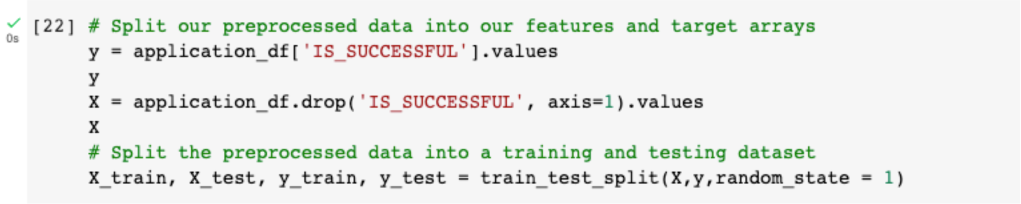
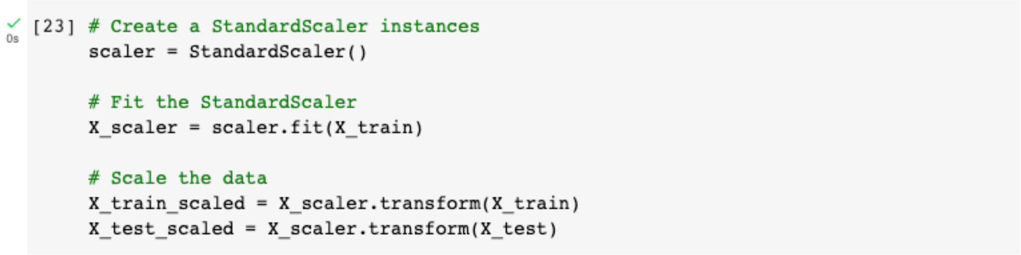
1. Deep learning was used to determine how successful applicants were for funding.

2. Results:

- Data Preprocessing



```
[22] # Split our preprocessed data into our features and target arrays
     y = application_df['IS_SUCCESSFUL'].values
     y
     X = application_df.drop('IS_SUCCESSFUL', axis=1).values
     X
     # Split the preprocessed data into a training and testing dataset
     X_train, X_test, y_train, y_test = train_test_split(X,y,random_state = 1)
```

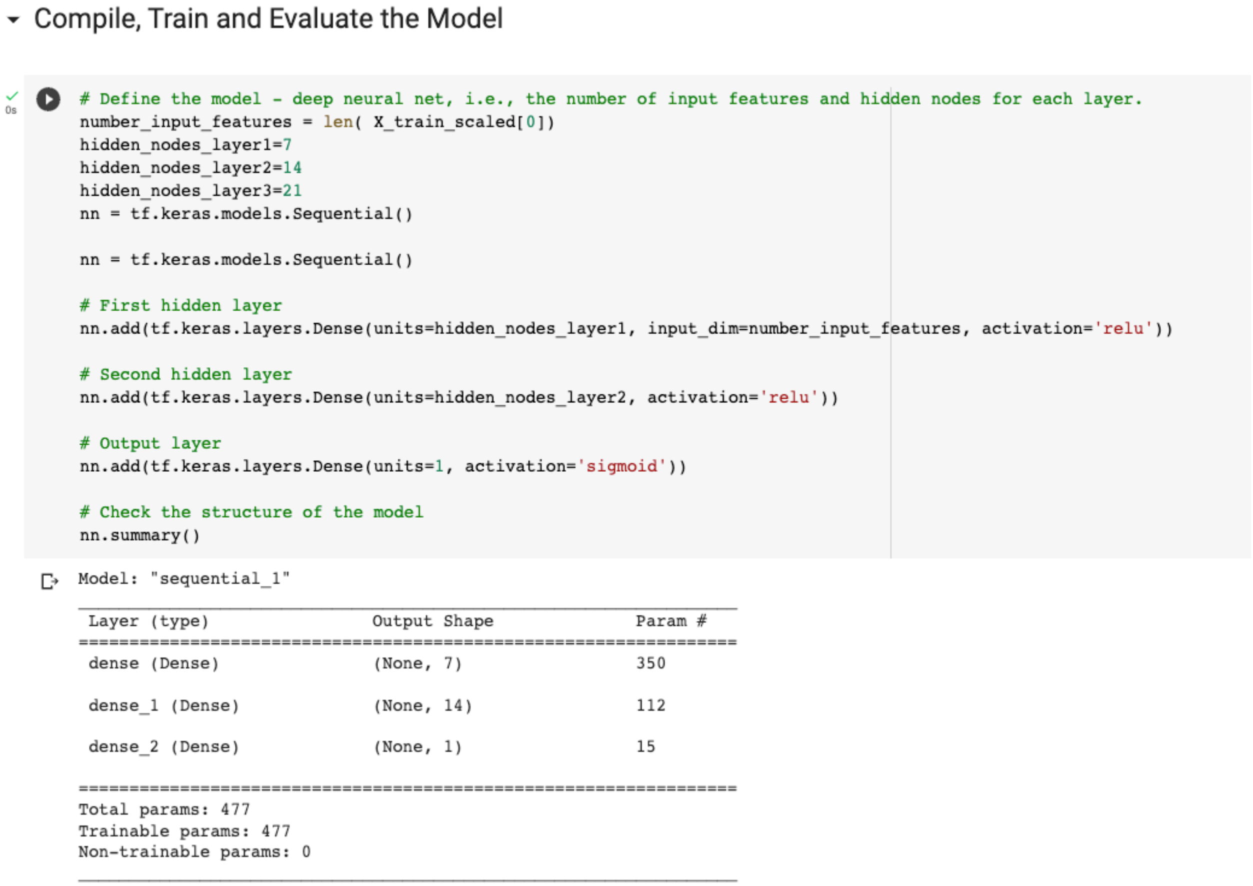


```
[23] # Create a StandardScaler instances
     scaler = StandardScaler()

     # Fit the StandardScaler
     X_scaler = scaler.fit(X_train)

     # Scale the data
     X_train_scaled = X_scaler.transform(X_train)
     X_test_scaled = X_scaler.transform(X_test)
```

- Compiling, Training, and Evaluating the Model



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

-

3. Summary: The model was able to predict at 79% accuracy.