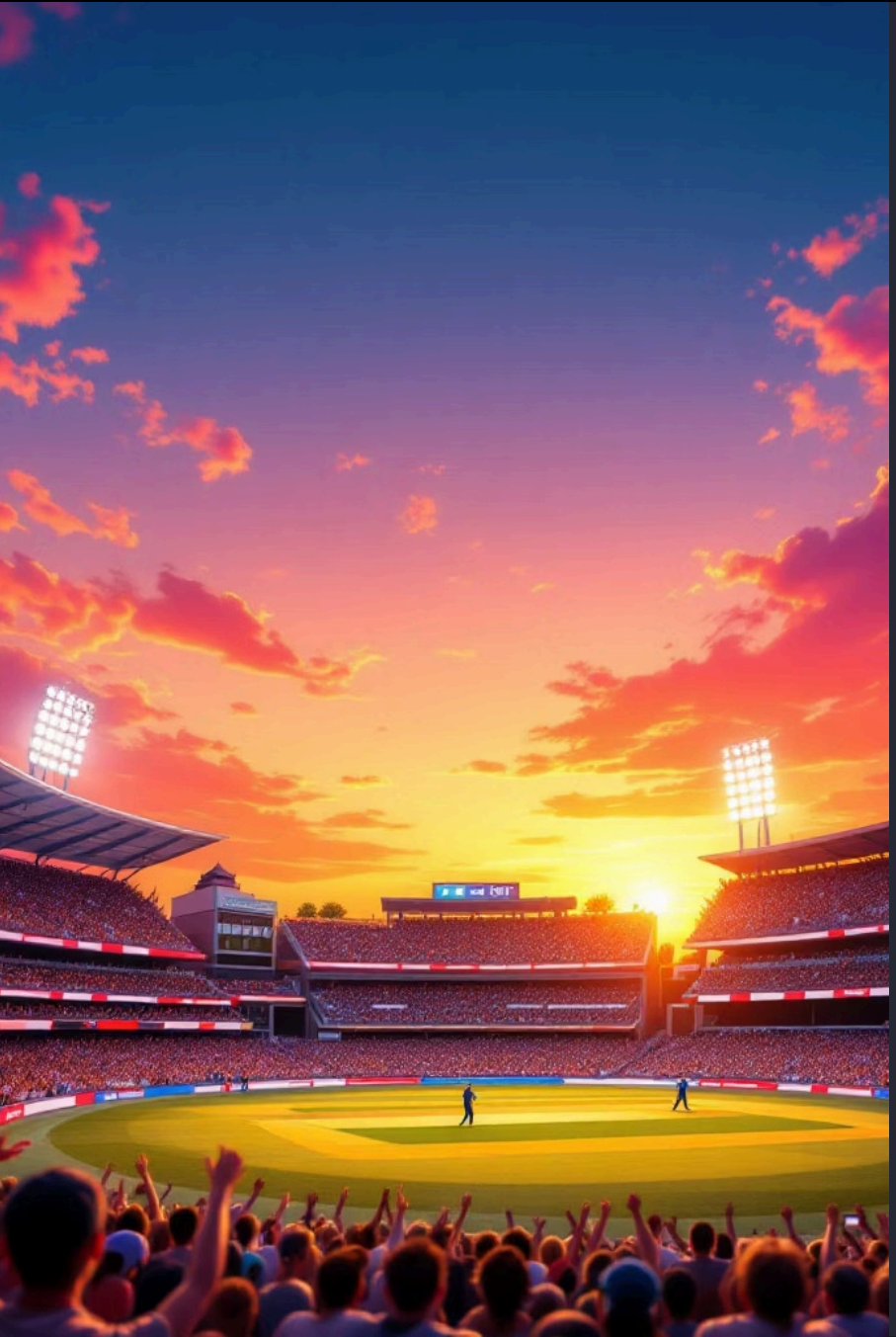


IPL Score Prediction Using Machine Learning

Problem Statement

In T20 cricket, predicting the final innings score during a live match is challenging due to dynamic match conditions.

