Hyperparameter Tuning and Performance

1. Tuning Method Chosen:

ARIMA MODEL:

We used **Grid Search** because it allows us to explore all combinations of the parameters p, d, and q to find the optimal configuration that minimizes the **Akaike Information Criterion (AIC)**.

- **ARIMA model**: ARIMA (AutoRegressive Integrated Moving Average) is used for time series forecasting and it is important to find the right combination of p (autoregressive order), d (differencing), and q (moving average order) to make the model perform well on the given data.
- **Why Grid Search?** Grid Search exhaustively evaluates all possible combinations of hyperparameters in a specified range. In the case of ARIMA, tuning p, d, and q ensures that the model captures both the temporal dependencies and the stationary nature of the time series data.

GRU MODEL:

We choose **KerasTuner's Hyperband** method for hyperparameter tuning because:

- **Hyperband** optimizes the search for the best model by using a combination of random search and bandit algorithms. This method efficiently explores hyperparameters by evaluating configurations and allocating resources based on the models' performance.
- It's suitable for deep learning models like GRU, which require extensive experimentation with different hyperparameters.
- **Hyperband** balances speed and accuracy by allocating more resources to promising configurations and speeding up the search process.

---

2. Hyperparameters Tuned and Why:

ARIMA MODEL:

- **p (AR order)**: The number of lag observations in the AR (AutoRegressive) model. It captures the past behavior of the time series and is crucial for understanding the influence of previous observations on future ones.
- **d (Differencing order)**: The number of times the series is differenced to make it stationary. Differencing is required when the data has trends or is non-stationary. Tuning d ensures that the series is stationary, which is a prerequisite for accurate time series forecasting.
- **q (MA order)**: The number of lagged forecast errors in the MA (Moving Average) model. Tuning q allows the model to account for the noise or shocks in the time series data, improving its ability to forecast accurately.

GRU MODEL:

a) units_1 and units_2 (Number of GRU units in each layer):

- We tuned the **number of units** in both GRU layers because this parameter controls the model's capacity to learn temporal dependencies in the time series data. Too few units may lead to underfitting, while too many could cause overfitting and longer training times.
- We selected a range of 50 to 200 with a step of 50 to test the impact of different layer sizes on the model's performance.

b) Optimizer:

- We tuned the **optimizer** to choose between **Adam** and **RMSprop**. Adam is commonly used in time series forecasting models as it adapts the learning rate, making it suitable for handling noisy data. RMSprop helps stabilize learning when dealing with non-stationary data.
- Tuning the optimizer helps find the best method to minimize loss during training, affecting convergence speed and accuracy.

c) Epochs and batch_size:

- These are standard training parameters, with **epochs** set to 10 and **batch_size** set to 32. These were tuned to balance training time and model performance.

---

3. Performance Metrics:

ARIMA MODEL:

- **RMSE:** 0.04110368480433161 (A low RMSE indicates that the model has effectively captured the trends in the data and provides reliable forecasts)

- **MAE:** 0.03026895838134875 (The low MAE value indicates that, on average, the model's predictions are close to the actual values.)

GRU MODEL:

- RMSE improved significantly to 0.0176 (lower value = better model performance).
- MAE also improved to 0.0118, indicating better prediction accuracy with less average error between predicted and actual values.