**Alphabet Soup Funding Success Machine Model**

**Overview**

The purpose of this analysis is to create a machine learning model that can accurately predict whether or not a venture was successful after receiving funding from Alphabet Soup. I analyzed 34,000 organizations that have received funding in the past from the company to create this model.

**Results**

**Data Processing**

* The target variable in this data set is the “IS\_SUCCESSFUL”. It is one of the only columns with binary values.
* The identification data from the “EIN” and “NAME” columns for the applicants were removed as they are not necessary for the machine learning model.
* The features of the data set include:
  + APPLICATION\_TYPE - the Alphabet Soup application type
  + AFFILIATION- Industry of the applicant
  + CLASSIFICATION- Government organization classification
  + USE\_CASE- how the funding will be used
  + ORGANIZATION- Type of organization
  + STATUS- active or inactive status
  + INCOME\_AMT- income classification
  + SPECIAL\_CONDISERATIONS- whether special considerations were taken for the application
  + ASK\_AMT- Funding requested

**Compiling, Training, & Evaluating**

* The model uses the 43 input features with 2 hidden layers. The first hidden layer has 80 nodes and the second layer has 30 nodes. Bothe of these layer use the ReLU activation function because we do not have any negative values to analyze. These are all positive, mostly integer values. I used the sigmoid activation function in the final layer because we are trying to predict an outcome, which can never be negative, just not existent. Sigmoid function exists between 0 and 1 and there for cannot be negative.
* I was not able to reach the target model performance. I achieved about 72.5% accuracy in my most
* I removed the “SPECIAL\_CONSIDERATIONS” column because there were only 27 applications out of 34,000 that needed special considerations. This seemed like extra data that did not add any value to the model. I also grouped all the applicants whose income was more than $1 together. There was a total of 24,388 applicants that did not report any income, while 9,911 have between $1 and 50+ million. It may be beneficial to split up the data into different data frames and analyze them separately. I also tried to create fewer bins for the classification type, as the top classification C100 had more than half the applicants. Lastly, I added a third final layer with 50 nodes and a sigmoid activation function. By making these changes, my accuracy fell slightly.

**Summary**

This model has room for improvement. I believe better subdivision of the data would allow for better outcomes. Creating a model that can predict only for organizations with no income and another model for organizations with income may help with the accuracy of the models. It would also be beneficial to divide the data by one set all being classified as a C1000 application and another set for the rest of the application classifications. C1000 had more than half of the applications, which could mean that the model is being overtrained by data in that classification type.