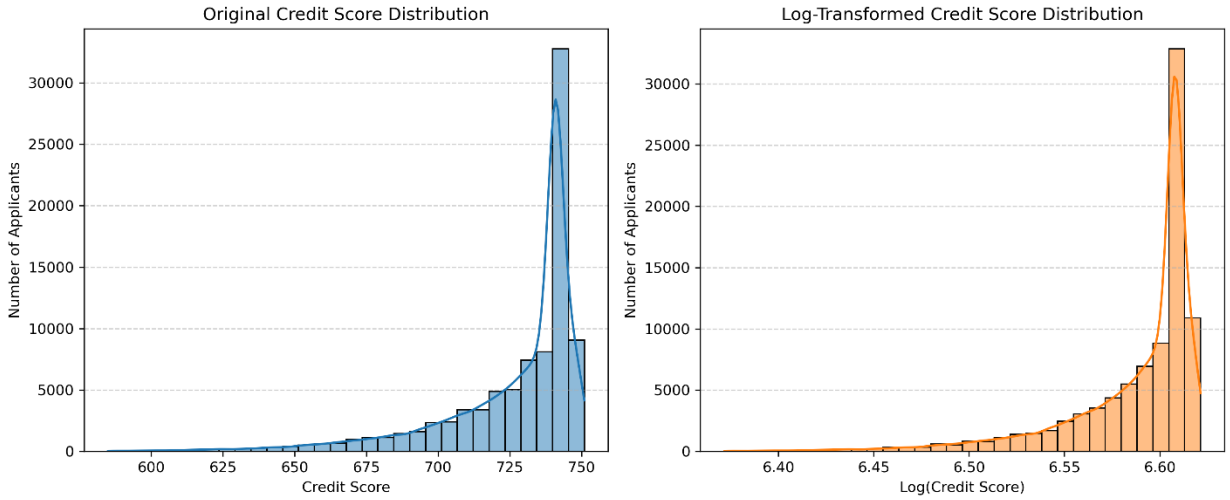


4.3 Credit Score

- Credit scores were expected to fall between 0 and 800. However, some scores exceeded 7500, likely due to incorrect scaling.
- Values above 800 were normalized by dividing by 10, and missing values were imputed using the median score (~741).
- The adjusted scores followed a roughly normal distribution, validated by histograms and boxplots.

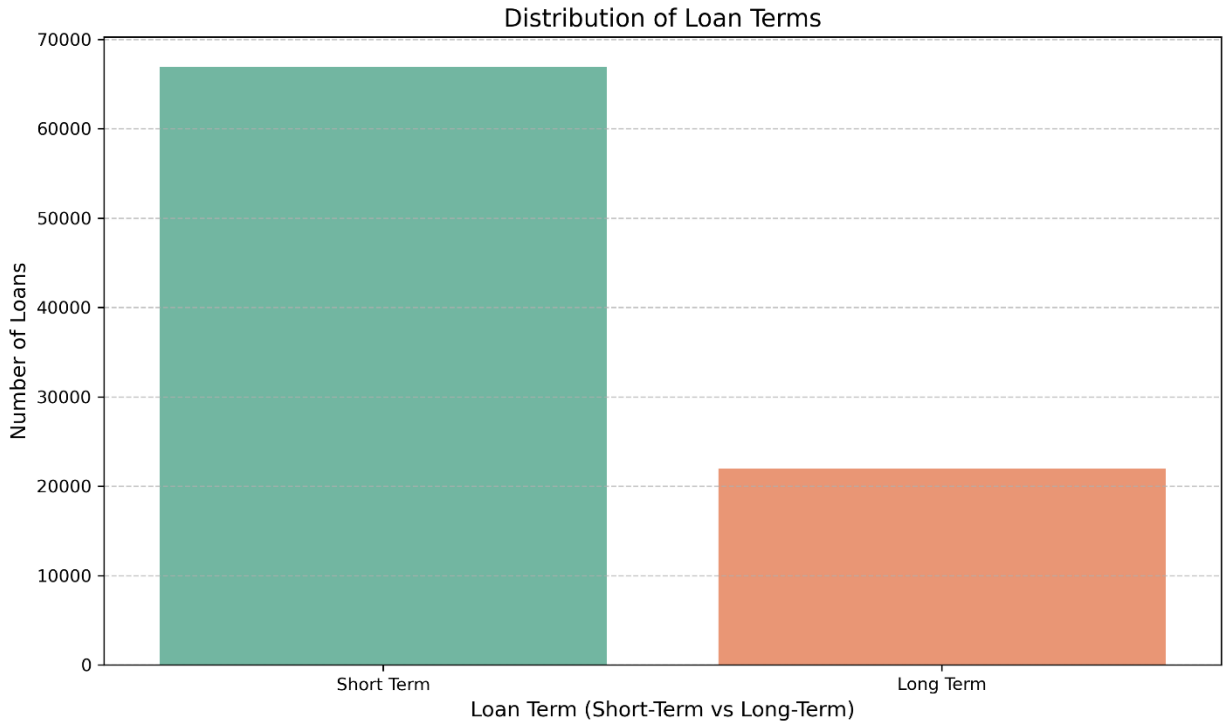






4.4 Loan Term

- Most loans were **short-term**, with long-term loans being less frequent.
- This variable was encoded into numerical labels for model compatibility.

