My initial approach for predicting passenger survival involved simply converting the csv file, `train.csv’, into a NumPy array using the csv and NumPy libraries imported and then manipulating the data to create weights by hand. However, this method proved less effective, leading to difficulties in utilizing the data and NumPy arrays outright. So, I shifted my focus to learning and using the Pandas library, which allowed for easier data handling via converting the csv data into a Pandas DataFrame, it became easier to adjust the data and do the computations to make predictions. With the DataFrame in place, the next step was to create arbitrary weights for the attributes I found relevant. The features I kept were passenger class, sex, age, number of siblings/spouses aboard, number of parents/children aboard and ticket fare to compute a weighted sum and apply a threshold to predict survival. This easy method, while conceptually simple, yielded poor results, with accuracy only reaching around 30% with my best weights.

After the last class when we dove into the Rosenblatt Perceptron algorithm, I attempted to implement the perceptron algorithm, to find better weights for my predictions. I arbitrarily decided to apply it by looping through the dataset 50 times to iteratively adjust the weights rather than aim for a certain percentage of accuracy before ending the “training”. The implementation of the Rosenblatt Perceptron algorithm led to a significant improvement in prediction accuracy, to approximately 78%. This substantial improvement demonstrates the algorithm's effectiveness in capturing patterns in the data and refining weight adjustments. Further iterations of the Perceptron algorithm could potentially enhance accuracy even more, if the data is linearly separable.