Credit Default Risk Modeling – Final Project Report

1. Project Overview

Objective
Build a reliable credit risk model using application-stage data to **identify and rank customers by default risk**. The focus is not on raw accuracy, but on producing a **stable, interpretable risk score** suitable for business decisioning.

Key Challenge
- Highly imbalanced target (~8% defaulters, ~92% non-defaulters)
- Large number of raw features (120+) with noise and missing values
- Application-only data (no behavioral or transactional history)

---

2. Understanding the Problem as a Risk-Ranking Task

Rather than treating this as a simple classification problem (default vs non-default), the problem was framed as:

> *Ranking customers from highest risk to lowest risk.*

This framing guided: - Metric selection (ROC-AUC over accuracy) - Evaluation strategy (deciles instead of single thresholds) - Model choice (stability and interpretability over complexity)

---

3. Data Exploration & Feature Structuring

The raw dataset was decomposed into **conceptual feature groups** to understand economic meaning and reduce noise:

- Identity & Demographics
- Income & Repayment Capacity
- Employment & Stability
- Assets & Wealth
- Credit History & External Risk
- Contactability & Digital Stability
- Location & Address Consistency
- Application Behavior & Timing

Each group was analyzed independently to: - Study distributions and missingness - Identify weak vs strong signals - Decide whether to keep, transform, aggregate, or drop features

This avoided blind modeling and ensured domain-aligned decisions.

---

4. Feature Engineering (Core Value Addition)

4.1 Ratio Features (High Impact)

Explicit economic ratios were created: - **Repayment Burden Ratio** = AMT_ANNUITY / AMT_INCOME_TOTAL
- **Credit-to-Income Ratio** = AMT_CREDIT / AMT_INCOME_TOTAL
- **Loan-to-Goods Value Ratio** = AMT_CREDIT / AMT_GOODS_PRICE

These ratios: - Linearized risk relationships - Captured affordability and leverage directly - Significantly improved separability for linear models

4.2 Aggregated Signals (Noise Reduction)

Several weak binary indicators were consolidated: - `contact_count` → summarized multiple phone/email flags - `location_mismatch_count` → summarized address/work inconsistencies - `assets_group` → combined car and real-estate ownership - `social_60_status` → summarized delinquency exposure in social circle

This reduced dimensionality while strengthening signal.

4.3 Feature Selection

After EDA and engineering: - Final feature set reduced to **~44 high-signal variables** - Redundant housing micro-features and noisy flags removed - Resulting feature space was compact, interpretable, and stable

---

5. Preprocessing Pipeline

A clean, leakage-free preprocessing pipeline was implemented:

- **Numerical features**: median imputation + standard scaling
- **Categorical features**: mode imputation + one-hot encoding
- All preprocessing performed **inside an sklearn Pipeline**

This ensured: - Proper train/test separation - Reproducibility - Production-ready structure

---

6. Model Selection

Baseline Model: Logistic Regression

Chosen because: - Strong for monotonic tabular data - Stable under regularization - Interpretable - Widely accepted in credit risk

Class imbalance handled via `class_weight='balanced'`.

---

7. Model Evaluation Strategy

7.1 Step 1 – Cross-Validation (Stability Check)

- 5-fold Stratified CV on training data
- Mean ROC-AUC ≈ **0.744**
- Std ≈ **0.0018**

Interpretation: - Performance is stable - No overfitting - Results are not due to lucky splits

---

7.2 Step 2 – Threshold Analysis

Different probability thresholds were evaluated to study precision–recall trade-offs.

Key findings: - Recall remains very high even at higher thresholds - Precision increases slowly - No sharp cutoff between defaulters and non-defaulters

Conclusion: - Single global threshold is inappropriate - Model should not be used as a hard classifier

---

7.3 Step 3 – Risk Decile (Bucket) Analysis (Most Important)

Customers were ranked by predicted risk and split into 10 equal-sized deciles.

Observed default rates:

- Top decile: **~26% default rate**
- Bottom decile: **~1% default rate**
- Baseline: ~8%

Key Observations: - Perfect monotonic decrease across deciles - ~3× risk lift in top decile - ~8× safer bottom decile

This confirms the model's effectiveness as a **risk-ranking tool**.

---

8. Model Comparison

A Random Forest model was trained for comparison.

- ROC-AUC ≈ **0.744** (similar to logistic regression)

Conclusion: - No meaningful non-linear gain - Engineered features already capture core signal - Logistic regression is the appropriate final model

---

9. Final Conclusions

- EDA and feature engineering materially improved model quality
- Logistic regression provides stable and interpretable risk ranking
- Risk decile analysis demonstrates strong business value
- Model is suitable for application-stage credit decisioning

  The model’s value lies in ranking customers by risk, not in binary prediction accuracy.

---

10. What I Would Do Next in Production

- Add missing-value indicator features
- Monitor population stability over time
- Periodically recalibrate score thresholds
- Integrate business cost functions

What I Learned From This Project

1. Problem Framing Matters More Than Algorithms

- Credit risk is a ranking problem, not a classification problem
- Accuracy can be actively misleading in imbalanced settings

2. Feature Engineering Beats Model Complexity

- Ratio features can outperform complex models
- Aggregating weak signals improves stability

3. Evaluation Must Match Business Reality

- ROC-AUC for ranking quality
- Deciles for business usefulness
- Thresholds are policy decisions, not model decisions

4. Stability Is as Important as Performance

- Low CV variance is critical in regulated domains

- A slightly lower but stable AUC is preferable

5. Simpler Models Can Be Better

- Logistic regression matched Random Forest performance
- Interpretability and stability often outweigh marginal gains

6. End-to-End Thinking Is Essential

- EDA, modeling, and evaluation are tightly connected
- Skipping early steps leads to weak conclusions later

Final Takeaway
This project demonstrates a full-cycle, production-style credit risk modeling workflow, emphasizing domain understanding, feature engineering, stability, and business-aligned evaluation over brute-force modeling.