**Overview**: The purpose of this analysis was to develop a binary classifier to predict whether applicants will be successful if funded by Alphabet Soup. The dataset provided included various features about the applicants, and the goal was to use these features to classify applicants into those likely to be successful and those unlikely to succeed. The project involved preprocessing the data, building and optimizing a neural network, and evaluating the model's performance.

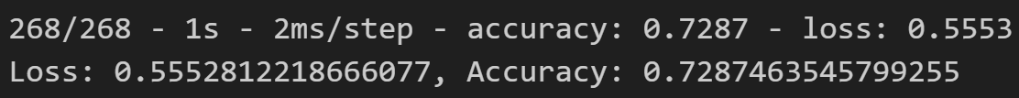
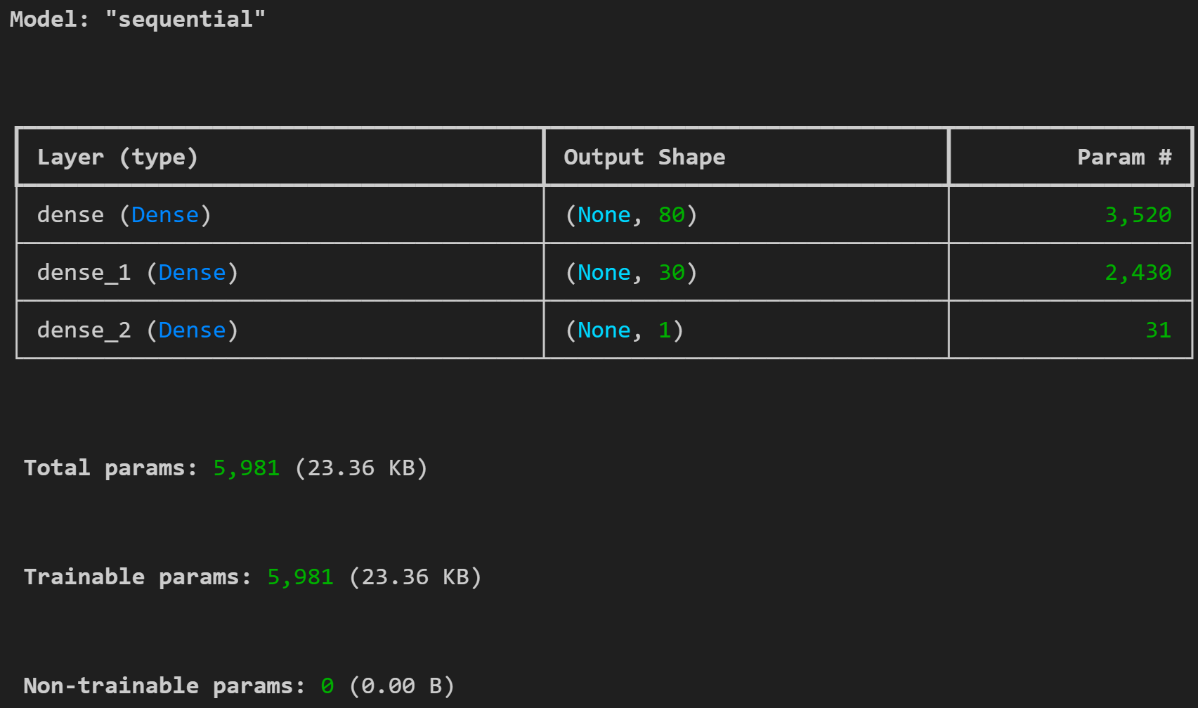
**Results**:

**Data Preprocessing**

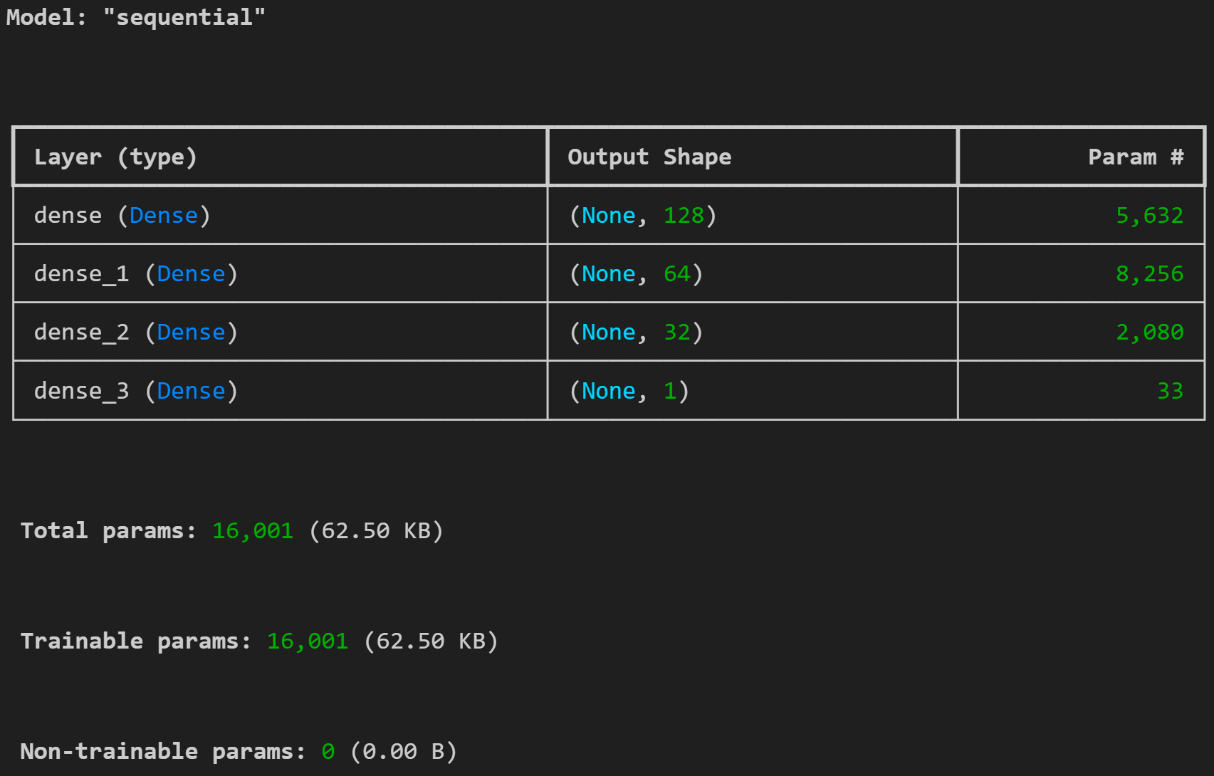
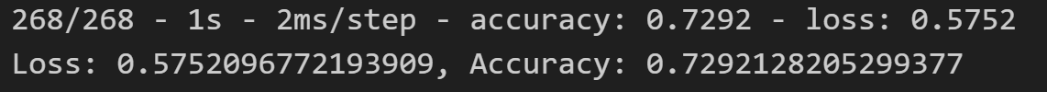
* The target variable for this model is whether an applicant is successful or not. It was encoded as a binary label where 1 = success and 0 = failure.
* The features used to predict success include various numerical and categorical data.
* The "EIN" and "Name" columns were dropped as they weren’t relevant for prediction.

