**Write Up:**

**1. Overview** of the Analysis: Explain the purpose of the analysis.

A: I wanted to create a binary classifier to predict whether applicants will be successful if funded by Alphabet Soup.

In order to be able to predict this, the approach is to create a machine learning model that can solve the task and be able to optimize the result through tuning the parameters.

**2. Results**: Using bulleted lists and images to support your answers, address the following questions.

* Data Preprocessing
  + What variable(s) are considered the target(s) for your model?

A: The “IS\_SUCCESSFUL” column is the target for the model. The variable for it is target\_y.

After splitting into training and testing data sets, for training, the variable for it is y\_train. For testing set, it is y\_test.

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

A: Most of the rest of the columns are considered features, the variable for them is features\_x.

After splitting into training and testing data sets, for training, the variable for it is x\_train. For testing set, it is x\_test.

*See screenshot of notebook below:*

Text

Description automatically generated

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

A: "EIN" and "NAME" columns were removed, because they are not features. They are not useful information for what we are trying to predict in this project.

*See screenshot of notebook below:*

Text

Description automatically generated with medium confidence

* 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?

A: Please see below for answers to the above 3 questions:

I’ve tested 3 combinations of different number of neurons, batch size and number of layers:

nn\_1 = build\_and\_test\_model(n\_hidden\_neurons=12, batch\_size=128, n\_layers=3)

nn\_2 = build\_and\_test\_model(n\_hidden\_neurons=4, batch\_size=64, n\_layers=3)

nn\_3 = build\_and\_test\_model(n\_hidden\_neurons=8, batch\_size=64, n\_layers=5)

The accuracy of nn\_1 is about: 0.732 (see below screenshot)



The accuracy of nn\_2 is about: 0.726 (see below screenshot)



The accuracy of nn\_3 is about: 0.730 (see below screenshot)



Based on the above results, I would select nn\_1 for the neural network model, since it has the highest accuracy among the three combinations/attempts.

I haven’t been able to achieve the target model performance (0.75) yet.

During the 3 attempts, I tried to increase the model performance by trying out different combinations of number of neurons, batch size and number of layers. So far, from the 3 combinations, we can see that “nn\_1”, which has a combo of 12 neurons, a batch size of 128, and 3 layers, gave us the best accuracy among them; but to achieve the target model performance of 0.75, I would keep testing different combinations of these parameters, and try to find a combo with optimized results.

1. **Summary**: Summarize the overall results of the deep learning model. Include a recommendation for how a different model could solve this classification problem, and explain your recommendation.

A: In summary, in order to be able to predict whether the loan applicant will be success, I’ve created a neural network classification model that so far has a 73.2% accuracy.

I’ve tested different combinations of the parameters, but I haven’t been able to achieve the 75% accuracy desired. As mentioned above, we can keep testing different combinations of the parameters of number of neurons, batch size and number of layers, or we can also look deeper into what other features might also determine “Is Successful”, and keep adjusting the model to come up with optimized results.

Another model that could also be used for this project would be the Random Forest Classification, because it is also good with the binary classification problem. Also, the performance of Random Forest Classification could be comparable with the neural network model when it only has few hidden layers. And also, since there are fewer parameters to work with, and overall more intuitive, Random Forest Classification could be a good alternative to the deep learning model for this project.