Creditworthiness Prediction Report

Kavyansh Jain

1 Introduction

This report presents the analysis and modeling of the Statlog (German Credit Data) dataset to predict creditworthiness using a random forest classifier. The dataset, sourced from the UCI Machine Learning Repository, contains 1000 instances with 20 attributes (7 numerical, 13 categorical) and a binary target variable (1 for good credit, 2 for bad credit). The project encompasses comprehensive exploratory data analysis (EDA), preprocessing with outlier handling and feature encoding, model training with cost-sensitive learning, and deployment via a Streamlit web application. A cost matrix prioritizes minimizing misclassifications of bad credits as good (cost=5) over good as bad (cost=1).

2 Data Description

The dataset includes the following attributes:

Type	**Attributes**
Numerical	Duration, Credit Amount, Installment Rate, Residence Since, Age, Existing Credits, Number of Dependents
Categorical	Checking Status, Credit History, Purpose, Savings Status, Employment, Personal Status, Other Parties, Property Magnitude, Other Payment Plans, Housing, Job, Own Telephone, Foreign Worker
Target	Creditworthiness (1=Good, 2=Bad)

Table 1: Dataset Attributes

No missing values were found, ensuring data completeness.

3 Exploratory Data Analysis

3.1 Numerical Variables

- **Credit Amount**: Highly right-skewed (range: 250 to 18424 DM, mean: 3271 DM), with significant outliers indicating large loans.
- **Duration**: Right-skewed (range: 4 to 72 months), suggesting longer loans are associated with higher risk.
- **Age**: Ranges from 19 to 75 years, with younger individuals slightly more likely to have bad credit.

3.2 Categorical Variables

- **Checking Status**: 'A14' (no checking account) is the most frequent (40%) and strongly associated with good credit.
- **Credit History**: 'A34' (critical account/other credits existing) correlates with good credit, likely due to managed debts.
- **Purpose**: Common purposes include 'A43' (radio/television) and 'A40' (new car), with varying creditworthiness associations.

3.3 Target Distribution

The target variable shows 700 good credits (70%) and 300 bad credits (30%), indicating moderate class imbalance. This imbalance necessitates cost-sensitive learning to prioritize correct identification of bad credits, as misclassifying them incurs a higher cost.

3.4 Relationships with Target

- Bad credits are associated with higher credit amounts and longer durations, suggesting riskier loan profiles.
- Checking status 'A11' (<0 DM) has a higher proportion of bad credits (50%).
- Credit history 'A34' shows a higher proportion of good credits, possibly reflecting stable debt management.

3.5 Correlation Analysis

A moderate correlation (0.62) exists between Credit Amount and Duration, expected as larger loans often require longer repayment periods. Other correlations are weak (<0.3), indicating no significant multicollinearity.

3.6 Additional Insight: Class Imbalance Impact

The 70:30 class imbalance (good:bad) could bias the model toward predicting good credit, potentially increasing false negatives (predicting bad credit as good), which are costly per the cost matrix. Class weights in the model mitigate this by penalizing misclassifications of bad credits more heavily, improving recall for the bad credit class.

4 Preprocessing

Preprocessing included:

- **Outlier Handling**: Log transformation was applied to 'credit$_amount$'and'$duration$'$toreduceskewness$
$likedistributionsthatenhancemodelstability.$**Encoding** : $Ordinalencodingwasusedfororordinalvari$
$hotencodingwasappliedtonominalvariables(e.g.,$'$checking_status$', '$purpose$')$$tocreatebinaryfeatu$

- **Target Mapping**: The target was mapped to 0 (good) and 1 (bad) for binary classification.

Additional Insight: Log transformation significantly reduced the impact of extreme values in 'credit$_amount'(e.g., 18424DM)and'duration'(e.g., 72months), asevidencedbymorebalancedhistogran transformation.Thispreprocessingsteplikelyimprovedthemodel'sabilitytogeneralizeacrossdiverseloa

5 Model Training and Evaluation

A random forest classifier was trained within a pipeline that integrates preprocessing and modeling. Class weights {0:1, 1:5} were applied to align with the cost matrix. Hyperparameter tuning was performed using grid search, optimizing for the F1-score to balance precision and recall given the class imbalance.

5.1 Performance Metrics

The model was evaluated on a 20% test set using precision, recall, F1-score, and a cost-based metric derived from the confusion matrix:

	Predicted Good	Predicted Bad
Actual Good	True Negative	False Positive (Cost=1)
Actual Bad	False Negative (Cost=5)	True Positive

Table 2: Confusion Matrix with Costs

Total cost = FP $\times$ 1 + FN $\times$ 5. The F1-score and cost metric reflect the model's ability to minimize costly false negatives.

5.2 Feature Importances

Top features include Checking Status, Log(Credit Amount), and Log(Duration), indicating their strong influence on creditworthiness. Features with importance <0.01 were identified but retained to preserve information.

5.3 Additional Insight: Feature Importance Stability

Feature importance analysis revealed that 'checking$_status'consistentlyrankedhighest, suggestingthatacco

6 Deployment

A Streamlit web application enables users to input financial attributes via dropdowns and numerical fields, using the trained model to predict creditworthiness. The model and preprocessor were exported using 'joblib' to ensure consistent predictions. The app provides probabilities for bad credit, enhancing interpretability.

7 Conclusion

The random forest model effectively predicts creditworthiness, with preprocessing steps ensuring robust handling of outliers and categorical variables. The class weights successfully address the class imbalance, prioritizing the identification of bad credits. The Streamlit app

provides a user-friendly interface for real-time predictions. Future improvements could include threshold tuning to further minimize the total cost and exploring feature selection to reduce model complexity by dropping low-importance features.