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

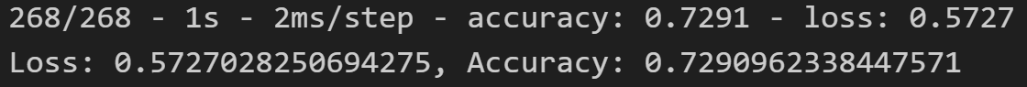
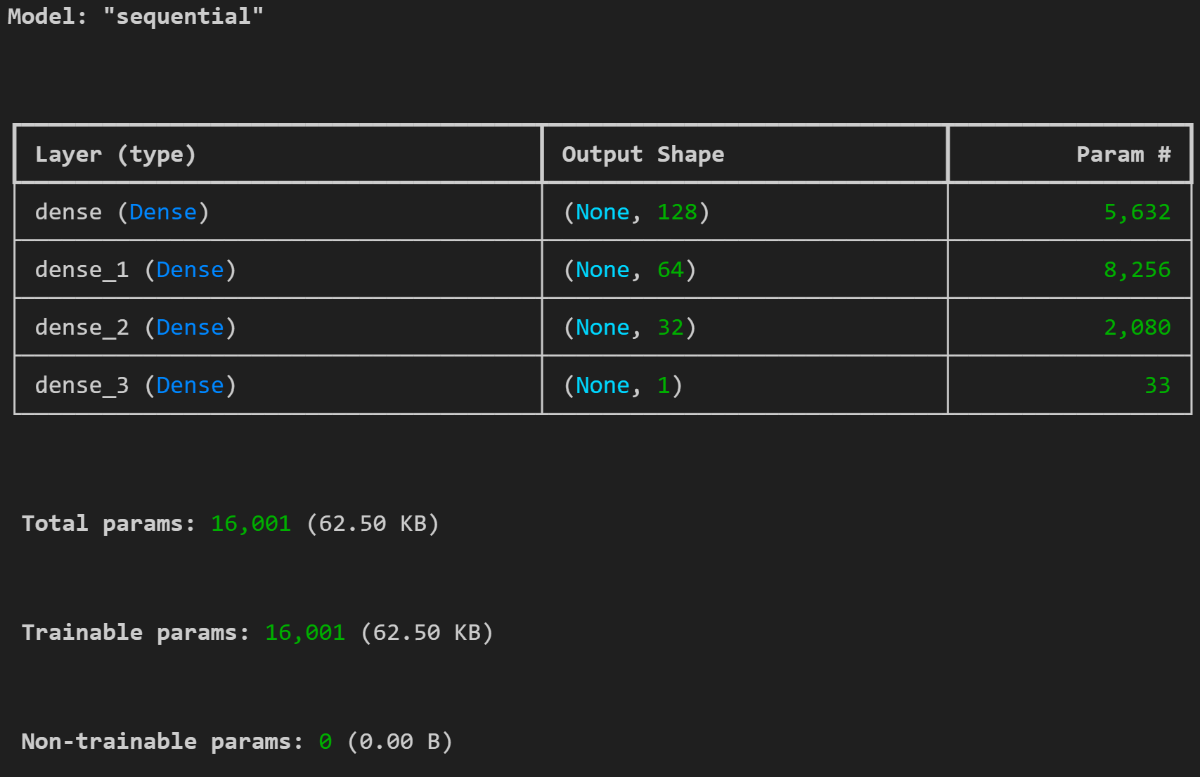
* For my initial model, I added 2 hidden layers and an output layer. I also set the epochs to 100. I chose this set-up because the initial starter code contained this information. It was also a good starting point for future optimization. Unfortunately, the target model performance was not achieved as the model only achieved 0.7287:



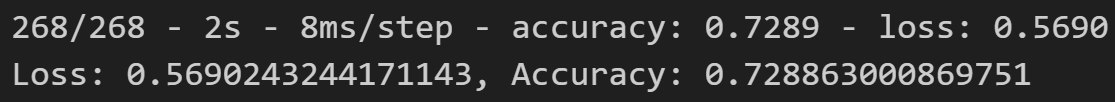
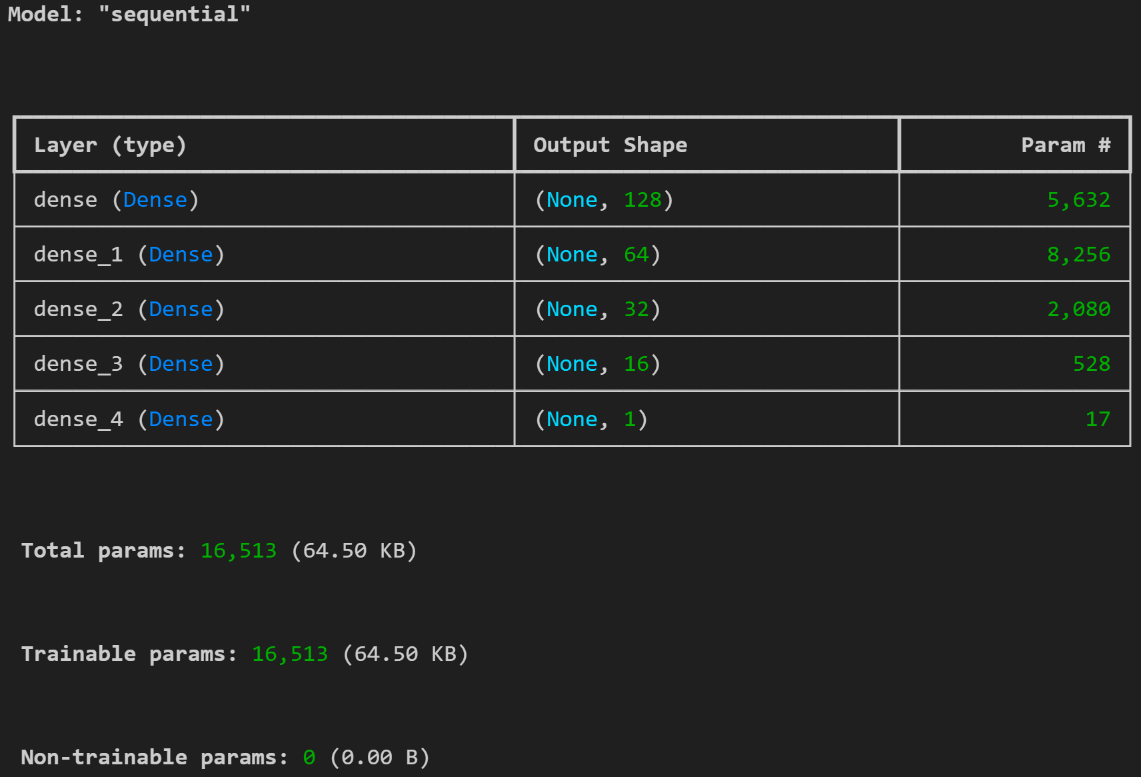
* I modified and re-ran my model 5 times using the following criteria to attempt to increase the model’s performance:
  + **Model 1:** I checked for missing data, dropped rows where STATUS = “1” and SPECIAL\_CONSIDERATIONS= “N”, then dropped these columns, and added a 3rd hidden layer. I kept the epochs set to 100. The target model performance was not achieved as the model only achieved 0.7292:



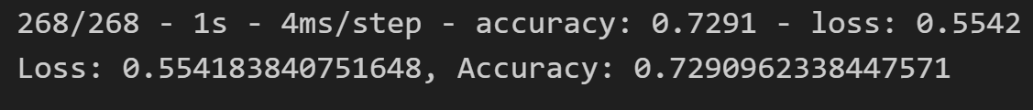
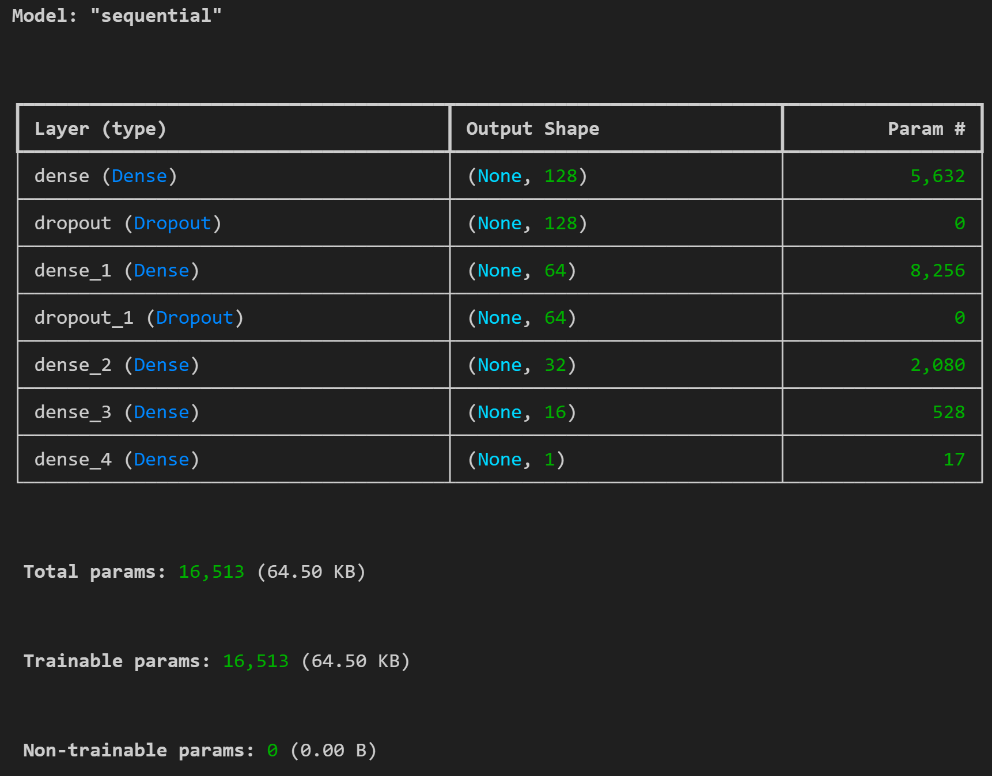
* **Model 2:** I kept everything the same as Model 1 except, changed Dense\_1 hidden layer activation from “relu” to “tanh” and increased the epochs to 150. The target model performance was not achieved as the model only achieved 0.7291:



