Write a Report on the Neural Network Model

The report should contain the following:

1. **Overview** of the analysis: Explain the purpose of this analysis.

2. **Results**: Using bulleted lists and images to support your answers, address the following questions:

- Data Preprocessing
  - What variable(s) are the target(s) for your model?
  - What variable(s) are the features for your model?
  - What variable(s) should be removed from the input data because they are neither targets nor features?
- Compiling, Training, and Evaluating the Model
  - How many neurons, layers, and activation functions did you select for your neural network model, and why?
  - Were you able to achieve the target model performance?
  - What steps did you take in your attempts to increase model performance?

3. **Summary**: Summarise 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.

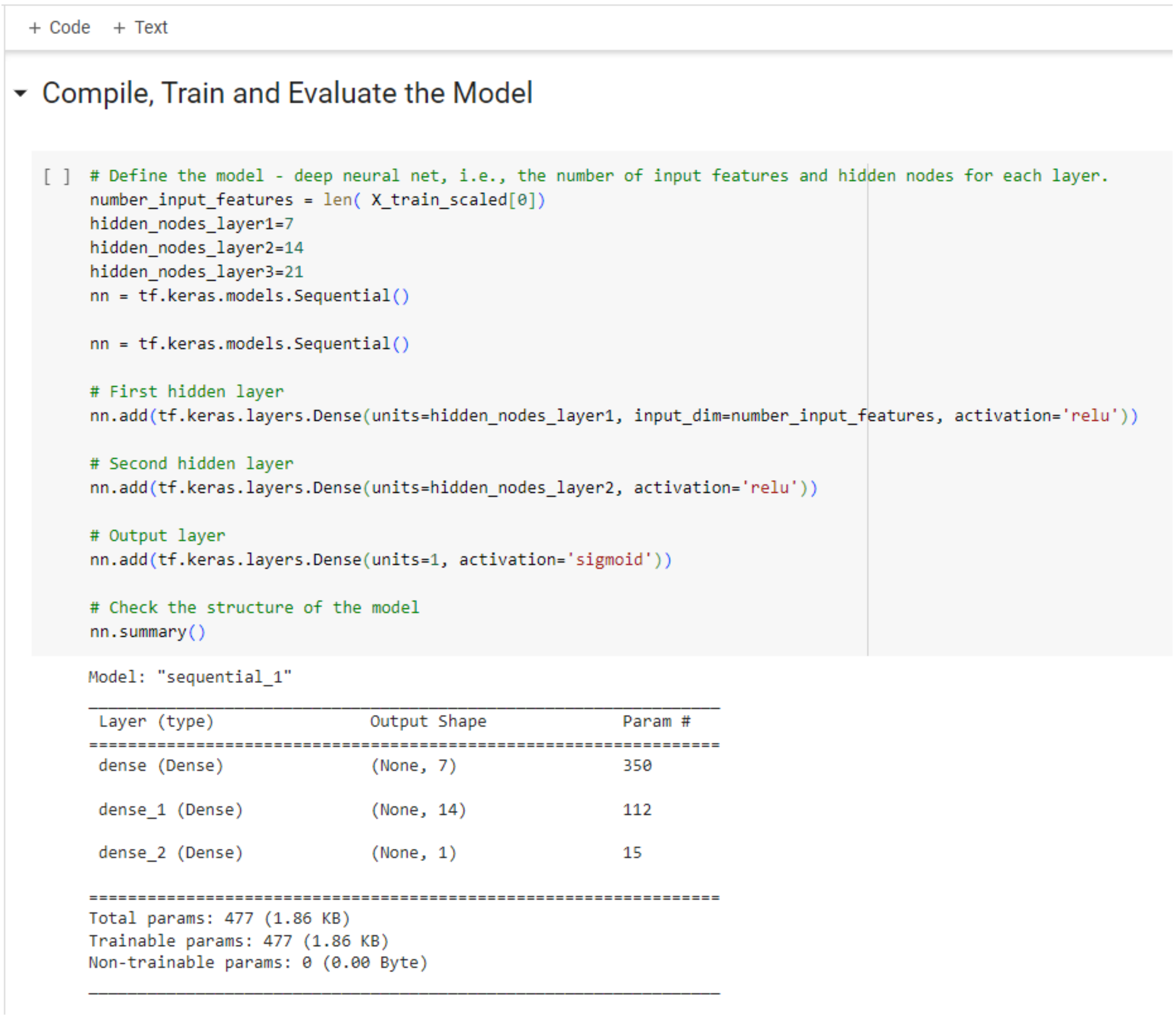
Deep learning and neural networks were used to determine if applicants would be successfully funded by Alphabet Soup.

