Class 28: Cross-validation

Dr. Glasbrenner

December 6, 2017

Setup

```
# Configure settings for compiling to HTML and PDF
knitr::opts_chunk$set(
  echo = TRUE, eval = TRUE, fig.width = 5, fig.asp = 0.618, out.width = "70%",
  dpi = 120, fig.align = "center", cache = TRUE, warning = FALSE)
# Load required packages
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(modelr))
suppressPackageStartupMessages(library(broom))
load(url("http://fall17.cds101.com/R/mariokart_cross_validation.RData"))
# Load and clean Mario Kart dataset
mariokart <- read_rds(url("http://fall17.cds101.com/datasets/mariokart.rds")) %>%
  filter(totalPr <= 100) %>%
  select(totalPr, cond, stockPhoto, duration, wheels)
```

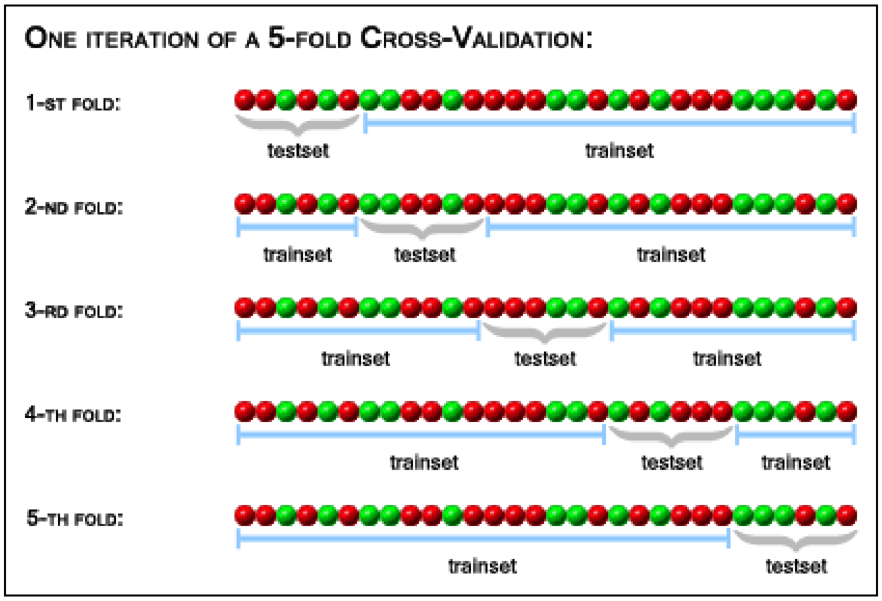
Get a blank working RMarkdown file:

```
download.file("http://fall17.cds101.com/documents/class28.rmarkdown", destfile = "class28.Rmd")
```

k-fold cross-validation

K-fold cross-validation is one kind of cross-validation procedure that's available. These methods allow us to estimate how robust the model is by systematically removing different chunks of the dataset and repeating the fitting process. The picture below illustrates what it means:

```
knitr::include_graphics("cross-validation-schematic.png")
```



The above example shows a "5-fold", or $k = 5$, cross-validation. There is a testing set, and the remaining $k - 1$ folds are used when fitting the model. After performing a fit, you apply the model to the training fold to see how well it does, and then calculate the mean-squared prediction error, which gives an estimate of how well the model works as a predictor. You can also calculate an R^2 for the prediction.

We'll look at how well two different kinds of folding choices perform on the Mario Kart dataset, $k = 2$ and $k = 10$. To compare different models, we should split the data exactly the same way each time. To do that, we set the `seed =` input on the `mariokart_cv` function.

