XAI Case Study: German Credit Risk

José Ridruejo Tuñón

1 Introduction and Objectives

For my final project, I chose the German Credit dataset. The goal is to train a machine learning model that can predict whether a customer is ”good” or ”bad” in terms of the risk they pose to the bank, and for the model to explain the reasons behind each decision. I chose this specific dataset because it’s a typical example of tabular data where hidden biases are often found, making it perfect for this case study and allowing me to propose improvements to the baseline model. The stakeholder would be a bank, specifically its lending division. It’s a high-risk decision, as rejecting a loan significantly impacts people’s lives. If the model answers ”No,” the bank must explain the reason to the customer. Real-world regulations also require these explanations. Therefore, I built a model and then applied various XAI techniques to explain it, so it’s not just a black box.

2 Part 1: Data Analysis and Preprocessing

Before starting with the model, I analyzed the data in the first notebook, `Preprocessing.ipynb`. I loaded the dataset from the UCI repository and first assigned the target variable to 0 and 1 to standardize it. Then I plotted the target distribution, and it’s clear that the distributions are unbalanced because there are many more ”Good” loans (around 70%) than ”Bad” ones (30%). This could be a problem since the model might simply learn to predict ”Good” loans consistently. To address this, I decided to split the data into three sets: Training (60%), Validation (20%), and Test (20%). I used stratified splitting to ensure that all three sets had exactly the same proportion of ”Bad” loans as the original data. I also plotted histograms of numerical characteristics such as loan amount, loan term, and age. I think these might also be skewed because there are many small loans and few very large ones.

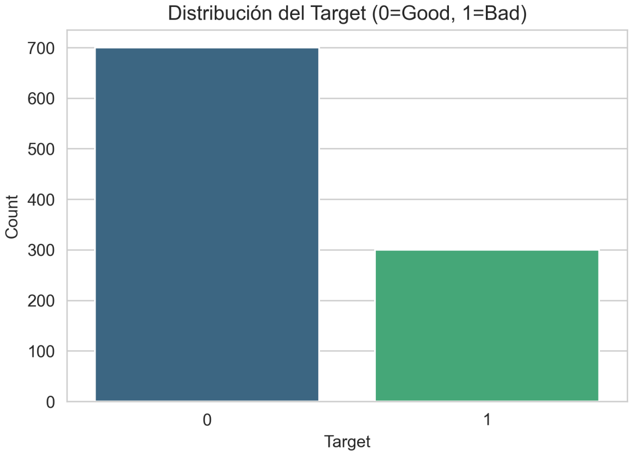


Figure 1: **Class Balance.** There are many more ’Good’ credits than ’Bad’ ones.

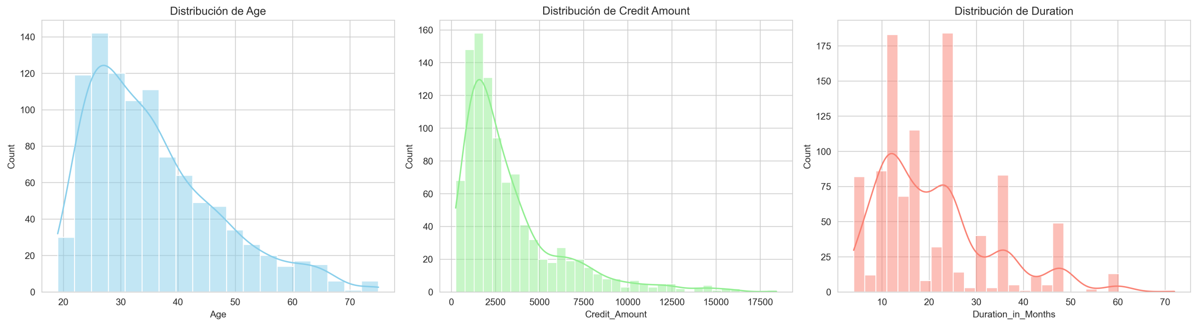


Figure 2: **Distributions.** Duration and Amount are skewed to the right.

3 Part 2: Baseline Model

In the second notebook, `Modeling.ipynb`, I programmed the machine learning model. I had to create a preprocessing pipeline with 'ColumnTransformer' so the model could process the data correctly. For the numerical variables, I used 'SimpleImputer' to fill in missing values with the median and 'StandardScaler' to normalize the range to improve convergence. For the categorical variables, 'OneHotEncoder' converts them to a format the model can understand (0 and 1). I chose a Random Forest Classifier as the model because it is very robust and works well with tabular data. It builds many decision trees and averages their predictions, achieving nonlinearities without overfitting (which would happen with a tree) and without requiring excessive hyperparameter tuning. This allowed me to quickly modify the parameters in Model Improvement since there were only two.

