CSE 471 Project Writeup
Team Name

David Ganey Kerry Martin Evan Stoll Michael Theut
Ben Roos

April 7, 2015

Contents

List of Figures

List of Tables

1 Part 1

2 Part 2

Part 2 of the project describes another task given by the CEO of the CactusCard Credit Company, wherein a program should be developed to provide "a *machine learning based* *approach* to identify individuals who should/should not be given the CactusCardPlus." This section will formulate this problem and describe the steps taken to solve it.

2.1 Problem Formulation

The problem itself is a machine-learning problem, which at its core simply means a program designed to recognize patterns. Machine learning can encompass a wide variety of problems, which can be divided into "regression" problems and "classification" problems. In this case, the consumer research division has given the team a dataset which contains information about customers *for whom the decision to award or not to award the CactusCard has already been made.* This is important information – because customers can either be given the card or not (a binary *classification*), this falls under the realm of "classification" machine learning problems. Additionally, the fact that the classification program will be trained with data means this is a "supervised" learning problem.

As a supervised classification machine, the program must be able to read the data set and, with that information, draw conclusions about which credit variables have the largest impact on whether or not the customer was given the card. To be an effective tool for the CactusCard corporation, the dataset supplied by the consumer research division should only include proper decisions – cases where the card was awarded to the correct individual and denied to individuals who would have abused the line of credit. Otherwise, the machine will learn "bad habits" and will draw conclusions from the data which are not useful to the company.

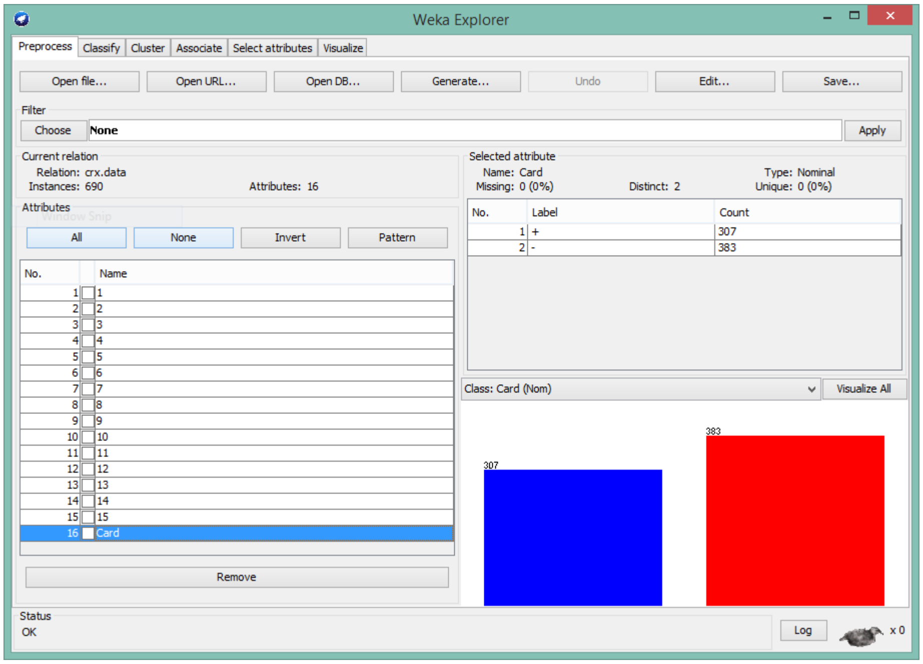
The dataset itself is comprised of 15 credit attributes for the individuals, as well as an indication of whether the individual received a card. The credit attributes are not specified, though the problem specification does include the datatypes for each attribute. (For example, attribute 1 can be either "b" or "a", and therefore represents some binary information about the customer such as whether they are male or female).

The goal of this component of the project is to demonstrate a program which can use the dataset to learn how the credit attributes affect the decision to award the card, and from there use that knowledge to predict card decisions within a certain degree of accuracy.

2.2 Solution Proposals

Weka, a software tool developed at the University of Waikato, is a software package with implementations of a large number of machine learning algorithms. Weka can be used in multiple ways. One option is to simply use the Weka API to access the algorithms from Java code. This is advantageous when one wishes to write a full machine learning system. Another option is to use the Weka Explorer, which is a wrapper around the algorithms. This GUI tool allows the user to load a data set and run the algorithms on it, then view the results in various formats.

It is this method which is proposed to solve the task given by the CEO. Using the Weka Explorer, we have a straightforward method for gaining insight into the effect of the 15 credit attributes. We can use the Explorer to test various machine learning algorithms, and based on their success rates, determine which should be used in the future to make the actual issue decision.



