James Cuzella
Justin Guerra
Data Mining
Project 5.1

Data Mining on Communities and Crime

Project Goals and Dataset

For our project, we decided to look at a dataset that contained census and crime data. Our goal was to try to find some model to predict whether or not any given community would have a low or high crime rate. The dataset was gathered from the University of California Irvine (UCI) Online Machine Learning Repository [1]. It was donated by Michael Redmond from La Salle University in Philadelphia, Pennsylvania, and contains aggregated data from the 1990 US Census, the 1995 US FBI Uniform Crime Report, and the 1990 US Law Enforcement Management and Administrative Statistics Survey. Due to the nature of this data, some have argued that such data could have been doctored or manipulated beforehand for political reasons [2]. However, our dataset's age is one factor that would make such manipulation useless for any current political motives. The other factor that makes this argument a non-issue is that all the fields have been normalized to values between 0 and 1. This preserves the rough ratios of the values **within** each attribute, however it does not preserve relationships **between** any other attributes [1]. Because the data has been processed in this way, it makes it easier to work on using attributes as discrete predictors of the crime rate, rather than giving the ability to look at other attributes and how they are related. The class variable of the dataset is "ViolentCrimesPerPop," this is defined as the per capita violent crimes. It was calculated using the community population and the sum of the number of crimes considered violent in the US: murder, rape, robbery, and assault [1]. One interesting thing to note about this is that there was some controversy in certain Midwestern cities concerning how rapes were counted, which led to some missing values and incorrect calculations of the class variable. Because of this, those cities were omitted from this dataset.

Preprocessing

After first gathering the CSV data and assembling it into a WEKA arff file, the first attempts at running a classifier on it were extremely unfruitful. After running the ZeroR classifier on the data, it predicted a value of 0.238 for the value in every case, resulting in a relative absolute error of 100%. In order to increase the accuracy of our classifier, it was apparent that a lot of preprocessing was necessary. The main reason that the classifier had a hard time is that the values for "ViolentCrimesPerPop" were normalized to from 0-1 with a mean of 0.238 and standard deviation of 0.233. Because the average value was 0.238, that's what the classifier ended up choosing as the best statistical prediction. The other issue with the data was that the first 5 fields were non-predictive. These fields were: state, county, community, communityname, and fold. Out of these, only fold and communityname were potentially useful for predictive purposes. However, because WEKA was able to perform its own data folding, we decided to get rid of the "fold" field. The next issue with the data was that many fields had missing values, which could result in a bad classifier model for certain communities where these data fields were unavailable. In order to remedy this problem, we first reduced the number of attributes of our dataset, removing all of the fields with missing values, as well as the non-predictive fields.

After removing all of these fields, the classifier still had a hard time predicting accurately due to the class variable having such a large range of values. To fix this, we decided to reduce the numeric field to a nominal binary field, having just values of 0 and 1. A “0” in this attribute would now mean that the original numeric value was <= 0.4, being considered a low crime rate, and a “1” would mean > 0.4 for a high crime rate. In order to perform this data processing, the data was first converted from comma separated values into a space separated value format so that awk scripts could be run. We managed to find an awk script that would do this, as well as convert the “ViolentCrimesPerPop” field to a two class variable based on our chosen threshold. Finally, fields that were detected as having missing values were removed, and the data was converted back into comma separated value (CSV) format. Afterwards, a diff on the removed fields had to be performed, and this list of removed fields was also removed from the arff header. The removed fields can be found in the “preprocessing-work/diff-work/” directory in our git repo. The awk scripts used can be found in the “preprocessing-work/” directory, along with other intermediate files created during the process.

Mining Strategy

When starting to explore our dataset, we were unsure of where to begin. We knew we wanted to classify data, but were unsure of what algorithms would be best. To find an optimal algorithm, we used the Weka experimenter extensively. As shown in the experiment files (experiment1-4.arff) a large variety of different algorithms were tested. Rule based, tree based, lazy, and Bayesian classifiers were experimented with. We did experiments with ANN’s, but abandoned that approach after running MultilayerPerceptron for 14 hours with no results. After finding the optimal classifiers this way, we then started to tweak the algorithms to see if we could get better results.

