IPL Score Prediction

Observations:

From the dataset and preprocessing steps, we can observe that:

- The dataset consists of features such as **venue, batting team, bowling team, batsman, bowler, and total runs scored**.
- Certain unnecessary columns like **date, runs, wickets, overs, runs_last_5, wickets_last_5, mid, striker, and non-striker** were dropped to improve model efficiency.
- Categorical variables (**venue, bat_team, bowl_team, batsman, and bowler**) were **label-encoded** for numerical processing.
- The **features were normalized** using MinMaxScaler to ensure better performance in model training.
- The dataset was split into **training (70%) and testing (30%) sets**.

From the **Neural Network training results**, we can observe that:

- The **Huber loss** was used as the loss function to handle outliers in run values.
- The model was trained for **50 epochs**, with validation loss **decreasing gradually**, indicating good learning.
- The final **training loss** and **validation loss** suggest that the model successfully learned patterns in the data.

From the **model evaluation metrics**, we observe that:

- The **Mean Absolute Error (MAE)** for the trained model on the test set is **19.48**, meaning that on average, the predicted total runs differ by approximately **19 runs** from the actual total.
- The **Neural Network performed well**, showing a stable decrease in loss over epochs.

Conclusion:

- The **Neural Network Model** provides **the best performance** based on the given data.
- The **MAE of 19.48** suggests that the model can predict total runs with a reasonable level of accuracy.
- The model can be further improved by tuning hyperparameters, adding more relevant features, or using alternative models such as XGBoost or Random Forest for comparison.
