R Notebook

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Classify the wine dataset with KNN

Step 1: Download data——

Step 2: Exploring and preparing the data —-

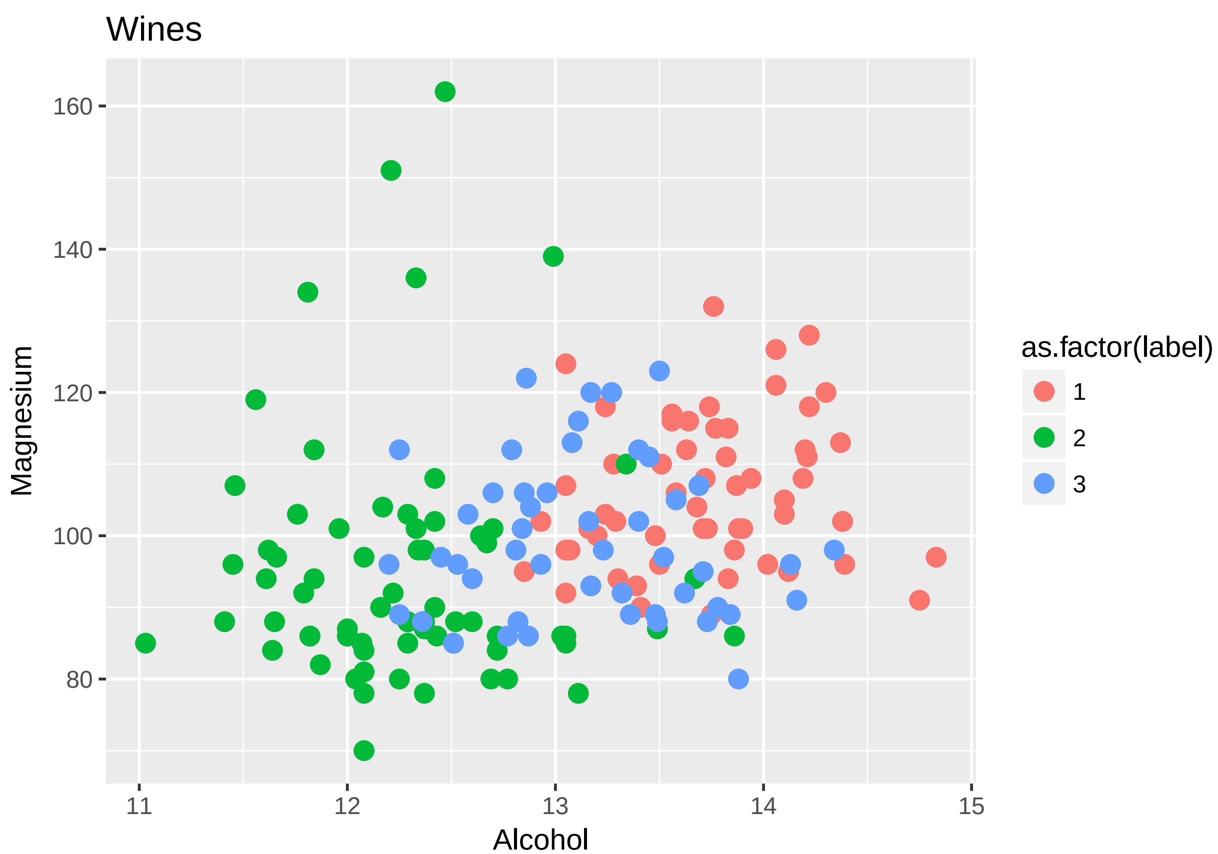
```
# import the CSV file
wines<- read.csv("wines.csv")
names(wines) <- c("label",
                  "Alcohol",
                  "Malic_acid",
                  "Ash",
                  "Alcalinity_of_ash",
                  "Magnesium",
                  "Total_phenols",
                  "Flavanoids",
                  "Nonflavanoid_phenols",
                  "Proanthocyanins",
                  "Color_intensity",
                  "Hue",
                  "OD280_OD315_of_diluted_wines",
                  "Proline")
head(wines)
```

```
##   label Alcohol Malic_acid  Ash Alcalinity_of_ash Magnesium Total_phenols
## 1     1   13.20       1.78 2.14              11.2       100          2.65
## 2     1   13.16       2.36 2.67              18.6       101          2.80
## 3     1   14.37       1.95 2.50              16.8       113          3.85
## 4     1   13.24       2.59 2.87              21.0       118          2.80
## 5     1   14.20       1.76 2.45              15.2       112          3.27
## 6     1   14.39       1.87 2.45              14.6        96          2.50
##   Flavanoids Nonflavanoid_phenols Proanthocyanins Color_intensity  Hue
## 1       2.76                 0.26            1.28            4.38 1.05
## 2       3.24                 0.30            2.81            5.68 1.03
## 3       3.49                 0.24            2.18            7.80 0.86
## 4       2.69                 0.39            1.82            4.32 1.04
## 5       3.39                 0.34            1.97            6.75 1.05
## 6       2.52                 0.30            1.98            5.25 1.02
##   OD280_OD315_of_diluted_wines Proline
## 1                         3.40    1050
## 2                         3.17    1185
## 3                         3.45    1480
```

