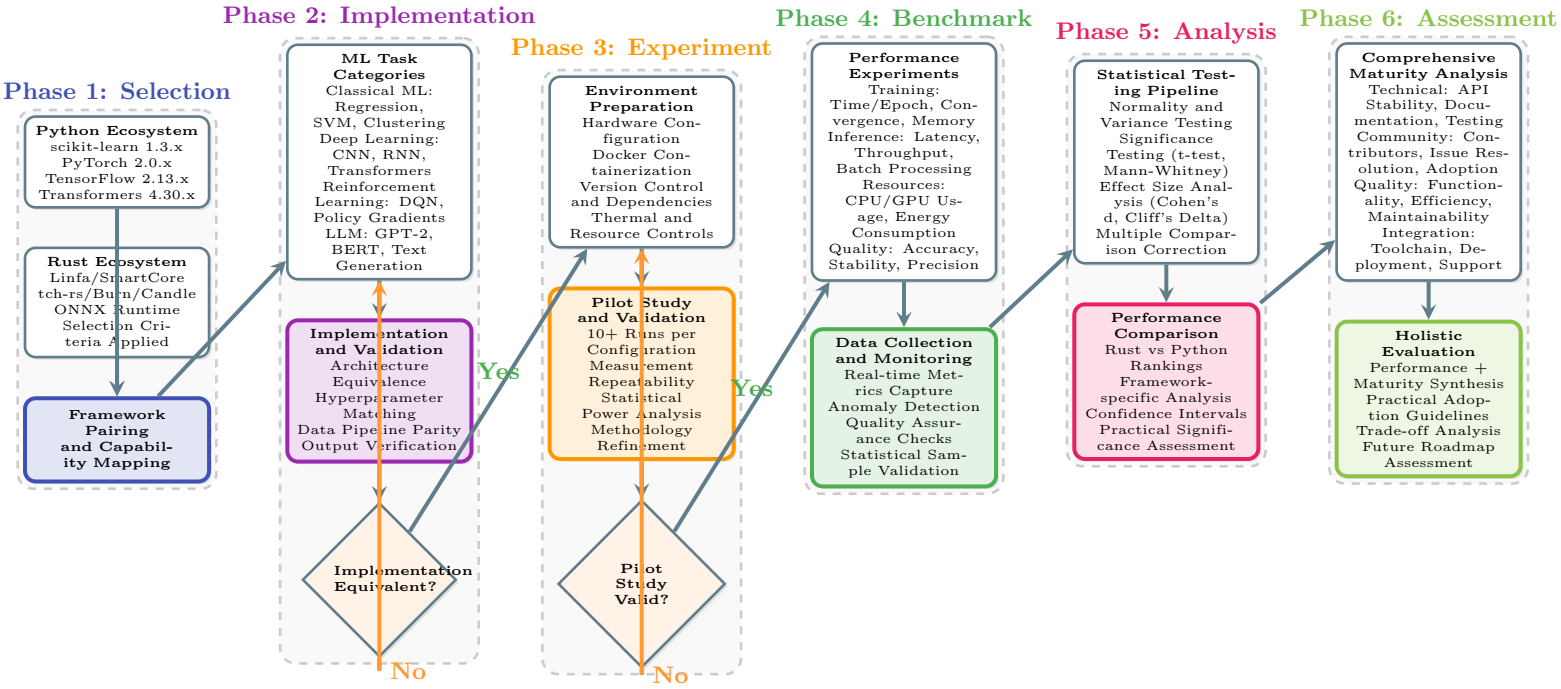
Phase 1: Selection
Python Ecosystem
scikit-learn 1.3.x
PyTorch 2.0.x
TensorFlow 2.13.x
Transformers 4.30.x
Rust Ecosystem
Linfa/SmartCore
tch-rs/Burn/Candle
ONNX Runtime
Selection Criteria Applied
Framework Pairing and Capability Mapping
Phase 2: Implementation
ML Task Categories
Classical ML: Regression, SVM, Clustering
Deep Learning: CNN, RNN, Transformers
Reinforcement Learning: DQN, Policy Gradients
LLM: GPT-2, BERT, Text Generation
Implementation and Validation
Architecture Equivalence
Hyperparameter Matching
Data Pipeline Parity
Output Verification
Implementation Equivalent?
Yes
No
Phase 3: Experiment
Environment Preparation
Hardware Configuration
Docker Containerization
Version Control and Dependencies
Thermal and Resource Controls
Pilot Study and Validation
10+ Runs per Configuration
Measurement Repeatability
Statistical Power Analysis
Methodology Refinement
Pilot Study Valid?
Yes
No
Phase 4: Benchmark
Performance Experiments
Training: Time/Epoch, Convergence, Memory
Inference: Latency, Throughput, Batch Processing
Resources: CPU/GPU Usage, Energy Consumption
Quality: Accuracy, Stability, Precision
Data Collection and Monitoring
Real-time Metrics Capture
Anomaly Detection
Quality Assurance Checks
Statistical Sample Validation
Phase 5: Analysis
Statistical Testing Pipeline
Normality and Variance Testing
Significance Testing (t-test, Mann-Whitney)
Effect Size Analysis (Cohen's d, Cliff's Delta)
Multiple Comparison Correction
Performance Comparison
Rust vs Python Rankings
Framework-specific Analysis
Confidence Intervals
Practical Significance Assessment
Phase 6: Assessment
Comprehensive Maturity Analysis
Technical: API Stability, Documentation, Testing
Community: Contributors, Issue Resolution, Adoption
Quality: Functionality, Efficiency, Maintainability
Integration: Toolchain, Deployment, Support
Holistic Evaluation
Performance + Maturity Synthesis
Practical Adoption Guidelines
Trade-off Analysis
Future Roadmap Assessment