Mark Klobukov

CS 383 HW5: SVMs

Professor Matthew Burlick

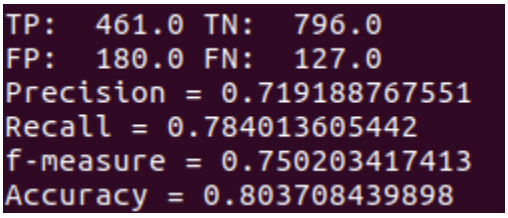
2/27/2018

The sklearn Python library was used in this assignment. It is installed on Tux, so there should be no additional installations required to run my code according to the instructions in readme.txt

Part 1: Binary SVM Results

Metric	Value
Total testing feature vectors	1564
# true positives	461
# true negatives	796
# false positives	180
# false negatives	127
Precision	0.719
Recall	0.784
F-Measure	0.750
Accuracy	0.804

When the program finishes execution, the following will get displayed to the screen (the numbers are the same as in the table above).

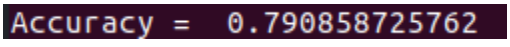


Part 2: Multi-Class SVM Results

Same classifier type as in Part 1 was used here.

Accuracy	0.791

When program finishes execution, the accuracy value gets displayed to the screen:



Part 3: Confusion Matrix

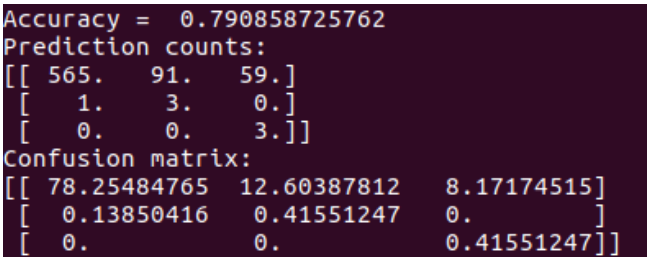
Without calculating percentages, the confusion matrix looks as follows:

	Class 1	**Class 2**	**Class 3**
Pred. Class1	565	91	59
Pred. Class 2	1	3	0
Pred. Class 3	0	0	3

Percentages for the confusion matrix were calculated. Each entry was divided by the total number of observations in the testing dataset.

	Class 1	**Class 2**	**Class 3**
Pred. Class1	78.25 %	12.60 %	8.17 %
Pred. Class 2	0.14 %	0.42 %	0 %
Pred. Class 3	0	0	0.42 %

When running the code as instructed in the readme.txt, the following will be printed to screen:



It appears that many samples are misclassified as Class 1. The first row of the confusion matrix has percentages much greater than the subsequent two rows. Given that a sample is classified as class 2 or class 3, the confidence in that classification is high. However, the chances of a sample being classified as class 2 or class 3 are low – overwhelming majority of the observations are classified as class 1.

The model is biased to classify samples as Class 1. The possible explanation for this is that in the dataset, there are many more samples with label 1 than the other two (namely, 1654 samples with label 1, 295 samples with label 2, and 176 samples with label 3). The problem may be solved if more observations with labels 2 and 3 are added so that the classifier can learn from more data.

The values in the confusion matrix, as well as the accuracy value from part 2, do not exactly correspond to those found in the assignment instructions. The possible explanation for this is that the way I shuffled the data was different. I used Numpy and provided the zero seed to the np.random.seed() function. It is possible that different distributions of samples went into the instructor's training and testing sets if a different way of randomization or providing the seed was used.