

# Caret

Victor Garcia

## Introduction

Le packet caret ( est un ensemble de fonctions qui tentent de rationaliser le processus de création de modèles prédictifs. Le package contient des outils pour:

fractionnement des données prétraitement sélection de fonctionnalité réglage du modèle à l'aide du rééchantillonnage estimation de l'importance variable ainsi que d'autres fonctionnalités.

Il existe de nombreuses fonctions de modélisation différentes dans R. Certaines ont une syntaxe différente pour l'apprentissage du modèle et / ou la prédiction. Le package a commencé comme un moyen de fournir une interface uniforme aux fonctions elles-mêmes, ainsi qu'un moyen de standardiser les tâches courantes (tel que le réglage des paramètres et l'importance des variables).

## Visualisation

La fonction plot est un wrapper pour différents "lattice plots" pour visualiser les données. Par exemple, les figures suivantes montrent le tracé par défaut des résultats continus générés à l'aide de la featurePlot fonction.

Pour les ensembles de données de classification, les iris données sont utilisées à des fins d'illustration.

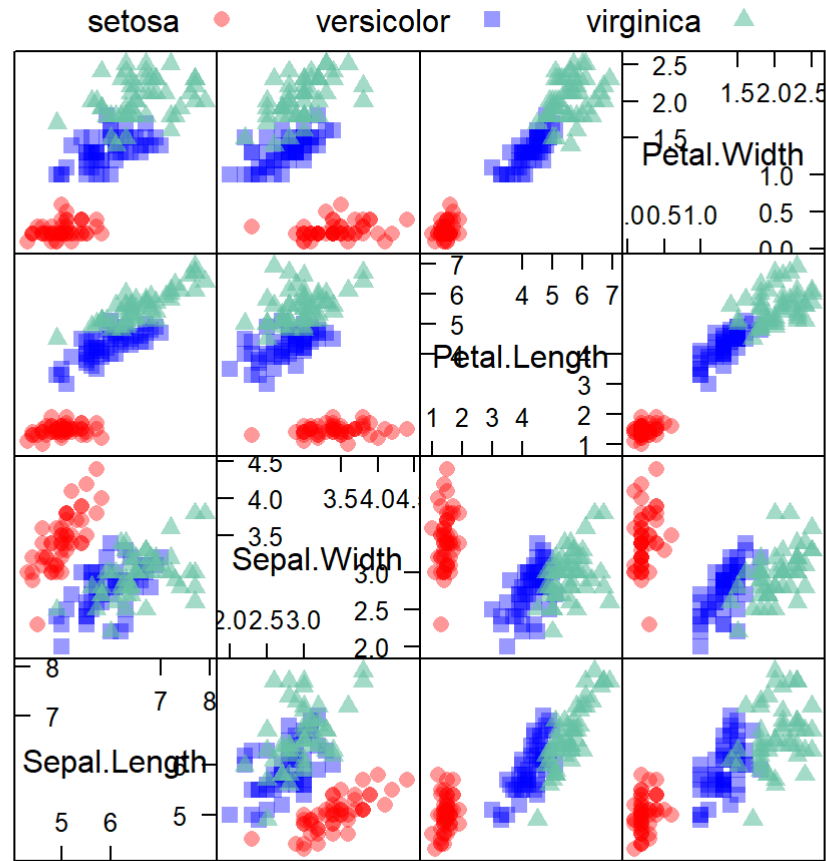
```
str(iris)
```

```
## 'data.frame': 150 obs. of 5 variables:
## $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
## $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
## $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
## $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
## $ Species : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
```

