**Overview of the Analysis:**

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

**Results**

* **Data Preprocessing**:

**What variable(s) are the target(s) for your model?**

* “IS\_SUCCESSFUL” variable as our target Variable

**What variable(s) are the features for your model?**

* + - APPLICATION\_TYPE
    - AFFILIATION
    - CLASSIFICATION
    - USE\_CASE
    - ORGANIZATION
    - STATUS
    - INCOME\_AMT
    - SPECIAL\_CONSIDERATIONS
    - ASK\_AMT

**What variable(s) should be removed from the input data because they are neither targets nor features?**

* + - “EIN” and “NAME”

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

**How many neurons, layers, and activation functions did you select for your neural network model, and why?**

* + - I decide to use 2 hidden layers with 80 and 30 neurons. I chose 80 and 30 is a reasonable initial choice. Because...

**Dat Complexity:** if having data with many features, having a larger first hidden layer (80 neurons) can help capture the wide range of information.

**Simplification:** The Secondlayer with fewer neurons (30 neurons) can serve to simplify the representation and reduce overfitting. It helps prevent the network from memorising the training data and encourages it to learn more general features.

**Were you able to achieve the target model performance?**

* + - No, Model achieving 73% accuracy which is close to the target result 75%

**What steps did you take in your attempts to increase model performance?**

* + - Add more neurons to hidden layer.
    - Add more hidden layers.
    - Use different activation functions for the hidden layers.
    - Add or reduce the number of epochs to the training.
    - Adjust the input data.
    - Dropping more or fewer columns.
    - Creating more bins for rare occurrences in columns.
    - Increasing or decreasing the number of values for each bin.

**Summary**:

After attempting many steps as discussed earlier, I could achieve the targeted results (75%). Dropping more or fewer columns resulting model keep crashing. I would suggest that to bring back column “NAME” into the model that can classify information and give out better accuracy.

I would recommend to use Keras Tuner library to further optimize model in order to achieve better accuracy score.