

PCON's Annual Machine Learning Hackathon '26

January 1, 2026

Welcome to **PCON's Annual Machine Learning Hackathon '26**, a 48-hour challenge.

Rules

1. All solutions must be developed strictly during the hackathon period.
2. Participants may use any programming language, framework, or machine learning library.
3. Each participant must work on **only one** problem statement.
4. Participants must create a **new GitHub repository** for the hackathon submission, and all work must be original.
5. Commit messages should be clear and meaningful.
6. No commits are allowed after the submission deadline.
7. The project repository must contain a **README** including:
 - Setup and execution instructions
 - Explanation of features and approach
 - Model details and evaluation methodology
8. Submissions must strictly adhere to the selected problem statement.

Judging Criteria

Submissions will be evaluated based on the following criteria:

1. **Model Performance (50%):** Accuracy and effectiveness of the model based on the provided evaluation metric.
2. **Creativity & Approach (20%):** Innovation in feature engineering, modeling strategy, and problem-solving.
3. **Explainability & Clarity (15%):** Clear explanation of model choices, features, and results.
4. **Code Quality (15%):** Clean, readable, modular, and well-documented code.

0.1 Problem Statement 1: Classification

26 me to duniya khatam hee :(

0.1.1 Problem Description

Build a machine learning model to classify whether a team will win at an event (**1 for winner, 0 otherwise**). Use creative feature engineering to uncover patterns in team dynamics, skills, and behavior. You may earn or lose *PCOINS* based on predictions from your model.

0.1.2 Dataset

You are provided with **50,000 entries** from several events across colleges, including the following features:

- **leetcode_hours (avg)**: Average hours spent on LeetCode.
- **leetcode_problems (avg)**: Average problems solved on LeetCode.
- **team_size**: Number of team members.
- **experience**: Experience level (categorical: **beginner**, **intermediate**, **expert**).
- **branch**: Engineering branch (e.g., CSE, Mechanical).
- **gaming_hours**: Hours spent gaming.
- **instagram_hours**: Hours spent on Instagram.
- **social_skill_points**: Score representing social skills.
- **connections_among_seniors**: Number of connections with seniors.
- **connections_with_faculty**: Number of connections with faculty.
- **connections_among_juniors**: Number of connections with juniors.
- **have_freshman**: Boolean indicating if the team has freshmen.
- **all_freshman**: Boolean indicating if the team consists only of freshmen.

The dataset also includes a binary target variable indicating whether the team won the event.

0.1.3 Objectives

Develop a good enough classifier (e.g., logistic regression, random forest, or neural nets).

0.1.4 Evaluation

Use eval.csv (it contains 10K entries with the above qualities). Submit your predictions as a single CSV file with 10001 lines.

Submission Format

- First line: `output`
- Followed by 10,000 binary predictions (0 or 1)

Scoring Rules

Your model must predict whether a given team will win or lose the event. You'll earn PCOINS for correct predictions according to the following metrics:

Notation:

- tp (True Positives)
- fp (False Positives)
- tn (True Negatives)
- fn (False Negatives)

Scoring Function:

$$\text{Precision} = \begin{cases} \frac{tp}{tp + fp}, & \text{if } tp + fp > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\text{Recall} = \begin{cases} \frac{tp}{tp + fn}, & \text{if } tp + fn > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\text{F1-score} = \begin{cases} \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, & \text{if } \text{Precision} + \text{Recall} > 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\text{Score} = \text{F1-score} \times 10000$$

Your final score will be the sum of all PCOINS earned.

0.2 Problem Statement 2: Regression

Hum pe to ho ji naww :O

0.2.1 Problem Description

Engineer a regression model to predict the number of participants attending tech fest events. Use creative feature engineering to uncover patterns in event appeal, logistics, and student behavior.

Think: Does a famous guest boost turnout? How does weather or event timing influence participation?

0.2.2 Dataset

You are provided with **50,000 entries** from various tech fest events across colleges, including the following features:

- **event_type:** Event category.
- **guest:** Guest speaker type.
- **organising_department:** Hosting department (e.g., CSE, ECE, ME, MM, EE, CE, ECM, PIE).
- **timing:** Time of day (Morning, Afternoon, Evening, Late Night).
- **day_of_week:** Day type (Weekday, Weekend).
- **promotion_level:** Social media reach.
- **event_duration:** Event length in hours.
- **venue_capacity:** Maximum venue capacity.
- **registration_fee:** Registration cost (INR).
- **social_media_buzz:** Number of mentions/shares.
- **concurrent_events_count:** Number of overlapping events.
- **weather_condition:** Weather condition (Sunny, Cloudy, Rainy, Stormy).

The target variable is **recorded_participant_count**, an integer representing the number of participants.

0.2.3 Objectives

Develop a good enough regression model that can predict the number of participants attending a tech fest event.

You are expected to handle categorical and numerical variables appropriately, and minimize prediction error in order to maximize the total *PCOINS* earned by the the scoring function below.

0.2.4 Evaluation

An evaluation dataset `eval.csv` containing **10,000 entries** is provided. Submit your predictions as a single CSV file containing predicted participant counts for each entry.

Submission Format

Submit the prediction file in CSV format with exactly **10,000 rows**, each containing a single predicted participant count.

Scoring Rules

Your model must predict the number of participants attending a tech fest event. You'll earn PCOINS for correct predictions according to the following metrics:

Notation:

- y : True attendance value (actual number of attendees)
- \hat{y} : Predicted attendance value (your model's prediction)
- capacity: Maximum capacity of the venue
- $\tau = 0.1$: Tolerance parameter
- e : Prediction error

Scoring Function:

$$\tau = 0.1$$

$$e = \frac{|y - \hat{y}|}{\text{capacity}}$$

$$\text{PCOINS} = \frac{1}{1 + \left(\frac{e}{\tau}\right)^2}$$

The PCOINS value ranges from 0 to 1, where 1 represents a perfect prediction and values closer to 0 indicate larger errors. Your final score for this problem is the **sum of all PCOINS** earned across all predictions.