DATA MINING CHARITIES

CSC558 – Assignment5

Abstract

This document details my process of finding, cleaning, and planning a project around a dataset pertaining to charities in the United States and how reputable they may be based on their earnings and their spending habits.

Tyler Stoney

tston529@live.kutztown.edu

4.2.a - Source

The data set was found on Kaggle at

https://www.kaggle.com/cyaris/charities-in-the-united-states/kernels

while the data contained in the set was scraped from the site

https://www.charitynavigator.org/

This aggregate site collects and compiles its data directly from each charity's website.

4.2.b - Goals

By analyzing this data, I hope to determine a relation between the charity's type or category and the likelihood to which the charity will be worth donating. I will be starting fresh analysis on this data, however, for part of my analyses, I will be using a charity's "rating," a metric assigned to each charity by the aggregate site which initially compiled the list. The overall rating is a factor of two sub-rating scores, rating the charity by a factor of expenses vs contributions, and the site's assessment of the charity's accountability and transparency.

On top of that, other intriguing correlations come to mind. I would like to find any correlations between the location of the charity and its reputability, if they exist. Could the possibility exist that certain states have laws pertaining to charities that are more exploitable, and as a result, the charity business flourishes?

4.2.c - Pre-Production

4.2.c.1 – Finding a Set

Choosing a dataset consumed a considerable amount of time. Initially, I had been looking for a set for which no solutions had been posted, since my confidence in finding a project was initially low, and piggybacking on another's work seemed more complicated than starting from scratch. I had found a set on credit card fraud and factors that play into detection of it, and as I began to get comfortable with the idea of working with this set, the thought that its data was made up struck me: the set's name began with "Abstract data set" and the person who had submitted it provided no description of the set, nor any sources from which the data may have been pulled, so I panicked and backed out; the hunt was back on.

I stumbled upon another newly submitted set, this time, on charities in the United States, and after noticing a happy lack of solutions and the existence of a veritable source, I jumped on the opportunity to work on it. It was time to clean the data.

4.2.c.2 – Cleaning the Data

4.2.c.2.a – Cleaning in Python

For the most part, the data was readable, self-explanatory, and as far as variability of strings is concerned, consistently written. Despite my laud towards the quality of the web scrape, there was still a considerable amount of tidying up to do. The most immediately glaring issue was with strings. While ultimately this might not pose a problem in terms of number crunching, many cells had several extra characters (a quotation mark, several newline escape characters, and varying amounts of tab escape

characters) prepending them. The organization type, another string column, had helpful data to a certain point: initially, each cell was given a broad category, then a short description of the charity's work, delimited by a colon. I saw an opportunity in this column, in that I can look at charity reputations and their types, and draw conclusions as to which types of charities are most work donating. For this, I decided to remove the description of the organization, leaving only the general type ("Religion," "Heath," etc.) Python and a csv manipulation module named pandas made quick work of these modifications.

Numbers often prove more important than strings in data analysis, however, so my cleaning extended to columns regarding compensation and expense percentages, in which more escape characters were jumbled in on either side of the numbers, and in cases of what are supposed to be blank fields, the curious string "=--\r\n" filled the cell. A column containing percentages was formatted as a string (%5.06) so I decided to remove the extra character, and turn it to a workable number (0.0506). Python and pandas again came to the rescue.

4.2.c.2.b – Cleaning in Excel

Microsoft Excel could handle the rest of the cleaning from here: I deleted three columns, as they would not help at all: the cities, as I had enough location data from the state attribute (plus this could be turned into a nominal attribute), the column containing the charities' URLs, and the column in which the scraped site in question had posited an advisory on some charities (this column was, from a quick skim over the spreadsheet, left entirely blank except for one charity).

Cleaning the remaining data was a matter of removing formatting for columns. I would rather work with numbers as stale, lifeless looking digits (à la, 12345) rather than a formatted string version ($12,345), and so Excel proved its worth again. The final touch was to turn any instances of the string "Not reported" to the number -1, as these strings were in a column otherwise populated by numbers, and positive numbers, at that. By doing this, when the data is imported into Weka, the black sheep of the attribute will be the only negative number, making it easy for a human to re-interpret into its original meaning.