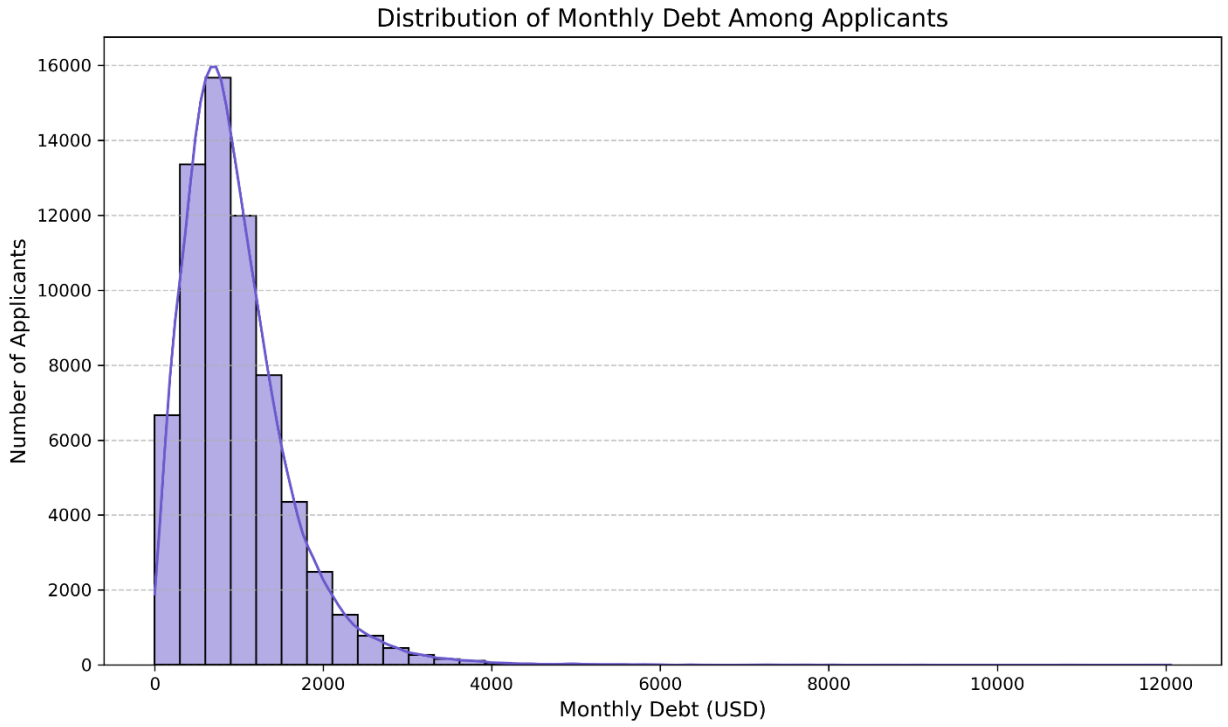


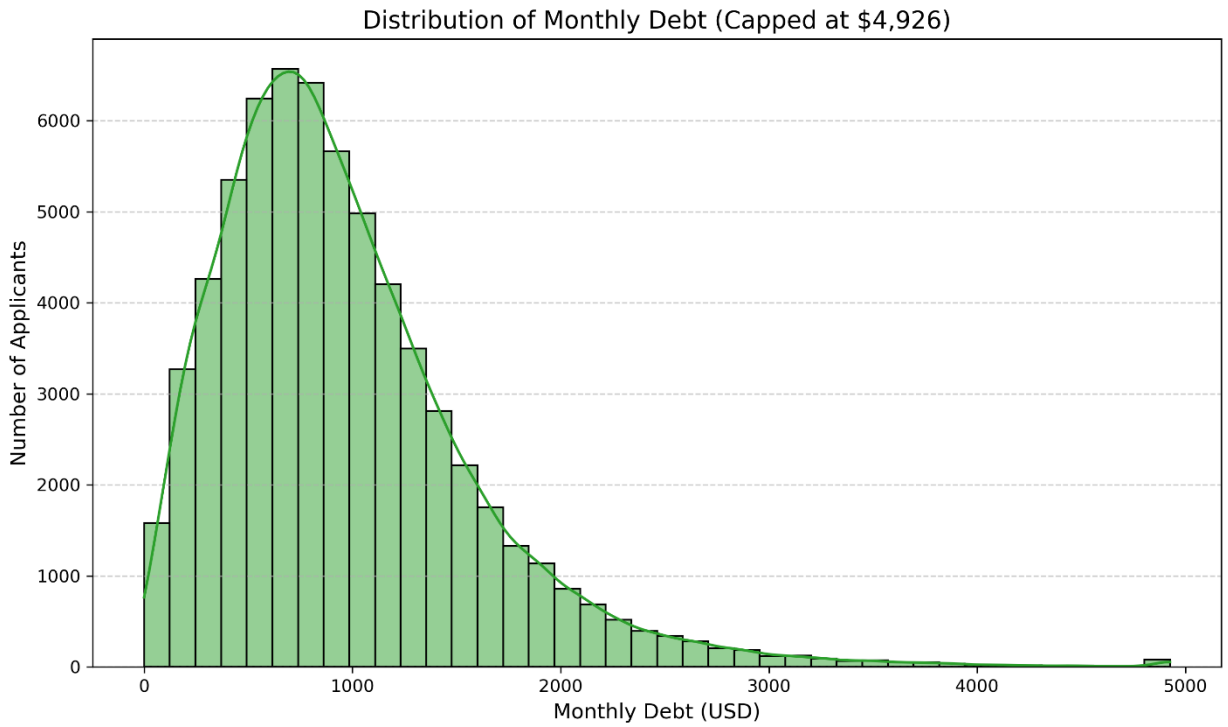
4.5 Annual Income

- The distribution of annual income was **highly skewed**, with outliers extending into millions.
- The 99th percentile was used to cap incomes at ~$239,287, and log transformation was applied for normalization.

