Interpretable Model Comparison for Bank Credit Approval Prediction

CENG 562 – Fall 2025

30/11/2025

Oğuzhan ALPERTÜRK 2315752

Arda ÇAVUŞOĞLU 2448249

Yağızcan PANÇAK 2697761

1. Abstract

This project investigates the problem of predicting bank credit approval using classical and interpretable machine learning models. Credit approval is a high-impact financial decision where transparency and fairness are as important as predictive accuracy. We use the UCI Credit Approval Dataset, which contains both numerical and categorical attributes representing applicants' demographic and financial characteristics. Three classical models—Logistic Regression, Decision Tree, and Random Forest—are implemented within unified preprocessing and evaluation pipelines. The models are assessed using 5-fold stratified cross-validation, with accuracy and ROC-AUC as primary metrics. Preliminary findings show that while Random Forest achieves the highest predictive performance, Logistic Regression and Decision Tree offer clear interpretability through coefficients, decision rules, and feature importance analysis. These early results highlight the trade-off between model accuracy and explainability in credit decision contexts.

2. Introduction and Background

Credit approval is one of the most consequential decision-making processes in the financial sector. Banks routinely evaluate applicants based on demographic and financial attributes, attempting to balance profitability with fairness and regulatory compliance. As these decisions increasingly rely on automated systems, ensuring that machine learning models are both accurate and interpretable has become essential. An opaque or overly complex model may lead to distrust, ethical concerns, and difficulty in explaining decisions to customers and auditors. Therefore, understanding how different classical ML models perform and how interpretable their internal decision mechanisms are is critical for responsible AI deployment in finance.

The objective of this project is to analyze the relationship between predictive performance and interpretability across three widely used classical machine learning models: Logistic Regression, Decision Tree, and Random Forest. Specifically, we aim to answer the following questions:

1. How well do these models predict credit approval using heterogeneous tabular data?
2. What are the key features influencing the models' decisions?
3. How does model interpretability differ across linear, rule-based, and ensemble approaches?

Relevant literature provides strong foundations for this comparison. Logistic Regression has long been used in credit scoring due to its interpretability and stability. Decision Trees, introduced by Quinlan (1986), offer transparent decision pathways and human-readable rules. Random Forests (Breiman, 2001) improve predictive accuracy through ensembling while still providing feature importance measures. Recent work on explainability further emphasizes the need to understand model behavior, especially in sensitive decision-making domains.

In this context, our project extends existing approaches by applying all three methods under a unified preprocessing and evaluation framework, enabling a fair and structured comparison. Our work aligns with the proposal's emphasis on understanding the trade-off between accuracy and interpretability, which remains an open and practically relevant challenge in financial ML systems.

3. Data and Methodology

3.1 Data Description

This study uses the UCI Credit Approval Dataset, a widely used benchmark for credit-scoring research. The dataset contains 690 instances, each representing a credit application, and includes 15 heterogeneous attributes (A1–A15) describing demographic, financial, and employment-related features. The target variable is binary, indicating whether the application was *approved* (+) or *rejected* (-). As stated in the original documentation and acknowledged in our proposal attribute names are anonymized to protect sensitive information.

The dataset contains missing values encoded using the "?" symbol. To ensure consistent preprocessing across models, a shared preprocessing module was developed. Missing values are replaced with NaN, after which:

- Numeric features (A2, A3, A8, A11, A14, A15) undergo median imputation.
- Categorical features (A1, A4–A7, A9–A10, A12–A13) undergo most-frequent imputation followed by One-Hot Encoding.

The target variable is mapped from {+, -} to {1, 0}. All models use the same train–test splits internally through cross-validation, ensuring no data leakage. This unified data preparation pipeline guarantees fair comparison and reproducibility across methods.

3.2 Methodology

The task is formulated as a binary classification problem, where the goal is to predict whether a credit application should be approved. Following the project proposal three classical machine learning models were selected for comparative analysis:

1. Logistic Regression – a linear and interpretable baseline model
2. Decision Tree – a rule-based model that provides clear decision paths
3. Random Forest – an ensemble method designed to improve predictive performance while retaining feature importance interpretability

All models were implemented using scikit-learn and embedded in Pipeline objects combining preprocessing and model fitting. This design ensures that preprocessing occurs within each fold during evaluation, preventing information leakage.

Logistic Regression Pipeline

- Preprocessing: median imputation (numeric), most-frequent imputation + one-hot encoding (categorical), and StandardScaler for numeric features
- Model parameters: penalty="l2", solver="liblinear", max_iter=1000
- Interpretability: coefficients and odds ratios extracted after fitting the model on the full dataset

This model serves as the benchmark for interpretability.

Decision Tree Pipeline

- Preprocessing: identical imputation and one-hot encoding steps
- Model parameters: criterion="gini", max_depth=3, random_state=42
- Interpretability: extraction of full decision rules and visualization of the learned tree structure

A shallow depth was intentionally chosen to maintain readability.

