Report: Alphabet Soup

Overview

A deep-learning neural network model was trained and evaluated on Alphabet Soup Foundation records in an effort to use features of applications to predict how effectively funding will be spent.

If successful, this classifier could be used by the Alphabet Soup Foundation to preemptively identify successful investments based on basic characteristics of the funding application. In the long term, this could drastically improve the charity's impact by channeling their funds towards valuable projects.

Results

Data Preprocessing

- The features used to build the model are as follows:
    - Application Type
    - Classification
    - Use Case
    - Organization
    - Status
    - Income Amount
    - Special Considerations
    - Ask Amount
- The target variable for the model is "Success", a predetermined binary variable encoding whether Alphabet Soup believes their funding to the applicant was used effectively.
- Identifiers (EIN and Applicant Name) were removed from the data set prior to model building.

Compiling, Training, and Evaluating the Model

- An initial neural network model was compiled with 3 hidden layers, composed of 10, 8, and 6 nodes, respectively, with a RelU activation function. Based on computational restraints and the decent sample size obtained from the Alphabet Soup records (over 34,000), this simple deep learning model was a reasonable place to start.
- Target performance was 75% accuracy. The original neural network achieved an accuracy of 73.1%.
- Steps were taken to optimize the model as follows:
    - It was observed that almost 90% of applications in this dataset asked for exactly $5000. However, amounts applied for include sums as high as $8.5B. This extreme skew could be impacting or confusing the model. To alleviate the skew, this numerical variable was binned into amounts equal to $5000, between $5000-10K, and greater than $10K. This did not have a meaningful impact on the accuracy of the model (0.732)
    - Overfitting did not appear to be an issue with the original model – accuracy on the training set was comparable to accuracy on the test data. Thus, it is reasonable to

expand the structure of the neural network to improve accuracy. This was accomplished in two more optimization steps.

    - First, an additional layer with 4 nodes was added to the original model. This did not significantly improve the model’s accuracy (0.731).
    - Second, all four layers were given 10 nodes. This also did not significantly improve the model’s accuracy (0.732).
- See Table 1 below for a Summary of these optimization steps.

Table 1. Summary of accuracy and loss statistics for the four tested neural network models.

	Original NN	**Binned Ask Amount**	**Increased Hidden Layers**	**Increased Nodes**
Accuracy	0.731	0.732	0.730	0.732
Loss	0.556	0.554	0.559	0.554

Summary

After three rounds of optimizing the deep-learning neural network model, we were unable to meet the 75% accuracy performance target.

We recommend exploring other modeling techniques to develop a classifier for this data. Popular and effective strategies for binary classification include Logistic Regression, k-Nearest Neighbors, Decision Trees, and Support Vector Machine models. Each of these can be optimized using a number of parameters that can be explored (programmatically, in some cases). As the application features are already relatively few in this dataset, feature selection techniques (PCA, t-SNE, Lasso) could be explored but may not help much.