Drug_Classification

Hanyu Chen

2023-02-08

```
#install.packages("caret")
library("caret")
```

```
## Warning: package 'caret' was built under R version 4.1.2
```

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

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

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

```
data<- read.csv("drug200.csv")
summary(data)
```

```
##       Age            Sex                BP            Cholesterol
##  Min.   :15.00   Length:200         Length:200         Length:200
##  1st Qu.:31.00   Class :character   Class :character   Class :character
##  Median :45.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :44.31
##  3rd Qu.:58.00
##  Max.   :74.00
##     Na_to_K           Drug
##  Min.   : 6.269   Length:200
##  1st Qu.:10.445   Class :character
##  Median :13.937   Mode  :character
##  Mean   :16.084
##  3rd Qu.:19.380
##  Max.   :38.247
```

```
which(is.na(data)) # no missing value
```

```
## integer(0)
```

```
dim(data) # shape of (162,6)
```

```
## [1] 200   6
```

```r
sapply(data,class) # check the data type of each column
```

```
##         Age         Sex          BP Cholesterol     Na_to_K        Drug
##   "integer" "character" "character" "character"   "numeric" "character"
```

```r
head(data)
```

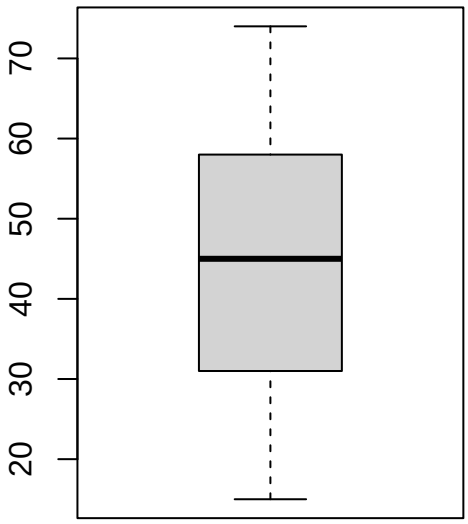
```
##   Age Sex     BP Cholesterol Na_to_K  Drug
## 1  23   F   HIGH        HIGH  25.355 DrugY
## 2  47   M    LOW        HIGH  13.093 drugC
## 3  47   M    LOW        HIGH  10.114 drugC
## 4  28   F NORMAL        HIGH   7.798 drugX
## 5  61   F    LOW        HIGH  18.043 DrugY
## 6  22   F NORMAL        HIGH   8.607 drugX
```

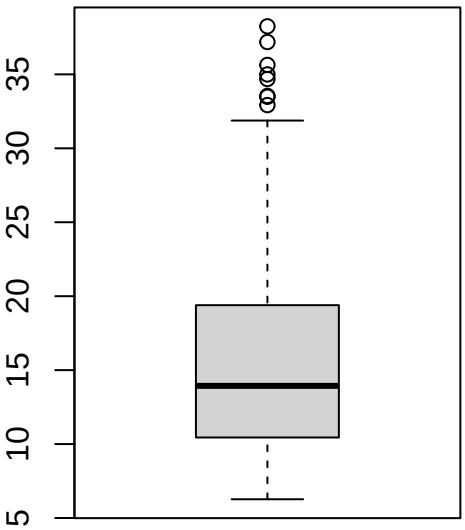
```r
data$Drug <-factor(data$Drug)
data$BP <- factor (data$BP)
data$Cholesterol <- factor(data$Cholesterol)
data$Sex <- factor(data$Sex)
summary(data)
```

```
##       Age        Sex          BP     Cholesterol     Na_to_K          Drug
##  Min.   :15.00   F: 96   HIGH  :77   HIGH  :103   Min.   : 6.269   drugA:23
##  1st Qu.:31.00   M:104   LOW   :64   NORMAL: 97   1st Qu.:10.445   drugB:16
##  Median :45.00           NORMAL:59                Median :13.937   drugC:16
##  Mean   :44.31                                    Mean   :16.084   drugX:54
##  3rd Qu.:58.00                                    3rd Qu.:19.380   DrugY:91
##  Max.   :74.00                                    Max.   :38.247
```

