**Analysis Report**

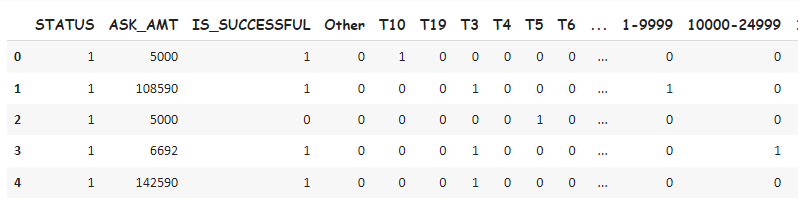
**Background:**

Alphabet Soup, a non-profit foundation needs to filter their applicants for the best chance of success.  Use machine learning and neural networks knowledge, I have built a binary classifier for predicting applicants' potential success, which shall assist Alphabet Soup in better reward from their funding.

**Data source:** a csv file containing 34000+ historic records

**Pre-process**

* Read csv into Pandas and remove extraneous EIN and NAME
* Check unique values, decide input point numbers
* Pick cut-off and combine rare to ‘Other’, encode categorical variables to numeric (APPLICATION, CLASSIFICATION)
* Split to feature X and target y arrays, then into training and testing datasets
* Standard scale, fit with training data, transform



Q1: What variable(s) are the target(s) for your model?

Answer: “IS\_SUCESSFUL”, 1-Yes, 0-No

Q2: What variable(s) are the features for your model?

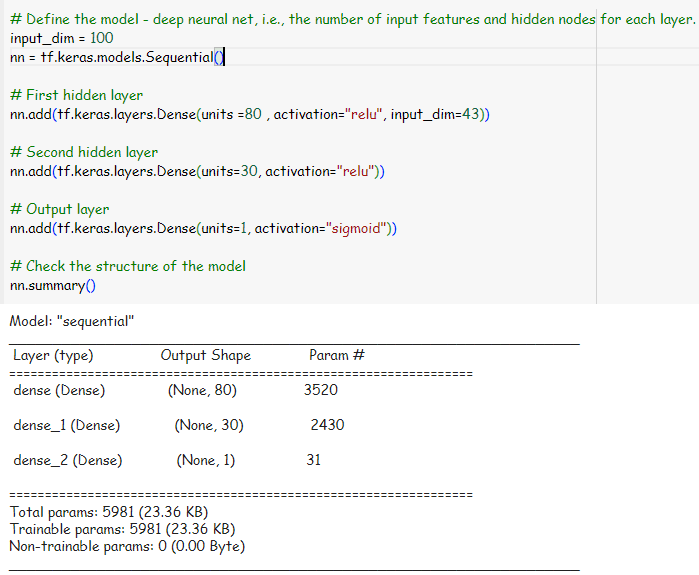
Answer: “APPLICATION DATA”, “CLASSIFICATION”

Q3: What variable(s) should be removed from the input data because they are neither targets nor features?

Answer: “EIN” and “NAME”

**Compile, Train, Evaluate Model**

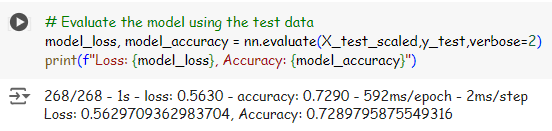
* use TensorFlow, build neural network model, assign input feature and node numbers
* set up 1st, 2nd hidden layers and output layer with appropriate activation function
* check model structure, compile and train
* evaluate model with test data, get loss and accuracy
* export HDF5



Q4: How many neurons, layers, and activation functions did you select for your neural network model, and why?

Answer: 1st hidden layer, 80 neurons, relu activate; 2nd hidden layer, 30 neurons, relu activate; output layer, 1 neuron, sigmoid activate

**Result**

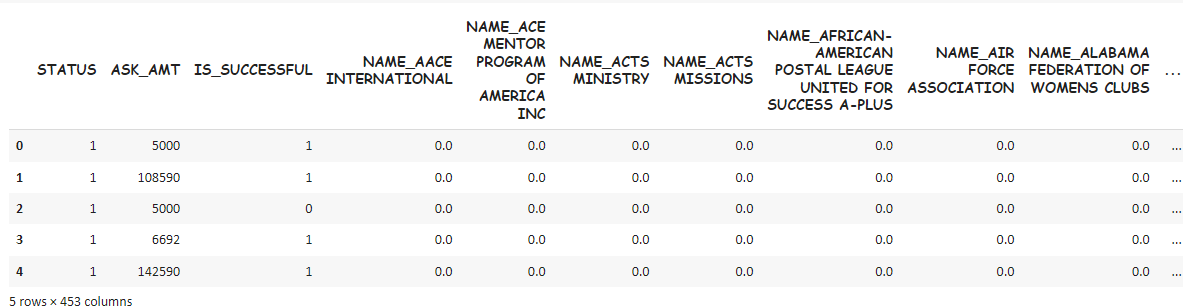
****

Q5: Were you able to achieve the target model performance?