JRip

Rule based classifiers were very accurate on our data set, and JRip turned out to be the most accurate. JRip is a rule based classifier that has two main stages. The first stage builds up a rule set, while the second optimizes the rule set. JRip uses a greedy approach to add to antecedents until the rule is 100% correct. It tries every possible combination of attributes, until it finds the combination with the highest information gain. The last stage of the grow phase prunes the rules. JRip accomplishes optimizing by taking the rule set from the grow stage, and creating two variant rule sets. The early grow stages are applied to these variant rule sets. After these steps the rules that have the shortest length are chosen to be in the final rule set.

James Cuzella
Justin Guerra
Data Mining
Project 5.1

Without any tweaking JRip gave fairly accurate classification (~86.4%). We found that altering the number of folds led to higher accuracy and less error. The optimal number of folds turned out to be three. With three folds the accuracy increased to ~87.5 and the root relative square error fell by 2%. The number of optimizations did not have a significant impact. A single optimization was .1% more accurate than using two optimizations, but also had .6% higher root relative square error. Increasing the optimizations past two resulted in lower accuracy, and had no impact on the root relative square error. Disabling pruning led to longer rules (which was expected because the algorithm optimizes for short rules), but also led to lower accuracy.

Here are the rules the most optimal JRip provided:

```
 JRIP rules:
===========

(PctIlleg >= 0.35) and (PctKids2Par <= 0.34) => ViolentCrimesPerPop=1 (191.0/41.0)
(PctIlleg >= 0.29) and (HousVacant >= 0.09) and (FemalePctDiv >= 0.56) and (racePctHisp >= 0.06) and
(MalePctDivorce >= 0.66) => ViolentCrimesPerPop=1 (26.0/2.0)
(PctIlleg >= 0.33) and (racePctWhite <= 0.53) and (PctHousLess3BR >= 0.59) => ViolentCrimesPerPop=1
(59.0/16.0)
(PctIlleg >= 0.26) and (pctWInvInc <= 0.36) and (numbUrban >= 0.04) and (RentMedian >= 0.44) =>
ViolentCrimesPerPop=1 (18.0/3.0)
(PctYoungKids2Par <= 0.62) and (racepctblack >= 0.25) and (PctWorkMom <= 0.53) =>
ViolentCrimesPerPop=1 (58.0/23.0)
(PctKids2Par <= 0.55) and (NumStreet >= 0.02) and (agePct12t29 <= 0.5) and (PctWOFullPlumb <= 0.16)
=> ViolentCrimesPerPop=1 (12.0/2.0)
 => ViolentCrimesPerPop=0 (1630.0/97.0)
```

RandomForest

The results of the experimenter showed that the RandomForest algorithm was the most promising tree based classifier. RandomForest works by taking n trees and builds them considering m random features. At each node of the tree, m variables are randomly chosen to base the decision on. Each tree is fully grown without any pruning.

Like JRip, RandomForest's default values provided solid results (~88.5% accuracy). We found that the accuracy peaked when increasing the number of trees from 10 to 14 trees (~89.27% accurate). With any more trees than 14 the accuracy began to decline; 15 trees had ~89.17% accuracy and 16 trees had ~88.87% accuracy. In addition, the default value used for number of features appeared to be optimal. The default value is log(M+1) where M is the number of inputs. In our data set the optimal number of features turned out to be seven. We were somewhat surprised that increasing number of features didn't increase the accuracy, however, it makes sense that the authors would chose an optimal default value.

