

AlphabetSoupCharity_Optimization_Final

Explanation:

1. **Stratified Splitting:** The most important change is adding `stratify=y` to the `train_test_split` function:

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    random_state=78, stratify=y)
```

- **Reason:** Stratified splitting ensures that the proportions of the target variable (IS_SUCCESSFUL, which is 0 or 1) are the same in both the training and testing sets. This is essential for imbalanced datasets (where one class is much more frequent than the other). Without stratification, you might end up with a training set that has a very different distribution of successful/unsuccessful applications than the test set, leading to a model that performs well on the training data but poorly on the test data (overfitting) or a model that performs well on the test data that is not truly reflective of the accuracy of the model. This was the most likely cause of the lower-than-expected accuracy.
2. **Added More Hidden Layers and Nodes:** Added a *fourth* hidden layer and further increased the number of nodes in the hidden layers:

```
hidden_nodes_layer1 = 100 # Increased neurons
hidden_nodes_layer2 = 80  # Increased neurons
hidden_nodes_layer3 = 30  # Increased neurons
hidden_nodes_layer4 = 10
```

- **Reason:** A deeper and wider network (more layers and more neurons per layer) has more capacity to learn complex patterns. The original model might have been too simple to capture the nuances of the data. Increasing the model's capacity can lead to overfitting, but there are other techniques to mitigate that (like dropout, which could also have been added).