Charitable Project Funding Predictive Model Analysis

Introduction

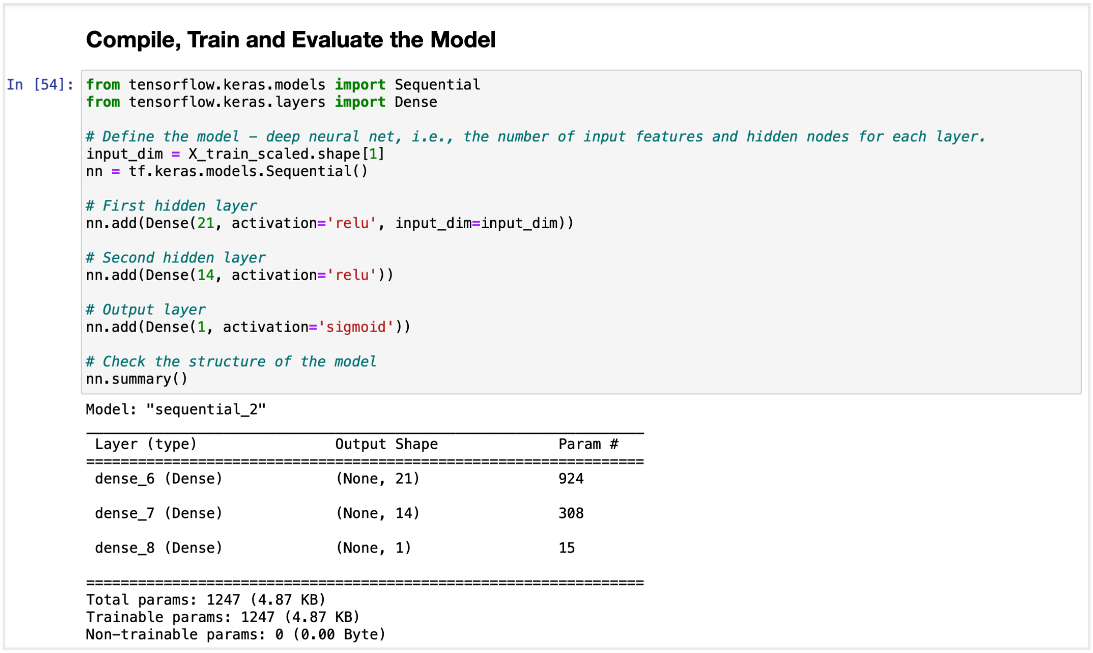
The primary objective of this analysis was to develop a binary classifier model capable of predicting the funding success of charitable projects. The process involved comprehensive data preprocessing to optimize the dataset for model training, followed by the compilation, training, and evaluation of the model's predictive accuracy. Subsequently, an optimization strategy focused on the "NAME" column was implemented to further enhance the model's performance.

Data Preprocessing

Several crucial operations were executed to prepare the dataset for effective model training. Irrelevant identification columns (EIN and NAME) were dropped, and application types and classification counts below a specified threshold were binned to manage column items effectively. Categorical data was converted to numeric binaries using dummy variables. The target variable for prediction, 'IS_SUCCESSFUL,' denoted the success or failure of project funding. Features used for prediction encompassed a range of variables, including "APPLICATION_TYPE," "AFFILIATION," "CLASSIFICATION," "USE_CASE," "ORGANIZATION," "STATUS," "INCOME_AMT," "SPECIAL_CONSIDERATIONS," and "ASK_AMT." Numerical data underwent scaling using the standardscaler function, and the pre-processed data was split into training and testing sets.

Model Compilation, Training, and Initial Evaluation

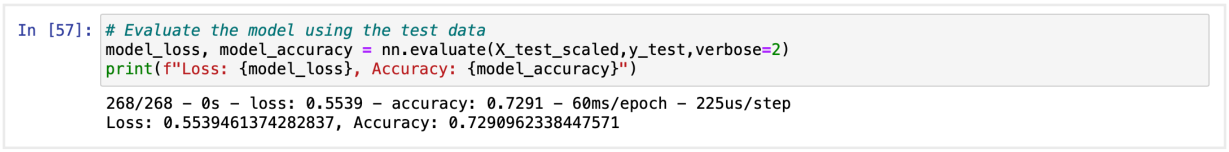
The model aimed for a benchmark accuracy of 75%, utilizing a sequential architecture with an initial input layer, one hidden layer, and an output layer. The training was performed on the scaled dataset and evaluated against the testing dataset. The initial model achieved a predictive accuracy of approximately 72%, slightly below the target benchmark.



