**Report on Neural Network Model for Alphabet Soup**

**Overview of the Analysis**

The primary purpose of this analysis is to develop a binary classification model that predicts the success of organizations funded by Alphabet Soup based on their features. By employing machine learning techniques, particularly a neural network, we aim to identify patterns in the data that correlate with successful funding outcomes. This will help Alphabet Soup select applicants with the highest probability of using funding effectively, ultimately increasing the impact of their financial resources.

**Results**

**Data Preprocessing**

* **Target Variable(s):**
  + **IS\_SUCCESSFUL**: Indicates whether the funding was utilized effectively (1 for successful, 0 for unsuccessful).
* **Feature Variable(s):**
  + **APPLICATION\_TYPE**: Type of application submitted.
  + **AFFILIATION**: Sector of industry the organization is affiliated with.
  + **CLASSIFICATION**: Classification of the organization.
  + **USE\_CASE**: Use case for which funding is requested.
  + **ORGANIZATION**: Type of organization.
  + **STATUS**: Active status of the organization.
  + **INCOME\_AMT**: Income classification of the organization.
  + **SPECIAL\_CONSIDERATIONS**: Any special considerations noted for the application.
  + **ASK\_AMT**: Amount of funding requested.
* **Variable(s) to Remove:**
  + **EIN**: Employer Identification Number.
  + **NAME**: Name of the organization.
* **Unique Values Analysis:**
  + Analyzed categorical columns with more than 10 unique values, combined rare categories into an "Other" category as needed.
* **Encoding:**
  + Categorical variables were converted to numerical values using pd.get\_dummies().
* **Data Splitting:**
  + The dataset was divided into feature array X and target array y, and split into training and testing datasets using train\_test\_split.
* **Scaling:**
  + Features were scaled using StandardScaler to normalize the data.

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

* **Neurons and Layers:**
  + **Input Layer:** 43 input features.
  + **First Hidden Layer:** 16 neurons with the ReLU activation function.
  + **Second Hidden Layer:** 12 neurons with the ReLU activation function.
  + **Output Layer:** 1 neuron with the Sigmoid activation function.
* **Model Structure:**
  + The model includes an input layer, two hidden layers, and an output layer.
* **Model Compilation:**
  + Compiled using Adam optimizer and binary cross-entropy loss function.
* **Training:**
  + Trained for 100 epochs, saving model weights every five epochs.
* **Model Evaluation:**
  + Evaluated on the test dataset, achieving a loss of 0.539 and an accuracy of 72.70% (test accuracy) with a training accuracy of 74.04% with an overall accuracy of 73%.
* **Performance Improvement Attempts:**
  + First Attempt - Changed the number of epochs from 100 to 60 since graph showed that the accuracy increased and decrease when the model learns and added two more hidden layers. But the accuracy dropped a little.
  + Second Attempt -Changed the number of epochs back to 100, added another hidden layer, changed the activation function of output layer to tanh function and increased the number of neurons in the 3rd and 4th layer by 2 each. Accuracy increased a little than the original model.
  + Third Attempt - Changed the number of epochs to 80, added another hidden layer, with 8 neurons. Accuracy increased than the original model. But couldn’t reach 75% accuracy.

**Visualizations**

**Model Training Accuracy and Loss Over Epochs**



Original Model

Accuracy generally increases while loss decreases as the model learns.



