R 机器学习初探

班马

Saturday, August 12, 2017

本周内容：基于 caret 的 R 机器学习入门

教材位置：

http://www.10tiao.com/html/403/201706/2650629555/1.html

MATLAB 机器学习宣传手册
https://github.com/simon19891101/Books/tree/master/MATLAB

http://pan.baidu.com/s/1c0FRdW 《R for Data Science - Import, Tidy, Transform, Visualize and Model Data》(感谢@想分享)

http://pan.baidu.com/s/1gf6Bw35 《R 实战第二版中文》(感谢@想分享)

网易公开课： 用户名 fairyha456@163.com 密码 datakong1234

学习要求：

1.了解机器学习的概念，目的
2.在 R 中安装 caret package
3. 对 iris 数据集进行预处理
4. 将数据集分割为训练集和测试集
5. 使用随机森林(random forest)对 iris 中的任意 feature 进行建模并预测
6. 使用交叉验证(cross validation)预估模型性能

考核方式：

使用 R Markdown 输出 pdf 报告至 github 中，报告需包含对机器学习入门认识，对随机森林算法的认识，对 iris 数据集的探索分析，使用 caret 的建模并评估的代码以及结果

正文：

PART 1:

对 iris 数据集的探索分析：

caret: Classification and Regression Training 分类与回归训练
Misc functions for training and plotting classification and regression models.

加载包，准备数据
install.packages("caret")#安装包
library("caret")#加载包

```
data(iris)#数据准备
iris.data<-iris
str(iris.data)#查看数据集结构
```

```
## '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 ...
```

了解数据

查看数据集类别变量数据的分布

```
percentage<-prop.table(table(iris.data$Species))*100
cbind(freq=table(iris.data$Species),percentage=percentage)
```

```
##            freq percentage
## setosa       50   33.33333
## versicolor   50   33.33333
## virginica    50   33.33333
```

使用 head()函数或者 tail()函数进行预览数据集

```
  head(iris.data)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
```

```
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
  tail(iris.data)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 145          6.7         3.3          5.7         2.5 virginica
## 146          6.7         3.0          5.2         2.3 virginica
## 147          6.3         2.5          5.0         1.9 virginica
## 148          6.5         3.0          5.2         2.0 virginica
## 149          6.2         3.4          5.4         2.3 virginica
## 150          5.9         3.0          5.1         1.8 virginica
```