k = 2

cond model

```
result <- mariokart_cv(
  data = mariokart, k = 2,
  lm_formula = formula(totalPr ~ cond),
  nruns = 100, seed = 152)
cv_report(result)
```

```
## # A tibble: 1 x 4
##   mse_mean   mse_sd r_squared_mean r_squared_sd
##      <dbl>    <dbl>          <dbl>        <dbl>
## 1 55.76271 2.243083      0.3174578   0.03514986
```

wheels + cond model

```
result <- mariokart_cv(
  data = mariokart, k = 2, lm_formula = formula(totalPr ~ wheels + cond),
  nruns = 100, seed = 152)
cv_report(result)
```

```
## # A tibble: 1 x 4
##   mse_mean   mse_sd r_squared_mean r_squared_sd
##      <dbl>    <dbl>          <dbl>        <dbl>
## 1 24.90933 1.074566      0.6939019   0.01657214
```

wheels + stockPhoto + cond model

```
result <- mariokart_cv(
  data = mariokart, k = 2,
  lm_formula = formula(totalPr ~ wheels + stockPhoto + cond),
  nruns = 100, seed = 152)
cv_report(result)
```

```
## # A tibble: 1 x 4
##   mse_mean mse_sd r_squared_mean r_squared_sd
##      <dbl>  <dbl>          <dbl>        <dbl>
## 1  25.1483 1.1784      0.6910377    0.0176153
```

duration + wheels + stockPhoto + cond model

```
result <- mariokart_cv(
  data = mariokart, k = 2,
  lm_formula = formula(totalPr ~ duration + wheels + stockPhoto + cond),
  nruns = 100, seed = 152)
cv_report(result)
```

```
## # A tibble: 1 x 4
##   mse_mean   mse_sd r_squared_mean r_squared_sd
##      <dbl>    <dbl>          <dbl>        <dbl>
## 1 25.78106 1.529286      0.6830405    0.0221466
```

```
result <- mariokart_cv(
  data = mariokart, k = 2,
  lm_formula = formula(totalPr ~ duration),
  nruns = 100, seed = 152)
cv_report(result)
```

```
## # A tibble: 1 x 4
##   mse_mean   mse_sd r_squared_mean r_squared_sd
##      <dbl>    <dbl>          <dbl>        <dbl>
## 1 73.82857 3.352116     0.09931143   0.04754692
```

k = 10

cond model

```
result <- mariokart_cv(
  data = mariokart, k = 10,
  lm_formula = formula(totalPr ~ cond),
  nruns = 100, seed = 320)
cv_report(result)
```

```
## # A tibble: 1 x 4
##   mse_mean    mse_sd r_squared_mean r_squared_sd
##      <dbl>     <dbl>          <dbl>        <dbl>
## 1 55.22924 0.5279726       0.246825   0.04951977
```

wheels + cond model

```
result <- mariokart_cv(
  data = mariokart, k = 10, lm_formula = formula(totalPr ~ wheels + cond),
  nruns = 100, seed = 320)
cv_report(result)
```

```
## # A tibble: 1 x 4
##   mse_mean    mse_sd r_squared_mean r_squared_sd
##      <dbl>     <dbl>          <dbl>        <dbl>
## 1 24.49117 0.3063059      0.6491363   0.03163265
```

wheels + stockPhoto + cond model

```
result <- mariokart_cv(
  data = mariokart, k = 10,
  lm_formula = formula(totalPr ~ wheels + stockPhoto + cond),
  nruns = 100, seed = 320)
cv_report(result)
```

```
## # A tibble: 1 x 4
##   mse_mean    mse_sd r_squared_mean r_squared_sd
##      <dbl>     <dbl>          <dbl>        <dbl>
## 1  24.5795 0.3324297      0.6487389    0.0309606
```

duration + wheels + stockPhoto + cond model

```
result <- mariokart_cv(
  data = mariokart, k = 10,
  lm_formula = formula(totalPr ~ duration + wheels + stockPhoto + cond),
  nruns = 100, seed = 320)
cv_report(result)
```

```
## # A tibble: 1 x 4
##   mse_mean    mse_sd r_squared_mean r_squared_sd
##      <dbl>     <dbl>          <dbl>        <dbl>
## 1 25.00116 0.3832303      0.6423503   0.03257053
```

Exercise

Try and find a set of parameters that works better than cond + wheels

cond	stockPhoto	wheels	duration	r^2
X				0.3174578
	X			0.3311888
		X		0.09
			X	0.10
X		X		0.6939019
X	X	X		0.6910377
X	X	X	X	0.6830405
		X	X	0.6379804
X		X	X	0.6841935
	X		X	0.08686714
	X	X	X	0.6356733