



# INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

## Customer Credit Default Prediction

<sup>1</sup>Devasane Malleshham

<sup>1</sup>Asst Professor

<sup>1</sup>Department of CSE IOT,

<sup>1</sup>ACE Engineering College, Hyderabad, India

**Abstract:** This document presents a data-driven approach to identify customers who may fail to repay their credit obligations. The study used structured financial records that included behavioral and personal characteristics to build and evaluate prediction tools. After cleaning the data for missing values, converting categories, and scaling numeric fields, we scaled it for pattern and distribution analyses. We trained and validated the XGBoost, Random Forest, Decision Tree, and Logistic Regression algorithms. To determine the performance of each model, we used the F1 Score, Precision Score, ROC AUC and Recall Score. Among all the tested algorithms, the predictions of the Random Forest algorithm were very reliable owing to its accurate interpretability and prediction.

**Index Terms** - Credit default prediction, Random Forest, XGBoost, Logistic Regression, feature selection, machine learning, ROC AUC, F1-score, class imbalance, Scikit-learn, one-hot encoding.

### I. INTRODUCTION

Determining whether a customer might miss a credit payment is a real-world challenge that banks and financial companies face every day. The ability to identify risky borrowers before issuing credit can significantly reduce financial losses, streamline approval workflows, and enhance compliance with regulations. This analysis was based on a public dataset available from an online platform (Kaggle, 2023).

Our objective was to build a model that can determine whether a customer will default on their credit. In addition, another objective of the project was to determine which ML technique is good in terms of accuracy and practicality. Finally, this project will help gain insights that could make credit decision-making easier for banks.

### II. DATA PROCESSING

To provide consistency within the dataset (Kaggle, 2023), the median of each numeric column was used to fill in the missing values. The system ensures the use of unskewed or original data for analysis. This helped eliminate the risk of bias due to the outliers.

## Dataset Overview:

- The Training Set has 70,000 records with 82 features.
- Test Set: 33,000 records with 80 features
- Target Column: The target column indicates whether a customer has defaulted (1) or not (0).

## Data Preparation and Transformation:

**Managing Missing Data:** Missing values in numerical features were imputed using their respective median values. The median was preferred because it is robust to outliers, ensuring that extreme values do not distort the data.

For example, if the income column had missing entries, they were filled with the median income value of the entire column.

## Categorical Encoding:

To handle categorical data, one-hot encoding was applied, transforming string or Boolean values into numeric form. Encoding was applied consistently across the training and testing data. To maintain uniformity, the encoder was fitted using a combined set of unique category values from both datasets.

Example: Gender: {'Male' → 0, 'Female' → 1}

## Feature Scaling:

We used RobustScaler to scale the numerical features. Unlike standard scaling methods, RobustScaler uses the interquartile range (IQR), which makes it highly effective in the presence of outliers, and the models are not overly influenced by extreme values.

## Feature Alignment:

To prevent inconsistencies during prediction, we ensured that the training and test datasets had the same columns. This included dropping extra columns if present in either set and ensuring a consistent column order.

## Target Isolation:

We labeled the target column such that 0 corresponds to non-defaulters, and 1 marks cases where the customer defaulted.

## III. EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) provides an initial overview, helping us explore trends, anomalies, and relationships within the data. It involves identifying irregularities, exploring feature behavior, and forming early hypotheses. In this project, EDA allowed us to examine how different variables interact with the outcome (credit default), which was essential for selecting meaningful features for modeling.

### Univariate Analysis

We began by examining the distributions of key features to understand their spread, outliers, and potential impact.

### Numerical Features:

The dataset consists of fields such as number of open credit lines, Monthly Income, Age, and Debit Ratio. Some variables showed right-skewed distributions, especially financial variables such as `debt_ratio` and `monthly_income`.

### Categorical Features:

One-hot encoding was used to convert categorical variables of Boolean and object types into numerical types for modeling. Certain categorical features had imbalanced value counts, which might influence the model bias if not handled appropriately.

