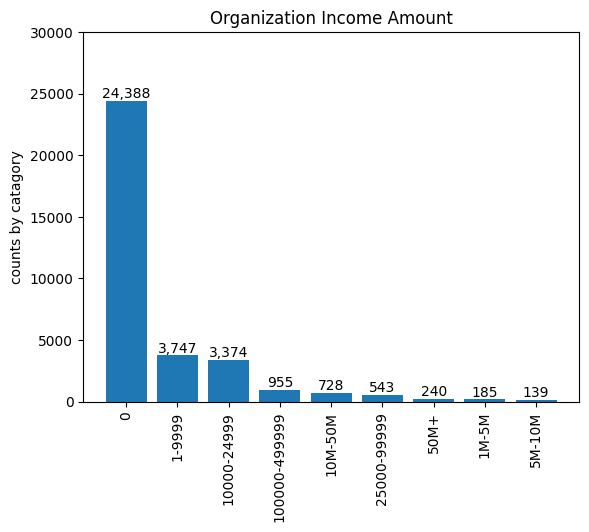
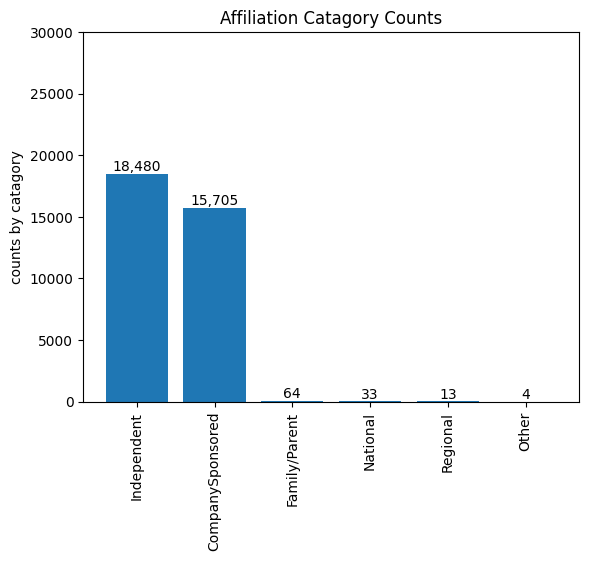
Introduction

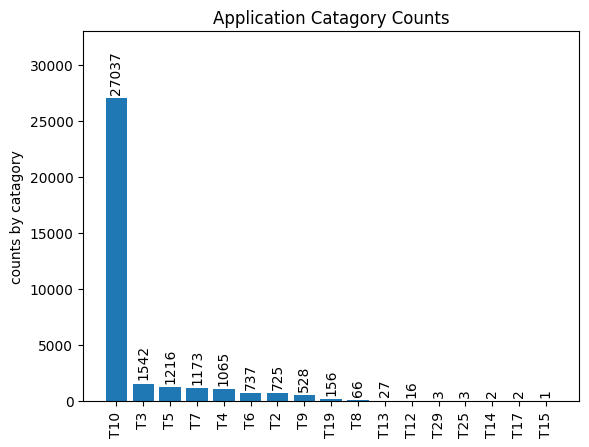
This analysis is of the Alphabet Soup funding data set. This has over 19568 organization (see cell 3 of the model, where the unique values are), these applications fall into 17 types and 71 classifications. The organizations are listed by affiliation and the use cases for the money. The amount requested ranges from 0 to 50 million. Of the records 24,388 of the 34,296 (that's 71%) list 0 income, this is an imbalance in the data set.



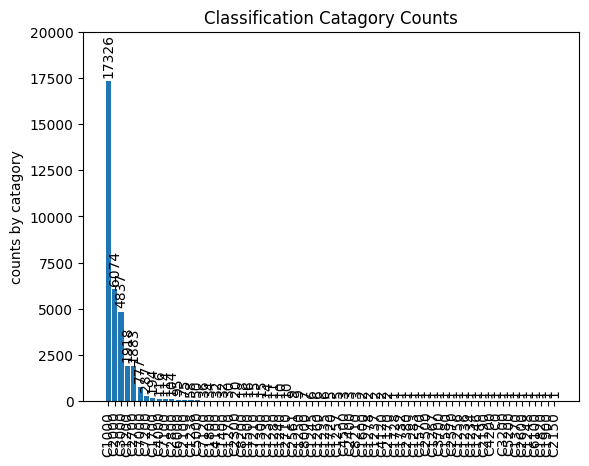
When looking at the 'affiliation' of the organizations, we see that 18,480 are 'independent', 15,705 are 'company sponsored' and the remaining 114 are either 'family/parent', 'national', 'regional', or 'other' (for exact breakdown, see Data Ex file).



