STAT 471: Midterm Exam

[Name]

March 22, 2020

Contents

Instructions

- This exam is open-book / open-notes / open-internet. However, it is individual work. Communication among students is prohibited.
- Please complete your homework in R Markdown, using this document as a starting point. Show your R code using code chunks and add your text answers using **bold text**.
- When you are ready to submit, please compile your R Markdown file into a PDF. Then, submit this PDF through Gradescope.
- While base R programming is acceptable, I strongly encourage you to use the `tidyverse` to complete your assignment. However, points will not be deducted if you use base R programming.
- In addition, please make sure to have clear labels and sensible titles on your plots.
- **Make sure that `readmission_clean.csv` and `plot_glmnet.R` are in your working directory before beginning the exam.**

Introduction: Predicting readmission for diabetes patients

Background and goals

Diabetes is a chronic medical condition affecting millions of Americans, but if managed well, patients can lead relatively normal lives. However, if improperly managed, diabetes can lead to patients being continuously admitted and readmitted to hospitals. Hospital readmissions represent a failure of the health system to provide adequate support to the patient and are extremely costly to the system. The goals are therefore to

1. identify important factors associated with readmission, and
2. predict whether a given patient will be readmitted.

Readmission data

In this exam, we will investigate a dataset originally from the Center for Clinical and Translational Research at Virginia Commonwealth University, covering diabetes patients across 130 U.S. hospitals from 1999 to 2008. Three former STAT 471 students Spencer Luster, Matthew Lesser and Mridul Ganesh brought this data set into the class through their final project. In this exam, we will use a cleaned subset of this data.

First, let's load a few libraries:

```r
library(glmnet)
library(scales)
library(tidyverse)
```

Let's also source `plot_glmnet.R` to get access to the `plot_glmnet` function (make sure this file is in your working directory).

```r
source("plot_glmnet.R")
```

Finally, let's load the data:

```r
readmission = read_csv("readmission_clean.csv")
readmission
```

```
## # A tibble: 99,492 x 26
##    race  gender age_group num_outpatient num_inpatient num_emergency
##    <chr> <chr>  <chr>              <dbl>         <dbl>         <dbl>
##  1 Cauc~ Female 80+                    0             0             0
##  2 Cauc~ Female 80+                    0             0             0
##  3 Cauc~ Male   20-59                  0             0             0
##  4 Afri~ Female 20-59                  0             0             0
##  5 Cauc~ Male   20-59                  0             0             0
##  6 Afri~ Male   60-79                  0             0             0
##  7 Cauc~ Female 20-59                  0             0             1
##  8 Cauc~ Male   80+                    0             0             0
##  9 Cauc~ Male   60-79                  0             0             0
## 10 Afri~ Female 60-79                  0             0             0
## # ... with 99,482 more rows, and 20 more variables: num_medications <dbl>,
## #   num_diagnoses <dbl>, adm_source <chr>, adm_type <chr>,
## #   time_in_hospital <dbl>, num_lab_procedures <dbl>, num_procedures <dbl>,
## #   discharge <chr>, max_glu_serum <chr>, A1Cresult <chr>, med_changed <chr>,
## #   med_prescribed <chr>, insulin <chr>, metformin <chr>, glimepiride <chr>,
## #   glipizide <chr>, glyburide <chr>, pioglitazone <chr>, rosiglitazone <chr>,
## #   readmitted <dbl>
```

Each row corresponds to a hospital admission of a patient. There are 26 total variables (a response `readmitted` and 25 features), described below:

Demographic variables

- `race`: patient's race
- `gender`: patient's gender
- `age_group`: patient's age group

Medical history

- `num_outpatient`: number of outpatient visits by the patient in the year prior to the current admission
- `num_inpatient`: number of inpatient visits by the patient in the year prior to the current admission
- `num_emergency`: number of emergency visits by the patient in the year prior to the current admission
- `num_medications`: number of total medications the patient has taken
- `num_diagnoses`: number of total diagnoses the patient has

Hospital admission details

- `adm_source`: who referred the patient to the hospital
- `adm_type`: type of admission
- `time_in_hospital`: length of stay in the hospital (in days)
- `num_lab_procedures`: number of lab procedures performed
- `num_procedures`: number of non-lab procedures performed
- `discharge`: where the patient was discharged

Clinical results

- `max_glu_serum`: results of glucose serum test
- `A1Cresult`: results of A1c test

Medication details

- `med_changed`: whether any medication was changed
- `med_prescribed`: whether any medication was prescribed
- `insulin`: type of change (if any) to insulin medication
- `metformin`: type of change (if any) to insulin medication
- `glimepiride`: type of change (if any) to glimepiride medication
- `glipizide`: type of change (if any) to glipizide medication
- `glyburide`: type of change (if any) to glyburide medication
- `pioglitazone`: type of change (if any) to pioglitazone medication
- `rosiglitazone`: type of change (if any) to rosiglitazone medication

