Alphabet Soup.

1. **Overview** of the analysis: A nonprofit organization, Alphabet Soup, wants a tool that can help select applicants for funding with the best chance of success in their ventures. Using machine learning and neural networks and neural networks, to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup.
2. **Results**: Data Preprocessing
   * + EIN and NAME—Identification columns (Dropped as they are non-beneficial)
     + APPLICATION\_TYPE—Alphabet Soup application type (Categorized and rare values replaced with "Other")
     + AFFILIATION—Affiliated sector of industry (One-hot encoded and grouped for infrequent categories)
     + CLASSIFICATION—Government organization classification (One-hot encoded and grouped for infrequent categories)
     + USE\_CASE—Use case for funding (Target variable: 1 = Successful, 0 = Unsuccessful)
     + ORGANIZATION—Organization type (Target variable: 1 = Successful, 0 = Unsuccessful)
     + STATUS—Active status (Target variable: 1 = Successful, 0 = Unsuccessful)
     + INCOME\_AMT—Income classification (Target variable: 1 = Successful, 0 = Unsuccessful)
     + SPECIAL\_CONSIDERATIONS—Special considerations for application (Target variable: 1 = Successful, 0 = Unsuccessful)
     + ASK\_AMT—Funding amount requested (Target variable: 1 = Successful, 0 = Unsuccessful)
     + IS\_SUCCESSFUL—Was the money used effectively (Target variable: 1 = Successful, 0 = Unsuccessful)
   * Compiling, Training, and Evaluating the Model
     + **Compiling:** Loading the data into Google Collab and modifying the dataset by removing unnecessary column and grouping. Then splitting the dataset into training and testing sets.
     + **Training**: Using the Sequential Neural Network to input layer based on feature count.
     + **Evaluating**: Training the model to evaluate accuracy using the test data. Saving the trained model.
3. **Summary**: Optimizing the model using TensorFlow and achieving a target predictive accuracy higher than 75%. This was achieved by dropping a few columns, creating more bins for rare occurrences in columns, and manipulating the hidden layers. Splitting the preprocessed data into a features array, x, and a target array, y. Then using these arrays and the train\_test\_split function to split the data into training and testing datasets. Finally after training and testing the models, transforming the data set.