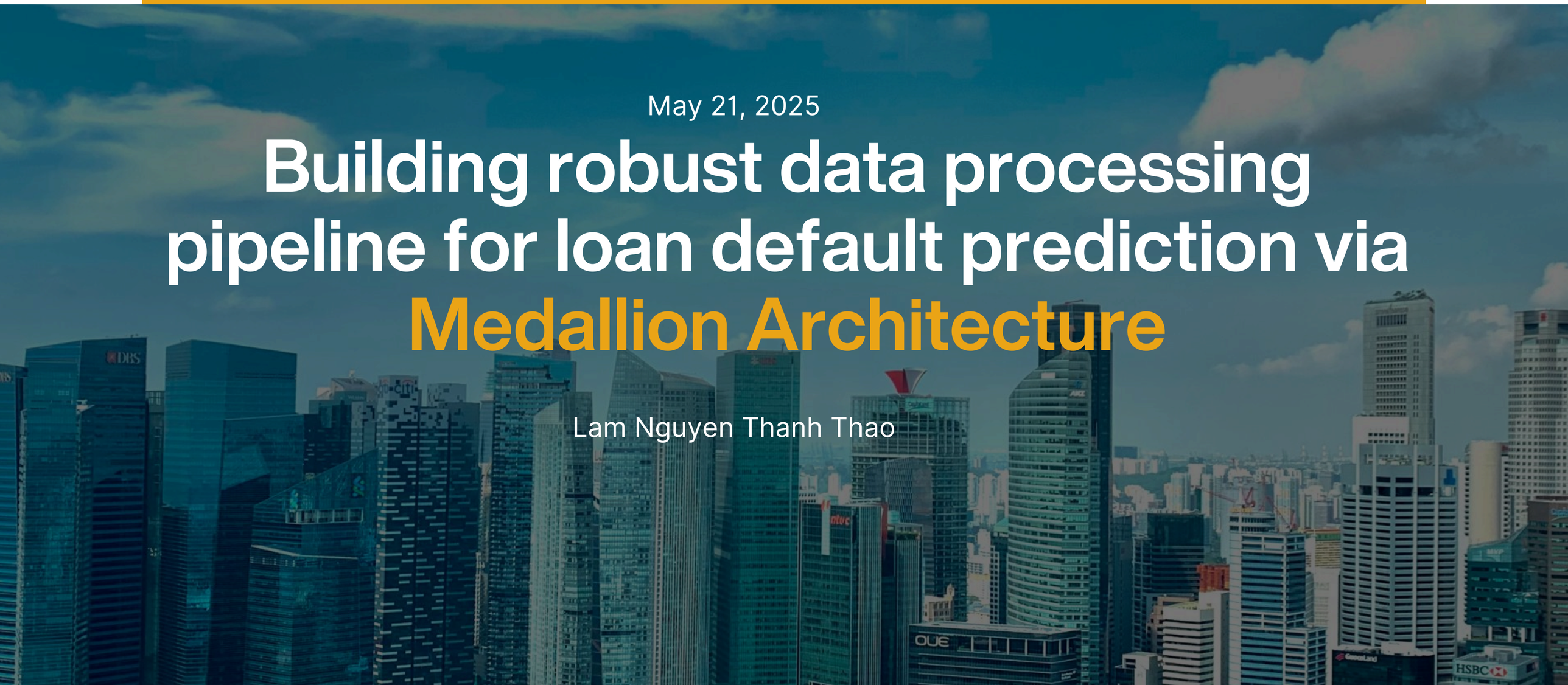


May 21, 2025

Building robust data processing pipeline for loan default prediction via Medallion Architecture

