# Charity Funding Predictor

## Overview Of The Analysis

**The purpose of the analysis** is to predict whether or not applicants for funding will be successful.

**Funding organization**: The non-profit foundation Alphabet Soup.

**Data**: CSV file containing more than 34,000 organizations that have received funding from Alphabet Soup over the years.

**Algorithms used**: Machine Learning, Neural Network, Deep Neural Networks.

**Tools**: Pandas, Scikit-Learn, TensorFlow.

## Results

- **Data Preprocessing**

**Target column**: *IS_SUCCESSFUL*.

**Features**: *APPLICATION_TYPE*—Alphabet Soup application type; *AFFILIATION*—Affiliated sector of industry; *CLASSIFICATION*—Government organization classification; *USE_CASE*—Use case for funding; *ORGANIZATION*—Organization type; *INCOME_AMT*—Income classification; *ASK_AMT*—Funding amount requested.

**Removed variables**: *EIN*, *NAME* - Identification columns;

*STATUS* and *SPECIAL_CONSIDERATIONS* removed at optimization step.

**Modified variables**: *APPLICATION_TYPE, CLASSIFICATION* were binned to group rare values.

*ASK_AMT was binned to group similar values at the optimization step.*

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

**Neural network model**:

**Initial NN** was built using one input layer, one hidden layer and one output layer with number of neurons - 80 (input layer), 30 (hidden layer), and activation functions - ‘relu’ for hidden layers and ‘sigmoid’ for output layer.

Initial model scored 0.7265 on accuracy.

```
# Evaluate the model using the test data
model_loss, model_accuracy = nn.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
8575/8575 - 0s - loss: 0.5578 - acc: 0.7263
Loss: 0.557812534073699, Accuracy: 0.7262973785400391
```

**Optimized NN** was tuned with Keras to get best hyperparameters for number of neurons, number of hidden layers, type of activation functions.

Optimized model scored 0.7977 on accuracy.

```
# Evaluate the top 3 models against the test dataset
top_model = tuner.get_best_models(3)
for model in top_model:
    model_loss, model_accuracy = model.evaluate(X_test_scaled,y_test,verbose=2)
    print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
268/268 - 0s - loss: 0.5417 - accuracy: 0.7977
Loss: 0.5417230129241943, Accuracy: 0.7976676225662231
268/268 - 1s - loss: 0.5942 - accuracy: 0.7958
Loss: 0.5941565632820129, Accuracy: 0.7958017587661743
268/268 - 1s - loss: 0.4773 - accuracy: 0.7956
Loss: 0.47731533646583557, Accuracy: 0.7955685257911682
```

To optimize NN following steps had been done:

- **Input data adjusted:**
- more columns dropped (*STATUS* and *SPECIAL_CONSIDERATIONS*);
- bins modified *APPLICATION_TYPE* - number of bins was reduced*; CLASSIFICATION* - the number of values for each bin was decreased; *INCOME_AMT* - additional bin was created; *ASK_AMT* was binned to group similar values - the result of grouping was saved into ‘bins’ column.
- **Model parameters adjusted**:
- Kerastuner library was used to train Sequential model with hyperparameter options.
- Number of neurons, hidden layers, types of activation functions were adjusted.
- Model with the best hyperparameters was selected.

## Summary

Keras Sequential Model was used to build a binary classifier to be able to predict if the funding applicant might be successful.

The optimized model was able to achieve the target model performance of greater than 0.75 - Optimized Model scored 0.7977.

To receive this score, input data were adjusted and model tuned with hyperparameters.