Eksploracja Danych w R

Spis treści

Przegląd danych

Charakterystyka:	Rozmiar:	Ilość atrybutów:	Charakterystyka atrybutów:	Powiązane zadania:
wielowymiarowe	5 x 1372	5	liczby rzeczywiste	klasyfikacja (ang. classification)

Dane zostały wydobyte ze zdjęć oryginalnych i podrobionych banknotów. Do digitalizacji użyto kamery przemysłowej zwykle używanej do kontroli wydruku. Zdjęcia mają rozmiar 400 x 400 pikseli, są w skali szarości o rozdzielczości około 660 dpi. Narzędzie `Wavelet Transform` zostało użyte do wydobycia danych ze zdjęć.

Atrybuty:

1. wariancja (ang. variance),
2. współczynnik skośności (ang. skewness),
3. kurtoza (ang. curtosis),
4. entropia (ang. entropy),
5. klasa (ang. class).

Mamy dla analizy zbiór danych, który zawiera 5 kolumn oraz 1372 wierszy. Pierwsze cztery kolumny zawierają dane wydobyte ze zdjęć. Ostatnia piąta kolumna wskazuje czy banknot są oryginalna. Gdzie 1 -- oryginał, `0` -- podróbka. Dane są liczby rzeczywistymi.

Źródło: https://archive.ics.uci.edu/ml/datasets/banknote+authentication

Statystyki ogólne

Ładowanie pakietów:

```
library(caret)
library(data.table)
library(dplyr)
library(PerformanceAnalytics)
library(rpart.plot)
```

Ładowanie zbioru danych:

```
url = 'http://bit.ly/banknote-auth'
df = data.frame(fread(url))
names(df) = c('variance', 'skewness', 'curtosis', 'entropy', 'class')
```

Sprawdzimy strukturę danych:

```
str(df)
```

Wynik:

```
'data.frame':   1372 obs. of  5 variables:
 $ variance: num  3.622 4.546 3.866 3.457 0.329 ...
 $ skewness: num  8.67 8.17 -2.64 9.52 -4.46 ...
 $ curtosis: num  -2.81 -2.46 1.92 -4.01 4.57 ...
 $ entropy : num  -0.447 -1.462 0.106 -3.594 -0.989 ...
 $ class   : int  0 0 0 0 0 0 0 0 0 0 ...
```

Widzimy na wyniku powyżej, że zbiór zawiera 1372 wierszy oraz 5 kolumn. R poprawnie rozpoznał typy danych: `int` dla kolumny `class` (znaczenia 1 lub 0) oraz `num` dla pozostałych.

Wyświetlimy pierwsze pięć wierszy:

```
head(df, 5)
```

variance	skewness	curtosis	entropy	class
3.62160	8.6661	-2.8073	-0.44699	0
4.54590	8.1674	-2.4586	-1.46210	0
3.86600	-2.6383	1.9242	0.10645	0
3.45660	9.5228	-4.0112	-3.59440	0
0.32924	-4.4552	4.5718	-0.98880	0

Wyświetlimy statystyki podsumowujące (ang. summary statistics):

```
summary(select(df, -class))
```

Wynik:

```
   variance          skewness          curtosis          entropy       
Min.   :-7.0421   Min.   :-13.773   Min.   :-5.2861   Min.   :-8.5482  
1st Qu.:-1.7730   1st Qu.: -1.708   1st Qu.:-1.5750   1st Qu.:-2.4135  
Median : 0.4962   Median :  2.320   Median : 0.6166   Median :-0.5867  
Mean   : 0.4337   Mean   :  1.922   Mean   : 1.3976   Mean   :-1.1917  
3rd Qu.: 2.8215   3rd Qu.:  6.815   3rd Qu.: 3.1793   3rd Qu.: 0.3948  
Max.   : 6.8248   Max.   : 12.952   Max.   :17.9274   Max.   : 2.4495  
```

Na wyniku powyżej można sprawdzić minimalne i maksymalne znaczenia, medianę oraz średnią arytmetyczną dla każdego atrybutu.

