# UE23CS352A: Machine Learning Hackathon

# Hackman

HARSHA DIWAKAR (PES1UG23AM110)

HARINI A (PES1UG23AM109)

RITVIK SAYE (PES1UG23AM085)

ARYAN S (PES1UG23AM069)

## 1. Key Observations

Developing the Hangman agent was challenging due to the need to balance probabilistic reasoning and adaptive learning. Designing effective state representations and tuning the reward structure required several refinements. The key insight was that combining HMM-based contextual predictions with RL-based decision-making significantly improved performance and adaptability.

## 2. Strategies

### HMM Design:

The HMM modelled letter-to-letter transitions using the training corpus. Transition probabilities captured contextual dependencies (e.g., likelihood of one letter following another), while emission probabilities were updated dynamically as letters were revealed. This enabled the agent to make context-aware predictions.

### RL Design:

- **State:** Represented the current word pattern, guessed letters, and remaining attempts.
- **Actions:** Selection of the next letter to guess.
- **Rewards:** Positive for correct guesses and wins; negative for wrong or repeated guesses. This setup reinforced accurate, non-redundant decision-making.

## 3. Exploration vs. Exploitation

An ε-greedy strategy was applied. Initially, higher exploration encouraged discovering effective letter patterns. As learning progressed, ε decayed gradually, favoring exploitation of high-value actions learned through experience.

## 4. Future Improvements

With additional time, the agent could be enhanced by:

- Incorporating deep RL for function approximation.
- Expanding HMM states using character n-grams for richer context.
- Adaptive reward scaling to better handle rare or long words.
- Parallelizing self-play for faster convergence.