Discussion —— Shangming Zhuo

1.Play with your sorting hat. Are all 10 questions important to create the sorting hat? If you were to remove some questions to improve user experience, which questions would you remove and justify your answer.

Not necessarily. For example, among the 10 questions, ones like "Preferred pet?" and "Favorite subject?" are not strongly related to house assignment. If I were to remove some questions, I would likely remove these two. Additionally, even among the remaining questions, their relevance to house classification varies—some are strongly correlated, while others are weakly correlated.

2.If you were to improve the sorting hat, what technical improvements would you make? Consider:

2.1 How could you improve the model's accuracy or efficiency?

First, during the training data collection phase, I would consider gathering more responses from users who already know their house assignments. Second, I would apply techniques such as Mutual Information and Information Gain to quantify the correlation between each question and the sorting result, and then adjust the influence of each question in the model based on its relevance (mutual information score).

2.2 What additional sensors or hardware could enhance the user experience?

For hardware, I would consider adding an RGB animation LED that lights up in the corresponding house color when the sorting result is revealed. Additionally, I might use a larger screen to display the questions, as the current text is too small and can be difficult to read.

For sensors, I would add a camera sensor to detect users' gaze and facial expressions, enabling the Sorting Hat to assign probabilistic weights to options based on hesitation or eye movements, thus enhancing both accuracy and immersion.

2.3 Does decision tree remain suitable for your choice of new sensors? If yes, carefully justify your answer. If not, what ML model would you use and explain why.

Decision trees are not well-suited for handling continuous and high-dimensional inputs like gaze direction or facial expressions, as they rely on axis-aligned splits and struggle to capture subtle, non-linear patterns. Such models tend to either oversimplify the decision boundaries or overfit when dealing with rich sensor data. A lightweight neural network (e.g., MLP or CNN) is more appropriate, as it can learn smooth and complex relationships from these inputs, and can be optimized for embedded devices using TinyML techniques.