Compile, Train and Evaluate the Model

In [54]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.
input_dim = X_train_scaled.shape[1]
nn = tf.keras.models.Sequential()

# First hidden layer
nn.add(Dense(21, activation='relu', input_dim=input_dim))

# Second hidden layer
nn.add(Dense(14, activation='relu'))

# Output layer
nn.add(Dense(1, activation='sigmoid'))

# Check the structure of the model
nn.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 dense_6 (Dense)             (None, 21)                924       
                                                                 
 dense_7 (Dense)             (None, 14)                308       
                                                                 
 dense_8 (Dense)             (None, 1)                 15        
                                                                 
=================================================================
Total params: 1247 (4.87 KB)
Trainable params: 1247 (4.87 KB)
Non-trainable params: 0 (0.00 Byte)
```

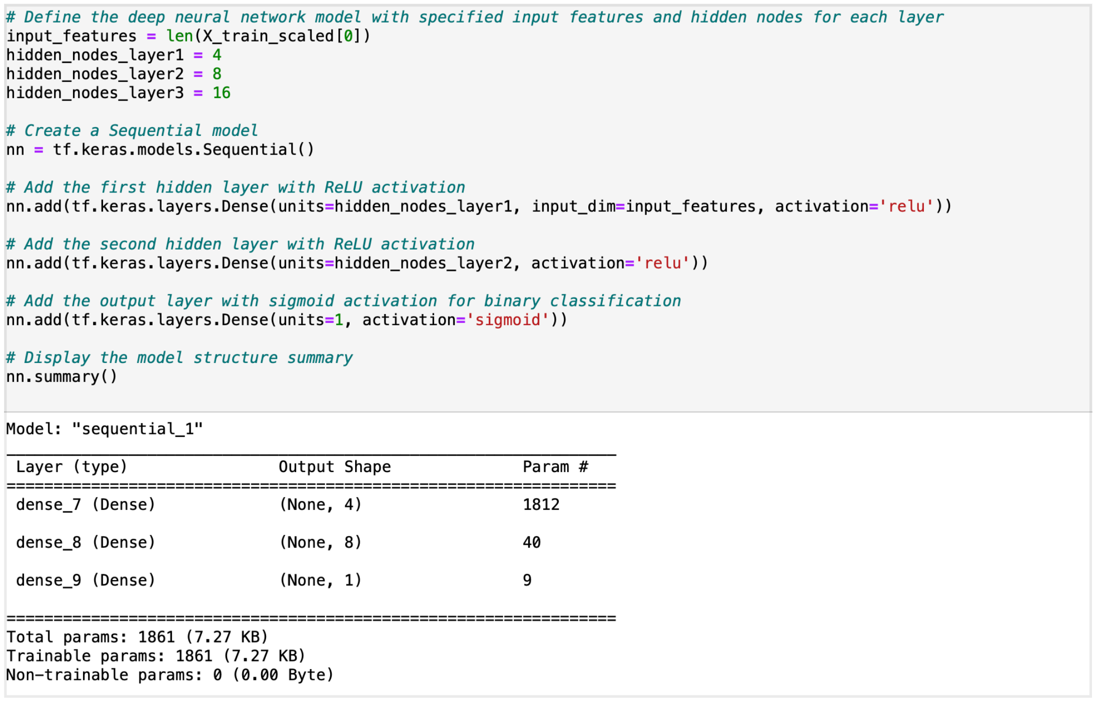


In [57]:
```
# Evaluate the model using the test data
model_loss, model_accuracy = nn.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")
```