Random Forest Pipeline

- Preprocessing: same imputation and one-hot encoding strategy
- Model parameters: n_estimators=200, class_weight="balanced", random_state=42
- Interpretability: feature importance scores and ranked feature lists

Random Forest is expected to outperform single estimators while allowing examination of aggregated feature contributions.

Evaluation Framework

As defined in both the guidelines and the proposal:

- All models are evaluated using 5-fold Stratified Cross-Validation.
- Primary metrics are Accuracy and ROC-AUC.
- Interpretability-specific outputs (coefficients, decision rules, feature importances) are generated for qualitative comparison.
- Visualization tools (tree plots, coefficient bar charts, importance charts) support interpretability analysis.

This methodology enables a controlled, systematic, and fair comparison of classical machine learning models in terms of predictive performance and transparency.

4. Preliminary Results and Discussion

We evaluated all three models using 5-fold stratified cross-validation, following the plan outlined in the proposal.
The performance results are summarized in Table 1.

Model	Mean Accuracy	Std	Mean ROC-AUC
Logistic Regression	0.859	0.036	0.934
Decision Tree	0.820	0.040	0.910
Random Forest	0.871	0.032	0.960

Table 1. Cross-Validation Performance of Models

These results demonstrate several important patterns:

1. Random Forest achieved the highest overall performance, outperforming both Logistic Regression and Decision Tree in accuracy and ROC-AUC. This aligns with expectations from ensemble learning research, where aggregating multiple trees reduces variance and captures nonlinear relationships more effectively.
2. Logistic Regression performed strongly despite its simplicity, reaching a ROC-AUC of 0.934. This indicates that the dataset contains a significant linear signal and that the chosen preprocessing pipeline (median imputation, scaling, and one-hot encoding) suits the model well.
3. Decision Tree produced the lowest accuracy, largely due to its intentionally constrained depth of 3. This limitation was deliberate to preserve interpretability, consistent with the project's objectives. In exchange for reduced complexity, the model provides fully readable decision structures.
4. Interpretability insights revealed complementary strengths across models:
    - Logistic Regression highlighted features such as A15, A9_t, and A6_x as strong linear predictors.
    - Decision Tree splits emphasized attributes like A9, A3, and A15, providing human-readable rules.
    - Random Forest ranked numeric variables (e.g., A3, A11, A15) as highly influential through feature importance scores.
5. No major anomalies or unexpected behaviors were observed. However, the performance gap between Decision Tree and the other two models suggests that future work may explore deeper or pruned trees, cost-complexity regularization, and better hyperparameter optimization.

Overall, the preliminary experiments support the project's central theme: understanding the trade-off between predictive performance and model interpretability. Random Forest offers the best accuracy, while Decision Tree and Logistic Regression provide clearer explanations, each in a different form.

5. Next Steps

Several tasks remain to be completed before the final report and presentation. These steps aim to strengthen both the experimental depth and the interpretability analysis of the models.

5.1 Hyperparameter Tuning

We will perform systematic tuning for all three models to improve performance and reduce variance. This includes:

- Logistic Regression: regularization strength (C), penalty variations
- Decision Tree: tree depth, minimum samples per split/leaf, cost-complexity pruning
- Random Forest: number of trees, max depth, max features, bootstrap strategy

5.2 Additional Evaluation Metrics

We plan to extend the evaluation with precision, recall, F1-score, confusion matrices, and calibration curves to provide a more comprehensive analysis of misclassification behavior.

5.3 Enhanced Interpretability Analysis

Following the interpretability goals stated in the proposal, we will:

- Visualize logistic regression coefficients and odds ratios in more detail
- Extract and analyze decision paths for deeper or pruned trees
- Compute permutation feature importance for the Random Forest model
- Investigate local explanations (e.g., LIME or SHAP) where appropriate

5.4 Robustness and Sensitivity Experiments

We will examine model behavior under data perturbation, such as noise injection, missing value scenarios, and shuffled categories, to evaluate stability and generalization.

5.5 Final Report Preparation

We will consolidate all analyses into the final 6–10 page report, incorporating:

- Full experimental results
- Refined visualizations (coefficient plots, tree diagrams, feature importance charts)
- Comparative discussion on the performance–interpretability trade-off
- Clear conclusions and recommendations for credit approval systems

These planned steps align with the project's initial work-plan and will ensure a thorough and well-supported final submission.

6. References

Quinlan, J. (1987). Credit Approval [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5FS30.

Quinlan, J. R. “Induction of Decision Trees.” Machine Learning, 1986.

Breiman, L. “Random Forests.” Machine Learning, 2001.

Ribeiro, M. T. et al. “Why Should I Trust You?” KDD, 2016.

Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*, 357–362. https://doi.org/10.1038/s41586-020-2649-2

McKinney, W., & others. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56).

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95.