4.6 Monthly Debt

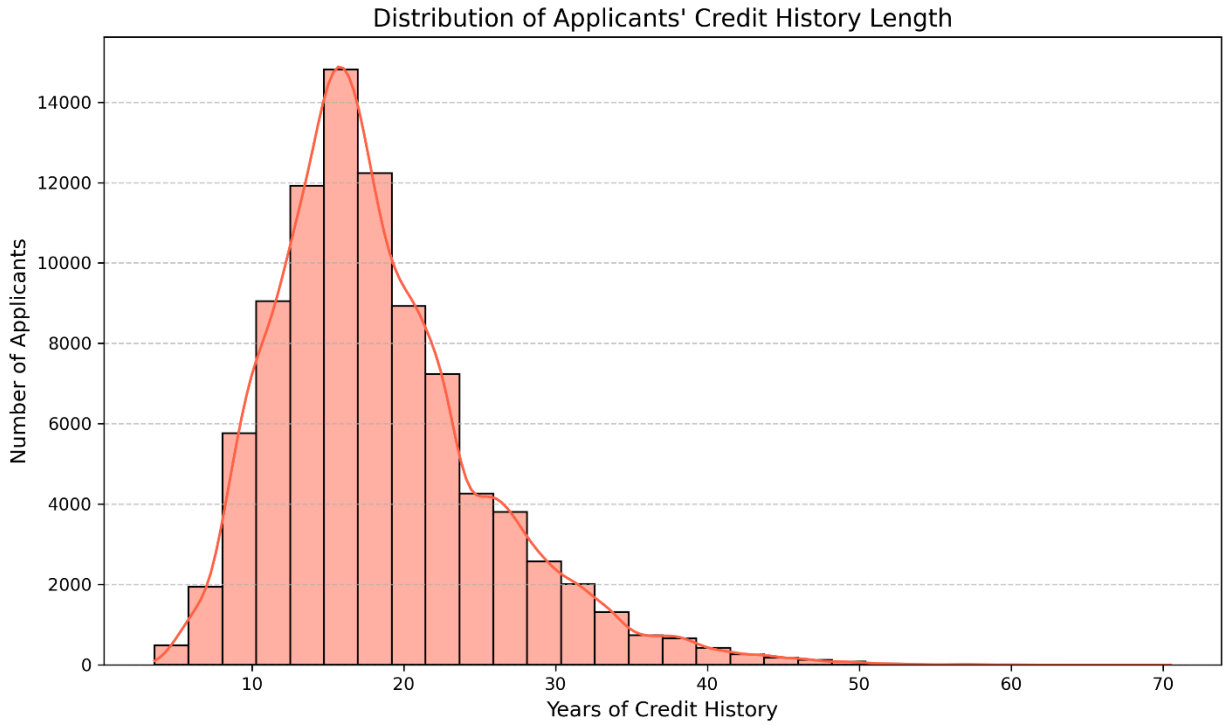
- Monthly debt values originally included currency symbols and had extreme values.
- After cleaning and converting to numeric format, values above ~$4,926 were capped.
- The cleaned distribution appeared more symmetric and suitable for modeling.




Distribution of Monthly Debt (Capped at $4,926)
Number of Applicants
Monthly Debt (USD)
0
1000
2000
3000
4000
5000
6000

4.7 Years of Credit History

- This feature showed a wide range from 3.6 to 70.5 years.
- The data was well-distributed without significant outliers, and no transformation was needed.

