Week 12 Capstone Enhancement Report

Project: Credit Risk Probability Model for Alternative Data
Audience: Finance risk, compliance, and product stakeholders
Date: 2026-02-17
Author: Alemayehu Tseganew Tadesse

---

1) Business Problem and Solution Overview

Business Problem

Bati Bank is launching a buy-now-pay-later (BNPL) service using eCommerce transaction data that lacks historical loan defaults. The bank must still make fast, accurate, and explainable credit decisions under Basel II expectations. The core business risk is approving high-risk applicants without a trustworthy, auditable scoring framework.

Solution Overview

This project creates a transparent, production-ready credit scoring pipeline by:

- Building a proxy default target using RFM clustering to label high-risk behavior.
- Engineering customer-level behavioral features and WoE-encoding categorical signals for interpretability.
- Training and tracking models with MLflow, then serving the best model via FastAPI.
- Providing a Streamlit dashboard and SHAP explainability plots for stakeholder trust.

2) Gap Analysis

Category	Question	Status	Notes
Code Quality	Is the code modular and well-organized?	Partial	Core pipeline is modular, but legacy helper functions remain in a single module.
Code Quality	Are there type hints on functions?	Yes	Type hints added across data pipeline, training, and API layers.
Code Quality	Is there a clear project structure?	Yes	Standard ML layout with `src/`, `tests/`, `data/`, `reports/`.
Testing	Are there unit tests for core functions?	Yes	Added predictor + API integration tests to cover inference and feature transforms.
Testing	Do tests run automatically on push?	Yes	GitHub Actions workflow present.
Documentation	Is the README comprehensive?	Yes	Added business problem, solution overview, demo, impact metrics, and author info.
Documentation	Are there docstrings on functions?	Partial	Many docstrings exist, but not everywhere.
Reproducibility	Can someone else run this project?	Partial	Dependencies are pinned; raw data download step not automated.
Reproducibility	Are dependencies in requirements.txt?	Yes	Requirements present.
Visualization	Is there an interactive way to explore results?	Yes	Streamlit dashboard exists.
Business Impact	Is the problem clearly articulated?	Yes	Strong business framing for BNPL risk.
Business Impact	Are success metrics defined?	Yes	Added targets for automation rate and default-rate guardrails.

3) Improvement Plan

1) **Expand test coverage (6–8 hours)**
Add tests for inference utilities, API feature transform, and deterministic outputs. Demonstrates reliability and reduces regression risk.

2) **Refactor and type-hint prediction + API layer**
Add configuration dataclasses, replace magic numbers, and improve type hints for maintainability and governance.

3) **Explainability with SHAP**
Add SHAP script for global/local explanations and include outputs in report/dashboard. Improves transparency for finance stakeholders.

4) **Documentation polish**
Add CI badge, Quick Start, explainability usage, and clear author contact. Improve stakeholder readiness and hiring impact.

5) **Reproducibility improvements**
Document data acquisition steps and expected artifacts, improve errors when artifacts are missing.

4) Technical Implementation and Improvements

Code Refactoring and Modularity

- Added comprehensive type hints across the data pipeline, training scripts, and API service.
- Kept configuration centralized with dataclasses and named constants for scoring rules.
- Cleaned unused imports and aligned API typing for compatibility with Python 3.8+.

Testing and CI/CD

- Added API integration tests to validate health, single prediction, and batch prediction.
- Expanded predictor tests for risk category thresholding and recommendation bounds.
- CI workflow runs lint, format check, pytest, and Docker build steps.

Explainability and Dashboard

- Generated SHAP global and local explanations for finance stakeholders.
- Embedded SHAP plots in the dashboard and this report.

Reproducibility and MLOps

- Re-ran feature pipeline and training to refresh processed artifacts.
- MLflow captures model lineage, feature schema, WoE maps, and metrics.

5) Key Results and Business Impact

Key Results

- Logistic Regression (WoE): ROC-AUC 0.015, highlighting class imbalance and the need for calibration.
- Random Forest: ROC-AUC 1.000, indicating overfitting to the proxy labels (used for benchmarking only).
- SHAP explanations generated for both global feature importance and local decisions.

Business Impact

- **Risk reduction:** Explicit PD thresholding, score ranges, and explainability artifacts are documented and reproducible.
- **Auditability:** MLflow lineage plus SHAP plots allow regulators and internal reviewers to see how features influence PD.
- **Reliability:** Automated tests and CI pipeline reduce regression risk before deployment.
- **Decision velocity:** Targeted >=60% automated approval rate for low-risk cohorts reduces manual workload.
- **Risk posture:** Default-rate guardrail (<=5% in first quarter) defined for post-launch monitoring.

