This repository is home to the code that was used to tackle the University of Pennsylvania's Data Science and Visualization Bootcamp's Neural Network Challenge. For this challenge we were tasked with reading in an Alphabet Soup CSV file to determine whether the funding of different charities was considered a success or not. Using knowledge of pandas and the tools we were given such as scikit learn and TensorFlow I was able to create a deep learning neural network to answer the question of whether it was a success.

When first running the data through the neural network model the accuracy of the data was around 72.34%. That was a result of running 2 hidden ReLu layers with 100 and 80 neurons. The model was then fit to run through 100 epochs to test the accuracy and loss of the model. A screenshot of a computer

Description automatically generated

After the initial run through of cleaning the data then creating a neural network model, I was tasked with optimizing the data so that I could reach a minimum accuracy of 75%. To optimize the data, I removed more columns that I did not feel were necessary for the answer, and then added a third hidden layer with tanh. After doing so I was able to increase the accuracy of the neural network model to 78.86%.

A screenshot of a computer

Description automatically generated

There are a few things that I would like to discuss to clarify the above conclusion. There are several variables that must be defined to better explain what the model was looking at. First, the accuracy of the neural network was based on a binary variable "IS\_SUCCESSFUL". This variable either returned a 1 if the charity was determined to be a success or a 0 if it was unsuccessful. Second, the variables that determined the features of the model were all the remaining columns in the data frame except for "IS\_SUCCESSFUL." Finally, not all the columns were necessary to use as the features of the model, therefore I removed them such as "EIN", "STATUS", "SPECIAL\_CONSIDERATIONS". All those columns gave indications towards the charities funding or labeling however, they were not so important when determining the success of the funding.

I was able to optimize the data to the required accuracy. To do so I needed to use 3 hidden layers (2 ReLu and 1 Tanh), along with varying neurons.

In conclusion the accuracy of the model could have been better even after optimizing it to the 75%. After attempts at fine tuning the data and changing the hidden layers, neurons, etc. there was still a lot of loss and inaccuracy. It would probably be a better choice to use something that could handle a variety of data types and outliers such as Random Forest Classifiers. TensorFlow does with the numeric values but there were a lot of other types of information in this data set that could have helped determine the success of the charity.