Analysis

**Overview** of the analysis: We were tasked with creating a model that can help a foundation select the applicants for funding with the best chance of success in their ventures. We had to use the features in the provided dataset to create a binary classifier that can predict whether applicants will be successful if funded.

The dataset provided contains more than 34,000 organizations that have received funding from over the years. This data was used as training to make predictions with the model.

1. **Results**:

* Data Preprocessing
  + What variable(s) are the target(s) for your model?
    - The target for the model is the column that identifies whether loans made in the past were successful ventures or not.
  + What variable(s) are the features for your model?
    - The features are the other 44 dimensions left in the dataframe after separating out the target and dropping EIN and name columns.
  + What variable(s) should be removed from the input data because they are neither targets nor features?
    - The EIN and name columns weren’t needed and were removed.
* Compiling, Training, and Evaluating the Model
  + How many neurons, layers, and activation functions did you select for your neural network model, and why?
    - I chose 70 neurons, three layers, and ReLU activation function. I chose ReLU because it’s faster and isn’t impacted by the vanishing gradient problem.
  + Were you able to achieve the target model performance?
    - Yes. I was able to achieve accuracy greater than 75% on the second attempt at optimization.
  + What steps did you take in your attempts to increase model performance?
    - I chose a few different configurations of neurons, layers, and activation functions to achieve greater accuracy. I also tried altering the number of dimensions that were used to train the model. The model that was the most accurate was the one where the names were binned.

1. **Summary**: The first model did a decent job of predicting successful loan outcomes. For the first optimization, I added more neurons for a total of 290. Accuracy actually decreased in the first attempt at optimization. The added neurons overtrained the training data for the model, so it didn’t do well with data it hadn’t seen yet. I was able to achieve greater than 75% accuracy with the third attempt at optimization. Instead of removing the name column, I chose to bin the names. The extra dimensions created by binning the names assisted in training the model.