1. **Overview** of the analysis:
   * The non-profit foundation Alphabet Soup wants 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.
2. **Results**:

* Data Preprocessing
  + What variable(s) are the target(s) for your model?
    - In this case we will use “IS\_SUCCESSFUL” variable as our target it is the main indicator to see if the money will be use effectively after applying our model
  + What variable(s) are the features for your model? The rest of the data, excluding “EIN” and “NAME” will be used as our features:
    - 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” will be removed as it is an identification column.
* Compiling, Training, and Evaluating the Model
  + How many neurons, layers, and activation functions did you select for your neural network model, and why?
    - For the first model, I decide to use 2 hidden layers with 90 and 35 neurons. I chose 90 and 35 as a rule of thumb, number of neurons normally 2 or 3 times of the input features, which is 43 in this case.
  + Were you able to achieve the target model performance?
    - No, first model only getting around 73% accuracy which is close to the target result 75%
  + What steps did you take in your attempts to increase model performance?
    - I will attempt to:
    - Adjust the input data to ensure that no variables or outliers are causing confusion in the model, such as:
    - Dropping more or fewer columns.
    - Creating more bins for rare occurrences in columns.
    - Increasing or decreasing the number of values for each bin.
    - Add more neurons to a hidden layer.
    - Add more hidden layers.
    - Use different activation functions for the hidden layers.
    - Add or reduce the number of epochs to the training regimen.

1. **Summary**:
   1. After attempting various steps as mentioned above, I still could get/achived the targeted results (75%). The only steps that I cant do is “dropping more or fewer columns” as the model keep crashing on me when I compile the result. I reckon if we bring back column “NAME” into the model, it can classify information and give out better accuracy.

Text

Description automatically generated

* 1. I would recommend to using Keras Tuner library to try to further optimize your model and get better accuracy score.