Korelacja

Dla analizy skorzystamy z funkcji `cor()`:

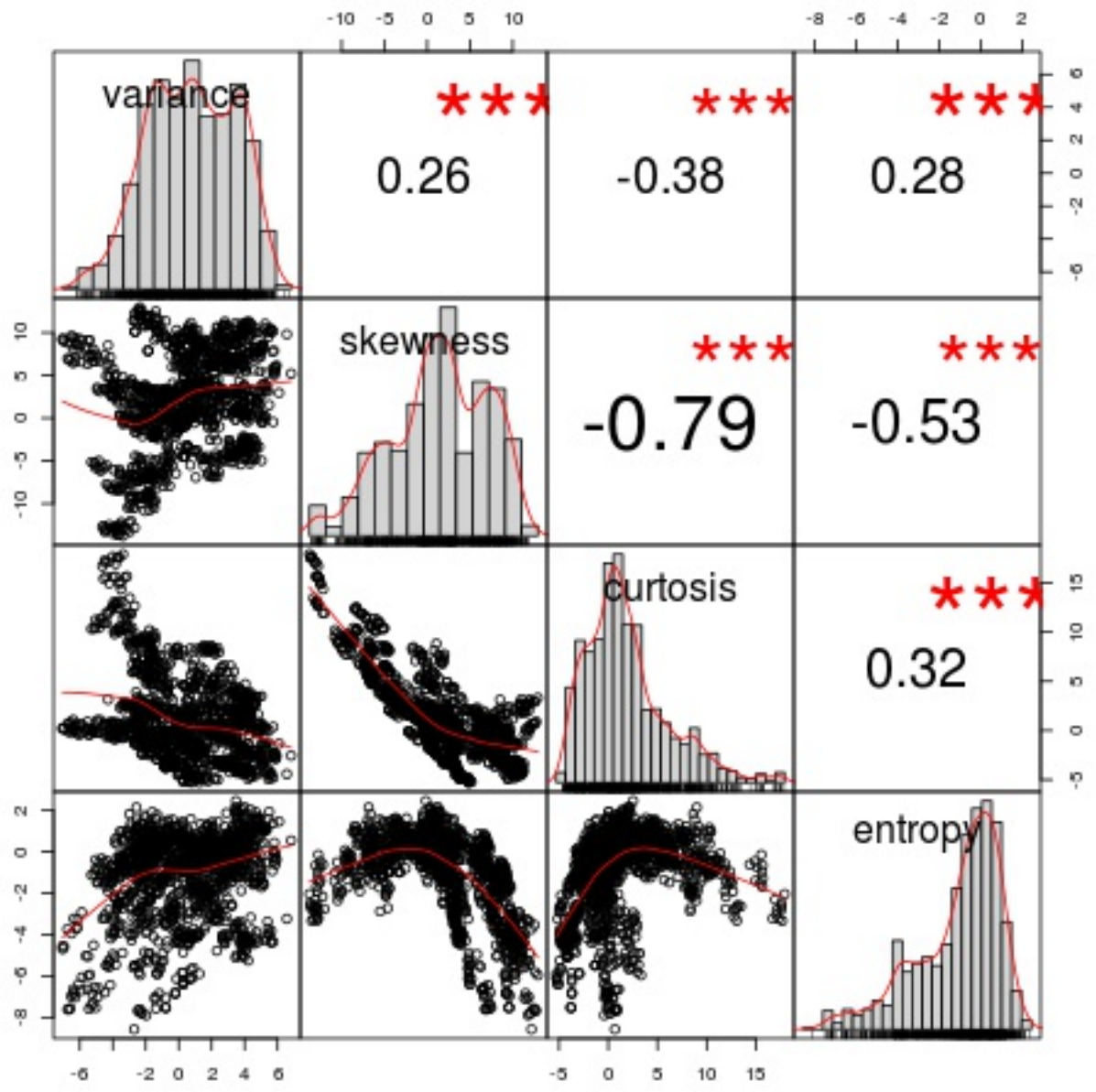
```
cor(df)
```

	variance	skewness	curtosis	entropy	class
variance	1.0000000	0.2640255	-0.3808500	0.27681670	-0.72484314
skewness	0.2640255	1.0000000	-0.7868952	-0.52632084	-0.44468776
curtosis	-0.3808500	-0.7868952	1.0000000	0.31884089	0.15588324
entropy	0.2768167	-0.5263208	0.3188409	1.00000000	-0.02342368
class	-0.7248431	-0.4446878	0.1558832	-0.02342368	1.00000000

