***Report:***

**Overview:**

The module challenge aims to build a machine learning model for Alphabet Soup, a nonprofit organization, to predict the success of funding applicants based on historical data. The goal is to develop a binary classification model using a dataset containing over 34,000 funded organizations. Alphabet Soup seeks a model that can identify applicants most likely to succeed with their funding. In this process, a neural network is designed and trained to predict success based on various features from the dataset.

**Results:**

This module challenge is using the following techniques and skills:

* **Data Preprocessing**
* **What variable(s) are the target(s) for your model?**
* The target variable for the model is **IS\_SUCCESSFUL**, which indicates whether an organization successfully utilized the funding provided by Alphabet Soup.
* **What variable(s) are the features for your model?**
* The features used to predict the likelihood of success include:
* **APPLICATION\_TYPE**
* **AFFILIATION**
* **CLASSIFICATION**
* **USE\_CASE**
* **ORGANIZATION**
* **STATUS**
* **INCOME\_AMT**
* **SPECIAL\_CONSIDERATIONS**
* **ASK\_AMT**
* **What variable(s) should be removed from the input data because they are neither targets nor features?**
* The following variables should be excluded as they are identifiers and irrelevant for prediction:
* **EIN**
* **NAME**
* **Compiling, Training, and Evaluating the Model**
* **How many neurons, layers, and activation functions did you select for your neural network model, and why?**
* **Layers:**
* **Input Layer**: Corresponds to the number of features in the dataset.
* **Hidden Layers**: Two hidden layers, with 64 neurons in the first layer and 32 neurons in the second.
* **Output Layer**: A single neuron with a **sigmoid** activation function, suitable for binary classification.
* **Activation Functions:**
* **ReLU (Rectified Linear Unit)**: Used for hidden layers to avoid the vanishing gradient problem and promote faster training.
* **Sigmoid**: Used in the output layer to produce probabilities between 0 and 1, making it ideal for binary classification tasks.
* **Were you able to achieve the target model performance?**

The target was to achieve an accuracy greater than 75%. The model's performance is evaluated based on the optimization process. Success depends on how well the hyperparameters, network architecture, and other features were fine-tuned. If accuracy exceeds 75%, the target is met; otherwise, further optimization would be necessary.

* **What steps did you take in your attempts to increase model performance?**
* Categorical features were encoded properly using **pd.get\_dummies()**, and rare categories were grouped into an "Other" category.
* The number of neurons in the hidden layers was adjusted, and the network’s depth was experimented with to improve the model’s learning capacity.
* Key hyperparameters, such as learning rate, batch size, and the number of epochs, were adjusted to enhance model performance.
* Techniques like **dropout** is introduced to prevent overfitting and improve generalization.
* Different optimizers (e.g., **Adam**) and the number of epochs were fine-tuned to prevent underfitting or overfitting, ensuring a more accurate and stable model.

**Summary:**

This project involved the design, training, and optimization of a neural network model to predict the success of funding applicants based on organizational data. The steps included data preprocessing, model evaluation, optimization, and the creation of a comprehensive report detailing the model's performance and recommendations for improvement.