**Sprint-1 Minimum Viable Product**

**Introduction**

In this Sprint, the purpose was to create a minimal viable working product. The following sections contain the User Stories I worked on with a detailed description of the Tasks I worked on.

**User Stories**

I worked on the following User Stories:

[**NQR: Fundamental ML Pipeline (FNN / RF / ANFIS) for Next-Quarter Return Prediction #597**](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/597)

**Conditions of Satisfiability:**

* Data Integrity: No missing or malformed ratio inputs; NaNs handled.
* Model Validity: All three models accept the same input shape and produce probabilities in [0, 1].
* Performance: Full inference (load + predict) runs ≤ 1 s on Codespaces CPU.
* Robustness: Pipeline tolerates new tickers or missing quarters without crashing.
* Integration: DecisionAgent always returns BUY or SELL (no HOLD).

**Definition of Done:**

* Ratio fetching and storage scripts implemented and unit-tested.
* FNN, RF, and ANFIS model code written, trained, and validated.
* Inference endpoint loads all three models and emits correct JSON.
* Crew AI decision agent defined and tested end-to-end.
* Integration tests and documentation (README, examples) completed.
* Backtesting harness implemented and performance report generated.

**Tasks**

[NQR.1 Ratio Data Retrieval & Preparation #598](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/598)

[NQR.2 Model Training Pipeline #629](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/629)

[NQR.2.1 Define FNN architecture and training loop (3 ph) #692](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/692)

[NQR.2.2 Train and validate FNN; log accuracy (2 ph) #693](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/693)

[NQR.2.3 Train Random Forest with hyperparameter tuning (2 ph) #694](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/694)

[NQR.2.4 Train ANFIS model and verify convergence (3 ph) #695](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/695)

[NQR.2.5 Compare validation metrics and select thresholds (1 ph) #696](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/696)

[NQR.3 Inference API Development #697](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/697)

[NQR.3.1 Create ModelInferAgent script to load all three models (2 ph) #698](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/698)

[NQR.3.2 Given input ratios, compute and format { fnn\_prob, rf\_prob, anfis\_prob } JSON (1 ph) #699](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/699)

[NQR.3.3 Benchmark inference latency and optimize if >1 s (1 ph) #700](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/700)

[NQR.4 Crew AI Decision Agent #701](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/701)

[NQR.4.1 Define Crew AI prompt logic for majority-vote or threshold rules (2 ph) #702](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/702)

[NQR.4.2 Implement and test DecisionAgent with edge-case probabilities (1 ph) #703](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/703)

[NQR.5 Integration & Testing #704](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/704)

[NQR.5.1 Write end-to-end integration test: fetch → ratios → infer → decision (2 ph) #705](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/705)

[NQR.5.2 Validate outputs for a sample ticker set (1 ph) #706](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/706)

[NQR.5.3 Document pipeline, usage examples, and agent prompts in README (2 ph) #707](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/707)

[NQR.6 Backtesting & Evaluation #708](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/708)

[NQR.6.1 Prepare historical test sets and data conversion (2 ph) #709](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/709)

[NQR.6.2 Implement backtesting harness to simulate the full infer → decision pipeline over history (3 ph) #710](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/710)

[NQR.6.3 Calculate backtesting performance metrics (accuracy, return, drawdown) and analysis (2 ph) #711](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/711)

[NQR.6.4 Generate backtesting report and visualizations (2 ph) #712](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/712)

**Tasks I Worked On**

[NQR.1 Ratio Data Retrieval & Preparation #598](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/598)

I developed python code to pull quarterly statements and then compute ratios. I have also written unit tests for this. The task was estimated at 6 person hours but it took me 17 person hours to complete.

**Summary Table of Work**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| UserStory GitHub Issue ID | User Story | Story Points | Task GitHub Issue ID | Task | Task Hours | Status | Actual Hours |
| [NQR](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/597) | [Fundamental ML Pipeline (FNN / RF / ANFIS) for Next-Quarter Return Prediction #597](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/597) |  | [NQR.1](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/598) | [Ratio Data Retrieval & Preparation #598](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/598) | 6 | Complete | 17 |

**Summary Table of Commits**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Commit Number | Commit Description (exactly as in github) | User Story | Task |
| June 2nd, 2025 | da6b658e231772bd41d706680a4317d13ddb9b57 | [NQR Datafetch and ratio calculation](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/pull/625/commits/da6b658e231772bd41d706680a4317d13ddb9b57) | [NQR](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/597) | [NQR.1](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/598) |