PSL Match and Player Prediction System Using Machine Learning

Ammar Jamil
Department of Computer Science
Bahria University
Islamabad, Pakistan

Wajahat Gul
Department of Computer Science
Bahria University
Islamabad, Pakistan

***Abstract*—Cricket is one of the most popular sports in Pakistan, and the Pakistan Super League (PSL) has become a major event for fans, analysts and media. People are always interested in knowing which team has a higher chance of winning, how many runs a batsman may score, and how many wickets a bowler might take. This paper presents a simple machine-learning-based PSL prediction system that uses past match data to forecast match outcomes and individual player performance. The system uses a cleaned and structured dataset containing PSL seasons from 2016 to 2025. Basic features such as teams, venue, toss result, player form and previous performance are used to train classification and regression models. The system predicts (1) which team has a higher chance of winning, (2) expected runs of a batsman and (3) expected wickets of a bowler. The predictions are made available through an interactive HBL PSL-themed website interface. The aim of this project is not to guarantee perfect predictions, but to show how machine learning can help understand cricket patterns in a simple and practical way using commonly available PSL data.**

I. Introduction

Cricket is deeply connected with Pakistani culture, and the Pakistan Super League (PSL) has gained a huge following in a short time. Fans, TV channels and social media users regularly try to guess match results and player performance. Many people make their predictions based on emotions, personal opinions or limited information. However, machine learning allows us to study real data and find patterns that humans usually miss.

Most discussions among cricket fans are about three main things: Who will win the match? Which batsman will perform well? And which bowler will take important wickets? These questions can be explored using past PSL data, as cricket performance usually follows some trends related to team strength, venue, player form and toss conditions.

Although some websites provide cricket analysis, there is no simple ML-based prediction tool focused on PSL that can give easy and understandable predictions for Pakistani users. Creating such a system is useful for students, analysts, casual fans and local sports platforms.

Machine learning gives a good approach because cricket data has many factors, and it is hard to create manual rules for all situations. ML models can learn patterns from past matches and then give reasonable predictions for upcoming games. Our project focuses on a simple version of this idea.

In this work, we develop a PSL Match and Player Prediction System that predicts:

- which team is likely to win a match,
- how many runs a batsman may score,
- how many wickets a bowler may take.

The predictions are presented through an interactive HBL PSL-themed website that allows users to select match features and receive data-driven predictions. The goal is not to replicate full professional analytics but to build a student-level ML system that uses available PSL data effectively.

II. Problem Definition and Motivation

A. Problem Overview

The main problem is that PSL fans, especially in Pakistan, usually depend on guesswork when thinking about match results or player performance. There is no easy tool that uses past PSL records to give logical predictions. People want simple but data-based answers, not complex expert analysis.

B. Technical Problem Statement

From a technical perspective, the system performs three tasks:

- **Team Win Prediction (Classification):** Predict whether Team A or Team B will win.
- **Batsman Score Prediction (Regression):** Predict expected runs based on form and past performance.
- **Bowler Wicket Prediction (Regression):** Predict expected wickets of a bowler.

The input includes features from past PSL seasons such as match venue, toss result, teams playing, batsman stats and bowler stats. The output is a numerical or categorical prediction displayed through an HBL PSL-branded web interface.

C. Real-World Impact

This project can help:

- **Fans:** Understand cricket more logically instead of emotional guessing.
- **YouTubers / Analysts:** Create better content with data-backed predictions.
- **Fantasy League Users:** Make better choices for selecting players.
- **Students:** Learn practical ML concepts using a dataset related to Pakistan.

D. Stakeholders

Key beneficiaries include:

- PSL fans and social media creators,
- sports bloggers and analysts,
- fantasy cricket players,
- students learning data science,
- cricket websites wanting prediction features.

E. Motivation

We chose this topic because:

- PSL is highly popular in Pakistan and easy to relate with,
- cricket datasets are easy to understand and work with,
- predictions like "team win", "runs" and "wickets" are simple ML tasks,
- the project fully fits the requirements of Assignment 3 and Assignment 4,
- the topic is interesting, practical and doable within 20 days.

F. Why did we choose this problem?

We selected PSL match and player prediction because it combines strong educational value with clear social and economic relevance. Socially, cricket is Pakistan's most-followed sport; accurate, data-driven insights help fans and content creators make more informed commentary. Industrially, sports analytics is a growing area broadcasters, fantasy-sports platforms and sports media companies value automated prediction tools for content, highlights and recommendation services.

