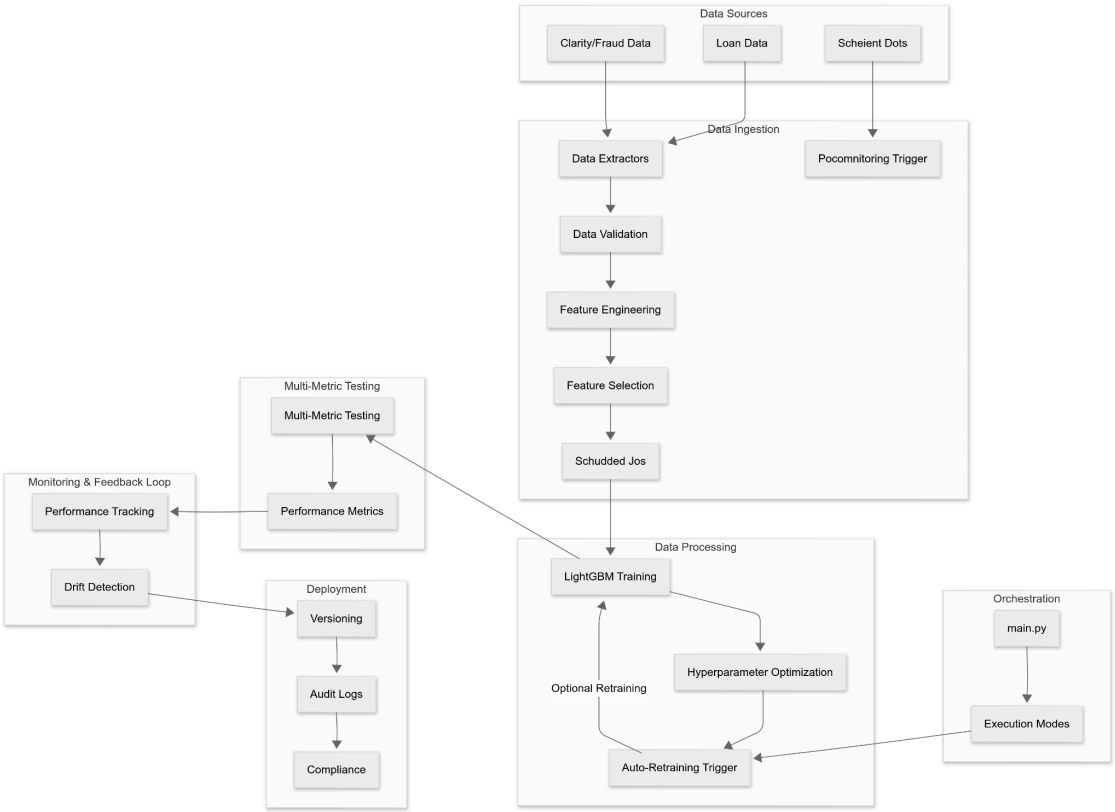
Loan Risk Prediction Automated ML System

System Architecture

The MoneyLion Loan Risk Prediction system is an automated machine learning pipeline designed to continuously assess and improve loan risk predictions. It integrates data from various sources including loan applications, payment histories, and fraud check variables. The system performs advanced data preprocessing, feature engineering, and model training using LightGBM. It includes modules for model evaluation, deployment, and ongoing performance monitoring with drift detection. Governed by configurable workflows and robust logging, the pipeline ensures reproducibility, versioning, and compliance. This scalable and modular architecture supports continuous learning, enabling MoneyLion to maintain accurate and reliable risk scoring for loan applicants.



Component	Description
Data Sources	Includes multiple datasets: - Loan Data: Historical loan information - Clarity/Fraud Data: Underwriting and fraud detection data - Scheint Dots: External/auxiliary data source
Poco Monitoring Trigger	Triggers data ingestion processes from external

	sources or monitors input readiness
Data Extractors	Extracts raw data from the source systems and prepares it for validation and transformation
Data Validation	Ensures the quality, consistency, and completeness of the ingested data
Feature Engineering	Creates new features from the validated data, including transformations and combinations
Feature Selection	Selects the most relevant features for modeling
Scheduled Jobs	Automates pipeline steps like data extraction and training at defined intervals
LightGBM Training	Trains the core risk model using LightGBM (a fast, efficient gradient boosting framework)
Hyperparameter Optimization	Tunes model parameters for optimal performance
Optional Retraining	Allows retraining of the model based on performance metrics or new data
Auto-Retraining Trigger	Automatically initiates retraining when certain thresholds are met
Multi-Metric Testing	Evaluates model using various metrics (classification, regression, etc.)
Performance Metrics	Outputs model evaluation scores used for deployment decisions
Performance Tracking	Monitors deployed model's performance over time
Drift Detection	Identifies changes in input data distribution or model prediction behavior
Versioning	Tracks different versions of the model and pipeline artifacts
Audit Logs	Records every operation for compliance and debugging
Compliance	Ensures the system meets regulatory and internal standards
main.py (Orchestration)	Entry point for executing pipeline steps using different modes (training, evaluation, etc.)
Execution Modes	Allows the system to run in different configurations: training, evaluation, or deployment