# Report on the Neural Network Model for Alphabet Soup Charity

## Overview of the Analysis

The purpose of this analysis was to develop and evaluate a deep learning model for predicting the success of funding applications at Alphabet Soup Charity. By leveraging a neural network, the goal was to accurately classify whether a given applicant will be successful based on various features derived from historical data.

---

## Results

### Data Preprocessing

- \*\*Target Variable:\*\*

- The target variable for the model is `IS\_SUCCESSFUL`, which indicates whether a funding application was successful.

- \*\*Feature Variables:\*\*

- The feature variables selected for the model include applicant data such as `APPLICATION\_TYPE`, `AFFILIATION`, `CLASSIFICATION`, `USE\_CASE`, `ORGANIZATION`, `INCOME\_AMT`, and other categorical and numerical data.

- \*\*Removed Variables:\*\*

- Variables that were removed include irrelevant or redundant data columns such as `EIN` and `NAME`, ‘SPECIAL\_CONSIDERATIONS’. These identifiers did not contribute meaningfully to the predictive power of the model.

---

### Compiling, Training, and Evaluating the Model

- \*\*Neurons, Layers, and Activation Functions:\*\*

- \*\*Attempt 1 (Baseline Model):\*\*

- Architecture: 2 hidden layers with 80 and 30 neurons respectively.

- Activation function: ReLU for hidden layers and sigmoid for the output layer.

- Performance: Loss = 0.5579, Accuracy = 73.05%.

- \*\*Attempt 2 (Enhanced Model):\*\*

- Architecture: Increased neurons to 128 and 64 in two hidden layers.

- Regularization techniques such as dropout were applied.

- Performance: Loss = 0.5730, Accuracy = 72.80%.

- \*\*Attempt 3 (Optimized Model):\*\*

- Architecture: Added kernel regularization (L2) and dropout to reduce overfitting.

- Performance: Loss = 0.5717, Accuracy = 72.85%.

- \*\*Final Initialized Model:\*\*

- Architecture: 3 hidden layers with 256, 128, and dropout layers.

- Performance: Loss = 0.5802, Accuracy = 72.38%.

- \*\*Target Model Performance:\*\*

- The target model accuracy of 75% was not achieved despite several optimization attempts.

- \*\*Steps to Improve Model Performance:\*\*

- Increased the number of neurons in hidden layers.

- Added additional hidden layers.

- Applied dropout and regularization techniques.

- Adjusted the number of epochs during training.

---

## Summary

The overall results of the deep learning model indicate that while the baseline accuracy was decent at 73.05%, subsequent attempts at optimization did not yield significant improvements. The final initialized model achieved a slightly lower accuracy of 72.38%.

### Recommendations for Alternative Models

To further improve the predictive accuracy of the classification problem, the following models are recommended:

1. \*\*Random Forest Classifier:\*\*

- Can handle imbalanced data effectively and provides feature importance.

- Suitable for complex datasets with categorical features.

2. \*\*XGBoost:\*\*

- A powerful gradient boosting technique that often outperforms neural networks in tabular data classification tasks.

3. \*\*Support Vector Machines (SVMs):\*\*

- Effective for high-dimensional spaces.

- Particularly useful when clear decision boundaries exist.

These models may better capture relationships in the data and achieve the target performance metric.