**Charity Funding Report**

**1. Overview**

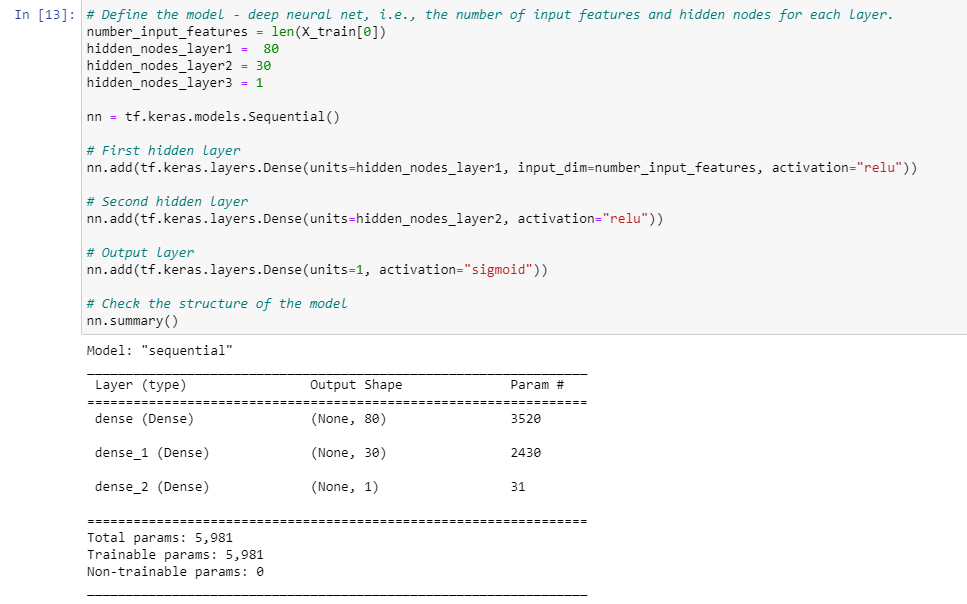
The non-profit foundation Alphabet Soup wants to select applicants for its charity funding program, and needs to find a system that will help it to select the applicants for funding with the best chance of success in their ventures. The main goal of this analysis is to create a binary classifier by using the appropriate machine learning model that can predict whether or not applicants will be successful if funded by Alphabet Soup. From Alphabet Soup’s business team, we received a CSV containing more than 34,000 organizations that have received funding from Alphabet Soup over the years. The dataset contains 12 columns that capture metadata about each beneficiary.

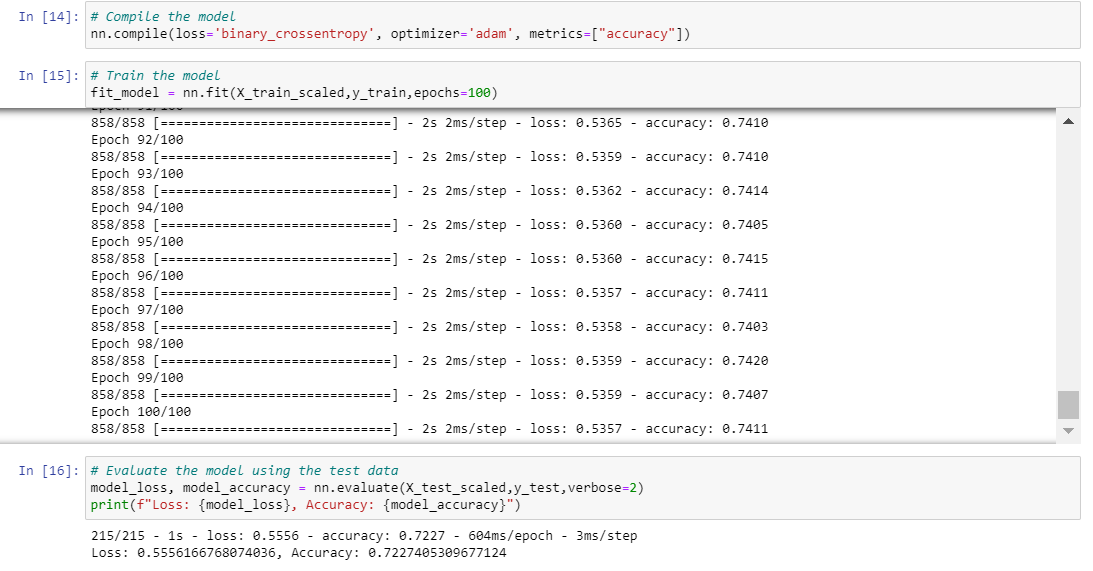
**2. Results: Data Preprocessing, and Compiling, Training and Evaluating the Model**

