FinTech Unit 14 Homework: Grading Rubric

Criteria	Ratings				Points	Feedback
Data Prep for Training and Testing • **FNG** values used to predict future closing prices. • **Past Closing Prices** used to predict furture closing prices. • **MinMaxScaler** applied to the X and Y values to scale the data for the model. • **X_train** and **X_test** reshaped to fit the model requirements (samples, time steps, features)	**35 Points Mastery** • Completed 4 out of 4 requirements • Code runs without error and produces the assigned results • Code accounts for all possible scenarios • Code is free of bugs	**34 > 28 Points Approaching Mastery** • Completed 3 out of 4 of requirements • Code runs without error • Code produces results as expected 80% of the time	**28 > 23 Points Progressing** • Completed fewer than 2 out of 4 requirements • Code runs without error • Code produces results, but not necessarily the correct results	**23 > 0 Emerging** • Completed 1 or none out of the 4 requirements • Code runs with error	0	
Build and train custom LSTM RNNS • **Notebook created to fit data using FNG Values.** • **Notebook created to fit data using closing prices.** **Evaluate the Perfomance of each model.** • **Lowest loss model determined.** • **Model** with best accuracy tracking determined. • **Window Size** determined for the model.	**35 Points Mastery** • Completed 5 out of 5 requirements • Code runs without error and produces the assigned results • Code accounts for all possible scenarios • Code is free of bugs	**34 > 28 Points Approaching Mastery** • Completed 4 out of 5 of requirements • Code runs without error • Code produces results as expected 80% of the time	**28 > 23 Points Progressing** • Completed 3 out of 5 requirements • Code runs without error • Code produces results, but not necessarily the correct results	**23 > 0 Emerging** • Completed 2 or fewer out of the 5 requirements • No submission • Code runs with error	0	
Coding Conventions/Formating • Appropriate header, name, short description at top of the notebook • Imports are at the top of the file, just after any headers or subheads. • Files read in from relative file path • Functions and variable names are descriptive, lowercase, with words separated by underscores • Clean code, no repetition, maintainable and highly reusable code. • Appropriate code wrapping and cell sizes • Appropriate subheads as needed	**10 Points Mastery**	**9 Points Approaching Mastery**	**8 Points Progressing**	**8 > 0 Emerging**	0	
Deployment/Submission • Files submitted in personal repo • Appropriate directory structure with correct files needed to run scripts • Appropriate commit messages • Appropriate README	**10 Points Mastery**	**9 Points Approaching Mastery**	**8 Points Progressing**	**8 > 0 Emerging**	0	
Documentation/Comments • Code is well commented with concise, relevant comments	**10 Points Mastery**	**9 Points Approaching Mastery**	**8 Points Progressing**	**8 > 0 Emerging**	0	

LETTER GRADE

TOTAL POINTS 0   **F**