```
268/268 - 0s - loss: 0.5539 - accuracy: 0.7291 - 60ms/epoch - 225us/step
Loss: 0.5539461374282837, Accuracy: 0.7290962338447571
```

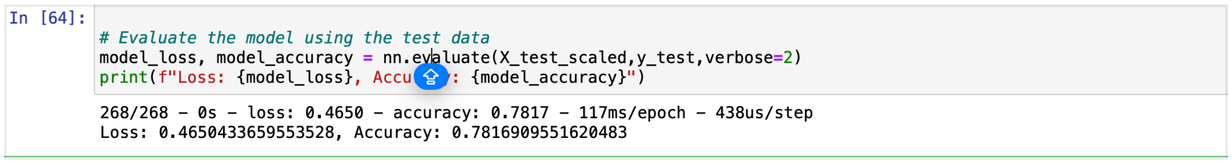
Model Optimization

To enhance model performance, a targeted optimization strategy was employed, focusing on the binning of the "NAME" column. This strategic categorization aimed to uncover meaningful patterns and associations, contributing to improved predictive accuracy.



```
# Define the deep neural network model with specified input features and hidden nodes for each layer
input_features = len(X_train_scaled[0])
hidden_nodes_layer1 = 4
hidden_nodes_layer2 = 8
hidden_nodes_layer3 = 16

# Create a Sequential model
nn = tf.keras.models.Sequential()

# Add the first hidden layer with ReLU activation
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer1, input_dim=input_features, activation='relu'))

# Add the second hidden layer with ReLU activation
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer2, activation='relu'))

# Add the output layer with sigmoid activation for binary classification
nn.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))

# Display the model structure summary
nn.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_7 (Dense)             (None, 4)                 1812

 dense_8 (Dense)             (None, 8)                 40

 dense_9 (Dense)             (None, 1)                 9

=================================================================
Total params: 1861 (7.27 KB)
Trainable params: 1861 (7.27 KB)
Non-trainable params: 0 (0.00 Byte)
```



```
In [64]:
# Evaluate the model using the test data
model_loss, model_accuracy = nn.evaluate(X_test_scaled,y_test,verbose=2)
print(f"Loss: {model_loss}, Accu[illegible]y: {model_accuracy}")
```

```
268/268 - 0s - loss: 0.4650 - accuracy: 0.7817 - 117ms/epoch - 438us/step
Loss: 0.46504335659553528, Accuracy: 0.7816909551620483
```

Outcome

The optimization process yielded a substantial improvement, with the model's accuracy now standing at an impressive 78%. This signifies a notable advancement over the initial accuracy of approximately 72%. The strategic binning of the "NAME" column played a pivotal role in extracting relevant information, enabling the model to make more informed predictions regarding the success of charitable project funding.

Results

- **What was the initial predictive accuracy of the model?**
  - The initial model achieved a predictive accuracy of approximately 72%.
- **What was the benchmark accuracy set for the model?**
  - The benchmark accuracy set for the model was 75%.
- **What was the purpose of optimizing the model, and what strategy was employed?**
  - The purpose of optimization was to enhance the model's performance. The optimization strategy focused on the binning of the "NAME" column to derive more meaningful patterns.
- **What was the outcome of the optimization process?**

    - The model achieved a notable advancement, with the accuracy reaching an impressive 78%, a substantial improvement over the initial accuracy.
- **How did the strategic binning of the "NAME" column contribute to the model's improvement?**
    - Binning the "NAME" column strategically played a pivotal role in extracting relevant information, allowing the model to make more informed predictions regarding the success of charitable project funding.
- **What further refinements can be explored to enhance model performance?**
    - Further refinements may be explored in feature engineering and model fine-tuning to continue improving predictive accuracy.
    -

Overall Summary

In summary, the analysis successfully developed a binary classifier model for predicting charitable project funding success. The optimization strategy, particularly focusing on the "NAME" column, significantly improved the model's accuracy to 78%. Ongoing exploration of refinements is recommended to continually enhance the model's predictive capabilities

Alternative Model Consideration

Considering the nature of the problem, an ensemble model, such as a Random Forest or Gradient Boosting, could be explored. Ensemble models often perform well in scenarios where multiple factors influence the outcome, as is the case with charitable project funding. Their ability to capture complex relationships and interactions among features could provide a more nuanced understanding, potentially leading to further improvements in predictive accuracy.