

Мастер академске студије
Рачунарство и аутоматика

Рачунарство високих перформанси
у информационом инжењерингу

Основи класификације

(материјали за предавања)

1. Класификација
2. Регресија
3. Метод најближих суседа
4. Метод потпорних вектора
5. Стабла одлучивања
6. Извори и литература

Класификација

задатак у којем треба утврдити којој од могућих класа припада посматрана појава

класа представља вредност обележја

утврђивање припадности класи се начелно изводи на основу вредности других одабраних обележја посматране појаве

резултат је одабрана класа

одабрана класа се може али и не мора поклапати са стварном класом којој припада посматрана појава

Класификација

врсте класификације

бинарна

две могуће класе у разматрању

n -арна

n (више од две) могућих класа у разматрању

Основне улоге обележја у класификацији

циљно обележје

(зависно обележје, одговор)

обележје чију вредност за неку појаву треба одредити на основу вредности других обележја

циљно обележје је категоријско (квалитативно)

обично постоји једно циљно обележје

ознака Y

предикторско обележје

(независно обележје, предиктор)

обележје чија се вредност за неку појаву користи у одређивању вредности циљног обележја

предикторско обележје може бити или категоријско (квалитативно) или нумеричко (квантитативно)

обично постоји више предикторских обележја

ознака $X = (X_1, X_2, \dots, X_p)$

Класификациони модел

(класификатор)

модел који начелно успоставља везу између циљног обележја и предикторских обележја

може настати у поступку обучавања

постојање варијација у врсти структуре и нивоу сложености

сврха

предвиђање

разумевање

Поступак обучавања

формирање модела и подешавање параметара ради постизања што бољих перформанси у класификацији

обучавање се изводи на основу расположивих података

за појаве заступљене у подацима познате су вредности циљног обележја

Перформансе у класификацији

перформансе могу варирати зависно од коришћених података
метод који је најпогоднији за један случај не мора бити
најпогоднији за неки други случај

тежити методи која обезбеђује ниску варијансу (енгл. *variance*) и
ниску пристрасност (енгл. *bias*)

теорема о непостојању бесплатног ручка

Оцена перформанси

грешка обучавања

процент погрешно класификованих појава међу појавама које су коришћене током обучавања

грешка тестирања

процент погрешно класификованих појава међу појавама које нису коришћене током обучавања

Класификација

Оцена перформанси

матрица конфузије – пример за бинарни случај

		Стварна класа	
		0 (не, није, нема, -)	1 (да, јесте, има, +)
Процењена класа	0 (не, није, нема, -)	број стварно негативних (енгл. <i>true negative</i> , <i>TN</i>)	број лажно негативних (енгл. <i>false negative</i> , <i>FN</i>)
	1 (да, јесте, има, +)	број лажно позитивних (енгл. <i>false positive</i> , <i>FP</i>)	број стварно позитивних (енгл. <i>true positive</i> , <i>TP</i>)

Оцена перформанси

матрица конфузије – пример за бинарни случај

тачност (*ACC*)

$$(TP + TN) / (TP + TN + FP + FN)$$

прецизност (*PREC*)

$$TP / (TP + FP)$$

осетљивост (*SENS*)

$$TP / (TP + FN)$$

специфичност (*SPEC*)

$$TN / (TN + FP)$$

скор *F1*

$$2 / (1 / PREC + 1 / SENS)$$

Процена грешке тестирања помоћу појава за обучавање валидациони приступ (енгл. *validation set approach*)

насумична подела расположивих појава у скуп за обучавање и валидациони скуп

модел бива подешаван на основу скупа за обучавање

подешени модел се примењује над валидационим скупом

рачуна се грешка над валидационим скупом

Процена грешке тестирања помоћу појава за обучавање појединачна унакрсна валидација (енгл. *leave-one-out cross-validation*)

расположиве појаве се распоређују у валидациони скуп, који обухвата само једну појаву, и скуп за обучавање, који обухвата све остале појаве

модел бива подешаван на основу скупа за обучавање

подешени модел се примењује над валидационим скупом

рачуна се грешка над валидационим скупом

понавља се по истом принципу распоређивање појава у валидациони скуп и скуп за обучавање, при чему нека друга појава постаје једини елемент валидационог скупа

поново се по истом принципу изводе подешавање модела, примена модела и рачунање грешке

понављање се изводи док свака појава тачно једном не буде део валидационог скупа

по завршетку понављања рачуна се аритметичка средина за грешку над валидационим скупом

Процена грешке тестирања помоћу појава за обучавање

k-тострука унакрсна валидација (енгл. *k-fold cross-validation*)

расположиве појаве се насумично распоређују у *k* скупова исте или приближно исте величине

за сваки појединачни скуп изводи се посебан поступак

појединачни скуп служи као валидациони скуп

сви остали скупови заједно служе као скуп за обучавање

модел бива подешаван на основу скупа за обучавање

подешени модел се примењује над валидационим скупом

рачуна се грешка над валидационим скупом

по завршетку поступка за сваки појединачни скуп, рачуна се аритметичка средина за грешку над валидационим скупом

Методи класификације

разноврсни методи на располагању

примери метода које се могу користити у класификацији

регресија

логистичка регресија

метод најближих суседа

метод потпорних вектора

стабла одлучивања

појединачна

вишеструка

Класификација

статистичко учење (енгл. *statistical learning*)

„велики скуп алата за разумевање података” (по Џејмсу и сарадницима)

машинско учење (енгл. *machine learning*)

„аутоматска детекција смислених образаца у подацима” (по Шалев-Шварцу и Бен-Давиду)

врсте машинског учења

надгледано учење (енгл. *supervised learning*)

потпомогнуто учење (енгл. *reinforcement learning*)

ненадгледано учење (енгл. *unsupervised learning*)

Скупови података коришћени у примерима

скуп података **abalone**

скуп података *Abalone*

аутори *Warwick Nash, Tracy Sellers, Simon Talbot, Andrew Cawthorn* и *Wes Ford*
електронска локација (доступни скуп података и пратеће информације)

<https://archive.ics.uci.edu/dataset/1/abalone>

<https://doi.org/10.24432/C55C7W>

репозиторијум *UC Irvine Machine Learning Repository* на електронској локацији
<http://archive.ics.uci.edu/>

скуп података дониран за репозиторијум 30. 11. 1995.