使用 summary 函数获取数据集摘要，对于数值型变量返回 5 个数字化特征：最小值，第一分位数，中位数，均值，第三分位数和最大值，因子型变量，返回每个类别的频数

```
summary(iris.data)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

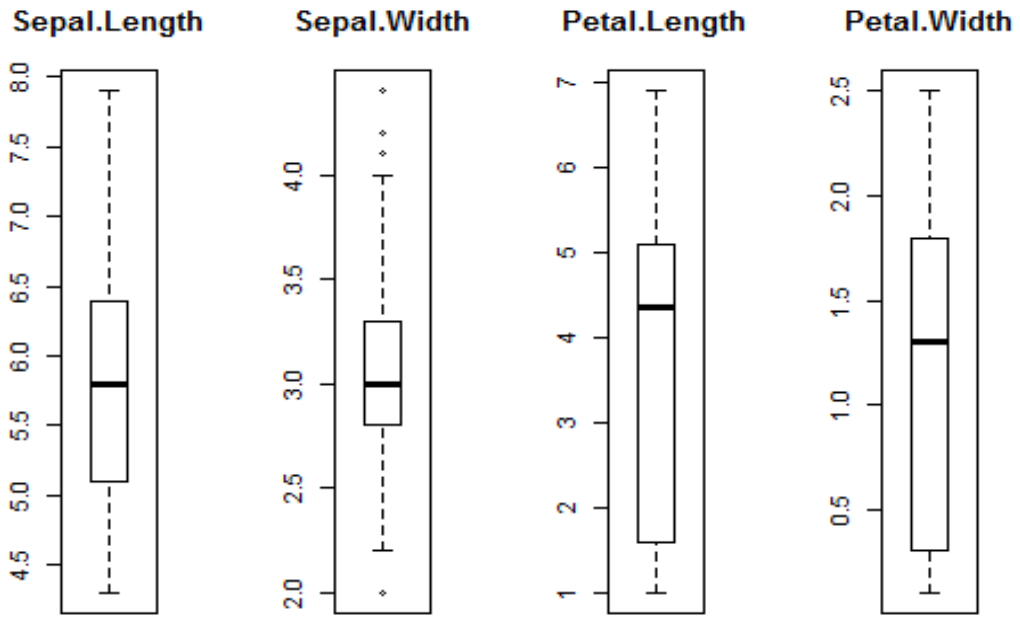
数据集可视化

单变量：了解每个属性的相关信息

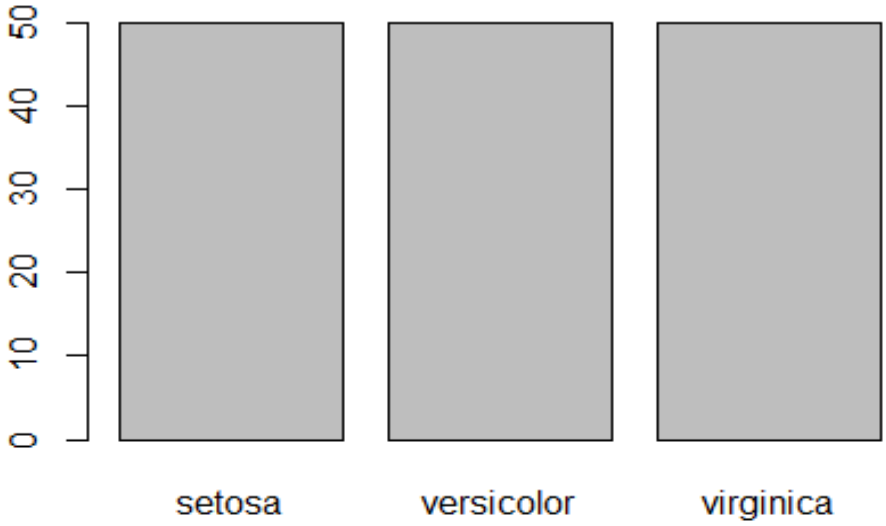
```
input.val<-iris.data[,1:4]
head(input.val)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1          5.1         3.5          1.4         0.2
## 2          4.9         3.0          1.4         0.2
## 3          4.7         3.2          1.3         0.2
## 4          4.6         3.1          1.5         0.2
## 5          5.0         3.6          1.4         0.2
## 6          5.4         3.9          1.7         0.4
```

```
par(mfrow=c(1,4))#绘图区域划分
for(i in 1:4)
{boxplot(input.val[,i],main=names(iris.data)[i])}
```

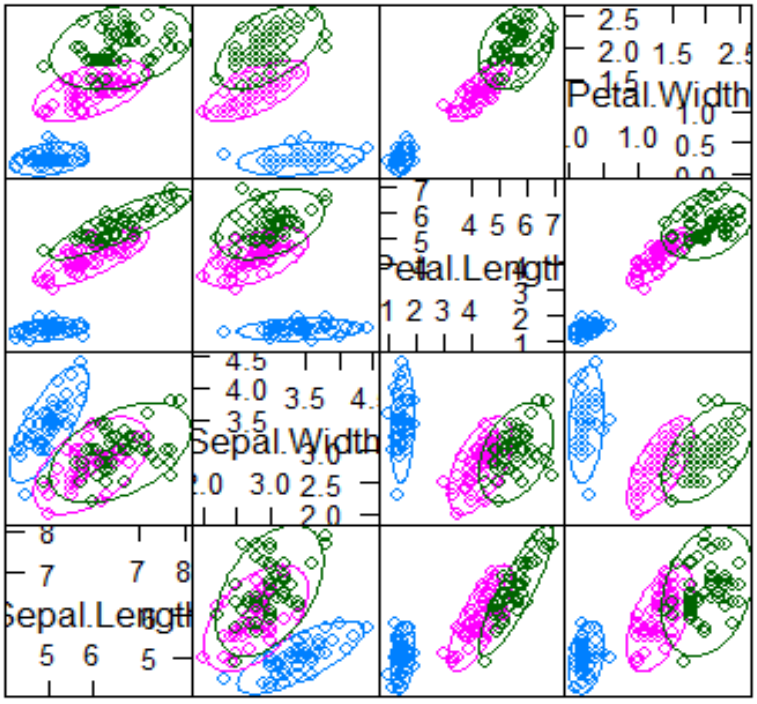



