**Deep Learning Analysis**

**Questions for Analysis**

\* Data Preprocessing

\* What variable(s) are considered the target(s) for your model?

\* What variable(s) are considered to be the features for your model?

\* What variable(s) are neither targets nor features, and should be removed from the input data?

\* Compiling, Training, and Evaluating the Model

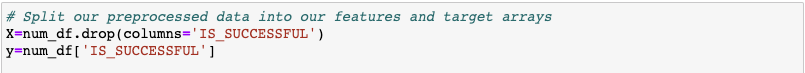
\* How many neurons, layers, and activation functions did you select for your neural network model, and why?

\* Were you able to achieve the target model performance?

\* What steps did you take to try and increase model performance?

**Findings**

The target variable I chose for my model was IS\_SUCCESSFUL because the goal was to try to find out if we could build a model that could successfully predict is Alphabet Soup funded a project if the applicant would use the money effectively. The features for my model were all the other columns besides the identifier columns (EIN and NAME) which could be dropped. I then created bins for the ‘APPLICATION\_TYPE’ values and the ‘CLASSIFICATION’ to limit the number of columns that would be created once I used get\_dummies.



Text

Description automatically generated with medium confidence

Table

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

I ended up building a model with 6 hidden layers that took 43 (all besides identifier) features. Each layer was as follows 1: (30 nodes, relu activation), 2: (16 nodes, relu activation), 3: (11 nodes, relu activation), 4: (16 nodes, relu activation), 5: (16 nodes, relu activation), 6: (16 nodes, relu activation), output layer activation was sigmoid.

Table

Description automatically generated

I was unable to achieve more than 75 percent accuracy on the test data with any of the models I tried. I ran a features selection to see if limiting the number of input features would have any effect and I ran a Keras Tuner to find the best hyperparameters for my deep learning model.

Graphical user interface, text, application, email

Description automatically generated

I do believe that I may be able to achieve a slightly higher accuracy rating, but nothing of much significance by increasing the number of hidden layers, increasing the number of nodes in each layer, and potentially by switching between different activations in different layers. I could also run more epochs to potentially increase the accuracy. Adjusting bin size to increase either the number of classes or application types may have also had an effect.