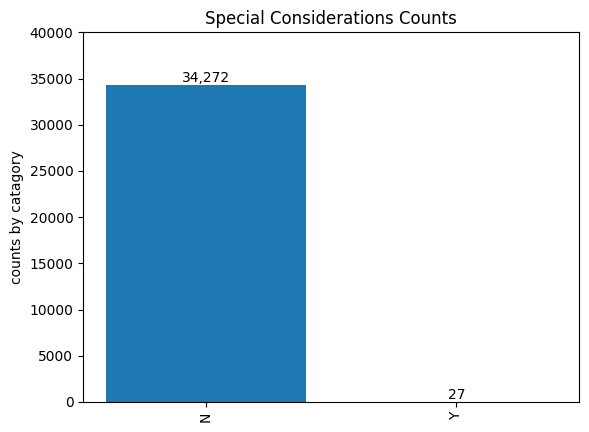
In the type of application, we see that 27,037 are category T10, with the next largest category having 1,542 and combining that with the other 3 counts for places 2-5 we get only 4,996. The next 'level' in the counts is below 1,000, with 2 categories having values in the 700s, one in the 500s, one in the 100s and 8 below 100.



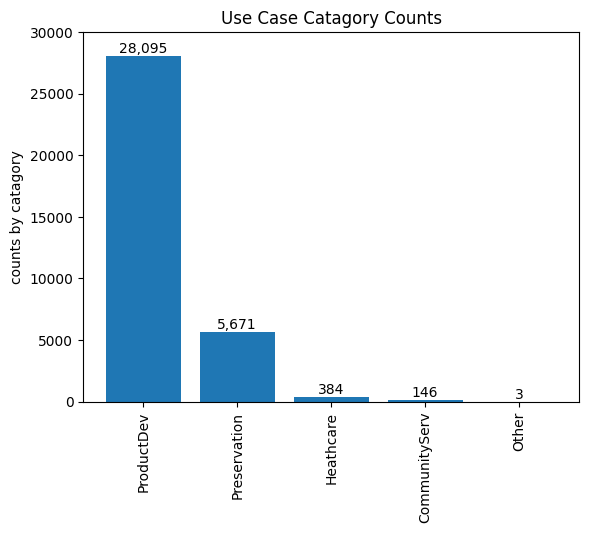
In the affiliation category the top 2 of the 6 categories made up 18,480 and 15,705 of the 34,299 records. The application classifications had 71 types and the graph looks like a classic Pareto distribution.



The special consideration category had only 27 applications that merited it.

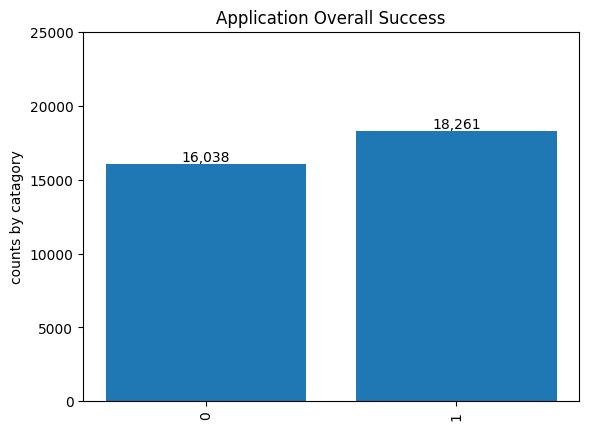


For Use Cases 28,095 of the 34,299 records were for Product Development, with the next highest category being for Preservation (5,671). The graph of the records also has the same uneven distribution.



For types of organizations the 'association' category was the largest covering 23,515 of the records. The status and special considerations are binary categories that have almost all their data in one of the categories, with the other having 5 and 27 records respectively.

For the project success 16,038 were 0, and 18,261 were 1.

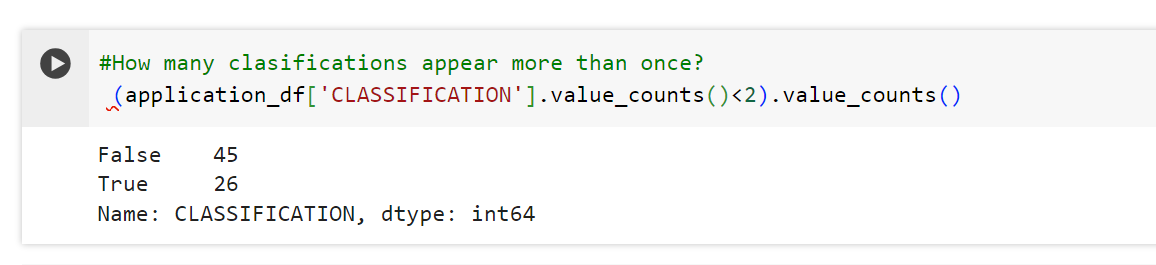


Model 1

The Data was read into a Google Colab notebook and the EIN and organization name columns were dropped form the first model. This was because these are the identification columns and biasing the model against specific groups is \*not ideal\* (to put it mildly). The data was examined (numbers of unique values per column, data type, descriptions of how the data fell in categories).

