Rebecca Melo

February 15, 2022

**Charity Funding Predictor: Report on the Neural Network Model**

The purpose of this analysis is to create an algorithm to predict whether or not applicants for funding will be successful. Machine Learning and neural networks will be used to create a binary classifier that is capable of predicting whether applicants will be successful if funded.

Data Preprocessing

The target variable for the model will be "IS\_SUCCESSFUL". This column has values of 1 and 0 which helps us determine if the charity fund is successful (1) or not successful (0). After columns "EIN" and "Name" are dropped, the remaining columns are features for the model.

Dropped the EIN and NAME columns then determined the number of unique values for each column. Looked at APPLICATION value counts, and CLASSIFICATION Value counts for binning. Used the number of data points for each unique value to pick a cutoff point to bin "rare" categorical variables together in a new value, `Other`, and then check if the binning was successful. Use `pd.get\_dummies()` to encode categorical variables.

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

Created neural network with two hidden layers as shown below. The number of hidden nodes were selected based on the number of features initially. The number of hidden nodes for the first layer should be at least equal to the number of features. The started code allowed me to start with less hidden nodes initially because it would be easier computation, and see how accurate that is. Also, I want to maintain a balance between training accuracy and testing accuracy and avoid over fitting of the model.

The model generated is a sequential model with two layers. A total of 477 parameters to train model. I was able to get a testing accuracy of 0.72688 for the first model, as shown below.

Table

Description automatically generated

Text

Description automatically generated with medium confidence

Optimization of the Model

Optimization 1:

Using the original Jupyter notebook I decided to put NAME column back in the model. I looked at NAME value type for binning. The count of the NAME indicated how many times an organization was funded. For all NAME less than 5 added to Other category. Name Column went from Length: 19568 to Length: 403.

Graphical user interface, text, application, email

Description automatically generated

My reasoning for including the NAME back in model was I thought that having received funding before would be a predictor of future funding. Adding NAME back into model gave Sequential model with two layers still and parameters went from 477 to 3,620. Testing accuracy increase from 0.72688 to 0.78636 as shown below.

**Table

Description automatically generated**

**Text

Description automatically generated**

**Optimization 2:**

I decided to bin based on ASK\_AMT and followed the same steps as above. Adding the ASK\_AMT bin into model, I also gave Sequential model with three layers still and parameters went from 477 to 3,627. The accuracy increased from 0.78636 to 0.78974.

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

Graphical user interface, text

Description automatically generated

**Summary**: Overall I was able to get an accuracy of 0.78974.using trial and error. This involved binning the following columns:

* APPLICATION\_TYPE
* CLASSIFICATION
* NAME
* ASK\_AMT

In summary, it can be concluded that it is recommended for a deep learning model to have multiple layers and large enough number of neuron, and the right activation function and balancing categorical input features to reach the accuracy for a binary classification models.