Filter: None

Current relation
Relation: crx.data
Instances: 690
Attributes: 16

Selected attribute
Name: Card
Type: Nominal
Missing: 0 (0%)
Distinct: 2
Unique: 0 (0%)

No.	Label	Count
1	+	307
2	-	383

Attributes

No.		Name
1	□	1
2	□	2
3	□	3
4	□	4
5	□	5
6	□	6
7	□	7
8	□	8
9	□	9
10	□	10
11	□	11
12	□	12
13	□	13
14	□	14
15	□	15
16	□	Card

Class: Card (Nom)

Status
OK

Figure 1: The Weka Explorer interface

2.3 Test Plan

The team will evaluate several popular machine learning algorithms in the Weka Explorer. The success of each algorithm will be determined by a number of factors. The team must consider runtime, as the algorithm cannot take an extreme amount of time to complete or it will be useless. The most important factor, however, is the accuracy of the classification. Using random chance to classify card issuance would theoretically have an accuracy rate of approximately 50%, so the team must find an algorithm with a higher percentage than that. To ensure CactusCard's continued financial success, the algorithm must predict card issuance correctly at least 80% of the time.

2.4 Solution 1

Support Vector Machines (SVM) are, according to Russell and Norvig, "the most popular approach for 'off-she-shelf' supervised learning"' [**?**]. SVMs do a good job of classification because they separate groups as much as possible (called a "maximum margin separator"). Additionally, SVMs can work in multiple dimensions (creating a "hyperplane") which will be important with the high number of credit attributes in the data set. Finally, SVMs are considered a nonparametric method, because they store some of the training examples, and yet like parametric models, they often avoid overfitting the data [**?**].

To use an SVM on the data set provided by the CEO, Weka supports the addition of a library called libsvm. This Java library contains an implementation of the SVM algorithm which hooks into the Weka Explorer interface. The team will set the Explorer to use varying numbers of crossfold validation passes, in order to evaluate the classifier. This will split the dataset, train the SVM with a subset of the dataset, and then evaluate the SVM's effectiveness at predicting the card issuance decision by comparing the SVM's predictions to the actual data. The results from running the libsvm algorithm on the dataset are shown in the table below:

As we can see from the table, no significant difference exists when adjusting the number of crossfold validation passes. All of the attempts fall well within the expected runtime, and predict with approximately 55% accuracy whether someone was or was not given a

Crossfold validations	5	10	15
Time taken to build model (s)	.19	.24	.19
Correctly classified (%)	382 (55.36%)	385 (55.80%)	384 (55.65%)
Incorrectly classified (%)	308 (44.64%)	305 (44.20%)	306 (44.35%)

Table 1: A table showing the results of the libsvm applied to the data set

CactusCard. While this is consistently better than randomly guessing, it only exceeds that percentage by approximately 5%. While SVMs are good classifiers, in this instance this algorithm is not a suitable choice for the company, as it fails the criteria established in Section 2.3.

2.5 Solution 2

Another type of machine learning algorithm is called a "multilayer perceptron." This is a type of neural network, which maps from inputs (in this case, the credit attributes) to outputs (whether someone was given a card). The MLP is suitable for this task because it uses "backpropagation," which is a supervised learning technique.

The multilayer perceptron is significantly more complex than the SVM, but Weka does include an implementation in the Explorer. Using the same technique as above, the team will evaluate this algorithm with various levels of crossfold validation to determine the optimal setting. The results are seen below:

Crossfold validations	5	10	15
Time taken to build model (s)	8.46	8.47	8.51
Correctly classified (%)	584 (84.64%)	572 (82.90%)	572 (82.90%)
Incorrectly classified (%)	106 (15.36%)	118 (17.10%)	118 (17.10%)

Table 2: A table showing the results of the multilayer perceptron applied to the data set

Once again, it is clear that adjusting the number of crossfold validations has little to no effect on the accuracy of the classifier. As a result, the recommendation of the team is to use 5 validations on data sets of this size, so the algorithm does not take too long to run. While the time taken to build the data model was always nearly the same, the total runtime for all passes of the validation increased dramatically at 15 validations.

Despite the lack of variability among the validation numbers, we can see immediately that the multilayer perceptron is a good choice for the CactusCard company. In all cases, the success of the classifier was above the criteria set above in Section 2.3. The MLP algorithm averaged approximately 83% correctness, which is significantly higher than random chance. This algorithm would be a good choice for future data sets, where it could use the information it learned from this training set to choose who receives a card.

3 Part 3

4 References

5 Appendices