Deep Learning Model for Alphabet Soup Performance Report

1. **Overview of the Analysis**

The purpose of this analysis is to create a deep learning model to help Alphabet Soup predict whether an applicant will be successful in their venture based on several features provided in the dataset. By building a binary classification model, we aim to guide Alphabet Soup in making informed funding decisions, increasing the likelihood of success among funded applicants.

1. **Results**

***Data Preprocessing***

Target Variable: the target variable for this analysis is IS\_SUCCESSFUL, which indicates whether an applicant was successful after receiving funding from Alphabet Soup.

Feature Variables:

The features include APPLICATION\_TYPE, AFFILIATION, CLASSIFICATION, USE\_CASE, ORGANIZATION, STATUS, INCOME\_AMT, SPECIAL\_CONSIDERATIONS, ASK\_AMT

Removed Variables:

EIN and NAME columns were removed from the dataset as they are identification columns and do not contribute to the predictive model.

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

Neurons, Layers, and Activation Functions:

The first neural network architecture was as follows:

First hidden layer: 80 neurons with the relu activation function.

Second hidden layer: 30 neurons with the relu activation function.

Output layer: 1 neuron with the sigmoid activation function for binary classification.

Epochs: 100

The relu activation function was chosen for the hidden layers because it helps in capturing non-linear relationships, while the sigmoid function was used in the output layer since this is a binary classification problem.

Model Performance:

The model achieved an accuracy of 72.29%, which fell short of the target accuracy of 75%.

A screenshot of a computer program

Description automatically generated

***Steps Taken to Improve Performance:***

Several optimization techniques were applied to improve the model's performance:

1. Adding more neurons to each hidden layer to increase the model's learning capacity and increased the number of epochs. The result is slightly increased to 72.36%.

A screenshot of a computer code

Description automatically generated

1. Created more bins for rare occurrences in columns, the result is slightly increased to 72.61%.

A screenshot of a computer program

Description automatically generated

1. Adjusting the activation functions by experimenting with tanh, selu, and relu, the result slightly decreased to 72.16%

A screenshot of a computer

Description automatically generated

Despite trying several optimization techniques, the model did not reach the desired performance.

***Summary***

The deep learning model achieved a reasonable accuracy of 72.29%, but did not meet the 75% target performance. Several optimization techniques were applied, such as increasing neurons, adding layers, adjusting activation functions, but the accuracy gains remained limited. To further improve the model, recommend exploring other machine learning models for this classification problem such as Random Forest: A random forest classifier may perform better with this dataset, as it handles categorical features well and can capture complex feature interactions. Also, we could try Gradient Boosting method to help improve model performance, as they are known to work well for structured tabular data.