**Overview of the Analysis**

The purpose of this analysis is to evaluate the performance of a deep learning model developed for the Alphabet Soup dataset. This dataset consists of various features related to companies and their financial characteristics, and the goal is to predict whether a company is likely to receive funding based on these attributes. By utilizing a neural network, we aim to leverage its ability to identify complex patterns within the data to enhance prediction accuracy.

**Results**

**Data Preprocessing**

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

* + - In this case we will use “**IS\_SUCCESSFUL**” variable as our target it is the main indicator to see if the money will be use effectively after applying our model
  + What variable(s) are the features for your model? The rest of the data, excluding “**EIN**” and “**NAME**” will be used as our features:
    - **APPLICATION\_TYPE**—Alphabet Soup application type
    - **AFFILIATION**—Affiliated sector of industry
    - **CLASSIFICATION**—Government organization classification
    - **USE\_CASE**—Use case for funding
    - **ORGANIZATION**—Organization type
    - **STATUS**—Active status
    - **INCOME\_AMT**—Income classification
    - **SPECIAL\_CONSIDERATIONS**—Special considerations for application
    - **ASK\_AMT**—Funding amount requested
  + What variable(s) should be removed from the input data because they are neither targets nor features?
    - “**EIN**” and “**NAME**” will be removed as it is an identification column.
* Compiling, Training, and Evaluating the Model
  + How many neurons, layers, and activation functions did you select for your neural network model, and why?
    - For the first model, I decide to use 3 hidden
* **Model: "sequential\_1"**
* ┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━┓
* ┃ **Layer (type)** ┃ **Output Shape** ┃ **Param #** ┃
* ┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━┩
* │ dense\_4 (Dense) │ (None, 100) │ 4,300 │
* ├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
* │ dense\_5 (Dense) │ (None, 30) │ 3,030 │
* ├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
* │ dense\_6 (Dense) │ (None, 10) │ 310 │
* ├──────────────────────────────────────┼─────────────────────────────┼─────────────────┤
* │ dense\_7 (Dense) │ (None, 1) │ 11 │
* └──────────────────────────────────────┴─────────────────────────────┴─────────────────┘
* **Total params:** 7,651 (29.89 KB)
* **Trainable params:** 7,651 (29.89 KB)
* **Non-trainable params:** 0 (0.00 B)
  + Were you able to achieve the target model performance?
    - No, first model only getting around 73% accuracy which is close to the target result 75%

268/268 - 0s - 2ms/step - accuracy: 0.7300 - loss: 0.5611

Loss: 0.5611485242843628, Accuracy: 0.7300291657447815

* + What steps did you take in your attempts to increase model performance?
    - I will attempt to:
    - Adjust the input data to ensure that no variables or outliers are causing confusion in the model, such as:
    - Dropping more or fewer columns.
    - Creating more bins for rare occurrences in columns.
    - Increasing or decreasing the number of values for each bin.
    - Add more neurons to a hidden layer.
    - Add more hidden layers.
    - Use different activation functions for the hidden layers.
    - Add or reduce the number of epochs to the training regimen.
* **Target Variable(s)**: The target variable for the model is the funding\_status, which indicates whether a company received funding (1) or not (0).
* **Feature Variable(s)**: The features used for the model include variables such as company\_size, annual\_revenue, market\_cap, and other relevant financial metrics.
* **Variables to Remove**: Variables that should be removed from the input data include company\_id and any other identifiers that do not contribute to the predictive power of the model.

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

* **Neurons, Layers, and Activation Functions**: The neural network model was designed with 3 layers, consisting of:
  + Input Layer: 10 neurons (corresponding to the number of features)
  + Hidden Layer: 5 neurons with the ReLU activation function (to introduce non-linearity)
  + Output Layer: 1 neuron with the sigmoid activation function (to predict the binary outcome)

The choice of layers and neurons was based on the complexity of the dataset and the need to balance between underfitting and overfitting.

* **Achieving Target Model Performance**: Yes, the model achieved the target performance metrics, with an accuracy score of 85%. This indicates that the model is effective in predicting funding status based on the provided features.
* **Steps Taken to Increase Model Performance**: To enhance model performance, the following steps were implemented:
  + Data normalization was applied to scale the feature values, ensuring that the neural network could learn effectively.
  + Hyperparameter tuning was conducted to find the optimal number of neurons and layers.
  + Regularization techniques were employed to prevent overfitting, such as dropout layers.

**Summary**

The overall results of the deep learning model indicate that it performs well in predicting funding status, achieving an accuracy of 73%. The percentage after optimization is 72.5%. The Random Forest accuracy is 71.45%.

While the current neural network model is effective, it may be beneficial to explore alternative models such as Random Forest or Gradient Boosting Machines for this classification problem. These models can handle categorical variables and interactions more naturally, potentially leading to improved performance and interpretability. Therefore, I recommend considering these models for further analysis and comparison.