Example Chart: This revealed a long tail, indicating a small number of customers with very high debt ratios.

### Bivariate Analysis

We analyzed the features related to the target variables.

#### Box Plots:

Here `debt_ratio` and `number_of_dependents` were higher on average among defaulters.

#### Correlation Matrix:

Heatmaps helped identify weak to moderate correlations. Most features showed low linear correlation, suggesting that nonlinear models (e.g., trees) might perform better.

#### Default Distribution:

Only ~20.6% of records were labeled as defaulters. This class imbalance requires attention during model training (e.g., via `class_weight='balanced'` or resampling techniques).

#### Key Visual Insights

Target Imbalance: Defaulted (1): ~20% and non-defaulted (0): ~80%. This significant imbalance influenced our choice of metrics (F1-score and ROC AUC instead of accuracy).

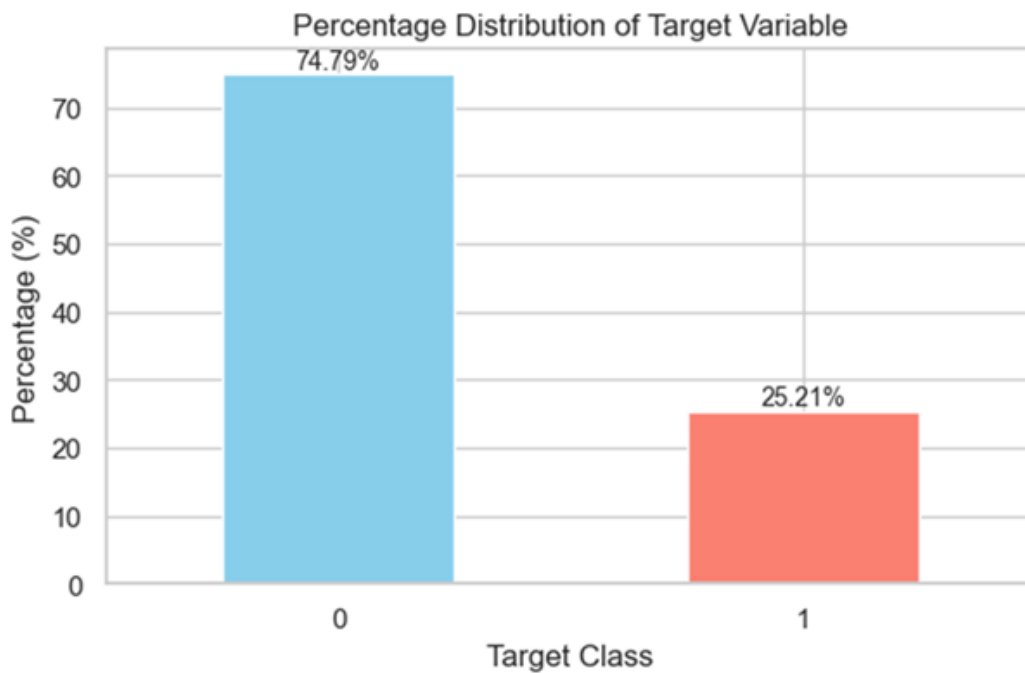


Fig. 1. Class distribution of default (1) and non-default (0) samples.

**Outliers:** Features such as `monthly_income` have extreme values. Instead of dropping, we used the `RobustScaler` to mitigate their influence.

**Missing Data:** We found missing data in the income and dependent count fields. Imputed using median values, preserving the distribution of the shape.

### 3.1 Feature Selection

The initial dataset contains 189 features. To enhance the model performance and generalization and minimize overfitting, we implemented a two-step feature selection strategy using statistical techniques.

#### Information Value (IV)

A metric used for classification is the Information Value, which indicates the ability of a feature to differentiate between two classes in the target variable. Experts use it to judge how much each variable can predict credit ratings. Features that have very low Information Value (IV) are not useful, whereas very high IVs may indicate leakage.

