**Project Information**

Title : **IPL Match Winner Prediction**

Name : Dhiwagar K

# Introduction

Cricket, particularly the Indian Premier League (IPL), is one of the most popular sports events worldwide. Predicting match outcomes using machine learning has become an exciting challenge for researchers and analysts. This project leverages machine learning algorithms to predict the winner of IPL matches based on historical data. The report documents each phase of the project including data loading, cleaning, exploratory analysis, preprocessing, model training, evaluation, hyperparameter tuning, model comparison, and final conclusions.

# Data Loading and Understanding

The IPL dataset was loaded using Pandas. It contains match-related information such as team names, toss decisions, match winners, venue, and other attributes. Displaying the first few rows of the dataset helped in understanding the structure and data distribution.

Code Example:  
  
import pandas as pd  
df = pd.read\_csv('ipl\_dataset.csv')  
df.head()

Observations:

The dataset contains both categorical and numerical columns.

* Categorical features include 'team1', 'team2', 'toss\_winner', and 'venue'.
* Numerical features include 'id', 'season', and 'dl\_applied'.
* The target variable is 'winner'.

# 2. Data Cleaning

The dataset required cleaning before training machine learning models:

* Missing values were handled by removing or imputing them appropriately.
* Categorical columns such as team names and toss decisions were converted into numerical form using Label Encoding and One-Hot Encoding.
* Outliers or inconsistent records were identified and removed to avoid bias.

Example:  
  
from Sklenar. preprocessing import Label Encoder  
le = Label Encoder()  
df['team1'] = le.fit\_transform(df['team1'])

# 3. Exploratory Data Analysis (EDA)

Exploratory Data Analysis provided insights into match patterns and team performance:

* Bar charts were used to show the number of matches won by each team.
* Pie charts showed toss outcomes (bat/field decisions).
* Bar charts demonstrated city across matches.

Observations:

* Teams like Mumbai Indians and Chennai Super Kings have consistently high win rates.
* Toss outcomes have some influence on match results but are not the only factor.

# 4. Data Preprocessing

Before model training, the data was preprocessed:

* Features (X) and target labels (y) were separated.
* The dataset was split into training (80%) and testing (20%) sets.
* Numerical features were scaled where necessary to ensure fair treatment across models.

# 5. Model Training

Multiple machine learning models were trained to compare their performance:  
 Logistic Regression  
 Support Vector Machine (SVM)  
 K-Nearest Neighbors (KNN)  
 Decision Tree  
 Random Forest  
 XGBoost

Each model was evaluated using cross-validation and test data predictions. The evaluation metrics included Accuracy, Precision, Recall, and F1 Score.

# 6. Model Evaluation

The models were compared using accuracy, precision, recall, and F1-score. Confusion matrices were generated for each model to visualize classification performance. Random Forest and XGBoost generally outperformed the other models in terms of F1 Score.

# 7. Hyperparameter Tuning

The best-performing model was tuned using RandomizedSearchCV and GridSearchCV. This process involved selecting the optimal hyperparameters like max\_depth, n\_estimators, and learning\_rate. After tuning, the XGBoost model showed significant improvement in performance.

# 8. Model Comparison

A comparison table was created to summarize all models:  
- Logistic Regression: Performed moderately but lacked in capturing complex relationships.  
- SVM: Worked well with linear data but not optimal for this dataset.  
- KNN: Sensitive to parameter 'k' and less effective with large feature sets.  
- Decision Tree: High variance and overfitting issues.  
- Random Forest: Robust and performed consistently well.  
- XGBoost: Best performer after hyperparameter tuning.

# 9. Conclusion

This project demonstrates the use of machine learning to predict IPL match outcomes.  
Ensemble models, especially XGBoost, performed better than simpler algorithms.  
After hyperparameter tuning, XGBoost achieved the best balance of accuracy, precision, recall, and F1-score.  
This work highlights the importance of data cleaning, preprocessing, and continuous retraining for reliable sports analytics.