# Overview:

A wide variety of factors must be considered in order for Alphabet Soup to effectively select candidates for funding out of the many applications they review each year. Because applications can number in the thousands, a deep learning program has been designed to review applications and help select the candidates most likely to apply funding to successfully achieve their program goals. The program will examine data from past datasets to allow for the probable classification of applications into successful and unsuccessful categories.

# Results:

## Data Preprocessing

* + The dataset column “Is\_Successful” is used as the target value for the model, and was dropped from the dataframe to be used as the feature set.

Text

Description automatically generated

## Model variables are:

* + **EIN** and **NAME**—Identification columns
  + **APPLICATION\_TYPE**—Alphabet Soup application type
  + **AFFILIATION**—Affiliated sector of industry
  + **CLASSIFICATION**—Government organization classification
  + **USE\_CASE**—Use case for funding
  + **ORGANIZATION**—Organization type
  + **STATUS**—Active status
  + **INCOME\_AMT**—Income classification
  + **SPECIAL\_CONSIDERATIONS**—Special considerations for application
  + **ASK\_AMT**—Funding amount requested
* The Name and EIN columns were excluded from the dataset as they reference specific company identifiers and do not contain values useful to generating a deep learning model to predict future behavior. Since they are neither targets nor useful features, they were removed from the dataset.

Graphical user interface, text, application, email

Description automatically generated

## Compiling, Training, and Evaluating the Model

* + The best model scores were achieved with a model of 3 hidden layers of 80, 90, and 30 neurons, with using “relu” as the initial function and “sigmoid” for the remaining layers. Because the aim of the project was to classify the data into two categories, the “relu” regression function was the logical choice as a primary function.
  + Unfortunately, the models fell slightly short of the goal of 75% accuracy, but came very close to that number with an optimization of 74.6%
  + The initial neural network was set up with two hidden layers of 80 and 50 neurons, respectively. The next attempt at optimization added a third hidden layer of neurons, with 90 nodes in the first hidden layer, 60 in the second, and 30 in the third. This did not substantially improve the model’s performance, so the next optimization binned values of requested grant and donation amounts for easier evaluation of the model. In this model, 3 hidden layers were also retained in the model, with 80, 90, and 30 nodes, respectively, and the epochs for the model were increased from 100 to 500. This model did see some improvement in scores, but still provided only a 74.6% accuracy score. In the third optimization, an automated function was created to search for the best hyperparameters for model creation, but even with automized optimization, the highest accuracy score for a model was only 73.2%.

# Summary

Although the targeted model performance was not achieved in the optimizations attempted for this project, a useful model was obtained and could be used to assist in classification of future aid requests submitted to Alphabet Soup. However, further optimization could be done on the model itself. Using automation to find the best hyperparameters is perhaps the most efficient way to build an enhanced model. Initial optimization testing suggests that increasing the range of the auto-tuning parameters, or building more epochs into the target testing, may provide an increased model accuracy.