Comprehension Check: Cross-validation

Q1

1/1 point (graded)

Generate a set of random predictors and outcomes using the following code:

```
set.seed(1996) #if you are using R 3.5 or earlier
set.seed(1996, sample.kind="Rounding") #if you are using R 3.6 or later
n <- 1000
p <- 10000
x <- matrix(rnorm(n*p), n, p)
colnames(x) <- paste("x", 1:ncol(x), sep = "_")
y <- rbinom(n, 1, 0.5) %>% factor()

x_subset <- x[ ,sample(p, 100)]
```

Because `x` and `y` are completely independent, you should not be able to predict `y` using `x` with accuracy greater than 0.5. Confirm this by running cross-validation using logistic regression to fit the model. Because we have so many predictors, we selected a random sample `x_subset`. Use the subset when training the model.

Which code correctly performs this cross-validation?

○
```
fit <- train(x_subset, y)
fit$results
```

●
```
fit <- train(x_subset, y, method = "glm")
fit$results
```

○
```
fit <- train(y, x_subset, method = "glm")
fit$results
```

○
```
fit <- test(x_subset, y, method = "glm")
fit$results
```

✓

? **Hint (1 of 1):** Use the caret `train` function. The `results` component of the output of `train` shows you the accuracy. Ignore the warnings.

Next Hint

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

Q2

1/1 point (graded)

Now, instead of using a random selection of predictors, we are going to search for those that are most predictive of the outcome. We can do this by comparing the values for the $y = 1$ group to those in the $y = 0$ group, for each predictor, using a t-test. You can do perform this step like this:

```
install.packages("BiocManager")
BiocManager::install("genefilter")
library(genefilter)
tt <- colttests(x, y)
```

Which of the following lines of code correctly creates a vector of the p-values called `pvals` ?

○ `pvals <- tt$dm`

○ `pvals <- tt$statistic`

○ `pvals <- tt`

● `pvals <- tt$p.value`

✓

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

Q3

1/1 point (graded)

Create an index `ind` with the column numbers of the predictors that were "statistically significantly" associated with `y`. Use a p-value cutoff of 0.01 to define "statistically significantly."

How many predictors survive this cutoff?

108

✓ **Answer:** 108

108

Explanation

The number of predictors that survive the cutoff can be found using this code:

```
ind <- which(pvals <= 0.01)
length(ind)
```

Submit

You have used 1 of 10 attempts

Answers are displayed within the problem

Q4

1/1 point (graded)

Now re-run the cross-validation after redefinining `x_subset` to be the subset of `x` defined by the columns showing "statistically significant" association with `y`.

What is the accuracy now?

0.7564915

✓ **Answer:** 0.754

0.7564915

Explanation

The accuracy can be calculated using the following code:

```
x_subset <- x[,ind]
fit <- train(x_subset, y, method = "glm")
fit$results
```

Submit You have used 1 of 10 attempts

Answers are displayed within the problem

Q5

1/1 point (graded)

Re-run the cross-validation again, but this time using kNN. Try out the following grid `k = seq(101, 301, 25)` of tuning parameters. Make a plot of the resulting accuracies.

Which code is correct?

●

```
fit <- train(x_subset, y, method = "knn", tuneGrid = data.frame(k = seq(101, 301,
25)))
ggplot(fit)
```

○

```
fit <- train(x_subset, y, method = "knn")
ggplot(fit)
```

○

```
fit <- train(x_subset, y, method = "knn", tuneGrid = data.frame(k = seq(103, 301,
25)))
ggplot(fit)
```

○

```
fit <- train(x_subset, y, method = "knn", tuneGrid = data.frame(k = seq(101, 301,
5)))
ggplot(fit)
```

✓

Submit You have used 1 of 2 attempts

Answers are displayed within the problem

Q6

1/1 point (graded)

In the previous exercises, we see that despite the fact that `x` and `y` are completely independent, we were able to predict `y` with accuracy higher than 70%. We must be doing something wrong then.

What is it?

○ The function `train` estimates accuracy on the same data it uses to train the algorithm.

○ We are overfitting the model by including 100 predictors.

● We used the entire dataset to select the columns used in the model.

○ The high accuracy is just due to random variability.

✓

Explanation

Because we used the entire dataset to select the columns in the model, the accuracy is too high. The selection step needs to be included as part of the cross-validation algorithm, and then the cross-validation itself is performed **after** the column selection step.
As a follow-up exercise, try to re-do the cross-validation, this time including the selection step in the cross-validation algorithm. The accuracy should now be close to 50%.

Submit You have used 1 of 2 attempts

ℹ Answers are displayed within the problem

Q7

1/1 point (graded)

Use the `train` function with kNN to select the best `k` for predicting tissue from gene expression on the `tissue_gene_expression` dataset from dslabs. Try `k = seq(1,7,2)` for tuning parameters. For this question, do not split the data into test and train sets (understand this can lead to overfitting, but ignore this for now).

What value of `k` results in the highest accuracy?

1 ✓ **Answer:** 1

1

Explanation

The following code will allow you to pick the best value of `k`:

```
data("tissue_gene_expression")
fit <- with(tissue_gene_expression, train(x, y, method = "knn", tuneGrid = data.frame( k
= seq(1, 7, 2))))
ggplot(fit)
fit$results
```

Submit

You have used 1 of 10 attempts

Answers are displayed within the problem

Ask your questions or make your comments about Cross-validation here! **Remember, one of the best ways to reinforce your own learning is by explaining something to someone else, so we encourage you to answer each other's questions (without giving away the answers, of course).**

Some reminders:

- Search the discussion board before posting to see if someone else has asked the same thing before asking a new question.
- Please be specific in the title and body of your post regarding which question you're asking about to facilitate answering your question.
- Posting snippets of code is okay, but posting full code solutions is not.
- If you do post snippets of code, please format it as code for readability. If you're not sure how to do this, there are instructions in a pinned post in the "general" discussion forum.

Discussion: Cross-validation

Show Discussion

Topic: Section 4: Distance, Knn, Cross-Validation, and Generative Models / 4.2.1: Cross-validation