**Deep Learning Model Analysis**

**Overview**

The non-profit foundation Alphabet Soup wants a tool that can help it select the applicants for funding with the best chance of success in their ventures. With your knowledge of machine learning and neural networks, you’ll use the features in the provided dataset to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup.

From Alphabet Soup’s business team, you have received a CSV containing more than 34,000 organizations that have received funding from Alphabet Soup over the years. Within this dataset are a number of columns that capture metadata about each organization, such as:

* **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

Use Machine learning and neural networks to analyse whether the applicants will be successful if they were to be funded by the Alphabet Soup Charity.

**Data Pre-Processing**

* The target is the column “IS\_SUCESSFUL” as indicates whether or not the money is effectively used by using the numbers 0 and 1.
* The columns “EIN” and “NAME” were removed and the features used as columns are below:

1. **APPLICATION\_TYPE**—Alphabet Soup application type
2. **AFFILIATION**—Affiliated sector of industry
3. **CLASSIFICATION**—Government organization classification
4. **USE\_CASE**—Use case for funding
5. **ORGANIZATION**—Organization type
6. **STATUS**—Active status
7. **INCOME\_AMT**—Income classification
8. **SPECIAL\_CONSIDERATIONS**—Special considerations for application
9. **ASK\_AMT**—Funding amount requested**IS\_SUCCESSFUL**—Was the money used effectively

* Next, the categorical data was converted into numeric data.

**Compile, Train, and Evaluate the Model**

The Keras Sequential model was used.

Graphical user interface, text

Description automatically generated

As shown in the image above:

* Input features is 43
* Output layer size is 1
* Hidden nodes in layer 1 is 85
* Hidden nodes in layer 2 is 30.

**The final result;**

Text

Description automatically generated

**As shown above, the accuracy was 72.96% and so does not meet the 75% target.**

**Optimization 1**

The image below shows the code to define the model.

* Text

  Description automatically generatedThe “APPLICATION\_TYPES” and “CLASSIFICATIONS” count was reduced from 500 and 1000 to 100 and 100 respectively. The “APPLICATION\_TYPES” and “CLASSIFICATIONS” below 100 were binned as the “Other” category.
* The number of Input features was increased from 43 to 50.
* The image shows hidden nodes 1, 2 and 3 being 99, 99 and 34 respectively.

**The results obtained a 72.86% accuracy compared to the required 75%.**

Graphical user interface, text

Description automatically generated

**Optimization 2**

* The CLASSIFICATIONS had a count of less than 150.
* The Input features was reduced to 47.
* Hidden node layers 1, 2 and 3 were 93, 93 and 32 respectively.
* For hidden layers 1 and 2, the activation function “tanh” was used rather than “relu”.

Text

Description automatically generated

Text

Description automatically generated

**Results show an accuracy of 73.08% which is less than both previous models.**

Text

Description automatically generated

**Optimization 3**

* A new function “kerastuner” was used in order to be able to automate the tuning for the hyperparameters. The function would further allow the best selection of the activation functions as well as the number of neurons within each layer and the number of hidden layers

Graphical user interface, text, application

Description automatically generated

* As shown in the image to the left, the most accurate value obtained was 73.99.
* Text

  Description automatically generatedThe best hyperparameters for the model, determined by the automation are shown to the left.

**Result shows the model gave a result of 73.14%, which is less than the previous model.**

Text

Description automatically generated

**Optimization 4**

* The column “NAME” remains in the data.
* Columns “EIN”, “ORGANISATION” and “CLASSIFICATION” were removed.
* The “APPLICATION\_TYPES” and “CLASSIFICATIONS” count was reduced to 500 and 1000 respectively.
* The “NAME” count was binned with values less than 21.
* The input features was 156 as the “NAME” column was added.
* Both layers 1 and 2 obtained hidden nodes of 311 and 105 respectively.
* The activation functions in both layers is “tanh”.

Text

Description automatically generated

**Result – The model produces an accuracy value of 76.50% which is greater than the required 75%.**

Graphical user interface, text

Description automatically generated

**Random Forrest Classifier Results**

Graphical user interface, text

Description automatically generated

**Logistic Regression Results**

**Graphical user interface, text

Description automatically generated**

**Summary of Results**

1. The initial Pre-process optimization gave an accuracy score of 72.96% which is less than the required 75%.
2. Following the first optimization, the model was further optimized by:

* Increasing and decreasing the input features, epochs and hidden neurons.
* Altering the binning values for the columns “APPLICATION\_TYPE” and “CLASSIFICATION”.
* Adding hidden layers and applying different activation functions to the model.
* Adding and removing different columns from the data set.

1. Once the “NAME” column had been added back into the data, the accuracy increased above 75% with the final optimization obtaining a final value of 76.5%.
2. Finally, there was a slight difference between the accuracies for each optimization, however the differences between the values were insignificant.
3. The Supervised Machine Learning techniques can be seen to be very useful in being able to build the tool. As shown above, the Random Forrest Classifier model has a training data score of 81.12% compared to the Logistic Regressions 72.37%.
4. The RFC and LR models however have a similar value for the testing data score with both scoring 71% and 72% respectively.
5. As the RFC model provided a value of 81%, it can be considered and studied when comparing to the neural network’s models.