3.1 Results

After training the model on the training set I evaluated it on the validation set to see how well it generalizes.

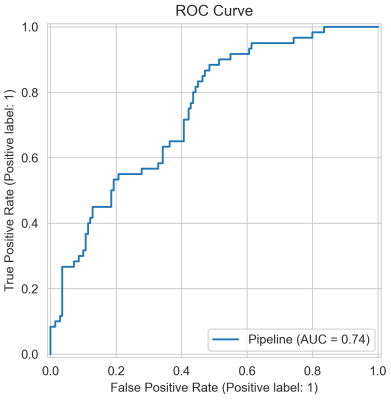


Figure 3: **ROC Curve.** The model achieves an AUC of 0.74.

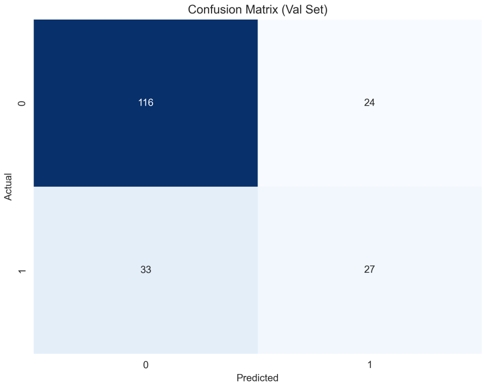


Figure 4: **Confusion Matrix.** Visualizing the prediction errors.

The model achieved a ROC-AUC score of 0.74, as shown in Figure 3, demonstrating a good ability to distinguish between positive and negative credit risks and is definitely better than random sampling. The Confusion Matrix in Figure 4 shows that while accuracy for good credits is high, recovery for bad credits is lower, meaning the model still fails to identify a significant number of risky customers (false negatives). This is important in credit risk assessment, as lending money to someone who will not repay is usually more expensive than denying a loan to a good customer. However, as a baseline model, this performance is good enough for the explainability analysis.

4 Part 3: Explainability (XAI)

`XAI.ipynb` contains the core of the project where XAI techniques are applied. To explain the black box model, two types of methods were used: global and local explanation methods.

4.1 Global Explanations

To begin, I analyzed the global importance of the features using Permutation Feature Importance (PFI), which involves altering one variable at a time and measuring the decrease in model performance. As shown in Figure 5, the checking and savings account balances and the loan term were the most important factors. This makes sense since financial stability and loan term are fundamental to risk. I also tested Partial Dependence Plots (PDPs). Figure 6 shows that as the term in months increases, the expected risk also increases, thus confirming that long-term loans are generally riskier for the bank.

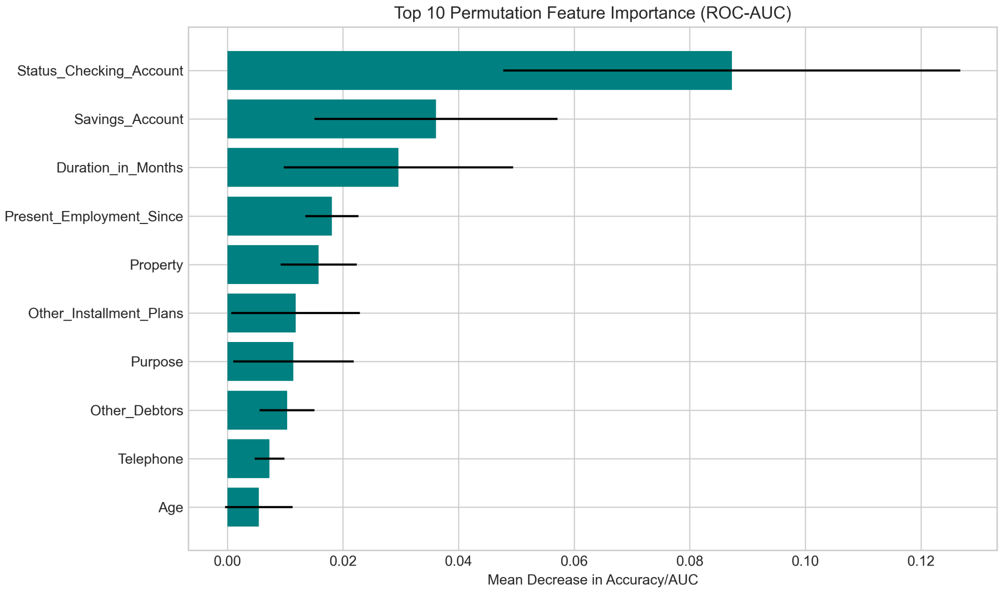


Figure 5: Permutation Feature Importance identifying key variables.

