

# Overview

The purpose of this analysis was to find the best applicant for funding based on the nonprofit foundation Alphabet Soup.

## Results

Data Preprocessing:

- The target variable came from the column 'IS\_SUCCESSFUL' where 1 is considered to be successful and 0 is considered to be unsuccessful.
- The features would be the remaining columns within the dataset.
- The variables that should be removed because they are neither targets nor features are 'Name' and 'EIN'.

Compiling, Training, and Evaluating the Model:

- For this evaluation 5,901 parameters were used with 3 layers of training models, the activation functions used were relu and sigmoid.
- Relu and sigmoid activations were used because they are easy to train and provide great performance. Sigmoid activation is a key factor for neural networks to learn complex problems.

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
dense_3 (Dense)	(None, 80)	3440
dense_4 (Dense)	(None, 30)	2430
dense_5 (Dense)	(None, 1)	31
Total params: 5,901		
Trainable params: 5,901		
Non-trainable params: 0		

268/268 - 0s - loss: 0.5580 - accuracy: 0.7271 - 429ms/epoch - 2ms/step  
Loss: 0.5580423474311829, Accuracy: 0.7271137237548828

## What steps did you take in your attempts to increase model performance?

In a second attempt to increase model performance the keras tuner was used to find the best hyperparameter. The model was tested against relu, tanh and sigmoid functions with neurons ranging from 1 to 10 and hidden layers from 1 to 6 with a single output layer. The results concluded that tanh was the best option for activation function along with 7 neurons and 3 layers.

```
# Get best model hyperparameters
best_hyper = tuner.get_best_hyperparameters(1)[0]
best_hyper.values
```

```
{'activation': 'tanh',
 'first_units': 7,
 'num_layers': 3,
 'units_0': 5,
 'units_1': 1,
 'units_2': 5,
 'units_3': 1,
 'units_4': 5,
 'units_5': 1,
 'tuner/epochs': 20,
 'tuner/initial_epoch': 7,
 'tuner/bracket': 1,
 'tuner/round': 1,
 'tuner/trial_id': '0048'}
```

---

Despite finding the most accurate parameters for the model, the accuracy score remains the same in comparison to the initial test.

```
# Evaluate best model against full test data
best_model = tuner.get_best_models(1)[0]
model_loss, model_accuracy = best_model.evaluate(X_test_scaled, y_test, verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
268/268 - 0s - loss: 0.5615 - accuracy: 0.7272 - 398ms/epoch - 1ms/step
Loss: 0.5615042448043823, Accuracy: 0.7272303104400635
```

---

## **Summary:**

Based on the initial testing, the neural model provided a fair prediction for charity success that would allow Alphabet Soup better insight on selecting appropriate applicants. As a recommendation, using PCA for future models could reduce the amount of features along with the amount of noise for better accuracy.