به نام خدا

# تمرین چهارم

# مبانی داده‌کاوی

## طبقه‌بندی و محاسبه دقت آن

نام و نام خانوادگی: علی رضائی نژاد

شماره دانشجویی: ۹۶۰۱۸۴۱۵۶

مشخصه درس: ۹۱۳۵۱

نام استاد: خانم امینه امینی

بر خلاف روند تمرین‌های پیشین، برای طبقه‌بندی به فیلدِ کلاس (Seedtype (Class)) نیاز داریم و برای این تمرین نیازی به حذف آن از ستون‌های مجموعه داده نیست. در خطوط ابتدایی اعداد ۱ تا ۳ را به نوع دانه گندم مربوطه نسبت دادم تا خوانایی نتایج بهبود یابد. برای یادآوری:

Class1 = Kama  Class2 = Rosa  Class3 = Canadian

برای تقسیم مجموعه داده به دو بخش **train** و **test** که برای طبقه‌بندی و پیش‌بینی لازم است، در هر دو شیوه‌ی طبقه‌بندی، سه چهارم داده را به عنوان **training** و باقی را برای **testing** اختصاص دادم. همچنین این دیتاست فاقد داده‌های **missing** است. الگوریتم خود را روی بخش آموزش اجرا می‌کنیم و درخت تصمیم‌گیری را برایش رسم می‌کنیم. از بیست و پنج درصد باقی مانده از دیتاست، برای تست و پیش‌بینی استفاده خواهیم کرد.

In [1]:
```
seeds <- read.csv("D:\\Daneshga\\T8\\Amini\\Datasets\\Iris Alternatives\\Wheat Seeds Dataset\\Book1.csv")

seeds$Seedtype..Class.[which(seeds$Seedtype..Class.==1)]="Kama"
seeds$Seedtype..Class.[which(seeds$Seedtype..Class.==2)]="Rosa"
seeds$Seedtype..Class.[which(seeds$Seedtype..Class.==3)]="Canadian"
seeds$Seedtype..Class. = factor(seeds$Seedtype..Class.)

library(ggplot2)
library(lattice)
library(caret)
library(rpart)
#library(e1071)

size <- floor(0.75 * nrow(seeds))
smpl <- sample(seq_len(nrow(seeds)), size = size)
train <- seeds[smpl, ]
test <- seeds[-smpl, ]

#summary(seeds)
```

# Decision Tree

# درخت تصمیم‌گیری

کتابخانه‌ی **rpart** و زیرمجموعه‌ی آن **rpart.plot** (برای نمایش درخت) در این زمینه تابع‌های مورد نظرمان را در اختیار قرار می‌دهند.

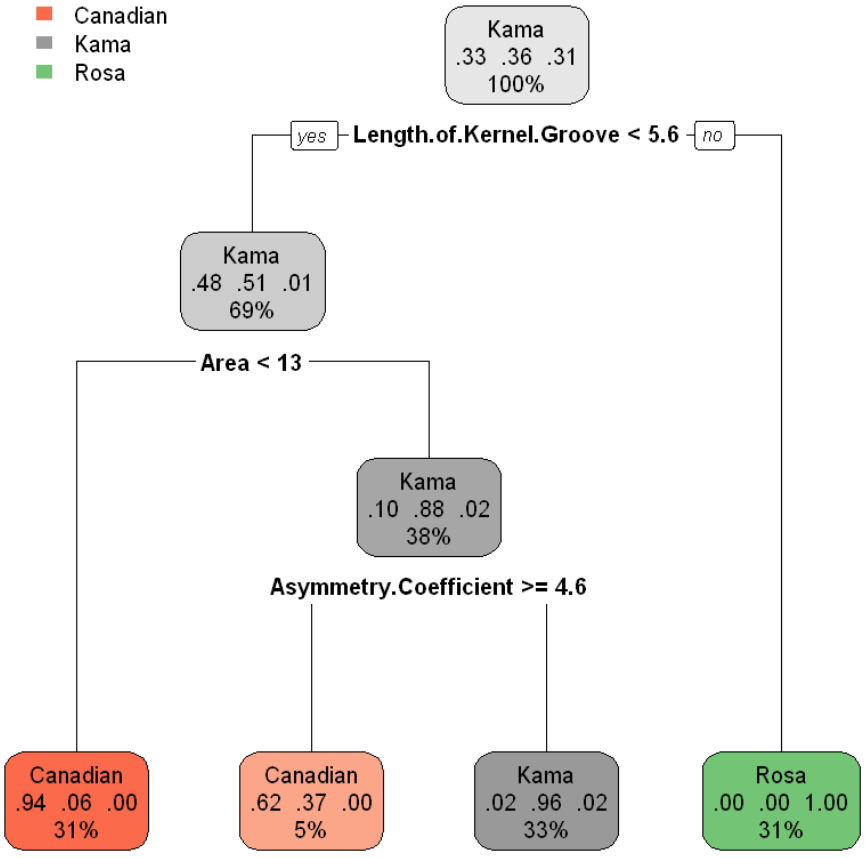
In [2]:
```
tree <- rpart( Seedtype..Class. ~ . , data = train , method = 'class')
print(tree)
```

```
n= 157

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 157 101 Kama (0.331210191 0.356687898 0.312101911)
   2) Length.of.Kernel.Groove< 5.597 109  53 Kama (0.477064220 0.513761468 0.009174312)
     4) Area< 12.71 49   3 Canadian (0.938775510 0.061224490 0.000000000) *
     5) Area>=12.71 60   7 Kama (0.100000000 0.883333333 0.016666667)
      10) Asymmetry.Coefficient>=4.6065 8   3 Canadian (0.625000000 0.375000000 0.000000000) *
      11) Asymmetry.Coefficient< 4.6065 52   2 Kama (0.019230769 0.961538462 0.019230769) *
   3) Length.of.Kernel.Groove>=5.597 48   0 Rosa (0.000000000 0.000000000 1.000000000) *
```

