**Overview of the analysis**

The analysis and preprocessing steps aim to prepare the data for effective model training and evaluation. These steps help address issues like data cleanliness, categorical variable handling, and model architecture selection. Our ultimate goal is to develop a machine learning model that accurately predicts the target variable while generalizing well to new, unseen data

**Data Preprocessing**

What variable(s) are the target(s) for your model?

The target variable for my machine learning model was 'IS\_SUCCESSFUL.' This variable is used as the target in a binary classification problem, where the goal is to predict whether a charity organization will be successful (IS\_SUCCESSFUL = 1) or not (IS\_SUCCESSFUL = 0) based on the input features and characteristics of each organization.

What variable(s) are the features for your model?

The features included columns such as 'APPLICATION\_TYPE,' 'AFFILIATION,' 'CLASSIFICATION,' 'USE\_CASE,' 'ORGANIZATION,' 'STATUS,' 'INCOME\_AMT,' 'SPECIAL\_CONSIDERATIONS,' and 'ASK\_AMT,'

What variable(s) should be removed from the input data because they are neither targets nor features?

The variables 'EIN' and 'NAME' were the two variables that should be removed from the input data because they were neither target variables nor features used for prediction. These variables were likely identifiers and organization names, respectively, which did not provide meaningful information for the model's predictive task.

**Compiling, Training, and Evaluating the Model**

How many neurons, layers, and activation functions did you select for your neural network model, and why?

Going down to one hidden layer helped reduce the risk of overfitting and allow the model to focus on the most important patterns in the data. Going down from 200 neurons to 20 neurons also increased accuracy by likely reducing model complexity, speeding up training, and increasing memory efficiency.

Were you able to achieve the target model performance?

I got close at Loss: 2.7257988452911377, Accuracy: 0.6963265538215637

What steps did you take in your attempts to increase model performance?

* Decrease layers
* Decreasing neurons
* Increasing the Number of Epochs
* Decreasing the Batch Size
* Switching the optimizer from ‘sgd’ to ‘adam’

**Summary**

The deep learning model's overall results, as discussed earlier, have shown some improvements in accuracy by making various modifications to the model architecture, such as reducing the number of neurons and changing the optimizer to 'adam.' However, the results still fall short of the target accuracy of 75%. Here's a summary of the overall results and a recommendation for a different approach:

Summary of Results:

The model was developed for a binary classification problem to predict whether a charity organization is successful ('IS\_SUCCESSFUL' = 1) or not ('IS\_SUCCESSFUL' = 0) based on various input features.

Multiple architectural adjustments were made, including changes to the number of neurons and optimizer, which improved accuracy compared to the initial model.

Despite the improvements, the model has not achieved the desired accuracy level of 75%, indicating room for further enhancement.