**Alphabet Soup Charity Binary Classifier Analysis:**

The nonprofit foundation Alphabet Soup wants a tool that can help it to select the applicants for funding with the best chance of success in their ventures. For this purpose, data of 34299 rows \* 12 columns has been analyzed using machine learning and neural networks, a binary classifier is created to predict if the applicants will be successful if they were funded by Alphabet Soup.

Below are the columns:

* EIN and NAME—Identification columns
* APPLICATION\_TYPE—Alphabet Soup application type
* AFFILIATION—Affiliated sector of industry
* CLASSIFICATION—Government organization classification
* USE\_CASE—Use case for funding
* ORGANIZATION—Organization type
* STATUS—Active status
* INCOME\_AMT—Income classification
* SPECIAL\_CONSIDERATIONS—Special considerations for application
* ASK\_AMT—Funding amount requested.
* IS\_SUCCESSFUL—Was the money used effectively. (Target variable)

**A -** In the first study, data has been preprocessed by dropping identification columns, checking unique values, creating bins for CLASSIFICATION and APPLICATION\_TYPE columns, converting categorical data to numeric, scaling the data, identifying future and target arrays, splitting as train – test. Then, binary classification model, compiled, trained, and evaluated to calculate the model’s loss and accuracy, as shown below:

nn = tf.keras.models.Sequential()

# First hidden layer

nn.add(tf.keras.layers.Dense(units=80, activation="relu", input\_dim=35))

# Second hidden layer

nn.add(tf.keras.layers.Dense(units=30, activation="relu"))

# Output layer

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

with 100 epochs below are the results for first study:

**Loss: 0.5617768168449402, Accuracy: 0.7254810333251953**

**B** - In the second part, optimization methods have been applied to the model to achieve a target predictive accuracy higher than 75%.

Below are some steps taken for this purpose:

1. Contribution of each feature to accuracy has been checked by dropping each

column separately under same model parameters and with Epoch 100.

Dropped Column Loss Accuracy

ASK\_AMT 0,562 0,7195

USE\_CASE 0,57 0,7223

ORGANIZATION 0,57 0,7229

INCOME\_AMT 0,56 0,7220

AFFILIATION 0,613 0,653

SPECIAL\_CONSIDERATIONS 0,56 0,7252

STATUS 0,56 0,7237

CLASSIFICATION 0,58 0,7174

APPLICATION\_TYPE 0,59 0,7067

It has been seen that dropping none of them is not contributing to increase

accuracy of 0,7255.

1. However, since INCOME\_AMT column has 24388 “0” value, it has been dropped.
2. Creating more bins for rare occurrences in columns,
3. In ASK\_AMT column according to value\_counts() there are 25398 “5000” value. The first binning has been created with 9 bins, the loss and accuracy were: 0,582 and 0,7205 under same parameters.

amount\_bins = [0, 5000, 10000, 20000, 50000, 100000, 200000, 500000, 1000000, np.inf]

amount\_labels = [1, 2, 3, 4, 5, 6, 7, 8, 9]

1. Second binning, with 2 bins the loss and accuracy were: 0,566 and 0,7228 under same parameters.

amount\_bins = [0, 5000, np.inf]

amount\_labels = [1, 2]

So, the second one is used for optimization.

1. Under the same parameters one more hidden layer is added to model as activation function “relu” with 10 neurons. The loss and accuracy were 0,56 and 0,7198. So, it didn’t increase accuracy and was not included in optimization.
2. Under same parameters as output function “tanh” is tested. (loss: 0,59, accuracy: 0,7173) but it didn’t give higher accuracy than “sigmoid” function.
3. Under same parameters elu – elu - sigmoid activation functions were used (80-30-1 neurons) epoch=100 and the loss: 0,556 accuracy: 0,7198.
4. Train-test split is made with 0,20 so training data has been tried to be increased. The results were still low (loss: 0,56 accuracy: 0,7206) so did not included to final optimization.
5. Increasing the number of epochs to 200 (loss:0,56 accuracy: 0,7187) did not increase the accuracy either.
6. Also, adding more neurons (90,40) (loss: 0,57 accuracy: 0,7187) did not change the situation.
7. Finally, after testing above steps for the final optimization; INCOME\_AMT column has been dropped, ASK\_AMT is used with 2 bins, 85-35-1 neurons used for layers with relu-relu-sigmoid activation functions, with 100 epochs, the final loss is 0,56 and the accuracy is 0,7206 which are still below 0,75.
8. In conclusion, to increase accuracy of this model more data and/or features are needed.