**AlphabetSoup Campaign Success Modeling Report – Neural Network Modeling and Prediction Success Rate**

**Overview:**

In response to the request from AlphabetSoup to investigate and create a machine learning model using deep neural networks to predict if an applicants’ campaigns will be successful if funded by AlphabetSoup.

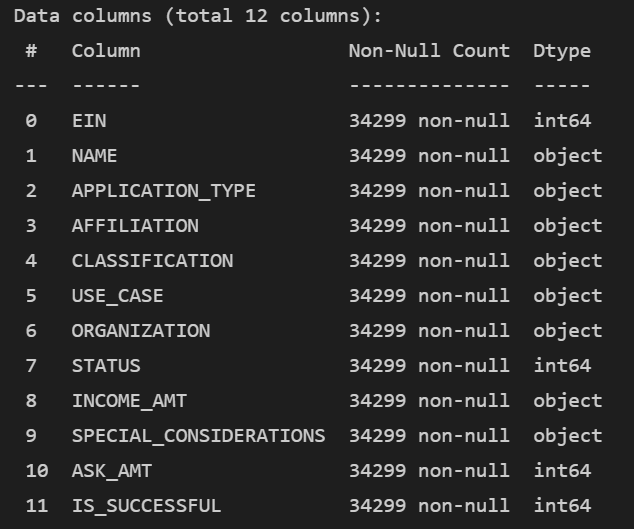
AlphabetSoup has provided historical data of ~34,000 applicants/organizations that received funding and if the campaign was “Successful”. This data will be leveraged though various processing, modeling and optimization tools to drive to the most accurate modeling of success rates possible with a goal of 75% or more accuracy. The data, tools and outcomes are captured below within the Results and Summary details.

**Results:**

**Data Preprocessing:**

* Initial review of the data provided indicated that the target for the model was if the applicant/campaign successful, using the “IS\_SUCCESSFUL” column (see Fig. 1).
* Two variables or data points in the data set were identified as not being meaningful to the model outcome, “EIN” and “NAME”; these are strictly identifying names/identification numbers of the applicant (see Fig. 1).
* The remaining features in the data set were used to in the model (see Fig. 1):
  + 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

**Fig. 1** - Data Columns and Data Types form the AlphabetSoup Historical Data:



After the target variable were identified and the identification columns (EIN/NAME) were removed, the unique counts of each data variable were reviewed to identify variables that have 10 or more unique values and further determining the number of each of the unique variable types (see Fig. 2).

Fig2 – Unique Feature types by Feature:

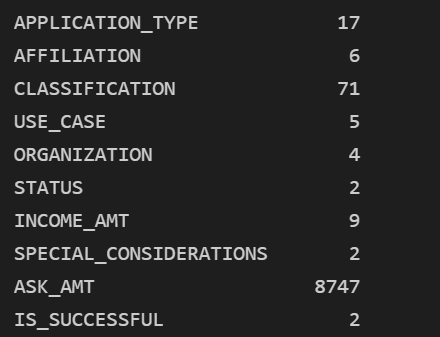
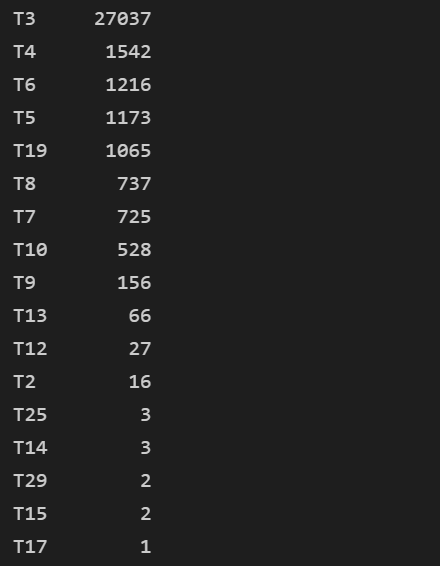
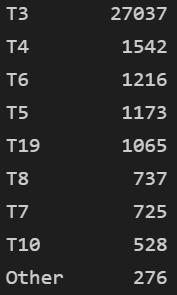


Fig. 3 – Unique Application Types:



The application type was reviewed with 17 application types. Any type with a count less than 500 our approximately half of the application types but very view of the actual applicants were in these types. These applicants were all grouped as “other” to eliminate potential errors our impacts form these outliers or dilution.

Fig. 4 – Unique Application Types after Binning



Next the Classification feature was reviewed for potential feature reduction for the same reasons as above.

Fig. 5 – Unique Classification counts > 1 (71 unique classification exist)

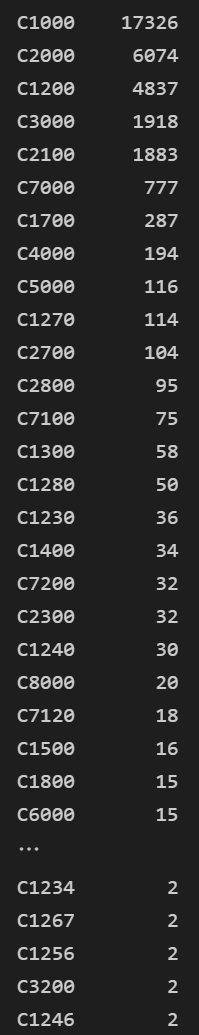
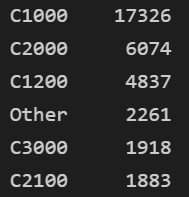


