**Overview:**

The nonprofit foundation Alphabet Soup wants a tool that can help it select the applicants for funding with the best chance of success in their ventures. With knowledge of machine learning and neural networks, I’ll use the features in the provided dataset to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup.

From Alphabet Soup’s business team, I have received a CSV containing more than 34,000 organizations that have received funding from Alphabet Soup over the years. Within this dataset are a number of columns that capture metadata about each organization, such as:

EIN and NAME—Identification columns

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 consideration for application

ASK\_AMT—Funding amount requested

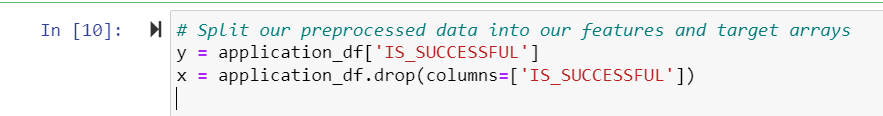
IS\_SUCCESSFUL—Was the money used effectively

The purpose of this analysis is to build and optimize a deep learning model to predict the success of charity donations using the Alphabet Soup dataset. The goal is to achieve a predictive accuracy higher than 75%.

**Results:**

1. **Data Preprocessing**

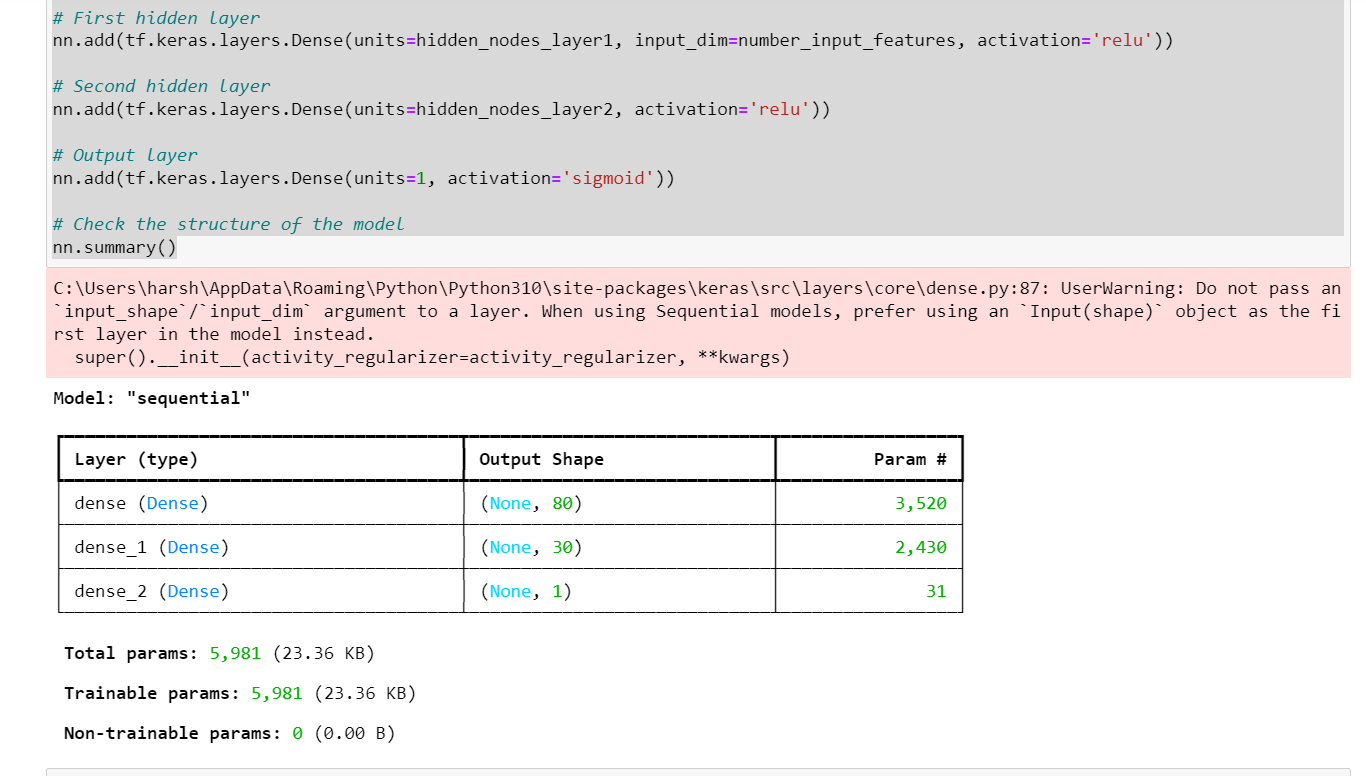
* Target Variable(s): The target variable for the model is the binary IS\_SUCCESSFUL column, indicating whether a charity donation was successful.
* Feature Variable(s): The feature variables include all other columns that provide information about the donations and the organizations.
* Removed Variable(s): Variables that are neither targets nor features, such as EIN and NAME, are removed from the input data to avoid unnecessary noise in the model.



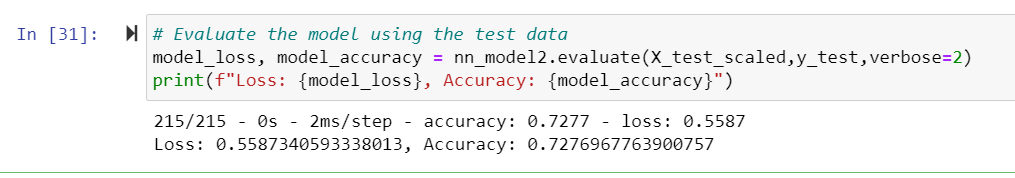
1. **Compiling, Training, and Evaluating the Model**

* Neurons, Layers, and Activation Functions: The neural network model consists of three layers:
* First hidden layer: 80 neurons, ReLU activation function
* Second hidden layer: 30 neurons, ReLU activation function
* Output layer: 1 neuron, Sigmoid activation function

These parameters were chosen to provide sufficient capacity for the model to learn complex patterns in the data while avoiding overfitting.



* Target Model Performance: **The target performance of 75% accuracy was not achieved despite various optimization attempts. The highest accuracy obtained was approximately 72.76%.**



Optimization Steps: Several methods were used to optimize the model, including:

* **Adjusting Input Data**: Dropped irrelevant columns and binned categorical variables with rare occurrences.
* **Increasing Model Complexity**: Added more neurons and hidden layers to the model.
* **Changing Activation Functions**: Tested different activation functions for hidden layers.
* **Adjusting Training Parameters**: Increased epochs to 300, set batch size to 32, and used a 20% validation split.

**Summary**

Despite multiple attempts to optimize the model, the target accuracy of 75% was not achieved. The final model reached an accuracy of approximately 72.76%. To further improve performance, other machine learning models, such as Random Forest or Gradient Boosting, could be explored. These models are often effective in handling complex datasets with many categorical variables and may provide better predictive accuracy.