1. **Overview**:

Alphabet Soup wants a tool that can help it select the applicants for funding with the best chance of success in their ventures. With machine learning, use the provided dataset to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup.

1. **Results**:

* Data Preprocessing
  + The target variable for the model is the WAS\_SUCCESSFUL column.
  + The feature variables for the model are “NAME, APPLICATION\_TYPE, AFFILIATION, CLASSIFICATION, USE\_CASE, ORGANIZATION, STATUS, INCOME\_AMT, SPECIAL\_CONSIDERATIONS, and ASK\_AMOUNT” columns.
  + The column “EIN” was removed from the data due to not being used as a target or a feature.
* Compiling, Training, and Evaluating the Model
  + For the model I used 75 neurons, 4 layers and added batch normalization to it, with ReLU as the activation function. I wanted the neurons and layers to be impactful, but not too heavy so there’s no chance of overfitting, and after different combination trials 75/4 seemed the most effective for the model. I used the batch normalization, as it helps make training deep neural networks more efficient and effective, by addressing issues related to activation distributions and gradient flow. After trials of different activators, I think the overall swiss army effect of the ReLU activator was the most useful, although the tanh activator showed good results as well.
  + The target achievement for the challenge was 75%, with the model created for the challenge consistently hitting 77%-80%, showing room for improvement but pushing into a good success range.
  + To increase model performance, I began the process by removing the “EIN” and scaling down the size of the other features, to streamline the data. After the selection of the features, I went about trimming the features, finding the balance where the data was meaningful and also scaled down. As explained above, I used batch normalization to increase efficiency, helping to enhance the ReLU activation. Finally, I cycled through the different neuron counts, adding and subtracting layers to the model, and switched through activation functions to find the right combination.

1. **Summary**:

**The model chosen had adequate results, consistently pushing the upper 70% threshold, but was by no means a for sure result when it came to applicant se**lection. The model could be enhanced with the addition of more data, possibly another feature that would provide more clarity for the model, currently there just isn’t enough information for the model to produce highly accurate results. Another approach to this problem would be to possibly pivot to Q Learning, keeping the current approach in mind, but reinforcing it with some more strategic thinking.