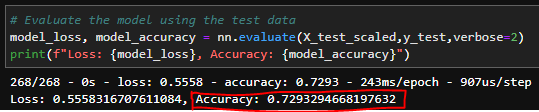
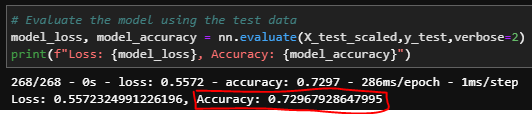
Overview:

The general purpose of this analysis was to create a neural network model, which could take in a set of categorical data and predict an outcome. More specifically, a set of application data for funding from Alphabet Soup where the model predicts success of the project.

Results:

* Data Preprocessing:
  + The target variable for this data set is the project’s success or failure given by the column “IS\_SUCCESSFUL”.
  + The rest of the columns in the data set (ones not removed below) are features in the model.
  + “EIN” and “NAME” were removed as they are identification columns and thus should have no impact on the outcome of the model.
* Compiling, Training, and Evaluating the Model:
  + First attempt:
    - Guided by starter code.
    - Accuracy = 0.729329
  + Second attempt:
    - Kept preprocessing the same.
    - Created method to optimize hyperparameters.
    - Accuracy = 0.733528
  + Third attempt:
    - Removed “STATUS”, 5 out of 34299 IDs had a value of 0 (0 & 1 possible values).
    - Removed “SPECIAL\_CONSIDERATIONS”, 27 out of 34299 IDs had a value of N (Y & N possible values).
    - Removed “ASK\_AMT”, 25398 out of 34299 IDs had a value of 5000 and the high end is in the billions.
    - Grouped “CLASSIFICATION”, previously grouped everything below 1000 instances together, but this attempt split the previous group at 75 instances. >1000 = no group, 76-1000 = “Other1”, <76 = “Other2”.
    - Used same hyperparameters as in the second attempt.
    - Accuracy = 0.730729

Summary:

The above neural networks performed fairly well with ~0.73 accuracy, however I would not let this model take over the awarding of funding. I would recommend attempting to improve the accuracy before trusting the model, but as is could be used as a tool to screen applications and order them by likelihood of success to review more efficiently. Alternatively, a random forest model could be generated based off this data given its categorical nature.