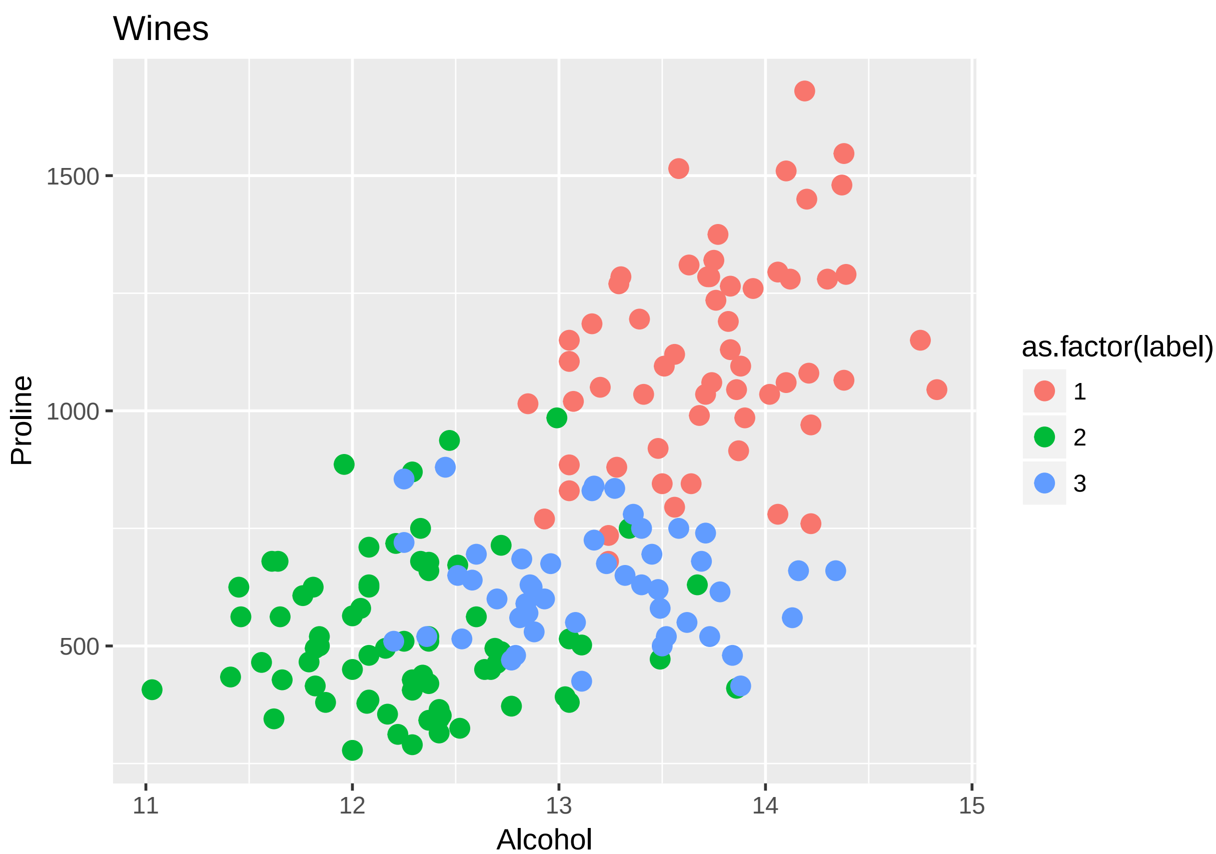
```
## 4                              2.93      735
## 5                              2.85     1450
## 6                              3.58     1290
```

```r
# examine the structure of the wbcd data frame
#str(wines)
```

```r
library(ggplot2)
plt1 <- ggplot(wines, aes(x = Alcohol, y = Magnesium, colour = as.factor(label))) +
geom_point(size=3) +
ggtitle("Wines")
plt2 <- ggplot(wines, aes(x = Alcohol, y = Proline, colour = as.factor(label))) +
geom_point(size=3) +
ggtitle("Wines")
plt1
```




```r
plt2
```




```
# table of labels
table(wines$label)
```

```
##
##  1  2  3
## 58 71 48
```

```
# recode label as a factor
wines$label <- factor(wines$label, levels = c("1","2","3"),
                      labels = c("A", "B", "C"))

# table or proportions with more informative labels
round(prop.table(table(wines$label)) * 100, digits = 1)
```

```
##
##    A    B    C
## 32.8 40.1 27.1
```

```
# create normalization function
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

# normalize the wbcd data
wines_n <- as.data.frame(lapply(wines[2:14], normalize))
# add the label column
wines_n$label<-as.factor(wines[,1])
```

```r
# set up trainning and test data sets
set.seed(3033)
indx <- sample(1:nrow(wines_n), as.integer(0.7*nrow(wines_n)))
#View(indx)
#indx

training <- wines_n[indx,1:13]
testing <- wines_n[-indx,1:13]

train_labels <- wines_n[indx,14]
test_labels <- wines_n[-indx,14]
```

```r
trl<-wines_n[indx,]
tel<-wines_n[-indx,]
```

```r
# check missing data
anyNA(training)
```

```
## [1] FALSE
```

```r
anyNA(testing)
```

```
## [1] FALSE
```

