

# Data Science Linear Regression

The textbook for the Data Science course series is [freely available online](#).

## Learning Objectives

- The basics of machine learning
- How to perform cross-validation to avoid overtraining
- Several popular machine learning algorithms
- How to build a recommendation system
- What regularization is and why it is useful

## Course Overview

There are six major sections in this course: introduction to machine learning; machine learning basics; linear regression for prediction, smoothing, and working with matrices; distance, knn, cross validation, and generative models; classification with more than two classes and the caret package; and model fitting and recommendation systems.

### Introduction to Machine Learning

In this section, you'll be introduced to some of the terminology and concepts you'll need going forward.

### Machine Learning Basics

In this section, you'll learn how to start building a machine learning algorithm using training and test data sets and the importance of conditional probabilities for machine learning.

### Linear Regression for Prediction, Smoothing, and Working with Matrices

In this section, you'll learn why linear regression is a useful baseline approach but is often insufficiently flexible for more complex analyses, how to smooth noisy data, and how to use matrices for machine learning.

### Distance, Knn, Cross Validation, and Generative Models

In this section, you'll learn different types of discriminative and generative approaches for machine learning algorithms.

### Classification with More than Two Classes and the Caret Package

In this section, you'll learn how to overcome the curse of dimensionality using methods that adapt to higher dimensions and how to use the caret package to implement many different machine learning algorithms.

## Model Fitting and Recommendation Systems

In this section, you'll learn how to apply the machine learning algorithms you have learned.

## Section 1 - Introduction to Machine Learning Overview

In the **Introduction to Machine Learning** section, you will be introduced to machine learning.

After completing this section, you will be able to:

- Explain the difference between the **outcome** and the **features**.
- Explain when to use **classification** and when to use **prediction**.
- Explain the importance of **prevalence**.
- Explain the difference between **sensitivity** and **specificity**.

This section has one part: **introduction to machine learning**.

### Notation

There is a link to the relevant section of the textbook: [Notation](#)

#### Key points

- $X_1, \dots, X_p$  denote the features,  $Y$  denotes the outcomes, and  $\hat{Y}$  denotes the predictions.
- Machine learning prediction tasks can be divided into **categorical** and **continuous** outcomes. We refer to these as **classification** and **prediction**, respectively.

### An Example

There is a link to the relevant section of the textbook: [An Example](#)

#### Key points

- $Y_i$  = an outcome for observation or index  $i$ .
- We use boldface for  $\mathbf{X}_i$  to distinguish the vector of predictors from the individual predictors  $X_{i,1}, \dots, X_{i,784}$ .
- When referring to an arbitrary set of features and outcomes, we drop the index  $i$  and use  $Y$  and bold  $\mathbf{X}$ .
- Uppercase is used to refer to variables because we think of predictors as random variables.
- Lowercase is used to denote observed values. For example,  $\mathbf{X} = \mathbf{x}$ .

## Comprehension Check - Introduction to Machine Learning

1. True or False: A key feature of machine learning is that the algorithms are built with data.

- ☒ A. True  
☐ B. False

2. True or False: In machine learning, we build algorithms that take feature values ( $X$ ) and train a model using known outcomes ( $Y$ ) that is then used to predict outcomes when presented with features without known outcomes.

- ☒ A. True  
☐ B. False

## Section 2 - Machine Learning Basics Overview

In the **Machine Learning Basics** section, you will learn the basics of machine learning.

After completing this section, you will be able to:

- Start to use the **caret** package.
- Construct and interpret a **confusion matrix**.
- Use **conditional probabilities** in the context of machine learning.

This section has two parts: **basics of evaluating machine learning algorithms** and **conditional probabilities**.