датотека *abalone.zip* (преузето 27. 3. 2024)

електронска локација

<https://archive.ics.uci.edu/static/public/1/abalone.zip>

подаци у датотеци *abalone.data*

лиценца *Creative Commons Attribution 4.0 International (CC BY 4.0)*

електронска локација

<https://creativecommons.org/licenses/by/4.0/legalcode>

скуп података *Abalone* читан, обрађиван и анализиран језиком *R*

активности над скупом података и резултати представљени у наставку

датотеке визуализација генерисане за формат *TIFF* (15,5 x 15,5 cm или 20,5 x 15,5 cm, 300 DPI)

Скупови података коришћени у примерима

скуп података **abalone**

подаци о абалонима с Тасманије (Аустралија)

4177 записа

9 обележја

пол, дужина, пречник, висина, различите тежине и број прстенова

коришћен скуп података *Abalone*

подаци из датотеке *abalone.data* након учитавања су даље обрађивани и анализирани

Скупови података коришћени у примерима

скуп података **abalone** – припрема

```
1 # install.packages("tidyverse")
2
3 library(readr)
4 library(dplyr)
5 library(magrittr)
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

УЛАЗ

Скупови података коришћени у примерима

скуп података **abalon** – припрема

```
1 abalon <- read_csv("abalone.data",
2                   col_names=c("sex", "length", "diameter",
3                               "height", "weight_whole",
4                               "weight_shucked", "weight viscera",
5                               "weight_shell", "rings"),
6                   col_types="fdddddddi")
7
8 abalon %<>%
9   mutate(id=1:nrow(abalon), age=rings + 1.5) %>%
10  select(id, everything())
11
12 abalon %<>%
13   mutate(age_cat=factor(ifelse(age < median(age), "young", "old"),
14                       levels=c("young", "old")))
15
16
17
18
19
20
```

УЛАЗ

Скупови података коришћени у примерима

скуп података **abalon** – припрема

```
1 set.seed(5)
2
3 abalon.tst <- slice_sample(abalon, prop=0.2)
4 abalon.trn <- setdiff(abalon, abalon.tst)
5 nrow(abalon.tst)
6 nrow(abalon.trn)
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

УЛАЗ

1. Класификација
- 2. Регресија**
3. Метод најближих суседа
4. Метод потпорних вектора
5. Стабла одлучивања
6. Извори и литература

Линеарна регресија

општи облик

једнострука линеарна регресија

$$Y = \beta_0 + \beta_1 X_1 + \epsilon$$

вишеструка линеарна регресија

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \epsilon$$

елементи

циљно обележје

предикторска обележја

параметри (коефицијенти)

случајна грешка

Линеарна регресија

потребно спровести оцењивање параметара

може се искористити метод најмањих квадрата

примарно погодна за нумеричка циљна обележја

потребно проверити испуњеност више предуслова за примену

Линеарна регресија

пример

```
1 library(ggplot2)
2
3 linr <- lm(age ~ diameter + weight_whole, data=abalon)
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

УЛАЗ

Линеарна регресија

пример

```
> summary(linr)
```

```
Call:
```

```
lm(formula = age ~ diameter + weight_whole, data = abalon)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max
-5.2786 -1.6905 -0.7151  0.8972 15.9518
```

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	4.2254	0.2858	14.783	<2e-16	***
diameter	16.8749	1.0858	15.541	<2e-16	***
weight_whole	0.3925	0.2197	1.786	0.0741	.

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 2.638 on 4174 degrees of freedom
```

```
Multiple R-squared:  0.3307, Adjusted R-squared:  0.3304
```

```
F-statistic: 1031 on 2 and 4174 DF,  p-value: < 2.2e-16
```

КОНЗОЛА

Линеарна регресија

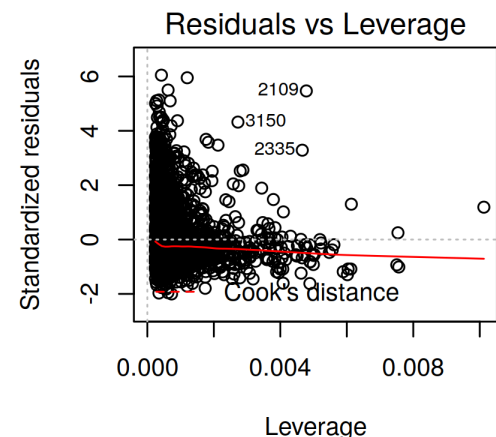
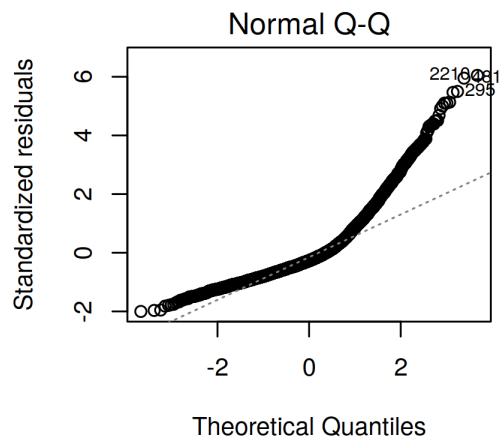
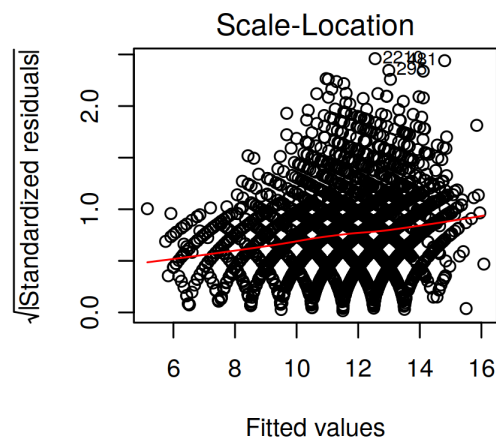
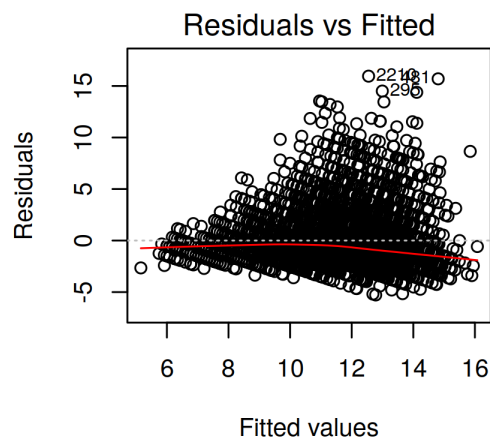
пример

```
> select(filter(abalon, id==1), age, diameter, weight_whole)
# A tibble: 1 x 3
  age diameter weight_whole
<dbl>   <dbl>   <dbl>
1  16.5    0.365    0.514
> predict(linr, newdata = abalon[1,])
      1
10.58652
>
```

