# Deep Learning Challenge - Alphabet Soup Foundation

## Overview

The non-profit foundation Alphabet Soup wants a tool that can help it select the applicants for funding with the best chance of success in their ventures. With your knowledge of machine learning and neural networks, you'll use the features in the provided dataset to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup.

## Results

### Data Preprocessing

- What variable(s) are the target(s) for your model?
  - **IS_SUCCESSFUL**

- What variable(s) are the features for your model?
  - **APPLICATION_TYPE**
  - **AFFILIATION**
  - **CLASSIFICATION**
  - **USE_CASE**
  - **ORGANIZATION**
  - **STATUS**
  - **INCOME_AMT**
  - **SPECIAL_CONSIDERATIONS**
  - **ASK_AMT**

- What variable(s) should be removed from the input data because they are neither targets nor features?
  - **EIN**
  - **Name**

**Compiling, Training, and Evaluating the Model**

**In this challenge, only 2 approaches were used to reach the 75% accuracy goal.**

**In the first model I  built:**

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 10)                480

 dense_1 (Dense)             (None, 10)                110

 dense_2 (Dense)             (None, 1)                 11

=================================================================
Total params: 601
Trainable params: 601
Non-trainable params: 0
_________________________________________________________________
```

- 10 neurons, 2 hidden layers and relu activation were selected as this is the first attempt to the first model.

```
# Train the model
fit_model = nn_model.fit(X_train_scaled, y_train, epochs=50)
```

- 50 epochs were selected as to observe how the model is performing.

```
268/268 - 1s - loss: 0.5586 - accuracy: 0.7230 - 536ms/epoch - 2ms/step
Loss: 0.5585763454437256, Accuracy: 0.7230320572853088
```

- The target model performance (75%) was not achieved, the model only has 72.30% of accuracy.

## Optimization

```
# Look at NAMES value counts for binning
N_counts = application_df['NAME'].value_counts()
N_counts
```

```
PARENT BOOSTER USA INC                                          1260
TOPS CLUB INC                                                    765
UNITED STATES BOWLING CONGRESS INC                               700
WASHINGTON STATE UNIVERSITY                                      492
AMATEUR ATHLETIC UNION OF THE UNITED STATES INC                  408
                                                                ...
ST LOUIS SLAM WOMENS FOOTBALL                                      1
AIESEC ALUMNI IBEROAMERICA CORP                                    1
WEALLBLEEDRED ORG INC                                              1
AMERICAN SOCIETY FOR STANDARDS IN MEDIUMSHIP & PSYCHICAL INVESTIGATI    1
WATERHOUSE CHARITABLE TR                                           1
Name: NAME, Length: 19568, dtype: int64
```

## In this model, the name column was kept.

```
Other                                          21022
PARENT BOOSTER USA INC                          1260
TOPS CLUB INC                                    765
UNITED STATES BOWLING CONGRESS INC               700
WASHINGTON STATE UNIVERSITY                      492
                                                ...
CASCADE 4-H FOUNDATION                            10
FREE & ACCEPTED MASONS OF WASHINGTON              10
NEW MEXICO GARDEN CLUBS INC                       10
NATIONAL ASSOCIATION OF HISPANIC NURSES           10
UNION OF CALIFORNIA STATE WORKERS                 10
Name: NAME, Length: 223, dtype: int64
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 10)                2770

 dense_1 (Dense)             (None, 10)                110

 dense_2 (Dense)             (None, 10)                110

 dense_3 (Dense)             (None, 10)                110

 dense_4 (Dense)             (None, 1)                 11

=================================================================
Total params: 3,111
Trainable params: 3,111
Non-trainable params: 0
_________________________________________________________________
```

- 10 neurons, 4 hidden layers and relu activation were selected as this is the first attempt to the first model.

```
# Train the model
fit_model = nn_model.fit(X_train_scaled, y_train, validation_split=0.15, epochs=100)
```

- 100 epochs were selected to increase the accuracy.

```
268/268 - 0s - loss: 0.4719 - accuracy: 0.7766 - 363ms/epoch - 1ms/step
Loss: 0.4718588590621948, Accuracy: 0.7765597701072693
```

- The target model performance (75%) was achieved.

**Summary**

The Keeping the Name column was the important procedure which shows the importance of the dataset shape before you preprocessing.

In the optimization, I tried to keep increase the hidden layers from 2 to 4, increase the epochs and added a neurons split to observe if the Name column is essential in building the model or it will reach oversampling stage.