Charity Funding Predictor

A model to predict if an applicant for funding the non-profit organization Alphabet Soup would be successful or not was created in this project.

With the information we had, I was able to use the feature “Is successful” as our target by telling us whether the applicants were successful or not.

To create the model we had to clean the data that was given, by removing unnecessary features like names and Id’s of the organizations, as well as binding some fields to reduce the amount of outliers.

After that the model considered this as features:

* Application Type
* Affiliation
* Classification
* Use Case
* Organization
* Status
* Income amt
* Special Considerations
* Ask Amt

Application Type and Classification had to be binded into smaller classifications due to many outliers.

After cleaning and scaling the data, the categorical data was turned into numeric to be ready to create the first model.

The categorical data increased significantly the amount of features from 9 to 43 resulting as having 43 features as inputs for the model.

Model #1

The first Dense layer was created with 43 inputs, then two hidden layers with 80 and 30 neurons using Relu as activating functions. The last output layer had one neuron with sigmoid as activation function. The model was trained with 100 epochs.

Loss: 0.5572 - accuracy: 0.7250

**Trying to improve the model to at least 75%**

Model #2

The first Dense layer was created with 43 inputs, then three hidden layers with 80, 40 and 30 neurons using Relu as activating functions. The last output layer had one neuron with sigmoid as activation function. Also the epoch was modified to 200 to see if more training would result in a better accuracy.

Loss: 0.5714465379714966, Accuracy: 0.7240816354751587

*Note: Adding a third layer did not improve the model but only made it worse.*

Model #3

The first Dense layer was created with 43 inputs, then two hidden layers with 80 and 30 neurons using Sigmoid as activating functions. The last output layer had one neuron with sigmoid as activation function. Also the epoch stayed as 200 like the previous model to see if more training would result in a better accuracy.

Loss: 0.5609973669052124, Accuracy: 0.7267638444900513

*Note: The model improved just a little bit by using the same activation in all layers.*

Model #4

After seeing all those results I looked for more outliers that might have been confusing the model. Resulting in the reduction of records to train the model excluding the outliers of the feature Ask amt. With that, and the help of three hidden layers with 80, 60 and 20 neurons, and using relu for the first two and the las two sigmoid as activation, I was able to improve the accuracy of the model to 75%

**Loss: 0.5458167195320129, Accuracy: 0.75**

*Note: The feature amount asked it was important to consider in the model, unfortunately had two many values out of range that needed to not be part of training the model, therefore those records were excluded in the search for a better model.*