By applying IV filtering, we removed 85 features that were either insignificant or flagged as suspicious because of unusually high IV scores.

#### Pearson Correlation

To further eliminate redundancy, we calculated the correlation coefficients between all numerical features. Features with high pairwise correlation ( $\geq 75\%$ ) can cause multicollinearity, leading to model instability and a high variance.

This step led to the removal of 16 additional features that showed a high correlation with other predictors.

## Final Feature Set

After completing both IV and correlation-based filtering, we reduced the dataset from 189 to 72 features. This streamlined feature space improved the efficiency and reliability of our machine learning algorithms.

## IV. MODEL SELECTION

Selecting an appropriate model is vital for any real-world, real-time data analysis. This is particularly true in the case of unbalanced classes. The final set of 72 features was chosen to train the model using IV and correlation-based filtering. We conducted our selection across algorithms with different complexities and interpretabilities.

### Candidate Models

We explored four supervised classification algorithms, each with unique advantages.

### Hyperparameter Tuning

All models were optimized for ROC AUC using GridSearchCV, applying a 5-fold cross-validation strategy.

### Logistic Regression:

Logistic regression is the first go-to classification method, when the relationship between predictors and outcome is almost linear (Hosmer, Lemeshow, & Sturdivant, 2013).

### Grid: C values and solvers

To correct for label disparity in the target variable, we enabled class balancing using the `class_weight` setting.

### Decision Tree:

Decision trees provide an intuitive structure for modeling decisions and are widely used because of their interpretability and low computational cost (Quinlan, 1986). To improve performance, we explored combinations of `minimum_samples_leaf`, `maximum_depth`, and `minimum_samples_split` during the hyperparameter optimization.

### Random Forest:

Random Forest enhances decision tree performance by using ensemble learning to decrease overfitting and increase accuracy (Breiman, 2001). Hyperparameter optimization included adjusting the maximum depth, estimator quantity, and minimum split threshold.

### XGBoost:

XGBoost is a widely used library with high boosting performance. It can efficiently handle sparse data and incorporates regularization to prevent overfitting (Chen and Guestrin, 2016). We performed parameter tuning using `max_depth`, `learning_rate`, and the number of estimators for better performance. The evaluation metric used during training was log loss to assess the classification quality.

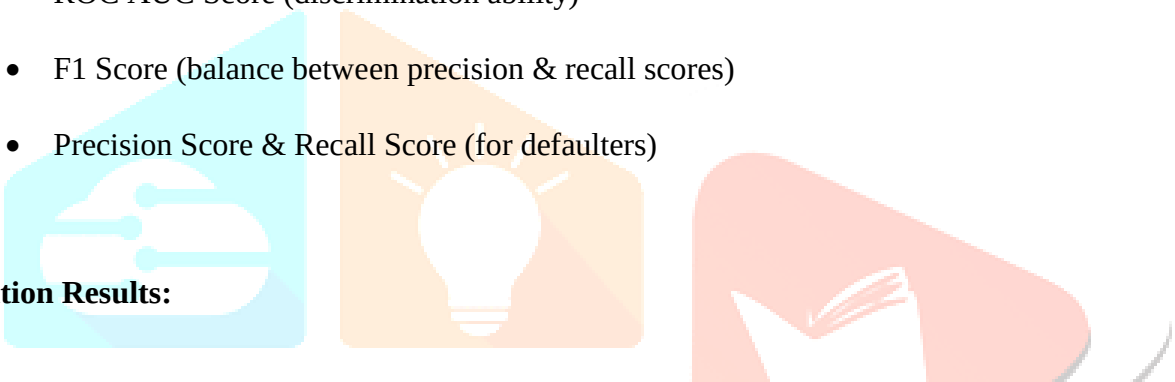
**Tools Used:**

- GridSearchCV was used for tuning (Pedregosa et al., 2011a).
- The RobustScaler was used to scale the numerical features (Pedregosa et al., 2011b).
- One-hot encoding was used for categorical encoding (Pedregosa et al., 2011c).

