CS57300: Data Mining

Assignment 3

Name: Mohammad Haseeb
Purdue ID: mhaseeb@purdue.edu

Due: March 8, 2019

*Note: I used **1** late day for this assignment. Total late days consumed: **1/3**.*

1 Preprocessing

The files 'testingSet.csv' and 'testSet.csv' are stored in the same directory from which the script 'preprocess-assg3.py' was run. The output from the script is shown in Figure 1.

```
data 164 $ python3 preprocess-assg3.py
Mapped vector for female in column gender: [1].
Mapped vector for Black/African American in column race: [0, 1, 0, 0].
Mapped vector for Other in column race_o: [0, 0, 0, 0].
Mapped vector for economics in column field: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0].
```

Figure 1: Output of preprocess-assg3.py

2 Logistic Regression and Linear SVM

(i) Figure 2 shows the output after running Logistic Regression. It takes approx. 20 seconds to run this script on data.cs.purdue.edu.

```
data 179 $ python3 lr_svm.py trainingSet.csv testSet.csv 1
lr_svm.py:27: RuntimeWarning: overflow encountered in exp
  def sigmoid(x): return 1/(1+np.exp(-x))
Training Accuracy LR: 0.65
Testing Accuracy LR: 0.65
```

Figure 2: Output of lr_svm.py for LR

(ii) Figure 3 shows the output after running SVM. It takes approx. 45 seconds to run this script on data.cs.purdue.edu.

```
data 180 $ python3 lr_svm.py trainingSet.csv testSet.csv 2
Training Accuracy SVM: 0.56
Testing Accuracy SVM: 0.55
```

Figure 3: Output of lr_svm.py for SVM

3 Learning Curves and Performance Comparison

3.1 Learning Curves

The learning curves for the algorithms are shown in Figure 4. It takes approx. 8.5 minutes to run this script.

We can observe from the graph that as the size of the training data increases, the performance of NBC increases. This is as expected as for a small training set there could be conditional probabilities that our model does not learn but appear in training set. Also note that we are not using Laplacian Correction. For both LR and SVM, we observe that with increasing size of the training data, their performance decreases given the current set of η and λ.

3.2 Performance Comparison and Hypothesis Testing

I will formulate a hypothesis about the performance difference between NBC and SVM.

Let the null hypothesis H_0 and alternate hypothesis H_1 be defined as follows:

- H_0: The average accuracies of NBC = the average accuracies of SVM
- H_1: The average accuracies of NBC != the average accuracies of SVM

To test whether the means of these two paired samples are significantly different, I will use the Paired Student's t-test (Wikipedia).

I use the following formula to calculate the t-value:

$$t = \frac{\bar{X}_D - \mu_0}{\frac{s_D}{\sqrt{n}}}$$

$\bar{X}_D$ and s_D are the average and the standard deviation of the differences between the pairs. The constant μ_0 is zero if we want to test whether the average of the difference is significantly different (which we do). The degree of freedom used is $n - 1$, where n represents the number of pairs (6 in our case).

I used calculations shown in Figure 5 to calculate t where *nbc_avg_accuracies* and *svm_avg_accuracies* are actual values from *cv.py*.

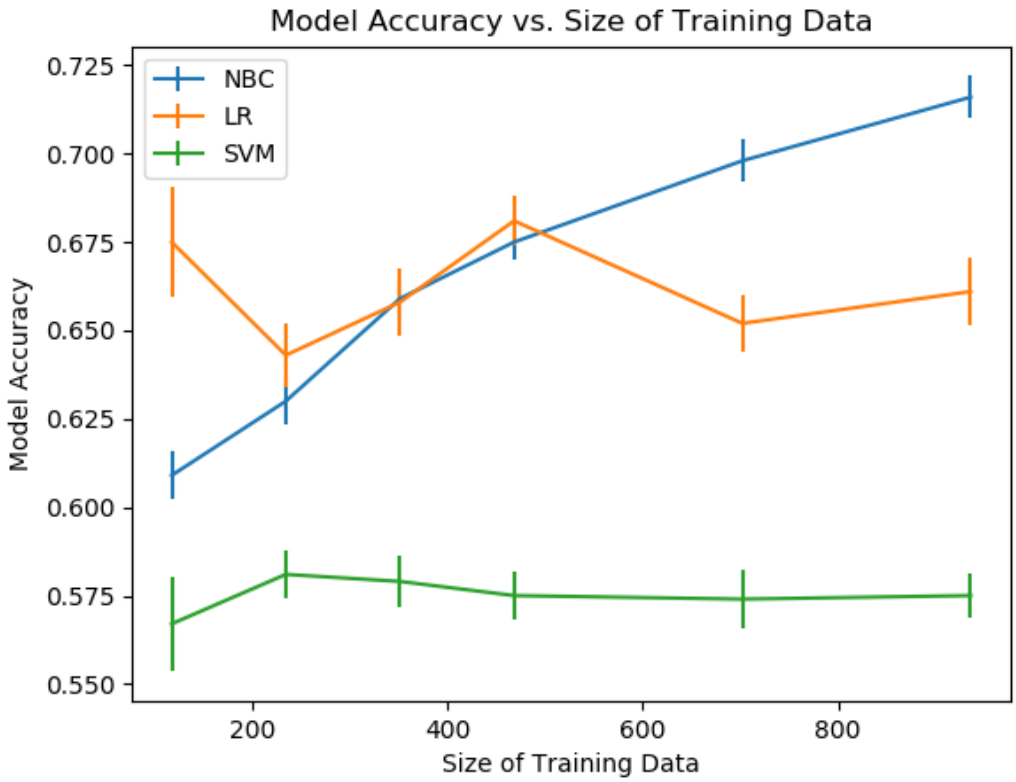

Figure 4: Learning Curves for NBC, LR, and SVM

The corresponding value for $t = 5.18163247138324$ is $p = 0.003521$ for a two-tailed hypothesis and a Significance Level of 0.05. The same values are found if I use the Python library *scipy* as is shown in Figure 6.
Now, because $p < 0.05$ (our selected significance level), we can reject H_0 and accept the alternate hypothesis H_1. Therefore, the observed data supports my hypothesis (i.e., the observed differences are significant).

```
>>> nbc_avg_accuracies = [0.61, 0.63, 0.66, 0.68, 0.7, 0.72]
>>> svm_avg_accuracies = [0.57, 0.58, 0.58, 0.57, 0.57, 0.57]
>>> z = list(zip(nbc_avg_accuracies, svm_avg_accuracies))
>>> diffs = [abs(p[0]-p[1]) for p in z]
>>> sd = statistics.stdev(diffs)
>>> mean = statistics.mean(diffs)
>>> t = mean/(sd/(math.sqrt(6)))
>>> t
5.18163247138324
```

Figure 5: Calculations for t

```
>>> from scipy import stats
>>> t_statistic, pvalue = stats.ttest_rel(nbc_avg_accuracies, svm_avg_accuracies)
>>> t_statistic
5.1816324713832405
>>> pvalue
0.0035205163968591796
```

Figure 6: Calculations for t and p using *scipy*