Personally, the topic is highly motivating for the student team: data is locally relevant, readily available, and the problem maps well to standard ML learning objectives. Economically, modest improvements in prediction quality can support fantasy-league decisions and advertising/engagement strategies for sports platforms. Together, these factors make PSL prediction an ideal student project with practical impact.

G. Why machine learning is suitable and why not rule-based (if-else) systems

Cricket match outcomes and player performance depend on many interacting variables (team composition, recent form, venue, toss, match context, opposition, and stochastic events). A rule-based system (if-else) would require enumerating a combinatorial number of conditions and tuning many thresholds, which quickly becomes unmanageable and brittle.

Machine learning is suitable because:

- ML learns complex, non-linear relationships and interactions from historical data without hand-crafted rules.
- Models can generalize from past seasons to unseen matchups and can be retrained as new data arrives.
- ML supports probabilistic outputs (winning probabilities, expected runs) rather than brittle binary rules.

Therefore, ML provides scalability, adaptability and a principled way to estimate prediction uncertainty that if-else rules cannot provide robustly.

III. Machine Learning Feasibility

A. Problem Classification

The system uses:

- **Binary / Multiclass Classification:** For predicting match winner (or multi-class for win/loss/tie/no-result).
- **Regression:** For predicting runs and wickets.

B. Why ML is Suitable

Machine learning works well because:

- Cricket performance depends on many factors, not simple rules.
- ML learns hidden patterns from data that humans cannot easily see.
- PSL has enough past matches to train basic models (with careful validation).

C. Data / Dataset Summary

The dataset contains PSL matches from 2016–2025, including:

- match results,
- team names,
- venue,
- toss winner and decision,
- player stats: runs, balls, wickets, overs, strike rate, economy.

D. ML Feasibility

Problem types:

- Team Win Prediction: **Binary / Multiclass Classification**.
- Batsman Runs Prediction: **Regression**.
- Bowler Wickets Prediction: **Regression**.

Target outputs:

- Win probability for each team (e.g., Lahore Q: 62%).
- Expected runs (numerical) or interval (e.g., 34–42).
- Expected wickets (numerical or distribution, e.g., 0–2).

Input features (explicit list at least 10):

1) Home/Away / Neutral venue (categorical)
2) Venue name (categorical)
3) Team A recent form (wins in last 5 matches) (numeric)
4) Team B recent form (wins in last 5 matches) (numeric)
5) Head-to-head win percentage between teams (numeric)
6) Toss winner and toss decision (bat/bowl) (categorical)
7) Average team batting score at venue (numeric)
8) Key batsman current form (e.g., average in last 5 innings) (numeric)
9) Key bowler economy/strike rate (numeric)
10) Player availability/final XI indicators (binary flags)
11) Day/Night match (categorical)
12) Pitch indicator or proxy (batting/pacing tendency) (categorical)
13) Weather condition flag or rain probability (numeric) if available

Possible datasets / data sources:

- PSL historical match scorecards and ball-by-ball data (ESPN Cricinfo).
- Public Kaggle cricket/PSL datasets (match-level and player-level).
- Official PSL statistics pages and CSV exports (if available).
- Third-party sports data APIs (e.g., CricAPI) for programmatic access.
- Manual local collection and cleaning of PSL match CSVs.

Data collection challenges:

- Inconsistent team / player name spellings across seasons.
- Missing fields such as pitch report or precise weather history.
- Incomplete final-XI information prior to match start.
- Licensing or scraping restrictions for some sources.

Ethical and privacy considerations:

- Use only public, non-sensitive player data (public performance stats).
- Be transparent about prediction limitations; include disclaimers.
- Avoid promoting gambling or irresponsible betting; explicit safety notice recommended.

Risk of bias and mitigation:

- Historical data bias (e.g., established players overrepresented).
- Venue bias (some grounds host more matches for particular teams).
- Mitigation: time-aware cross-validation, re-weighting, careful feature selection, and repeated experiments.

Evaluation metrics:

- *Classification (match winner):* Accuracy, Confusion Matrix, Precision, Recall, F1-score(for probabilties).
- *Regression (runs/wickets):* Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), for predicted ranges.
- *Practical metrics:* Ranking metrics for fantasy selection (top-K recall).

IV. Proposed Methodology

Fig. 1 shows the system workflow. We clean the PSL dataset, extract useful features, train ML models and finally predict match winner, batsman runs and bowler wickets through an HBL PSL-themed website.