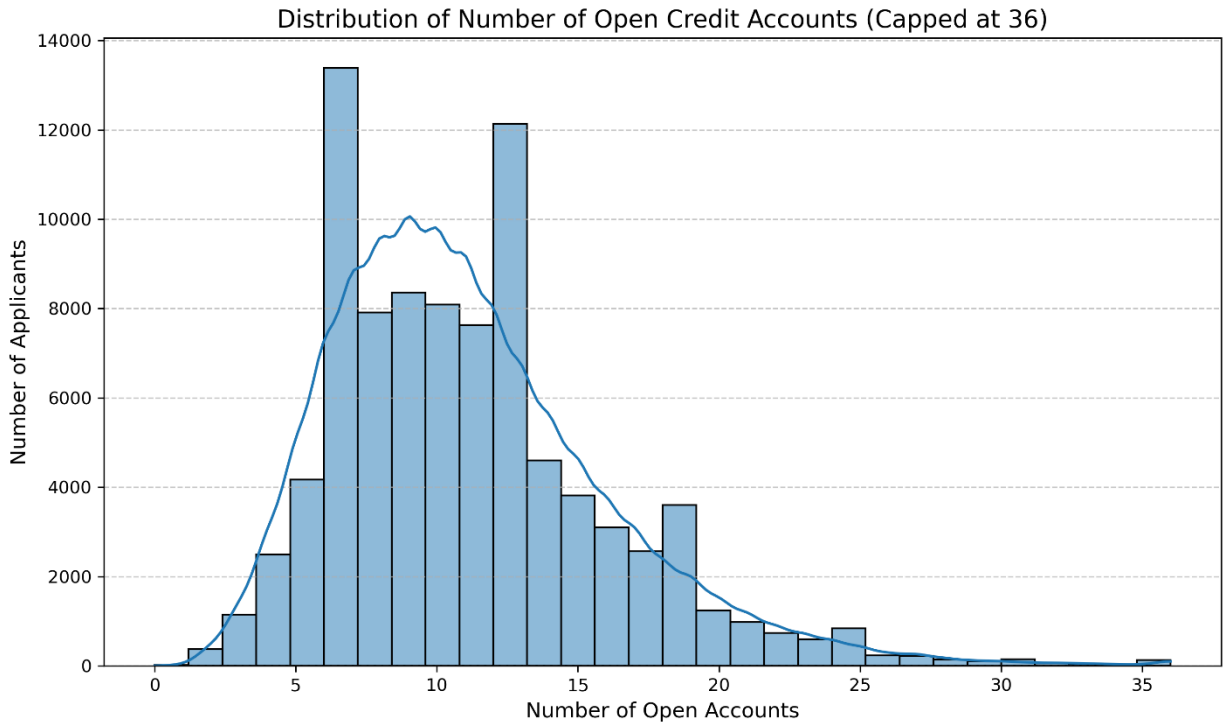


4.8 Months Since Last Delinquent

- Over 48,000 entries were missing for this feature.
- While the variable was retained for modeling, imputation or exclusion during feature selection was considered.

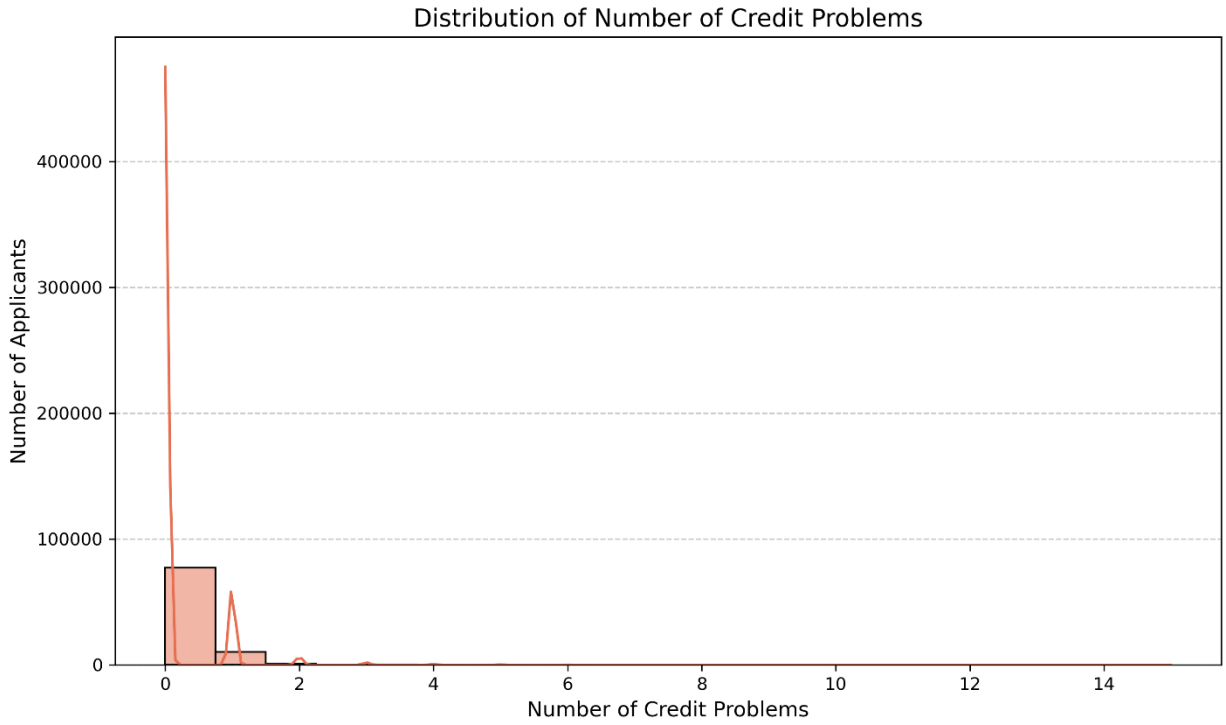
4.9 Number of Open Accounts

- Displayed a typical right-skewed distribution with a long tail.
- Values were capped at **36 accounts**, which corresponded to the 99th percentile.