КОНЗОЛА

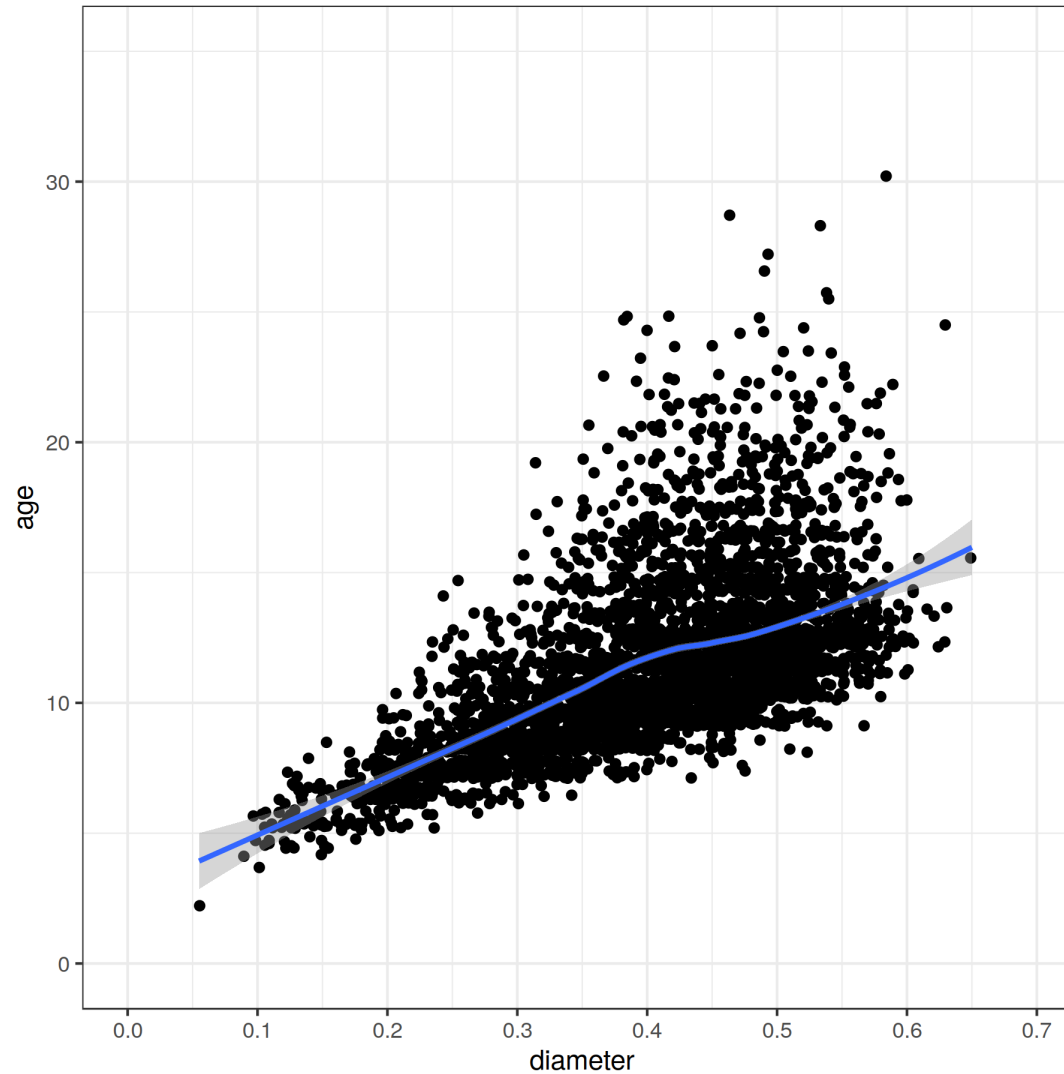
Линеарна регресија

пример



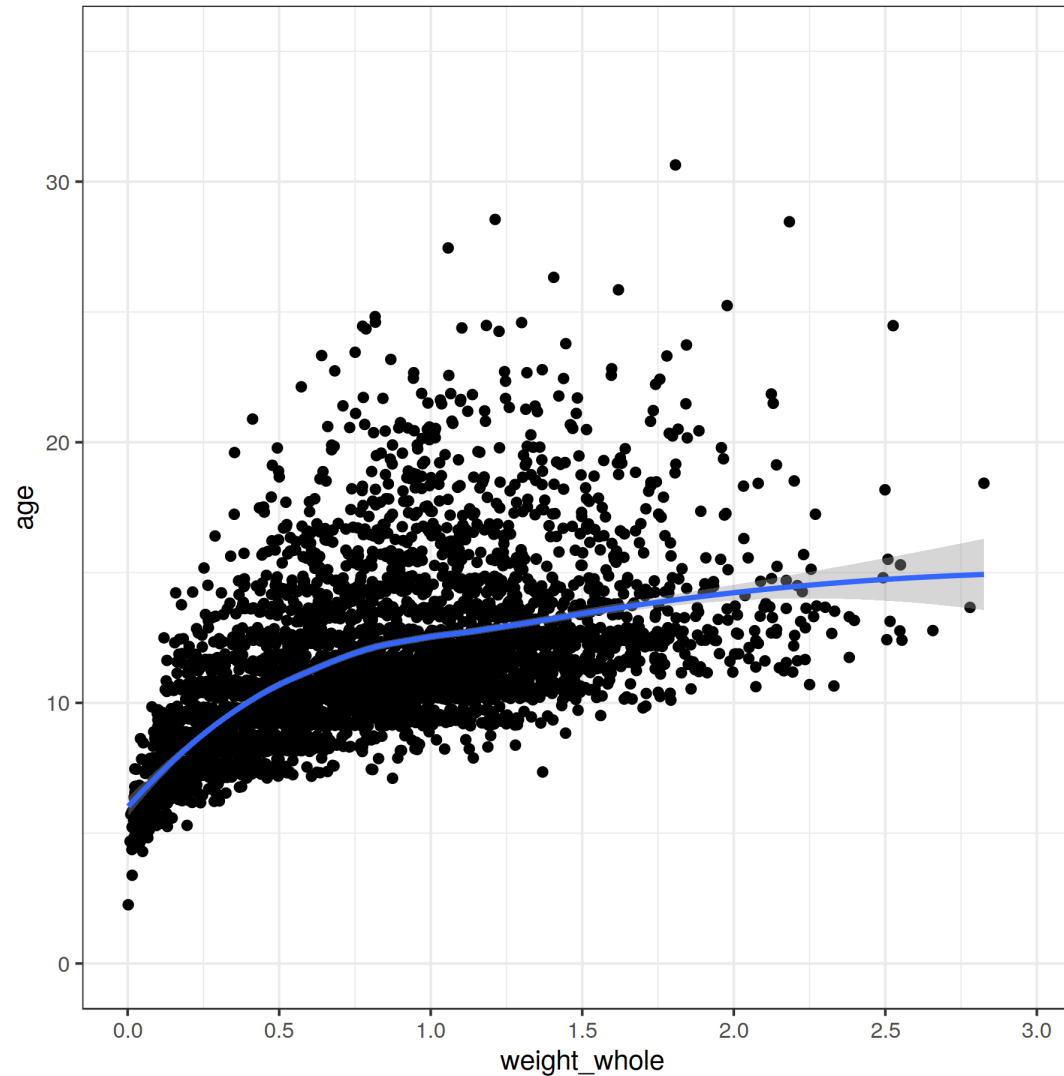
Линеарна регресија

пример



Линеарна регресија

пример



Логистичка регресија

општи облик

једнострука логистичка регресија

$$p(X) = Pr(Y = 1|X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X$$

вишеструка логистичка регресија

$$p(X) = Pr(Y = 1|X) = \frac{e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p}}$$