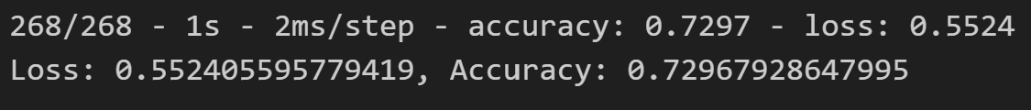
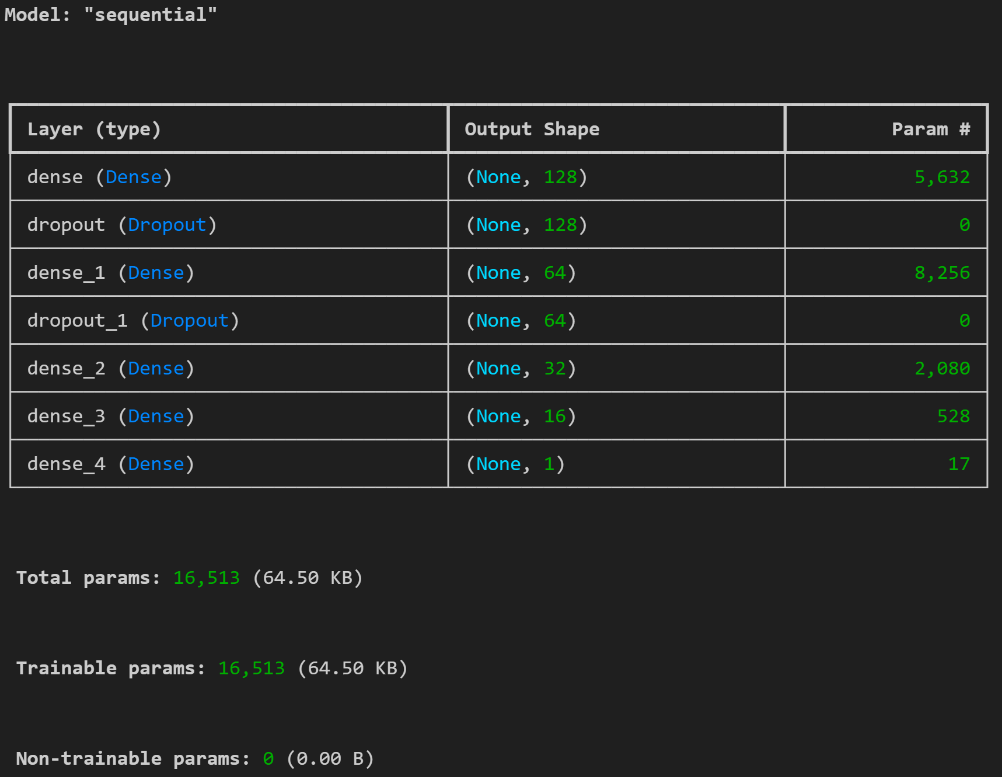
* **Model 3:** I kept everything the same as Model 2 except this time, I added a fourth layer where activation = “relu” and set the epochs to 200. The target model performance was not achieved as the model only achieved 0.7289:



* **Model 4:** I kept everything the same as Model 3, except I added 0.2 dropout to reduce overfitting after the first hidden layer and after the second. I then compiled the model and added a lower learning rate of 0.0005. Finally, I added a batch size of 32 before training the model. Epochs remained at 200. The target model performance was not achieved as the model only achieved 0.7291:



* **Model 5:** I kept everything the same as Model 4, except this time, I changed the fourth hidden layer’s activation from “relu” to “tanh” and the epochs to 250. The target model performance was not achieved as the model only achieved 0.7297:



* **Summary**: Summarize the overall results of the deep learning model. Include a recommendation for how a different model could solve this classification problem, and then explain your recommendation.
* The deep learning model successfully predicted the likelihood of success for applicants but did not meet the target of 75% accuracy. Despite multiple attempts at optimization, including adding more hidden layers, adjusting activations, and using techniques like dropout, the model's performance did not progress past an accuracy of ~73%.
* Recommendation: If I had more time, I would explore more advanced techniques like random forest or attempt to continue adjusting the hyperparameters (e.g. learning rate, epochs) which may enhance results.