**CS5402 Final Project**

Drew Buelter and Trevor Ross

<https://github.com/trrevvorr/ML-final_proj>

December 6th - 2016

**Project Summary**

The goal of our project was to predict if a given person made over $50,000 given census data from 1988. The dataset we worked with included the following attributes:

* Age
  + Continuous
* Workclass
  + Private
  + Self-emp-not-inc
  + Self-emp-inc
  + Federal-gov
  + Local-gov
  + State-gov
  + Without-pay
  + Never-worked
* Education Number
  + Continuous
* Marital-Status
  + Married-civ-spouse
  + Divorced
  + Never-married
  + Separated
  + Widowed
  + Married-spouse-absent
  + Married-AF-spouse
* Occupation
  + Tech-support
  + Craft-repair
  + Other-service
  + Sales
  + Exec-managerial
  + Prof-specialty
  + Handlers-cleaners
  + Machine-op-inspct
  + Adm-clerical
  + Farming-fishing
  + Transport-moving
  + Priv-house-serv
  + Protective-serv
  + Armed Forces
* Relationship
  + Wife
  + Own-child
  + Husband
  + Not-in-family
  + Other-relative
  + Umarried
* Race
  + White
  + Asian-Pac-Islander
  + Amer-Indian-Eskimo
  + Other
  + Black
* Sex
  + Female
  + Male
* Capital-gain
  + Continuous
* Capital-loss
  + Continuous
* Hours-per-week
  + Continuous

Rather than simply implementing one algorithm to accomplish our task, we decided to try several different algorithms to see which ones perform the best. The algorithms we used included PLA, pocket PLA, linear regression, and logistic linear regression.

**Implementation**

Since a large portion of the data was in the form of strings, before we were able to implement the algorithms mentioned above, we had to transform our raw data into a usable format. In order to do this, we simply assigned integer values to each possible value of an attribute that consisted of strings. Additionally, we had to deal with some data that was incomplete. Because our dataset was large, we decided to simply get rid of records that were incomplete. After deleting those fields we were still left with a training dataset of approximately 27,000 records and a test dataset of about 13,000 records. Once the data was properly transformed we were able to test the four algorithms on it. Pseudo-code for the various algorithms is shown below:

**PLA**

Initialize w0

For t=0,1, …

1. Find the next mistake of wt called (xn(t),yn(t))
2. Correct mistakes by

Until no mistakes or last iteration

Return last w

**Pocket PLA**

Initialize

For t=0,1,…

1. Find the next mistake of wt called (xn(t),yn(t))
2. Correct mistakes by

Until no mistakes or last iteration

Return

**Linear Regression**

1. Given D={xn,yn}, construct x and y
2. Calculate
3. Return

**Logistic Linear Regression**

Initialize

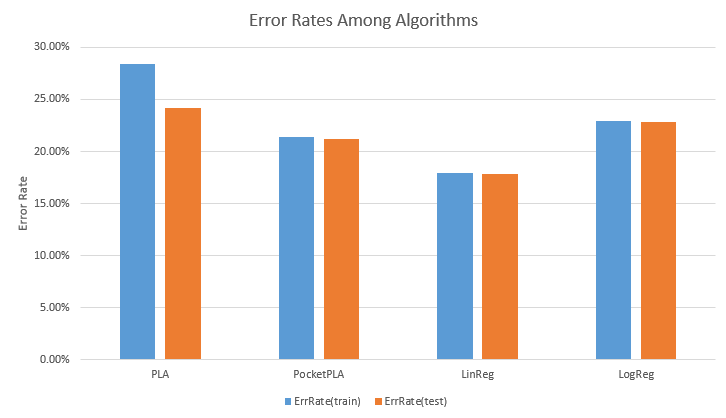
For t=0,1, …

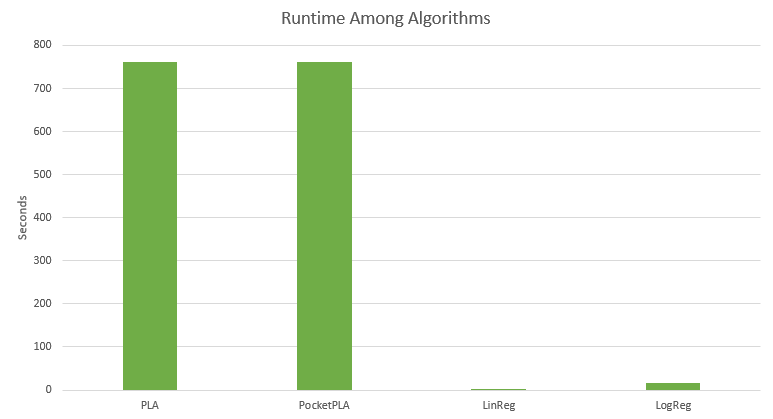
1. Compute
2. Update

Until

Return

**Results**





|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Error Rate (train) | Error Rate (test) | Time (s) |
| PLA | 28.45% | 24.20% | 761.789 |
| PocketPLA | 21.43% | 21.19% | 761.789 |
| LinReg | 17.94% | 17.83% | 0.202 |
| LogReg | 22.93% | 22.80% | 17.208 |

The results above show the various algorithms’ performance on our testing and training data. Both the training and testing error rates were very similar for each algorithm which tells us that overfitting was not an issue. This was expected as both our training and testing datasets were very large. Another result of such large datasets were time constraints on runtime. The PLA and PocketPLA algorithms could only be run with one iteration. Ideally, we would have preferred several hundred iterations to better tune *w*, but time did not allow. Linear regression, on the other hand, finished very quickly. It performed the best out of all the algorithms as it had the lowest error rate and runtime. Given more processing power or time, the PocketPLA might have been able to compete with, or even surpass linear regression; however, within the scope of the project we were not able to test this.

**Related Work**

Many other researchers have used the same dataset we did for their work. One example is the work of Ron Kohavi and Barry Becker who used the dataset to test various algorithms. The performance of those algorithms is show below (our best result is shown in bold):

|  |  |
| --- | --- |
| Algorithm | Error Rate (test) |
| IDTM (Decision Table) | 14.46% |
| HOODG | 14.82% |
| C4.5 | 15.54% |
| Voted ID3 | 15.64% |
| Naive-Bayes | 16.12% |
| **Linear Regression** | **17.83%** |
| Nearest-neighbor | 21.42% |

**Conclusion and Future Work**

Given time and computational limits, the linear regression algorithm was the best performing algorithm for our application. However, with more time or computational power, the PocketPLA might be able to compete with the linear regression algorithm. In the future we plan to optimize code and run the PLA algorithm with more iterations to see how it compares with the linear regression algorithm when allowed more iterations. All source code and analysis can be found in our repository, located at: <https://github.com/trrevvorr/ML-final_proj>