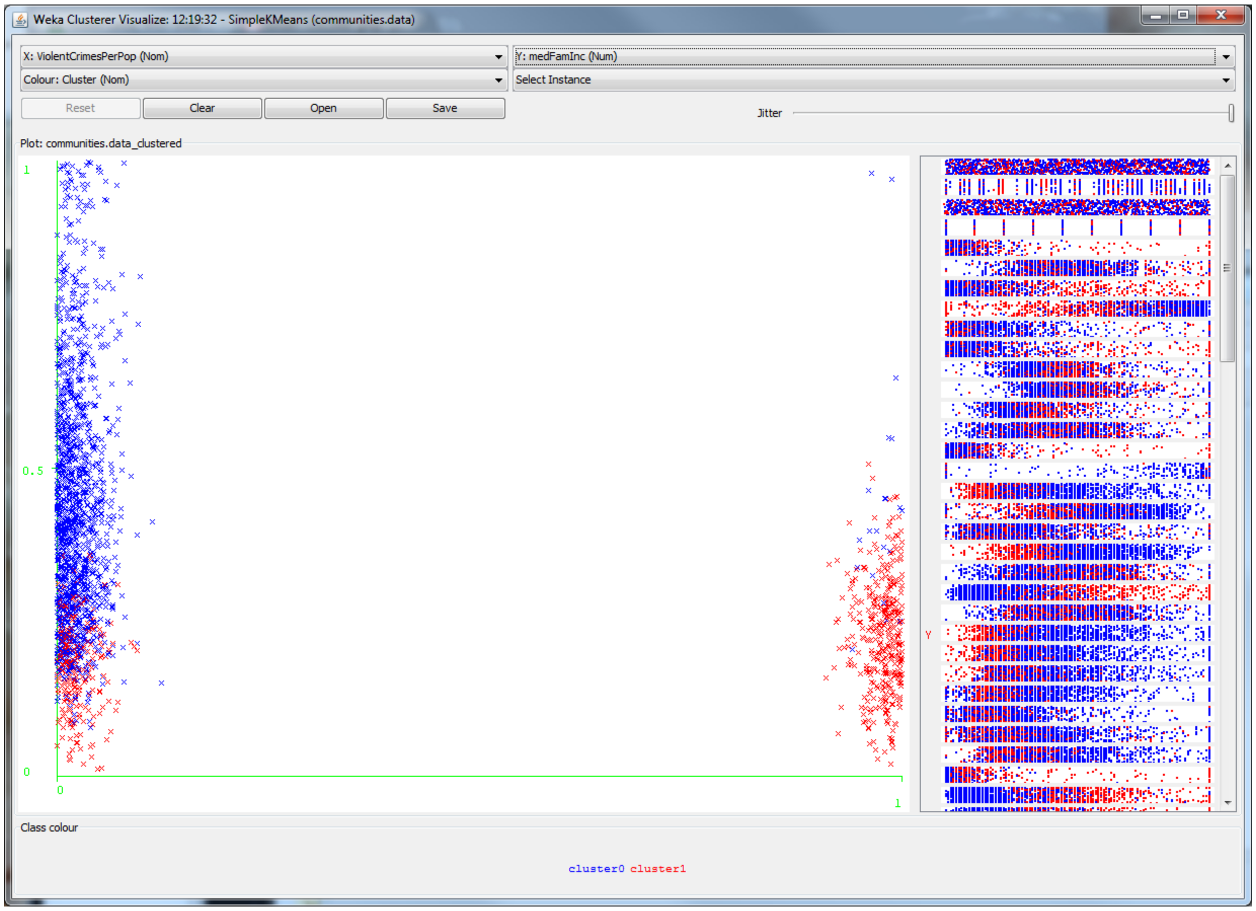
```
=== Confusion Matrix ===

    a    b   <-- classified as
 1556   64 |    a = 0
  150  224 |    b = 1
```

Cluster Analysis

In addition to classifying algorithms, we also played around a fair amount with clustering. For clustering we chose to use the KNN algorithm. In our experimentation, it appeared that n=2 provided the most meaningful clusters. Some of our results can be found below.

James Cuzella
Justin Guerra
Data Mining
Project 5.1



Weka Clusterer Visualize: 12:19:32 - SimpleKMeans (communities.data)

X: ViolentCrimesPerPop (Nom)
Y: medFamInc (Num)
Colour: Cluster (Nom)
Select Instance

Plot: communities.data_clustered

Figure 1: Violent Crime v. Median Income

Figure 1 shows the relation between violent crime and income. Cluster1 has a much higher rate of violent crime, and also had a low median income. This clustering also shows that income is not the only determinate of violent crime though. Lower violent crime rates have a large disparity in income ranges. In addition, there are a few affluent areas that have a high violent crime rate. It would be interesting if we had more data on these areas to determine why the crime rate is higher. For example, it could be possible that there are a large amount of armed robberies in these areas because of the wealth and maybe also a small police force. Or possibly there are just a lot of crimes of passion. Overall these results might show that having more money is a big incentive not to commit violent crimes.