A. Workflow Steps

- **Step 1: Data Collection** gather match and ball-by-ball data from ESPN Cricinfo, Kaggle and PSL official sources.
- **Step 2: Data Cleaning** remove missing values, resolve name inconsistencies, handle missing XI information.
- **Step 3: Feature Extraction** venue, toss, previous match stats, rolling averages, head-to-head stats.
- **Step 4: Model Training** classification for match winner; regression for runs/wickets.

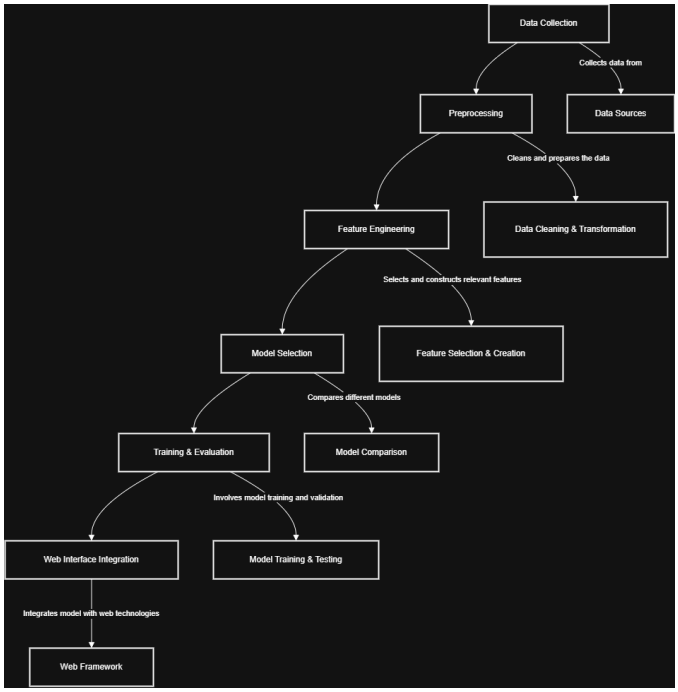

Fig. 1: High-level pipeline: Data Collection → Preprocessing → Feature Engineering → Model Selection → Training & Evaluation → Web Interface Integration

- **Step 5: Model Evaluation** compute metrics, perform calibration and error analysis.
- **Step 6: Web Interface Development** create HBL PSL-themed website where users can input match features and receive predictions.
- **Step 7: Integration** connect trained ML models to website backend for real-time predictions.

B. Methodology steps

V. HBL PSL Website Interface

The prediction system will be integrated into an HBL PSL-themed website that provides an intuitive and visually appealing interface for users. The website will feature:

A. Website Structure

- **Home Page:** Welcome screen with HBL PSL branding, brief introduction to the prediction system, and navigation to three main prediction modules.
- **Match Winner Prediction Page:** Users select two teams from dropdowns, choose venue, input toss details, and click ”Predict Winner” to receive win probability predictions with confidence intervals.
- **Batsman Performance Page:** Users select a batsman, opponent team, venue, and recent form indicators. The system predicts expected runs with a range estimate.
- **Bowler Performance Page:** Users select a bowler, opponent team, venue, and bowling conditions. The system predicts expected wickets.

TABLE I: Workflow steps and tools

Step	Description & Tools/Methods
Data Collection	Collect PSL match and player stats from ESPN Cricinfo, Kaggle, and official PSL sources. Tools: Python (requests, BeautifulSoup), CSV exports.
Preprocessing	Clean names, handle missing values, normalize numeric fields, one-hot encode categorical variables. Tools: Pandas, NumPy.
Feature Engineering	Create rolling-form stats (last N matches), head-to-head features, venue aggregates, player-impact features. Tools: Pandas.
Model Selection	Try Logistic Regression, Random Forest, XGBoost for classification; Random Forest, Gradient Boosting, and Linear models for regression. Tools: Scikit-learn, XGBoost.
Training	Time-aware cross-validation, hyperparameter tuning (GridSearch/RandomSearch). Tools: Scikit-learn, Optuna.
Evaluation	Evaluate with metrics above; calibration checks and error analysis. Tools: Scikit-learn, matplotlib.
Web Interface	Build HBL PSL-themed website with HTML/CSS/JavaScript frontend; Flask/Django backend for model serving; interactive feature selection forms.
Integration	Connect trained models to web backend; implement RESTful API endpoints for predictions; add visualization charts using Chart.js.

