Programming Assignment 4

Assigned: Nov. 5
Due: Nov. 21

Overview

In this assignment, you will implement the Naive Bayes algorithm on a synthetic dataset.

WARNING: DO NOT USE CODE FOR NAIVE BAYES THAT YOU FIND ON THE WEB. You should write all the code yourself from scratch, except for the library functions for parsing the input, for reading from an Excel spreadsheet and for the logarithm function.

Dataset

The dataset is in an Excel file, linked on the course web page. It has 1000 rows and 6 columns. The first five, imaginatively named `a1, a2, a3, a4, a5` are the predictive attributes; they take values between 1 and 4. The sixth (`a6`) takes values between 1 and 3.

The data set has been constructed so that Naive Bayes should predict, certainly better than chance, but nowhere near perfectly.

There are no null values in the data set.

You may, and indeed should, hard-code the above features of the dataset into your program. However, you should not hard-code the actual content of the dataset as one enormous data statement. Your code will be tested on a different dataset with the identical structure but completely different values.

Input and Output

The input to your program is four parameters on a single input line:

- The number of rows to use for training. These will be the first rows of the dataset.
- The number of rows to use for testing. These will be the last rows of the test set.
- Optionally `-v` for verbose output, described below.

For example, the input `200 300 -v` should run Naive Bayes with verbose output, using lines 1-200 of the data set for training and lines 701-1000 for testing.

The output of the program is, on one line:

- If verbose output is specified then the values of the negative log probabilities (defined in the next section) in the form specified on the next page.
- In all cases, on one line: the overall accuracy on the test set. and the precision and recall for the specific category X.a6=3 in the test set.

Naive Bayes

The program will have two stages. In the training stage, the program will compute the relevant log probabilities over the training set. In the testing stage, it computes the classification attribute over each instance in the test set and tallies the accuracy, precision, and recall.

The log probabilities is the negative log probability of a 0.1 Laplacian correction of the relevant frequencies. That is:

$$lp(X.a_6 = v) = -\log_2\left(\frac{\#_T(X.a_6 = v) + 0.1}{|T| + 0.3}\right) \text{ for } v = 1, 2, 3$$

$$lp(X.a_i = u|X.a_6 = v) = -\log_2\left(\frac{\#_T(X.a_i = u, X.a_6 = v) + 0.1}{\#_T(X.a_6 = v) + 0.4}\right) \text{ for } v = 1, 2, 3;\ i = 1..5,\ u = 1..4$$

In the formulas above T is the training set; $|T|$ is the number of instances in T; and $\#_T(\phi)$ is the number of instances in T that satisfy condition ϕ.

In the testing stage, for each instance $X.a_1 = u_1 \dots X.a_5 = u_5$ in the test set, compute the value of the sum

$$lp(X.a_1 = u_1|X.a_6 = v) + \dots + lp(X.a_5 = u_5|X.a_6 = v) + lp(X.a_6 = v)$$

for $v = 1, 2, 3$; choose the value of v with the smallest sum; and compare it to the labeled value $X.a_6$.

It is altogether unlikely that you will ever run into a tie, but if you do, break it arbitrarily.

Format for verbose output

```
lp(X.a6=1)    lp(X.a6=2)    lp(X.a6=3)


lp(X.a1=1|X.a6=1)    lp(X.a1=2|X.a6=1)    lp(X.a1=3|X.a6=1) lp(X.a1=4|X.a6=1)
lp(X.a1=1|X.a6=2)    lp(X.a1=2|X.a6=2)    lp(X.a1=3|X.a6=2) lp(X.a1=4|X.a6=2)
lp(X.a1=1|X.a6=3)    lp(X.a1=2|X.a6=3)    lp(X.a1=3|X.a6=3) lp(X.a1=4|X.a6=3)


lp(X.a2=1|X.a6=1)    lp(X.a2=2|X.a6=1)    lp(X.a2=3|X.a6=1) lp(X.a2=4|X.a6=1)
lp(X.a2=1|X.a6=2)    lp(X.a2=2|X.a6=2)    lp(X.a2=3|X.a6=2) lp(X.a2=4|X.a6=2)
lp(X.a2=1|X.a6=3)    lp(X.a2=2|X.a6=3)    lp(X.a2=3|X.a6=3) lp(X.a2=4|X.a6=3)


lp(X.a3=1|X.a6=1)    lp(X.a3=2|X.a6=1)    lp(X.a3=3|X.a6=1) lp(X.a3=4|X.a6=1)
lp(X.a3=1|X.a6=2)    lp(X.a3=2|X.a6=2)    lp(X.a3=3|X.a6=2) lp(X.a3=4|X.a6=2)
lp(X.a3=1|X.a6=3)    lp(X.a3=2|X.a6=3)    lp(X.a3=3|X.a6=3) lp(X.a3=4|X.a6=3)


lp(X.a4=1|X.a6=1)    lp(X.a4=2|X.a6=1)    lp(X.a4=3|X.a6=1) lp(X.a4=4|X.a6=1)
lp(X.a4=1|X.a6=2)    lp(X.a4=2|X.a6=2)    lp(X.a4=3|X.a6=2) lp(X.a4=4|X.a6=2)
lp(X.a4=1|X.a6=3)    lp(X.a4=2|X.a6=3)    lp(X.a4=3|X.a6=3) lp(X.a4=4|X.a6=3)


lp(X.a5=1|X.a6=1)    lp(X.a5=2|X.a6=1)    lp(X.a5=3|X.a6=1) lp(X.a5=4|X.a6=1)
lp(X.a5=1|X.a6=2)    lp(X.a5=2|X.a6=2)    lp(X.a5=3|X.a6=2) lp(X.a5=4|X.a6=2)
lp(X.a5=1|X.a6=3)    lp(X.a5=2|X.a6=3)    lp(X.a5=3|X.a6=3) lp(X.a5=4|X.a6=3)


Accuracy=⟨accuracy⟩. Precision=⟨precision⟩. Recall=⟨recall⟩.
```