Alphabet Soup Foundation Applicant Selection Model

By Sarah McLain

18 Aug 2023

Introduction: Predicting Funding Success for Nonprofits

In the world of nonprofit work, it's important to spend money in ways that make a positive difference. Alphabet Soup, a group that helps nonprofits, wants to get better at picking which organizations to fund. They're using machine learning to create a tool that can guess if a nonprofit will do well with their money.

The provided dataset contains a comprehensive collection of over 34,000 organizations that have received funding from Alphabet Soup. Each organization is characterized by a range of attributes, including identification data, application types, sector affiliations, government classifications, funding purposes, and more.

By analyzing these attributes and employing advanced machine learning techniques, this study seeks to construct a predictive model capable of categorizing applicants as successful or unsuccessful recipients of Alphabet Soup's funding. This tool holds the potential to significantly streamline the funding selection process, ensuring that resources are allocated to ventures that are likely to achieve meaningful outcomes. Through the application of machine learning algorithms to this real-world dataset, we aim to provide Alphabet Soup with valuable insights that will aid in the efficient allocation of resources.

Preprocessing & Setting up the Model

In the preprocessing phase of the data, we focused on enhancing the quality of features for our machine learning model. The first thing that was done was to remove the Name and EIN columns of the data as these variables will not hold influence on whether or not a program was successful. Next, the data was binning to group infrequent application types and classifications into an "Other" category. This step streamlined the data and made it more manageable. For application types and classifications that appeared fewer than 500 and 1000 times respectively, this approach ensured that we maintain significant representation in our data without overwhelming the model with excessive categorical variables.

Additionally, we transformed categorical data into a numerical format using the pd.get_dummies method. This conversion facilitated compatibility with our neural network architecture.

The model architecture features a deep neural network that includes two hidden layers. To start with the initial model was kept simple with the first hidden layer, containing 10 units and the second hidden layer containing 5; the ReLU activation function was employed for both layers. These choices were made to prevent an initial overfitting of the data. The output layer comprises a solitary neuron activated by the sigmoid function, well-suited for binary classification tasks.

The model's loss was minimized using the binary cross-entropy loss function, and we employed the Adam optimizer. These choices were influenced by their effectiveness in training deep neural networks. Our model underwent 100 epochs during training to iteratively fine-tune its parameters.

Results Initial model

When tested against the evaluation data, the model showed a loss of roughly 0.5547 and an accuracy of approximately 72.62%. These outcomes suggest that there is potential for enhancing the model's performance. To achieve better results, further modifications and refinements should be explored in order to enhance its accuracy.

Results

Test data:

Model Loss	0.5547
Model Accuracy	72.62%

Optimizing the Model

Steps that were taken:

1. Reinstated the 'Name' column and created bins. This increased the number of features in X_train from 43 to 398. The model was kept the same as the initial model.

   Results

   Test data:

Model Loss	0.4509
Model Accuracy	78.58%

2. Increase epochs from 100 to 200 to give the model more time to discover patterns.

   Results
   Test data:

Model Loss	0.4605
Model Accuracy	78.35%

3. Add another hidden layer, increase units in the layers, change activation of layers. Change epochs back to 100.
   Model:
   - Layer 1: 100 units, activation reLU, input dim 398
   - Layer 2: 30 units, activation sigmoid
   - Layer 3: 10 units, activation sigmoid
   - Output Layer: units 1, activation sigmoid

Results

Test data:

Model Loss	0.4600
Model Accuracy	78.83%

Conclusion

In the effort of enhancing the model's accuracy, a significant finding emerged with the reintroduction of the 'Name' column. This change in the preprocessing of the data increased the number of features of the data from 43 to 399, subsequently, increasing the model’s accuracy from 72.62% to 78.58%. The discernible impact of this specific feature underscores its pivotal role in capturing essential patterns within the data. Despite other explorations conducted in model architecture, data preprocessing, and hyperparameters, it became evident that the reinstatement of this feature held the most pronounced influence on accuracy enhancement. This observation underscores the intricate web of factors influencing model performance, encompassing data quality, feature selection, and the complex nature of predicting nonprofit funding success. This exploration underscores the notion that the path to accurate predictions traverses multiple dimensions, where advancement is frequently molded by intricate interplays within the domain of intricate data.