**Report on the Neural Network Model**

**The purpose of the model and the target:**

I wanted to create a model to predict if the application will be successful or not.

**The data set:**

as it's shown in the plot that I created in my starter code, the data is very nonlinear and doesn’t follow any specific shape or algorithms like sinus shape or any other rhythms.

A blue and green dots

Description automatically generated

**The features and target:**

in this model I used application\_df\_encoded['IS\_SUCCESSFUL'] as my target. I used the rest of the columns as my features. (in one of my optimization tests I used PCA to reduce the number of features to 2 feature, but that method surprisingly reduced my accuracy score).

**Activation Function, layers and Neurons:**

I used relu and tanh as my hidden layers because these activation functions are usually suitable to use for none linear data. i used a high number of neurons in my first layer and a medium size in my tanh layer, because the data shape was complicated, I wanted to see if a high number of neurons would help with a better accuracy and prediction.

in another test I used softmax instead of sigmoid for my output layer, but that significantly reduced the accuracy results, therefore sigmoid is a better last layer for this data set.

Using mathematical methods to identify the best activation function for each layer, and the best number of neurons for each layer can significantly increase the accuracy rate, for example functions like “tuner” from the keras\_tuner could help to develop the more suitable activation functions and number of units. Adjusting the numbers of hidden layers and adjusting the type of layer (other than Dense) could also improve the model.

The steps I tool to modify the results:

* Reducing the number of features by using PCA
* Adding an extra hidden layer
* Changing the activation functions
* Changing the unit numbers