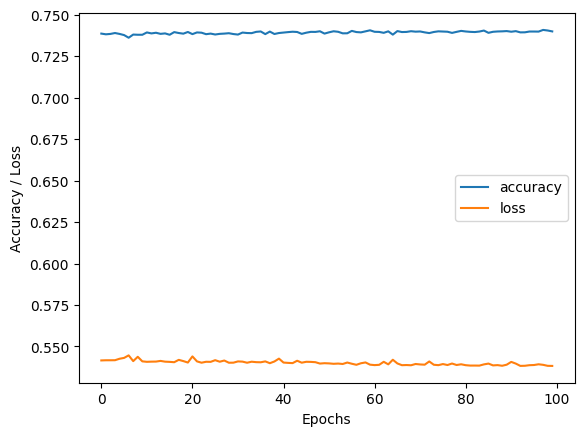
First Attempt

Accuracy decreased a little compared to the original one



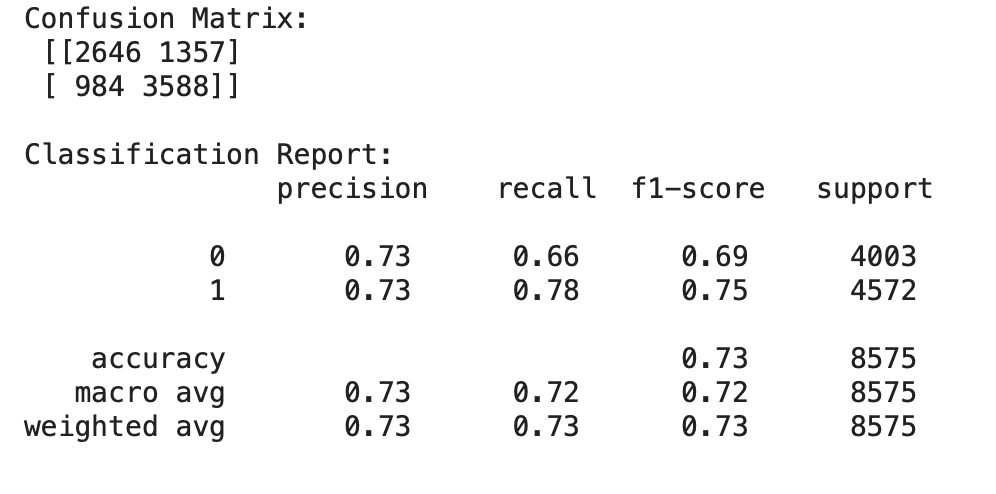
Second Attempt

Accuracy increased a little compared to the original model



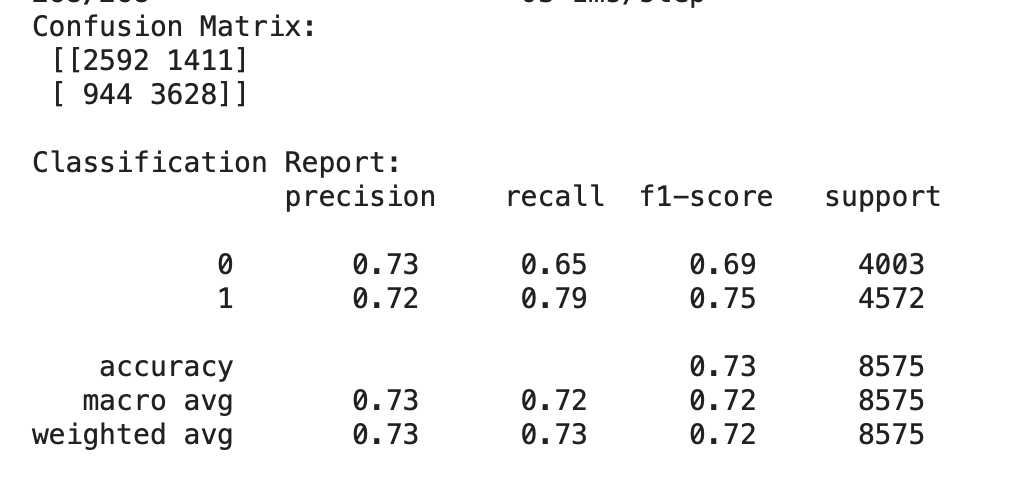
Third Attempt

Accuracy increased upto 74.09%



Classification report and the confusion matrix of original model

Overall Accuracy - 73%



Classification report and the confusion matrix of final model

Overall Accuracy - 73%

**Summary**

Overall, the deep learning model developed for predicting the success of Alphabet Soup funding applications achieved an accuracy of 73%. The optimizations used increased the training accuracy a very little but dropped the test accuracy hence the overall accuracy remained the same. This structured approach to preprocessing and model design allowed for effective learning from the dataset.

**Recommendations for Alternative Models**

While the neural network provided satisfactory results, exploring alternative models could enhance prediction accuracy:

* **Random Forest Classifier:** Handles categorical variables well and reduces overfitting through averaging.
* **Gradient Boosting Machines (GBM):** Captures complex relationships in the data, potentially improving accuracy.

These alternatives could provide different insights into the data and lead to better decision-making for funding selections.