Step 3: Training a model on the data —-

```r
# load the "class" library
library(class)

test_pred <- knn(train = training[1:13], test = testing[1:13],
                      cl = train_labels, k = 4)

head(testing)
```

```
##      Alcohol Malic_acid       Ash Alcalinity_of_ash Magnesium
## 1  0.5710526  0.2055336 0.4171123        0.03092784 0.3260870
## 4  0.5815789  0.3656126 0.8074866        0.53608247 0.5217391
## 13 0.9789474  0.1956522 0.5508021        0.04123711 0.2282609
## 15 0.6842105  0.2114625 0.7165775        0.34020619 0.4565217
## 16 0.8605263  0.2332016 0.7272727        0.48453608 0.5434783
## 18 0.8315789  0.1679842 0.5989305        0.30412371 0.4130435
##    Total_phenols Flavanoids Nonflavanoid_phenols Proanthocyanins
## 1      0.5758621  0.5105485            0.2452830       0.2744479
## 4      0.6275862  0.4957806            0.4905660       0.4447950
## 13     0.7310345  0.7067511            0.5660377       0.7570978
## 15     0.6448276  0.5421941            0.3207547       0.3312303
## 16     0.6275862  0.5907173            0.3773585       0.4921136
## 18     0.8000000  0.7573840            0.3584906       0.4574132
##    Color_intensity       Hue OD280_OD315_of_diluted_wines   Proline
## 1        0.2645051 0.4634146                    0.7802198 0.5506419
## 4        0.2593857 0.4552846                    0.6080586 0.3259629
## 13       0.3515358 0.6260163                    0.5347985 0.6219686
## 15       0.5136519 0.6504065                    0.5897436 0.7360913
```

```
## 16       0.4197952 0.4796748                    0.5054945 0.7146933
## 18       0.6331058 0.6097561                    0.5677656 1.0000000
```

```r
head(test_pred)
```

```
## [1] A A A A A A
## Levels: A B C
```

