Reducing LSTM Output Variability

Currently, our test case is as such:

- Input: Dataset that contains AAPL hourly prices for the past few years
- After feeding through the LSTM,
- Output: it produces predictions around our target range however, the predictions are too scattered.

Ex. Current price of AAPL is 184.46

- LSTM predicts AAPL price in two hours
- After 3 runs, comes up with 185.45, 184.27, and 184.16.

Actual Price of AAPL after two hours is 184.26

Normally a 1.49 difference in a $180 stock isn’t very significant. However, In this case 185.45 and 184.16 is a big difference in outputs because one shows a decreasing movement from the original price and the other shows an increase. There are two ways we researched to fix that variability:

- **Random Seed Generation**
- **L1 Regularization**

Random Seed Generation

A random seed serves as an anchor point for algorithms that involve randomness. This includes processes like the initialization of weights, the shuffling of data, and any other random process in the model's training phase. It basically gives the LSTM a uniform starting point in two ways

- By setting consistent weight values
- By creating consistency in data shuffling

L1 Regularization

prevents from overfitting – learning the training data so well, including its noise and details, that it performs poorly on new data. It works by applying a penalty for complexity: it discourages the model from using too many or too large parameters/weights, The result is a simpler, more generalized model that focuses on the most important features of the data. We’ll implement L1 Regularization if our Random Seed Generation Integration isn’t sufficient