Analysis

Overview

The purpose of this analysis is to determine if applicants would be successfully funded by the organization Alphabet shop which in the past has funded more than 34,000 organizations.

- What variable(s) are the target(s) for your model?

The target variable for the model is “IS_SUCCESSFUL” and is verified by the value, 1 was considered yes and 0 was no.

- What variable(s) are the features for your model?

  APPLICATION_TYPE, AFFILIATION, CLASSIFICATION, USE_CASE, ORGANIZATION, STATUS, INCOME_AMT, SPECIAL_CONSIDERATIONS, ASK_AMT

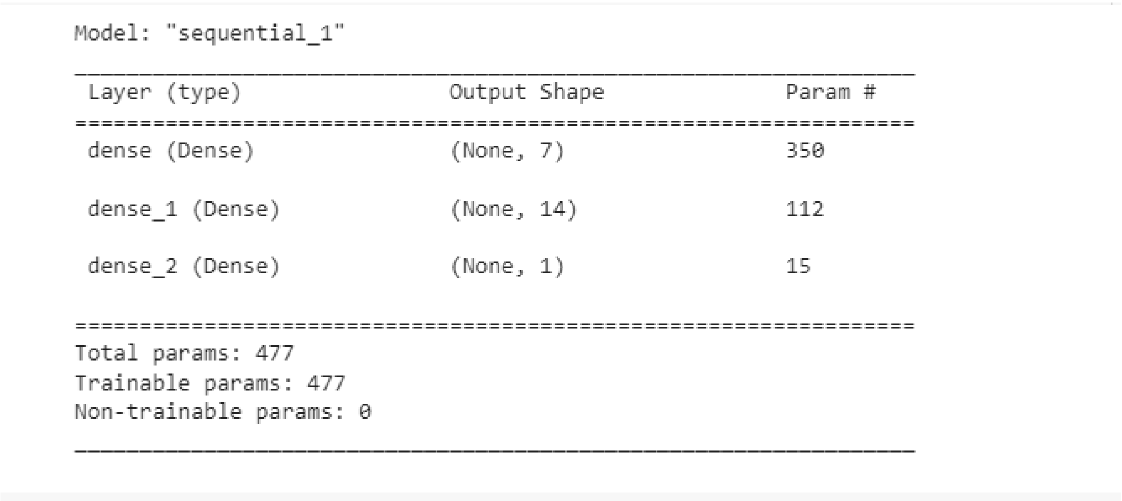
- What variable(s) should be removed from the input data because they are neither targets nor features?

  We removed any irrelevant information, columns such as EIN and Name were dropped from the dataset.

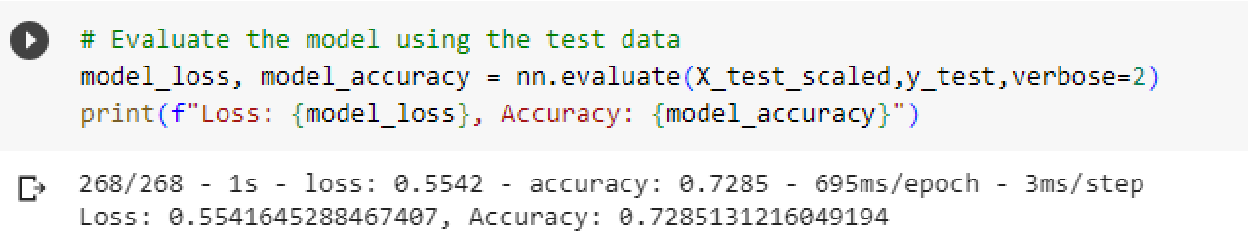
Compiling, Training, and Evaluating the Model

- How many neurons, layers, and activation functions did you select for your neural network model, and why?

A Neural Network was applied on each model multiple layers, three in total. The number of features dictated the number of hidden nodes. The first attempt resulted in a score of 72% which is below our desired score of 75%.