4.2.c.2.c – Moving to Weka

I knew I wanted to make the charities' organization types the target, so I shifted that column to the end.

I slightly modified a csv-to-arff converter script found on GitHub to handle string values. Weka's built-in converter didn't appreciate my dataset, so I opted to do it in this more "hands-on" way.

Running it through the converter and attempting to open the new file as-is posed some issues, as well. Weka threw several errors, mostly due to the nature of some of the string-based values. Weka seems to parse data with spaces, regardless of whether it is used in a string, as if they are two separate values, which throws off the balance of the rest of the arff file. It was here that I decided to throw away the columns "compensation_leader_title" and "charity_name," to aid in creation of a successful file, with the added thought that the columns wouldn't be of any use either way. The leader's title column had too much variability in the way the strings were named and the way the titles were distributed (some columns were listed "CEO," others "Chief Executive Officer") and many had more than one title assigned to it. The column containing the percent of their spending should suffice. The charity name, like the title column, had too much variability, since the nature of the column only allowed unique objects.

I defined the state column as a string, as Weka does not seem to like the way this Python converter interprets nominal values, but that was not a huge issue, since Weka has a filter to convert into nominals from a string, so I opted to take that route, and named it a string in the context of the Python script.

4.2.c.3 – Final Data

I will be using any combination of the following attributes to perform my analysis:

Attribute Name	Type	Attribute Description
accountability_score	Numeric	A metric assigned by the aggregate charity site mostly based on the charity's public Form 990
administrative_expenses	Numeric	Cost of general legal services, accounting, office management, human resources, etc.
compensation_leader_compensation	Numeric	Amount paid to a charity's president or CEO
compensation_leader_expense_percent	Numeric	Percent of total expenses the CEO/president's salary represents
excess_or_deficit_for_year	Numeric	The difference between a charity's total revenue and its total expenses.
financial_score	Numeric	A metric assigned by the aggregate charity site, based on their financial performance
fundraising_expenses	Numeric	This measure reflects what a charity spends to raise money.
net_assets	Numeric	The difference between a charity's assets and its liabilities.
total_contributions	Numeric	Amount contributed towards the charity's cause
other_revenue	Numeric	Revenue from investments, rents, special events, sales of inventory, and other unrelated business income.
overall_score	Numeric	Derived attribute, distance between the other two scoring metrics
payments_to_affiliates	Numeric	Amount the charity pays to any companies affiliated with it
program_expenses	Numeric	This measure reflects the percent of its total expenses a charity spends on the programs and services it exists to deliver.
state	Nominal	State in which the charity is based
organization_type	Target - Nominal	Target attribute describing, in general, what the organization funds

4.2.d - Who Could Use This?

Charities are first and foremost businesses, and it is a stereotype of this industry that they pay themselves first, and put their intended cause on the backburner. Armed with the information from this research, donors with deep pockets can make a more informed decision as to whether a charity is a

worthwhile investment. Businesses may use charity donations as a means of benefitting from a tax write-off; it would look better for the face of the company in question if the charities to which they donate were proven to be reputable, or had a high likelihood of being reputable.

Should this type of research become more known, charities themselves may consider modifying their business model, or at least becoming more transparent with their funds.

Since this dataset contains locations for the charities, another potential use for this would be to find a correlation between the state in which a charity is based and its net earnings.

As mentioned in the goals section, perhaps there is a correlation between charity reputation and the state in which it is based. If such a correlation is strong enough, there may be a missing link not covered by this dataset: perhaps a law in that state provides a loophole for charities to act not in accordance with their stated purpose.

4.2.e - Prospective Methods

4.2.e.1 - Techniques

Since this dataset focuses primarily on monetary values, most of the calculations will be mathematical, rather than some form of string frequency, so to make heads or tails of the source site's scoring system, I will be performing calculations on the monetary data, but will eventually move to classifying the data based on the nominal attributes I had cleaned at the beginning – the charity's state and organization types. It is the classification on the organization type that I believe to be the most important facet of this project; since money spent vs money earned can be turned into a percentage, the real classifier may be what the organization purports to serve.

