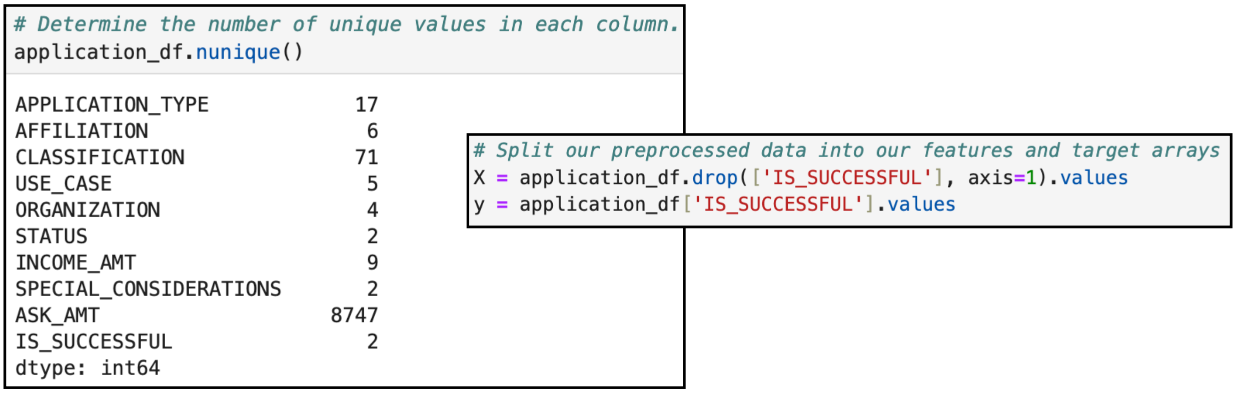
Neural Network Report

Overview

This model takes information from Alphabet Soup's previous business ventures to create a neural network model to predict if a business venture will be successful. The model take in the application type, affiliation, classification, funding use, organization type, status, income classification, special considerations and requests funding amount and outputs if it is successful.

Results

- Data Processing
  - The target of the model is if the venture is successful
  - The variables of the model are application type, affiliation, classification, funding use, organization type, status, income classification, special considerations.

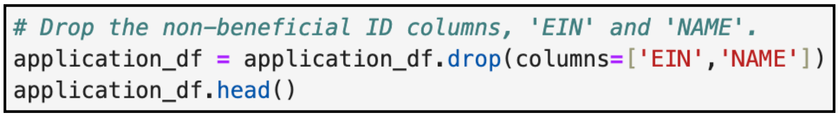


```
# Determine the number of unique values in each column.
application_df.nunique()

APPLICATION_TYPE            17
AFFILIATION                  6
CLASSIFICATION              71
USE_CASE                     5
ORGANIZATION                 4
STATUS                       2
INCOME_AMT                   9
SPECIAL_CONSIDERATIONS       2
ASK_AMT                   8747
IS_SUCCESSFUL                2
dtype: int64
```

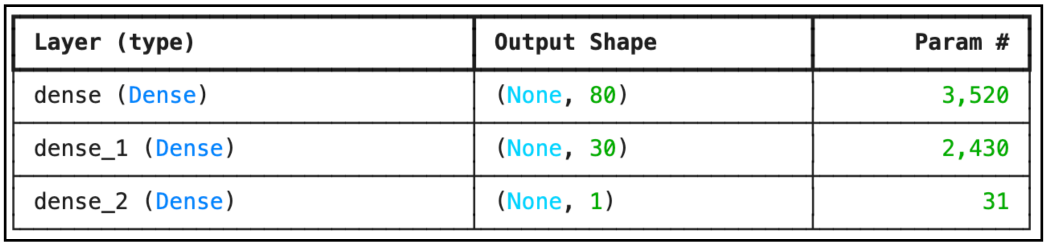
```
# Split our preprocessed data into our features and target arrays
X = application_df.drop(['IS_SUCCESSFUL'], axis=1).values
y = application_df['IS_SUCCESSFUL'].values
```

  - The EIN and name of the organizations were removed since they are identifier and the do not contribute to the success of the venture.

