**Neural Network Model Report**

1. **Overview**

With the given information, Alphabet Soup wants to develop a tool that can help select the applicants for funding with the best chance of success in their ventures. They provide a dataset with about 34,000 organizations that have received funding from the organization over the years. They, therefore, require us to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup.

1. **Results**

**Data Pre-processing**

* The dataset provided has 12 columns.
* The target variable for the data was the “IS\_SUCCESSFUL” variable.
* The features for the model are “APPLICATION\_TYPE”, “AFFILIATION”, “CLASSIFICATION”, “USE\_CASE”, “ORGANIZATION”, “STATUS”, “INCOME\_AMT”, “SPECIAL\_CONSIDERATIONS”, “ASK\_AMT”, “IS\_SUCCESSFUL”
* The “NAME” and “EIN” columns were removed from the input data because they were neither targets nor features at first. In the optimized model, I chose to include the “NAME” column as part of the features to change the result and optimize the results.

**Compiling, Training, and Evaluating the Model**

* I selected to have 2 hidden layers and 1 output layer.
* The first hidden layer had 80 neurons while the second hidden layer had 30 neurons.
* In the first model, I was able to achieve an accuracy of 72.98% which was below the target performance.
* To increase the performance of the model I included “NAME” variable to be part of the features. Additionally, I binned the rare values in the category in a new value.
* I tried to increase a third hidden layer but this did not aid in improving the results.
* I kept altering the number of neurons in the hidden layers to get the optimal number of neurons that will improve the model performance and ended up settling with 80 for the first layer and 50 for the second hidden layer.
* I tried changing the activation for the output layer. I tried the “tahn” and the “leakey RELU” functions. However, the Sigmoid function still performed better and I, therefore, settled on it.

1. **Summary**

I managed to achieve an accuracy of 73.8% which was still below the target accuracy. The result was still really low and would need to be improved further.