

# REPORT ON THE NEURAL NETWORK MODEL

---

## Overview of the Analysis

The purpose of this analysis was to build a deep learning model capable of predicting the success of funding applications for a nonprofit foundation Alphabet Soup. This foundation wants a tool that can help it select the applicants for funding with the best chance of success in their ventures. This model will help create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup and determine whether an organization will be successful based on various features extracted from the provided dataset.

---

## Results

### Data Preprocessing

1. First the dataset was read from here [https://static.bc-edx.com/data/dl-1-2/m21/lms/starter/charity\\_data.csv](https://static.bc-edx.com/data/dl-1-2/m21/lms/starter/charity_data.csv) and it got imported into application\_df dataframe.
2. This csv file contains over 34,000 organizations that have received funding from Alphabet Soup organization, using metadata columns to predict the success of future applicants.
3. **Target Variable:** IS\_SUCCESSFUL (binary classification - 1 for success, 0 for failure).
4. **Feature Variables:** This dataset contains several columns that capture metadata about each organization, such as:
  - a) EIN and NAME: Identification columns
  - b) APPLICATION\_TYPE: Alphabet Soup application type
  - c) AFFILIATION: Affiliated sector of industry
  - d) CLASSIFICATION: Government organization classification
  - e) USE\_CASE: Use case for funding
  - f) ORGANIZATION: Organization type
  - g) STATUS: Active status
  - h) INCOME\_AMT: Income classification
  - i) SPECIAL\_CONSIDERATIONS: Special considerations for application
  - j) ASK\_AMT: Funding amount requested
  - k) IS\_SUCCESSFUL: Was the money used effectively

The feature variables consist of every alternate column in the application\_df dataframe, determined by excluding the 'IS\_SUCCESSFUL' column from the original dataframe.

5. **Removed Variables:** EIN and NAME were dropped as they are identifiers are not targets nor features for the dataset. Therefore, they do not contribute meaningfully to the model.

## Compiling, Training, and Evaluating the Model

### 1. Model Architecture:

- **Layers:**
  - There were 2 hidden layers.
  - The first hidden layer had 15 neurons with tanh activation.
  - The second hidden layer had 10 neurons with relu activation.
  - The output layer had 1 neuron with relu activation for binary classification.
- **Why these choices?**
  - tanh and relu are effective activation functions for extracting non-linear relationships in hidden layers.
  - relu was selected for binary classification in the output layer.

### 2. Performance:

- The baseline model achieved an accuracy of approximately 73%.
- Despite optimizations, including adding more neurons, changing activation functions, and increasing epochs, and trying new activations like sigmoid the target accuracy of 75% could not be achieved.

### 3. Optimization Steps:

- Added a third hidden layer and dropout layers to prevent overfitting.
- Increased the number of neurons in the hidden layers.
- Adjusted the number of epochs.
- Experimented with different activation functions, including relu, tanh, and sigmoid in different layers.

---

## Summary

The deep learning model performed well but did not achieve the 75% target accuracy. While optimizations improved the model slightly, further enhancements could include:

1. Using a different machine learning model which may better handle categorical features and imbalanced datasets.
2. Applying feature engineering to create more meaningful variables and activations from the data.

In conclusion, while the neural network demonstrated promise, a tree-based model might offer higher accuracy for this classification problem due to its ability to capture complex feature interactions. We can evaluate the model using the test data to determine the loss and accuracy for to create different model could solve this classification problem.