4.2.e.2 - Tools

I intend to use Weka for most of my classifications. Weka will allow me to easily categorize attributes by placing them in bins, with each bin representing a fixed range on that attribute's data. For extra analysis, I may write some Python scripts to classify, or rather, quickly calculate statistics in relation to some nominal attributes.

4.2.f – Anything Else?

The majority of the project's preliminary stage is laid out in the previous sections, so no further information needs to be relayed.

5.2.a – Additional Data

No additional data was collected for this part of the analysis.

5.2.b – Results?

I hate to end any experiment with an "inconclusive" conclusion, but given my hours of analysis and contemplation, it certainly seems like very little, if any, correlations can be drawn on classifying charity types with their reputability. Overall, it seems that charities are too independent to stereotype. In other words, no, I did not achieve the results I had anticipated, but this could be a positive outcome, considering the business type. Since charities stand to make money for a cause, and since no evident correlation between charity cause and reputation exist, we can optimistically say that the charities with a low score are just a few "bad eggs" in the bunch and that their performance is not representative of the cause for which their organization stands.

The best classification I came across was through a decision tree's classification after running it through a service in Weka which ranks attributes by predictability; after using the analysis from this, my best classification correctly identifies just over 39% of the attributes.

From this part of my analysis, the best attribute to predict the organization's type is its payments to its affiliates, with a predictability of about 20%, not-quite twice as accurate as a random guess.

Part two of my analysis proved a bit more useful in terms of learning, rather than pure predictions. Once I removed the redundant attributes from the set and targeted the overall score, the classifiers had to analyze each of the attributes more closely. In both sets of tests before and after removing redundant data, a crucial element in classifying the overall score, and therefore the reputation of each charity, was how much money (as a percent of the charity's total funds) is used to pay the head of the organization (CEO, president, etc.). I did not expect to find an attribute as predictive as this, given the state of my earlier analyses.

5.2.c – Steps to Classification

Initially, I went in to the project quite optimistically, hoping to run a few classifiers and see the high correlations they would spit out. This turned out to be far from what happened. J48 tree was initially my "go-to" classifier in this project, since within the data, there are cases of partially redundant data- the final assigned score for each charity is obtained by measuring the distance between the financial and accountability scores – and trees generally perform well under those circumstances. While the final score is not as close to either as, say, a simple average of the two, there is still some distant correlation among the three.

Classifying by Organization Type

Running J48 on my desired target attribute (the charity's type) yielded poor results. A ten-fold cross-validation correctly classified just over 25% of the attributes with a kappa statistic of 0.1207, while training the set on only about 1,200 (running a 15% split on the data) correctly classified 22%, yielding a kappa statistic of 0.0888:

10-fold cross:

Correctly Classified Instances	2055	25.0916 %

Incorrectly Classified Instances	6135	74.9084 %
Kappa statistic	0.1207	
Mean absolute error	0.1402	
Root mean squared error	0.3377	
Relative absolute error	90.1581 %	
Root relative squared error	121.0761 %	
Total Number of Instances	8190	

15% split:

Correctly Classified Instances	1535	22.0514 %
Incorrectly Classified Instances	5426	77.9486 %
Kappa statistic	0.0888	
Mean absolute error	0.1456	
Root mean squared error	0.344	
Relative absolute error	93.6363 %	
Root relative squared error	123.3029 %	
Total Number of Instances	6961	

There are eleven possible targets to choose from, so this roughly 25% accuracy is 2.5 to 3 times as accurate as a random guess. This was not predictive enough for my tastes so I moved on to other classification methods.

The nature of this dataset dictates that each instance is independent of one another, making methods other than trees, most notably any type of nearest-neighbor algorithm – kstar, IBk, etc. - potentially perform a worse under identical circumstances. My suspicions were confirmed when running IBk under several different trials yielded an accuracy of 20.94% on a 10-fold cross-validation with a kappa statistic of 0.0727:

Correctly Classified Instances	1715	20.9402 %
Incorrectly Classified Instances	6475	79.0598 %
Kappa statistic	0.0727	
Mean absolute error	0.1438	
Root mean squared error	0.3789	
Relative absolute error	92.4263 %	
Root relative squared error	135.8504 %	
Total Number of Instances	8190	

This dataset was not looking performant at all.