Step 4: Evaluating model performance —-

```r
# load the "gmodels" library
library(gmodels)

# Create the cross tabulation of predicted vs. actual
CrossTable(x = test_labels, y = test_pred,
           prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  54
##
##
##              | test_pred
##  test_labels |         A |         B |         C | Row Total |
## -------------|-----------|-----------|-----------|-----------|
##            A |        15 |         0 |         0 |        15 |
##              |     1.000 |     0.000 |     0.000 |     0.278 |
##              |     0.938 |     0.000 |     0.000 |           |
##              |     0.278 |     0.000 |     0.000 |           |
## -------------|-----------|-----------|-----------|-----------|
##            B |         1 |        22 |         0 |        23 |
##              |     0.043 |     0.957 |     0.000 |     0.426 |
##              |     0.062 |     0.957 |     0.000 |           |
##              |     0.019 |     0.407 |     0.000 |           |
## -------------|-----------|-----------|-----------|-----------|
##            C |         0 |         1 |        15 |        16 |
##              |     0.000 |     0.062 |     0.938 |     0.296 |
##              |     0.000 |     0.043 |     1.000 |           |
##              |     0.000 |     0.019 |     0.278 |           |
## -------------|-----------|-----------|-----------|-----------|
## Column Total |        16 |        23 |        15 |        54 |
##              |     0.296 |     0.426 |     0.278 |           |
## -------------|-----------|-----------|-----------|-----------|
##
```

```
##
```

```r
#we can see the error is not very high, only 2 mistake.
```

