Background

Alphabet Soup wants to develop a tool that can help with selecting applicants for funding that have the best chance of success in their ventures. In this project a a nueral network machine learning model is developed to help identify the applicants that have the best chance of succeeding.

To build the model a CSV file containing more than 34,000 organizations that have received funding from Alphabet Soup over the years was used. Within this dataset are a number of columns that capture metadata about each organization, such as:

\* \*\*EIN\*\* and \*\*NAME\*\*—Identification columns

\* \*\*APPLICATION\_TYPE\*\*—Alphabet Soup application type

\* \*\*AFFILIATION\*\*—Affiliated sector of industry

\* \*\*CLASSIFICATION\*\*—Government organization classification

\* \*\*USE\_CASE\*\*—Use case for funding

\* \*\*ORGANIZATION\*\*—Organization type

\* \*\*STATUS\*\*—Active status

\* \*\*INCOME\_AMT\*\*—Income classification

\* \*\*SPECIAL\_CONSIDERATIONS\*\*—Special consideration for application

\* \*\*ASK\_AMT\*\*—Funding amount requested

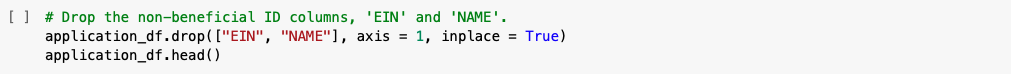
\* \*\*IS\_SUCCESSFUL\*\*—Was the money used effectively

Instructions

Step 1: Preprocess the Data

The following steps were taken to clean up the data and suitable for analysi

EIN and NAME columns were removed to from the dataset as they where not relevant to the analysis



The data had no rows with Null values as confirmed by the code below.

Graphical user interface

Description automatically generated with medium confidenceThe code below indicated that the data had duplicated columns. However on close inspection it appeared that different organisation with similar profiles were flagged as duplicates. Therefore it was decided not to delete the columns that were flagged as duplicates.

Graphical user interface, application

Description automatically generated

3. The number of unique values and data types in each column were determined using the nunique() and dtypes functions.The IS\_SUCCESSFUL column has two integers that were unique making it ideal target variable

Table

Description automatically generated

The CLASSIFICATION and APPLICATION\_TYPE type columns had 17 and 71 unique values respectively. This is too large for encoding. Therefore binning was used to reduce the unique values in the CLASSIFICATION and APPLICATION\_TYPE columns. To do this all unique values that had counts less that 200 value counts were put into the “other” category. This reduced the value counts in this column to 9 from 17

Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

The CLASSIFICATION column had 71 unique values. This was reduced to 71 to 5 buy filtering out all unique values that had less 1000 value counts. This meant that the classification column had 5 unique values . After the filtering a new dataframe X was formed

Graphical user interface

Description automatically generated with medium confidence

The categorical data in the X dataframe was then converted into numeric using the pandas get\_dummies function. Thereafter the data was split into two for training and testing. The testing data was 33% of the data size. To avoid any bias between the large and small numbers the X\_train and X\_test data was scaled using the Standardscaler().



Graphical user interface, text, application, email

Description automatically generated

### Step 2: Compile, Train, and Evaluate the Model

A deep neural network with two hidden layers was built with an hidden layer activation function of relu. The Output node was binary classifier model with only one output: that decided if the funding application was going to be successful or not. And an output layer activation of sigmoid as the model output is binary classification between 0 and 1. This model had an accuracy of 73.2%

Text

Description automatically generated

### Step 3: Optimize the Model

The model build had an accuracy of 73.2%, this was below the expected 75%. To optimise the model and reach this target, a function was created to find the parameters that can achieve this target. With keras turner it was possible to find parameters that could improve the accuracy to 74%.

Graphical user interface, text, application

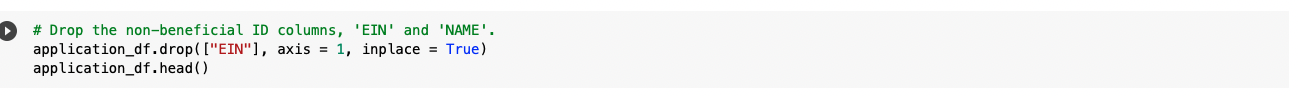
Description automatically generated

A picture containing shape

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Since keras turner could only improve the efficiency marginally it possibly means that the accuracy could not be improved further or the data processing needed a bit more. manipulating. Therefore to further improve the efficiency the code in jupyter notebook was re-run this time with only dropping the “EIN” column “NAME” column from the dataframe as opposed to dropping the both EIN and NAME columns from the dataframe as was done in the first analysis.

This improved the efficiency to 80% as can be seen in the. The code for this optimisation can be seen in the ML\_Starter\_Code\_optimised2 jupyter notebook.

Graphical user interface, text

Description automatically generated

So adding the NAME column improved the efficiency to 80%. This was possible because adding the NAME helps the model to recognise data from each company. This is perhaps vindicated by the fact that in the dataframe without the NAME column rows that weren’t duplicates were highlighted as duplicates.