The application types were counted by category (add graph) and there was a very obvious distribution. The categories that had fewer than 500 total results were grouped into the category 'other' (276 results) in the hope that this might make a more meaningful grouping.

The classifications have 71 categories that are distributed in a very uneven manner. Of the 71 categories about 45 have more than 1 record. Groupings with less than 550 records were reduced to the category of 'other'. This reduced the number of categories to 7, with the other category now containing 1484 results, or the second lowest count. Hopefully this will produce a more meaningful grouping for machine learning.



All this summarization meant that going into the learning, all but 2 columns had been reduced to less than 10 categories each.

The categories were converted into binary and the column that measures success was put into the y value then removed from the data, so that it could be the answer key for testing and training. The rest of the values were set to X. The data was then split into testing and training data, then scaled and fit to the model.

The model was set up with 80, 30 and 1 neurons in the three layers. The activation functions were relu for the first layer and then sigmoid for the other two. The model was compiled and then trained for 50 epochs. At the end of this the accuracy was 0.7343, and the loss was 0.5465. The model was saved.

Model 2- Adding back the organization names

Having the suggestion that the model would respond to organization names as a parameter I got a count of the number of times an organization's name showed up in the data and then did an initial grouping of all organizations which appeared less than 500 times, which made it far and away the largest category in that column (31,574 of the total 34,299), the next largest had only 1,260 records. The model was rerun and the new values for the loss and accuracy were 0.5361, and 0.7352 respectively.

Model version 3

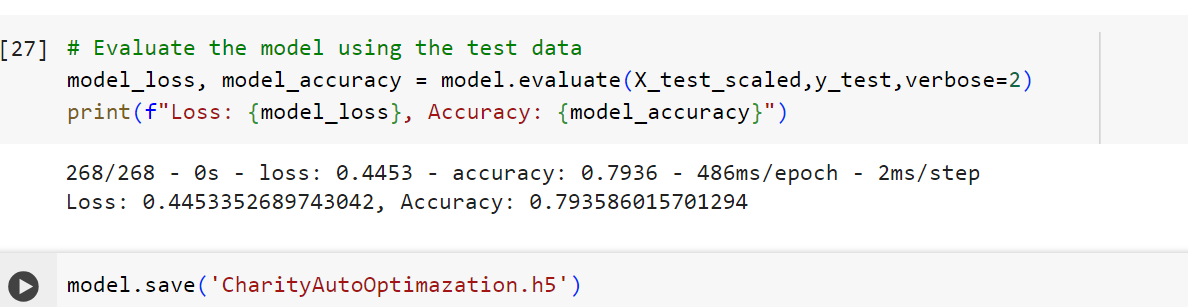
After playing with the numbers in the organization name category a bit, grouping organizations that had less than a count of 6 resulted in a loss of 0.4421 and an accuracy of 0.7934. This resulted in the organization name category being reduced to 355 from 19,568.

Model version 4

I attempted an auto-optimization of the network hidden layers without the organization name column. The best result That I got was loss 0.5487, accuracy 0.7380 with 5 hidden layers.

Model version 5

I reintroduced the Organization name, grouped all the organizations that had less than 5 applications and reran the auto-optimization. This resulted in an accuracy of 0.7936, and a loss of 0.4453 with 3 hidden layers.



Conclusion

The attempts at optimization without taking the organization into account were not as successful as they could have been. The data has several categories that are much more likely than others, however the overall success of the applications is around 53%. Altering the size of categories was not the approach that this sort of data needed.

The introduction of classifying if the organization had significant experience with the process (by grouping the organization names) was the most effective method of improving model performance. There were efficiency gains in neural network structure with the auto-optimization (model 4), but to truly see a performance gain, that needed to be combined with the information about the organization.

This result suggests that there may be a way to improve project organization success though mentorship, or some kind of help getting the projects organized in a way that increases their success. The next step might be to collect data about how projects were run and see if we can correlate that to success.