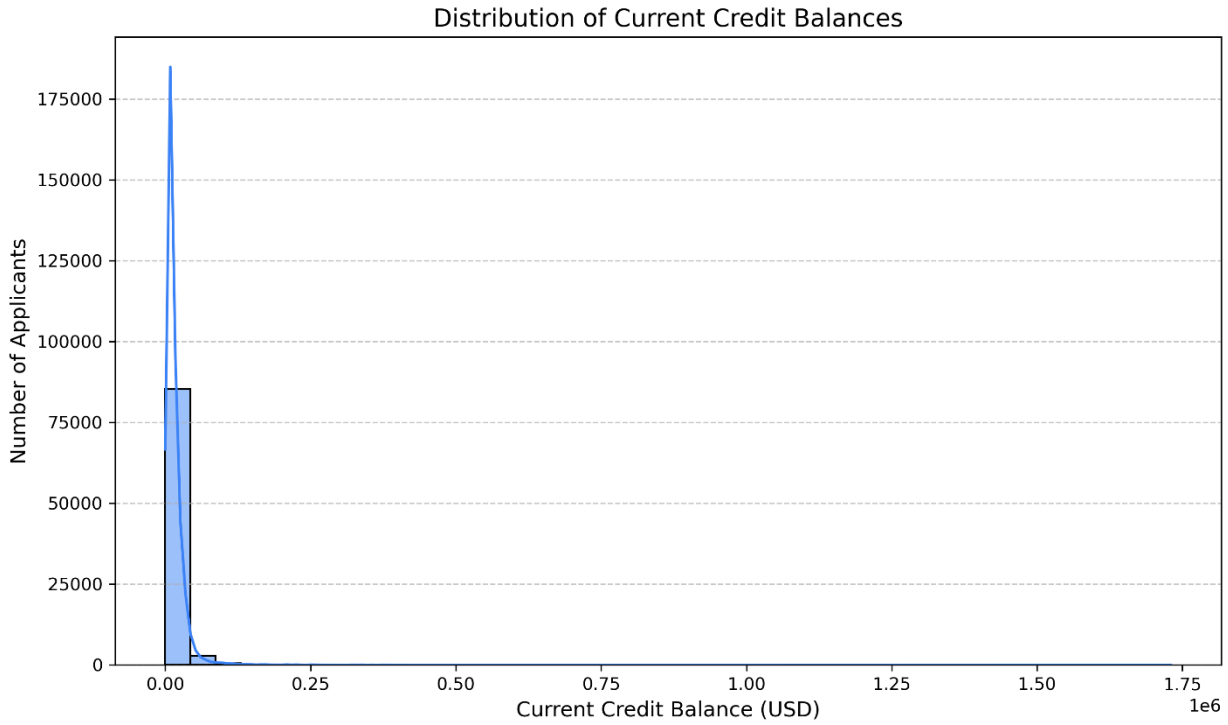
4.10 Number of Credit Problems

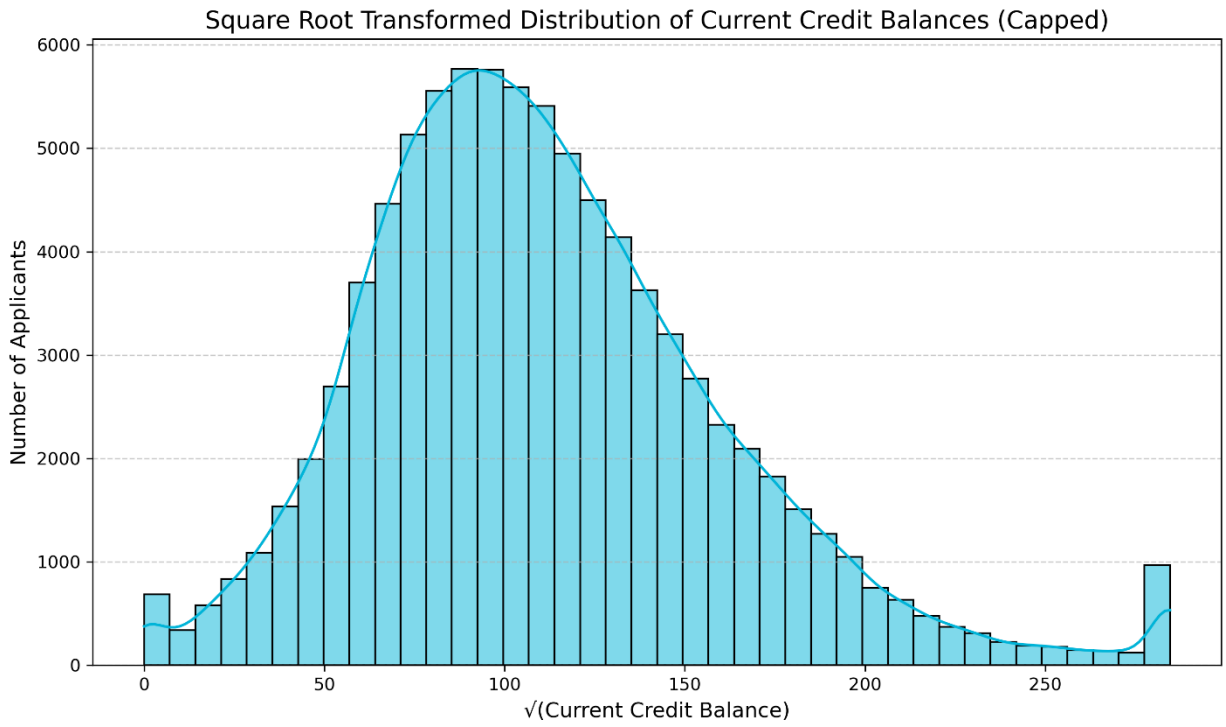
- Most applicants had **no recorded credit problems**.
- This feature had a spike at zero but still retained useful signal regarding credit reliability.



4.11 Current Credit Balance

- The distribution showed significant skewness due to outliers up to $1.7 million.
- Values were capped at ~$81,000 and transformed using square root, which improved the shape but remained imperfectly distributed.