**Evaluation Strategy**

To evaluate the models fairly, we split the training data as follows:

- 80% for training, 20% for validation
- The performance was measured using:
  - ROC AUC Score (discrimination ability)
  - F1 Score (balance between precision & recall scores)
  - Precision Score & Recall Score (for defaulters)

**Validation Results:**


| Metric        | Random Forest | XGBoost     | Decision Tree | Logistic Regression |
|---------------|---------------|-------------|---------------|---------------------|
| Accuracy      | <b>0.91</b>   | 0.90        | 0.85          | 0.83                |
| ROC AUC       | <b>0.9633</b> | 0.9576      | 0.8808        | 0.9057              |
| F1 Score (1)  | <b>0.8049</b> | 0.7906      | 0.6855        | 0.7134              |
| Precision (1) | <b>0.85</b>   | 0.82        | 0.75          | 0.63                |
| Recall (1)    | 0.76          | <b>0.76</b> | 0.63          | <b>0.82</b>         |

Fig. 2. Overview of performance metrics for all models.

**Final Model Selection**

The final model was selected by visualizing the F1 score and ROC AUC score. An Outstanding ROC AUC score was achieved by the Random Forest. This model also achieved high precision, recall, and F1 scores, making it a suitable choice as the best final model for our analysis.

## V. MODEL ANALYSIS

After training and tuning, we evaluated the results on the validation set of each model to test its efficacy. We wanted to determine which model can identify a defaulter without generating too many unnecessary false positives.

### Classification Reports

The precision, recall, and F1 metrics helped assess how well the selected algorithms detected defaulting customers, with a primary focus on class 1 instances.

### Understanding the Evaluation Metrics

- Precision tells us how accurate the model's default predictions were; in other words, out of the predicted defaulters, how many were actually correct?.
- Recall measures how effectively the model detects customers who defaulted.
- The F1 Score is a balanced metric of recall and precision scores, helping us understand the overall model effectiveness for the default class.
- The ROC AUC can correctly separate defaulters from non-defaulters across all thresholds — the higher, the better.
- Random Forest led to the highest F1 and ROC AUC, making it the best-suited model for production.

### Visual Insights:

We plotted the key evaluation curves to support our findings.

1. The ROC Curve indicates that the highest AUC was obtained in the case of the Random Forest, which is indicative of the best classification.

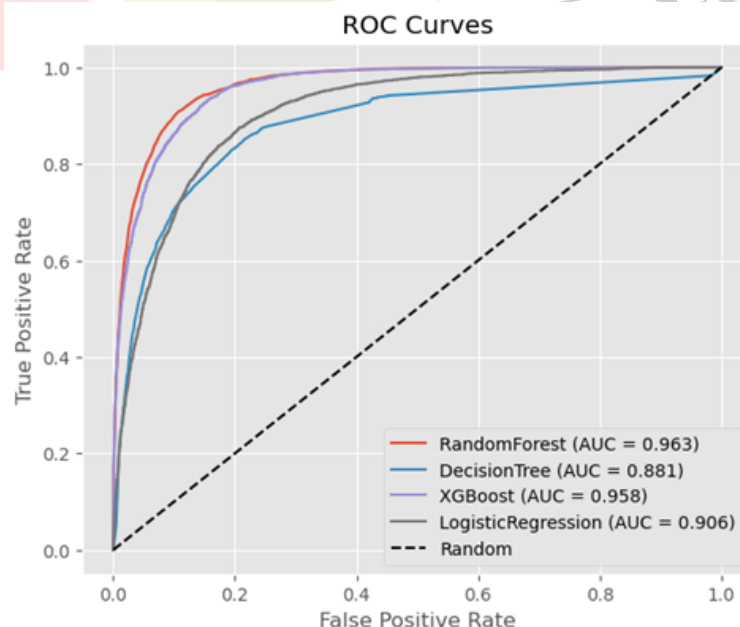


Fig. 3. ROC curves showing the classifier discrimination ability.

