Adam Gincel

CS559

Assignment 1

Running the attached code:

The attached folder contains a file `hw1.py` – this can be run in a command line as long as Python 3 is installed with the command `python hw1.py`. The program will ask for the file name of the .csv file you look to analyze (these files must be in the same folder as the .py file itself), and will later on ask for a learning rate and epoch for which to calculate Logistic Regression over.

Indices.csv is included in this download as it is necessary to the interpretation of the datasets. Data_NESARC_tr and Data_NESARC_ts.csv were not included as they are large files to include in email attachments, but they are what the program was tested with.

Results:

On the test set, with a learning rate of 0.001 and with 10000 iterations, the resulting weight vector was:

```
[0.26772052872680824, 0.07724065579186402, 0.3595809234537087, 0.13632017239516664, -
0.31489936732223256, -0.5511900406490273, -0.12137705883069107, 0.28457558212098977,
0.20706744295079196, -0.37772575806907754, -0.3268630237947166, -0.46126338077685114, -
0.9150855733895837, -1.5516807072546444, -0.8528328135401421, 0.6006271340953866,
0.13788620387161146, 0.5353849200334654, -0.9038737959016473, -0.024985623462197272, -
0.07650191293317943, 0.10385167397199484, -0.06750063545198015, 0.383668330857784,
0.17278415926346724, -0.17126277707083676, 1.1912219738067606, -0.0056177324794245195, -
```

0.003228814277641164, 0.38089299192391, 0.8088237998121726, 0.28067948717817703, 0.02582411266518653, 0.2800113493247946, 0.14462570078984702, 0.40417830481236444, 1.3965656481349278, 0.9592019632770385]

This resulted in an 81.6161% success rate, and an error percentage of 18.3839%. These values are fairly similar through various learning rates and epochs – I believe they are a fairly powerful prediction measurement without overfitting to the data given.

Running another test over the training set yielded the following weight vector:

[0.24835266192436217, 0.181494624078222, 0.22678005273339594, 0.31470818249603794, -0.3028179611715537, -0.500583634274711, 0.028400124721642904, 0.3389532211943889, 0.11252277508344387, -0.3424363425772464, -0.3951740665086544, -0.35898278448544435, -0.8348828102631164, -1.2111439951376601, -0.901762083450253, 0.7046316618316657, 0.21601288995106058, 0.5229776294791654, -0.7840302118835486, -0.011781922593151417, -0.1050656989693357, 0.3037101944970366, -0.15196736991224732, 0.4224007324727235, 0.2474353127121483, -0.0687031240150973, 1.302693534979546, -0.0044992869490420246, 0.13006392578813972, 0.43479704531245655, 1.0015639369048257, 0.41234969900009827, -0.09793775244601202, 0.2361388510518646, 0.206694135559054, 0.2977862416436988, 0.5120714820551325, 0.6988737604123327]

This had a success rate of 81.81% and an error rate of 18.19%.

Analysis:

With the generated weight vectors, greater weight magnitudes denote greater importance to the prediction of the result. Higher positive values are correlated towards a positive result (yes), and negative values correlated towards a negative result (no). In this case, it is very clear that region, height, and weight were among some of the highly important fields necessary for predicting whether or not someone had ever drunk alcohol before. Changing, adding, and removing these

fields would be interesting as it would let us hypothesize which of these fields is most important in making good guesses. We would see obvious fluctuations in our accuracy percentage as we chose better or worse fields. For example, removing all fields except for region would probably yield much less accuracy than adding a new field about, say, access to alcohol in the house. Interestingly, very similar results are observed among learning rates between 0.1 and 0.001 and epochs between 250 and 10000. This suggests a very high correlation between the chosen data fields and the field we are trying to predict. If large differences were observed with higher learning rates and epochs, we might assume that we were overfitting to one specific sample size. Having a similar accuracy percentage across two datasets and multiple settings makes it fairly clear that we are likely not overfitting.