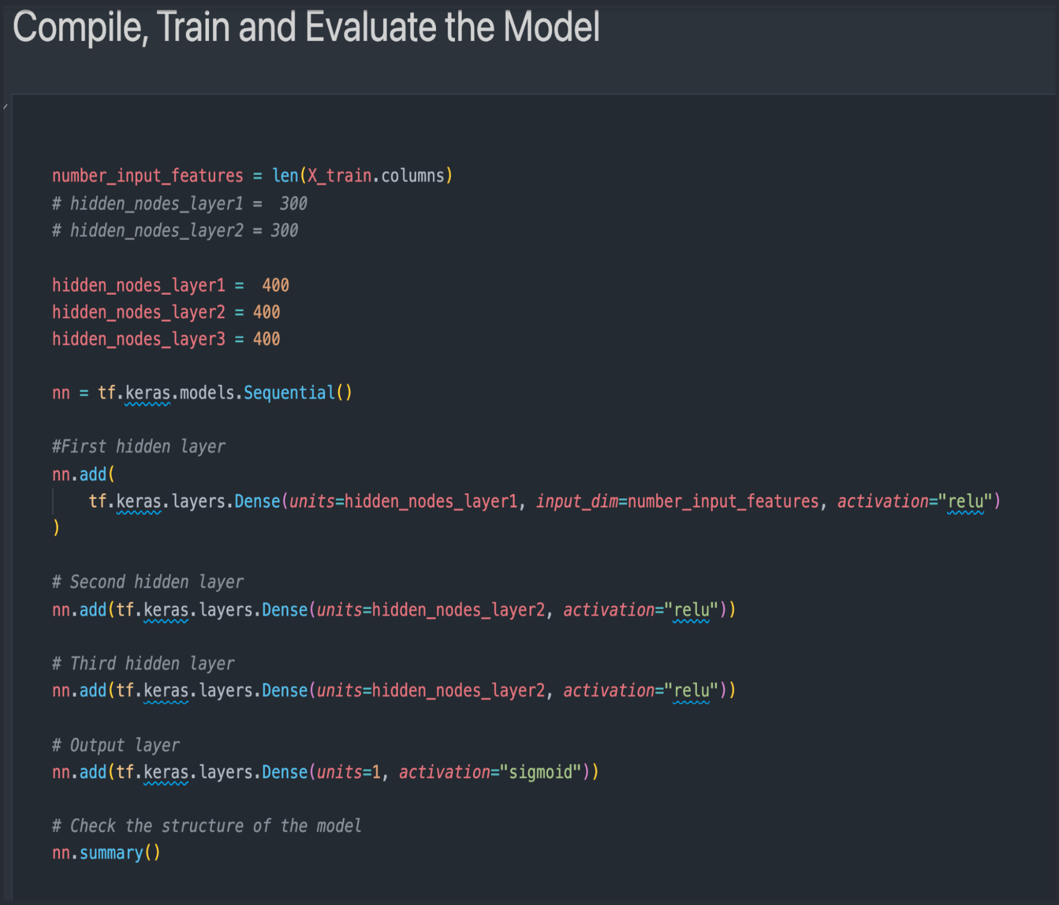
Analysis

The dataset was used to create an algorithm to predict whether or not applicants would be successfully funded and contained more than 34,000 organizations that have received funding from Alphabet Soup. Within the dataset, there are ten columns containing metadata about the organizations.

