

Machine Learning Hackathon '26

@>--;--

authored by

Ayush Jayaswal
Chandrima Hazra

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. Organizers reserve the right to audit submissions and request source code at any time.
4. **GPU usage is prohibited.** All training must be performed on CPU only to ensure fair competition.
5. Please structure your solution neatly as notebooks or script files.

Submission Guidelines

1. For each submission, you'll receive a score depending on how accurate your predictions are.
2. Participants cannot make more than **3 submissions** for either of the two problems.
3. The **best score out of 3** submissions is reflected on the leaderboard.
4. The **final total score** you get is the number of **PCOINS** you've earned.
5. You must make your submission [here](#)

Problem 1: Classification

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'll earn *PCOINS* based on predictions from your model.

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. The dataset can be [found here](#).

Objectives

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

Evaluation

Use [eval1.csv](#) (it contains 10K entries with the above qualities). Submit your predictions as a single CSV file with exactly 10,000 lines of binary output.

Submission Format

- No headers
- There must not be any alphanumeric characters other than 0 and 1
- The *i*'th entry in the file must correspond to the *i*'th entry of the eval set

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:

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

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

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

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

$$\text{PCOINS earned} = \text{Score}$$

Problem 2: Regression

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?

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.

The dataset can be [found here](#).

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.

Evaluation

An evaluation dataset [eval2.csv](#) containing **10,000 entries** is provided. Submit your predictions as a single CSV file with exactly **10,000 rows** of integer output, each row being predicted participant counts for each entry.

Submission Format

- No headers
- There must not be any alphanumeric characters other than 0 and 1
- The i'th entry in the file must correspond to the i'th entry of the test set

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:

$$\tau = 0.1$$

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

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

Each prediction receives a Score between 0 and 1, where 1 indicates a perfect prediction and values closer to 0 indicate larger errors.

Your final score (**PCOINS**) is the sum of all Scores.