Deep Learning Project Analysis:

Overview:

This project was to create a tool that would be able to predict which applicants would have the highest chance of success if they were to be funded by Alphabet Soup. There was a csv with more than 34,000 organizations to analyze. The purpose of this analysis is to showcase what was made and how it analyzes the data given.

Results:

Data Preprocessing:

- The EIN and NAME columns were removed to clean the dataset.

```python
# Drop the non-beneficial ID columns, 'EIN' and 'NAME'.
#  YOUR CODE GOES HERE
application_df = application_df.drop(columns = ['EIN', 'NAME'])
```
[3]

```python
# Determine the number of unique values in each column.
#  YOUR CODE GOES HERE
for x in application_df.columns:
    print(x, len(application_df[x].unique()))
```
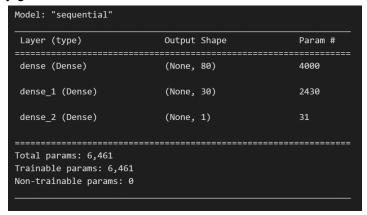[4]

```
···  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
```

- APPLICATION_TYPE and CLASSIFICATION were grouped into an "other" category where cutoff points were determined for "rare" categorical variables.
- pd.get_dummies() was used to encode variables to determine whether it was successful through 1s and 0s

Compiling, Training, and Evaluating the Model:
- There were 3 layers for each model
- 6441 params were created and the attempt was 73% accuracy which was below the 75% accuracy goal.

```
Model: "sequential"

_____
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 80)                4000

 dense_1 (Dense)             (None, 30)                2430

 dense_2 (Dense)             (None, 1)                 31

=================================================================
Total params: 6,461
Trainable params: 6,461
Non-trainable params: 0
_____
```

```python
# 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: 0.5631 - accuracy: 0.7293 - 164ms/epoch - 613us/step
Loss: 0.5631100535392761, Accuracy: 0.7293294668197632
```

Summary:

I was unable to save and export my results to an HDF5 file and I was unable to optimize it through google colab.