## Scatterplot Matrix

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

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

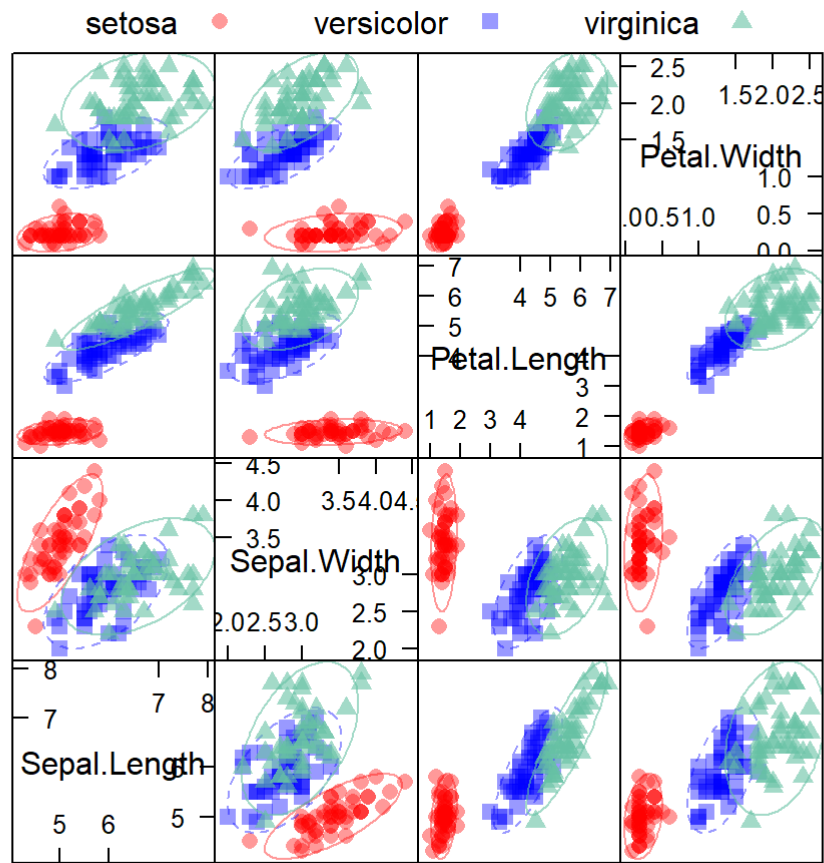


Scatter Plot Matrix

##Scatterplot Matrix avec Ellipses

```
featurePlot(x = iris[, 1:4],
            y = iris$Species,
            plot = "ellipse",

            auto.key = list(columns = 3))
```

