Weekly Report: Nusrat Jahan Nuha-Alpha AI

Week-04 (May 12 - May 16)

1. Course Documentation

Mathematics for Machine Learning: Linear Algebra

- Reviewed the concept of vector spaces and how vectors operate within defined dimensions.
- Studied the definition and importance of a basis—a minimal set of linearly independent vectors that span a vector space.
- Differentiated between linear independence and linear dependence to identify valid bases and eliminate redundancy.
- Explored the application of changing bases to transform vector representations between coordinate systems.
- Applied Gaussian elimination to solve systems of linear equations and determine matrix rank.
- Learned to compute the inverse matrix.
- Emphasized the foundational role of these concepts in advanced mathematics and real-world applications like data science and machine learning.

2. TESLA-S (Tesla Stock Price Forecasting) Project (Cont.)

Completed Tasks

- Model Evaluation for Deep Learning Models (LSTM, GRU)
- Pipeline Creation for Ensemble Model
- Model Evaluation for The Ensemble model

A. Model Evaluation for Deep Learning Models (LSTM, GRU)

Implemented and evaluated deep learning models LSTM and GRU, for time series forecasting. Both models were trained on sequential stock data of Tesla using a sliding time window approach to capture temporal dependencies. After training, assessed performance using standard metrics such as MSE, RMSE, MAE, and R^2. The results highlighted the LSTM model's strong ability to model long-term dependencies, while the GRU model provided a more lightweight alternative with competitive accuracy and faster training times.

B. Pipeline Creation for Ensemble Model

An ensemble pipeline was developed by combining predictions from a trained Random Forest model and an LSTM model. Using a custom `EnsembleModel` class, I integrated both models with adjustable blend weights. The `find_optimal_weights` function applied linear regression to determine the

most effective weighting for blending RF and LSTM outputs, improving prediction robustness. This ensemble approach leveraged the strengths of both tree-based and sequential deep learning models to produce more stable and accurate forecasts.

C. Model Evaluation for The Ensemble model

The ensemble model was evaluated using the same metrics as the individual models (MSE, RMSE, MAE, R^2) and showed improved performance over standalone models in most cases. The ensemble achieved a better balance between bias and variance by combining the complementary predictive power of RF and LSTM. Visual plots of predicted vs. actual prices demonstrated enhanced alignment with ground truth, confirming the effectiveness of the blending strategy. The pipeline also included dynamic visualization for performance comparison and interpretation.

3. Session on Test Driven Development (TDD)

Had an insightful session on Test-Driven Development (TDD), which emphasized the importance of writing test cases before beginning the actual coding process. This proactive approach helps ensure code reliability, maintainability, and clarity of design. Also explored various testing strategies, including unit, integration, and functional testing, and discussed their roles in software quality assurance.

Research and Documentation Progress

Began documenting testing methodologies, focusing initially on unit testing and pytest frameworks.

A. **Unit Testing**

Unit testing involves writing test cases for individual components or functions to verify their correctness in isolation. This helps catch bugs early, simplifies debugging, and supports modular design. We have Python's built-in `unittest` module to create structured test suites with assertions, setup methods, and automated test runners.

B. **Pytest**

Pytest is a powerful and user-friendly testing framework for Python that offers a simpler syntax and more flexibility compared to unittest. It supports parameterized tests and better reporting, making it ideal for both small projects and complex test suites. For example, writing basic test functions and using assertions to validate code behavior under various inputs.