Step 5: Improving model performance —-

```r
# use the scale() function to z-score standardize a data frame
wines_z <- as.data.frame(scale(wines[2:14]))
# add the label column
wines_z$label<-as.factor(wines[,1])
```

```r
# set up trainning and test data sets
set.seed(3033)
indx <- sample(1:nrow(wines_z), as.integer(0.7*nrow(wines_z)))
#indx

training1 <- wines_z[indx,1:13]
testing1 <- wines_z[-indx,1:13]

train_labels1 <- wines_z[indx,14]
test_labels1 <- wines_z[-indx,14]
```

```r
# check missing data
anyNA(training1)
```

```
## [1] FALSE
```

```r
anyNA(testing1)
```

```
## [1] FALSE
```

```r
#start time
strt<-Sys.time()

test_pred <- knn(train = training[1:13], test = testing[1:13], cl = train_labels, k = 1)
CrossTable(x = test_labels, y = test_pred,prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  54
##
##
##              | test_pred
##  test_labels |         A |         B |         C | Row Total |
## -------------|-----------|-----------|-----------|-----------|
##            A |        15 |         0 |         0 |        15 |
```

```
##               |     1.000 |     0.000 |     0.000 |     0.278 |
##               |     0.833 |     0.000 |     0.000 |           |
##               |     0.278 |     0.000 |     0.000 |           |
## -------------|-----------|-----------|-----------|-----------|
##             B |         3 |        20 |         0 |        23 |
##               |     0.130 |     0.870 |     0.000 |     0.426 |
##               |     0.167 |     1.000 |     0.000 |           |
##               |     0.056 |     0.370 |     0.000 |           |
## -------------|-----------|-----------|-----------|-----------|
##             C |         0 |         0 |        16 |        16 |
##               |     0.000 |     0.000 |     1.000 |     0.296 |
##               |     0.000 |     0.000 |     1.000 |           |
##               |     0.000 |     0.000 |     0.296 |           |
## -------------|-----------|-----------|-----------|-----------|
## Column Total |        18 |        20 |        16 |        54 |
##               |     0.333 |     0.370 |     0.296 |           |
## -------------|-----------|-----------|-----------|-----------|
##
##
```

```r
test_pred <- knn(train = training[1:13], test = testing[1:13], cl = train_labels, k=4)
CrossTable(x = test_labels, y = test_pred, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  54
##
##
##               | test_pred
##  test_labels |         A |         B |         C | Row Total |
## -------------|-----------|-----------|-----------|-----------|
##             A |        15 |         0 |         0 |        15 |
##               |     1.000 |     0.000 |     0.000 |     0.278 |
##               |     0.938 |     0.000 |     0.000 |           |
##               |     0.278 |     0.000 |     0.000 |           |
## -------------|-----------|-----------|-----------|-----------|
##             B |         1 |        22 |         0 |        23 |
##               |     0.043 |     0.957 |     0.000 |     0.426 |
##               |     0.062 |     0.957 |     0.000 |           |
##               |     0.019 |     0.407 |     0.000 |           |
## -------------|-----------|-----------|-----------|-----------|
##             C |         0 |         1 |        15 |        16 |
##               |     0.000 |     0.062 |     0.938 |     0.296 |
##               |     0.000 |     0.043 |     1.000 |           |
##               |     0.000 |     0.019 |     0.278 |           |
## -------------|-----------|-----------|-----------|-----------|
```

```
## Column Total |        16 |        23 |        15 |        54 |
##              |     0.296 |     0.426 |     0.278 |           |
## -------------|-----------|-----------|-----------|-----------|
##
##
```

```
test_pred <- knn(train = training[1:13], test = testing[1:13], cl = train_labels, k = 8)
CrossTable(x = test_labels, y = test_pred,prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  54
##
##
##              | test_pred
##  test_labels |         A |         B |         C | Row Total |
## -------------|-----------|-----------|-----------|-----------|
##            A |        15 |         0 |         0 |        15 |
##              |     1.000 |     0.000 |     0.000 |     0.278 |
##              |     0.882 |     0.000 |     0.000 |           |
##              |     0.278 |     0.000 |     0.000 |           |
## -------------|-----------|-----------|-----------|-----------|
##            B |         2 |        21 |         0 |        23 |
##              |     0.087 |     0.913 |     0.000 |     0.426 |
##              |     0.118 |     1.000 |     0.000 |           |
##              |     0.037 |     0.389 |     0.000 |           |
## -------------|-----------|-----------|-----------|-----------|
##            C |         0 |         0 |        16 |        16 |
##              |     0.000 |     0.000 |     1.000 |     0.296 |
##              |     0.000 |     0.000 |     1.000 |           |
##              |     0.000 |     0.000 |     0.296 |           |
## -------------|-----------|-----------|-----------|-----------|
## Column Total |        17 |        21 |        16 |        54 |
##              |     0.315 |     0.389 |     0.296 |           |
## -------------|-----------|-----------|-----------|-----------|
##
##
```

```
test_pred <- knn(train = training[1:13], test = testing[1:13], cl = train_labels, k=12)
CrossTable(x = test_labels, y = test_pred, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
```

```
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  54
##
##
##              | test_pred
##  test_labels |         A |         B |         C | Row Total |
## -------------|-----------|-----------|-----------|-----------|
##            A |        15 |         0 |         0 |        15 |
##              |     1.000 |     0.000 |     0.000 |     0.278 |
##              |     0.938 |     0.000 |     0.000 |           |
##              |     0.278 |     0.000 |     0.000 |           |
## -------------|-----------|-----------|-----------|-----------|
##            B |         1 |        22 |         0 |        23 |
##              |     0.043 |     0.957 |     0.000 |     0.426 |
##              |     0.062 |     1.000 |     0.000 |           |
##              |     0.019 |     0.407 |     0.000 |           |
## -------------|-----------|-----------|-----------|-----------|
##            C |         0 |         0 |        16 |        16 |
##              |     0.000 |     0.000 |     1.000 |     0.296 |
##              |     0.000 |     0.000 |     1.000 |           |
##              |     0.000 |     0.000 |     0.296 |           |
## -------------|-----------|-----------|-----------|-----------|
## Column Total |        16 |        22 |        16 |        54 |
##              |     0.296 |     0.407 |     0.296 |           |
## -------------|-----------|-----------|-----------|-----------|
##
##
```

```r
test_pred <- knn(train = training[1:13], test = testing[1:13], cl = train_labels, k = 14)
CrossTable(x = test_labels, y = test_pred,prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  54
##
##
##              | test_pred
##  test_labels |         A |         B |         C | Row Total |
## -------------|-----------|-----------|-----------|-----------|
##            A |        15 |         0 |         0 |        15 |
##              |     1.000 |     0.000 |     0.000 |     0.278 |
```

```
##               |     0.938 |     0.000 |     0.000 |           |
##               |     0.278 |     0.000 |     0.000 |           |
## -------------|-----------|-----------|-----------|-----------|
##             B |         1 |        22 |         0 |        23 |
##               |     0.043 |     0.957 |     0.000 |     0.426 |
##               |     0.062 |     1.000 |     0.000 |           |
##               |     0.019 |     0.407 |     0.000 |           |
## -------------|-----------|-----------|-----------|-----------|
##             C |         0 |         0 |        16 |        16 |
##               |     0.000 |     0.000 |     1.000 |     0.296 |
##               |     0.000 |     0.000 |     1.000 |           |
##               |     0.000 |     0.000 |     0.296 |           |
## -------------|-----------|-----------|-----------|-----------|
## Column Total |        16 |        22 |        16 |        54 |
##               |     0.296 |     0.407 |     0.296 |           |
## -------------|-----------|-----------|-----------|-----------|
##
##
```

```r
test_pred <- knn(train = training[1:13], test = testing[1:13], cl = train_labels, k=20)
CrossTable(x = test_labels, y = test_pred, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  54
##
##
##               | test_pred
##  test_labels |         A |         B |         C | Row Total |
## -------------|-----------|-----------|-----------|-----------|
##             A |        15 |         0 |         0 |        15 |
##               |     1.000 |     0.000 |     0.000 |     0.278 |
##               |     0.833 |     0.000 |     0.000 |           |
##               |     0.278 |     0.000 |     0.000 |           |
## -------------|-----------|-----------|-----------|-----------|
##             B |         3 |        20 |         0 |        23 |
##               |     0.130 |     0.870 |     0.000 |     0.426 |
##               |     0.167 |     1.000 |     0.000 |           |
##               |     0.056 |     0.370 |     0.000 |           |
## -------------|-----------|-----------|-----------|-----------|
##             C |         0 |         0 |        16 |        16 |
##               |     0.000 |     0.000 |     1.000 |     0.296 |
##               |     0.000 |     0.000 |     1.000 |           |
##               |     0.000 |     0.000 |     0.296 |           |
## -------------|-----------|-----------|-----------|-----------|
## Column Total |        18 |        20 |        16 |        54 |
```

```
##              |     0.333 |     0.370 |     0.296 |           |
## -------------|-----------|-----------|-----------|-----------|
##
##
```

```
#end time
print(Sys.time()-strt)
```

```
## Time difference of 0.2909791 secs
```

```
# we find the best is 1 mistake improve our model when k=12 and 14.
```

