In-Class ML Competition

Tadao Hoshino （星野匡郎）

Econometrics II: ver. 2019 Spring Semester

Competition Outline

Competition Outline

- **Date**: 23 July; **Time**: 10:40 – 12:00 (10 mins for preparation, 70 mins for coding and submission.); **Place**: 3-901 (this room).
  - Download the data and submit your answer through the **Course Navi**.
  - The submission folder will be automatically closed at 12:00.
- You are allowed to bring "anything" with you to the class room, including lecture slides, textbooks, pre-written **R** scripts, etc.
- But you are not allowed to communicate with others and also to ask me any technical questions.

Competition Outline

- **Available data**: $\mathbf{X}^{\text{train}}$, Y^{train}, $\mathbf{X}^{\text{test}}$.
- **Task**: to predict the true value of Y^{test} as accurately as possible, where Y^{test} is a dummy variable (i.e., classification problem).
  - IMPORTANT: do NOT submit the predicted binary responses, but submit the "classification scores" $s(\mathbf{X}^{\text{test}})$'s to compute the AUC.
  - It is not necessary to submit the **R** code used.
- **Evaluation**: The performance of your prediction algorithm will be evaluated by the AUC score.

* This competition accounts for 40% of your final grade:

$$40 = a + b \times \text{your AUC score},$$

where a and b will be determined later.

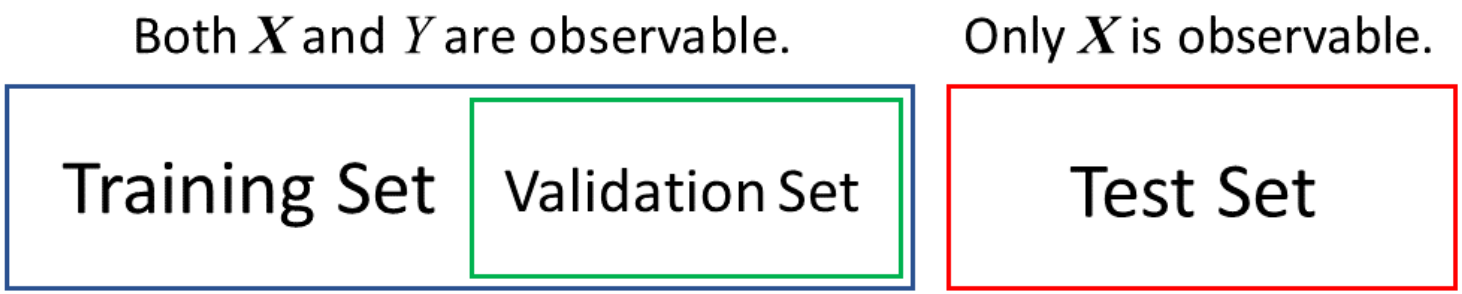
Cross Validation

Cross Validation

- True Y^{test} is unobservable $\Rightarrow$ you cannot compute any performance statistics including Accuracy and AUC.
- How can we compare alternative prediction models?

Cross Validation

- Further split the training set into a reduced training set and a "validation" set.
- Train each model on the reduced training set, and select the best model based on the results on the validation set.

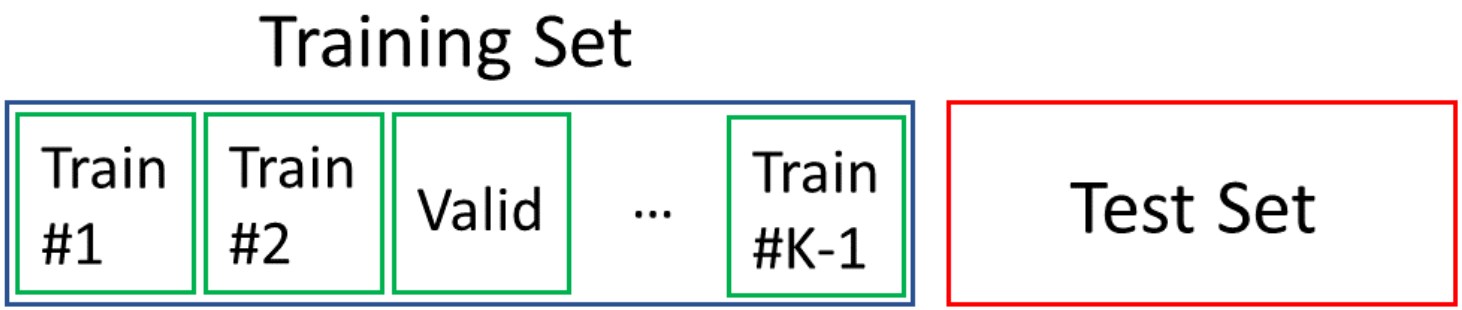


Cross Validation

K-fold Cross Validation

- The above mentioned method runs the risk of overfitting to a particular validation set (especially when the size of the training data is small).
  $\Rightarrow$ K-fold cross-validation approach:

