**Performance of Charity funding predictor model**

**Overview** **[[1]](#footnote-1)**

The primary purpose of this analysis is that using the features in the provided dataset to create a binary classifier that can predict whether applicants will be successful if funded by Alphabet Soup then optimize the predictor performance by optimization in dataset preprocess and model and show how much they affect the performance

**Result**

* **Data preprocessing:**
* **Target [[2]](#footnote-2)** :
* IS\_SUCCESSFUL is considered the target for the model as the propose is whether applicants will be successful if funded.
* **Feature which was deleted [[3]](#footnote-3)**
* EIN and NAME were dropped and weren’t considered being useful in first attempt
* For optimizing EIN, STATUS, SPECIAL\_CONSIDERATIONS were dropped as they are not correlated to target enough.
* **Features which were used [[4]](#footnote-4)**
* For the first attempt all the features were used exceptEIN, NAME**.**
* But for optimizing the NAME column considered be useful so all features were used exceptEIN, STATUS, SPECIAL\_CONSIDERATIONS .
* **Compiling, Training, and Evaluating the Model:**
* **Model** **before optimizing**: First a Sequential model with dense layers of Input layer, two hidden layers with 80 and 30 neurons and using a 'relu' activation function for both hidden layers. One 'sigmoid' activation function with one neuron for output layer. (Figure 1). Model was fitted in only on training dataset (Figure 2)

![Table

Description automatically generated]()

Figure 1: Model before optimizing

* **Model** **after optimizing:** a Sequential model with dense layers of Input layer, three hidden layers instead of two with 443 and 35, 3 neurons and using tanh activation function instead of relu for three hidden layers. One 'sigmoid' activation function with one neuron for output layer. As our input shape increased from 43 to 443, increasing the number of neurons in the first hidden layer improved the accuracy (Figure 3).**[[5]](#footnote-5)**

![Table

Description automatically generated]()

Figure 2:Optimized model

For achieving model performance firstly, I added neurons in first hidden as our input shape increases from 43 to 443, then I increased the neurons of second to 35 and then adding another hidden layer with 3 neurons. **[[6]](#footnote-6)**

**Summary:**

before any optimization in data preprocessing and modeling (Figure 3):

* Loss: 0.569678544998169, Accuracy: 0.7256559729576111

![A picture containing chart

Description automatically generated]()

Figure 3 : Loss and Accuracy before any optimization

**Step\_1&2**: In part of data preprocessing the NAME column wasn’t considered as a noisy column and instead it categorized and binned.

In part of data preprocessing, The STATUS, SPECIAL\_CONSIDERATIONS, and EIN columns were dropped as it doesn’t be correlated to target much enough .

* Loss: 0.4908713400363922, Accuracy: 0.7860058546066284

![Graphical user interface

Description automatically generated]()

Figure 4: Loss and Accuracy after dataset preprocessing optimization

**Step\_3:** In part of model making: Additional neurons are added to the hidden layers. Additional hidden layers are added. The activation function of hidden layers are changed from relu to tanh. Adding early stopping callback for decreasing overfitting:

1. Changing 80 neurons to 443:

* Loss: 0.452499121427536, Accuracy: 0.7845481038093567

1. Changing 30 neurons to 35:

* Loss: 0.45344942808151245, Accuracy: 0.7868804931640625

1. Adding a hidden layer by 3 neurons before output layer:

* Loss: 0.4491606652736664, Accuracy: 0.7879008650779724

So, we achieved the target model performance, our Loss was decease from 0.57 to 0.45 and accuracy was increased from 0.72 to .79 before and after optimization.[[7]](#footnote-7)

![A picture containing graphical user interface

Description automatically generated]()

Figure 5:loss and accuracy after optimization

1. Overview of the analysis: Explain the purpose of this analysis. [↑](#footnote-ref-1)
2. What variable(s) are considered the target(s) for your model? [↑](#footnote-ref-2)
3. What variable(s) are neither targets nor features, and should be removed from the input data? [↑](#footnote-ref-3)
4. What variable(s) are considered to be the features for your model? [↑](#footnote-ref-4)
5. How many neurons, layers, and activation functions did you select for your neural network model, and why? [↑](#footnote-ref-5)
6. What steps did you take to try and increase model performance? [↑](#footnote-ref-6)
7. Were you able to achieve the target model performance? [↑](#footnote-ref-7)