In [3]:
```
#install.packages("rpart.plot")
library(rpart.plot)
rpart.plot(tree)
```



## Naive Bayesian

## بیزین ساده

```
In [2]: bayes <- naiveBayes (Seedtype..Class. ~ . , data = train)
        bayes

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
 Canadian      Kama      Rosa
0.3184713 0.3439490 0.3375796

Conditional probabilities:
          Area
Y             [,1]      [,2]
  Canadian 11.93040 0.7242645
  Kama     14.36370 1.2145801
  Rosa     18.23736 1.4582803

          Perimeter
Y             [,1]      [,2]
  Canadian 13.27140 0.3324615
  Kama     14.30111 0.5775931
  Rosa     16.09094 0.6253899

          Compactness
Y               [,1]       [,2]
  Canadian 0.8503840 0.02145955
  Kama     0.8810426 0.01698652
  Rosa     0.8837075 0.01608682

          Length.of.Kernel
Y             [,1]      [,2]
  Canadian 5.237480 0.1319937
  Kama     5.506889 0.2344520
  Rosa     6.127755 0.2871774

          Width.of.Kernel
Y             [,1]      [,2]
  Canadian 2.862140 0.1473970
  Kama     3.250185 0.1787488
  Rosa     3.671132 0.1844607

          Asymmetry.Coefficient
Y             [,1]     [,2]
  Canadian 4.819160 1.487550
  Kama     2.601263 1.144380
  Rosa     3.643981 1.223242

          Length.of.Kernel.Groove
Y             [,1]      [,2]
  Canadian 5.120660 0.1586142
  Kama     5.073630 0.2625668
  Rosa     6.003736 0.2748143
```

برای استفاده از تابع naiveBayes نیاز به نصب کتابخانه‌ی e1071 و کتابخانه‌ی وابسته‌ی آن، proxy بود.

Cell اول برنامه در Jupyter Notebook به جز فراخوانی لایبرری های جدید، همانند درخت تصمیم‌گیری بود، به همین دلیل آن را تکرار نکردم.

همانند قبل، هفتاد و پنج درصد داده را به آموزش اختصاص دادم.

(برای نصب کتابخانه های e1071، rpart.plot و proxy از فرامین زیر در محیط Jupyter Notebook استفاده شد.)

```
install.packages("proxy")
install.packages("e1071")
install.packages("rpart.plot")
```

## مقایسه‌ی دقت طبقه‌بندی Accuracy Assessment

کتابخانه‌ی caret کار ما را در سنجش معیارهای Accuracy، Percision و Recall بسیار راحت می‌کند. صفحه ۲۴ داکیومنتیشن این کتابخانه که در زیر لینک آن قرار داده شده شامل توضیحات تابع confusionMatrix است که بدین منظور از آن استفاده کردم.