At this point, I needed to determine which attributes were performing the best, if any. Weka has a feature in the tab labeled “Select Attributes,” which analyzes the attributes and how much they can potentially contribute to classifying the target attribute. There were two cases I considered here: ranks in relation to my default target attribute (organization_type) and those in relation to the overall_score attributed to that instance. Under several search methods and evaluator combinations, targeting the default target attribute with all the attributes intact showed that the state was often one of, if not the highest factor in predictability. This was interesting, as the other metrics that were used to create the three scores were mostly numerical; the state should have had nothing to do with how the

organizations were ranked. If this was throwing off predictability, it would certainly help to throw it out, then. Most of the remaining tests I performed did not include the state attribute.

I swapped to looking at this project from another point of view: if the score is not easily used to predict the organization's type, then how predictive would the organization's type be in predicting the score? Ranker needs the target attribute to be nominal, so I discretized the organization's score into 20 bins, so each bin now has a range of about 1.7 score points, from the highest to the lowest-scored charity. Running the ranker showed that the organization type is only about 7% predictive. This is what helped start to form my conclusion that the charity's type is not easily predicted through its performance.

Running J48 again after removing the state showed some increase in performance, correctly classifying about 27% of the instances:

Correctly Classified Instances	2199	26.8498 %
Incorrectly Classified Instances	5991	73.1502 %
Kappa statistic	0.1412	
Mean absolute error	0.1361	
Root mean squared error	0.3372	
Relative absolute error	87.499 %	
Root relative squared error	120.905 %	
Total Number of Instances	8190	

Coming back to the idea that there might still be more attributes that are hindering performance, I once again turned to the "Select Attributes" tab, and started playing around with classifiers after removing attributes that were deemed to be less predictive. J48 did degrade, but only marginally, even after removing five attributes: program expenses, total_contributions, fundraising_expenses, financial_score, and excess_or_deficit_for_year – all attributes deemed less predictive to the computer, but attributes that I would have imagined to be important:

Correctly Classified Instances	2081	25.409 %
Incorrectly Classified Instances	6109	74.591 %
Kappa statistic	0.1223	
Mean absolute error	0.1391	
Root mean squared error	0.3337	
Relative absolute error	89.4094 %	
Root relative squared error	119.6762 %	
Total Number of Instances	8190	

Rather than stopping here and deeming this dataset completely unusable, I gave it one last shot, still working on my hunch that there is some combination of attributes that is hindering the performance of my classifiers. Fortunately, Weka has a feature under the classifiers which lets the user perform a classification after piping in the results of an attribute evaluation; Weka weights the classifications according to the importance of these attributes, sometimes removing the attribute or attributes which could be a hindrance to the result. I ran BayesNet out of curiosity and right off the bat, there was a slight difference in accuracy; this outperformed my highest classifer by 7% on a 66% train/test split:

Ranked attributes:
0.2081 12 payments_to_affiliates
0.0415 1 accountability_score

0.0386 9 total_contributions
0.0356 8 net_assets
0.0343 11 overall_score
0.0323 5 excess_or_deficit_for_year
0.0319 3 compensation_leader_compensation
0.0303 10 other_revenue
0.0258 13 program_expenses
0.0245 2 administrative_expenses
0.0206 4 compensation_leader_expense_percent
0.0167 6 financial_score
0.013 7 fundraising_expenses

Correctly Classified Instances	910	32.675 %
Incorrectly Classified Instances	1875	67.325 %
Kappa statistic	0.1623	
Mean absolute error	0.1425	
Root mean squared error	0.277	
Relative absolute error	91.6 %	
Root relative squared error	99.272 %	
Total Number of Instances	2785	

J48 performed slightly better than its previous run. Running Random Forest in this manner, however, gave the best results of all my analyses, along with a temporary header applied to the arff file before the final prediction is given (it gave the same table for ranked attributes as above):

Header of reduced data:
@relation 'cf-weka.filters.unsupervised.attribute.StringToNominal-R14-weka.filters.unsupervised.attribute.Remove-R14-weka.filters.unsupervised.attribute.Remove-V-R12,1,9,8,11,5,3,10,13,2,4,6-7,14'

