# The caret package

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

# The caret R package

## the caret package

The **caret** package (short for *C*lassification *A*nd *RE*gression *T*raining) is a set of functions that attempt to streamline the process for creating predictive models. The package contains tools for:

Links

`train` Model List

Topics

- Main Page
- Data Sets
- Visualizations
- Pre-Processing

http://caret.r-forge.r-project.org/

# Caret functionality

- Some preprocessing (cleaning)
  - preProcess
- Data splitting
  - createDataPartition
  - createResample
  - createTimeSlices
- Training/testing functions
  - train
  - predict
- Model comparison
  - confusionMatrix

# Machine learning algorithms in R

- Linear discriminant analysis
- Regression
- Naive Bayes
- Support vector machines
- Classification and regression trees
- Random forests
- Boosting
- etc.

# Why caret?

| `obj` Class | Package | `predict` Function Syntax |
|---|---|---|
| `lda` | `MASS` | `predict(obj)` (no options needed) |
| `glm` | `stats` | `predict(obj, type = "response")` |
| `gbm` | `gbm` | `predict(obj, type = "response", n.trees)` |
| `mda` | `mda` | `predict(obj, type = "posterior")` |
| `rpart` | `rpart` | `predict(obj, type = "prob")` |
| `Weka` | `RWeka` | `predict(obj, type = "probability")` |
| `LogitBoost` | `caTools` | `predict(obj, type = "raw", nIter)` |

http://www.edii.uclm.es/~useR-2013/Tutorials/kuhn/user_caret_2up.pdf

# SPAM Example: Data splitting

```
library(caret); library(kernlab); data(spam)
inTrain <- createDataPartition(y=spam$type,
                               p=0.75, list=FALSE)
training <- spam[inTrain,]
testing <- spam[-inTrain,]
dim(training)
```

```
[1] 3451   58
```

# SPAM Example: Fit a model

```
set.seed(32343)
modelFit <- train(type ~.,data=training, method="glm")
modelFit
```

```
Generalized Linear Model

3451 samples
  57 predictors
   2 classes: 'nonspam', 'spam'

No pre-processing
Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 3451, 3451, 3451, 3451, 3451, 3451, ...

Resampling results

  Accuracy  Kappa  Accuracy SD  Kappa SD
  0.9       0.8    0.02         0.04
```

# SPAM Example: Final model

```
modelFit <- train(type ~.,data=training, method="glm")
modelFit$finalModel
```

```
Call:  NULL

Coefficients:
      (Intercept)              make           address               all             num3d
        -1.78e+00        -7.76e-01        -1.39e-01         3.68e-02         1.94e+00
              our              over            remove          internet            order
         7.61e-01         6.66e-01         2.34e+00         5.94e-01         4.10e-01
              mail          receive             will           people           report
         4.08e-02         2.71e-01        -1.08e-01        -2.28e-01        -1.14e-01
         addresses             free         business            email              you
         2.16e+00         8.78e-01         6.49e-01         1.38e-01         6.91e-02
            credit             your             font          num000            money
         8.00e-01         2.17e-01         2.17e-01         2.04e+00         1.95e+00
                hp              hpl           george          num650              lab
        -1.82e+00        -9.17e-01        -7.50e+00         3.33e-01        -1.89e+00
              labs           telnet           num857             data           num415
```

# SPAM Example: Prediction

```
predictions <- predict(modelFit,newdata=testing)
predictions
```

```
  [1] spam    spam    spam    nonspam nonspam nonspam spam    spam    spam    spam    spam
 [12] spam    spam    spam    spam    spam    spam    spam    nonspam spam    spam    spam
 [23] nonspam spam    nonspam nonspam spam    spam    spam    spam    spam    spam    spam
 [34] spam    spam    spam    spam    spam    spam    spam    spam    spam    spam    spam
 [45] spam    spam    spam    spam    nonspam spam    nonspam spam    spam    spam    spam
 [56] spam    nonspam nonspam spam    spam    spam    spam    spam    nonspam spam    spam
 [67] spam    spam    spam    spam    spam    spam    spam    spam    spam    spam    spam
 [78] nonspam nonspam nonspam spam    spam    nonspam spam    nonspam nonspam spam    spam
 [89] spam    spam    spam    spam    spam    spam    nonspam spam    spam    spam    spam
[100] spam    spam    spam    nonspam spam    nonspam spam    spam    spam    spam    spam
[111] spam    spam    spam    spam    nonspam spam    spam    spam    spam    spam    spam
[122] spam    spam    spam    spam    spam    spam    spam    nonspam spam    spam    nonspam
[133] spam    spam    spam    spam    spam    spam    spam    spam    spam    spam    spam
[144] spam    spam    spam    nonspam spam    spam    spam    spam    spam    spam    spam
[155] nonspam spam    nonspam spam    nonspam spam    spam    spam    spam    spam    spam
[166] spam    spam    spam    spam    spam    spam    spam    spam    spam    spam    spam
[177] spam    spam    spam    spam    spam    spam    spam    spam    spam    spam    spam
```

# SPAM Example: Confusion Matrix

```
confusionMatrix(predictions,testing$type)
```

```
Confusion Matrix and Statistics

          Reference
Prediction nonspam spam
   nonspam     665   54
   spam         32  399

               Accuracy : 0.925
                 95% CI : (0.908, 0.94)
    No Information Rate : 0.606
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.842
 Mcnemar's Test P-Value : 0.0235

            Sensitivity : 0.954
            Specificity : 0.881
         Pos Pred Value : 0.925
```

# Further information

- Caret tutorials:
  - http://www.edii.uclm.es/~useR-2013/Tutorials/kuhn/user_caret_2up.pdf
  - http://cran.r-project.org/web/packages/caret/vignettes/caret.pdf
- A paper introducing the caret package
  - http://www.jstatsoft.org/v28/i05/paper