Use library(caret) to train the model KNN

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(Rcpp)
library(ggplot2)
library(caret)
```

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

```
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
set.seed(3333)
knn_fit <- train(label ~., data = trl, method = "knn",
trControl=trctrl,
preProcess = c("center", "scale"),
tuneLength = 10)
knn_fit
```

```
## k-Nearest Neighbors
##
## 123 samples
##  13 predictor
##   3 classes: 'A', 'B', 'C'
##
## Pre-processing: centered (13), scaled (13)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 112, 111, 110, 111, 110, 111, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.9484737  0.9218625
##    7  0.9597985  0.9391761
```

```
##    9  0.9514652  0.9266650
##   11  0.9617521  0.9419621
##   13  0.9563714  0.9336441
##   15  0.9589355  0.9375835
##   17  0.9589355  0.9375835
##   19  0.9645299  0.9461287
##   21  0.9700855  0.9546871
##   23  0.9647436  0.9467811
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 21.
```

```
#k = 21, give the best accurarcy=0.97 and Kappa=0.95
```

```
# use the model to test the testing data
knnPredict <- predict(knn_fit,newdata = tel )
#Get the confusion matrix to see accuracy value and other parameter values
confusionMatrix(knnPredict, tel$label )
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  A  B  C
##          A 15  3  0
##          B  0 20  0
##          C  0  0 16
##
## Overall Statistics
##
##                Accuracy : 0.9444
##                  95% CI : (0.8461, 0.9884)
##     No Information Rate : 0.4259
##     P-Value [Acc > NIR] : 6.112e-16
##
##                   Kappa : 0.9161
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C
## Sensitivity            1.0000   0.8696   1.0000
## Specificity            0.9231   1.0000   1.0000
## Pos Pred Value         0.8333   1.0000   1.0000
## Neg Pred Value         1.0000   0.9118   1.0000
## Prevalence             0.2778   0.4259   0.2963
## Detection Rate         0.2778   0.3704   0.2963
## Detection Prevalence   0.3333   0.3704   0.2963
## Balanced Accuracy      0.9615   0.9348   1.0000
```