=== Summary ===

Correctly Classified Instances	3201	39.0842 %
Incorrectly Classified Instances	4989	60.9158 %
Kappa statistic	0.2321	
Mean absolute error	0.1384	
Root mean squared error	0.2628	
Relative absolute error	88.9514 %	
Root relative squared error	94.2286 %	
Total Number of Instances	8190	

In previous assignments, Random Forest seemed to perform well for me often, though the datasets provided for those were guaranteed to yield good results most of the time. Why the huge discrepancy between this score and the original data? Random Forest is a bootstrap aggregator method, in that it generates a bunch of trees by resampling data and spitting out the best result, but the “Forest” in its name comes from the fact that it does this repeatedly and averages the most predictive trees’ results. This, coupled with the weights it assigns to specific attributes, seems to explain why this outperformed

every classification I ran thus far. To compare it to the likelihood of a random guess again: this result is over four times as accurate as guessing.

Classifying by Overall Score

Once concluding that the organization type is not as easily stereotyped as I had expected, I attempted to classify the data on its overall score, rather than by its organization type. Running M5P immediately proved useful: it correctly classified 99.97% of its target attributes, and showed which attributes were used to get this astounding correlation.

```
LM num: 9
overall_score =
        0.4161 * accountability_score
        + 0 * administrative_expenses
        + 0.9783 * compensation_leader_expense_percent
        - 0 * excess_or_deficit_for_year
        + 0.5826 * financial_score
        + 0 * fundraising_expenses
        + 0 * net_assets
        - 0 * total_contributions
        + 0 * program_expenses
        + 0.1046
```

Above is an example of a leaf from the tree; scanning over the leaves it produces, it seems like the only attributes used are accountability score (which, despite this outlier example, is almost always has the highest coefficient). The tree itself, however, starts the branches with its financial score, then dips into the accountability score. I had anticipated this, since the overall score is, again, just the distance formula run on these two scores. Despite this obvious correlation, I did obtain the unexpected tidbit that the overall score can be classified by the organization leader's compensation, as a percentage of their organization's earnings.

Removing these two redundant scores should then force the classifiers to analyze the data themselves. By doing this, my correctly classified instances plummeted to 64%, but that is still relatively passable, compared to analysis on my previous target.

Correlation coefficient	0.6448
Mean absolute error	4.427
Root mean squared error	6.5623
Relative absolute error	75.1852 %
Root relative squared error	78.5744 %
Total Number of Instances	8190

The leaves and tree this time were radically different, as expected. The organization type played a role in predicting each score, but usually they were worth less than a percent per classification. The attribute that stood out, yet again, was the leader's compensation. The tree started each branch by looking at net assets, then went into program expenses and depending on the situation, re-looked at net assets, or looked at fundraising expenses or for the minute frequency that it occurred, the organization

type. The thing to take away from this is that the leader's compensation plays a role in determining the reputation of the organization.

5.2d – New Technique

As I had talked about in the last section, I had used the "Select Attributes" tab and the search methods and attribute evaluators inside it to give me an estimate as to how predictive each attribute may be. This proved useful, as it showed that for this dataset the state had a higher influence on the result than the organization type, and often, the majority of the score metrics. Removing this attribute boosted the classifications, if only by two percent.

The principles of this technique were applied to my classifications automatically by Weka, thanks to a classifier named "AttributeSelectedClassifier" which performs the same attribute-level analysis before running the classifier of the user's choice on the new weighted dataset, giving results more in-line with the importance of each attribute. This caused a jump of 14% in my correctly classified attributes.

5.2.e – Research for Commercial Purposes

Since this analysis was not as predictive as I had anticipated, the results might not bear much weight in any kind of research. I had suggested previously that perhaps the fact that charities cannot be classified into tidy bins is a good thing for the business. It suggests that, for the most part, they are doing what they claim to do, and that no classification can outright say "X and Y charities shouldn't be given donations, since they are an A and B-style charity." Perhaps if more research is to be done on this topic, one might consider putting less of a focus on the organization type, and look to classify charities for their financial metrics.

5.2.f – Anything Else?

Once again, everything I needed to say has been said, no further information about this project is required.