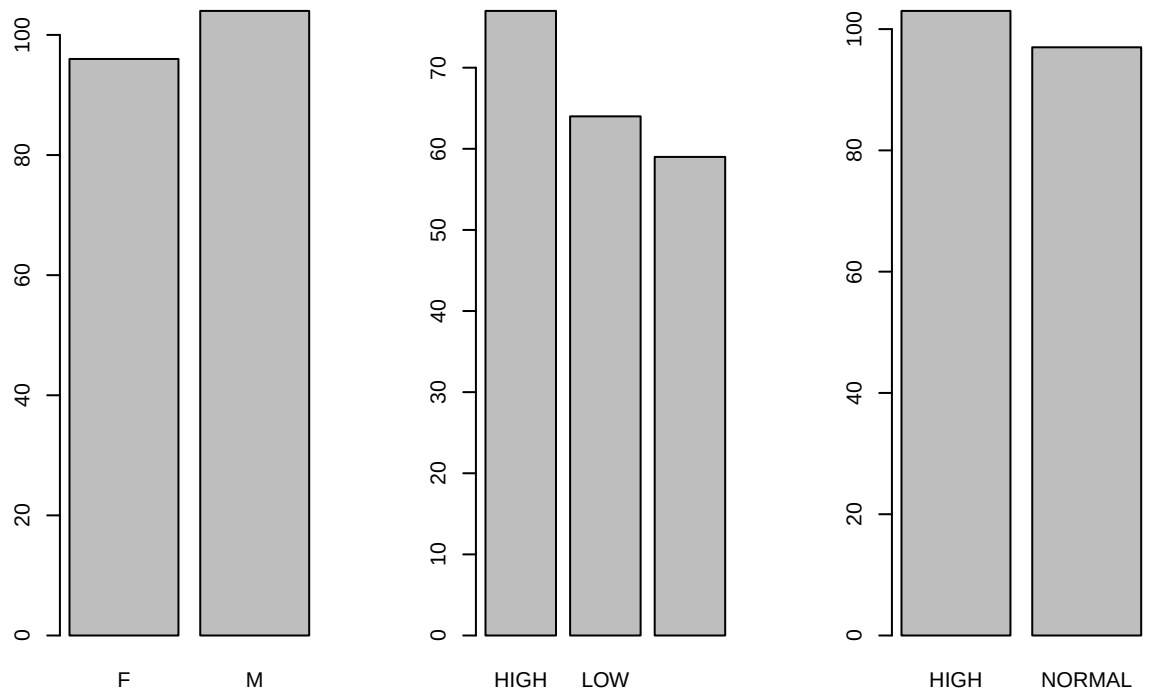
Visulization

```r
# Identify x and y
x <- data[,1:5]
y <- data[,6]
# boxplots to see distribution
par(mfrow = c(1:2))
boxplot(x$Age)
boxplot(x$Na_to_K)
```






```
par(mfrow=c(1,3))
title <- c("Sex","BP","Cholesterol")
  for (i in x){
    if (class(i) =="factor"){
      barplot(table(i))
  }
  }
```




```
#install.packages("fastDummies")
library("fastDummies")
data <- dummy_cols(data, select_columns = c("Sex","Cholesterol","BP"))
data <- subset(data,select = -c(Sex, Cholesterol, BP))
```

```
# Randomly create a list of 80% of the index that used for training
validation_index <- createDataPartition(data$Drug, p = 0.80, list = FALSE)
# 20% of the data used for validation
validation <- data[-validation_index,]
# 80% used to train and test
data<- data[validation_index,]
```

```
control <- trainControl(method = 'cv',
                        summaryFunction = defaultSummary,
                        number = 10,
                        savePredictions = TRUE)
metric <- 'Accuracy'
```

Build Models

```
set.seed(37)
knn_m <- train(Drug~.,
               data = data,
               method = "knn",
               metric = metric,
```

```
               trControl = control,
               tuneGrid = data.frame(k = seq(10,30,by = 1))) # Cross-Validation)
knn_m
```