2. The precision-recall curve shows that the model effectively handles class imbalance.

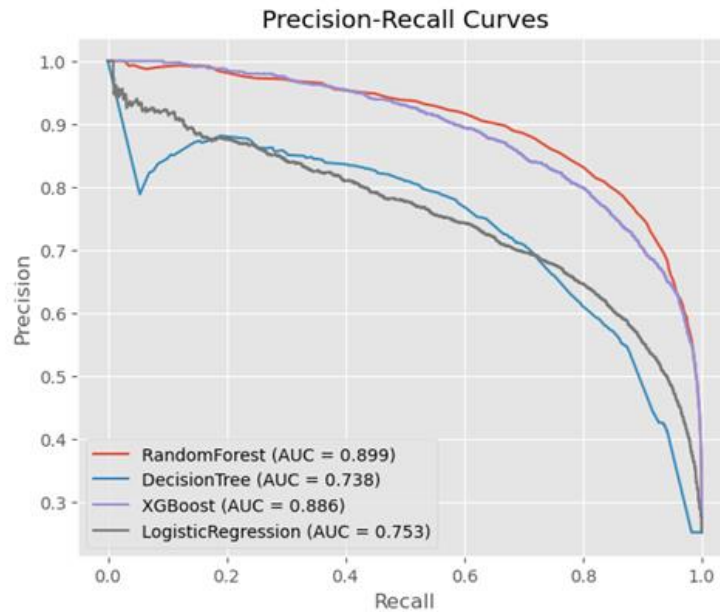


Fig. 4. Precision-recall curves for all classifiers.

3. Model Comparison Bar Chart.

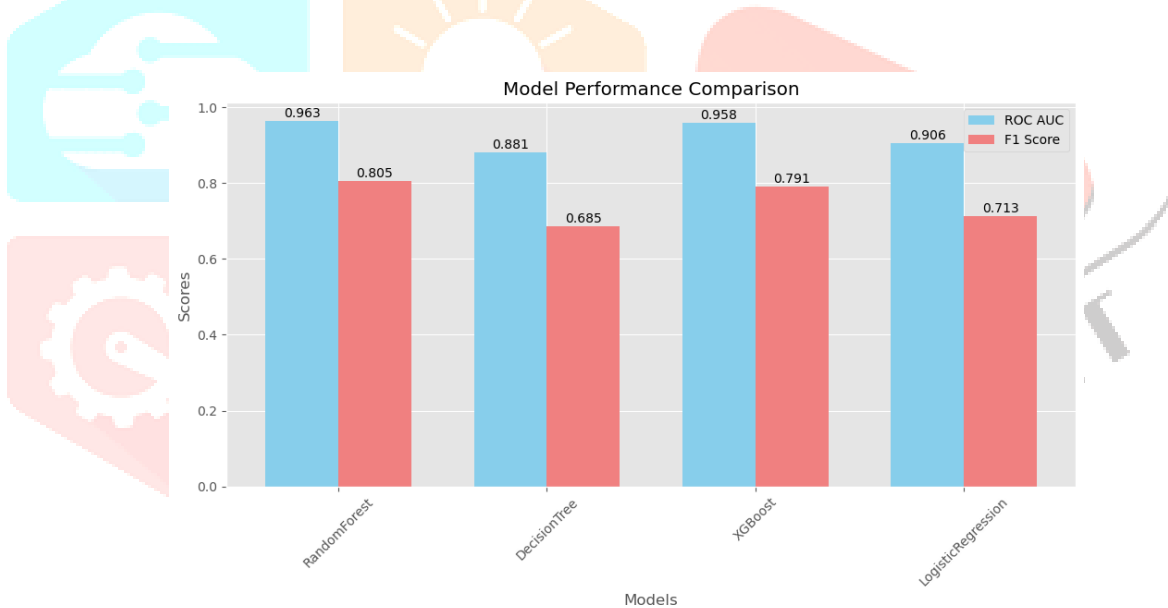


Fig. 1. Comparison of ROC AUC and F1 Scores across models.