```
par(mfrow=c(1,1))
output.val<-iris.data[,5]
plot(output.val)
```



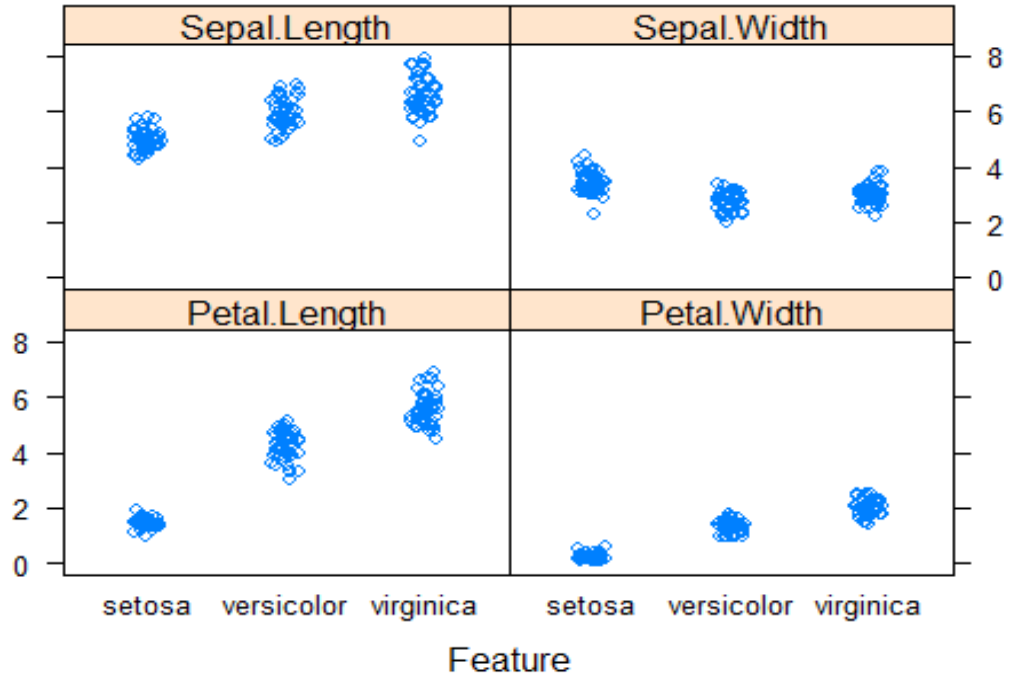
多变量：了解每个属性之间存在的关系

```
library(caret)