Data Processing

The dataset underwent preprocessing to eliminate irrelevant information, resulting in the removal of EIN and NAME from the model. The remaining columns were designated as features. In a subsequent test, NAME was reintroduced. Due to high fluctuation, CLASSIFICATION and APPLICATION_TYPE were replaced with 'Other.' The data was then divided into training and testing sets. The target variable for the model is "IS_SUCCESSFUL," with 1 representing yes and 0 representing no. An analysis of APPLICATION data was conducted, and CLASSIFICATION values were used for binning. Unique values served as cutoff points to group "rare" categorical variables under a new label, 'Other.' Subsequently, the success of binning was assessed. Categorical variables were encoded using 'pd.get_dummies().'

Compiling, Training, and Evaluation the Model

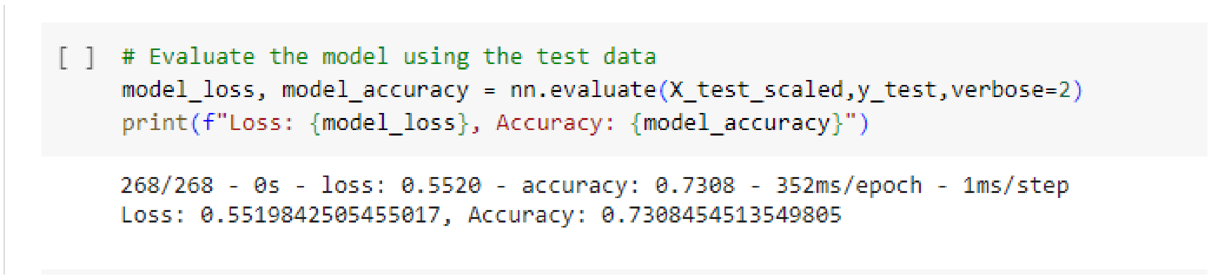
Each model underwent the application of a Neural Network with a total of three layers. The number of hidden nodes in each layer was determined by the number of features.



Compile, Train and Evaluate the Model

```
[ ] # Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.
    number_input_features = len( X_train_scaled[0])
    hidden_nodes_layer1=7
    hidden_nodes_layer2=14
    hidden_nodes_layer3=21
    nn = tf.keras.models.Sequential()

    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'))

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

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

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 7)                 350

 dense_1 (Dense)             (None, 14)                112

 dense_2 (Dense)             (None, 1)                 15

=================================================================
Total params: 477 (1.86 KB)
Trainable params: 477 (1.86 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

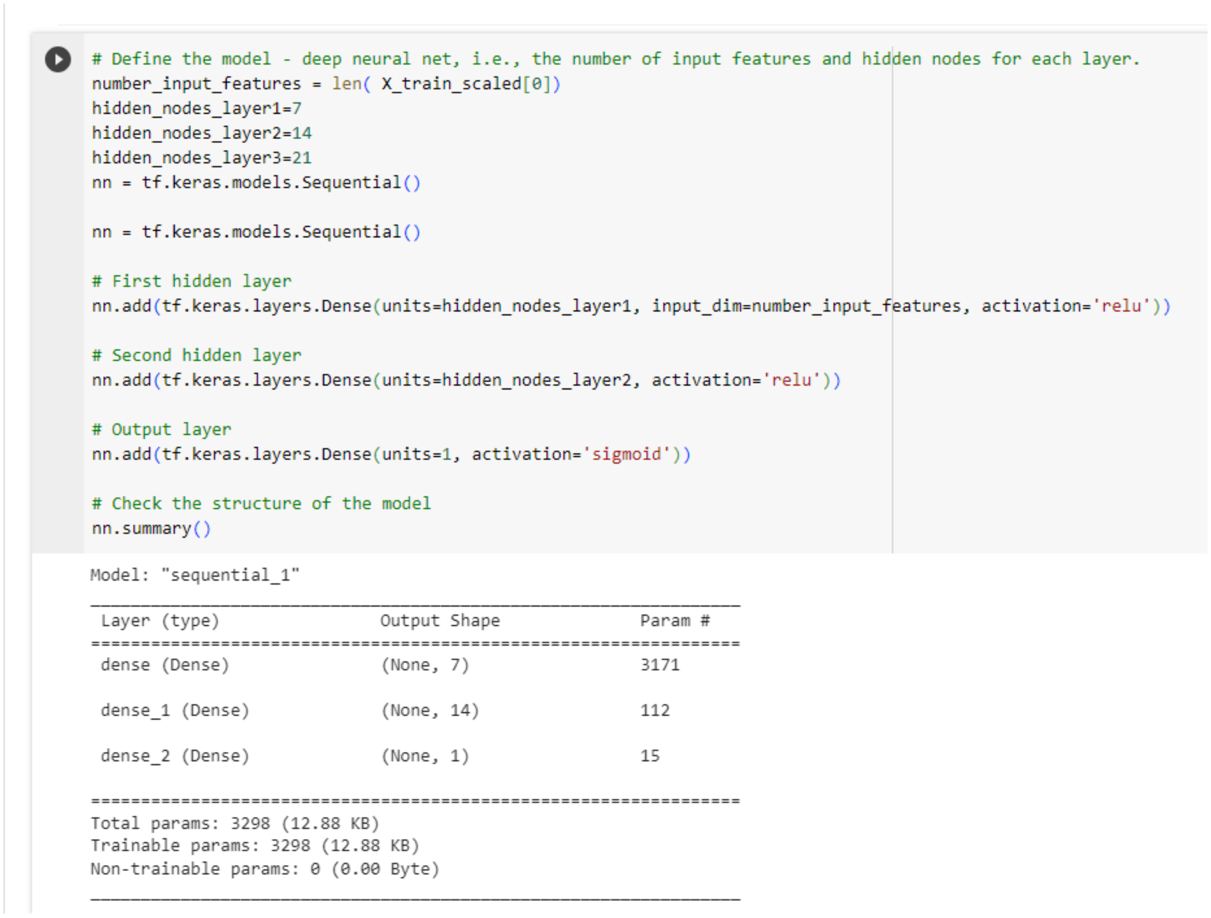


