**Report on Neural Network Model**

**Purpose of the Analysis**

A nonprofit foundation is seeking a tool that can help it select applicants with the highest chance of success in their ventures. Use neural networks to create a binary classifier that can predict a applicant’s success based on the provided dataset.

**Dataset**

Total organizations equated to 34,298 that received funding from the nonprofit foundation. Metadata is captured for each organization. The inputs are listed below.

* **EIN** and **NAME**—Identification columns
* **APPLICATION\_TYPE**—Alphabet Soup application type
* **AFFILIATION**—Affiliated sector of industry
* **CLASSIFICATION**—Government organization classification
* **USE\_CASE**—Use case for funding
* **ORGANIZATION**—Organization type
* **STATUS**—Active status
* **INCOME\_AMT**—Income classification
* **SPECIAL\_CONSIDERATIONS**—Special considerations for application
* **ASK\_AMT**—Funding amount requested
* **IS\_SUCCESSFUL**—Was the money used effectively

**Data Preprocessing**

1. Target variable: “IS\_SUCCESSFUL” column
2. Features used in training the model:
   * **NAME**
   * **APPLICATION\_TYPE**
   * **AFFILIATION**
   * **CLASSIFICATION**
   * **USE\_CASE**
   * **ORGANIZATION**
   * **INCOME\_AMT**
   * **ASK\_AMT**
3. Features removed because they are not targets or features:
   * **EIN –** anon-predictive id number
   * **NAME –**a non-predictive organization label
4. Binning: Reduce the number of **APPLICATION\_TYPES** labels from 17 to 9 by creating an ‘Other’ value label to consolidate all variables that have counts on an individual basis under 500.
5. Binning: Reduce the number of **CLASSIFICATION** labels from 71 to 46 by creating an ‘Other’ value label to consolidate all variables that have counts on an individual basis under 2.
6. Convert categorical data to numeric

**Compiling, Training, and Evaluating Model Optimization Results**

The final optimization, Sequential model had the following parameters

* Three layers
* Number of neurons:
  + - Layer 1: 75
    - Layer 2: 50
    - Layer 3: 25
  + Activation function types:
    - Layer 1: relu
    - Layer 2: relu
    - Layer 3: sigmoid

I took the following steps to increase model accuracy, but with no increased model performance:

1. Increased the number of hidden layers from 2 to 3.
2. Changed the ratio of neurons in layers 1, 2 and 3.
3. Changed activation functions from relu to sigmoid.
4. Reduce epochs.

**The final model:**

A picture containing text

Description automatically generated

I was not able to achieve target model performance greater than 75% from the initial model.

**Text

Description automatically generated**

**Different Model to Improve Results and Why Model Choice**

I recommend trying a LogisticRegression or a Random Forest Classifier under the premise that the data is linear. Therefore, a complicated neural network model does not add any increased accuracy. My adjustment of layers, neurons and activation functions did not increase model accuracy.

On the data front, I would recommend binning the ASK\_AMOUNT to 10 ranges to reduce the 8,747 unique data values to 10. This would help tune the model by creating more balance in the input data set. Remove SPECIAL\_CONSIDERATIONS because almost all of the values are N while Y is only 0.08% of total values.