Readmission indicator

- `readmitted`: whether the patient was readmitted to the hospital within 30 days of discharge

Train/test split

Let's split off 5000 observations for training and leave the rest for testing:

```r
set.seed(1) # set seed for reproducibility (DO NOT CHANGE)
n_total = nrow(readmission)
n_train = 5000
n_test = n_total-n_train
partition = sample(c(rep("train", n_train), rep("test", n_test)))
readmission_train = readmission %>%
  bind_cols(partition = partition) %>%
  filter(partition == "train") %>%
  select(-partition)
readmission_test = readmission %>%
  bind_cols(partition = partition) %>%
  filter(partition == "test") %>%
  select(-partition)
```

1 Exploratory data analysis (20 points total)

First, let's do some exploratory data analysis on our training data `readmission_train`.

1. (5 points) What fraction of the patients in the training data were readmitted?

2. (5 points)

    - Produce a bar plot to display the breakdown of the patients by age group.
    - What is the most prevalent age group in the training data?

3. (10 points)

   - Produce a plot to show the relationship between `time_in_hospital` and `readmitted`. [Hint: It may be useful to convert `readmitted` to a factor using `as.factor(readmitted)`.]
   - Using `summarise`, compute the median time in hospital separately for patients that were not readmitted and for those that were.
   - Do these suggest that readmission rates vary based on time in hospital, and if so, what is the direction of the relationship?

2 Association via logistic regression (35 points total)

Next, let's explore the factors associated with hospital readmission using logistic regression **on the training data.**

4. (10 points)

   - Run a logistic regression of `readmitted` on all of the features and print the summary.
   - Based on this summary, how many features are significantly associated with `readmitted` at the 0.05 level? [For the purposes of this question, treat dummy features for categorical variables as separate features.]
   - Which of these are also significant at the 0.01 level? [For the purposes of this question, treat dummy features for categorical variables as separate features.]

5. (15 points) Based on the summary above, insulin seems to be the medication most associated with readmission.

   - What is the appropriate statistical test of the null hypothesis that none of the *other* medications (namely, `metformin`, `glimepiride`, `glipizide`, `glyburide`, `pioglitazone`, and `rosiglitazone`) are associated with `readmitted` while controlling for all of the other features?
   - Carry out this test, and state the resulting p-value and the conclusion you would make at the 0.05 significance level.

6. (10 points) Suppose patient A and patient B match on all features, except patient A has `insulin` = "Steady" while patient B has `insulin` = "Up". Based on the fitted coefficients from question 4 above, what is the relationship between the predicted odds of `readmitted` for patient B and the corresponding odds for patient A? Express your answer quantitatively.

3 Prediction via (lasso) logistic regression (45 points total)

7. (15 points total)

i. (5 points)

   - Fit a 10-fold cross-validated logistic lasso regression to the training data, using the misclassification error for cross-validation.
   - Produce the cross-validation plot.
   - Why is the left-most number across the top of the CV plot 54, rather than 25, the latter being the total number of features in the data?

```r
set.seed(3) # set seed for reproducibility (DO NOT CHANGE)
# TODO
```

ii. (5 points) How many nonzero coefficients (not counting the intercept term) are there when `lambda = lambda.min` and when `lambda = lambda.1se`?

```r
# TODO
```

iii. (5 points) Code is provided below to produce the lasso trace plot. Based on this plot, which feature is the first to have a nonzero coefficient as `lambda` decreases from infinity to zero?

```r
# produce trace plot
plot_glmnet(glmnet_fit, lambda = NA, features_to_plot = 5)
```

8. (10 points) Define vectors `logreg_probabilities` and `glmnet_probabilities` containing the predicted probabilities for each of the test observations under the logistic model and the logistic lasso model (with `lambda = lambda.min`), respectively.

```r
glmnet_probabilities = # TODO
logreg_probabilities = # TODO
```

Let's put these together into a data frame, together with the test response:

```r
predictions = tibble(glmnet_probabilities,
                     logreg_probabilities,
                     readmitted = readmission_test %>% pull(readmitted))
```

9. (10 points)

   - For each of the above methods (logistic regression and lasso logistic regression with `lambda = lambda.min`), make a box plot to compare the predicted probabilities of readmission for patients that were and were not in fact readmitted. You should end up with two separate plots. [Hint: use `as.factor(readmitted)`.]
   - Do the predicted probabilities tend to be larger for test observations for which `readmitted` = 1?

10. (10 points)

    - Use `mutate` to add two columns to `predictions` called `logreg_predictions` and `glmnet_predictions`, which are the binary classifications of these two methods based on thresholding the probabilities at 0.5 [Hint: `as.numeric` converts logical variables to binary numeric variables].
    - Then, `summarise` the resulting data frame to obtain the misclassification error of both methods.
    - How do these misclassification errors compare?
    - In what sense is this conclusion consistent with the CV plot from question 7?