Lam Nguyen Thanh Thao

Paving the way for smarter lending: a Data Pipeline Initiative

Project description

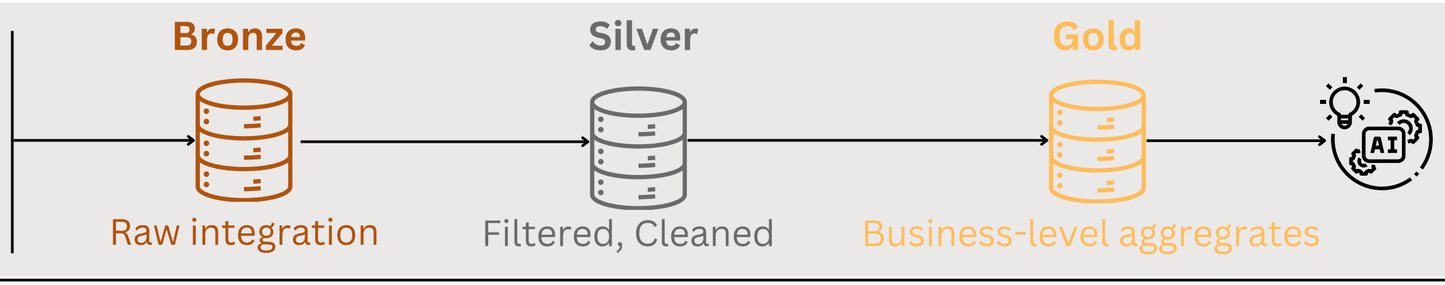
This project focuses on engineering a robust, end-to-end data pipeline to prepare high-quality data for developing an advanced loan default prediction model.

Key Benefits

- Reduce Credit Losses: More accurate identification of high-risk applicants.
- Optimize Lending: Smarter, data-driven loan origination decisions.
- Enhance Efficiency: Streamlined data preparation for future analytics.
- Strengthen Governance: Improved data quality and reliability.

Methodology

Employ the **Medallion Architecture** to process key customer data streams into validated, ML-ready assets, prioritizing quality and preventing data leakage.



Key Deliverables

An automated, end-to-end PySpark pipeline transforming raw data into structured, reliable **data storage (Bronze/Silver/Gold) for downstream ML prediction model,** which contains 2 key folders: **feature store** and **label store**.

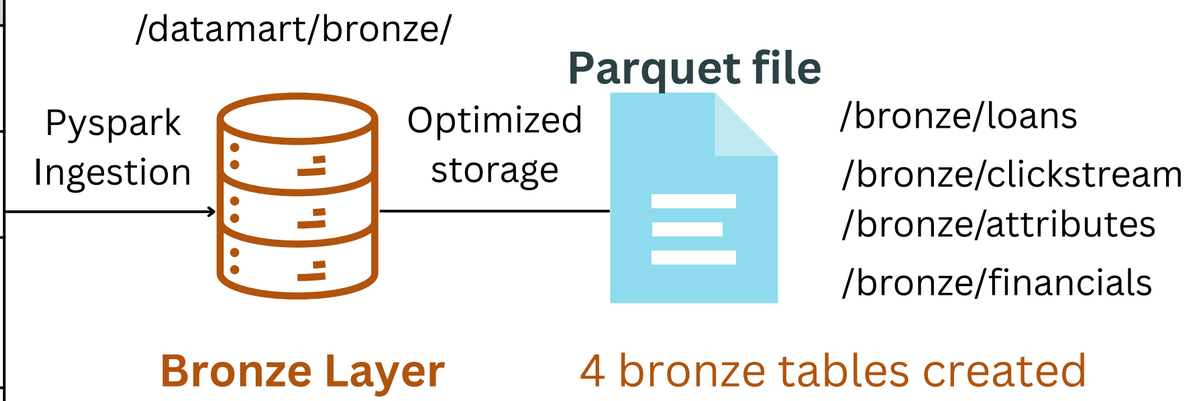
Laying the Groundwork - From Raw Data to Bronze Layer

First off, a Dockerized PySpark environment is established, then all raw source CSVs (Financials, etc.) are investigated and ingested 'as-is' into the Bronze Layer, creating an immutable ground-truth archive of efficient, partitioned Parquet files.

Investigate raw data tables

	Clickstream	Attributes	Financials	Loans
Rows	215,376	12,500	12,500	137,500
Columns	22	6	22	11
Unque Customer ID	8974	12500	12500	12500
Type of table	Snapshot	Overwrite	Overwrite	Fact (needs aggregate)
Snapshot Date range	Jan 2023 → Dec 2024	Jan 2023 → Jan 2025		Jan 2023 → Nov 2025

Archive data “as-is” into Bronze



- **Clickstream** contains multiple records per customer across time
→ a classic **snapshot table** used to anchor the timeline.
- **Attributes** and **Financials** each contain only 1 record per customer,
→ treated as **overwrite tables** with static values.
- **Loans** contains many rows per customer per date
→ **fact table** requiring aggregation by Customer_ID and snapshot_date to align temporally.