## Warning: package 'caret' was built under R version 3.1.3

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.1.3

library(ellipse)#椭圆包

## Warning: package 'ellipse' was built under R version 3.1.3

featurePlot(x=input.val,y=output.val,plot='ellipse')
```

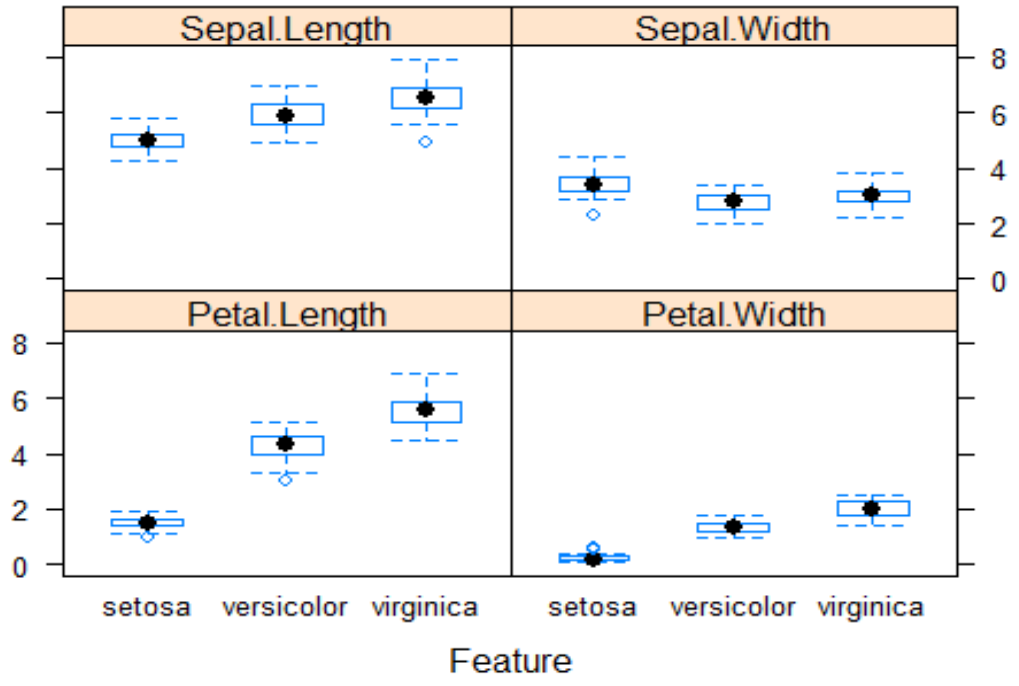


Scatter Plot Matrix

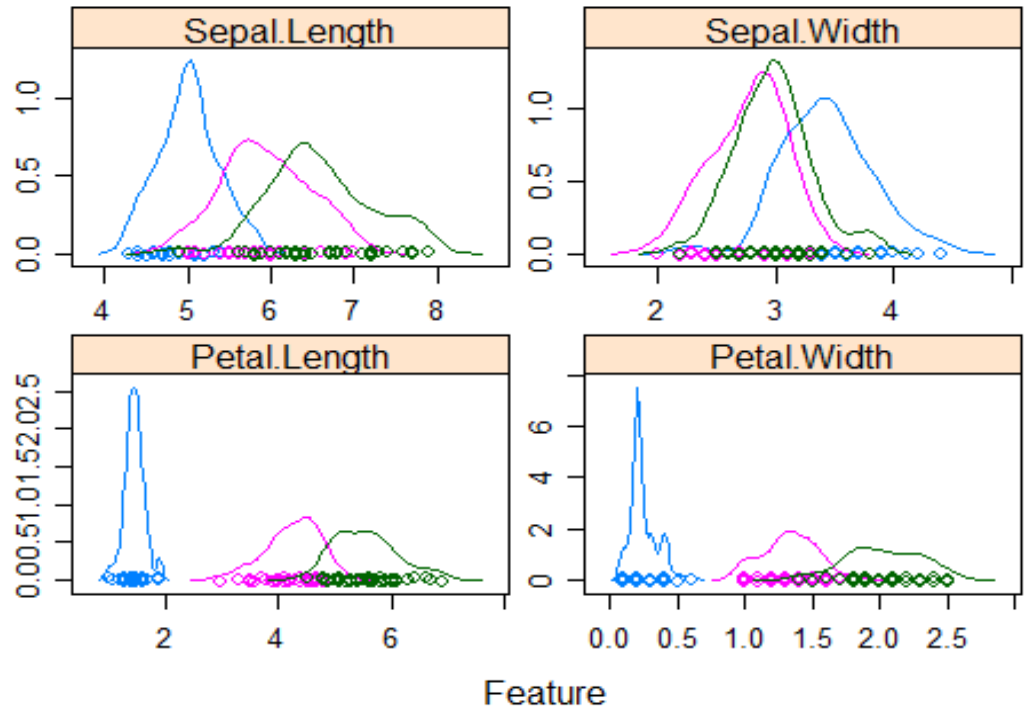
```
featurePlot(x=input.val,y=output.val,"strip",jitter=TRUE)
```




