CHARITY FUNDING PREDICTOR

OVERVIEW

Using deep learning and neural networks, a dataset of over 34,000 organizations was analyzed to determine the success of the applicants if funded by Alphabet Soup.

Results

Data Preprocessing

- What variable(s) are considered the target(s) for your model?
  - The target variable in this model was “IS_SUCCESFUL”
- What variable(s) are considered to be the features for your model?
  - Featured variables in the model were the remaining columns, after the EIN and NAME columns were removed (NAME added back in during the optimization of the model)
- What variable(s) are neither targets nor features, and should be removed from the input data?
  - The variable EIN held no significance to the model’s performance

Compiling, Training, and Evaluating the Model

- How many neurons, layers, and activation functions did you select for your neural network model, and why?
  - The model had 3 layers, the features determine the number of nodes. Sigmoid and Relu activation functions were selected.

```
# 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()
```

- Were you able to achieve the target model performance?
  - Yes I was able to achieve a 79% model 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}")
```

```
268/268 - 0s - loss: 0.4624 - accuracy: 0.7918 - 337ms/epoch - 1ms/step
Loss: 0.46235769987106323, Accuracy: 0.7918367385864258
```

- What steps did you take to try and increase model performance?
  - To optimize performance, “NAME” was added back into the model. This increased the model from 72% to 79%.

SUMMARY

The model produced an overall 79% accuracy score. The model could be optimized by adding more layers to better predict and classify data.