

## **EHB420E Final: Binary Classification of Coin Tosses Using Neural Network**

### **Data Collection:**

Since I lacked the knowledge on how to capture and process video and images, I was unable to record and analyze 100 coin tosses in that format. Instead, I opted to record the data in a structured form, noting the position where the coin fell and whether it landed on heads or tails. I considered the plane on which the coin fell as the xy plane, with the dropping point being treated as the origin. I expressed the position where the coin fell using x and y coordinates and saved the data accordingly. Additionally, I recorded whether it landed on heads or tails. I personally conducted some of the 100 tosses and sought assistance from online tools for the remainder. With the collected data, I created an Excel file named 'data' with three columns (x, y, and coin).

### **Data Preprocessing:**

Upon loading the dataset into a Pandas DataFrame, data preprocessing steps included inspecting and separating the features (X) and labels (y). The dataset was then split into training (80%) and test (20%) sets using the **train\_test\_split** function. This separation aimed to ensure the model's ability to generalize to new, unseen data.

### **Neural Network Architecture Selection:**

A simple feedforward neural network architecture was chosen for binary classification. The architecture consisted of an input layer with two nodes (for x and y coordinates) and an output layer with one node for binary classification. The **MLPClassifier** from Scikit-learn was used for its simplicity and effectiveness.

### **Model Training and Evaluation:**

The neural network was trained using the training set, and its performance was evaluated on the test set. Performance metrics such as accuracy, precision, and recall were calculated to assess the model's effectiveness. The code implementation ensured clarity and ease of understanding, with comments explaining each step.

### **Results:**

Due to the way I collected the data, my program does not fulfill all the desired requirements, but it is functional. My program predicts whether it is heads or tails based on where the coin landed. Therefore, it cannot predict where the coin will land.

Output:

```
Model Accuracy: 0.7
Enter x coordinat of coin's position(cm): 30
Enter y coordinat of coin's position(cm): 20
Prediction: Heads
```