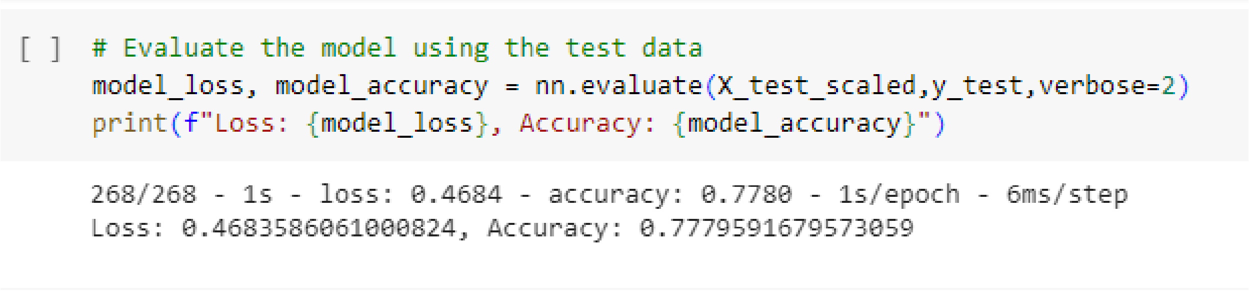
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 7)                 350

 dense_1 (Dense)             (None, 14)                112

 dense_2 (Dense)             (None, 1)                 15

=================================================================
Total params: 477
Trainable params: 477
Non-trainable params: 0
_________________________________________________________________
```



```
# 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}")
```

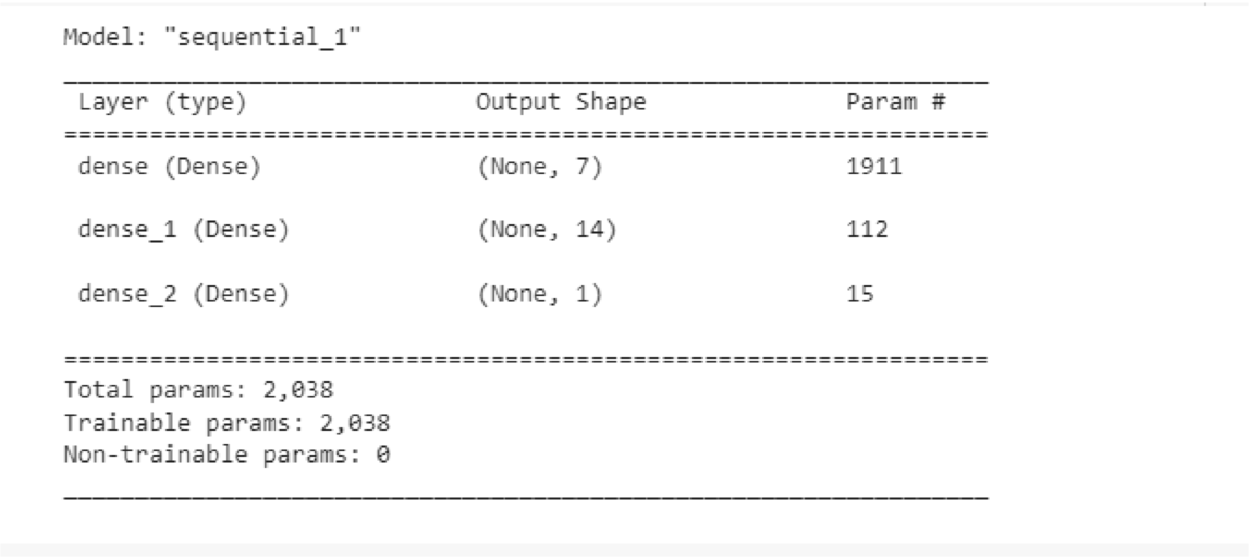
```
268/268 - 1s - loss: 0.5542 - accuracy: 0.7285 - 695ms/epoch - 3ms/step
Loss: 0.5541645288467407, Accuracy: 0.7285131216049194
```

In the second attempt, we added the NAME column back into the dataset, this resulted in a score of 79%, which is above our desired score of 75%. The total params went from 471 to 2038.



```
[ ] # 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: 0.4684 - accuracy: 0.7780 - 1s/epoch - 6ms/step
Loss: 0.4683586061000824, Accuracy: 0.7779591679573059
```



```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 7)                 1911

 dense_1 (Dense)             (None, 14)                112

 dense_2 (Dense)             (None, 1)                 15

=================================================================
Total params: 2,038
Trainable params: 2,038
Non-trainable params: 0
_________________________________________________________________
```

Summary.

In summary we can conclude that by changing the parameters such as adding the “NAME” column helped increase accuracy of the model. A supervised machine learning may be a better way to classify the groups and result. Due to the high number of input parameters, a random forest classifier may result in more accurate and reliable results. In the future, performing a deep learning algorithm using random forest classifier would provide improved accuracy.