Alphabet Soup Charity Neural Network Model Report

# Overview of the Analysis

The objective of this project was to develop a binary classification model that predicts whether a funding application to Alphabet Soup will be successful. Using a dataset of over 34,000 historical applications, we leveraged TensorFlow and Keras to create and optimize a deep learning model capable of identifying patterns in applicant features that correlate with success.

# Results

## Data Preprocessing

• Target Variable: IS\_SUCCESSFUL — a binary indicator of success (1) or failure (0).

• Feature Variables: APPLICATION\_TYPE, AFFILIATION, CLASSIFICATION, USE\_CASE, ORGANIZATION, STATUS, INCOME\_AMT, SPECIAL\_CONSIDERATIONS, ASK\_AMT.

• Removed Variables: EIN and NAME — identification columns not useful for model prediction.

• Encoding and Transformation: Rare categorical values were grouped into “Other” categories, categorical variables were one-hot encoded, data was split into training and testing sets, and numerical features were scaled using StandardScaler().

## Compiling, Training, and Evaluating the Model

• Input Features: 116 after one-hot encoding.

• Hidden Layers: 2 hidden layers with 80 and 30 neurons respectively.

• Activation Functions: ReLU for hidden layers, Sigmoid for output.

• Output Layer: 1 neuron for binary classification.

• Loss Function: Binary crossentropy.

• Optimizer: Adam.

• Callbacks: ModelCheckpoint to save weights every 5 epochs.

• Performance: Accuracy after optimization: ~75.1%; Loss: ~0.49.

## Model Optimization Attempts

• Architecture Tuning: Increased neurons, added a third layer, tested alternative activation functions.

• Training Adjustments: 100 epochs, batch size 32, early stopping and checkpointing.

• Data Adjustments: Binned low-frequency categories, tested dropping outlier ASK\_AMT values.

# Summary and Recommendation

The optimized neural network model achieved a predictive accuracy of approximately 75.1%, meeting the target. For future improvements, consider:  
- Trying alternative models like Random Forest or XGBoost.  
- Using hyperparameter tuning tools like Keras Tuner or Optuna.  
- Engineering new features or interaction terms.