**Neural Network Model Analysis for Predicting Charity Success**

**1. Introduction**

**1.1 Purpose of the Analysis**

The goal of this project was to develop a deep learning model using TensorFlow to predict whether applicants to Alphabet Soup, a fictitious non-profit organization, would be successful if funded. The model aims to improve funding allocation decisions, ensuring resources are directed toward the most promising applicants.

**2. Data Preprocessing**

**2.1 Features and Target**

* **Input Features:** Included application type, affiliated organization, classification, income amount, and other encoded categorical variables.
* **Target Variable:** Binary label indicating success (IS\_SUCCESSFUL), where 1 means the funded organization was successful.

**2.2 Processing Steps**

* One-hot encoding for categorical variables
* Dropping features with high cardinality (e.g., EIN and NAME)
* Splitting into training and testing sets
* Scaling features with StandardScaler()

**3. Model Architecture**

**3.1 Initial Neural Network Model**

* **Input Layer:** Matches number of features
* **Two Hidden Layers:**
  + Layer 1: 80 neurons, ReLU activation
  + Layer 2: 30 neurons, ReLU activation
* **Output Layer:** 1 neuron with sigmoid activation
* **Compilation:** Binary cross-entropy loss, Adam optimizer

**4. Results**

**4.1 What is the structure of your model?**

The neural network consists of:

* Input layer with nodes equal to the number of input features (43)
* Two hidden layers:
  + First hidden layer: 80 neurons, ReLU
  + Second hidden layer: 30 neurons, ReLU
* Output layer: 1 neuron with sigmoid activation

**4.2 What is the model’s performance?**

The initial model achieved a **73.15% accuracy** on the test data. This suggests reasonable performance but room for improvement.

**4.3 What steps did you take to try and improve model performance?**

* Added in Name to the beneficial columns
* Added an additional layer
* Adjusted the number of nodes
* Added the batch size to 32 during the model training

**4.4 Which model performed best and why?**

The optimized model used:

* Three hidden layers with sizes [7, 14, 21]
* Dropout layers to prevent overfitting
* Learning rate tuning with Adam(learning\_rate=0.001)

This model achieved slightly better accuracy (~79%) showing that increasing depth and regularization helped.

**4.5 What are some limitations of your model?**

* Moderate accuracy (~79%) indicates a trade-off between complexity and performance
* Potential overfitting without regularization
* Neural networks are less interpretable than simpler models
* Computationally expensive and require more tuning

**4.6 How could you further improve model performance?**

* Use feature selection to reduce dimensionality
* Use ensemble methods (e.g., stacking with random forest)

**5. Summary of Results**

The deep learning model can predict charity success with approximately 73–79% accuracy. While this performance is decent, it doesn't significantly outperform simpler models like logistic regression or random forests. However, the neural network model does scale well with data and can potentially uncover nonlinear patterns.

**6. Alternative Model Consideration**

**6.1 Suggested Alternative: Random Forest Classifier**

A **Random Forest** is a strong candidate because:

* It handles categorical data well
* Offers good interpretability via feature importance
* Generally robust against overfitting
* Performs well without extensive hyperparameter tuning

**6.2 Why Use Random Forest?**

* More interpretable: You can explain why decisions are made
* Faster to train and tune
* Performs competitively with neural networks on structured data
* Naturally handles missing data and outliers

**7. Conclusion**

The neural network model provides a solid baseline for predicting which applicants to Alphabet Soup are most likely to succeed. While it achieves a reasonable level of accuracy, improvements are possible through optimization, alternative models, and enhanced data processing. A Random Forest model may provide similar or better performance with added transparency and ease of interpretation, making it worth exploring in future iterations.