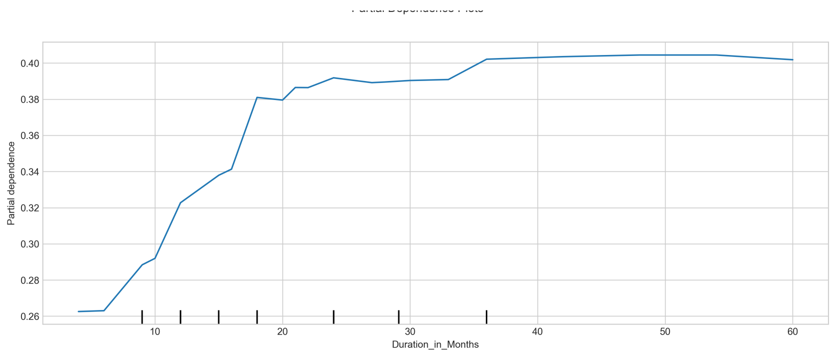
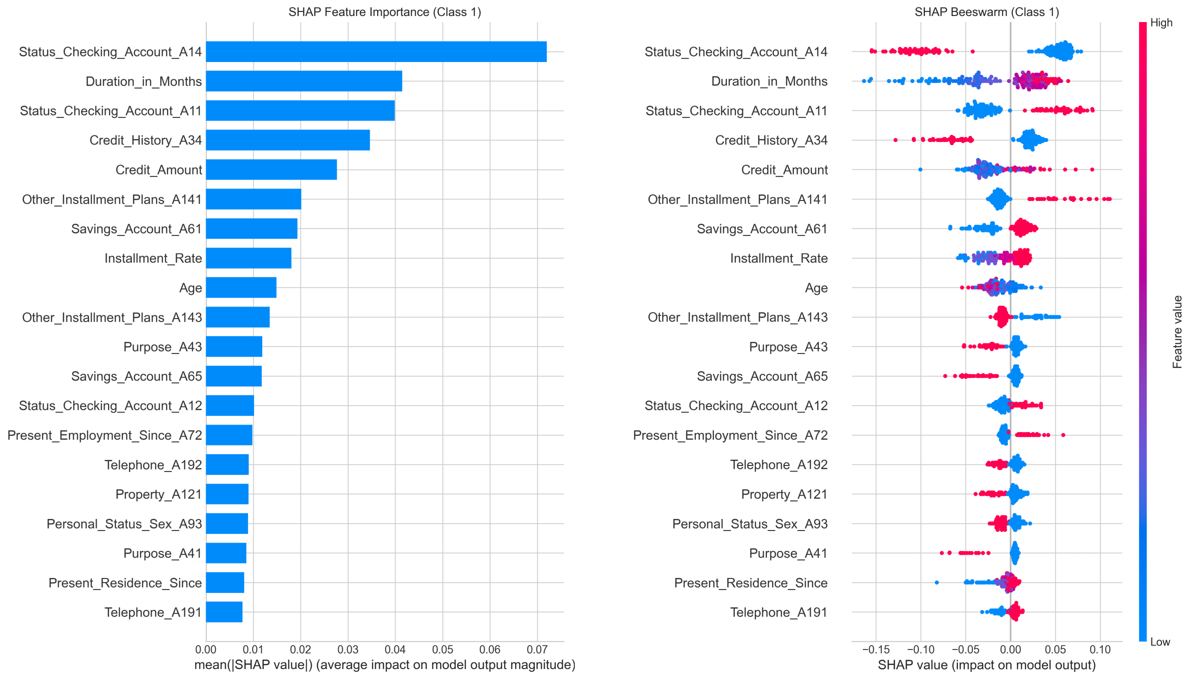


Figure 6: PDP Plot showing risk increasing with loan duration.

To test more complex algorithms, I applied SHAP (Shapley Additive Explanations), which assigns a contribution value to each feature based on game theory. The beeswarm plot in Figure 7 is a comprehensive summary where I can observe the impact of all features simultaneously. It shows that customers with low balances in their checking accounts make the model produce a riskier outcome, while those with savings reduce it. Although this conclusion is similar to PFI, it adds a directional value for each feature.