```
featurePlot(x=input.val,y=output.val,plot='box')
```




```
scales <- list(x=list(relation="free"), y=list(relation="free"))#分离层次的值
featurePlot(x=input.val,y=output.val,plot="density", scales=scales)#密度函数曲线
```



划分数据集： 分为训练数据集（80%）和验证数据集（20%）

```
validation.index<-createDataPartition(iris.data$Species,p=0.80,list=FALSE)
#获取原数据集的80%的行索引号？随机？
validation.data<-iris.data[-validation.index,]# 选择20%的数据用来验证模型
train.data <- iris.data[validation.index,]# 选择80%的数据用来训练和测试模型
```

10-折交叉验证

```
control<-trainControl(method="cv",number=10)
metric <- "Accuracy"#使用“精准度” 度量评估模型
#正确的预测实例的数量的比率，它通过正确的预测实例数除以总实例数乘于100%所得的一个百分数
```

构建模型：randomForest

```
library(e1071)

## Warning: package 'e1071' was built under R version 3.1.3

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.1.3

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

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

set.seed(7)
rf.model<-train(Species~.,data=train.data,method="rf",metric=metric,trControl=control)
```

查看模型精度

```
 pred.result <- predict(rf.model, validation.data)
 confusionMatrix(pred.result, validation.data$Species)

## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
```

```
##   setosa         10          0         0
##   versicolor      0         10         0
##   virginica       0          0        10
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.8843, 1)
##     No Information Rate : 0.3333
##     P-Value [Acc > NIR] : 4.857e-15
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virgi
nica
## Sensitivity                 1.0000            1.0000           1.
0000
## Specificity                 1.0000            1.0000           1.
0000
## Pos Pred Value              1.0000            1.0000           1.
0000
## Neg Pred Value              1.0000            1.0000           1.
0000
## Prevalence                  0.3333            0.3333           0.
3333
## Detection Rate              0.3333            0.3333           0.
3333
## Detection Prevalence        0.3333            0.3333           0.
3333
## Balanced Accuracy           1.0000            1.0000           1.
0000
```

PART 2:

机器学习入门认识：

通过本次作业，对机器学习入门有了比较入门级的认识，通过实操，对建模的过程也有了初步的了解。 但要针对具体的业务数据等进行建模还是有较大难度，还需加深学习。

随机森林算法的认识：

本以为要啃算法代码才可以用 RF 算法进行建模，看了教材之后原来导入包就可以！！挫败感减少了一丢丢。 在目前的理解层面来看的话，随机森林算法算是决策树的加强版。因为是多次分类，且取众数作为观测值的分类，精确度相对于决策树会高些。

对于教材上的内容暂没有很深的体会，先记一下以便后续加深学习：

随机森林的算法涉及对样本单元和变量进行抽样，从而生成大量决策树。对每个样本单元来 说，所有决策树依次对其进行分类。所有决策树预测类别中的众数类别即为随机森林所预测的这 一样本单元的类别。 假设训练集中共有 N 个样本单元，M 个变量，则随机森林算法如下。

(1) 从训练集中随机有放回地抽取 N 个样本单元，生成大量决策树。

(2) 在每一个节点随机抽取 m<M 个变量，将其作为分割该节点的候选变量。每一个节点处的 变量数应一致。

(3) 完整生成所有决策树，无需剪枝（最小节点为 1）。

(4) 终端节点的所属类别由节点对应的众数类别决定。

(5) 对于新的观测点，用所有的树对其进行分类，其类别由多数决定原则生成。生成树时没有用到的样本点所对应的类别可由生成的树估计，与其真实类别比较即可得到袋 外预测（out-of-bag，OOB）误差。无法获得验证集时，这是随机森林的一大优势。

随机森林算法可计算变量的相对重要程度相较于其他分类方法，随机森林的分类准确率通常更高。另外，随机森林算法可处理大规模 问题（即多样本单元、多变量），可处理训练集中有大量缺失值的数据，也可应对变量远多于样本 单元的数据。可计算袋外预测误差（OOB error）、度量变量重要性也是随机森林的两个明显优势。 随机森林的一个明显缺点是分类方法（此例中相当于 500 棵决策树）较难理解和表达。另外， 我们需要存储整个随机森林以对新样本单元分类。