```
# only 3 B mistaked as A, not very high for the testing data. Also, the accurarcy is 0.9444 and the Kap
```

```
library(ranger)
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ranger':
##
##     importance
## The following object is masked from 'package:dplyr':
##
##     combine
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(caret)
library(randomForest)
set.seed(3033)
rf <- randomForest(label ~ ., data = trl)
rf
```

```
##
## Call:
##  randomForest(formula = label ~ ., data = trl)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 3.25%
## Confusion matrix:
##    A  B  C class.error
## A 42  1  0  0.02325581
## B  1 45  2  0.06250000
## C  0  0 32  0.00000000
```

```r
library(caret)
ctrl <- trainControl(method = "repeatedcv",
                     number = 10, repeats = 10)

# auto-tune a random forest
grid_rf <- expand.grid(.mtry = c(1,2,4,6))

set.seed(300)
m_rf <- train(label ~ ., data = trl, method = "rf",
              metric = "Kappa", trControl = ctrl,
              tuneGrid = grid_rf)
m_rf
```

```
## Random Forest
##
## 123 samples
##  13 predictor
##   3 classes: 'A', 'B', 'C'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 110, 111, 111, 110, 111, 112, ...
```

```
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##   1     0.9821404  0.9730934
##   2     0.9756019  0.9632366
##   4     0.9658467  0.9484100
##   6     0.9626415  0.9435974
##
## Kappa was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 1.
```

```
#mtry = 1, Accurart is 0.9821404 and kappa is 0.9730934
```

```
rfPredict <- predict(m_rf,newdata = tel )
confusionMatrix(rfPredict, tel$label )
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  A  B  C
##          A 15  0  0
##          B  0 23  0
##          C  0  0 16
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.934, 1)
##     No Information Rate : 0.4259
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C
## 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             0.2778   0.4259   0.2963
## Detection Rate         0.2778   0.4259   0.2963
## Detection Prevalence   0.2778   0.4259   0.2963
## Balanced Accuracy      1.0000   1.0000   1.0000
```

```
# we find that the accuray is 1 and kappa is 1, excellent.
```

try to compare with C5.0

```
library(broom)
library(tidyr)
library(caret)
library(tictoc)
library(lattice)
library(ggplot2)
```

```r
library(ranger)
library(h2o)
```

```
##
## ----------------------------------------------------------------------
##
## Your next step is to start H2O:
##     > h2o.init()
##
## For H2O package documentation, ask for help:
##     > ??h2o
##
## After starting H2O, you can use the Web UI at http://localhost:54321
## For more information visit http://docs.h2o.ai
##
## ----------------------------------------------------------------------
##
## Attaching package: 'h2o'
## The following objects are masked from 'package:stats':
##
##     cor, sd, var
## The following objects are masked from 'package:base':
##
##     %*%, %in%, &&, ||, apply, as.factor, as.numeric, colnames,
##     colnames<-, ifelse, is.character, is.factor, is.numeric, log,
##     log10, log1p, log2, round, signif, trunc
```

```r
library(stats)
library(base)
library(randomForest)
```

```r
### c5.0 tree to give the first model and result
set.seed(300)
m <- train(label ~ ., data = trl, method = "C5.0")
```

```
## Warning: 'trials' should be <= 9 for this object. Predictions generated
## using 9 trials
## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials
## Warning: 'trials' should be <= 8 for this object. Predictions generated
## using 8 trials
## Warning: 'trials' should be <= 5 for this object. Predictions generated
## using 5 trials
```

```
## Warning: 'trials' should be <= 5 for this object. Predictions generated
## using 5 trials

## Warning: 'trials' should be <= 9 for this object. Predictions generated
## using 9 trials

## Warning: 'trials' should be <= 7 for this object. Predictions generated
## using 7 trials

## Warning: 'trials' should be <= 8 for this object. Predictions generated
## using 8 trials

## Warning: 'trials' should be <= 6 for this object. Predictions generated
## using 6 trials

## Warning: 'trials' should be <= 8 for this object. Predictions generated
## using 8 trials

## Warning: 'trials' should be <= 6 for this object. Predictions generated
## using 6 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 7 for this object. Predictions generated
## using 7 trials

