

## **Overview**

The purpose of this analysis is to determine if by using the provided data, a neural network model that is 75% accurate or more can be produced. This model is to be used to determine if an applicant's campaign will be successful if they are funded by Alphabet Soup. Attempts at this accuracy level were conducted 4 times. Each attempt changed at least one parameter in an effort to raise the accuracy score.

## **Results**

- Data Processing
  - Target(s)
    - In this model, the binary flag "IS\_SUCCESSFUL" is our target.
      - Values from this column will be used in "y\_test" and "y\_train"
  - Features
    - Categorical variables such as "APPLICATION\_TYPE", "CLASSIFICATION", and "USE\_CASE" are features to be used by our model.
      - Since these are categorical and not numerical, these fields will have to be encoded
    - Numerical data such "ASK\_AMT" is also a feature
    - The "STATUS" column is a binary column that is also a feature
  - Data to remove
    - The columns "EIN" and "Name" are used to identify the organization and are neither features nor targets. They should be dropped
- Compiling, Training, and Evaluating the Model
  - For my initial model, I selected:
    - 2 hidden layers
    - 43 inputs to the first layer with 80 outputs
    - ReLU
  - I was not able to achieve the target with these parameters, though i came close with ~73% accuracy
  - In order to try to increase the accuracy, I"
    - Increased the number of hidden layers to 5
      - Adding additional hidden layers did not impact the accuracy
    - Dropping the "AFFILIATION" and "ORGANIZATION" features with 2 layers
      - This dropped the accuracy by almost 10 percentage points
    - Binning more categories from the "APPLICATION\_TYPE" into the "Other" category and using 5 hidden layers
      - This brought the accuracy back near the 73% starting benchmark

## **Summary**

Overall, I was able to come close to achieving the target accuracy, but was not able to reach it. If I were to continue, I would move forward with a KNN model. I would do this to cut through the potential noise features.