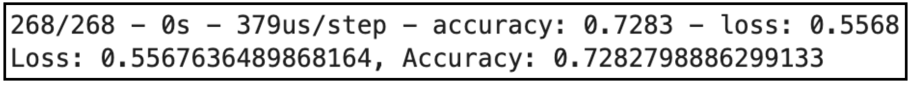


```
# Drop the non-beneficial ID columns, 'EIN' and 'NAME'.
application_df = application_df.drop(columns=['EIN','NAME'])
application_df.head()
```

- Compiling, Training, and Evaluating the Model
  - ALL MODELS: Each attempt has the same amount of input dimensions to keep all the necessary information for each model. The first hidden layer uses relu activation and sigmoid on the output layer. The models are all trained with *epochs=100*.
  - Original model: The original model has two hidden layers with relu activation with 80 and 30 neurons, respectively. This model has 72.8% accuracy.

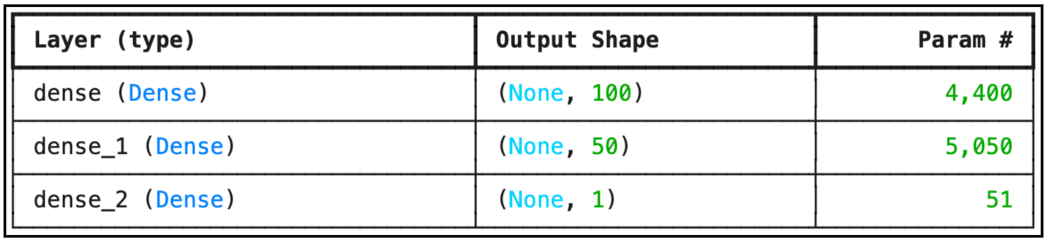


Layer (type)	Output Shape	Param #
dense (Dense)	(None, 80)	3,520
dense_1 (Dense)	(None, 30)	2,430
dense_2 (Dense)	(None, 1)	31

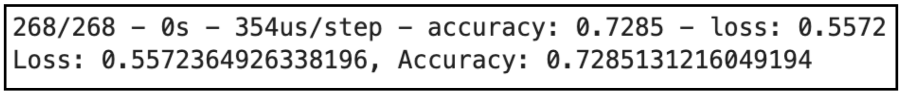


```
268/268 - 0s - 379us/step - accuracy: 0.7283 - loss: 0.5568
Loss: 0.5567636489868164, Accuracy: 0.7282798886299133
```

- Attempt #1: The hidden layer neurons on this model are 100 and 50, respectively. I wanted to increase the neurons to see it that would better handle the large data set, but this did not change the accuracy. This model resulted in 72.9% accuracy.

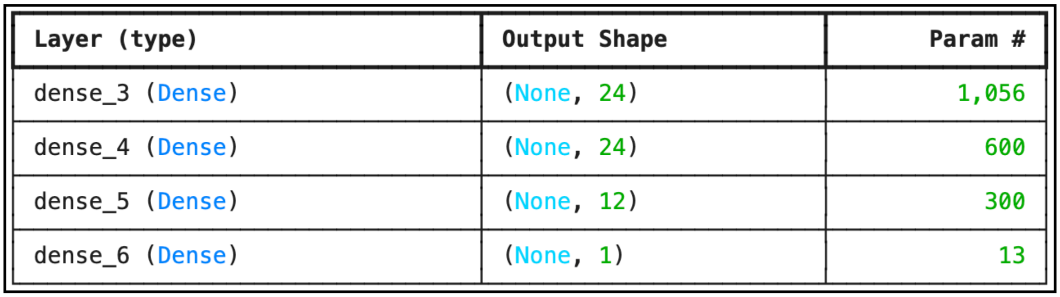


Layer (type)	Output Shape	Param #
dense (Dense)	(None, 100)	4,400
dense_1 (Dense)	(None, 50)	5,050
dense_2 (Dense)	(None, 1)	51

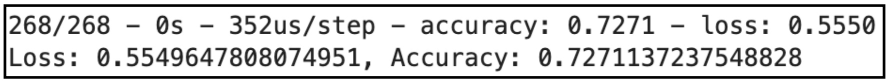


```
268/268 - 0s - 354us/step - accuracy: 0.7285 - loss: 0.5572
Loss: 0.5572364926338196, Accuracy: 0.7285131216049194
```

- Attempt #2: Since increasing the neurons did not improve accuracy, this model had a decrease in units. There is also an additional layer in this model using relu activation to process the data. The first two layers had 24 neurons and the third had 12. However this also had minimal change in the accuracy with 72.7%.