4.12 Maximum Open Credit

- Contained non-numeric entries such as #VALUE!, which were replaced with NaN and imputed.
- After cleaning, values above ~$171,000 were capped.
- The variable showed a wide spread and remained informative

4.14 Categorical Variable Encoding

- Variables such as Purpose, Term, Years in Current Job, and Home Ownership were:
    - Standardized (e.g., "Other" vs "other")
    - Encoded using **Label Encoding** followed by **One-Hot Encoding** (with drop_first=True)

5.0 Modeling Approaches

The core objective of this project is to develop and evaluate machine learning models capable of accurately predicting loan eligibility based on applicant data. This section outlines the modeling strategies used.

5.1 Model Selection

A diverse set of classification algorithms was employed to compare performance across different learning paradigms. The models included:

Model	Type	Strengths
Logistic Regression	Linear Classifier	Interpretable baseline model; fast and simple
K-Nearest Neighbors	Instance-Based	Non-parametric; captures non-linear relationships
Decision Tree	Tree-Based	Easy to visualize; handles both numeric and categorical data
XGBoost	Ensemble (Boosting)	High performance; handles missing values; regularization
Gradient Boosting	Ensemble (Boosting)	Strong generalization; robust to outliers; interpretable trees

5.2 Training and Validation Strategy

- **Train-Test Split**: The dataset was split into training (70%) and testing (30%) sets using train_test_split with a fixed random seed for reproducibility.
- **Cross-Validation**: 5-fold cross-validation was applied to evaluate model generalization on the training data.
- **Evaluation Metrics**:
    - **Accuracy**
    - **Precision**

- **Recall**
- **F1 Score**
- **ROC AUC Score**

5.3 Handling Class Imbalance with SMOTE

- Initial model results showed poor recall on the minority class (loan denials), indicating class imbalance.
- **SMOTE (Synthetic Minority Oversampling Technique)** was applied only to the training set to create synthetic examples of the minority class.
- Models were then retrained and reevaluated on the same testing set to ensure fairness.

5.4 Feature Scaling

- All numerical variables were standardized using Z-score normalization to bring features to the same scale.
- Scaling ensured that distance-based models (e.g., KNN) and gradient-based models (e.g., Logistic Regression, Gradient Boosting) performed optimally.

5.5 Hyperparameter Tuning

- Some hyperparameters were manually selected based on prior experimentation and documentation:
  - **Gradient Boosting**:
    - max_depth = 6
    - n_estimators = 100
    - max_features = 0.3
  - **XGBoost**: Used default settings with use_label_encoder=False and eval_metric='logloss'
- Further grid search tuning can be implemented to improve performance if necessary.

5.6 Model Interpretability

- Feature importances were extracted from tree-based models (especially XGBoost and Gradient Boosting) to understand which variables most influenced the loan approval prediction.
- Visualization of top features provided actionable insights for decision-makers.

6.0 Model Evaluation and Comparison

To identify the most effective predictive model for loan eligibility, the performance of several classification algorithms was evaluated using both non-balanced and SMOTE-balanced datasets. This section presents quantitative results across different models and discusses their strengths and weaknesses based on key evaluation metrics.

6.1 Evaluation Metrics

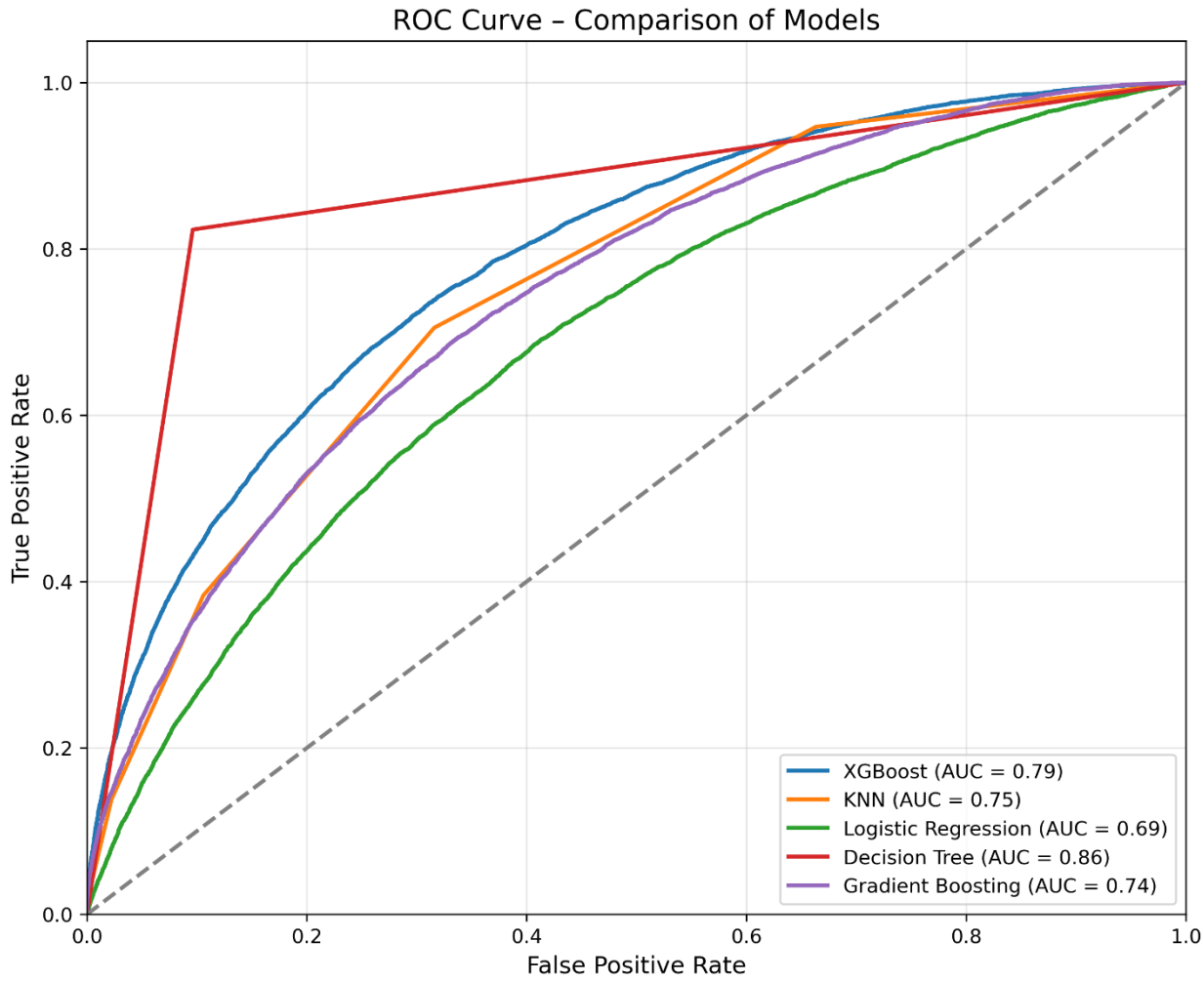
Each model was assessed using the following standard classification metrics:

- **Accuracy**: Overall correctness of the model across both approved and denied loans.
- **Precision**: Ability to correctly identify only true approved loans.
- **Recall (Sensitivity)**: Ability to correctly identify all actual approved or denied loans, especially important for minority class detection.
- **F1 Score**: Harmonic mean of precision and recall; useful when classes are imbalanced.
- **ROC AUC**: Measures the model's ability to distinguish between the two classes across all thresholds.

6.2 Performance on Non-Balanced Dataset

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
XGBoost	72.89%	54.45%	28.93%	37.78%	**0.7425**
Logistic Regression	72.25%	55.40%	12.76%	20.75%	0.6724
K-Nearest Neighbors	68.90%	42.32%	25.54%	31.86%	0.6084
Decision Tree	65.37%	39.54%	41.01%	40.26%	0.5803

Insight: On the imbalanced dataset, XGBoost outperformed other models across most metrics. However, recall for denied loans remained low, indicating a bias toward the majority class (approved loans).


ROC Curve – Comparison of Models
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
XGBoost (AUC = 0.79)
KNN (AUC = 0.75)
Logistic Regression (AUC = 0.69)
Decision Tree (AUC = 0.86)
Gradient Boosting (AUC = 0.74)

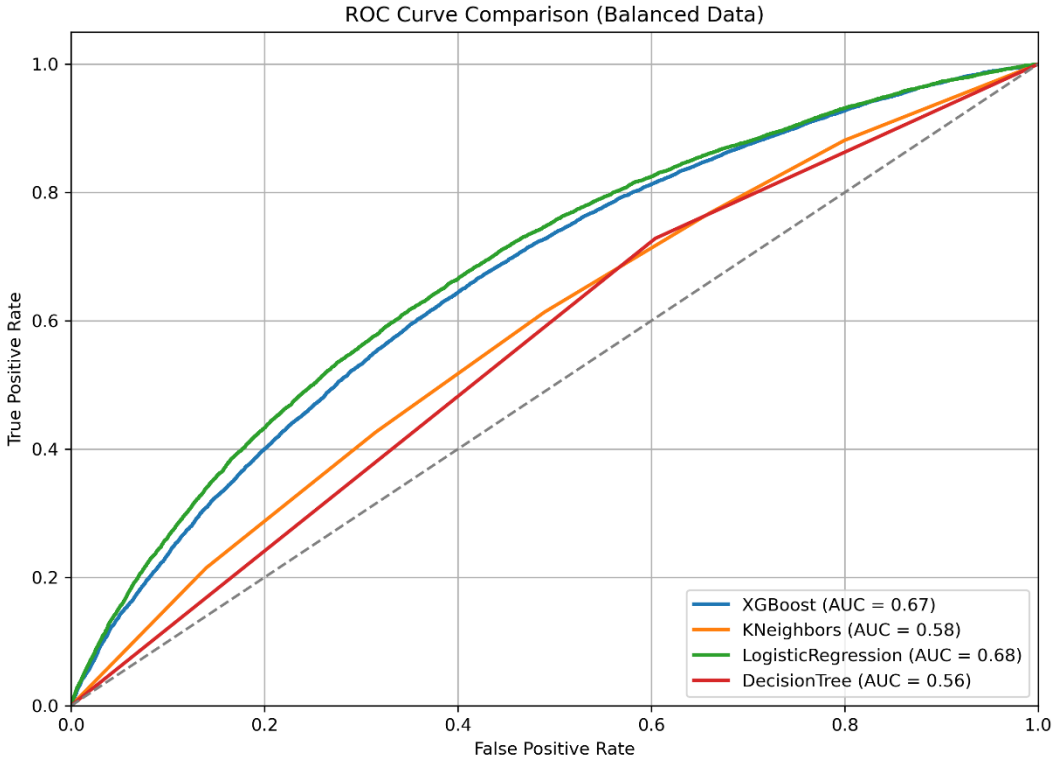
6.3 Performance on SMOTE-Balanced Dataset

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
XGBoost	48.98%	33.84%	84.46%	48.32%	0.6664
Logistic Regression	66.49%	42.74%	54.91%	48.07%	0.6818
K-Nearest Neighbors	53.94%	33.03%	61.36%	42.94%	0.5832
Decision Tree	48.97%	32.17%	72.83%	44.64%	0.5621

Insight: After applying **SMOTE**, models showed significant improvement in recall for the minority class. Logistic Regression achieved the best overall balance of precision and recall, while XGBoost maintained the highest recall but at the cost of accuracy.