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p$$

Логистичка регресија

потребно спровести оцењивање параметара

може се искористити метод максималне веродостојности

погодна за категоријска циљна обележја

потребно проверити испуњеност више предуслова за примену

Логистичка регресија

пример

```
1 logr <- glm(age_cat ~ diameter + weight_whole, data=abalon.trn,  
2           family="binomial")  
3  
4 ver.logr <- predict(logr, newdata=abalon.tst, type="response")  
5  
6 klas.logr <- factor(ifelse(ver.logr > 0.5, "old", "young"),  
7 levels=c("young", "old"))  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20
```

УЛАЗ

Логистичка регресија

пример

```
> summary(logr)
```

```
Call:
```

```
glm(formula = age_cat ~ diameter + weight_whole, family = "binomial",  
     data = abalon.trn)
```

```
Deviance Residuals:
```

Min	1Q	Median	3Q	Max
-2.8053	-0.5798	0.3073	0.6247	2.2990

```
Coefficients:
```

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-5.3649	0.4516	-11.879	< 2e-16	***
diameter	12.6844	1.7516	7.241	4.44e-13	***
weight_whole	1.5411	0.3876	3.976	7.01e-05	***

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)
```

КОНЗОЛА

Логистичка регресија

пример

```
> head(ver.logr)
      1      2      3      4      5      6
0.5711173 0.9864490 0.9856813 0.9726682 0.1762120 0.9339484
> head(klas.logr)
      1      2      3      4      5      6
old   old   old   old young   old
Levels: young old
>
```

КОНЗОЛА

Логистичка регресија

пример

```
1 library(caret)
2
3 cm.logr.tst <- confusionMatrix(data=klas.logr,
4 reference=abalon.tst$age_cat)
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

УЛАЗ

Логистичка регресија

пример

```
> cm.logr.tst
Confusion Matrix and Statistics

          Reference
Prediction young old
young    177  53
old      99 506

      Accuracy : 0.818
      95% CI   : (0.7901, 0.8436)
No Information Rate : 0.6695
P-Value [Acc > NIR] : < 2.2e-16

      Kappa : 0.5706

McNemar's Test P-Value : 0.0002623

      Sensitivity : 0.6413
      Specificity : 0.9052
```

КОНЗОЛА

Логистичка регресија

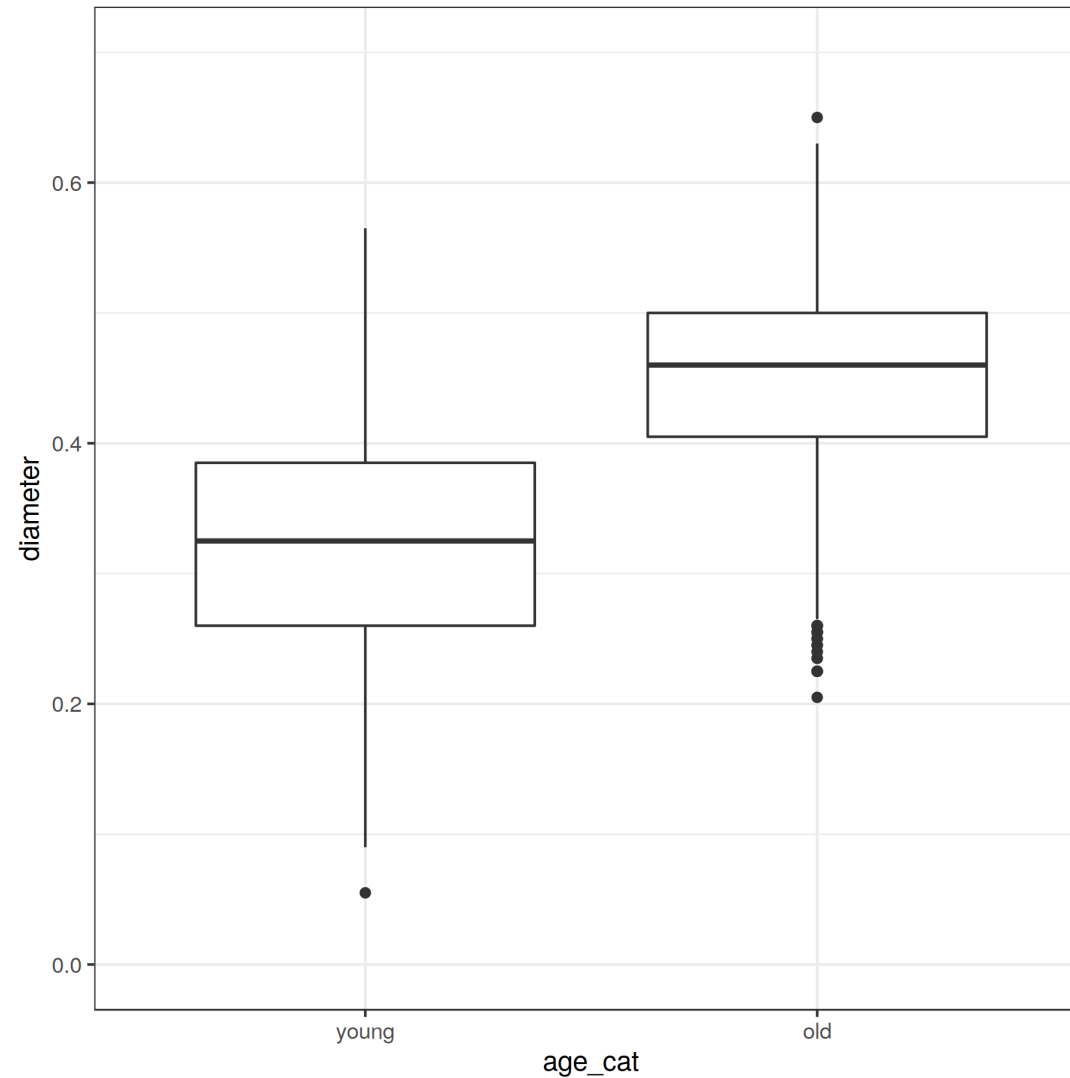
пример

```
> cm.logr.tst$overall[["Accuracy"]]  
[1] 0.8179641  
>
```

