We were assigned to determine if we could utilize machine learning and neural networks to predict whether applicants for a non-profit grant would be successful after being funded.

**Results:**

Target for all attempts was 'IS\_SUCCESSFUL'

Features were the remaining columns after dropping unnecessary columns as stated in the bullets below.

**Starter\_Code:** https://github.com/ClaytonsCreations/Non-Profit\_application\_deepLearning/blob/master/Starter\_Code.ipynb

Graphical user interface

Description automatically generated **Optimization:** https://github.com/ClaytonsCreations/Non-Profit\_application\_deepLearning/blob/master/AlphabetSoupCharity\_Optimization.ipynb

* Dropped the EIN, and NAME columns as it had been advised for the Starter\_Code and didn’t see the value of either column being added back.
* I then used binning on the APPLICATION TYPE and CLASSIFICATIONS columns but increase the unique values and placed less in the “Other” column. Based on the reading, deep learning works better with the more data it has available.
* For the Model, I included 2 layers at 50 and 100 as I had in the Starter\_Code.

Text

Description automatically generated

Very slight improvement from the Starter\_Code, not significant enough to feel comfortable continuing the same path.

**Optimization 2:** https://github.com/ClaytonsCreations/Non-Profit\_application\_deepLearning/blob/master/AlphabetSoupCharity\_Optimization2.ipynb

* Dropped the EIN, NAME, and STATUS columns as I felt the STATUS also wasn’t providing any useful information.
* I kept the binning of APPLICATION\_TYPE at the same level but changed CLASSIFICATION to have less unique and higher “Other” count.
* I attempted to adjust the hidden layers by significantly changing the values of layer1 and layer2 as well, I added a third layer.

Text

Description automatically generated

This reduced the accuracy slightly but was the lowest accuracy so far. Thinking about the data, I decided that maybe more data would be better.

**Optimization 3:** https://github.com/ClaytonsCreations/Non-Profit\_application\_deepLearning/blob/master/AlphabetSoupCharity\_Optimization3.ipynb

* I dropped EIN and STATUS as I felt they still weren’t useful, but for the first time I kept NAME.
* I kept the binning relatively minimal again wanting to keep as much data available to the model. Did not even create binning for CLASSIFICATION as I had in the last attempts.
* For the model, I did keep the 3 hidden layers, but based off some google searches, I reduced the numbers significantly.

Graphical user interface, application

Description automatically generated

This brought up the accuracy up over 6% from my last attempt brining it above the 75% requirement.

**Summary:**

It appears based on my 4 different models, adding the NAMES data and reducing the neurons in the layers made the biggest impacts. After my google searches I found that having too many neurons could be overfitting the model. We could have utilized the keras functional model vs the sequential that may have given us more flexibility to define multiple input and outputs. The functional model is more advanced but can handle more complex data. I recommend that because I feel if we were able to tie additional data together in addition to the “IS\_SUCCESSFUL” we could have done more with the “ASK\_AMT” and “INCOME\_AMT” to determine if those influenced them being successful and giving us more accuracy in our model.