**Data pre-processing:**

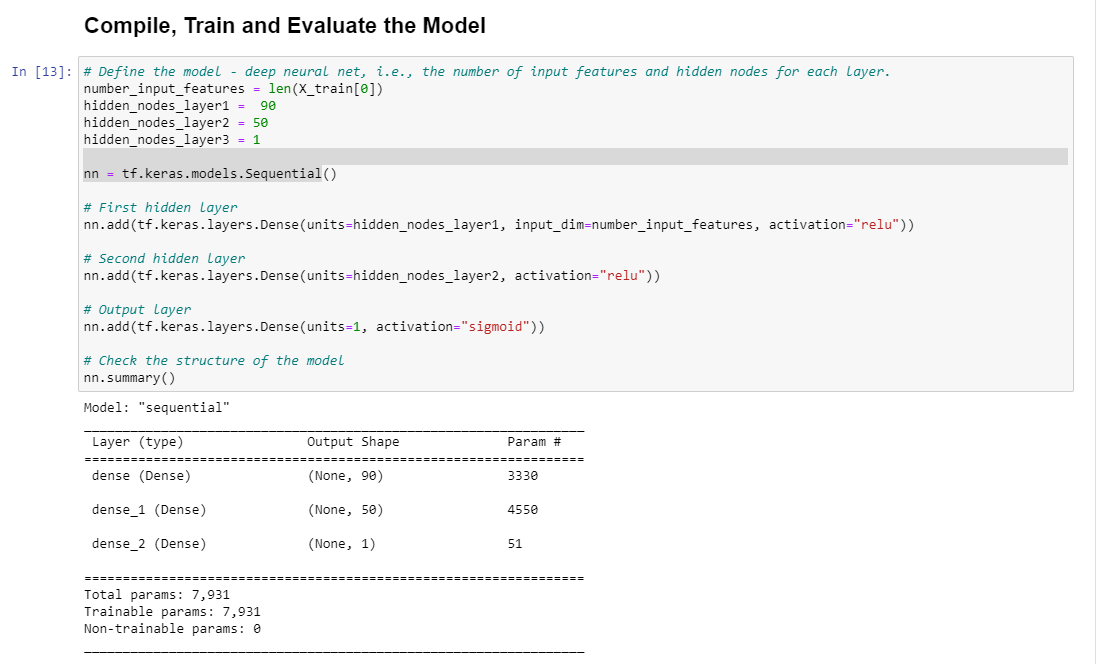
* A target is **IS\_SUCCESSFUL** for the model to find if the applicants will be successful, if funded.
* For the first attempt, EIN and NAME were dropped and weren’t considered being useful. All the features were used except EIN and NAME.
* For optimizing model, EIN, NAME, USE\_CASE and SPECIAL\_CONSIDERATIONS columns were dropped and all features were used except the columns removed.

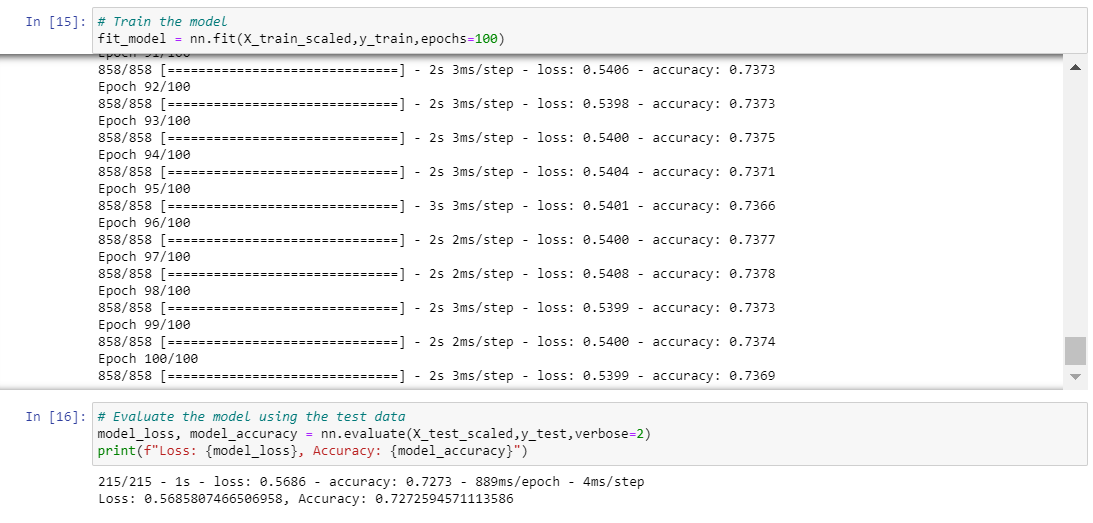
**Compiling, Training, and Evaluating the Model:**

In the first model (Start\_code\_Ilia.ipynb), I used two hidden layers with 80 and 30 neurons, and a 'relu' activation function for both hidden layers. 



The model achieves an accuracy of 72%, which is below the 85% accuracy required for a basic model. In order to improve the performance of the model, another optimizing model was built (AlphabetSoupCharity\_Optimzation.ipynb) starting by dropping four columns of EIN, NAME, USE\_CASE and SPECIAL\_CONSIDERATIONS and increased the number of neurons in the two hidden layers.





However, the performance of the model was not improved. The loss slightly increased from 0.5556 to 0.5686, and the accuracy rate increased slightly from 0.7227 to 0.7273.

**3. Summary**

The model's best accuracy rate was 73%, and after building another model, the value was not improved by increasing the number of layers, or dropping more columns in the dataset.