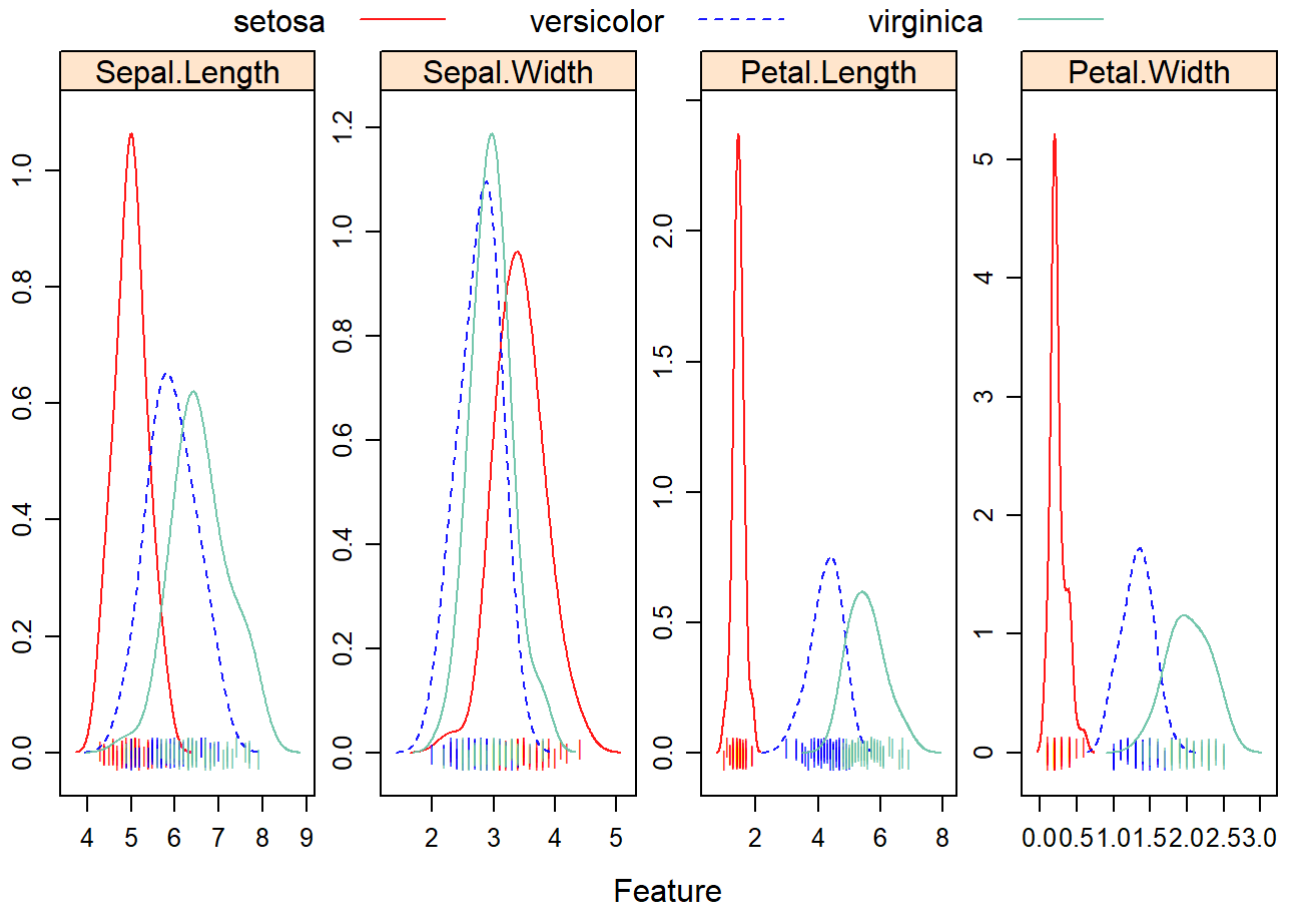


Scatter Plot Matrix

## ##Overlaid Density Plots

```
transparentTheme(trans = .9)
featurePlot(x = iris[, 1:4],
            y = iris$Species,
            plot = "density",

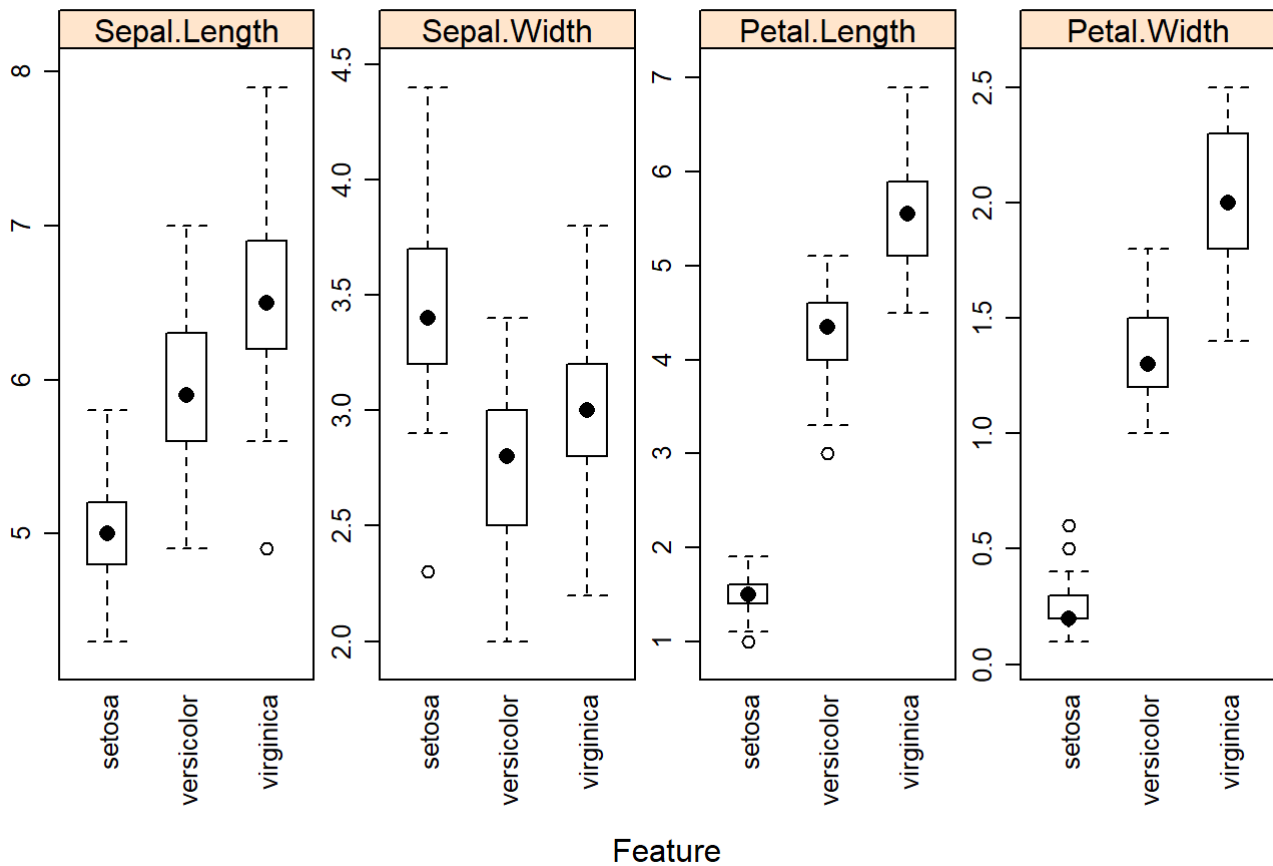
            scales = list(x = list(relation="free"),
                           y = list(relation="free")),
            adjust = 1.5,
            pch = "|",
            layout = c(4, 1),
            auto.key = list(columns = 3))
```



## Box Plots

```
featurePlot(x = iris[, 1:4],
            y = iris$Species,
            plot = "box",

            scales = list(y = list(relation="free"),
                          x = list(rot = 90)),
            layout = c(4,1 ),
            auto.key = list(columns = 2))
```