1. Randomly split the training set in K equally sized subsets.
2. Keep the k-th subset as a validation set, and train the model on the remaining $K-1$ subsets. Compute the AUC on this validation set, AUC_k.
3. Repeat this process from $k=1$ to $k=K$, and compute the average $\overline{\text{AUC}} = K^{-1}\sum_{k=1}^{K}\text{AUC}_k$. Finally, choose the best model in terms of $\overline{\text{AUC}}$.



Cross Validation

K-fold Cross Validation (cont.)

- A common choice for K is either 5 (80% for training and 20% for testing) or 10 (90% for training and 10% for testing), but there is no formal theoretical justification for these numbers.
- Repeated K-fold Cross Validation:
  - In a K-fold cross validation, only K estimates of model performance are obtained.
  - After reshuffling the data, run K-fold cross validation multiple times.

Sample Code

```
library(ROCR)
setwd("C:/Rdataset")
data <- read.csv("spam_train.csv")
data$type <- (data$type == "spam")

AUC <- function(s, Y){
    pred <- ROCR::prediction(s, Y)
    auc  <- performance(pred, "auc")@y.values[[1]]
    return(auc)
    }

K <- 10
N <- nrow(data) # Total sample size
n <- floor(N/K) # The size of each subset
data <- data[sample(N),] # Randomly shuffle the data
```

Sample Code (cont.)

```
CV <- function(k){
    ids <- (k - 1)*n + 1:n
    test  <- data[ids,]
    train <- data[-ids,]

    m1 <- lm(type ~., data = train)
    m2 <- glm(type ~., data = train, family = binomial(link = "logit"))

    s1 <- predict(m1, newdata = test)
    s2 <- predict(m2, newdata = test)

    auc1 <- AUC(s1, test$type)
    auc2 <- AUC(s2, test$type)

    return(c(auc1,auc2))
    }
```

Sample Code (cont.)

```
cvmat <- matrix(0,K,2) # Matrix of zeros of dimension (K, #models)
for(k in 1:K) cvmat[k,] <- CV(k)
colMeans(cvmat)
```

The above **R** code is available at the **Course Navi**.

Pre-Competition

Infant Birth Weight

Birth weight data[1]

- Training data: **bweight_train.csv** (including both $\mathbf{X}$ and Y)
- Test data: **X_test.csv** and **Y_test.csv**, where **Y_test.csv** will not be downloadable until 12:00.
- Submission file: **submission.csv**
- The csv files are uploaded on the **Course Navi** (not from my website).

[1] Obtained from Wooldridge's dataset:
`http://fmwww.bc.edu/ec-p/data/wooldridge/datasets.list.html`

Infant Birth Weight

Definitions of variables

Response variable

lbw3000 TRUE if birth weight $\leq$ 3,000 (kg), and FALSE otherwise.

Input variables

xage, xeduc, xrace x's age, x's education in years, and x's race ("white", "black" or "other"), respectively.
x = m $\Rightarrow$ mother; x = f $\Rightarrow$ father.

monpre month prenatal care began.

npvis total number of prenatal visits.

omaps, fmaps One-minute and five-minute Apgar scores, respectively.[2]

cigs average cigarettes per day.

drink average drinks per week.

[2] The Apgar score is the very first test performed on a newborn baby at 1 and 5 minutes after birth.

Infant Birth Weight

The task:

Compute classification scores for all 500 individuals in the test set, which are indexed by ID = 1, ..., 500, for the prediction of {`lbw3000 = TRUE`}.

Submission process:

1. Using the training data, develop your prediction model.
2. Based on your model, compute the classification scores for the observations in the test data. Typically, you can obtain them using the `predict()` function.[3]

[3] Here, it would be important to check that the obtained scores are not binary but continuous values.

Infant Birth Weight

Submission process (cont.):

3. Load the `submission.csv` file:

	A	B
1	ID	score
2	1	
3	2	
4	3	
5	4	

   store the obtained classification scores in the variable `score`, and overwrite the csv file:

```
submit <- read.csv("submission.csv")
submit$score <- s # s = classification score
write.csv(submit, "submission.csv") # overwrite the file
```

4. Submit this through **Course Navi**.