

The caret package

Jeffrey Leek

May 18, 2016

The caret R package

the caret package



The **caret** package (short for Classification And REgression Training) 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)
```

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

```
## Loading required package: ggplot2
```

```
##
```

```
## Attaching package: 'kernlab'
```

```
## The following object is masked from 'package:ggplot2':
```

```
##
```

```
##      alpha
```

```
inTrain <- createDataPartition(y=spam$type,  
                                p=0.75, list=FALSE)
```

```
training <- spam[inTrain,]
```

```
testing <- spam[-inTrain,]
```

```
dim(training)
```

SPAM Example: Fit a model

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

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

SPAM Example: Final model

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

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or
```


SPAM Example: Prediction

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

```
##      [1] spam      nonspam spam      nonspam spam      spam      spam
##      [9] spam      spam      nonspam nonspam spam      spam      spam
##     [17] spam      nonspam spam      nonspam spam      spam      spam
##     [25] spam      spam      nonspam nonspam spam      spam      spam
##     [33] spam      spam      spam      spam      spam      spam      spam
##     [41] spam      spam      spam      nonspam spam      spam      spam
##     [49] spam      spam      spam      spam      spam      spam      spam
##     [57] nonspam nonspam spam      nonspam spam      nonspam spam
##     [65] spam      spam      spam      spam      spam      spam      spam
##     [73] spam      nonspam spam      nonspam spam      nonspam spam
##     [81] spam      spam      spam      spam      spam      spam      spam
##     [89] nonspam spam      spam      nonspam spam      spam      spam
##     [97] spam      spam      spam      spam      spam      spam      spam
##    [105] spam      spam      spam      spam      nonspam spam      spam
##   [113] spam      spam      nonspam spam      nonspam spam      spam
```

SPAM Example: Confusion Matrix

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

```
## Confusion Matrix and Statistics
```

```
##
```

```
##           Reference
```

```
## Prediction nonspam spam
```

```
##   nonspam      663    70
```

```
##   spam         34   383
```

```
##
```

```
##               Accuracy : 0.9096
```

```
##               95% CI : (0.8915, 0.9255)
```

```
##   No Information Rate : 0.6061
```

```
##   P-Value [Acc > NIR] : < 2.2e-16
```

```
##
```

```
##               Kappa : 0.8079
```

```
##   Mcnemar's Test P-Value : 0.0005991
```

```
##
```

```
##               Sensitivity : 0.9512
```

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>