🏦 Bati Bank – Credit Risk Probability Model

Final Report
Prepared by: Yamlak Negash
Date: July 1, 2025
Program: 10 Academy – B5W5
Project: End-to-End Credit Scoring with Alternative Data

---

1. Executive Summary

This project addresses Bati Bank's initiative to offer a **Buy-Now-Pay-Later (BNPL)** service in partnership with an eCommerce platform. To manage risk associated with this new service, a machine learning-based **Credit Risk Probability Model** was developed to estimate customer risk using behavioral transaction data (Recency, Frequency, and Monetary patterns).

The goal was to build an interpretable, reproducible, and scalable pipeline to assess loan eligibility and suggest optimal lending conditions using alternative data, aligned with **Basel II Capital Accord** principles.

---

2. Business Context & Problem Framing

Basel II Compliance:
The Basel II Accord emphasizes **quantitative risk assessment** and **regulatory transparency**, requiring models to be interpretable, explainable, and well-audited. Hence, we carefully selected both interpretable and performant models.

Proxy Labeling Approach:
Due to the absence of a direct 'default' label, we engineered a binary proxy variable using **RFM-based segmentation** and unsupervised clustering to categorize customers as **high-risk (1)** or **low-risk (0)**.

Business Objective:
Build a model that can:

- Predict customer risk (creditworthiness)
- Return a risk probability score

- Help determine loan amount and duration
- Integrate with a live system via API

---

3. Technical Methodology

a. Data Exploration (EDA)

- **Source:** Xente eCommerce transactions
- **Size:** ~100,000 records, 17 columns
- **Analysis:**
    - High skewness in transaction amounts
    - Seasonality patterns in transaction time
    - Missing values handled using imputation strategies
    - FraudResult and Amount distributions revealed useful business trends

b. Feature Engineering

Developed using `sklearn.pipeline.Pipeline` in `src/data_processing.py`, including:

- Aggregate Features: total value, frequency, average amount
- Date Features: transaction hour, day, month
- Encodings: one-hot for channel/product category
- Scaling: standardization for numerical features
- Feature Selection: based on Weight of Evidence (WOE) and IV

c. Proxy Label Generation

Used **KMeans Clustering** on scaled RFM values to assign proxy labels:

- Cluster with lowest frequency and monetary value marked as **High Risk**
- Merged cluster labels with processed data as `is_high_risk`

---

4. Model Training & Evaluation

Models Trained:

- Logistic Regression (baseline, interpretable)
- Random Forest Classifier
- Gradient Boosting Classifier (best performing)

Evaluation Metrics:

Model	Accuracy	Precision	Recall	F1 Score	ROC-AUC
Logistic Regression	0.79	0.76	0.71	0.73	0.80
Random Forest	0.83	0.82	0.75	0.78	0.86
Gradient Boosting	**0.86**	**0.85**	**0.81**	**0.83**	**0.89**

Best model registered with **MLflow Model Registry**.

---

5. Deployment

Deployed the model using a RESTful API built with **FastAPI**:

- `/predict` endpoint returns credit risk probability
- Input validated using **Pydantic**
- Deployed locally using **Docker**

CI/CD Pipeline:

- GitHub Actions CI triggered on every push
- Runs:
    - `flake8` for linting
    - `pytest` for unit tests in `tests/test_data_processing.py`

---

6. Challenges & Mitigations

Challenge	Resolution
No default label	Used unsupervised proxy via RFM
Class imbalance	Stratified splitting, AUC as metric
Interpretability vs performance	Trained interpretable and tree-based models
Scaling to real-time use	Built FastAPI container with Docker

---

7. Key Takeaways & Future Recommendations

- **Proxy-based labeling** can be an effective workaround for real-world missing labels.
- A **hybrid model stack** (Logistic + GBT) balances regulation and performance.
- Future work:
    - Explore SHAP for deeper model interpretability
    - Integrate external credit bureau data
    - Launch A/B test with live customer segments

---

8. Appendix

- **Repository:** github.com/yamneg96/credit-risk-model
- **Artifacts:**
    - Notebooks: `EDA`, `Feature Engineering`, `Training`, `Deployment`
    - Scripts: `train.py`, `predict.py`, `api/main.py`
    - Dockerfiles and CI/CD YAML
- **Authors:** Yamlak Negash, AI Trainee @ 10 Academy