(a) **Global importance (bar).**

(b) **Beeswarm.**

Figure 7: **Global SHAP explanations**.

4.2 Local Explanations

Regarding local methods, these allow us to see individual predictions to demonstrate how the bank might explain a customer's rejection. I chose a specific high-risk applicant, let's call him Bob, to analyze his particular situation, and using the SHAP waterfall plot in Figure 8, I was able to see how each characteristic contributed to the final score. For Bob, it appears that what made him high-risk was the amount of credit requested, the long loan term, and the lack of savings. I also verified this result with LIME (Locally Interpretable and Model-Agnostic Explanations), which fits a simple linear model locally around the prediction. Figure 9 confirms the conclusion, although it adds importance to loan history, but overall, both methods agree on the main reasons for the negative result. A combination of these two metrics could be used in a real-world setting, and the variables that appear in SHAP and LIME would be the most reliable.

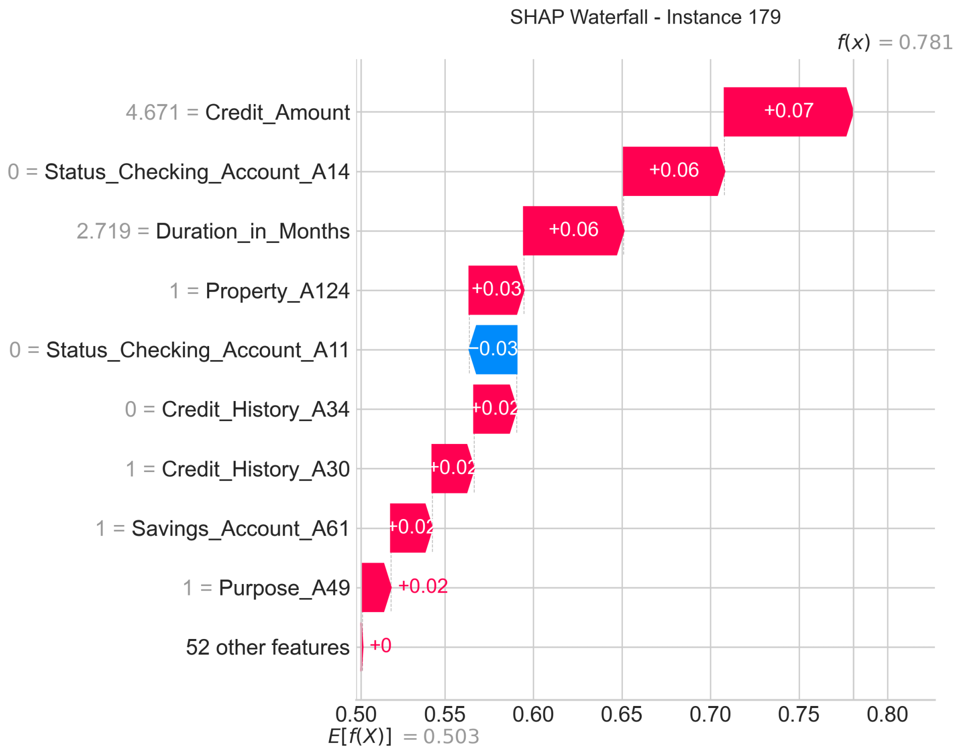


Figure 8: SHAP Waterfall explaining a specific decision.

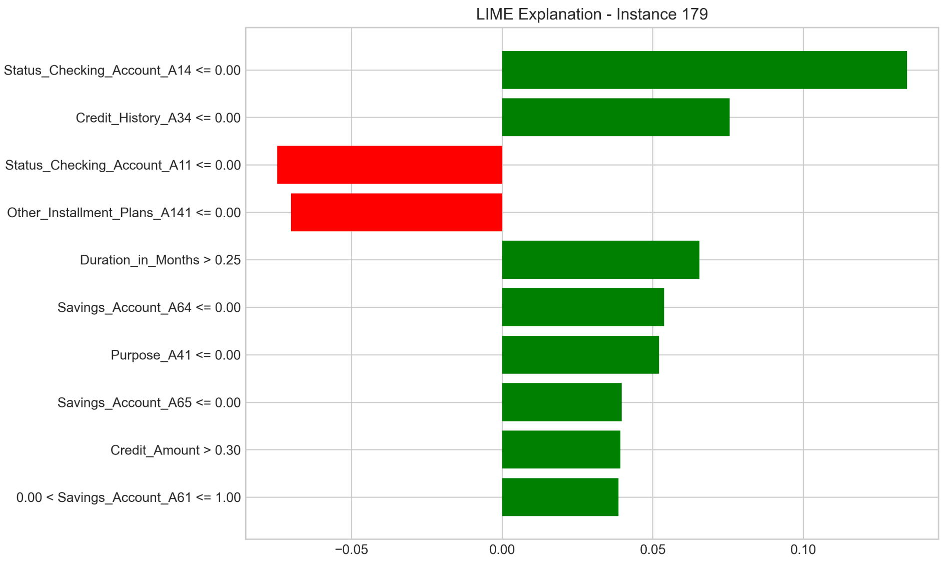


Figure 9: LIME explanation for the same instance.

