**Report on the Neural Network Model for Alphabet Soup Charity**

# Overview of the Analysis

The purpose of this analysis is to create a deep learning model using the Alphabet Soup Charity dataset, which contains data from various organizations that have received funding from Alphabet Soup over the years. The goal is to build a model that can predict which organizations are likely to be successful after receiving funding from Alphabet Soup, based on the features provided in the dataset.

# Results

## Data Pre-processing

The target variable for our model is **IS\_SUCCESSFUL**.

The features for our model are all the variables in the dataset except for **EIN** and **NAME**.

The variables **EIN** and **NAME** were removed from the input data because they are neither targets nor features.

## Compiling, Training, and Evaluating the Model

Our neural network model consists of three hidden layers with 7, 9, and 7 neurons respectively. We chose these values based on the hyperparameters generated by the Keras Tuner.

We used the ReLU activation function in the input layer and the Tanh activation function in the hidden layers. The output layer uses the sigmoid activation function.

We were able to achieve an accuracy score of 73.4% on the test set, which met our target performance.

To increase model performance, we tried different hyperparameters, such as changing the number of layers, the number of neurons in each layer, and the activation functions. We also used regularization techniques such as dropout to prevent overfitting.

## Summary

Our deep learning model achieved an accuracy score of 73.4% on the test set, which met our target performance. We were able to achieve this score by using a neural network with three hidden layers, consisting of 7, 9, and 7 neurons respectively, and the ReLU and Tanh activation functions.

## Recommendation

A different model that could be used to solve this classification problem is a Random Forest Classifier. Random Forests are a type of decision tree algorithm that work by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random Forests have been shown to work well on a wide range of classification problems and can handle large datasets with high-dimensional feature spaces. However, Random Forests are not as interpretable as neural networks, so there is a trade-off between interpretability and accuracy. If interpretability is not a concern, a Random Forest Classifier could be a good choice for this problem.