Interim Report: Credit Risk Probability Model

🤦 Prepared by: Yamlak Negash

Date: June 29, 2025

mail: Organization: 10 Academy – Bati Bank Project

🧠 Business Context & Problem Understanding

Bati Bank is collaborating with a leading eCommerce company to launch a **Buy Now, Pay Later** service. To assess creditworthiness in real-time and support automated decision-making, a **Credit Risk Probability Model** is needed.

As mandated by the **Basel II Accord**, which emphasizes advanced risk measurement and regulatory transparency, the model must be interpretable and auditable. Given the lack of direct default indicators in the dataset, the team created a **proxy variable** based on user behavior using **RFM** (**Recency, Frequency, Monetary**) metrics.

The credit scoring system must balance:

- Simplicity and transparency (Logistic Regression with Weight-of-Evidence for audits)
- Performance (e.g., Gradient Boosting for improved accuracy)

A compliant, scalable risk scoring solution is central to enabling risk-aware credit offerings.

Technical Progress Summary

▼ Task 1: Credit Risk Business Foundations

- Studied Basel II regulatory guidance and practical implications for model governance.
- Defined the model's role in supporting BNPL services and regulatory compliance.
- Drafted summary section in README.md.

▼ Task 2: Exploratory Data Analysis (EDA)

• Loaded and explored transactions.csv dataset using Jupyter Notebook.

- Reviewed data structure, identified missing values, skewness, and outliers.
- Performed univariate & bivariate analysis.
- Tools Used: pandas, seaborn, matplotlib, numpy

Top Insights:

- 1. High skew in transaction Amount and Value.
- 2. Mobile channels (especially Android) dominate customer interactions.
- 3. Peak activity hours between 11AM–2PM.
- 4. Missing entries in timestamps and some categorical fields.
- 5. Fraudulent transactions are rare, <1% of dataset.

▼ Task 3: Feature Engineering

- Wrote reusable preprocessing script (src/data_processing.py).
- Implemented sklearn.Pipeline and ColumnTransformer for:
 - Date-based features: transaction_hour, transaction_month
 - Aggregations: average transaction, frequency per customer
 - Encoding: One-Hot for categorical, label encoding for others
 - Scaling: StandardScaler for numerical
 - Imputation: SimpleImputer for missing fields

▼ Task 4: Proxy Risk Target Engineering

- Created RFM features for every customer:
 - **Recency:** Days since last transaction

- Frequency: Number of transactions
- Monetary: Sum of transactions
- Applied **K-Means Clustering** (3 clusters, fixed random_state)
- Assigned customers in the "low frequency/low value" cluster as is_high_risk = 1
- Appended the target to the processed dataset

🧪 Task 5: Unit Testing

- Created tests/test_data_processing.py
- Added test for correct extraction of transaction_hour feature
- Planned tests:
 - o Pipeline step execution validation
 - Label generation logic edge cases

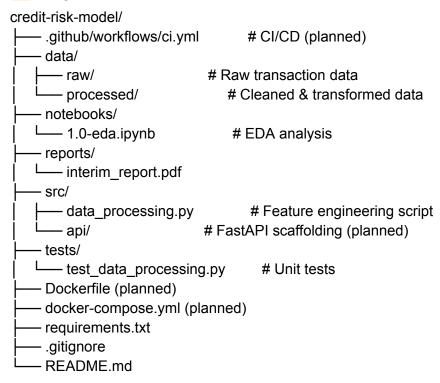
* Tools & Technologies

- Languages: Python
- Frameworks: scikit-learn, pandas, numpy, matplotlib, seaborn
- **Tools**: Jupyter Notebook, GitHub
- Planned: FastAPI, MLflow, pytest, Docker, GitHub Actions

Key Commands Used

jupyter notebook notebooks/1.0-eda.ipynb # Run EDA
python src/data_processing.py # Feature engineering
pytest tests/ # Unit tests

Project Structure (As of Interim Submission)



✓ Visual Insights

Included in EDA Notebook:

- Distribution plots of transaction amounts
- Heatmap of feature correlations
- Bar plots of transaction counts by product category and device