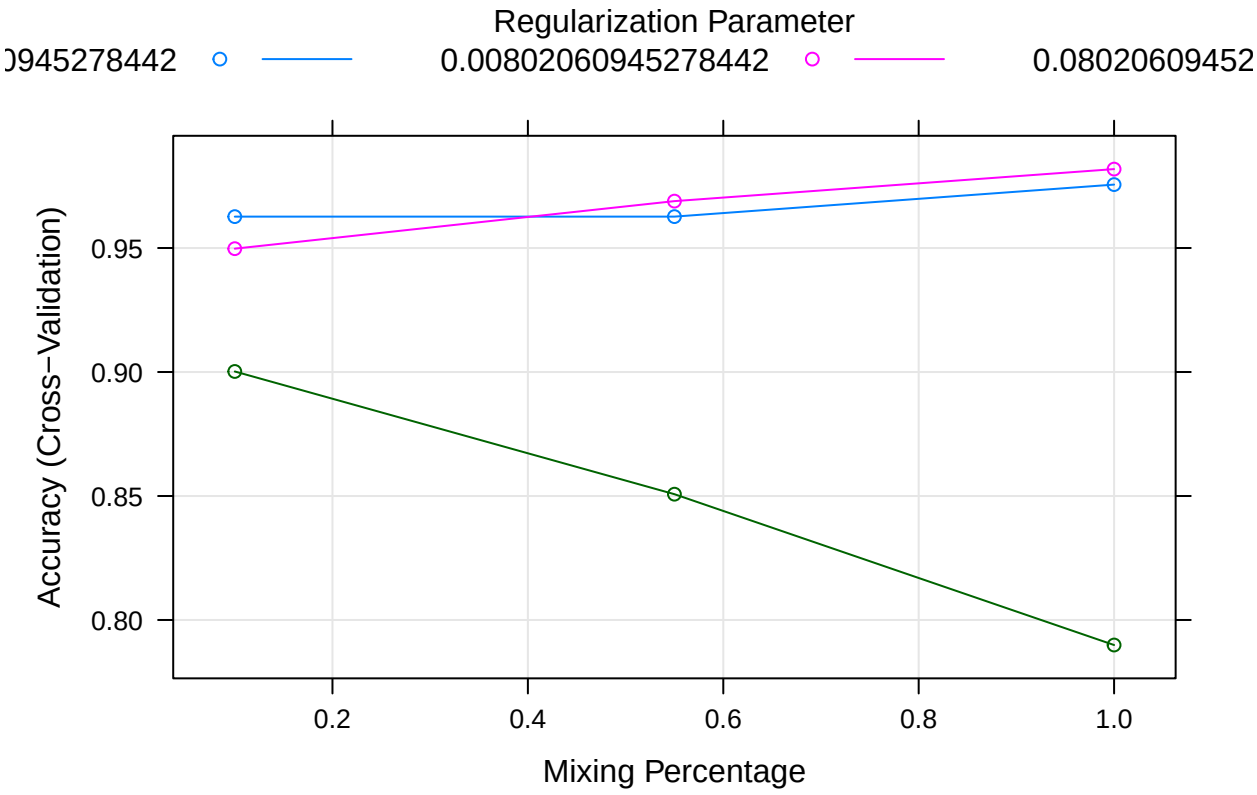
```
## k-Nearest Neighbors
##
## 162 samples
##   9 predictor
##   5 classes: 'drugA', 'drugB', 'drugC', 'drugX', 'DrugY'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 143, 146, 145, 145, 147, 146, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##   10  0.6775580  0.5361731
##   11  0.6728277  0.5214784
##   12  0.6641757  0.5076486
##   13  0.6283978  0.4557503
##   14  0.6473994  0.4858073
##   15  0.6809520  0.5266367
##   16  0.6600142  0.4943259
##   17  0.6667299  0.5044062
##   18  0.6598491  0.4953382
##   19  0.6535991  0.4827737
##   20  0.6545369  0.4836870
##   21  0.6670859  0.5031784
##   22  0.6733849  0.5116576
##   23  0.6719324  0.5021127
##   24  0.6402167  0.4569348
##   25  0.6405844  0.4589558
##   26  0.6588132  0.4801243
##   27  0.6458965  0.4595873
##   28  0.6588132  0.4775824
##   29  0.6476677  0.4588751
##   30  0.6455289  0.4546436
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 15.
```

```
set.seed(37)
glm_m <- train(Drug~.,
               data = data,
               method = "glmnet",
               metric = metric,
               trControl = control
              )
glm_m
```

```
## glmnet
##
## 162 samples
```

```
##   9 predictor
##   5 classes: 'drugA', 'drugB', 'drugC', 'drugX', 'DrugY'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 143, 146, 145, 145, 147, 146, ...
## Resampling results across tuning parameters:
##
##   alpha  lambda        Accuracy   Kappa
##   0.10   0.0008020609  0.9626535  0.9463588
##   0.10   0.0080206095  0.9497368  0.9290348
##   0.10   0.0802060945  0.9002090  0.8503392
##   0.55   0.0008020609  0.9626535  0.9463588
##   0.55   0.0080206095  0.9689035  0.9553454
##   0.55   0.0802060945  0.8507301  0.7712574
##   1.00   0.0008020609  0.9755702  0.9648750
##   1.00   0.0080206095  0.9818202  0.9740704
##   1.00   0.0802060945  0.7899033  0.6754540
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were alpha = 1 and lambda = 0.008020609.
```

```
plot(glm_m)
```



