Lacey Morgan

05/20/2023

*Predictive Model Development for Nonprofit Funding Programs*

# Introduction

Alphabet Soup, a nonprofit foundation, sought a tool to predict the success of applicants' ventures if funded. Utilizing machine learning and neural network techniques, I was tasked with creating a binary classifier based on a provided dataset with features detailing various organizations' information.

# Data Preparation

The dataset, comprising over 34,000 organizations funded by Alphabet Soup, included identifying details and metadata such as application type, affiliation, classification, use case, organization type, status, income amount, special considerations, funding request amount, and a measure of successful funding utilization.

I refined the dataset by adding a 'NAME\_COUNT' column, reflecting the frequency of each organization's name. After this, I removed the 'NAME' and 'EIN' columns. Additionally, the 'STATUS' column was dropped due to a highly skewed representation, with one class accounting for all but five instances. Next, the 'INCOME\_AMT' column was mapped to enhance its ordinality. The 'APPLICATION\_TYPE' and 'CLASSIFICATION' columns were binned, with less frequent classifications grouped under 'Others.' Lastly, categories were encoded, barring 'INCOME\_AMT,' as it was already adequately mapped. Finally, the two numeric columns, ‘ASK\_AMT’ and ‘NAME\_COUNT’ were scaled based on skewness and kurtosis.

# Model Creation

I utilized Keras Tuner to construct and optimize the neural network model architecture. The model was built using a Keras sequential model with an input shape of 48. The hyperparameters that were tuned included the number of hidden layers (ranging from one to four), the number of nodes in each layer (ranging from 100 to 1000), and the choice of activation function (ReLU, Sigmoid, or Tanh).A hyperband tuner performed the hyperparameter search, aiming for the best validation accuracy over a maximum of ten epochs. The best model and hyperparameters were identified and summarized.

# Model Performance

The model delivered a validation accuracy of 0.763, exceeding the target of 0.75. Upon testing, the model achieved a loss of 0.490 and accuracy of 0.763, demonstrating its predictive capability in estimating the success of funded ventures.

A screenshot of a computer

Description automatically generated

# Permutation Importance

This involves shuffling individual features in the validation data and measuring how much the model's performance decreases. A significant decrease indicates that the feature is important.

In implementing this on the model I can see what is important to the model. Higher values of feature importance indicate features that were most valuable to the model. The most important features were the NAME\_COUNT which I created and scaled and was far more important to the model than the others. The aggregation was followed in importance ORGANIZATION\_Association and then INCOME\_AMT which I had previously mapped to highlight ordinality.

A picture containing text, screenshot, number, font

Description automatically generated

# Conclusion

The developed predictive model, powered by machine learning and a Keras neural network, can effectively assist Alphabet Soup in selecting applicants with the highest likelihood of success. This tool will ensure more strategic and fruitful allocations of funding to maximize the organization's social impact.

# Next Steps

In order to further improve the performance of my model, there are a few next steps that I could take. Firstly, it would be advantageous to fine-tune the Keras tuner by refining the range of nodes and using smaller steps. This will allow for a more detailed exploration of the optimal architecture, potentially leading to better results.

In addition, incorporating cross-validation into our model evaluation process would be highly beneficial. By performing cross-validation, I can assess how the model is likely to perform on unseen data and obtain a more reliable estimate of its generalization capabilities.

Furthermore, it is worth considering the utilization of ensemble classification algorithms provided by scikit-learn. Ensemble methods, such as Random Forests, Gradient Boosting, Stacking and Voting, combine multiple models to improve predictive performance. These tools could potentially create a more effective model.

By implementing these suggested next steps, I can further enhance the effectiveness and reliability of the machine learning model, ultimately leading to improved performance and better predictions.