Silver Layer: Cleaning raw data into reliable information

Next, bronze data is loaded to be cleansed, standardized to create a trusted, analysis-ready source.

Cleansing step	Data Source				Rationale
	Financials	Attributes	Clickstream	Loans	
Standardize missing values (“_”, “NA” → None)	✓	✓	✓	Through Schema & Type Enforcement	Ensures consistent handling of absent data. Improve reliability & prevent errors in downstream analysis.
Validate data formats, remove noise “__52424” in income, invalid Age/SSN, etc.	✓	✓			Corrects inconsistencies. Leads to more accurate calculations and insights.
Enforce Correct Data Types (Casting)	✓	✓	✓	✓	Guarantees data integrity, enabling accurate financial/date-based logic and preventing analytical errors.
Validate/Standardize Categorical Values	✓ (Payment_Behavior)				Maintains consistency for key behavioral indicators, allowing for meaningful segmentation and risk profiling.
Structure Complex Text Fields	✓ (Type of Loan)				Prepares multi-value fields for feature extraction (e.g., loan counts), unlocking richer customer
Filter Incomplete Core Records (Customer_ID, snapshot_date nulls)	✓	✓	✓	✓	Removes unusable records, ensuring all data used for modeling is complete and reliably linkable.

Gold Layer: Overall logic

This layer is the final stage of our data transformation, where refined Silver data is engineered into curated, analysis-ready **'Feature Stores'** and **'Label Stores'** - specifically designed for our loan default prediction model.

datamart/silver/
financials_clean.csv

datamart/silver/
attributes_clean.csv

datamart/silver/
clickstream_clean.csv

datamart/silver/
loan_clean.csv

Gold Processing Engine

- Feature transformation
- Feature Engineering
- Final Data Aggregation

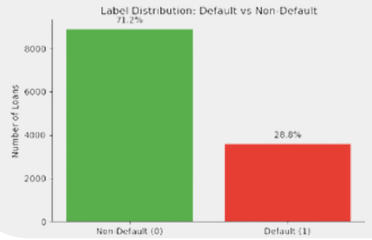
Feature store → Parquet file

Label store → Parquet file

- **ML-Ready:** Specifically structured for direct input into predictive models.
- **Business-Centric:** Features and aggregations are designed to answer specific business questions (predict default).
- **Leakage-Proofed:** Rigorous point-in-time logic minimizes data leakage.

Label definition

Label is defined by a loan's payment behavior at 6 months after it starts. If the loan is 30 or more days overdue at that point (Days Past Due ≥ 30): Label = 1 (Default); otherwise, Label = 0 (Non-Default).



- Moderate imbalance in label, which should be noted during training.
- To prevent target leakage, all features will be derived before or at the label snapshot date.

Gold logic deep dive: Transforming existing features

To select and refine the most impactful features, Exploratory Data Analysis (EDA) is conducted. This involves plotting existing feature distributions against the time-aligned 'default' label, guiding their transformation into optimal forms for ML input.

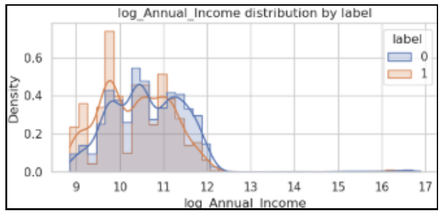
Financials

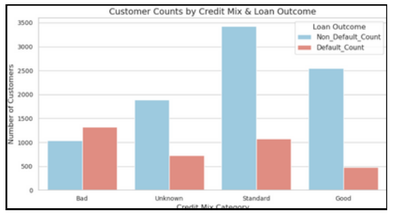
- **Imputed missing categoricals**, e.g.:
  - missing *Credit_Mix*: "Unknown";
  - missing *Outstanding_Debt*: median value.

→ Ensures robustness against outliers.

- **Parsed and quantified Credit History Age** text into *credit_history_months*
- **Applied log-scaling and outlier capping** for numeric stability(*e.g., log1p (Annual Income*))
- **One-hot encoded categorical fields** like *Credit_Mix* and *Payment_Behaviour*

→ ML-usable numerical format, capturing distinct risk profiles per category.





