**Overview of the analysis:**

Using machine learning and neural network skills, you'll use the attributes in the provided dataset were used to assist Bek in developing a binary classifier capable of predicting whether applicants will be successful if they are funded by Alphabet Soup.

Beks got a CSV comprising more than 34,000 groups that had received financing from Alphabet Soup throughout the years from Alphabet Soup's business team. A number of columns in this dataset record metadata on each organisation, 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

**Results**

***Data Preprocessing***

* The target variable was IS\_SUCESSFUL and it was a binary 0 or 1 which tells, if it was successful and money was used effectively
* The variables which were initially used were:
  + **APPLICATION\_TYPE <- was removed after analysis**
  + AFFILIATION
  + CLASSIFICATION
  + USE\_CASE
  + ORGANIZATION
  + STATUS
  + INCOME\_AMT
  + SPECIAL\_CONSIDERATIONS
  + **ASK\_AMT <- was removed after analysis**
  + IS\_SUCCESSFUL
* Initially APPLICATION\_TYPE and CLASSIFICATION were both re-binned and put into lesser categories to ensure that the categories are evenly distributed.
* The data was first ONE-Hot encoded for all categorical variables
* After encoding the data was standardised using StandardScaler and that was used to pre-process all the data
* The variables such as STATUS, EIN and NAME were variables which were neither targets nor features and were removed. After further analysis, it was found that APPLICATION\_TYPE and INCOME\_AMT had a very weak correlation with the IS\_SUCESSFUL and were let go of to do further testing

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

* Initially 2 hidden layers + 1 output layers were used to figure out the accuracy score with all the variables highlighted above– the score was found to be about 72%
* In the first iteration of changes – 3 hidden layers + 1 output layer were used to figure out the accuracy score – the score improved by 0.5% to 72.5%
* In the second change – variables such as ASK\_AMT and APPLICATION\_TYPE were found to be less valuable in comparison with others using Random Forest and importance of variables and re-ran through analysis of Random forest and saw that the score improved, after that the results were put in and tested it working with 3 hidden layers + 1 output layer and without the ASK\_AMT and APPLICATION\_TYPE variable
  + The results improved slightly but not too much
* In the third change – epoch was increased to be 500 times, it was found that changing epoch did not impact the performance as the results were stable after 100 epochs around 73% accuracy
* In the fourth change – the bucket of binarization was also tried to be changed to see if it caused any difference with CLASSIFICATIONO variable – it was found that there was no difference. Also looked at USE\_CASE variable to see if that could be bucketed and similar results were seen there – also made stratification to be enabled.

**Summary:**

The deep learning results were somewhat close to random forest results – it was found that some of the epochs – the results had better accuracy than 75% and it was an improvement from the starting bit where the values were around 60% to start off with. All over the results were promising but other ML algorithms like RandomForest performed just about equally and with much better pace – had slightly lesser accuracy. Other algorithms like XGBoost might have also performed in par with RandomForest but with more time, the accuracy would have been closer to 75% or even over 75%. Stratification improved the results and so did another hidden layer.

Using a GRIDSearchCV might have further improved and helped with understanding the hidden layers and neaurons but various different sizes of neurons and layers were tried – it was found that 2-3 hidden layers gave the accuracy and after that model was overfitting and causing the results to be lesser accurate.

There were bunch of outliers and that might have skewed the results as well – it would have been good to analyse those IQR for each variable and figure out if some of them were causing the ML model to perform incorrectly – that would have also improved the results.