Analysis: Predicting Charity Success

Introduction

The purpose of this analysis is to build a predictive model that can determine the success of charitable organizations. The model will be trained on a dataset containing various features related to charitable applications, such as application type, affiliation, classification, use case, organization type, income amount, and special considerations. The target variable is the "IS_SUCCESSFUL" column, which indicates whether a charity was successful in achieving its goals.

Data Preprocessing

Target and Features

Question: What variable(s) are the target(s) for your model? The target variable for the model is "IS_SUCCESSFUL." It represents whether a charitable organization was successful (1) or not (0) in achieving its goals. This binary classification allows us to predict the likelihood of success for new organizations.

Question: What variable(s) are the features for your model? The features for the model include all the columns in the dataset except for the target variable. These features provide information about the charitable applications and can help predict the success of an organization. By analyzing these features, we can identify patterns and relationships that contribute to the success of charitable initiatives.

Removed Variables

Question: What variable(s) should be removed from the input data because they are neither targets nor features? Two variables, "EIN" and "NAME," were removed from the input data as they are non-beneficial for predicting the success of charitable organizations. The "EIN" column contains unique identification numbers assigned to each organization, while the "NAME" column contains the names of the organizations. These variables do not provide meaningful information that can contribute to the predictive power of the model.

Compiling, Training, and Evaluating the Model

Neural Network Architecture

In this analysis, a neural network model is employed to predict the success of charitable organizations. The model consists of three layers:

1. Input layer: The input layer has a number of neurons equal to the number of features in the preprocessed dataset. It uses the "sigmoid" activation function to introduce non-linearity to the model.
2. Hidden layer 1: This layer consists of 150 neurons and uses the "sigmoid" activation function. The purpose of this layer is to learn complex relationships between the input features.
3. Hidden layer 2: There is no hidden layer 2 mentioned in the provided code. The absence of this layer might be due to the specific requirements of the dataset and problem at hand.
4. Hidden layer 3: This layer consists of 10 neurons and uses the "tanh" activation function. The "tanh" activation function introduces non-linearity and helps capture additional patterns in the data.

5. Output layer: The output layer has 1 neuron, representing the binary classification of the target variable. It uses the "sigmoid" activation function to produce a probability output.

The choice of the number of neurons and activation functions is based on experimentation and the complexity of the problem. The model aims to strike a balance between capturing complex patterns in the data and avoiding overfitting.

Model Performance

Before training the model, it is compiled using the "binary_crossentropy" loss function, "adam" optimizer, and the "accuracy" metric. The "binary_crossentropy" loss function is suitable for binary classification problems, and the "adam" optimizer is known for its efficiency and robustness.

The model is then trained on the training data for 100 epochs. Each epoch represents a full iteration over the training dataset, allowing the model to learn from the data and optimize its parameters.

After training, the model is evaluated on the test data to assess its performance. The evaluation metrics used are the loss value and accuracy. The loss value represents the discrepancy between the predicted and actual values, while the accuracy measures the proportion of correct predictions.

Question: Were you able to achieve the target model performance?

The model achieved an accuracy of approximately 72.6% on the test data, with a loss of 0.5578. While this accuracy may be considered reasonable, there is room for improvement.

Overall Results

The trained neural network model achieved an accuracy of approximately 72.6% on the test data. While this accuracy may be considered reasonable, there is room for improvement.

Alternative Model Considerations

While the deep learning model showed reasonable performance, other models could be considered for solving the same problem:

1. Random Forest: Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It can handle both categorical and numerical features and is less prone to overfitting. Random Forest models can be useful when there are complex interactions and non-linear relationships between features.
2. Support Vector Machine (SVM): SVM is a powerful classification algorithm that works well in high-dimensional spaces. It aims to find an optimal hyperplane that separates the data points of different classes. SVM can handle both linear and non-linear classification problems and can be effective when the dataset is not too large.
3. Gradient Boosting: Gradient Boosting is an ensemble method that combines multiple weak models (usually decision trees) to create a strong predictive model. It works by sequentially adding models, each correcting the mistakes of the previous models. Gradient Boosting models, such as XGBoost or LightGBM, often yield

excellent results and are widely used in various competitions and industry applications.

The choice of the alternative model depends on several factors such as the size and nature of the dataset, interpretability requirements, and computational resources. For example, if interpretability is crucial, a Random Forest model might be a better choice. If the dataset is extremely large, a gradient boosting model optimized for efficiency, like LightGBM, could be preferred. Ultimately, the decision would depend on the specific characteristics and constraints of the problem at hand.