

## Neural Network Model Report - same as README

For this week's challenge, a deep learning model was created in order to predict the success of applicants for funding. The purpose of this analysis was to observe the accuracy rate of this model, as well as considering different parameters that may affect the variable. After cleaning the data, which included dropping unnecessary columns, replacing the APPLICATION\_TYPES, filtering and replacing CLASSIFICATION values, and converting categorical data to numeric values, we were able to split the preprocessed data into training and testing datasets once they were in features and target arrays. The variable considered as a feature would be the number of all columns in each row, whereas the target arrays would include the values in the 'IS\_SUCCESSFUL' column. For purposes of receiving the most accurate data, that same column was dropped from the input data for the features, meaning that the features included the number of columns for each row except for this specific one.

For the second part of this challenge, we were asked to compile, train, and evaluate the model using deep learning. Initially, there were two hidden layers, one with 40 units and the other with 20 units, both using the 'relu' activation function. An output layer was created afterwards with a total of one unit and a 'sigmoid' activation function. Once these layers were created, a summary was performed to determine the structure of the model. It is shown that the model is sequential. The model was then compiled and trained by using the ".compile" function, and then the ".fit" function, respectively. The value of one hundred epochs was performed initially, resulting in a 72.8% accuracy.

In an attempt to create a 75% accuracy rate, the number of hidden layers, units in each layer, as well as the epoch values, were manipulated. After several tries (and I emphasize several, ranging between 8-10 attempts), I failed to increase the accuracy rate. I added an extra hidden layer, and decreased the amount of units to each layer. For the first hidden layer, there were 30 units, 15 in the second layer, 5 in the third layer, and lastly, 1 in the output layer, with a total of 1,871 parameters. I trained the model once again, but for the amount of attempts that I took, the epochs ranged between 75-200. I ended up training the data with just 150 epochs, with an accuracy rate of 72.7%. Although I had lost my data in one of the attempts, the highest accuracy rate that I received was 73.3%, not much different than what was shown.

Although I was unable to meet the 75% accuracy rate goal, my data remained consistent in each of the steps that I took in an attempt to increase that rate. I believe that this, however,

would still be considered as a high accuracy rate, and therefore conclude that there is a 72%-73% success rate in applicants for funding. This may be manipulated further by changing variables such as dropping more or fewer columns, creating more bins, using different activation functions, adding more neurons to a layer, and possibly decreasing or increasing the amount of epochs. My recommendation would be to change the activation function of each layer, since each function takes different data and parameters into consideration. I believe that by doing this, there may be a higher chance of increasing the accuracy rate, and accomplishing that goal.