```
[ ] # Evaluate the model using the test data
    model_loss, model_accuracy = nn.evaluate(X_test_scaled,y_test,verbose=2)
    print(f"Loss: {model_loss}, Accuracy: {model_accuracy}")

    268/268 - 0s - loss: 0.5520 - accuracy: 0.7308 - 352ms/epoch - 1ms/step
    Loss: 0.5519842505455017, Accuracy: 0.7308454513549805
```

The three-layer training model produced 477 parameters. In the initial try, the accuracy slightly exceeded 73%, falling short of the targeted 75%.

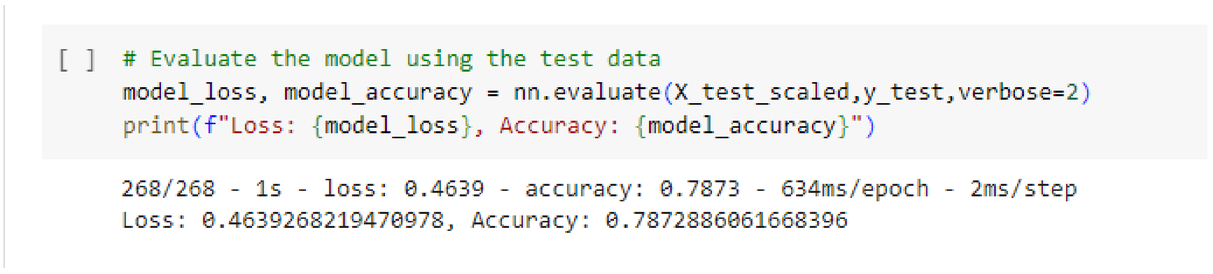
Optimisation

In the second iteration, 'NAME' was reintroduced into the dataset. This time, the model achieved an accuracy just below 79%, surpassing the target by slightly less than 4%. The model had a total of 3,298 parameters.



```
# Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.
number_input_features = len( X_train_scaled[0])
hidden_nodes_layer1=7
hidden_nodes_layer2=14
hidden_nodes_layer3=21
nn = tf.keras.models.Sequential()

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'))

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

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

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 7)                 3171

 dense_1 (Dense)             (None, 14)                112

 dense_2 (Dense)             (None, 1)                 15

=================================================================
Total params: 3298 (12.88 KB)
Trainable params: 3298 (12.88 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```



```
[ ]  # 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 - 1s - loss: 0.4639 - accuracy: 0.7873 - 634ms/epoch - 2ms/step
     Loss: 0.4639268219470978, Accuracy: 0.7872886061668396
```

Deep learning models are designed with multiple layers to enable a machine-based process. These models instruct computers to filter inputs through various layers, facilitating the learning of patterns and enabling accurate predictions and classifications of information.