Predicting Risk of Credit Applicants

Data Mining
Prof. Dawn Woodard
School of ORIE
Cornell University

Outline

1 Announcements

2 Credit Data Analysis

Announcements

- Reading for week 3 (listed on schedule in “Information” section on Blackboard): SPB Chap.5 through p.102; also Chap.7.
- Make sure you finished the readings for weeks 1 & 2 listed on Blackboard
- Questions?

Reading

Correction for week 3 reading:

- The definition of false positive rate and true positive rate in the text are VERY nonstandard. Ignore the box on p.102, and USE THE DEFINITIONS I'LL GIVE IN CLASS TODAY INSTEAD.

- The book uses "sensitivity" to refer to the true positive rate we will define in class today. It uses "specificity" to refer to the true negative rate we will define in class today. The terms "sensitivity" and "specificity" are correctly defined in SPB and are commonly used.

Reading

Note on week 3 reading:

- So far in class we have talked about training data and testing data.
- The book defines three categories: training, validation, and test data. We will talk about this distinction soon, but in the meantime just think of validation and test data as being data we did NOT use to train (fit) the model.

Credit Data Analysis

- Download the credit data from Blackboard (both files). The data is in “crx.data” and information about the dataset (metadata) is in “crx.names”

- Look at the metadata, noticing the section on missing data. Find an instance of missing data in the data file; they are coded using a “?”.

- Create a text (script) file to put all your code in for today; you may need to rerun it and you don’t want to have to retype.

Credit Data Analysis

- Read the dataset into R:

  ```
  > credit = read.table(“C:/temp/crx.data”, sep = “,”,
  na.strings = “?” )
  ```

- how many records and variables in the dataset? Does it match the metadata?

Credit Data Analysis

- Missing data are represented in R using “NA” values. Check that the missing values have been read in correctly by calling

  ```
  > sum( is.na( credit ) )
  ```

  This should be equal to the number of missing values in the dataset (why?) as reported in the metadata. How many missing values?

A. 0-100

B. 101-200

C. 201-300

D. 301-400

Credit Data Analysis

- Since we have only one dataset, we will need to split it into train and test datasets. Sample 460 records at random without replacement to create the training data, and use the rest to create the test dataset:

  ```
  nData = dim( credit )[1]
  # here we draw 460 indices at random:
  trainInd = sample( (1:nData), 460 )
  # create the training data:
  trainData = credit[ trainInd, ]
  # create the test data:
  testData = credit[ -trainInd, ]
  ```

  Why does this syntax work?

Credit Data Analysis

- Summarize the distribution of the first predictor for the training data using the “summary” function. Are there missing values for this predictor?

- Do the same for the second predictor, and notice that it is continuous.

- We do not yet know how to fit naive Bayes using continuous predictors (will learn later in course)

- For now we discretize this predictor, i.e. turn it into a finite set of values. One simple approach is to split at the median, creating one value if the predictor is less than its median, and another value if it is greater than its median. First find the median of the second predictor using the “median” function.

  ```
  medianV2 = median( trainData$V2, na.rm = T )
  trainData$V2 = as.factor( trainData$V2 > medianV2 )
  ```

Credit Data Analysis

- Call the “summary” function on the newly defined V2 variable to make sure you did this correctly. The counts are about the same in the two categories (as they should be!!).

- Perform the analogous transformation of all of the continuous predictors in the data set (i.e. split at the median of the predictor). Code for this is posted in the “code” folder on Blackboard.

- If you run the discretization code more than once for each variable you will get wierd results later on. Make sure you only discretize each predictor once.

- Perform the same transformation for all the continuous predictors in the **test** data set. For each continuous predictor variable, split the test data at the same value at which you split the training data. For instance for V2 split the test data at “medianV2”, the median from the training data, not at the median of V2 from the test data!! The code is also on Blackboard.

Credit Data Analysis

- Your professor has kindly altered the naive Bayes training and testing functions to handle missing data, as we discussed in class. Copy the altered naive Bayes functions (still called nb.train and nb.predict) from the naiveBayesMissing.R file (on Blackboard) to the R command prompt.

- Now train naive Bayes on the training data, and test on the testing data:

  ```
  creditTrained = nb.train( D = trainData )
  nb.predict( D = testData, nb = creditTrained )
  ```

- What is your error rate? What types of errors are being made (i.e. how many times did you predict a $+$ when actually $Y = -$ and vice versa?)?