- **Results Display:** Clean, visual presentation of predictions with percentage probabilities, confidence ranges, and supporting statistics displayed in charts and graphs.

B. User Interaction Flow

1) User navigates to desired prediction type (match/batsman/bowler).
2) User fills in required feature inputs via dropdowns and forms.
3) User clicks prediction button.
4) Backend processes features through trained ML model.
5) Results are displayed with visual indicators (bar charts, probability meters).
6) User can modify inputs and re-predict or navigate to other modules.

C. Technical Implementation

- **Frontend:** HTML5, CSS3, JavaScript with Bootstrap for responsive design and HBL PSL color scheme (green and yellow theme).
- **Backend:** Flask or Django framework to serve ML models and handle prediction requests.
- **Visualization:** Chart.js for interactive graphs showing team form, head-to-head records, and prediction confidence.
- **Model Integration:** Trained scikit-learn/XGBoost models serialized with pickle/joblib and loaded by backend API.

D. Website Mockup Description

Since we are building a web-based system, the interface will be designed to match HBL PSL branding with:

- Clean, modern layout with PSL team colors and logos
- Large, easy-to-read buttons and dropdowns
- Real-time prediction results with probability bars
- Historical performance charts and trend graphs
- Mobile-responsive design for accessibility on all devices
- Disclaimer section explaining prediction limitations

E. GUI Mockup Images

Create the mockups in Figma or draw.io and export high-resolution PNGs. Place them in the same folder as this LaTeX file and include the following files:

- `home.png`
- `match.png`

Embed them with the figures below (replace filenames as needed):

Fig. 2: HBL PSL website home page mockup showing main navigation and branding

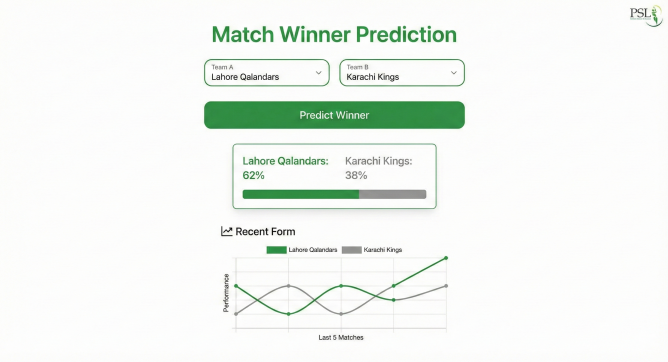

Fig. 3: Match prediction interface mockup with team selection and results display

VI. Expected Outcomes

We expect the system to:

- Provide reasonable predictions for PSL match winners with probability estimates,

- Estimate batsman runs and bowler wickets based on past trends and current form,
- Present predictions through an accessible HBL PSL-themed website,
- Offer interactive features allowing users to adjust match parameters,
- Help users understand basic cricket analytics through visual data presentation,
- Serve as an educational tool for understanding ML applications in sports.

VII. Conclusion

This project demonstrates how machine learning can be applied to PSL data to predict match outcomes and player performance through an accessible web interface. Although cricket is unpredictable, the system provides simple and helpful estimates based on historical data patterns. The HBL PSL-themed website makes these predictions available to fans, analysts, and casual users in an amazing format.

The project is practical for a 6th-semester student group and fits the Pakistani context well. With careful preprocessing, feature engineering, and proper evaluation (time-aware validation and calibration), this system provides value for PSL followers while demonstrating core machine learning concepts.

References

[1] M. Ankit and R. Singh, "Cricket match outcome prediction using machine learning", *International Journal of Computer Applications*, 2020.

[2] S. Shah, A. Naveed, "Data-driven analysis of PSL performance trends", *Journal of Sports Analytics*, 2022.

[3] K. Prasad, "Predicting player performance in T20 cricket", *IEEE Access*, 2021.

[4] F. Ahmed and M. Khan, "Team performance modelling in cricket using ML techniques", *Procedia Computer Science*, 2020.

[5] ESPN Cricinfo, PSL historical datasets (scorecards and ball-by-ball data), 2016–2025. [Online]. Available: https://www.espncricinfo.com

[6] Kaggle Datasets, "PSL Complete Dataset 2016-2025", Kaggle. [Online]. Available: https://www.kaggle.com/datasets/zeeshanahmad124586/psl-complete-dataset-2016-2025

[7] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[8] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer, 2009.