5 Part 4: Fairness and Model Improvement

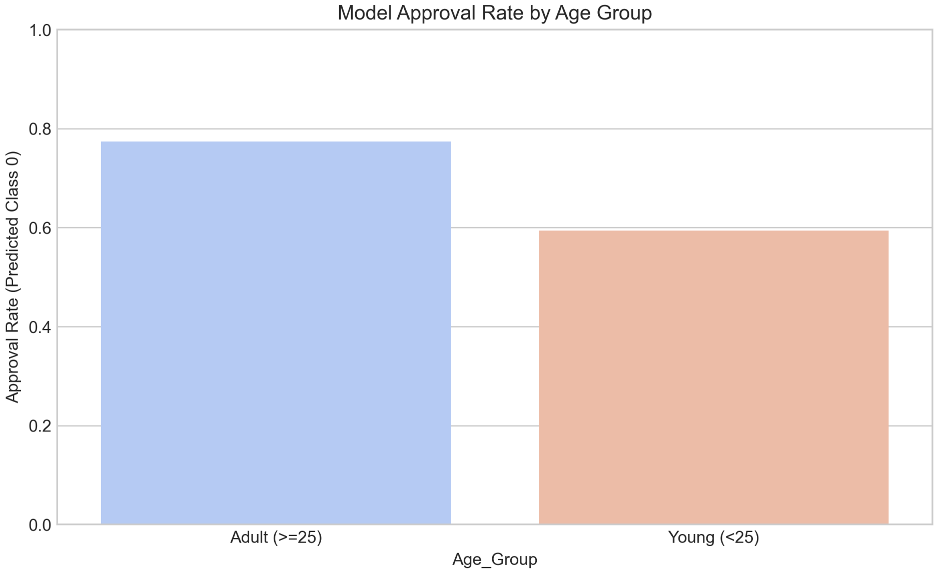


Figure 10: Age disparity in loan approvals.

Finally, in `Accionability.ipynb`, I focused on the fairness of the model's results depending on age, which can be considered a sensitive attribute. I divided the dataset into young applicants (under 25 years old) and adults (over 25 years old) for comparison. The results showed that the model approves fewer young applicants. While it clearly seems that the model has a bias, this could also be due to the smaller amount of data for young people, making the results more opaque. Even so, I calculated the error rates and found that the false positive rate for young people is almost double that of adults. This may mean that the model often incorrectly classifies good young clients as bad risks, likely penalizing them for their lack of credit history when, in these cases, other attributes such as poor financial behavior should be considered.

Age Group	**FPR**	**FNR**	**Count**
Adult (Over 25)	0.1545	0.5778	168
Young (Under 25)	0.2941	0.4667	32

Table 1: Error rates by age group.

To improve the model, or at least try to remove the bias (although this doesn't necessarily mean a complete improvement, I believe it's more important), I used the information obtained from the Permutation Feature Importance (PFI) analysis and the SHAP analysis. If the 'Age' variable introduced unnecessary noise and unfairness, it was best to remove it, which is what I did. However, I also wanted to optimize the model: I focused on the 10 most important features from my PFI analysis. Then I trained a simplified Random Forest with these selected features and reduced the number of estimators from 100 to 35. The results were quite good: this new model, in addition to being more efficient and eliminating the direct use of age, also outperformed the base model. It achieved an accuracy of 74.5% (compared to 71.5% for the base model) and an ROC-AUC score of 0.6940 (compared to 0.6393). Although the model improvement ends here, it would be interesting to thoroughly check that the 'Age' variable is not being inferred in some way in the other variables, although by reducing the number of features to 10, the probability is low.

References

[1] Ahmet Yalcin. *Credit Risk: Predict and Explain by XAI Algorithms.* Kaggle. Available at: `https://www.kaggle.com/code/ahmetyalcinn/credit-risk-predict-and-explain-by-xai-algorithms`.