آرگیومنت mode در صورتی که برابر با prec_recall تنظیم شود، علاوه بر Accuracy، دو معیار دیگر مد نظر ما را هم برای پیش‌بینی‌های هر کلاس به طور جداگانه محاسبه می‌کند.

caret.pdf (r-project.org)

# سنجش الگوریتم درخت تصمیم‌گیری

In [6]:
```
#install.packages("proxy")
#install.packages("e1071")

prediction <- predict(tree, test, type = 'class')
result <- table( test$Seedtype..Class. , prediction)
confusionMatrix(result, mode = "prec_recall")
```

```
Confusion Matrix and Statistics

          prediction
           Canadian Kama Rosa
  Canadian       18    0    0
  Kama            3   10    1
  Rosa            0    1   20

Overall Statistics

               Accuracy : 0.9057
                 95% CI : (0.7934, 0.9687)
    No Information Rate : 0.3962
    P-Value [Acc > NIR] : 1.239e-14

                  Kappa : 0.8557

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Canadian Class: Kama Class: Rosa
Precision                     1.0000      0.7143      0.9524
Recall                        0.8571      0.9091      0.9524
F1                            0.9231      0.8000      0.9524
Prevalence                    0.3962      0.2075      0.3962
Detection Rate                0.3396      0.1887      0.3774
Detection Prevalence          0.3396      0.2642      0.3962
Balanced Accuracy             0.9286      0.9069      0.9606
```

# سنجش الگوریتم بیزین ساده

In [3]:
```
prediction <- predict(bayes, test, type = 'class')
result <- table( test$Seedtype..Class. , prediction)
confusionMatrix(result, mode = "prec_recall")
```

```
Confusion Matrix and Statistics

          prediction
           Canadian Kama Rosa
  Canadian       19    1    0
  Kama            3   11    2
  Rosa            0    1   16

Overall Statistics

               Accuracy : 0.8679
                 95% CI : (0.7466, 0.9452)
    No Information Rate : 0.4151
    P-Value [Acc > NIR] : 1.096e-11

                  Kappa : 0.8

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Canadian Class: Kama Class: Rosa
Precision                     0.9500      0.6875      0.9412
Recall                        0.8636      0.8462      0.8889
F1                            0.9048      0.7586      0.9143
Prevalence                    0.4151      0.2453      0.3396
Detection Rate                0.3585      0.2075      0.3019
Detection Prevalence          0.3774      0.3019      0.3208
Balanced Accuracy             0.9157      0.8606      0.9302
```

مشاهده می‌شود که معیار Accuracy و Percision (در پیش‌بینی‌های هر سه کلاس) در الگوریتم درخت تصمیم‌گیری از بیزین برتر هستند. گرچه در Recall و فقط برای کلاس Canadian (کلاس شماره ۳) الگوریتم بیزین ساده بهتر در این Iteration بهتر عمل کرده است.

در نتیجه، الگوریتم درخت تصمیم‌گیری برای مجموعه داده‌ی seeds بهتر و دقیق‌تر پیش‌بینی می‌کند.

## خلاصه‌ای از فرم دیتاست و مفهوم Compactness:

| | A | P | C | L | W | AC | LG | Class |
|---|---|---|---|---|---|---|---|---|
| 1 | 15.26 | 14.84 | 0.871 | 5.763 | 3.312 | 2.221 | 5.22 | 1 |
| 2 | 14.88 | 14.57 | 0.8811 | 5.554 | 3.333 | 1.018 | 4.956 | 1 |
| 3 | 14.29 | 14.09 | 0.905 | 5.291 | 3.337 | 2.699 | 4.825 | 1 |
| . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . |
| 210 | 12.3 | 13.34 | 0.8684 | 5.243 | 2.974 | 5.637 | 5.063 | 3 |

1. A = Area مساحت
2. P = Perimeter محیط
3. C = Compactness دنسیتی/چگالی کل دانه
4. L = Length of Kernel طول هسته
5. W = Width of Kernel عرض هسته
6. AC = Asymmetry Coefficient ضریب عدم تقارن
7. LG = Length of Kernel Groove طول گروو دانه

Compactness به این صورت تعریف شده است:

$$C = \frac{4\pi A}{P^2}$$

a = Area مساحت (میلی‌متر مربع)

p = perimeter محیط (میلی‌متر)

π = 3.14