Powerplays and Predictions: Unlocking IPL Intelligence

Submission 02: Machine Learning using Scikit-learn

Ashwin Shastry Paturi, Akarsh Lakshmana, Divakar Reddy Ravi, Sara Verkiyani

Team-09, CS506\_02\_IN: Programming for Computing, Spring 2025, MS-Computer Science, City University of Seattle

[paturiashwinshastry@cityuniversity.edu](mailto:paturiashwinshastry@cityuniversity.edu), [lakshmanaakarsh@cityuniversity.edu](mailto:lakshmanaakarsh@cityuniversity.edu), [ravidivakarreddy@cityuniversity.edu](mailto:ravidivakarreddy@cityuniversity.edu), [verkiyanisara@cityuniversity.edu](mailto:verkiyanisara@cityuniversity.edu)

**Abstract**

This module focuses on developing machine learning models using the Scikit-learn library to analyze Indian Premier League (IPL) match data. After previously performing exploratory data analysis using Pandas, we now build and evaluate predictive models to forecast match outcomes and cluster team performances. We apply supervised learning techniques like logistic regression, decision trees, and random forests to classify match winners. We also employ K-Means clustering to uncover latent performance groupings among IPL teams. This step-by-step tutorial includes data preprocessing, feature engineering, training/testing split, model evaluation using accuracy and confusion matrices, and unsupervised segmentation of team strategies. The analysis demonstrates how Scikit-learn streamlines model development for real-world sports analytics.

**Keywords:** Scikit-learn, IPL, Classification, Clustering, Machine Learning, Python, Toss Outcome, KMeans, Team Performance, T20 Cricket

1. **Introduction**

Following the foundational data exploration in the Pandas module, this submission delves into practical machine learning applications using Scikit-learn. IPL, as a competitive and data-rich T20 league, offers meaningful challenges for classification and clustering tasks.

In this phase, we explore:

* Predicting match winners using classification models.
* Estimating toss impact with binary classification.
* Identifying team performance profiles through clustering.

Scikit-learn’s simple API for building pipelines, training models, and evaluating them makes it the perfect choice for applied machine learning over structured datasets like IPL match records.

1. **Overview of the Dataset**

The dataset contains match-level data for some IPL seasons. It has over 70 columns and includes information on:

* **Team-level data:** home team, away team, scores, captains, playing XIs
* **Match result information:** winner, toss decision, result type, player of the match
* **Time and place:** start/end dates, venue name, umpires
* **Scoring statistics:** runs, wickets, boundaries
* **Metadata:** result descriptions, match days, super over information

This dense dataset provides an ideal environment to work with and demonstrate Pandas' functionality, ranging from basic I/O to more involved group-by operations and aggregations.

1. **Scikit-learn Concepts and Their Applications**

**3.1 Data Preprocessing:** We cleaned the dataset, dropped null values, and created a binary target for toss-win impact. One-hot encoding was applied using `get\_dummies()`, and target labels were encoded using `LabelEncoder()`.

A screenshot of a computer

AI-generated content may be incorrect.

**3.2 Train-Test Split:** We used an 80-20 split to divide the data into training and test sets.

A black text on a white background

AI-generated content may be incorrect.

**3.3 Classification Models**

* Logistic Regression

A screenshot of a graph

AI-generated content may be incorrect.

* Decision Tree

A screenshot of a computer

AI-generated content may be incorrect.

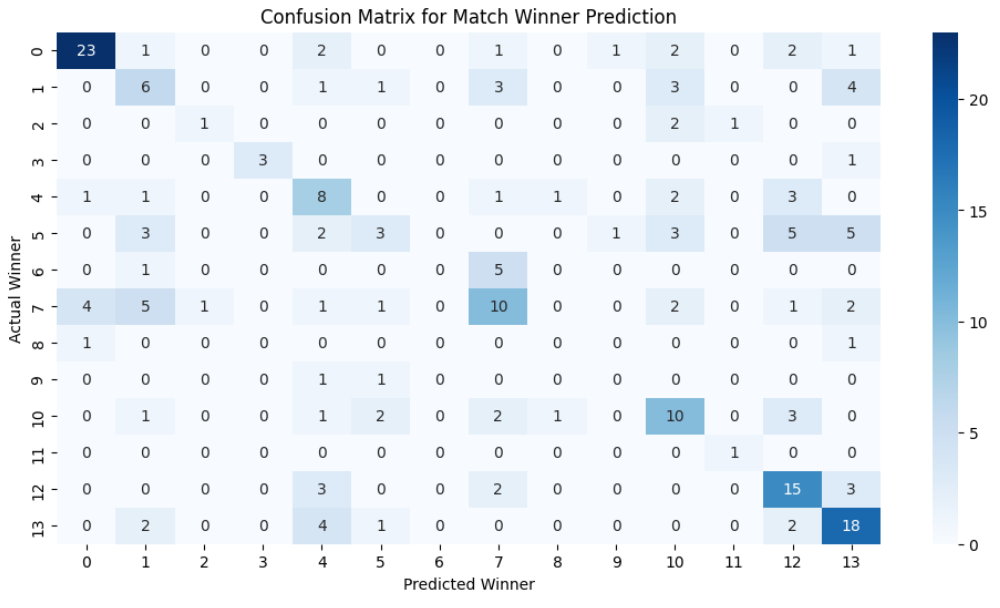
* Random Forest

A screenshot of a graph

AI-generated content may be incorrect.

Each model was evaluated using accuracy, a confusion matrix, and an F1 score.

**3.4 Evaluation Metrics:** We compared model accuracy and used Scikit-learn’s classification report for deeper insight into per-class performance.



**3.5 K-Means Clustering:** We averaged per-team performance (runs, wickets, boundaries), applied feature scaling, and K-Means to group teams into 3 clusters. This revealed high-performing vs. defensive team clusters.

**A diagram of a number of dots

AI-generated content may be incorrect.**

1. **Summary of Key Insights**

- Logistic Regression performed well even with basic encoding.

- Random Forest yielded the highest accuracy due to its ensemble nature.

- Clustering successfully identified team types based on numerical metrics.

**5. Learning Outcomes**

At the end of this module, we,

* Learned to implement Scikit-learn classification and clustering workflows.
* Understood the importance of feature encoding and standardization.
* Developed skills in evaluating and comparing models.
* Gained exposure to real-world sports analytics use cases.

**REFERENCES**

McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media.

IPL T20 | Indian Premier League Official Website. (n.d.). <https://www.iplt20.com/>

Indian Premier League (IPL) - All seasons. (2024, May 13). Kaggle. <https://www.kaggle.com/datasets/rajsengo/indian-premier-league-ipl-all-seasons>

Using Matplotlib — Matplotlib 3.10.1 documentation. (n.d.). <https://matplotlib.org/stable/users/index.html>

pandas documentation — pandas 2.2.3 documentation. (n.d.). <https://pandas.pydata.org/docs/>

Seaborn: statistical data visualization — seaborn 0.13.2 documentation. (n.d.). <https://seaborn.pydata.org/>

Geron, A. (2019). \*Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow\* (2nd ed.). O’Reilly Media.

Chollet, F. (2018). \*Deep Learning with Python\*. Manning Publications.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). *Scikit-learn: Machine learning in Python*. Journal of Machine Learning Research, 12, 2825–2830. [https://scikit-learn.org](https://scikit-learn.org/)

McKinney, W. (2010). *Data structures for statistical computing in Python*. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56). [https://pandas.pydata.org](https://pandas.pydata.org/)