КОНЗОЛА

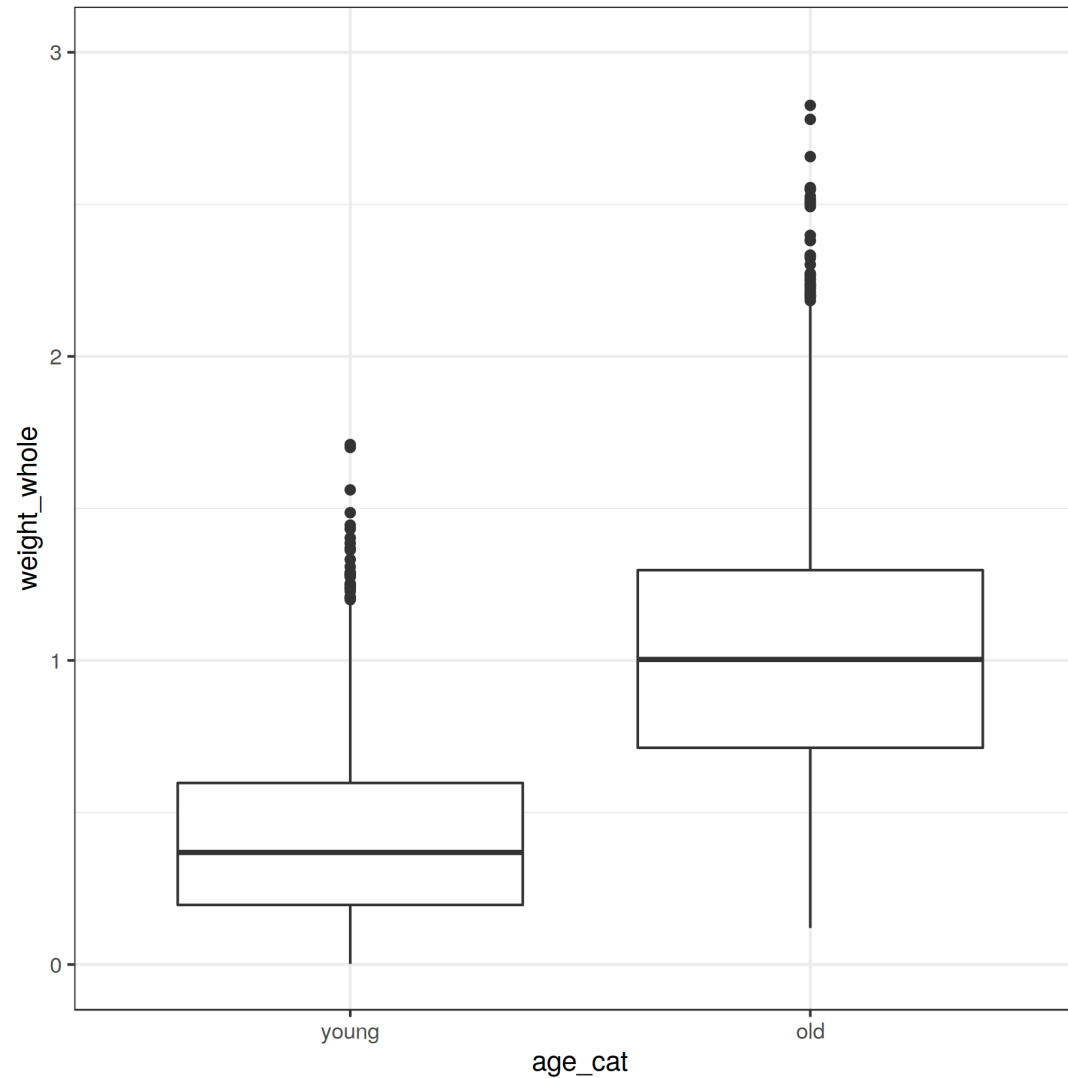
Логистичка регресија

пример



Логистичка регресија

пример



Садржај

1. Класификација
2. Регресија
- 3. Метод најближих суседа**
4. Метод потпорних вектора
5. Стабла одлучивања
6. Извори и литература

Метод најближих суседа

метод k најближих суседа

енгл. *k-nearest neighbours* (KNN)

нека су дати позитивни цео број k , појава x_0 и скуп K који садржи k појава за обучавање које су најближе појави x_0

могуће је проценити условну вероватноћу да појави x_0 одговара класа j

$$Pr(Y = j | X = x_0) = \frac{1}{k} \sum_{i \in K} I(y_i = j)$$

за класу која одговара појави x_0 одређује се класа са највећом процењеном вероватноћом

Метод најближих суседа

метод k најближих суседа

избор вредности k

могућа осетљивост на скале коришћених обележја

близина две појаве може бити одређена помоћу функције удаљености на основу вредности предикторских обележја

различите скале вредности код предикторских обележја могу утицати на удаљеност између појава

над предикторским обележјима може бити извршена стандардизација или поступак нормализације

Метод најближих суседа

стандардизација обележја

обухвата центрирање и скалирање вредности обележја

за обележје A нека су a_i вредност, μ_A аритметичка средина и σ_A стандардна девијација

тада је одговарајућа стандардизована вредност a_i^s

$$a_i^s = \frac{a_i - \mu_A}{\sigma_A}$$

Метод најближих суседа

Метод најближих суседа

пример

```
1 library(class)
2
3 klas.knn <- knn(select(abalon.trn, diameter, weight_whole),
4               select(abalon.tst, diameter, weight_whole),
5               abalon.trn$age_cat)
6
7 klas.knn.ska <- knn(scale(select(abalon.trn, diameter, weight_whole)),
8                    scale(select(abalon.tst, diameter, weight_whole)),
9                    abalon.trn$age_cat)
10
11
12
13
14
15
16
17
18
19
20
```

