

Lab 3: ESP32 Sorting Hat

Implementation:

The responses for the 10 questions will first be collected and used to train the decision tree. After the training, the model will be converted into the header format then it can be imported into the Arduino code.

For the codes filled in the Arduino, the button pressed variable will be set to true and the corresponding index stored with the answer.

Discussion:

- 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.

For classification, they should be equally important. But for the user experience, I felt like the question 10 is too vague, and the question 5 seems too subjective.

- If you were to improve the sorting hat, what technical improvements would you make? Consider:
 - How could you improve the model's accuracy or efficiency?

To improve the accuracy, I can enrich the dataset by collecting more responses and increasing the depth of the tree, or changing other tunable parameters. For efficiency, I can reduce the number of questions to reduce the branches and the leaves.

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

I can integrate a microphone to let the user choose options by voice, and also a gyroscope to watch the movement of the hat to start the sorting process.

- 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.

The raw data for voice data is not suitable for decision trees, because they are continuous and time-dependent. But pre-processing can be applied to convert the data to discrete inputs. We can also apply tiny ML if we do not want to convert the data to categorical.