**Data Preprocessing**

**What variable(s) are the target(s) for your model?**

* The 'IS\_SUCCESSFUL' column from application\_df is the target variable, this is what we are trying to predict. This shows if the money was used effectively.

**What variable(s) are the features for your model?**

* The feature variables we used are:
  1. AFFILIATION—Affiliated sector of industry
  2. CLASSIFICATION—Government organization classification
  3. USE\_CASE—Use case for funding
  4. ORGANIZATION—Organization type
  5. STATUS—Active status
  6. INCOME\_AMT—Income classification
  7. SPECIAL\_CONSIDERATIONS—Special considerations for application
  8. ASK\_AMT—Funding amount requested

**What variable(s) should be removed from the input data because they are neither targets nor features?**

* Identification columns: The "EIN" and "NAME" columns are identification columns that typically provide unique identifiers for each organization. These columns usually have no direct impact on the target variable and can be dropped without affecting the model's accuracy.

**Compiling, Training, and Evaluating the Model**

**How many neurons, layers, and activation functions did you select for your neural network model, and why?**

* In my first neural network model, I used a two-layer architecture with a specific choice for the number of neurons, layers, and activation functions.
* By selecting 80 neurons in the first hidden layer (units\_1 = 80), 30 neurons in the second hidden layer (units\_2 = 30) and using the ReLU activation function (activation="relu") for both hidden layers, you aimed to create a model with sufficient complexity to capture and learn meaningful patterns in the data. The choice of ReLU activation helps introduce non-linearity and allows the model to learn complex relationships between the input features and the target variable.
* Additionally, you used a single neuron in the output layer (units=1) with a sigmoid activation function (activation="sigmoid") to model the binary classification problem. The sigmoid activation function maps the output to a range between 0 and 1, representing the probability of the positive class.
* In summary, the model architecture with the chosen number of neurons, layers, and activation functions aimed to strike a balance between complexity and simplicity, allowing the model to learn and generalize well on the given classification task.

**Were you able to achieve the target model performance?**

* As you can see below, I was only able to achieve 73%, which was not the target model performance which was 75%.

**What steps did you take in your attempts to increase model performance?**

1. Increasing the number of neurons and epochs:

* By increasing the number of neurons in a layer, the model becomes more expressive and can capture complex patterns in the data. This allows for better representation of the underlying relationships between the features and the target variable, potentially leading to higher accuracy.
* Increasing the number of epochs gives the model more opportunities to learn from the data and adjust the weights. It allows the model to refine its predictions and find better parameter values, which can lead to improved accuracy. However, it's important to find a balance as increasing epochs excessively can lead to overfitting.
* As you can see, I was only able to achieve 73%.

1. Adding more layers to the model:

* Adding more layers can provide the model with additional capacity to capture and represent intricate relationships within the data. Each layer can learn different levels of abstraction, enabling the model to extract more meaningful features and potentially improving accuracy. Deep models with multiple layers could learn hierarchical representations of the data, which can be advantageous for complex problems.
* Again, I was only able to achieve 73%

1. Using a different activation function (tanh for the second layer):
2. Introducing a different activation function, such as tanh, can affect how the model interprets and transforms the inputs. Different activation functions have different properties and can capture different types of non-linearities. By using tanh, it introduces a different non-linearity that may better suit the problem at hand, potentially leading to increased accuracy.
3. Again, I was only able to achieve 73%
4. Utilizing an Automated Optimizer (such as a hyperparameter tuner):
5. Automated optimizers, like hyperparameter tuners, systematically explore various combinations of hyperparameters, such as activation functions, number of layers, number of neurons, and epochs. This exploration can help identify the most optimal combination of hyperparameters for your specific problem, potentially leading to higher accuracy. It saves you from manually trying out different combinations and allows the optimizer to leverage its search algorithms to find the best configuration.

* Again, I was only able to achieve 73%

## **Conclusion**

The deep learning model that I have developed was unable to achieve accuracy higher than 73%. To further improve the model's performance, I can consider the following steps:

1. Adding more data:

* Increasing the size of the training dataset can help the model learn from a larger and more diverse set of examples. This can improve the generalization capability of the model and potentially lead to higher accuracy. Collecting additional data relevant to the classification problem could provide the model with more information to make better predictions.

1. Checking data cleaning:

* Ensuring that the data is properly cleaned is crucial for model performance. Cleaning includes handling missing values, handling outliers, normalizing or standardizing features, and addressing any data quality issues. By thoroughly reviewing and cleaning the data, I can mitigate the impact of noise or irrelevant information that might be affecting the model's accuracy.

1. Exploring alternative machine learning algorithms:

* Trying a different algorithm, such as Random Forest, can provide valuable insights into the importance of different features. Random Forest can measure feature importance based on how effectively each feature contributes to the overall prediction. This analysis can help identify the key predictor columns, allowing you to focus on the most informative features and potentially improve accuracy.

1. Identifying feature importance and selecting relevant attributes:

* Analyzing feature importance helps determine which attributes have the most significant impact on the output. By identifying and selecting the most important attributes, you can reduce the noise and complexity in the model. Focusing on the most relevant features can enhance the model's ability to capture meaningful patterns and improve accuracy.

1. Addressing high bias and outliers:

* High bias in the model can be caused by outliers or skewed data points that deviate significantly from most of the dataset. Identifying and addressing these outliers can help improve the model's performance. Techniques such as outlier detection, data transformation, or stratified sampling can be applied to mitigate the impact of outliers and reduce bias in the model.

1. Binning the data:

* Binning continuous variables can be useful in certain scenarios. It can help simplify the relationship between variables and the target variable by grouping similar values into bins. This can reduce the complexity of the model and make it more robust to noise or fluctuations in the data, potentially leading to improved accuracy.

In summary, to improve the deep learning model's performance, I would consider adding more data, ensuring proper data cleaning, exploring alternative algorithms, identifying feature importance, addressing bias and outliers, and applying data binning techniques. Each step aims to enhance the model's ability to capture relevant patterns and reduce noise, ultimately improving accuracy in the classification problem. It is important to iterate and experiment with these steps, evaluating the impact on model performance and fine-tuning as necessary.