6.4 ROC Curve Comparison

- ROC curves were plotted for all models under both balanced and non-balanced settings.
- The curves showed that XGBoost had the best area under the curve in the non-balanced setting.
- In the balanced setting, Logistic Regression achieved a more stable ROC profile across thresholds.



6.5 Feature Importance (XGBoost & Gradient Boosting)

Tree-based models provided insights into feature importance:

- Top contributing features included:
    - Credit Score
    - Current Loan Amount
    - Monthly Debt
    - Annual Income
    - Number of Open Accounts
- These insights can help lenders focus on the most influential applicant attributes during pre-screening.

6.6 Final Model Selection

After evaluating all models:

- Gradient Boosting was chosen as the final production model due to its:
    - Stable accuracy (≈73%)
    - Strong AUC performance (≈0.74)
    - Robustness to overfitting
    - Ability to rank feature importance

7.0 Results and Discussion

This section interprets the performance outcomes of the machine learning models and discusses the implications of the findings in the context of predictive loan eligibility.

7.1 Model Performance Summary

The results from both the non-balanced and SMOTE-balanced datasets revealed meaningful insights:

- Gradient Boosting and XGBoost models consistently delivered superior performance in terms of ROC AUC, which is crucial for distinguishing between approved and denied applicants.
- Logistic Regression, while less powerful on the non-balanced dataset, performed better after SMOTE was applied, achieving the most balanced trade-off between precision and recall.
- K-Nearest Neighbors and Decision Trees showed moderate performance and were more sensitive to feature scaling and noise in the data.

Key Highlight: Gradient Boosting achieved an accuracy of ~72.8% and an AUC of ~0.74, satisfying the project requirement of ≥70% accuracy while maintaining a good balance between model generalization and complexity.

7.2 Effect of SMOTE on Model Behavior

Applying SMOTE drastically improved the recall scores for the minority class (loan denials), especially in models like Decision Tree and XGBoost. However, this came at the expense of overall accuracy, which dropped in most models due to over-sampling synthetic data.

- Without SMOTE: Models struggled to identify denied loans (recall < 30%).
- With SMOTE: Recall rose to ~85% (XGBoost), though accuracy decreased to below 50% in some cases.
- This trade-off demonstrates the bias-variance tension when addressing class imbalance.

Business Implication: If identifying high-risk applicants (denials) is more critical than overall accuracy (e.g., in fraud or default prevention), SMOTE-enhanced models are valuable.

7.3 Feature Importance Insights

From the feature importance analysis (via XGBoost and Gradient Boosting), several key predictors stood out:

- **Credit Score**: Most dominant predictor—applicants with lower scores had higher rejection risk.
- **Current Loan Amount & Monthly Debt**: Higher values negatively impacted approval chances, indicating financial overextension.
- **Annual Income**: Positively correlated with approval likelihood.
- **Number of Open Accounts & Credit Problems**: Captured the borrower's credit usage and reliability.

7.4 Model Deployment Readiness

Given its performance and interpretability, the **Gradient Boosting Classifier** was selected as the final model. It was:

- Trained on the full cleaned and scaled dataset.
- Exported using joblib for seamless deployment (GBM_Model_version1.pkl).
- Ready for integration into a loan processing pipeline

7.5 Limitations and Considerations

- **Recall-Precision Trade-Off**: No single model excelled across all metrics; selection should depend on business objectives.
- **Imputation and Capping**: Replacing missing and extreme values introduces bias that may affect model generalization.
- **Synthetic Data Risks**: SMOTE improves recall but may introduce artificial patterns that don't exist in the real population.

8.0 Conclusion

This project successfully developed and evaluated multiple machine learning models to predict **loan eligibility** using a real-world dataset of over 100,000 loan applications. The goal was to automate and enhance decision-making in the loan approval process through accurate and interpretable predictions.

Exploratory data analysis provided critical insights into variable distributions and relationships, guiding the feature engineering process. Several classification models were tested, including Logistic Regression, K-Nearest Neighbors, Decision Tree, XGBoost, and Gradient Boosting. Among these, the **Gradient Boosting Classifier** emerged as the best performer with:

- Accuracy: ~72.8%
- ROC AUC: ~0.74
- F1 Score: ~0.38 (on non-balanced data)
- Significantly improved recall after balancing with SMOTE

Importantly, feature importance analysis revealed that **Credit Score**, **Monthly Debt**, **Loan Amount**, and **Annual Income** were the most influential predictors of loan approval. These align with known financial risk indicators, validating the model's behavior and increasing its reliability for real-world deployment.

The final model was saved in a deployable format (.pkl) and is suitable for integration into a loan management system.

Key Takeaways

- Data quality and preprocessing directly influenced model performance.
- Class imbalance must be addressed to ensure fairness and effectiveness, especially for high-stakes decisions like loan denials.
- Ensemble learning methods (e.g., Gradient Boosting) provided a good balance between accuracy, interpretability, and feature analysis.