IPL Score Prediction Analysis

Introduction: Predicting IPL match scores is a crucial machine learning application in sports analytics. This project aims to analyze match data, preprocess key features, and train a machine learning model to predict the total score of a batting team. The methodology involves data preprocessing, exploratory data analysis (EDA), feature selection, and model training.

Skills used: Deep Learning, TensorFlow, Keras

Dataset Overview: The dataset used for IPL score prediction contains multiple features describing the match conditions, teams, players, and innings progress.

Dataset Features

1. **mid** - Match ID (not used for prediction)
2. **date** - Date of the match
3. **venue** - Stadium where the match was played
4. **bat_team** - Batting team
5. **bowl_team** - Bowling team
6. **batsman** - Current batsman
7. **bowler** - Current bowler
8. **runs** - Runs scored so far
9. **wickets** - Wickets lost so far
10. **overs** - Overs bowled so far
11. **runs_last_5** - Runs scored in the last 5 overs
12. **wickets_last_5** - Wickets lost in the last 5 overs
13. **total** (Target Variable) - Total predicted score of the batting team

The target variable **total** is what we aim to predict based on other attributes, i.e., we take **total** as the response variable.

Analysis of Data

1. **Data Distribution**
    - The dataset consists of multiple teams and stadiums across IPL seasons.
    - The average total score varies depending on the venue and teams.
    - Teams batting first tend to have higher scores compared to teams chasing.
    - Performance in the last 5 overs is a strong indicator of final scores.
2. **Feature Importance** Using feature selection techniques, the most influential factors in score prediction were identified:

- **Venue** - Some stadiums have higher average scores.
- **Batting Team** - Certain teams have historically higher scores.
- **Bowling Team** - Strong bowling teams tend to restrict runs.
- **Runs in Last 5 Overs** - Higher scores in the last 5 overs indicate a strong finish.
- **Overs Completed** - As overs increase, run prediction becomes more accurate.

Machine Learning Model Implementation

1. **Data Preprocessing**
    - **Handling Missing Values** - Not applicable as dataset was complete.
    - **Encoding Categorical Variables** - Used **Label Encoding** for venue, teams, batsmen, and bowlers.
    - **Feature Scaling** - Applied **MinMaxScaler** to normalize numerical variables.
2. **Model Selection and Training** The following machine learning models were tested:
    - **Neural Network (ANN)** - Multi-layered perceptron with ReLU activation.
    - **Loss Function** - Used Huber Loss to handle outliers effectively.
    - **Optimizer** - Adam optimizer for efficient learning.
    - **Train-Test Split** - 70% training data, 30% testing data.
3. **Model Performance** The final model was evaluated using **Mean Absolute Error (MAE)**:
    - **Mean Absolute Error (MAE)**: 19.48
    - **Validation Loss Improvement** - The model improved across 50 epochs, reducing loss from 21.57 to 18.99.

Key Insights from Model Performance

- The neural network performed well in predicting total scores with reasonable accuracy.
- Run progression in the last 5 overs and venue played a crucial role in determining scores.
- Categorical encoding helped improve prediction accuracy by allowing the model to learn team-specific patterns.
- Further tuning, such as additional features (e.g., pitch conditions, weather), could enhance accuracy.

This analysis provides valuable insights into IPL score prediction and demonstrates the impact of different match factors on final totals.
