LLM Schedule Optimizer - Main Flow

START
main.py

1. Load HDFS Jobs
jobs/loader.py
• Parse logs
• Extract features
• 5000 jobs loaded

2. Load Prompts
prompts/templates.py
• simple (10 tokens)
• standard (25 tokens)
• fewshot_1 (50 tokens)
• fewshot_3 (90 tokens)

3. Create Training Data
predictors/mlbp.py

5% Real LLM Calls
llm/gemini_client.py
• Call Ollama
• Cache results
• Label = correct?

95% Heuristic
• Job difficulty
• Prompt capability
• Pattern-based

4. Train XGBoost
• 20,000 samples
• 7 features
• Accuracy: 79%

5. Predict Probabilities
P(success | job, prompt)
Matrix: 5000 × 4

6. Baseline Evaluation
evaluation/baseline.py
• simple: 75.1% @ $11
• standard: 77.7% @ $18.5
• fewshot_1: 79.0% @ $31
• fewshot_3: 84.1% @ $51

7. Run Optimizers (3 runs each)

Random Search
• 1000 iterations
• ~20 solutions

NSGA-II
• Pop: 100, Gen: 50
• ~40 solutions
• IGD = 0 (BEST)

SPEA2
• Archive-based
• ~12 solutions

8. Extract Pareto Front
evaluation/pareto.py
• Filter dominated solutions
• Keep optimal trade-offs

9. Compute Metrics
• IGD (distance)
• Δ (spread)
• M_n (count)

10. Generate Outputs
• Pareto plot
• CSV tables
• Results summary

END
Results Saved

Legend

Data Loading
ML Training
Optimization
Evaluation
Output