Attributes

- **Validated *Age* range and binned into *Age_group*** (e.g., "<25", "25–39", etc.)

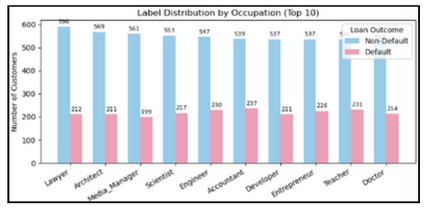
→ Remove noise/errors, capture non-linear patterns and reduces the risk of overfitting to specific ages.

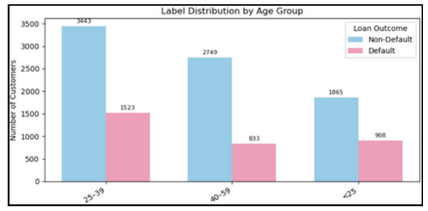
- **Handled missing *Occupation* with** Unknown.

→ Retain all rows and allow the model to learn from missingness explicitly.

- **One-Hot Encoded Age Group & Occupation**

→ For models to learn distinct effects of each group without assuming ordinal relationships.



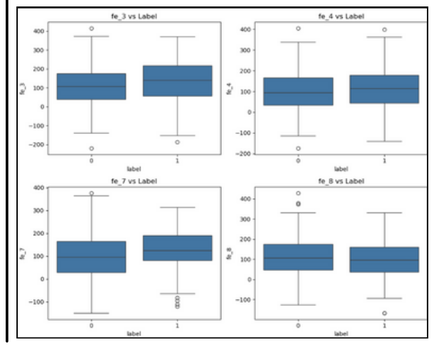


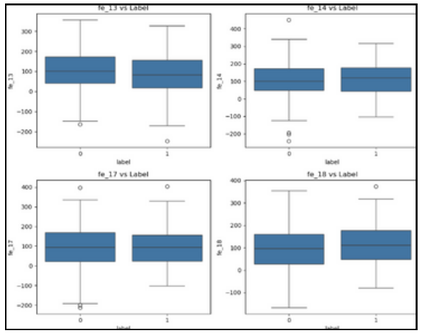
Clickstream

- **Capped outliers** across all fe_1 to fe_20 features to reduce skew.

→ Reduces impact of extreme/anomalous values on the model's learning process.

→ Improves model stability & robustness, leading to a more stable model that generalizes better to new, unseen data.





Gold logic deep dive: Engineering new features

After transforming existing features, new, meaningful features such as financial ratios, behavioral indicators, and customer profile insights features are engineered to better capture credit risk signals.

Creating new financials ratio

Feature	Description & Rationale
credit_history_months	Parsed from credit history text → converts to numeric total months of credit experience
num_loan_types	proxies for loan complexity & diversity
debt_to_income	Key creditworthiness ratio
emi_to_salary, investment_rate	Spending/investing habits normalized by income → reflect financial discipline
has_credit_limit_change	Binary flag for behavioral signal → indicates financial volatility
balance_to_debt, inq_per_loan	Encodes liquidity vs. debt and credit-seeking behavior

Deriving Behavioral & Profile Insights

Feature	Description & Rationale
Age_group	Binned into categories to capture non-linear effects of age
Occupation	One-hot encoded to allow the model to distinguish between job types
One-hot vectors	Applied to both categorical fields using StringIndexer + OneHotEncoder
fe_1 to fe_20	Behavioral features; capped outliers to reduce skew and overfitting

Gold logic deep dive: Final Aggregation & ML-ready data storing

Lastly, all engineered and time-aligned data sources are joined to construct the definitive gold feature_store, alongside creating a precise gold label_store – both primed for later machine learning model usage.