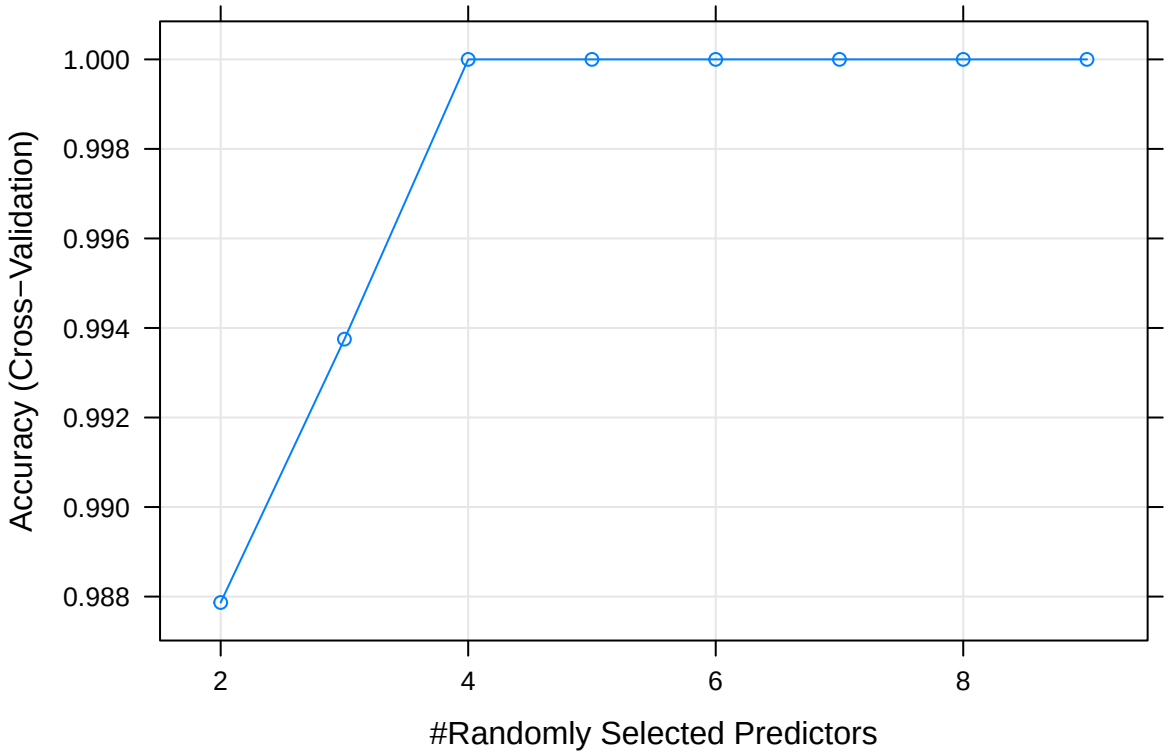
```
set.seed(37)
rf_m <- train(Drug~.,
               data = data,
               method = "rf",
               metric = metric,
               trControl = control,
              tuneLength = 30
              )
```

```
## note: only 8 unique complexity parameters in default grid. Truncating the grid to 8 .
```

```
# mtry = 3
print(rf_m)
```

```
## Random Forest
##
## 162 samples
##   9 predictor
##   5 classes: 'drugA', 'drugB', 'drugC', 'drugX', 'DrugY'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 143, 146, 145, 145, 147, 146, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   2     0.9878676  0.9826602
##   3     0.9937500  0.9911602
##   4     1.0000000  1.0000000
##   5     1.0000000  1.0000000
##   6     1.0000000  1.0000000
##   7     1.0000000  1.0000000
##   8     1.0000000  1.0000000
##   9     1.0000000  1.0000000
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 4.
```

