1. **Overview** of the analysis: Explain the purpose of this analysis.
2. **Results**: Using bulleted lists and images to support your answers, address the following questions.
    - Data Preprocessing
    - 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
    - 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
    - What variable(s) are neither targets nor features, and should be removed from the input data?
        - EIN and NAME
    - 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

    - Were you able to achieve the target model performance?
        - Yes, almost 75% accuracy
    - What steps did you take to try and increase model performance?
        - To increase model performance, decreased the application_counts [application_counts < 68 and classification_counts [classification_counts < 290 From the first notebook

3. **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 → 70+ % accuracy in predicting 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
            1. All offer a simpler approach to classify the data
            2. Perform just as well as this neural network
            3. All have about 72% accuracy