УЛАЗ

Метод најближих суседа

Метод најближих суседа

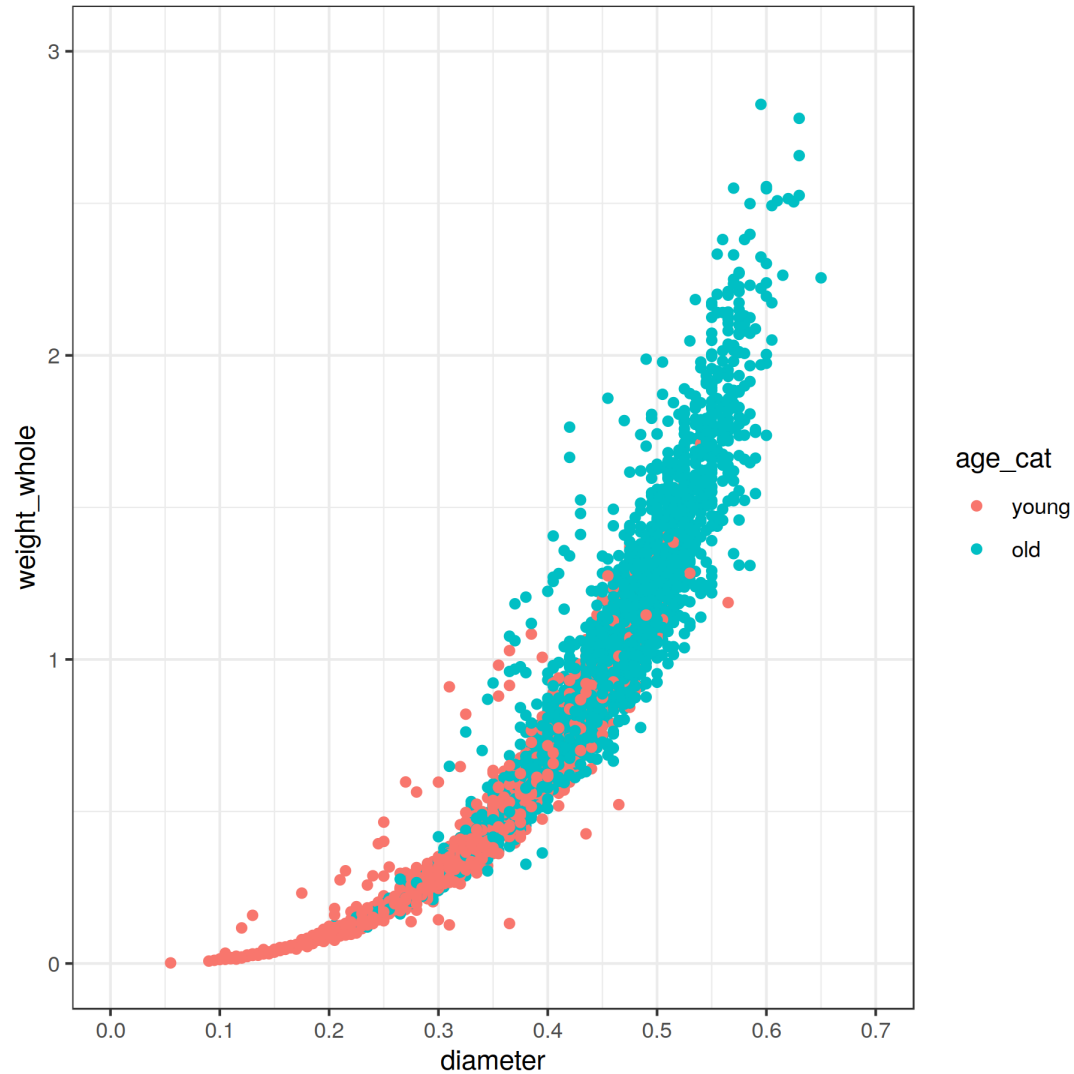
пример

```
> head(klas.knn)
[1] young old   old   old   young old
Levels: young old
> sum(klas.knn == abalon.tst$age_cat)
[1] 623
> sum(klas.knn == abalon.tst$age_cat) / nrow(abalon.tst)
[1] 0.7461078
> head(klas.knn.ska)
[1] young old   old   old   young old
Levels: young old
> sum(klas.knn.ska == abalon.tst$age_cat)
[1] 605
> sum(klas.knn.ska == abalon.tst$age_cat) / nrow(abalon.tst)
[1] 0.7245509
```

КОНЗОЛА

Метод најближих суседа

Метод најближих суседа пример



1. Класификација
2. Регресија
3. Метод најближих суседа
- 4. Метод потпорних вектора**
5. Стабла одлучивања
6. Извори и литература

Метод потпорних вектора

Метод потпорних вектора

класификатор потпорних вектора

енгл. *support vector classifier*

машина потпорних вектора

енгл. *support vector machine (SVM)*

Метод потпорних вектора

хиперраван

енгл. *hyperplane*

једначина хиперравни у p -димензионалном простору

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p = 0$$

хиперраван раздвајања

појаве које припадају различитим класама налазе се с различитих страна хиперравни

маргина

минимална удаљеност између појава и хиперравни раздвајања

оптимална хиперраван раздвајања (хиперраван максималне маргине)

хиперраван раздвајања код које је минимална удаљеност од појава највећа

хиперраван раздвајања не мора постојати у неком посматраном простору с појавама

Метод потпорних вектора

класификатор потпорних вектора

класификација појава применом хиперравни

класа дате појаве одређује се на основу тога с које стране хиперравни се појава налази

хиперраван се одређује у поступку решавања одговарајућег оптимизационог проблема

могуће да постоји појава из скупа за обучавање која се налази с неодговарајуће стране маргине или хиперравни
параметар C

хиперраван зависи од појава из скупа за обучавање које се налазе на маргини или с неодговарајуће стране маргине

потпорни вектори

сређени облик

нека је S скуп ознака потпорних вектора

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i \langle x, x_i \rangle$$

Метод потпорних вектора

машина потпорних вектора

примена хиперравни у раздвајању појава различитих класа

даљи развој класификатора потпорних вектора

могућа примена и у случају нелинеарног раздвајања класа

употреба кернел функције

простор с већим бројем димензија

параметар γ

код одабраних кернел функција

сређени облик

нека је S скуп ознака потпорних вектора а K кернел функција

$$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K(x, x_i)$$

Метод потпорних вектора

машина потпорних вектора

могућност n -арне класификације

приступ један-на-један (енгл. *one-versus-one*)

за сваки пар класа формира се један модел

модел се користи у испитивању припадности некој од класа из пара

користе се сви модели

најчешће одређена класа је резултат класификације

приступ сам-против-свих (енгл. *one-versus-all*)

за сваку класу формира се један модел

модел се користи у испитивању припадности главној класи модела или преосталим класама

користе се сви модели

најсигурније одређена класа је резултат класификације

Метод потпорних вектора

Метод потпорних вектора

пример

```
1 library(e1071)
2
3 suvm <- svm(age_cat ~ diameter + weight_whole, data=abalon.trn)
4
5 klas.suvm <- predict(suvm, newdata=abalon.tst)
6 cm.suvm.tst <- confusionMatrix(data=klas.suvm,
7 reference=abalon.tst$age_cat)
8
9
10
11
12
13
14
15
16
17
18
19
20
```

УЛАЗ

Метод потпорних вектора

Метод потпорних вектора

пример

```
> summary(svm)
```

```
Call:
```

```
svm(formula = age_cat ~ diameter + weight_whole, data = abalon.trn)
```

```
Parameters:
```

```
  SVM-Type:  C-classification  
  SVM-Kernel: radial  
    cost: 1
```

```
Number of Support Vectors: 1369
```

```
( 679 690 )
```

```
Number of Classes: 2
```

```
Levels:
```

```
young old
```

КОНЗОЛА

Метод потпорних вектора

Метод потпорних вектора

пример

```
> cm.svm.tst
Confusion Matrix and Statistics

          Reference
Prediction young old
  young   167  47
  old     109 512

      Accuracy : 0.8132
      95% CI   : (0.785, 0.8391)
  No Information Rate : 0.6695
  P-Value [Acc > NIR] : < 2.2e-16

      Kappa   : 0.5524

  McNemar's Test P-Value : 1.04e-06

      Sensitivity : 0.6051
      Specificity : 0.9159
```