Sprawdzanie korelacji w postaci tablicy powyżej. Ogólnie tablica musi mieć jedynki tylko po przekątnej, bo przykładowo wariacja koreluje z wariancją jako jeden do jednego. Jedynka nie po przekątnej wskazuję na duplikację danych (ang. duplicity), czyli mamy te same danych w różnych kolumnach. Uwagi potrzebują też znaczenia zbliżone do jedynki. W naszym przypadku to korelacja współczynnika skośności (ang. skewness) do kurtozy (ang. curtosis). Prawdopodobnie kolumnę `skewness` trzeba będzie usunąć, jednak sprawdzimy to później bardziej szczegółowo. Również ciekawy jest wynik `-0.72484314` jako korelacja klasy (ang. class) do wariacji (ang. variance). Tu zalecane będzie budowanie drzewa decyzyjnego.

Identyfikacji silnie skorelowanych zmiennych. Wyświetlimy wykres ze współczynnikiem korelacji dla atrybutów:

```
chart.Correlation(select(df, -class), histogram=T)
```



Wykres powyżej potwierdza wysokie znaczenie korelacji kolumny skewness: `-0.79`. Zanim usuniemy kolumnę skewness, zrobimy podsumowanie poziomu współczynnika korelacji:

```
df$class = ifelse(df$class=='1', 'Y', 'N')
df2 = select(df, -class)
cor_matrix = cor(df2)
summary(cor_matrix[upper.tri(cor_matrix)])
```

Wynik:

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.78690 -0.48995 -0.05841 -0.13906  0.27362  0.31884
```

Między atrybutami występuje korelacja ujemna. Znaczenie: `-0.78690`. Sprawdzimy atrybuty zmiennych ze współczynnikiem korelacji powyżej `0,75`. Skorzystamy z funkcji `findCorrelation()`:

```
names(df[findCorrelation(cor_matrix, cutoff = 0.75)])
```

Wynik:

```
'skewness'
```

Usuniemy kolumnę skewness i ponownie zrobimy podsumowanie poziomu współczynnika korelacji:

```
df2 = select(df2, -skewness)
cor_matrix = cor(df2)
summary(cor_matrix[upper.tri(cor_matrix)])
df = cbind.data.frame(df2, class = df$class) # dodamy `class`
```

Wynik:

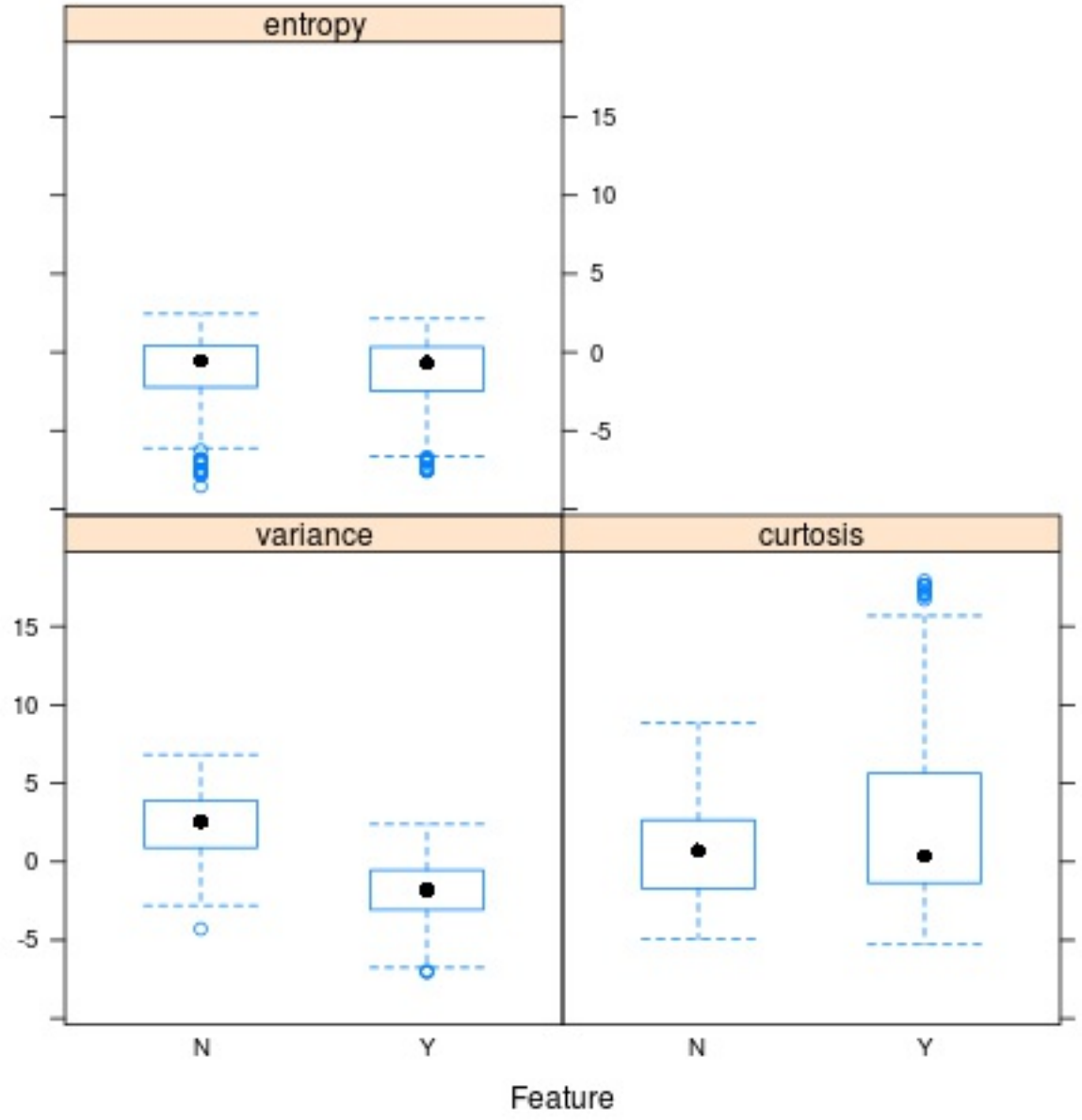
```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.38085 -0.05202  0.27682  0.07160  0.29783  0.31884
```

Wszystko w porządku. Przejdziemy do wizualizacji atrybutów.

Wizualizacja atrybutów

Skorzystamy z funkcji `featurePlot`:

```
featurePlot(x=select(df, -class), y=df$class, plot='box')
```



Ogólnie przygotowanie atrybutów oraz obserwacji wykonano dobrze. Jedynej uwagi zasługuje tylko kolumna `curtosis` i wyłączne dane przygotowane dla prawdziwych banknot (zobacz wykres powyżej).

Wariacja

Sprawdzimy atrybuty z wariacją bliską zera. Skorzystamy z funkcji `nearZeroVar()`:

```
nearZeroVar(select(df, -class), saveMetrics=T)
```

	freqRatio	percentUnique	zeroVar	nzv
variance	1.25	97.52187	FALSE	FALSE
skewness	1.20	91.54519	FALSE	FALSE
curtosis	1.00	92.56560	FALSE	FALSE
entropy	1.00	84.25656	FALSE	FALSE

Wszystko w porządku. Funkcja zwróciła FALSE dla wszystkich atrybutów.

Kombinacja liniowa

Sprawdzimy zaleznosci liniowe miedzy atrybutami:

```
findLinearCombos(select(df, -class))
```

Wynik:

```
$linearCombos
list()

$remove
NULL
```

Wszystko w porządku. Funkcja zwróciła `NULL. Przejdziemy do dystrybucji.

Dystrybucja

Wyświetlimy dystrybucję:

```
df2 = data.frame(table(df$class))
names(df2) = c('class', 'freq')
cbind(df2, percent=round((df2$freq/sum(df2$freq))*100, 1))
```

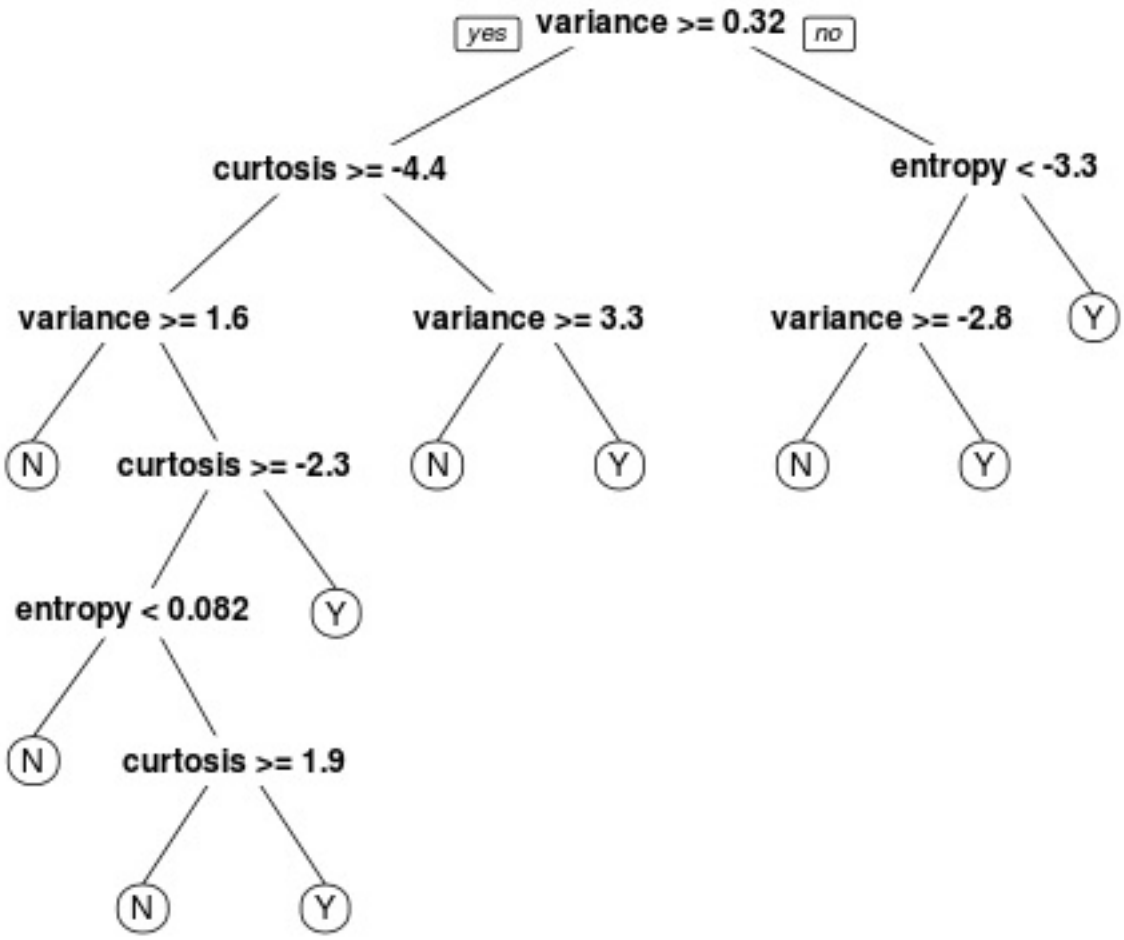
class	freq	percent
N	762	55.5
Y	610	44.5

Wszystko w porządku (zobacz tabelę powyżej). Prawie 50% do 50%. Czyli zbiór zawiera oryginalne banknoty oraz podróbki w dobrej proporcji.

Drzewo decyzyjne

Wyświtlimy drzewo decyzyjne:

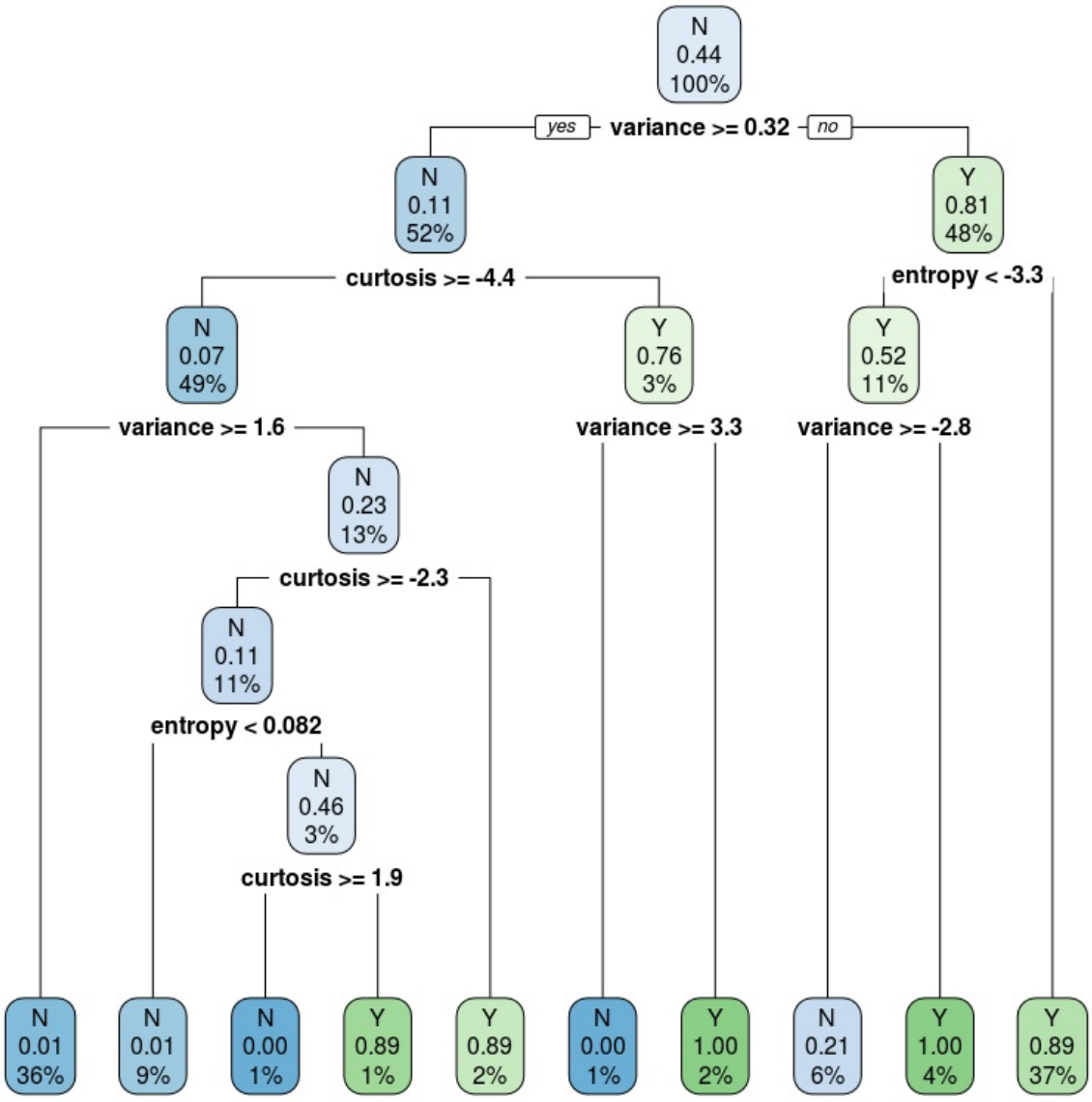
```
rtree_set = rpart(class ~., df)
prp(rtree_set)
```



Właśnie to, co zauważyliśmy jeszcze podczas sprawdzania korelacji. Na samej górze drzewa jest zmian `variance`, która dobrze koreluje z klasą (ang. class). Można zapisać reguły, które będą dobrze działały (zobacz powyżej). Skala problemu nie jest zbyt duża, by go manualnie rozwiązać. Uczenie maszynowe (ang. machine learning) nie będzie lepsze od klasycznego rozwiązania.

Teraz wyświtlimy drzewo decyzyjne oraz szczegóły dodatkowe:

```
rpart.plot(rtree_set)
```



Drzewo w takiej postaci odzwierciedla, w jaki sposób na podstawie atrybutów były podejmowane decyzje klasyfikujące. Zaletą tej reprezentacji jest jej czytelność dla człowieka. W prosty sposób można przekształcić ją do reprezentacji regułowej.