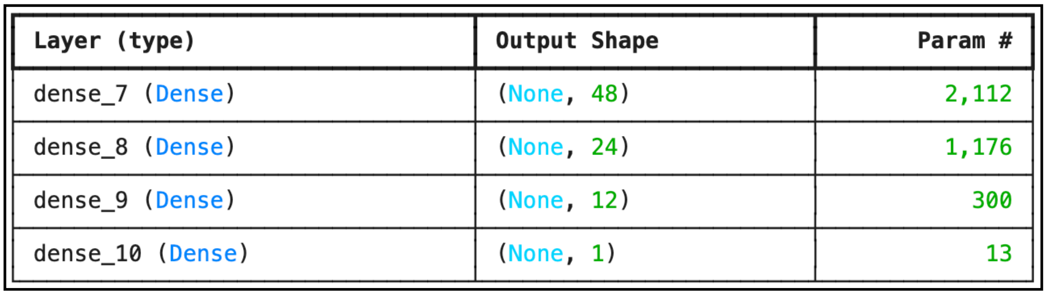


Layer (type)	Output Shape	Param #
dense_3 (Dense)	(None, 24)	1,056
dense_4 (Dense)	(None, 24)	600
dense_5 (Dense)	(None, 12)	300
dense_6 (Dense)	(None, 1)	13

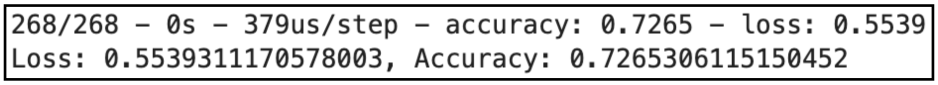


```
268/268 - 0s - 352us/step - accuracy: 0.7271 - loss: 0.5550
Loss: 0.5549647808074951, Accuracy: 0.7271137237548828
```

- Attempt #3: This model only increases the neurons in the first layer from 24 to 48 to find middle ground between the previous models. There was also minimal change in accuracy here at 72.6%.

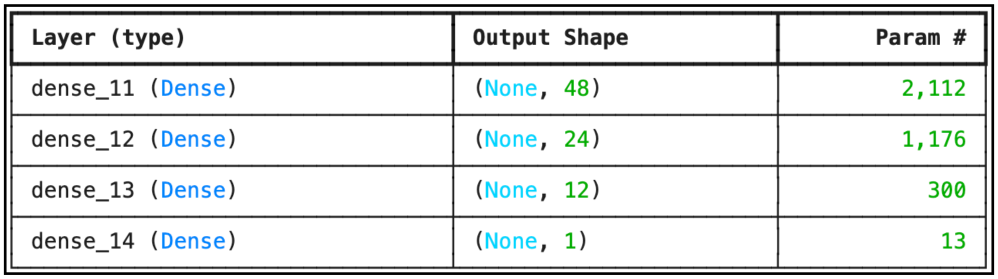


Layer (type)	Output Shape	Param #
dense_7 (Dense)	(None, 48)	2,112
dense_8 (Dense)	(None, 24)	1,176
dense_9 (Dense)	(None, 12)	300
dense_10 (Dense)	(None, 1)	13

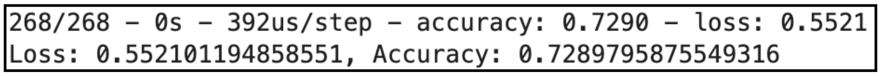


```
268/268 - 0s - 379us/step - accuracy: 0.7265 - loss: 0.5539
Loss: 0.5539311170578003, Accuracy: 0.7265306115150452
```

- Attempt #4: Since changes to the neurons and layers did not make much of a change, this model’s third layer uses sigmoid activation. The change in activation did little to the accuracy with 72.9%



Layer (type)	Output Shape	Param #
dense_11 (Dense)	(None, 48)	2,112
dense_12 (Dense)	(None, 24)	1,176
dense_13 (Dense)	(None, 12)	300
dense_14 (Dense)	(None, 1)	13



```
268/268 - 0s - 392us/step - accuracy: 0.7290 - loss: 0.5521
Loss: 0.552101194858551, Accuracy: 0.7289795875549316
```

Summary

This neural network model is about 72.9% accurate when predicting if Alphabet Soup’s funded ventures were successful. Despite a few attempts to optimize the model, the accuracy at 75% could not be reached with changes in layers, number of neurons, and activation.

Another model that can be used for this data is random forest. This would give us insight to the weight of the features and allow us to reconfigure that did that goes into the model.