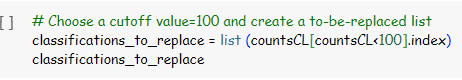
Answer: No. This model created 5981 parameters, and get 72.9% of accuracy.

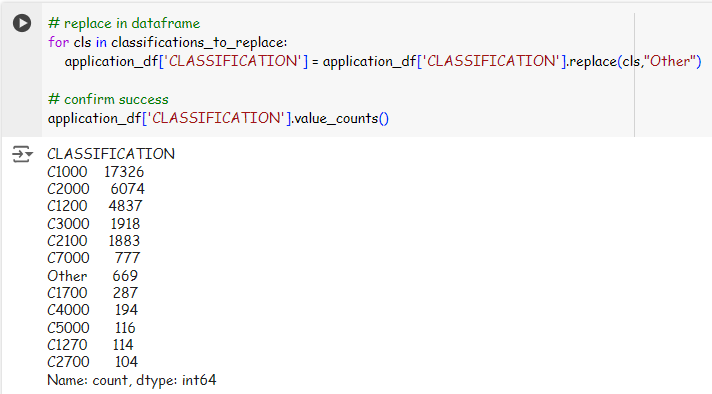
It is necessary to optimize the model to get an accuracy over the 75% threshold.

**Optimization:**

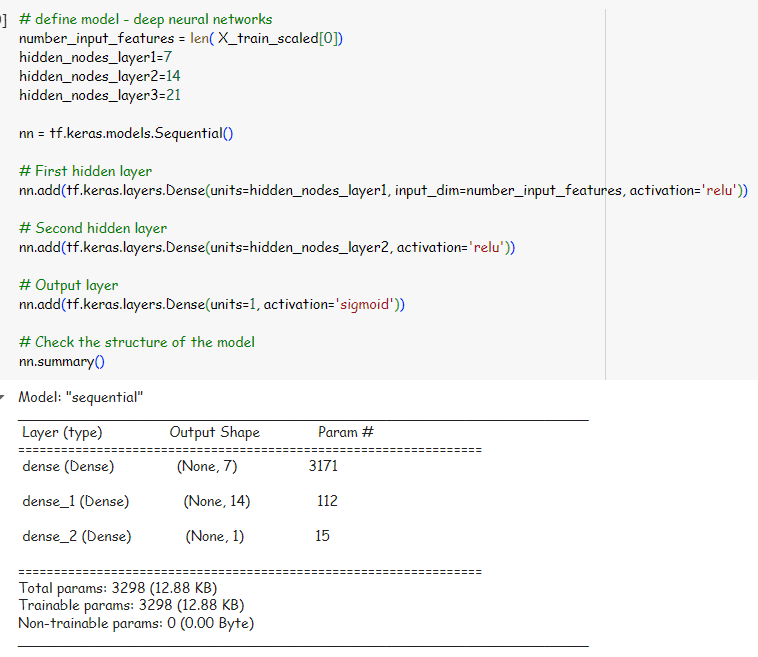
Only drop “EIN” and keep “NAME”, so I have more feature to analyze

Change “CLASSIFICATION” cut-off to 100, so I have more values to analyze

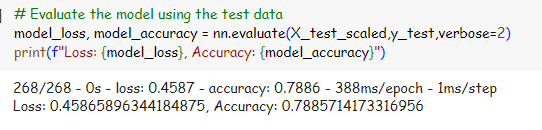




Adjust model settings, number of neurons in layers



**Optimized result:**



Q6: What steps did you take in your attempts to increase model performance?

Answer: I took 3 actions, 1- add “NAME” column back, so I have more column to use as feature; 2- increase number of values for “CLASSIFICATION” by change cut-off from 1000 to 100; 3- adjust number of neurons in layers. As a result, it returned an accuracy of 78.9%, which meet the requirement.