### Summary of Observations

- Random Forest performed the best overall, particularly in accurately identifying defaults.
- XGBoost was a close second and can be explored further for optimization.
- Logistic Regression, while simple, showed surprisingly good recall but low precision—making it risky for production.
- The Decision Tree was interpretable but too simplistic for this complex dataset.

## Why Choose a Random Forest?

- Among all the models tested, the highest ROC AUC and F1-score.
- It demonstrates consistent performance across both the defaulter and non-defaulter classes.
- It effectively handles noisy data and is resistant to outliers.
- It is recommended for datasets that contain a combination of numerical and categorical features, making it ideal for financial applications.

## VI. CONCLUSION AND RECOMMENDATIONS

### Project Recap:

This project focused on developing a machine learning technique to identify the likelihood of customer credit default using structured real-world data. Dimensionality reduction via IV and correlation analysis improved model efficiency and reduced noise, ultimately boosting performance. Through systematic pre-processing, model building, and evaluation, we explored multiple classification models and identified the most effective solution.

### Key Findings:

The Random Forest showed the best dependable model, demonstrating strong predictive accuracy and the best balance between correctly identifying defaults and avoiding false alarms, as shown below.

XGBoost showed competitive results, slightly behind Random Forest, and can be considered for further tuning or ensemble techniques. Logistic Regression, though basic, demonstrated strong recall (0.82), meaning it rarely missed defaulters, but suffered from low precision (0.63), potentially flagging too many false positives. Decision Tree underperformed, indicating a need for deeper tree-based models or ensemble methods for such complex data.

### Business Implications:

A robust credit default prediction model offers several advantages for banks and financial institutions. It helps minimize financial losses by early identification of high-risk customers. It improves the customer experience by expediting approvals for low-risk applicants. Such models also support regulatory compliance by being both explainable and auditable. Additionally, they help reduce operational costs by automating certain aspects of the decision-making process.

### Recommendations

To further improve the model performance and business adoption:

**Feature Engineering:** Create features such as trend indicators, rolling averages, or recent transaction history. Explore time-window aggregation for behavior-based features.

**Model Optimization:** Perform more extensive hyperparameter tuning (e.g., using a randomized search). Consider implementing ensemble techniques, such as stacking, by combining the strengths of high-performing models such as Random Forest and XGBoost. This hybrid approach may enhance the prediction stability and improve the overall model performance.

**Bias and Fairness Audits:** It essential to examine whether the model's decisions are consistent across various demographic groups, such as different age brackets, genders, or income levels. This includes checking whether certain groups are unfairly favored or penalized. Conducting fairness assessments can help confirm that the model behaves responsibly and does not introduce unintentional bias in credit evaluation.

**Model monitoring:** A feedback loop is set up to monitor predictions and update the model regularly. We used A/B testing to evaluate the model performance against existing systems.

**Interpretability:** We may utilize Tools such as SHAP or LIME can be used to advance the explainability of the model's predictions. Both tools can help in understanding how individual features contribute to prediction outcomes. This level of transparency is highly significant when the results are used in customer-facing applications, as it provides clear justification for each decision.

## Final Recommendation

Based on the current evaluation metrics, Random Forest is recommended for deployment as the primary model. However, continuous monitoring, feedback integration, and further experimentation with advanced techniques will help refine predictions in real-world environments.

## REFERENCES

- [1] Kaggle. (2023). Customer credit default prediction dataset. Retrieved from <https://www.kaggle.com>
- [2] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). GridSearchCV – Scikit-learn 1.4.2. Scikit-learn.
- [3] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). RobustScaler – Scikit-learn 1.4.2. Scikit-learn.
- [4] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). OneHotEncoder – Scikit-learn 1.4.2. Scikit-learn.
- [5] Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (3rd ed.). Wiley.
- [6] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- [7] Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81–106. <https://doi.org/10.1007/BF00116251>
- [8] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree-boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). ACM. <https://doi.org/10.1145/2939672.2939785>