Alphabet Soup Deep Learning Model Analysis

# Overview of the Analysis

The purpose of this analysis was to build and evaluate a deep learning model capable of predicting whether an organization will be successful in receiving funding from Alphabet Soup. The model was trained on a dataset containing historical application data, including categorical and numerical features related to each organization's mission, income, and operational characteristics.

# Results

## Data Preprocessing

• Target Variable: IS\_SUCCESSFUL (1 = successful, 0 = not successful)  
• Features: APPLICATION\_TYPE, AFFILIATION, CLASSIFICATION, INCOME\_AMT, ASK\_AMT, USE\_CASE, ORGANIZATION, STATUS, SPECIAL\_CONSIDERATIONS, among others after encoding  
• Removed Variables: EIN, NAME — these are identifiers and do not contribute to the prediction

## Compiling, Training, and Evaluating the Model

• Neural Network Architecture:  
 - Input Layer: Based on the number of features after encoding (e.g., 116)  
 - Hidden Layers: 2 layers with 80 and 30 neurons respectively  
 - Activation Functions: ReLU for hidden layers, Sigmoid for the output layer  
  
• Model Performance:  
 - Accuracy Target: 75%  
 - Final Accuracy Achieved: ~72.5% (based on validation data)  
  
• Model Optimization Attempts:  
 - Increased/decreased number of neurons  
 - Tried adding/removing layers  
 - Tuned epochs and batch size  
 - Used dropout to prevent overfitting  
 - Normalized numerical input features

# Summary

The deep learning model was moderately successful in predicting the likelihood of funding approval, achieving about 72.5% accuracy on the validation set. However, it fell short of the target threshold of 75%. The model could potentially be improved through further hyperparameter tuning or feature engineering.  
  
As an alternative to a neural network, I recommend trying a Random Forest Classifier. This model is capable of handling mixed feature types, managing non-linearities, and providing feature importance metrics for interpretability. Random forests also tend to perform well on classification problems with structured tabular data like this.