- Predict final IPL score after 5 overs
- Use live match statistics
- Apply machine learning models
- Deploy the model using Streamlit

Dataset Description

- Source: IPL Ball-by-Ball Dataset
- Total Records: ~76,000 rows
- Each row represents one ball

Key Features Used:

- Batting Team
- Bowling Team
- Runs (current score)
- Wickets fallen
- Overs completed
- Runs in last 5 overs
- Wickets in last 5 overs

Target Variable:

- Total (Final Innings Score)

Data Cleaning Steps

01

Removed irrelevant columns:

- match id, date, venue
- batsman, bowler
- striker, non-striker

02

Selected only consistent IPL teams:

- 8 core teams retained
- Removed temporary teams

03

Removed early overs (< 5 overs):

- Avoid unstable early predictions
- Improve model reliability

04

Handled categorical variables:

- One-Hot Encoding for teams

Key Insights

Wickets Distribution

- Most innings have 0–4 wickets during early/mid overs
- Very few instances of 8–10 wickets

Final Score Distribution

- Most IPL scores lie between 140–190
- Very few extreme scores (<100 or >220)

Selected Features

- Batting Team (One-Hot Encoded)
- Bowling Team (One-Hot Encoded)
- Current Runs
- Wickets Fallen
- Overs Completed
- Runs in Last 5 Overs
- Wickets in Last 5 Overs

Total features after encoding: 21+

Models Trained

- Decision Tree Regressor
- Random Forest Regressor
- XGBoost Regressor
- Linear Regression
- Support Vector Machine
- K-Nearest Neighbors

Model Performance Comparison

Decision Tree	Train Score: 99.98%	Test Score: 86.29%
Random Forest	Train Score: 99.04%	Test Score: 93.07%
XGBoost	Train Score: 88.71%	Test Score: 84.84%

Final Model: Random Forest Regressor

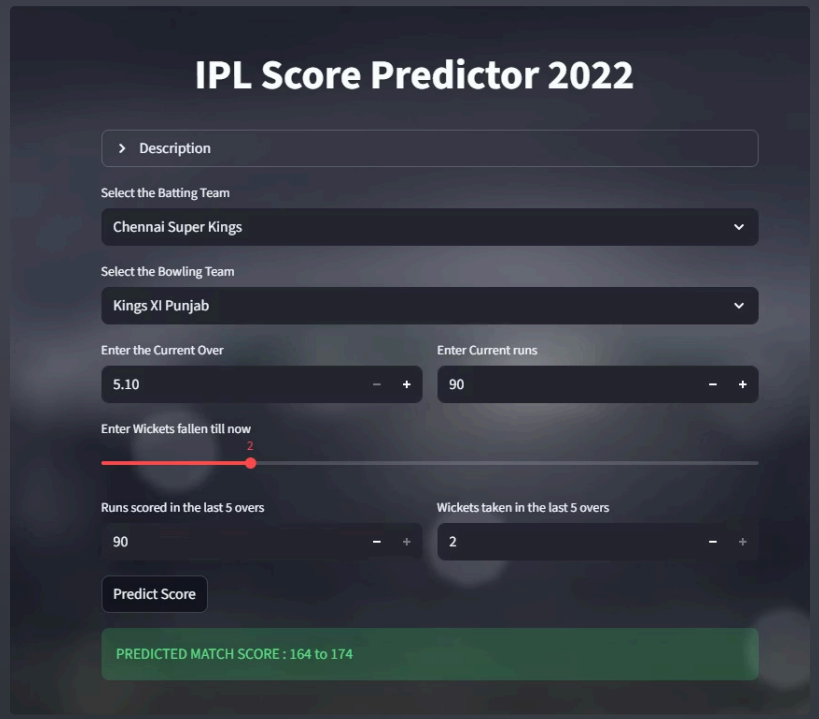
Reasons:

- Highest Test Accuracy (93.07%)
- Good Generalization
- Less Overfitting compared to Decision Tree
- Robust against noise

Evaluation Metric:

- R^2 Score

Random Forest provided best balance between bias and variance.



Web Application Features

- Select Batting Team
- Select Bowling Team
- Enter Current Overs
- Enter Current Runs
- Enter Wickets Fallen
- Enter Runs & Wickets in Last 5 Overs

Validation Implemented

- Overs must be ≥ 5
- Valid over format (6 balls per over)
- Batting and Bowling teams must differ
- Logical input constraints

Conclusion

- Successfully built an end-to-end ML pipeline
- Performed feature selection and data filtering
- Achieved 93% prediction accuracy
- Deployed model using Streamlit

The model effectively captures:

- Team impact
- Match progression
- Momentum factor