Anderson Igben

Rice Data Analytics and Visualization

Charity Funding Predictor

In the first model EIN and NAME columns were dropped from the dataset as a preprocessing step. Application Type and Classification was then analyzed for binning before the model was created. These were the targets while the other columns were the features. Once a split, training and testing dataset were created I went ahead and compiled train and evaluate the model. There were two layers and one output layer. The picture below visualizes what was done in code

Compile, Train and Evaluate the Model

In [25]:
```
# Define the model - deep neural net, i.e., the number of input features a
number_input_features = len( X_train_scaled[0])
hidden_nodes_layer1=90
hidden_nodes_layer2=50

nn = tf.keras.models.Sequential()

# First hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer1, input_dim=number_i

# 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_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_13 (Dense)            (None, 90)                4050

 dense_14 (Dense)            (None, 50)                4550

 dense_15 (Dense)            (None, 1)                 51

=================================================================
Total params: 8,651
Trainable params: 8,651
Non-trainable params: 0
_________________________________________________________________
```

Once completed the epochs was set to 100 and I went ahead to train the model. My final evaluation was set to an accuracy of Accuracy: 0.7269970774650574. On the other hand, to better improve this model and increase the accuracy score, in the preprocessing steps I dropped only the EIN columns and then evaluated the NAME columns. Columns for binning remained the same however I then added another layer to the model and kept the number of modes the same. The picture below depicts changes made.

```
In [19]: # Define the model - deep neural net, i.e., the number of input features ar
number_input_features = len( X_train_scaled[0])
hidden_nodes_layer1=90
hidden_nodes_layer2=50
hidden_nodes_layer3=25

nn = tf.keras.models.Sequential()

# First hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer1, input_dim=number_ir

# Second hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer2, activation="relu"))

#Added Thrid hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer3, activation="relu"))

# Output layer
nn.add(tf.keras.layers.Dense(units=1, activation="sigmoid"))

# Check the structure of the model
nn.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 90)                40230

 dense_1 (Dense)             (None, 50)                4550

 dense_2 (Dense)             (None, 25)                1275

 dense_3 (Dense)             (None, 1)                 26

=================================================================
Total params: 46,081
Trainable params: 46,081
Non-trainable params: 0
_________________________________________________________________
```

After proceeding to compile the model and train it as well, I increased the accuracy to an impressive 0.7905539274215698 which was better than the initial model and even more than the required limit of 75%.