In addition to information on violent crime, we also discovered other interesting trends in our data. In figure 2, the percentage of people with under a 9th grade education is plotted against the percentage of people on public assistance. It is clear that having such a low education corresponds to a much higher level of people living off public assistance. Figure 3 shows the percentage of illegitimate children plotted against the number of kids with two parents. There is a clear correlation between an illegitimate child being raised by a single parent throughout their lives. This is important because the red cluster has a higher violent crime rate. This shows that children born/raised by one parent are more likely to commit violent crimes than legitimate children raised by two parents.

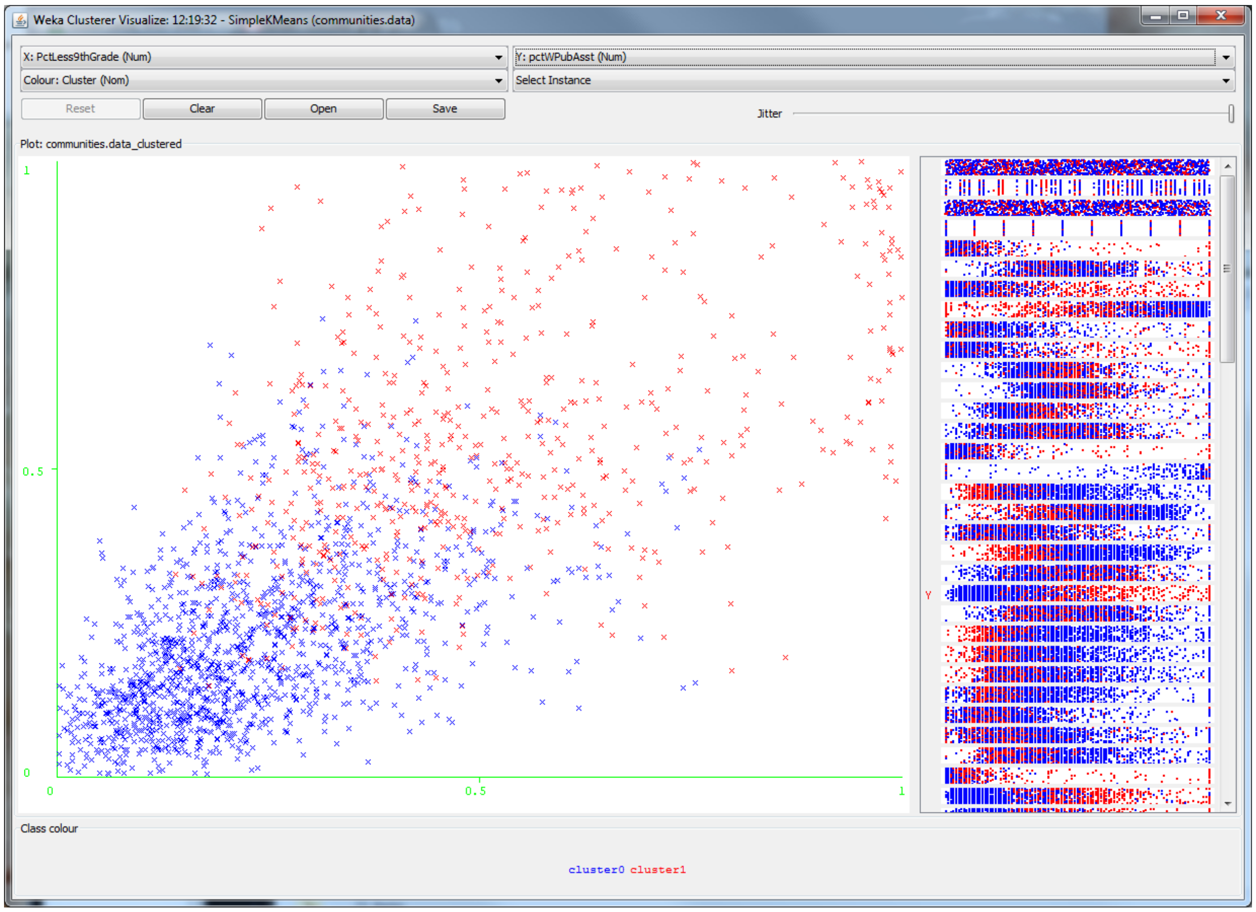


Figure 2: Percent with less than 9th grade education v. Percent with public assistance

James Cuzella
Justin Guerra
Data Mining
Project 5.1

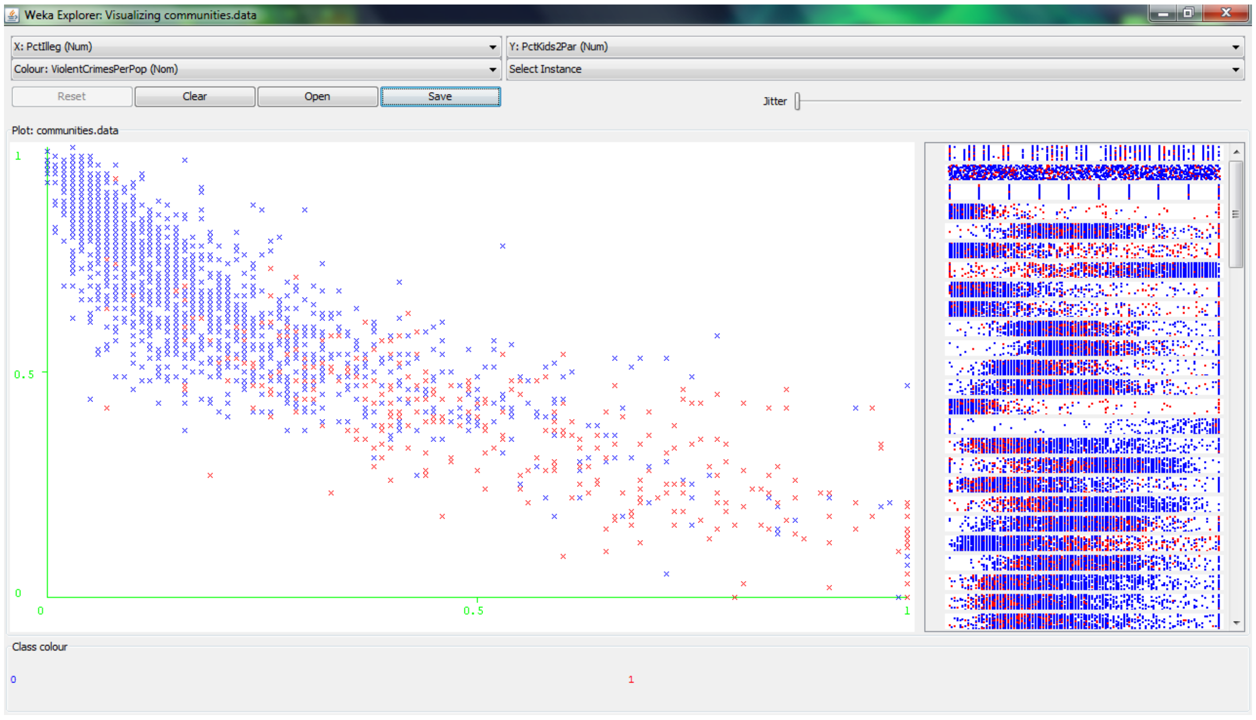


Figure 3: Percent illegitimate v. Percent with two parents

Works Cited

[1] Michael Redmond. (2009, July) UCI Machine Learning Repository - Communities and Crime Dataset. [Online]. http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime

[2] Claire Cain Miller. (2009, December) Local Governments Offer Data to Software Tinkerers. [Online]. http://www.nytimes.com/2009/12/07/technology/internet/07cities.html?_r=1