DETAILED GRADE BREAKDOWN

Assignment 2

Student ID: 48951

Repository: LLMsMultiAgentOrchestration_RNN_LSTM

Assessment Date: 2025-12-02

FINAL SCORE: 20.5 / 100

Performance Tier: Below Standard

Skills Summary

Skill	Score	Status
Project Planning	0.0/10	✗ Poor
Code Documentation	6.5/10	✓ Good
Config Security	5.0/10	■ Fair
Testing Quality	0.0/10	✗ Poor
Research Analysis	0.0/10	✗ Poor
Ui Ux	4.0/10	■ Fair
Version Management	4.0/10	■ Fair
Costs Pricing	0.0/10	✗ Poor
Extensibility	0.0/10	✗ Poor
Quality Standards	1.0/10	✗ Poor

Skill: Project Planning

Score: 0.0/10 points

Status: ✗ Poor

Points Breakdown

Criterion	Max Points	Earned	Status
PRD.md exists	2.0	0.0	✗
ARCHITECTURE.md exists	5.0	0.0	✗
Problem Statement	1.0	0.0	✗
Functional Requirements	1.5	0.0	✗
Success Metrics	0.5	0.0	✗
TOTAL	**10.0**	**0.0**	

What Was Found

• No PRD.md document found

• No ARCHITECTURE.md document found

• 2 graph images found (graph1_single_freq.png, graph2_all_freqs.png) but not architecture diagrams

What Was Missing

• No PRD.md document found

• No ARCHITECTURE.md document found

• No planning documents in README.md

How to Improve (+10.0 points)

1. Create a PRD.md with problem statement, functional requirements, and success metrics

2. Create an ARCHITECTURE.md with C4 diagrams showing system design

Skill: Code Documentation

Score: 6.5/10 points

Status: ✓ Good

Points Breakdown

Criterion	Max Points	Earned	Status
README.md >1KB	3.0	3.0	✓
Installation instructions	1.0	1.0	✓
Usage examples	1.0	1.0	✓
Code structure documented	2.0	0.0	✗
Python docstrings (>50%)	3.0	2.0	✗
TOTAL	**10.0**	**6.5**	

What Was Found

• README.md exists and is 2400 bytes (>1KB): +3 points

• Has reproduction/installation instructions: +1 point

• Has usage/evaluation examples: +1 point

• 2 out of 4 Python files have docstrings (50%): +1.5 points

What Was Missing

• Code structure not explicitly documented: 0 points

How to Improve (+3.5 points)

1. Add docstrings to data_gen.py and evaluate.py

2. Add a 'Project Structure' section to README explaining the codebase organization

Skill: Config Security

Score: 5.0/10 points

Status: ■ Fair

Points Breakdown

Criterion	Max Points	Earned	Status
No hardcoded secrets (CRITICAL)	5.0	5.0	✓
.env.example exists	2.0	0.0	✗
.gitignore exists	1.0	0.0	✗
Uses environment variables	2.0	0.0	✗
TOTAL	**10.0**	**5.0**	

What Was Found

• CRITICAL: No hardcoded API keys or secrets found: +5 points (baseline security)

• No .env.example file found: 0 points

• No .gitignore file found: 0 points

What Was Missing

• CRITICAL: No hardcoded API keys or secrets found: +5 points (baseline security)

• No .env.example file found: 0 points

• No .gitignore file found: 0 points

• No use of environment variables (os.getenv): 0 points

How to Improve (+5.0 points)

1. Add .gitignore to exclude data files, model checkpoints, and Python cache

2. Consider adding .env.example if future features require configuration

Skill: Testing Quality

Score: 0.0/10 points

Status: ✗ Poor

Points Breakdown

Criterion	Max Points	Earned	Status
Test files exist	3.0	0.0	✗
Multiple test files (>3)	2.0	0.0	✗
Test framework configured	2.0	0.0	✗
Test functions (>10)	3.0	0.0	✗
TOTAL	**10.0**	**0.0**	

What Was Found

• No test files found (test_*.py or *test*.py): 0 points

What Was Missing

• No test files found (test_*.py or *test*.py): 0 points

• No pytest.ini or test configuration: 0 points

• No tests directory: 0 points

How to Improve (+10.0 points)

1. Add unit tests for model.py (test LSTM architecture)
2. Add tests for data_gen.py (verify data generation correctness)
3. Add integration tests for train.py (test training pipeline)
4. Set up pytest with coverage reporting

Skill: Research Analysis

Score: 0.0/10 points

Status: ✗ Poor

Points Breakdown

Criterion	Max Points	Earned	Status
Jupyter notebooks exist	4.0	0.0	✗
Multiple notebooks (>2)	2.0	0.0	✗
Has visualizations/plots	2.0	2.0	✓
Analysis documentation	2.0	0.0	✗
TOTAL	**10.0**	**0.0**	

What Was Found

• No Jupyter notebooks found (.ipynb): 0 points

What Was Missing

• No Jupyter notebooks found (.ipynb): 0 points

• Visualizations exist as PNG files (2 graphs) but no notebooks: partial credit not applicable

• README documents pedagogical insights but no exploratory analysis notebooks

How to Improve (+10.0 points)