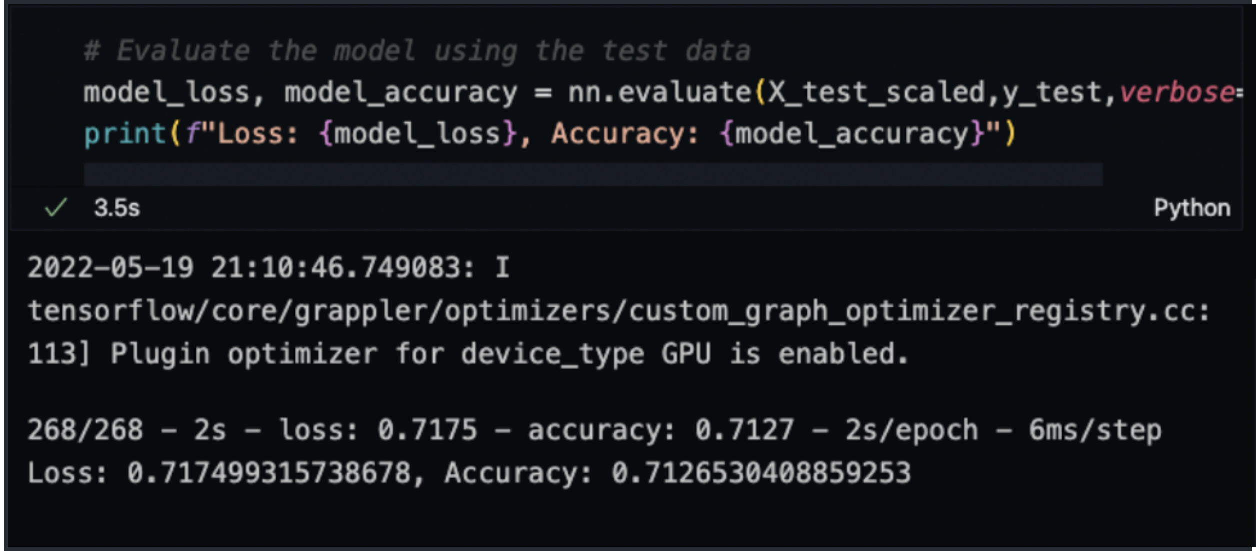
During the data processing, irrelevant information was removed or replaced. For example, the EIN and Name were dropped from the model due to high fluctuation, and classification and application types were replaced with "Other." Finally, categorical values were encoded by pd.get_dummies.



Compile, Train and Evaluate the Model

```
number_input_features = len(X_train.columns)
# hidden_nodes_layer1 =  300
# hidden_nodes_layer2 = 300

hidden_nodes_layer1 =  400
hidden_nodes_layer2 = 400
hidden_nodes_layer3 = 400

nn = tf.keras.models.Sequential()

#First hidden layer
nn.add(
    tf.keras.layers.Dense(units=hidden_nodes_layer1, input_dim=number_input_features, activation="relu")
)

# Second hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer2, activation="relu"))

# Third hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer2, activation="relu"))

# Output layer
nn.add(tf.keras.layers.Dense(units=1, activation="sigmoid"))

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

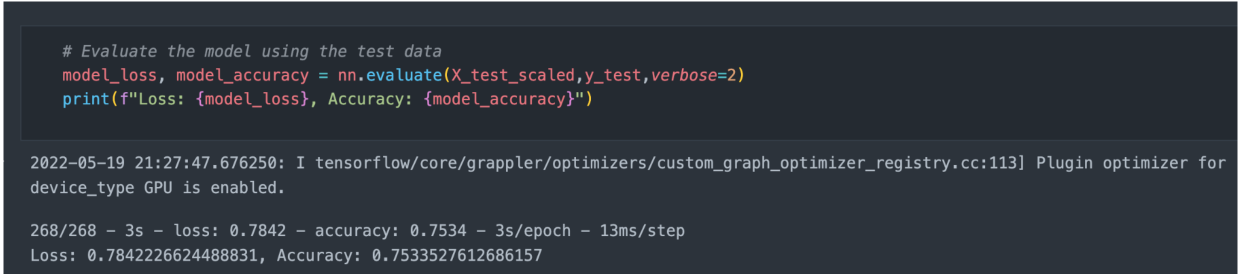
The neural network was applied to each model. The number of features dictated the number of hidden nodes. Two hidden nodes at 300 and 50 epochs were used for the first attempt. The first attempt reached 71% accuracy, so the model needed to be compiled a second time in order to reach 75% accuracy.



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

```
2022-05-19 21:10:46.749083: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:
113] Plugin optimizer for device_type GPU is enabled.

268/268 - 2s - loss: 0.7175 - accuracy: 0.7127 - 2s/epoch - 6ms/step
Loss: 0.717499315738678, Accuracy: 0.7126530408859253
```

The second attempt featured three nodes at 400 and fifty epochs. After training the model, the accuracy met the 75% required accuracy.



```
# 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}")
```

```
2022-05-19 21:27:47.676250: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:113] Plugin optimizer for
device_type GPU is enabled.

268/268 - 3s - loss: 0.7842 - accuracy: 0.7534 - 3s/epoch - 13ms/step
Loss: 0.7842226624488831, Accuracy: 0.7533527612686157
```