

Overview: The purpose of the analysis is to help select applicants for the Alphabet Soup Charity.

Results:

Data Preprocessing:

- Target variable is the “IS_SUCCESSFUL” column.
- Feature Variables Include all numeric values including application types, income, and special considerations.

Compiling, Training, and Evaluating the Model:

- My final attempt I chose to start at 40 neurons on the first layer, 20 neurons on the second layer, and 10 neurons on the last layer. I felt that starting close to the number of features and consolidating the number from there was an effective method. However, I was only able to reach an accuracy of 73%.

```
Attempt 3:

layer1 = 40 : activation function = relu
layer2 = 20 : activation function = relu
layer3 = 10 : activation function = relu
Final Accuracy: 73%
```

```
# Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.
number_input_features = len(X_train_scaled[0])
hidden_nodes_layer1 = 40
hidden_nodes_layer2 = 20
hidden_nodes_layer3 = 10

nn = tf.keras.models.Sequential()

# First hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer1, input_dim=number_input_features, activation="relu"))

# Second hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer2, activation="relu"))

# Third hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer3, activation="relu"))

# Output layer
nn.add(tf.keras.layers.Dense(units=1, activation="sigmoid"))

# Check the structure of the model
nn.summary()
```

Model: "sequential_2"

Layer (type)	Output Shape	Param #
dense_7 (Dense)	(None, 40)	1760
dense_8 (Dense)	(None, 20)	820
dense_9 (Dense)	(None, 10)	210
dense_10 (Dense)	(None, 1)	11

Total params: 2801 (10.94 KB)
Trainable params: 2801 (10.94 KB)
Non-trainable params: 0 (0.00 Byte)

Summary: Giving more resources or less resources isn't going to fix poor preprocessing. In the future I will spend more time going over the data to clean and prep it.