# Alphabet Soup Charity Funding Predictor: Neural Network Model Report

## Overview of the Analysis

The primary purpose of this analysis is to build a machine learning model to help Alphabet Soup, a nonprofit foundation, predict which funding applicants are most likely to succeed. By identifying successful applicants, Alphabet Soup can allocate its resources more effectively, supporting ventures that are likely to have a significant impact. The analysis employs a deep learning approach using a neural network model to make these predictions.

## Results

## Data Preprocessing

### Target Variable

The target variable for the model is IS\_SUCCESSFUL, which indicates whether the funding recipient used the funds effectively.

### Feature Variables

* EIN and NAME:
* APPLICATION\_TYPE: Type of application submitted.
* AFFILIATION: Affiliated sector of the industry.
* CLASSIFICATION: Government organization classification.
* USE\_CASE: Purpose for the funding request.
* ORGANIZATION: Type of organization.
* STATUS: Active status.
* INCOME\_AMT: Income classification.
* SPECIAL\_CONSIDERATIONS: Whether there are any special considerations.
* ASK\_AMT: Amount of funding requested.

### Variables Removed

The variables EIN and NAME were removed from the input data as they are identification numbers and do not contribute meaningfully to predicting the success of the funding.

## Compiling, Training, and Evaluating the Model

### Neural Network Configuration

**Neurons**

The model was initially configured with two hidden layers. The first hidden layer consisted of 80 neurons, and the second hidden layer had 30 neurons. The choice of neurons was based on trial and error, aiming to balance model complexity and computational efficiency.

**Layers**

The model used an input layer (corresponding to the number of input features), two hidden layers, and an output layer with a single neuron for binary classification.

**Activation Functions**

ReLU (Rectified Linear Unit) activation function was used for the hidden layers to introduce non-linearity, which helps the model learn complex patterns. A sigmoid activation function was used in the output layer to produce a probability output for binary classification.

### Model Performance

The initial model achieved an accuracy of around **72%**, which was below the target accuracy of **75%**.

## Steps to Increase Model Performance

### A. Adjusting Model Architecture

The number of neurons and layers was varied to find the optimal configuration. Experimenting with more layers and varying neuron counts was aimed at capturing more complex relationships in the data.

### Changing Activation Functions

Alternative activation functions, such as tanh and LeakyReLU, were tested in the hidden layers to see if they could improve learning capability.

### Increasing Epochs

The number of training epochs was increased to give the model more opportunities to learn patterns from the data.

### Implementing Dropout

Dropout layers were introduced to prevent overfitting by randomly setting a fraction of input units to 0 during training.

### Feature Engineering

Additional feature engineering techniques were explored, such as creating new features or combining existing ones to provide more meaningful input to the model.

## Summary

The deep learning model developed for Alphabet Soup was moderately successful, achieving a maximum accuracy of **72%** after optimization efforts. Although this performance fell slightly short of the desired threshold, the model still offers a useful tool for predicting applicant success.

## Recommendations for Alternative Models

While the neural network provided valuable insights, exploring alternative models might further improve prediction accuracy:

### Random Forest Classifier

**Reason**

Random Forests are robust to overfitting, especially with noisy data, and can handle a large number of input features effectively. They also provide feature importance metrics, which can be useful for identifying which features most strongly predict success.

### Gradient Boosting Machines (GBM)

**Reason**

GBM models, such as XGBoost or LightGBM, often outperform neural networks on structured/tabular data due to their ability to model complex relationships and interactions between features efficiently. They are also highly tunable, with various parameters that can be adjusted to optimize performance.

### Support Vector Machine (SVM)

**Reason**

SVMs are effective in high-dimensional spaces and with a clear margin of separation. They work well in binary classification tasks and can be a good alternative when neural networks do not provide sufficient accuracy.