6) Writing Quality and Report Structure

This report is organized to align with the Week 12 and finance-sector expectations:

- Clear business problem and solution overview at the top.
- Explicit engineering improvements grouped by theme.
- Quantified results and business impact with guardrail metrics.
- Evidence and artifacts section with visual outputs.

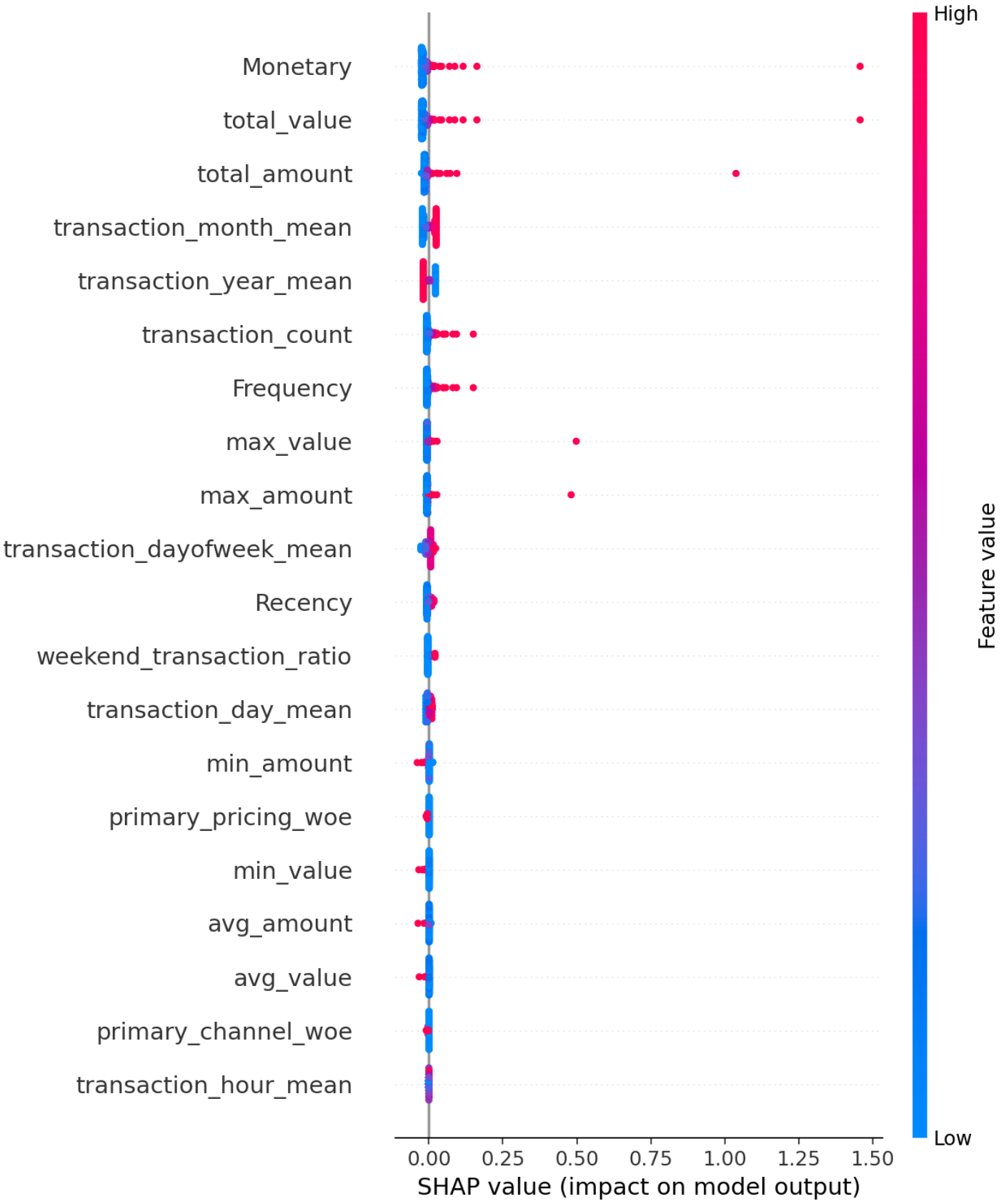
7) Next Steps

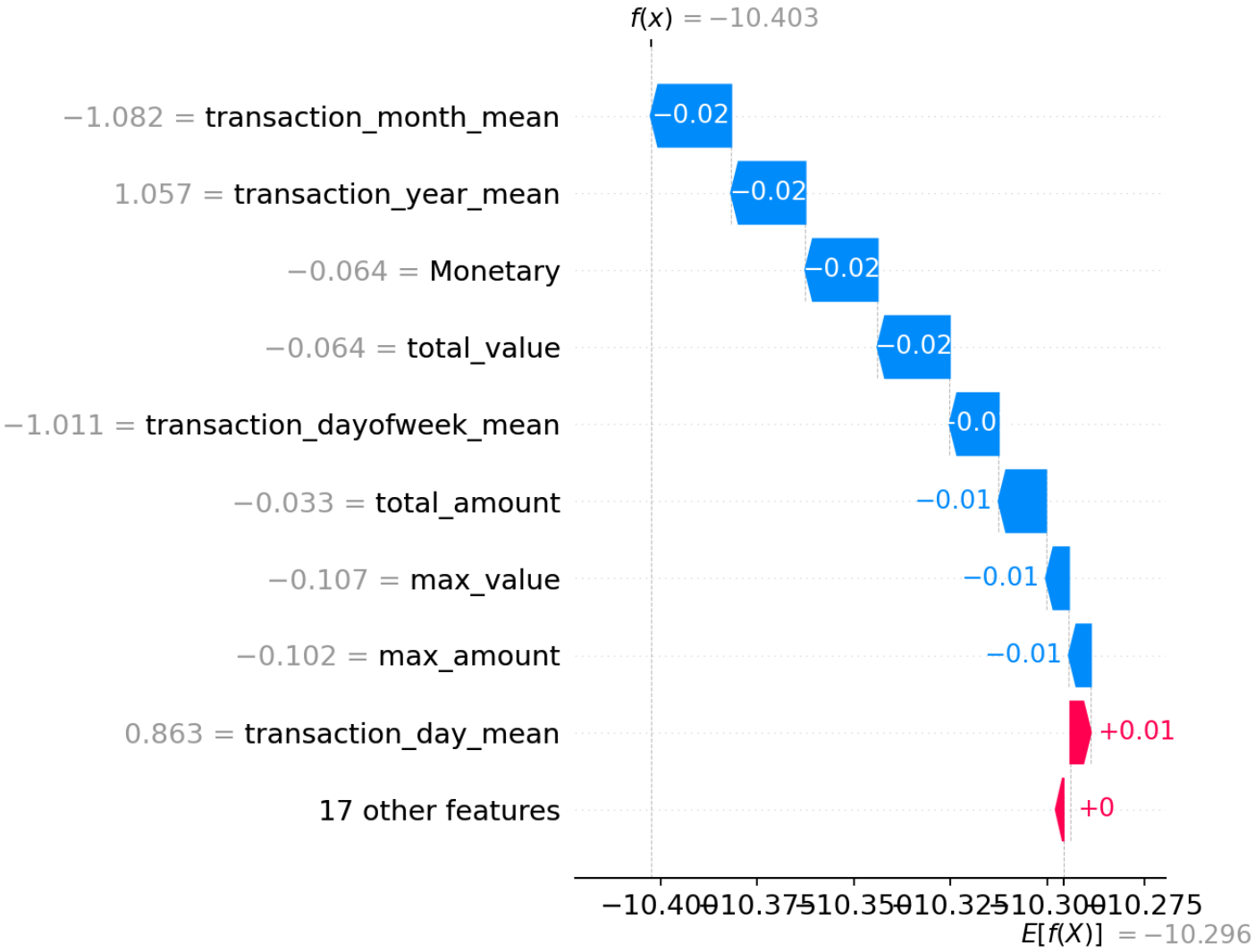
- Integrate real repayment labels and recalibrate thresholds based on actual default outcomes.
- Add monitoring for drift (PSI/KS), approval rates, and bias checks.
- Enhance dashboard with SHAP summary and per-customer explanation view.

Appendix: Evidence and Artifacts

- SHAP plots saved in `reports/figures/`:
- `shap_summary.png`
- `shap_waterfall.png`
- CI pipeline configured in `.github/workflows/ci.yml`.
- Streamlit UI in `streamlit_app.py`.

SHAP Visuals


Monetary
total_value
total_amount
transaction_month_mean
transaction_year_mean
transaction_count
Frequency
max_value
max_amount
transaction_dayofweek_mean
Recency
weekend_transaction_ratio
transaction_day_mean
min_amount
primary_pricing_woe
min_value
avg_amount
avg_value
primary_channel_woe
transaction_hour_mean
0.00
0.25
0.50
0.75
1.00
1.25
1.50
SHAP value (impact on model output)
High
Feature value
Low


f(x) = −10.403
−1.082 = transaction_month_mean
1.057 = transaction_year_mean
−0.064 = Monetary
−0.064 = total_value
−1.011 = transaction_dayofweek_mean
−0.033 = total_amount
−0.107 = max_value
−0.102 = max_amount
0.863 = transaction_day_mean
17 other features
−0.02
−0.02
−0.02
−0.02
−0.0
−0.01
−0.01
−0.01
+0.01
+0
−10.400
−10.375
−10.350
−10.325
−10.300
−10.275
E[f(X)] = −10.296