**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

## The neural network model we created has 1 input layer, 3 hidden layers with varying number of neurons, and 1 output layer. The activation functions used were "tanh" and "sigmoid". We used Kerastuner to automatically select the best hyperparameters for our model.

## We were able to achieve an accuracy of 0.7283 on the test data, which is close to the target model performance of 0.75.

## To increase model performance, we tried several methods including adding more hidden layers, increasing the number of neurons in each layer, changing the activation functions, and tuning the learning rate of the optimizer.

## Summary

## In summary, our deep learning model performed reasonably well in predicting whether an organization will be successful or not. However, we recommend trying a different model, such as a random forest classifier or a support vector machine, to see if it can achieve better performance on this classification problem. These models are known to work well with binary classification problems and may be more suitable for the data provided.

## 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.