### Caret package, training and test sets, and overall accuracy

There is a link to the relevant sections of the textbook: [Training and test sets](#) and [Overall accuracy](#)

#### Key points

- Note: the `set.seed()` function is used to obtain reproducible results. If you have R 3.6 or later, please use the `sample.kind = "Rounding"` argument whenever you set the seed for this course.
- To mimic the ultimate evaluation process, we randomly split our data into two — a training set and a test set — and act as if we don't know the outcome of the test set. We develop algorithms using only the training set; the test set is used only for evaluation.
- The `createDataPartition()` function from the **caret** package can be used to generate indexes for randomly splitting data.
- Note: contrary to what the documentation says, this course will use the argument `p` as the percentage of data that goes to testing. The indexes made from `createDataPartition()` should be used to create the test set. Indexes should be created on the outcome and not a predictor.
- The simplest evaluation metric for categorical outcomes is overall accuracy: the proportion of cases that were correctly predicted in the test set.

#### Code

```
if(!require(tidyverse)) install.packages("tidyverse")
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages -----
```

```
## v ggplot2 3.3.2    v purrr   0.3.4
## v tibble  3.0.3    v dplyr   1.0.2
## v tidyr   1.1.2    v stringr 1.4.0
## v readr   1.3.1    v forcats 0.5.0
```

```
## -- Conflicts -----
```

```
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
if(!require(caret)) install.packages("caret")
```

```
## Loading required package: caret
```

```
## Loading required package: lattice
```

```
##
```

```
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
```

```
##
```

```
## lift
```

```
if(!require(dslabs)) install.packages("dslabs")
```

```
## Loading required package: dslabs
```

```
library(tidyverse)
```

```
library(caret)
```

```
library(dslabs)
```

```
data(heights)
```

```
# define the outcome and predictors
```

```
y <- heights$sex
```

```
x <- heights$height
```

```
# generate training and test sets
```

```
set.seed(2, sample.kind = "Rounding") # if using R 3.5 or earlier, remove the sample.kind argument
```

```
## Warning in set.seed(2, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
```

```
## used
```

```
test_index <- createDataPartition(y, times = 1, p = 0.5, list = FALSE)
```

```
test_set <- heights[test_index, ]
```

```
train_set <- heights[-test_index, ]
```

```
# guess the outcome
```

```
y_hat <- sample(c("Male", "Female"), length(test_index), replace = TRUE)
```

```
y_hat <- sample(c("Male", "Female"), length(test_index), replace = TRUE) %>%  
  factor(levels = levels(test_set$sex))
```

```
# compute accuracy
```

```
mean(y_hat == test_set$sex)
```

```
## [1] 0.5238095
```

```
heights %>% group_by(sex) %>% summarize(mean(height), sd(height))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 3
##   sex      `mean(height)` `sd(height)`
##   <fct>      <dbl>      <dbl>
## 1 Female      64.9      3.76
## 2 Male       69.3      3.61
```

```
y_hat <- ifelse(x > 62, "Male", "Female") %>% factor(levels = levels(test_set$sex))
mean(y == y_hat)
```

```
## [1] 0.7933333
```

```
# examine the accuracy of 10 cutoffs
cutoff <- seq(61, 70)
accuracy <- map_dbl(cutoff, function(x){
  y_hat <- ifelse(train_set$height > x, "Male", "Female") %>%
    factor(levels = levels(test_set$sex))
  mean(y_hat == train_set$sex)
})
data.frame(cutoff, accuracy) %>%
  ggplot(aes(cutoff, accuracy)) +
  geom_point() +
  geom_line()
```



```
max(accuracy)
```

```
## [1] 0.8361905
```

```
best_cutoff <- cutoff[which.max(accuracy)]
best_cutoff
```

```
## [1] 64
```

```
y_hat <- ifelse(test_set$height > best_cutoff, "Male", "Female") %>%
  factor(levels = levels(test_set$sex))
y_hat <- factor(y_hat)
mean(y_hat == test_set$sex)
```

```
## [1] 0.8171429
```

## Comprehension Check - Basics of Evaluating Machine Learning Algorithms

1. For each of the following, indicate whether the outcome is continuous or categorical.

- Digit reader - categorical
- Height - continuous
- Spam filter - categorical
- Stock prices - continuous
- Sex - categorical

2. How many features are available to us for prediction in the `mnist` digits dataset?

You can download the `mnist` dataset using the `read_mnist()` function from the `dslabs` package.

```
mnist <- read_mnist()
ncol(mnist$train$images)
```

```
## [1] 784
```