КОНЗОЛА

Метод потпорних вектора

Метод потпорних вектора

пример

```
> cm.svm.tst$overall[["Accuracy"]]  
[1] 0.8131737  
>
```

КОНЗОЛА

1. Класификација
2. Регресија
3. Метод најближих суседа
4. Метод потпорних вектора
- 5. Стабла одлучивања**
6. Извори и литература

Формирање модела заснованих на стаблима одлучивања

стабло одлучивања

енгл. *decision tree*

метод *Random Forest*

поступак *XGBoost*

Стабло одлучивања

структура стабла одлучивања

посматраном чвору стабла одговара неки скуп појава које су из скупа за обучавање

скуп појава који одговара посматраном чвору обухвата појаве које се налазе у некој области простора вредности предикторских обележја

чвору стабла који је директно подређен посматраном чвору одговара неки подскуп од скупа појава који одговара посматраном чвору

скупови појава који одговарају чворовима који су директно подређени посматраном чвору заједно представљају једну партицију скупа појава који одговара посматраном чвору

партиција скупа појава настаје на основу вредности неког предикторског обележја

корену стабла одговара скуп појава који обухвата све појаве из скупа за обучавање
стабло одлучивања описује једну поделу простора вредности предикторских обележја на хијерархијски организоване области

Стабло одлучивања

формирање стабла одлучивања

примена поступка рекурзивне бинарне поделе

појаве за посматрани чвор стабла се разврставају (деле) у два подскупа на основу критеријума поделе чиме се добијају два подређена чвора

разврставање се изводи на основу одабраних вредности одабраног обележја обележје и вредности се бирају тако да разврставање појава у односу на класе буде што боље

повољност разврставања се одређује помоћу посебних показатеља (Џини индекс, ентропија...)

сваком подскупу одговара један нови директно подређени чвор стабла одлучивања

разврставање може бити спроведено и над сваким новим чвором разврставање за посматрани чвор се изводи ако није задовољен неки критеријум за завршетак

достигнут одређени минимални број појава у вези с чвором, достигнута одређена дубина у стаблу...

разврставање креће од корена стабла, којем одговарају све појаве из скупа за обучавање, и може бити даље примењивано на нове чворове стабла који настају

Стабло одлучивања

формирање стабла одлучивања

показатељи за одређивање повољности датог разврставања појава у два подскупа

утврђивање вредности одабраних показатеља за случај области r и скуп класа C

Џини индекс

$$G = \sum_{c \in C} t_c^r (1 - t_c^r)$$

ентропија

$$D = - \sum_{c \in C} t_c^r \log t_c^r$$

где је t_c^r удео појава из класе c међу свим појавама у области r

Стабло одлучивања

класификација за дату појаву

пролазак кроз стабло од корена к листовима

у сваком унутрашњем чвору испитује се критеријум поделе у односу на вредности предикторског обележја дате појаве

услов је придружен свакој од одлазних грана чвора

услов обухвата предикторско обележје и вредности

бира се она грана чији је придружени услов задовољен за дату појаву

следи се одабрана грана до наредног чвора

у сваком листу одређује се класа за дату појаву

одређена класа је вредност циљног обележја која је најзаступљенија међу појавама повезаним с листом

Стабло одлучивања

редукција („орезивање“) стабла

примењује се због ризика од преучености

стабло одлучивања може постати исувише велико те може бити погодно уклонити одређене делове

Стабла одлучивања

Стабло одлучивања

пример А

```
1 library(rpart)
2 library(rpart.plot)
3
4 dtree <- rpart(age_cat ~ diameter + weight_whole,
5               data=abalon.trn, method="class")
6
7 klas.dtree <- predict(dtree, newdata=abalon.tst, type="class")
8
9 cm.dtree.tst <- confusionMatrix(klas.dtree,
10                               reference=abalon.tst$age_cat)
11
12
13
14
15
16
17
18
19
20
```

УЛАЗ

Стабла одлучивања

Стабло одлучивања

пример А

```
> dtree
n= 3342

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

1) root 3342 1131 old (0.3384201 0.6615799)
 2) weight_whole < 0.626 1301 420 young (0.6771714 0.3228286)
 4) weight_whole < 0.34475 628 91 young (0.8550955 0.1449045) *
 5) weight_whole >= 0.34475 673 329 young (0.5111441 0.4888559)
 10) weight_whole < 0.44575 229 91 young (0.6026201 0.3973799) *
 11) weight_whole >= 0.44575 444 206 old (0.4639640 0.5360360) *
 3) weight_whole >= 0.626 2041 250 old (0.1224890 0.8775110) *
```

КОНЗОЛА

Стабло одлучивања

пример А

```
> cm.dtree.tst
Confusion Matrix and Statistics

          Reference
Prediction young old
young    157  43
old      119 516

      Accuracy : 0.806
      95% CI   : (0.7775, 0.8323)
No Information Rate : 0.6695
P-Value [Acc > NIR] : < 2.2e-16

      Карра : 0.5288

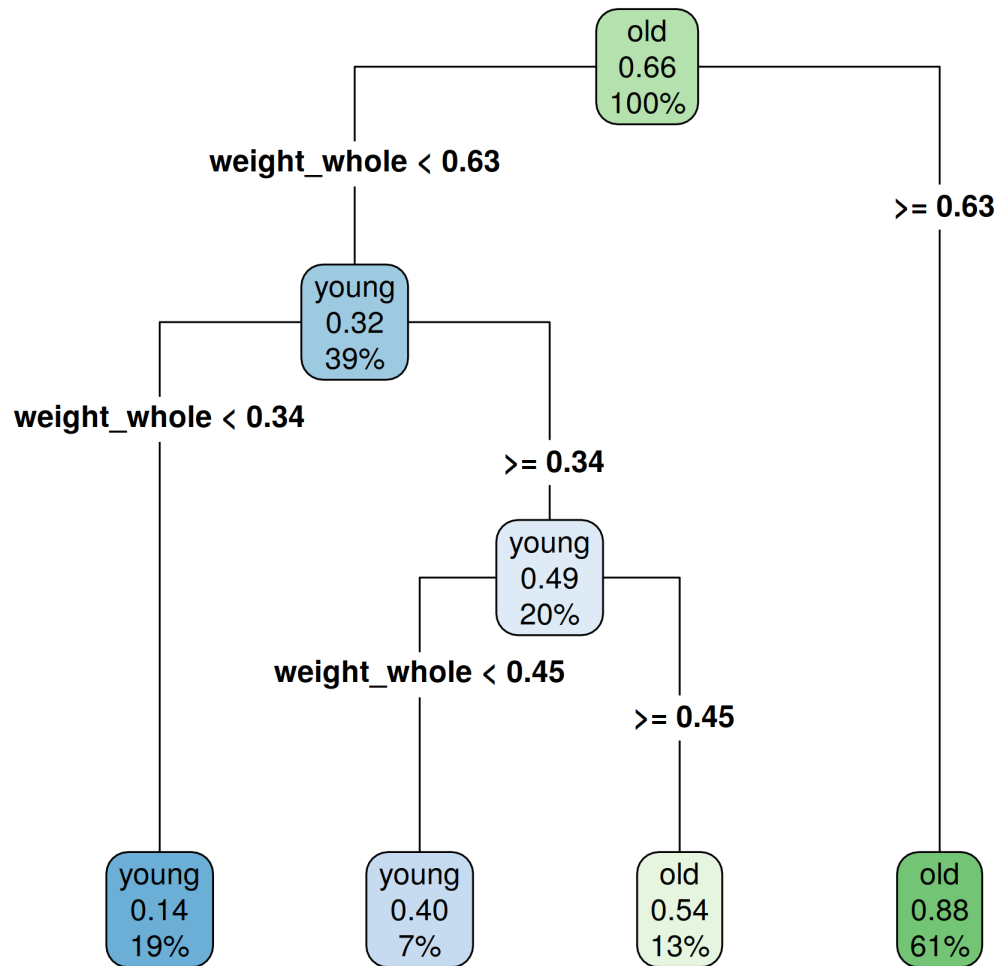
McNemar's Test P-Value : 3.803e-09

      Sensitivity : 0.5688
      Specificity : 0.9231
```