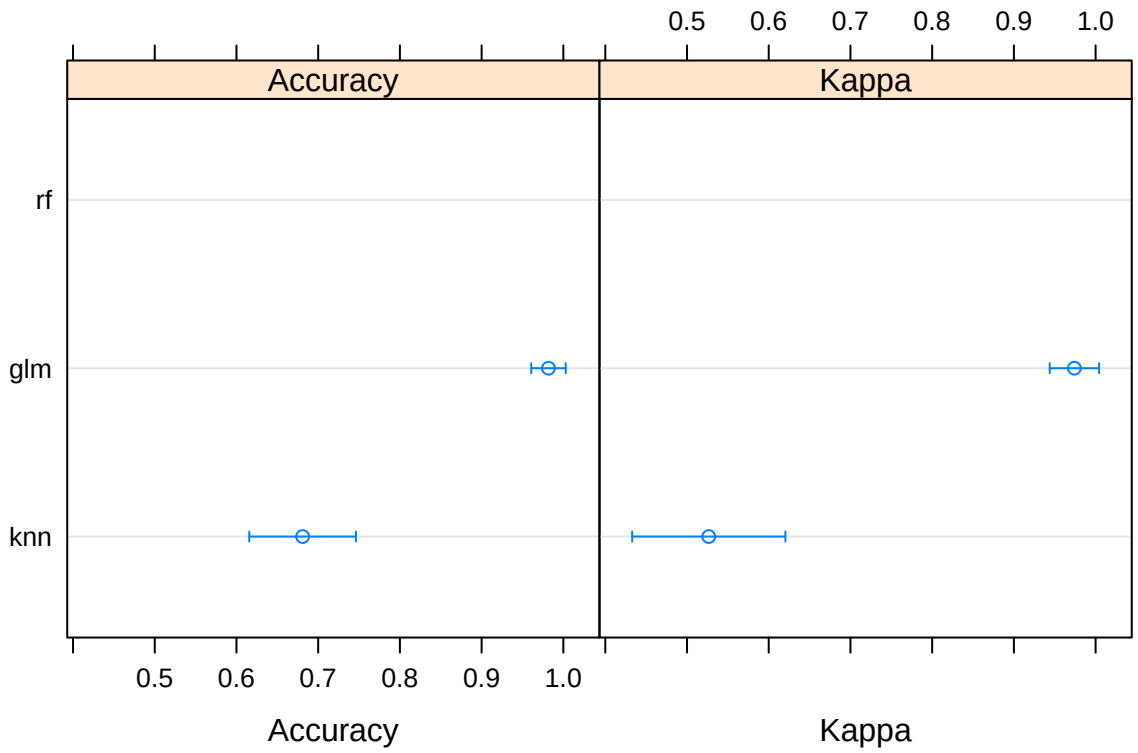
```
plot(rf_m)
```




```
results <- resamples (list(knn = knn_m, rf = rf_m, glm = glm_m))
print(summary(results))
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: knn, rf, glm
## Number of resamples: 10
##
## Accuracy
##          Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
## knn 0.5625000 0.6062500 0.6595395 0.6809520 0.7610294  0.8    0
## rf  1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  1.0    0
## glm 0.9333333 0.9605263 1.0000000 0.9818202 1.0000000  1.0    0
##
## Kappa
##          Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
## knn 0.3411765 0.4174350 0.5117162 0.5266367 0.6296935  0.7    0
## rf  1.0000000 1.0000000 1.0000000 1.0000000 1.0000000  1.0    0
## glm 0.9058824 0.9464286 1.0000000 0.9740704 1.0000000  1.0    0
```

```
dotplot(results) # Random Forest is the best
```



Confidence Level: 0.95

```
pred <- predict(rf_m, validation)
confusionMatrix(pred,validation$Drug)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction drugA drugB drugC drugX DrugY
##      drugA     4     1     0     0     0
##      drugB     0     2     0     0     0
##      drugC     0     0     3     0     0
##      drugX     0     0     0     9     0
##      DrugY     0     0     0     1    18
##
## Overall Statistics
##
##                Accuracy : 0.9474
##                  95% CI : (0.8225, 0.9936)
##     No Information Rate : 0.4737
##     P-Value [Acc > NIR] : 4.248e-10
##
##                   Kappa : 0.9222
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
```

```
##
##                      Class: drugA Class: drugB Class: drugC Class: drugX
## Sensitivity               1.0000      0.66667      1.00000       0.9000
## Specificity               0.9706      1.00000      1.00000       1.0000
## Pos Pred Value            0.8000      1.00000      1.00000       1.0000
## Neg Pred Value            1.0000      0.97222      1.00000       0.9655
## Prevalence                0.1053      0.07895      0.07895       0.2632
## Detection Rate            0.1053      0.05263      0.07895       0.2368
## Detection Prevalence      0.1316      0.05263      0.07895       0.2368
## Balanced Accuracy         0.9853      0.83333      1.00000       0.9500
##                      Class: DrugY
## Sensitivity               1.0000
## Specificity               0.9500
## Pos Pred Value            0.9474
## Neg Pred Value            1.0000
## Prevalence                0.4737
## Detection Rate            0.4737
## Detection Prevalence      0.5000
## Balanced Accuracy         0.9750
```

Random Forest model gets an accuracy of 0.975