## Warning: 'trials' should be <= 6 for this object. Predictions generated
## using 6 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 8 for this object. Predictions generated
## using 8 trials

## Warning: 'trials' should be <= 9 for this object. Predictions generated
## using 9 trials

## Warning: 'trials' should be <= 8 for this object. Predictions generated
## using 8 trials

## Warning: 'trials' should be <= 9 for this object. Predictions generated
## using 9 trials
```

```
## Warning: 'trials' should be <= 5 for this object. Predictions generated
## using 5 trials

## Warning: 'trials' should be <= 5 for this object. Predictions generated
## using 5 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 7 for this object. Predictions generated
## using 7 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 7 for this object. Predictions generated
## using 7 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
## using 1 trials

## Warning: 'trials' should be <= 6 for this object. Predictions generated
## using 6 trials

## Warning: 'trials' should be <= 6 for this object. Predictions generated
## using 6 trials

## Warning: 'trials' should be <= 4 for this object. Predictions generated
## using 4 trials

## Warning: 'trials' should be <= 4 for this object. Predictions generated
## using 4 trials

## Warning: 'trials' should be <= 4 for this object. Predictions generated
## using 4 trials

## Warning: 'trials' should be <= 4 for this object. Predictions generated
## using 4 trials

## Warning: 'trials' should be <= 7 for this object. Predictions generated
## using 7 trials

## Warning: 'trials' should be <= 1 for this object. Predictions generated
```

```
## using 1 trials

## Warning: 'trials' should be <= 8 for this object. Predictions generated
## using 8 trials

## Warning: 'trials' should be <= 9 for this object. Predictions generated
## using 9 trials

## Warning: 'trials' should be <= 8 for this object. Predictions generated
## using 8 trials

## Warning: 'trials' should be <= 9 for this object. Predictions generated
## using 9 trials

## Warning: 'trials' should be <= 8 for this object. Predictions generated
## using 8 trials
```

```
m
```

```
## C5.0
##
## 123 samples
##  13 predictor
##   3 classes: 'A', 'B', 'C'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 123, 123, 123, 123, 123, 123, ...
## Resampling results across tuning parameters:
##
##   model  winnow  trials  Accuracy   Kappa
##   rules  FALSE    1      0.8913776  0.8318133
##   rules  FALSE   10      0.9326047  0.8959589
##   rules  FALSE   20      0.9310704  0.8936118
##   rules   TRUE    1      0.8819750  0.8181153
##   rules   TRUE   10      0.9039278  0.8518311
##   rules   TRUE   20      0.9030667  0.8505844
##   tree   FALSE    1      0.8827631  0.8185151
##   tree   FALSE   10      0.9261146  0.8858830
##   tree   FALSE   20      0.9271079  0.8874290
##   tree    TRUE    1      0.8802430  0.8154014
##   tree    TRUE   10      0.9040771  0.8521893
##   tree    TRUE   20      0.9037585  0.8519172
##
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were trials = 10, model = rules
##  and winnow = FALSE.
```

```
p <- predict(m, trl)
confusionMatrix(data=p, trl$label)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  A  B  C
##          A 43  0  0
##          B  0 48  0
##          C  0  0 32
```

```
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9705, 1)
##     No Information Rate : 0.3902
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C
## 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             0.3496   0.3902   0.2602
## Detection Rate         0.3496   0.3902   0.2602
## Detection Prevalence   0.3496   0.3902   0.2602
## Balanced Accuracy      1.0000   1.0000   1.0000
```

```
rp <- predict(m, tel)
confusionMatrix(data=rp, tel$label)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  A  B  C
##          A 15  0  0
##          B  0 23  0
##          C  0  0 16
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.934, 1)
##     No Information Rate : 0.4259
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C
## 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             0.2778   0.4259   0.2963
## Detection Rate         0.2778   0.4259   0.2963
## Detection Prevalence   0.2778   0.4259   0.2963
## Balanced Accuracy      1.0000   1.0000   1.0000
```