**Report**

**Overview** of the analysis: Explain the purpose of this analysis.

The purpose of this analysis is to create a model that can effectively predict, based on certain variables (Application type, Affiliation, Classification among others), if the project will be successful or not. The goal is to help Alphabet Soup make better decisions when selecting which applicant should receive funding or not.

**Results**: Using bulleted lists and images to support your answers, address the following questions:

* Data Preprocessing
  + What variable(s) are the target(s) for your model?
* The target variable is “Is\_Successful” -Since the goal is identifying if the venture will be successful or not.
  + What variable(s) are the features for your model?
* The features are the following: "APPLICATION\_TYPE", "AFFILIATION", "CLASSIFICATION","INCOME\_AMT", "USE\_CASE", "ORGANIZATION", "SPECIAL\_CONSIDERATIONS"
  + What variable(s) should be removed from the input data because they are neither targets nor features?
* The variable “EIN” and the variable “Name”
* Compiling, Training, and Evaluating the Model
  + How many neurons, layers, and activation functions did you select for your neural network model, and why?
* Input Layer: 7 neuros (7 input features)
* First Hidden Layer: 80 neurons with ReLU activation.
* Second Hidden Layer: 40 neurons with ReLU activation.
* Output Layer: 1 neuron with Sigmoid activation.
  + Were you able to achieve the target model performance?
* I tried more than 3 different models but I was not able to achieve the accuracy of 75%
  + What steps did you take in your attempts to increase model performance?
* I changed the numbers of neuros for the first and second layer.

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

* The deep learning model achieved a loss of 0.5563 and an accuracy of 72.62% on the test data. This means that the performance is moderate but can definitely be improved. A Convolutional Neural Network could perhaps solve this problem better. The reason is that CNNs automatically learn features. Therefore, there would be no need for manual feature engineering. Also, CNNs use parameter sharing to detect features across different parts of the input, leading to fewer parameters compared to fully connected networks. As a result, this can prevent overfitting.