1. **Overview of the analysis:** The purpose of this analysis is to predict whether or not applicants for funding will be successful

2. **Results:**

Data Preprocessing

a. **What variable(s) are considered the target(s) for your model?**

The IS_SUCCESSFUL column in the classification_numeric dataframe that converted all the categorical to numeric values

b. **What variable(s) are considered to be the features for your model?**
   i. APPLICATION_TYPE—Alphabet Soup application type
   ii. AFFILIATION—Affiliated sector of industry
   iii. CLASSIFICATION—Government organization classification
   iv. USE_CASE—Use case for funding
   v. ORGANIZATION—Organization type
   vi. STATUS—Active status
   vii. INCOME_AMT—Income classification
   viii. SPECIAL_CONSIDERATIONS—Special consideration for application
   ix. ASK_AMT—Funding amount requested

c. **What variable(s) are neither targets nor features, and should be removed from the input data?**
   i. EIN and NAME

d. **How many neurons, layers, and activation functions did you select for your neural network model, and why?**
   i. Number of Neurons:
      # number of layer1 neurons = 2*(number of inputs=43) = 86 ~ 80
      hidden_nodes_layer1 = 80
      # number of layer2 neurons: Between (input=80) and (output=1 - classifier)
      hidden_nodes_layer2 = 30
   ii. We had 3 hidden layers: first hidden layer, second hidden layer, and output layers
   iii. Activation function was Sigmoid. Hidden layers activation function was Relu.
      ```
      # 1st Hidden Layer: 3520 params = [43 inputs (from input
      layer) * 80 neurons] + (80 bias terms)
      ```

```
# 2nd Hidden Layer: 2430 params = [80 inputs (from 1st hidden
layer) * 30 neurons] + (30 bias terms)
# Output Layer: 31 params = [30 inputs (from 2nd hidden layer)
* 1 neuron] + (1 bias term)

Total params: 6,141
Trainable params: 6,141
Non-trainable params: 0
```

e. **Were you able to achieve the target model performance?**

Yes I was close to 75% accuracy

f. **What steps did you take to try and increase model performance?**

To increase model performance I decreased the
`application_counts[application_counts < 68 and`
`classification_counts[classification_counts < 290`
`From the first notebook`

Summary: Summarize the overall results of the deep learning model. Include a recommendation for how a different model could solve this classification problem, and explain your recommendation.

- **Results Summary:**
    - Good results → 72-73% accuracy in predict whether or not applicants for funding will be successful

- **Recommendation:** A Different model like Decision Tree Classifier, Random Forest, or Logistic Regression = all are binary classification
    - All offer a simpler approach to classify the data
    - Perform just as well as this neural network
    - All have about 72% accuracy