Individual Assignment 4

Topics: K-NN and Naïve Bayes, Classification vs. Ranking

105078702 Wendy Huang

Submission instruction:

1. Upload a single PDF to Canvas. It should include all the screenshots and answers 1-8.
2. Hand in the printed copy of your document in class.

Data Details and Goal:

The file "user_changes.xlsx" contains information on EZTABLE restaurant reservations by members and an analysis. Worksheet "train" (also available as file train.txt) is the data from the Kaggle contest: it includes the first reservation made by EZTABLE members. For each reservation, there is information that the user entered at the time of booking, and column ***status*** which is entered by EZTABLE. ***status*** has two values that indicate a change that a user made on the EZTABLE website after his/her initial booking: "change" means that the user changed the reservation in some way. "canceled" means that the user canceled it online.

We want to build a model that predicts, right after a booking, whether the user will change/cancel that booking in the future (**status** = *changed* or *canceled*).

Data Preparation:

- Set random seed to 12345. (R: set.seed(12345))
- Take a random sample of 10,000 records (from worksheet *train*).
- Create column ***cancel_change*** that takes 1 if *status*={canceled or changed}, 0 otherwise.
- Create column ***time_to_booking*** from *datetime* (restaurant reservation date) and *cdate*
- Partition the data randomly into training/validation/test sets using 50%-30%-20%. Keep *booking_id, time_to_booking, people,* and *cancel_change*

Naïve Benchmark

1. Using the information in the training dataset, if a member just logged in but did not complete their booking (we have no information), would we classify the booking as ***cancel_change*** =1 or =0? What is the probability that this record will be canceled/change?

   Classification: __0____
   Probability of cancellation/change = ____21.48%_____

K-NN

Run a K-NN classifier on the partitioned datasets: Use predictors *people* and *time_to_booking* and outcome ***cancel_change*** (with 1 as the success class). Cutoff = 0.5. Normalize the predictor variables (XLMiner: "normalize"; R use preProcess() - see Table 7.2 in book); Let the validation set find the best k between 1-20 (XLMiner: "score on be k between 1 and specified value", choose 20 or the largest value allowed; R - see Table 7.3 in book)

2. From the output (XLMiner: worksheets KNNC_Output, KNNC_TestLiftChart) fill in:

   Best k = _____19_____

   Training overall error = _____21.88_____ %

   Validation overall error = _____21.36_____ %

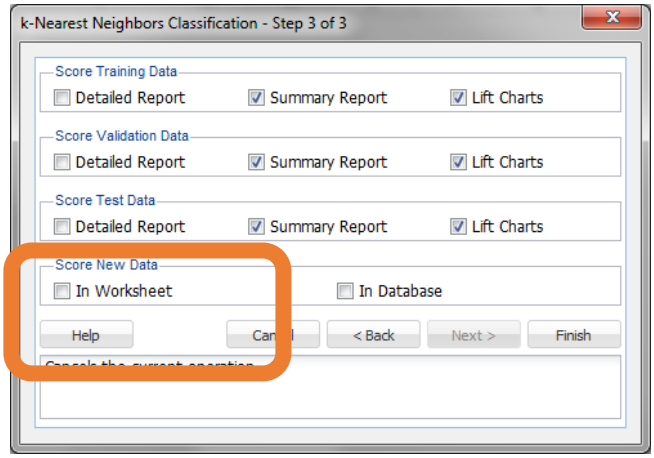
   Test overall error = _____21.05_____ %

   Test Lift for first decile = _____1.39_____ (R: use gains(), see Fig 5.7 in book)

3. Fill in the predicted probabilities and classifications **for the 10 records in the table below**.

   XLMiner: copy the three left columns into a new worksheet; re-run KNN, in Step 3 click "Score New Data" in Worksheet and choose this worksheet.

   R - see Table 7.4 in book.



Booking #	people	time_to_booking	KNN: Prob of 1	KNN Classification (0/1)
1	2	27	0.3157895	0
2	2	63	0.3000000	0
3	2	1.85	0.4210526	0
4	10	1.7	0.4210526	0
5	2	13	0.3157895	0
6	11	8.7	0.2631579	0
7	2	1.8	0.4210526	0
8	4	7	0.2631579	0
9	2	2.5	0.2105263	0
10	6	10	0.2105263	0

Naïve Bayes

From the SampleWS worksheet (or file SampleWS.txt), convert *people* into a categorical variable *Binned_people*, by splitting it into **two** categories with an equal number of records. Similarly, convert *time_to_booking* into *Binned_time_to_booking* with two categories with an equal number of records.

- XLMiner: Transform > Bin Continuous Data
- R: use function median: Binned_people <- ifelse(people < median(people), 1, 2)

Partition the data again (50-30-20%) and keep *booking_id, cancel_change*, and the two Binned columns.

Run a naïve Bayes classifier on the partitioned dataset using only the two new categorical predictors you created. Keep all other options as in the KNN. R: see Table 8.4 in textbook.

4. From the output (XLMiner worksheets NB_Output, NNB_TestLiftChart), fill in:

   Training overall error = _____21.43_____ %

   Validation overall error = ____20.52_____ %

   Test overall error = _____22.06_____ %

   Test Lift for first decile = ____1.147____ (R: use gains(), see Fig 5.7 in book)

5. Fill in the predicted probabilities and classifications for the 10 records in the table below. You'll need to add their Binned values before NB can score them. R: see Table 8.6 in book.

Booking #	people	time_to_booking	Binned_people	Binned_time_to_booking	NB: Prob of 1	NB Classification (0/1)
1	2	27	1	2	0.2488304	0
2	2	63	1	2	0.2488304	0
3	2	1.85	1	1	0.1991569	0
4	10	1.7	2	1	0.1839551	0
5	2	13	1	2	0.2488304	0
6	11	8.7	2	2	0.2309303	0
7	2	1.8	1	1	0.1991569	0
8	4	7	2	2	0.2309303	0
9	2	2.5	1	1	0.1991569	0
10	6	10	2	2	0.2309303	0

Exact Bayes

6. Use a pivot table to compute the **count** of *cancel_change*=1 and 0 in the **training dataset** as a function of the two categorical predictors.

	cancel_change=1	cancel_change=0
Binned_people = 1, Binned_time_to_booking = 1	652	2473
Binned_people = 1, Binned_time_to_booking = 2	772	3585
Binned_people = 2, Binned_time_to_booking = 1	720	2293
Binned_people = 2, Binned_time_to_booking = 2	1070	3435

7. Compute manually the **exact** Bayes conditional probabilities of a cancel_change=1, given *Binned_people* and *Binned_time_to_booking* as the two predictor variables. Show your calculation for the first record.

   Then, use cutoff=0.5 to get the classifications.

Booking #	**people**	**time_to_ booking**	**Binned_ people**	**Binned_time_ to_booking**	**Exact Bayes: Prob of 1**	**Exact Bayes Classification (0/1)**
1	2	27	1	2	0.2401991	0
2	2	63	1	2	0.2401991	0
3	2	1.85	1	1	0.2028625	0
4	10	1.7	2	1	0.2240199	0
5	2	13	1	2	0.2401991	0
6	11	8.7	2	2	0.3329185	0
7	2	1.8	1	1	0.2028625	0
8	4	7	2	2	0.3329185	0
9	2	2.5	1	1	0.2028625	0
10	6	10	2	2	0. 3329185	0

8. Compare the Naïve Bayes and exact Bayes scores for the 10 records.

   Are the resulting probabilities identical? No (choose one)

   Are the resulting classifications identical? Yes (choose one)

   Is the ranking (= ordering) of observations identical? No (choose one)