**Data Preprocessing**

* What variable(s) are the target(s) for your model?
  + IS\_SUCCESSFUL is the target variable since it is the outcome.
* What variable(s) are the features for your model?
  + Outside of IS\_SUCCESSFUL (target) and EIN and NAME (dropped) all other variables were used as features for the model. For the optimization method, STATUS was also not included.
* What variable(s) should be removed from the input data because they are neither targets nor features?
  + The status variable should be removed since it is neither and independent or dependent variable.

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

* How many neurons, layers, and activation functions did you select for your neural network model, and why?
  + There were three different methods used to try and optimize the model. The first method doubled the number of neurons. The second method added an additional hidden layer and increased the number of epochs (from 100 to 150). The third method utilized the keras auto-optimizer function.
* Were you able to achieve the target model performance?
  + No, the target model performance of 75% accuracy was not able to be achieved, even with the auto-optimizer function being utilized.
* What steps did you take in your attempts to increase model performance?
  + The first step I used was to increase the number of neurons. Compared to the original model, I doubled the amount of neurons for the two hidden layers. The second method that I used was to increase the number of epochs (from 100 to 150) and add an additional hidden layer. This slightly decreased the accuracy from 0.7263 to 0.7248. The last method I used was the auto-optimizer utilizing the keras auto-optimizer function. This improved the accuracy to 0.7275, but still did not achieve target model performance.

**Purpose of the Analysis**

* The purpose of this Analysis was to develop a model, utilizing previous data, that would be able to predict which applications got approved and which did not.

**Results**

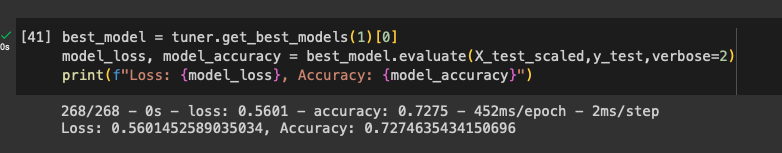
* Overall, none of the models were able to achieved the target performance. They were pretty close, but none were even able to cross the 73% mark. Utilizing optimization techniques helped to increase the accuracy of the model. The auto-optimization function from the keras package was able to create the most accurate model for the different methods that I tried.

**Different Model**

* You could potentially use a random forest plot to solve the same problem. This would generate comparisons that would tell you which type of candidate is more likely to get approved. You would be able to see comparisons at a much higher level, as well as a more granular level. This may also be beneficial because it could potentially show you which variables play a bigger role in predicting approval.

**Images**

Best Model:



Original Model:

A screenshot of a computer

Description automatically generated with low confidence