1. Create a Jupyter notebook showing exploratory data analysis
2. Add notebook documenting hyperparameter tuning experiments
3. Create visualization notebook showing model performance analysis

Skill: Ui Ux

Score: 4.0/10 points

Status: ■ Fair

Points Breakdown

Criterion	Max Points	Earned	Status
Screenshots/images (1+)	3.0	3.0	✓
Screenshots/images (5+)	3.0	1.2	✗
UI documentation	2.0	2.0	✓
User guide exists	2.0	0.0	✗
TOTAL	**10.0**	**4.0**	

What Was Found

• 2 visualization graphs found (graph1_single_freq.png, graph2_all_freqs.png): +3 points

• Visual analysis section in README references graphs: +1 point

What Was Missing

• No separate user guide: 0 points

• No UI design documentation (not applicable for ML project): 0 points

How to Improve (+6.0 points)

1. Add more detailed captions for graphs in README

2. Consider adding a USER_GUIDE.md for reproduction steps

Skill: Version Management

Score: 4.0/10 points

Status: ■ Fair

Points Breakdown

Criterion	Max Points	Earned	Status
Git commits >10	2.0	2.0	✓
Meaningful commit messages	2.0	2.0	✓
PROMPT_BOOK.md exists	5.0	0.0	✗
Branching strategy	1.0	0.0	✗
TOTAL	**10.0**	**4.0**	

What Was Found

• 21 commits (>10): +2 points

• Commit messages are descriptive (e.g., 'Adjusted noise data', 'Increase number of epochs'): +2 points

• No PROMPT_BOOK.md found: 0 points (critical for Assignment 2)

What Was Missing

• No PROMPT_BOOK.md found: 0 points (critical for Assignment 2)

• No branching strategy documented: 0 points

How to Improve (+6.0 points)

1. Create PROMPT_BOOK.md documenting all AI assistant interactions

2. Document the development workflow and any branching strategy used

Skill: Costs Pricing

Score: 0.0/10 points

Status: ✗ Poor

Points Breakdown

Criterion	Max Points	Earned	Status
Cost analysis document	5.0	0.0	✗
Cost mentions in docs	3.0	0.0	✗
Budget tracking	2.0	0.0	✗
TOTAL	**10.0**	**0.0**	

What Was Found

• No cost/pricing/budget documents found: 0 points

What Was Missing

• No cost/pricing/budget documents found: 0 points

• No cost references in README or documentation: 0 points

• No analysis of computational costs (GPU training time, cloud costs, etc.)

How to Improve (+10.0 points)

1. Add cost analysis document covering GPU training costs

2. Document computational requirements and estimated costs

3. Add budget considerations for scaling the model

Skill: Extensibility

Score: 0.0/10 points

Status: ✗ Poor

Points Breakdown

Criterion	Max Points	Earned	Status
Plugin/extension system	3.0	0.0	✗
Modular structure (3+ dirs)	3.0	0.0	✗
Interfaces/APIs	2.0	0.0	✗
Extension documentation	2.0	0.0	✗
TOTAL	**10.0**	**0.0**	

What Was Found

What Was Missing

• No plugin or extension system: 0 points

• No abstract interfaces or ABC classes: 0 points

• Code is monolithic without clear modular structure: 0 points

• No extension documentation: 0 points

How to Improve (+10.0 points)

1. Refactor to use abstract base classes for models

2. Create a modular architecture with separate modules for data, models, training

3. Add interfaces to support different model architectures

4. Document how to extend the system with new frequency patterns or model types

Skill: Quality Standards

Score: 1.0/10 points

Status: ✗ Poor

Points Breakdown

Criterion	Max Points	Earned	Status
Linting configuration	2.0	0.0	✗
CI/CD pipeline	3.0	0.0	✗
Code style guide	2.0	0.0	✗
Pre-commit hooks	2.0	0.2	✗
Project setup file	1.0	0.0	✗
TOTAL	**10.0**	**1.0**	

What Was Found

• Code shows some quality (good comments, clear structure): +1 point for internal quality

What Was Missing

• No linting configuration (.pylintrc, .flake8, pyproject.toml): 0 points

• No CI/CD pipeline (.github/workflows): 0 points

• No code style guide: 0 points

• No requirements.txt or setup.py: 0 points

How to Improve (+9.0 points)

1. Add requirements.txt listing all dependencies (torch, pandas, numpy)
2. Set up linting with pylint or flake8
3. Add GitHub Actions workflow for automated testing
4. Create a CONTRIBUTING.md with code style guidelines

Overall Assessment Summary

Key Strengths (8+ points)

• No skills scored 8 or above

Critical Gaps (<5 points)

• Project Planning: 0.0/10

• Testing Quality: 0.0/10

• Research Analysis: 0.0/10

• Ui Ux: 4.0/10

• Version Management: 4.0/10

• Costs Pricing: 0.0/10

• Extensibility: 0.0/10

• Quality Standards: 1.0/10

Recommended Actions

Immediate Priority:

• Create PROMPT_BOOK.md documenting all AI interactions

• Add requirements.txt with dependencies

• Create PRD.md and ARCHITECTURE.md

• Add .gitignore file

High Priority:

• Write unit tests for all Python modules

• Create Jupyter notebooks for exploratory analysis

• Add cost analysis document

• Set up linting configuration