Fig 6. – Unique Classification Counts after Binning

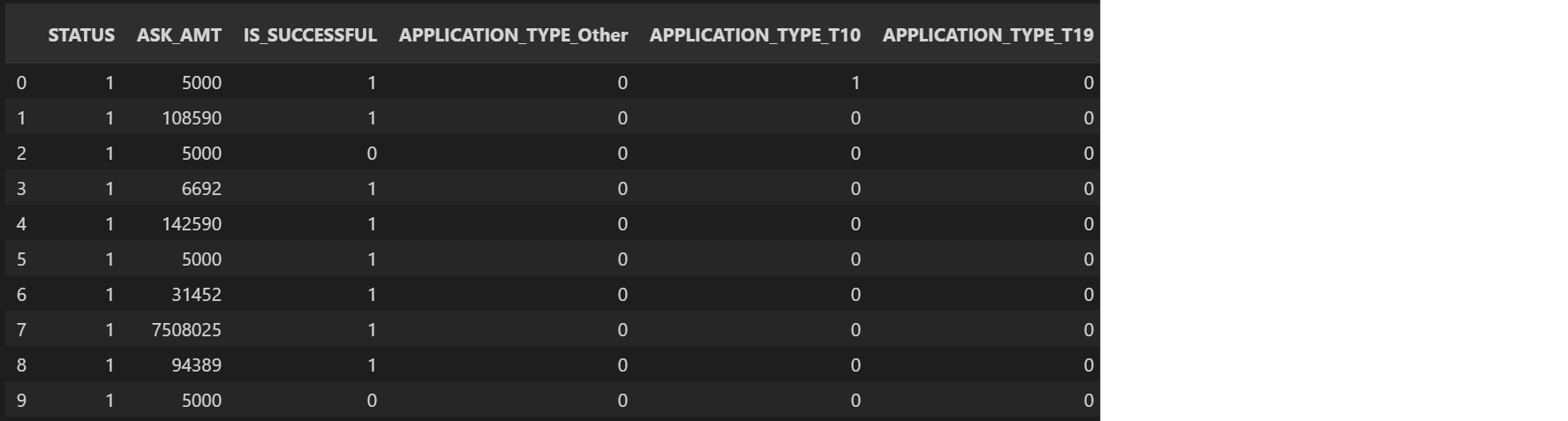


By leveraging binning, the classification feature has been reduced from 71 classifications to 6 as seen in Fig. 6.

While the Ask Amount for each applicant is a feature with greater than 10 unique values, it was not binned for the initial modelling as this may be viewed as a key feature in determining outcome.

To continue the data pre-processing from the data provided by AlphabetSoup, a programing feature called “GetDummies” was utilized to convert all categorical data to numeric data. This step is crucial in the process. Machine learning models require numeric data explicitly to predict outcomes.

Fig. 7 – Example of Categorical data converted to Numerical Data

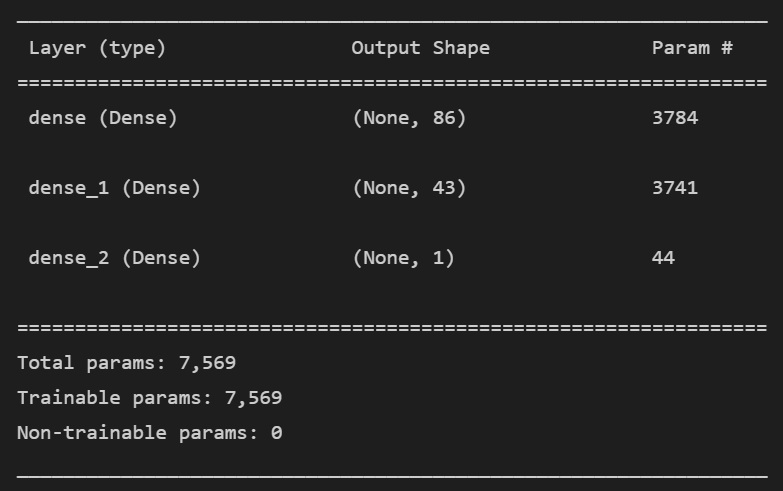


The last steps in the data pre preprocessing of the data is to split the data into testing and training sets for training and testing of the model to determine the accuracy using the data as process. Additionally, the data was scaled. Scaling of data is a key step in assuring that large variance in data feature values does not adversely influence the model by making all values fall within a similar scale.

**Compiling, Training and Evaluating the model:**

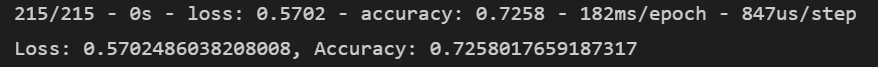
A neural network model uses a predetermined number of nodes or layers to evaluate or model outcomes based on the historical training data. Two input layers were leveraged in the initial model. The first input layer is using 86 nodes or twice the input features and the second node uses the number of features or 43. Lastly the number of training cycles or “Epochs” is set to 100. The neural network will begin training and weighting each feature within the epochs to determine which parameters across the features will predict more accurately the target outcome of the success of the campaign.

Fig. 8 – Initial parameters the model with train itself with:



As the Model Runs, each Epoch will determine the loss and accuracy of the parameters in the epoch to assist in defining the most accurate model. Lower Loss and Higher Accuracy is always the intended target.

Fig 9. – Initial Model Accuracy before optimization:



The initial model demonstrated .57 loss and .72 accuracy after the initial training. This is short of our goal of .75 or 75% accuracy in predicting the success of an applicant/campaign funded by Alphabet Soup.