The purpose of this analysis is to find the best tool that can predict whether applications will be successful if they are funded by the nonprofit foundation Alphabet Soup.

The target is the desired output for the given input X. Initially, the instructions wanted the data prepared by removing the EIN and NAME columns. The features for this model are all the columns after those two were removed. The target of the model is finding if the money was used effectively by displaying it as the output y. I did not remove any more columns of data because they were all important factors that should be used to determine how successful applicants will be.

I did three attempts to improve optimization. In the first run, I used input features listed for as many as there are for X\_trained\_scaled. For the neurons I used roughly a third of the 117 columns displayed in the one-hot encoding dataframe, which for the first layer is 35, and the second layer I used 5. I only used two hidden layers with relu for both activation functions, and lastly the output as sigmoid. I used this method for modeling, as usually having relu as hidden, and output as sigmoid is the general rule of thumb for a good outcome.The target model performance failed, the accuracy of the model was only 0.727 with a loss of 0.558. The first attempt ended up being the best result. In an attempt to optimize I added an additional layer because this allows more interactions between the variables; I added more neurons in the second attempt as well because it can speed up the model and may reduce loss. On the final attempt, I changed the activation function as usually it would drastically change the model interpreting the inputs, but still only gave a .726 for the accuracy. I also changed the epochs from 100 to 50, but that shouldn’t have gone down as adding more epochs would increase the likelihood of the model achieving optimal weight coefficients.

Overall the results did not reach over 75% accuracy given the three attempts to improve optimization.The data has high variance and is overfitting. There could possibly not be enough complexity in the training data and converges too quickly or test and training data are unbalanced. In order to fix this issue we can add more training data. Another option is to add training data and change the split of the training and testing date of the original input data.