## ## Scatter Plots

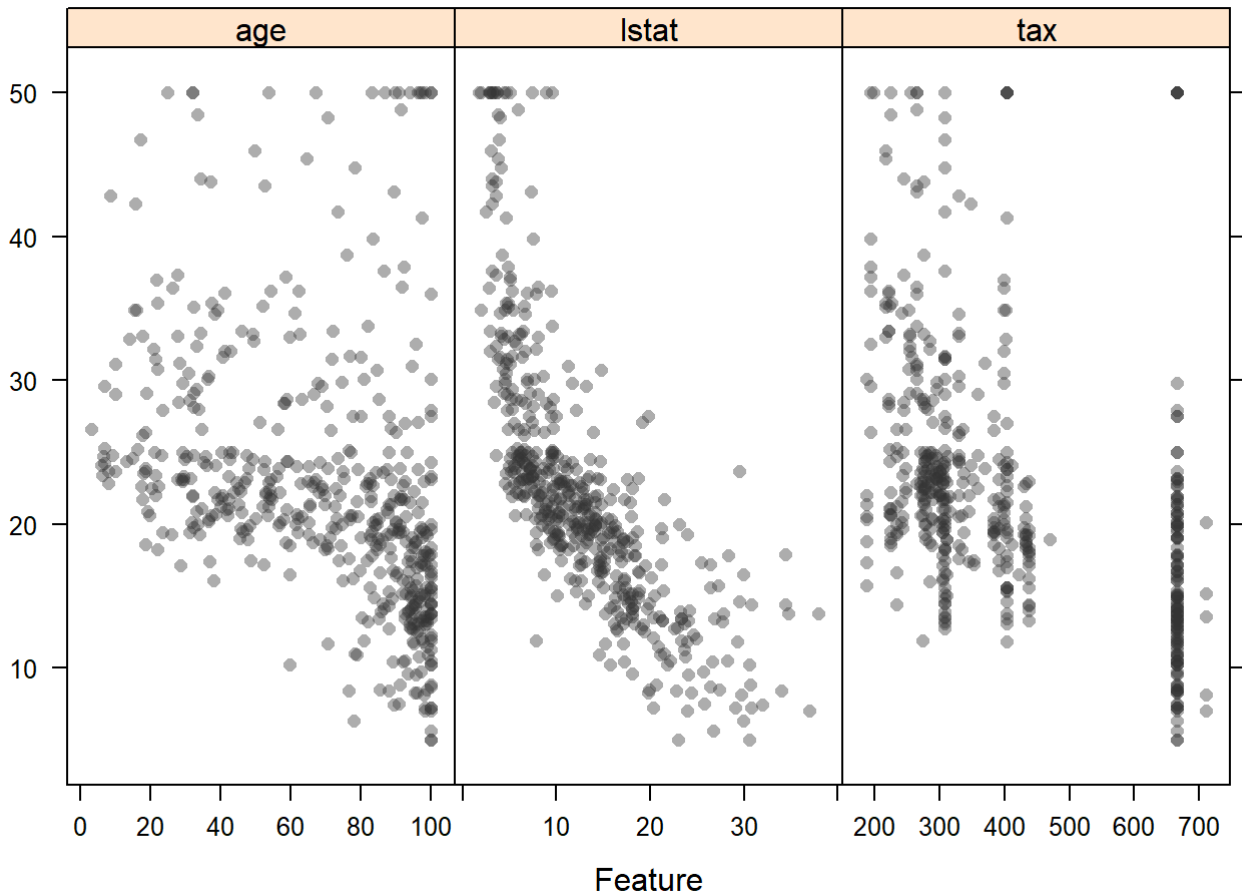
```
#install.packages("mlbench")
library(mlbench)
data(BostonHousing)
regVar <- c("age", "lstat", "tax")
str(BostonHousing[, regVar])
```

```
## 'data.frame':  506 obs. of  3 variables:
## $ age : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
## $ lstat: num  4.98 9.14 4.03 2.94 5.33 ...
## $ tax : num  296 242 242 222 222 222 311 311 311 311 ...
```

Lorsque les prédictions sont continues, featurePlot peut être utilisé pour créer des nuages de points qui permettront avec le résultat de pouvoir faire des prédictions.

Par exemple:

```
theme1 <- trellis.par.get()
theme1$plot.symbol$col = rgb(.2, .2, .2, .4)
theme1$plot.symbol$pch = 16
theme1$plot.line$col = rgb(1, 0, 0, .7)
theme1$plot.line$lwd <- 2
trellis.par.set(theme1)
featurePlot(x = BostonHousing[, regVar],
            y = BostonHousing$medv,
            plot = "scatter",
            layout = c(3, 1))
```



Notez que les échelles de l'axe des x sont différentes. La fonction utilise automatiquement `scale = list` (`y = list` (relation = "free")) afin que vous n'ayez pas à l'ajouter. On peut également passer des options à la fonction `treillis xyplot`. Par exemple, nous pouvons ajouter un nuage de points plus lisse en passant de nouvelles options :

```
featurePlot(x = BostonHousing[, regVar],
            y = BostonHousing$medv,
            plot = "scatter",
            type = c("p", "smooth"),
            span = .5,
            layout = c(3, 1))
```