Unifying processed data streams

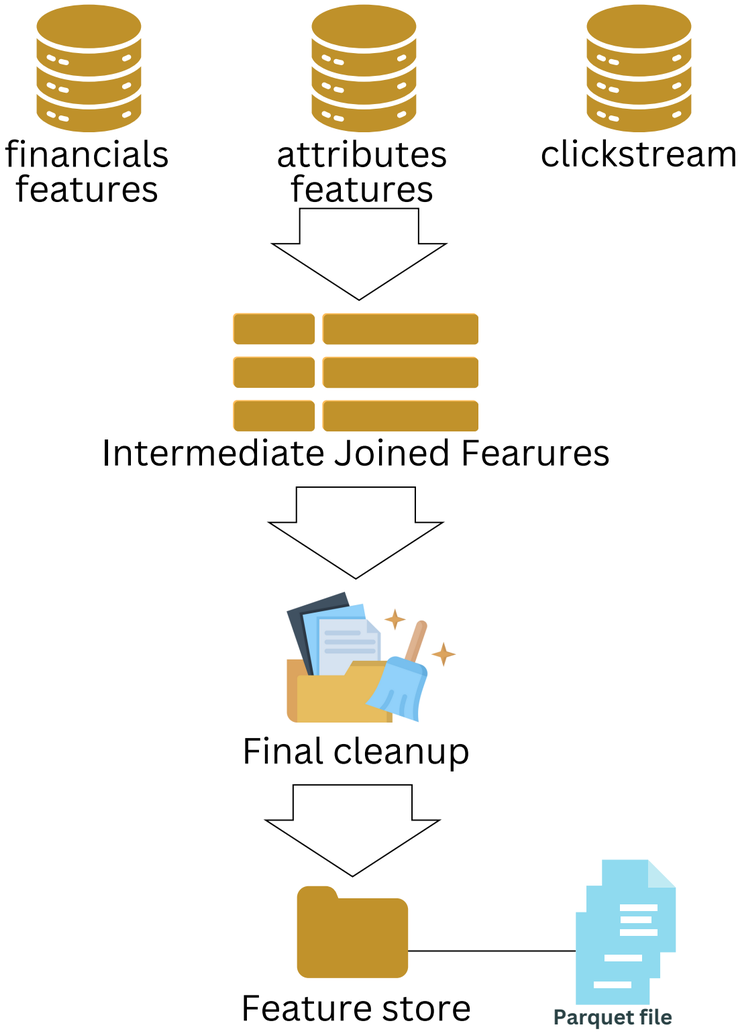
- The engineered and point-in-time aligned data from Financials (fin_all), Attributes (att_all), and Clickstream (clk_all) are joined together.
- Join Keys: "Customer_ID, feature_snapshot_date (and temporarily label, label_snapshot_date for alignment before they are dropped from the feature store).

Strategic Column Finalization & Cleanup

- Dropping unnecessary columns:
  - **Target Variables:** label, label_snapshot_date → Critical to prevent data leakage.
  - **PII:** Name, SSN → Ensures data privacy & security
  - **Redundant Raw/Intermediate Features**: Annual_Income (log_Annual_Income is used), Type_of_Loan (encoded versions exist), Loan_Types_Array, etc.
    → Simplifies feature set, reduces multicollinearity, focuses on most predictive forms.

Output & Characteristics

- Shape: 8974 rows, 45 columns.
- 4 Bronze tables, 4 Silver tables, 2 Gold tables
- Key Attributes: ML-ready, point-in-time correct, free of target leakage, PII removed, and focused on predictive signals



Conclusion: Delivering Value Today, Powering Predictions Tomorrow

Overall, this project has successfully implemented a fully automated PySpark data pipeline (Bronze, Silver, Gold) within a reproducible Dockerized environment.

Immediate Business Value & Impact

- Foundation for accurate, **high-performing loan default prediction model.**
- **Enhanced Risk Management:** Provides the means to more accurately identify and quantify credit risk, leading to better lending decisions.
- **Improved Data Governance & Reusability**: establishes a trusted, centralized data asset that improves data quality and can be leveraged for future analytical projects across the bank.

Strategic Path Forward & Beyond

- **Develop & Train Advanced ML Models:** Utilize the gold feature_store and label_store to build, train, and rigorously evaluate state-of-the-art loan default prediction models
- **Iterative Feature & Model Refinement:** Analyze model performance, feature importance, and business feedback to continuously enhance the feature set and model accuracy
- **Explore Model Deployment & Business Integration**

> With this robust data foundation, our team is now better positioned than ever to leverage the power of machine learning for superior risk management and enhanced profitability.

Thank You So Much