**Neural Network Analysis for Predicting Funding Success**

**1. Introduction**

This report details the development of a neural network model designed to predict whether organizations will effectively use funding provided by Alphabet Soup. The analysis covers data preparation, model design, training, evaluation, and ideas for alternative approaches. The overall goal is to build a binary classifier that forecasts success based on historical data.

**2. Data Preparation**

**2.1. Selecting the Target and Features**

* **Target Variable:**
  + IS\_SUCCESSFUL indicates if an organization used the funding successfully (1) or not (0).
* **Feature Variables:**
  + All remaining columns (such as APPLICATION\_TYPE, AFFILIATION, CLASSIFICATION, etc.) serve as features after removing non-informative identifiers.
* **Removed Columns:**
  + The columns EIN and NAME were dropped because they do not contribute predictive value.

**2.2. Processing Steps**

* **Grouping Rare Categories:**
  + In columns like APPLICATION\_TYPE and CLASSIFICATION, infrequent values were consolidated under the label “Other” to reduce noise.
* **Encoding and Scaling:**
  + Categorical features were converted to numerical values using one-hot encoding.
  + Features were scaled with a StandardScaler to normalize the data for training.

*Figure 1: Data Preprocessing Flow (Placeholder Image)*

**3. Model Architecture and Evaluation**

**3.1. Model Design**

* **Input Layer:**
  + The number of neurons is set to match the number of processed features.
* **Hidden Layers:**
  + **First Hidden Layer:** 80 neurons with ReLU activation.  
    *Purpose: Capture complex, non-linear relationships in the data.*
  + **Second Hidden Layer:** 40 neurons with ReLU activation.  
    *Purpose: Further refine learned patterns and reduce dimensionality.*
* **Output Layer:**
  + 1 neuron with sigmoid activation to output a probability for the binary classification.

*Simplified Model Flow:*

scss

Copy

Input -> Dense(80, ReLU) -> Dense(40, ReLU) -> Dense(1, Sigmoid)

**3.2. Training and Performance**

* **Training Process:**
  + The model was trained for 100 epochs using the Adam optimizer and binary crossentropy loss.
* **Performance Metrics:**
  + **Test Loss:** Approximately 0.7251
  + **Test Accuracy:** Approximately 44%

**3.3. Results Section: Answering Key Questions**

* **What is the target variable?**
  + The target variable is IS\_SUCCESSFUL.
* **What are the feature variables?**
  + All columns remaining after removing non-predictive identifiers (e.g., APPLICATION\_TYPE, AFFILIATION, etc.).
* **Which columns were removed?**
  + The EIN and NAME columns were dropped because they do not add predictive value.
* **How many neurons and layers were used?**
  + The network uses an input layer sized to the number of features, a first hidden layer with 80 neurons, a second hidden layer with 40 neurons, and an output layer with 1 neuron.
* **Was the target performance achieved?**
  + No; the current model reached an accuracy of around 44%, well below the 75% target.
* **What steps were taken to improve performance?**
  + Data grouping for rare categories, one-hot encoding, feature scaling, and experimenting with the network architecture (e.g., adjusting the number of neurons and layers) were implemented. Future improvements could include adding dropout layers or tuning hyperparameters further.

*Figure 2: Training Progress (Loss and Accuracy) (Placeholder Image)*

**4. Summary and Alternative Approach**

**4.1. Overall Model Summary**

The developed neural network achieved a test accuracy of approximately 44%. This performance suggests that the current approach does not capture all the complexities of the data, indicating a need for further refinement through model tuning, additional preprocessing adjustments, or alternative architectures.

**4.2. Alternative Modeling Approach**

**Random Forest Classifier**

* **Why Consider a Random Forest?**
  + **Ease of Handling Categorical Data:** Random forests can work well with categorical features, often requiring less extensive preprocessing.
  + **Feature Importance:** They provide insights into which variables are most influential.
  + **Robustness:** This model is generally less sensitive to outliers and does not require extensive hyperparameter tuning.
  + **Performance:** A random forest might better capture non-linear interactions and improve overall predictive performance.

**5. Conclusion**

In summary, this report has outlined the process of building a neural network to predict whether organizations will succeed with funding. While the initial model only achieved around 44% accuracy, further tuning and alternative methods like Random Forests may offer improved results. Continued experimentation with preprocessing techniques and model architecture is recommended to reach the target performance level.