КОНЗОЛА

Стабла одлучивања

Стабло одлучивања пример А



Стабла одлучивања

Стабло одлучивања

пример Б

```
1 library(rpart)
2 library(rpart.plot)
3
4 dtree2 <- rpart(age_cat ~ sex + length + diameter + height +
5               weight_shucked + weight_shell,
6               data=abalon.trn, method="class",
7               parms=list(split="information"),
8               control=list(minsplit=10, cp=0.001, maxdepth=7))
9
10 klas.dtree2 <- predict(dtree2, newdata=abalon.tst, type="class")
11
12 cm.dtree2.tst <- confusionMatrix(klas.dtree2,
13                                reference=abalon.tst$age_cat)
14
15
16
17
18
19
20
```

УЛАЗ

Стабло одлучивања

пример Б

```
> cm.dtree2.tst
Confusion Matrix and Statistics

          Reference
Prediction young old
young    184  54
old      92 505

      Accuracy : 0.8251
      95% CI   : (0.7977, 0.8503)
No Information Rate : 0.6695
P-Value [Acc > NIR] : < 2.2e-16

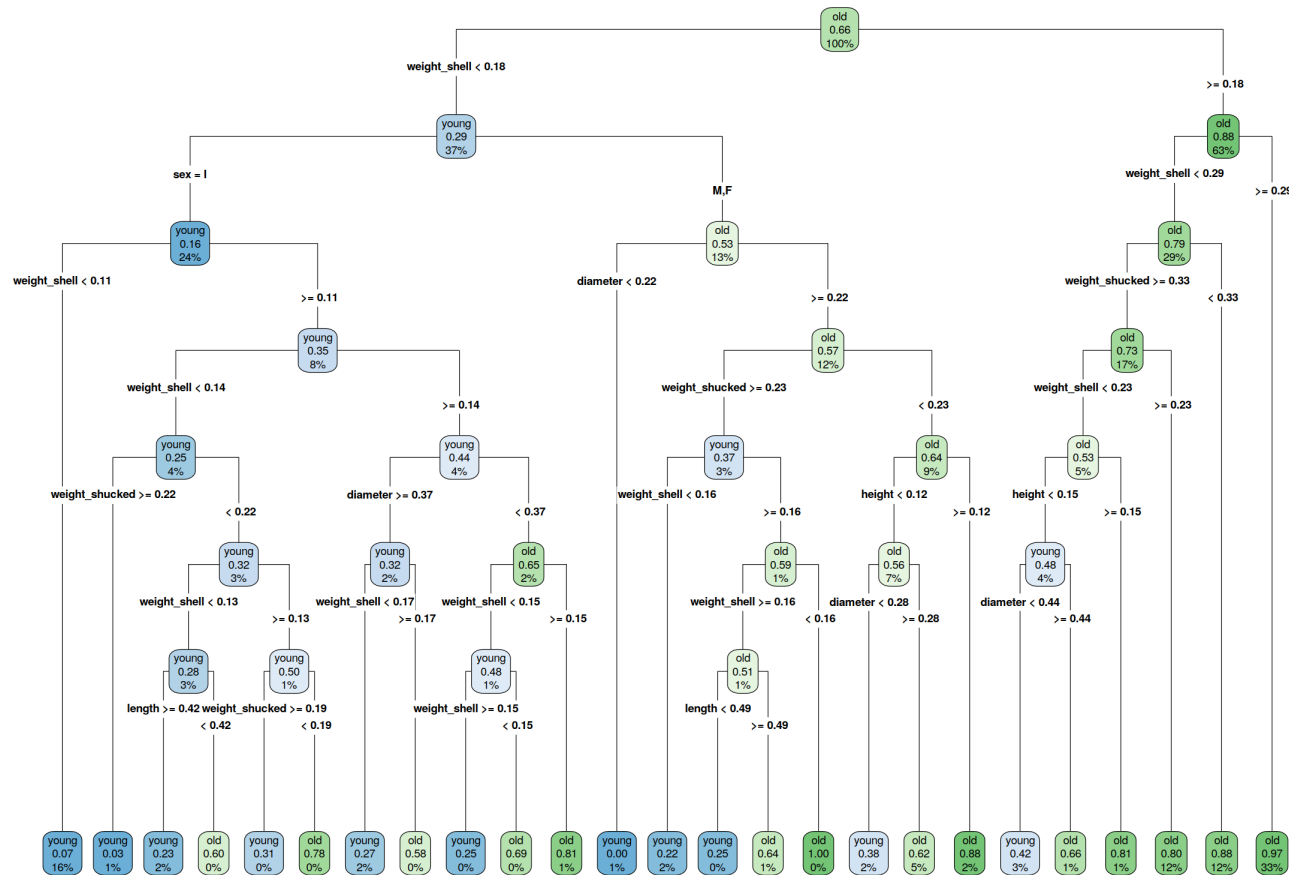
      Кappa   : 0.5907

McNemar's Test P-Value : 0.002198

      Sensitivity : 0.6667
      Specificity : 0.9034
```

КОНЗОЛА

Стабло одлучивања пример Б



Стабла одлучивања

Стабло одлучивања

пример Б – редукција („орезивање“)

```
1 dtree2.pruned <- prune(dtree2, cp=0.02)
2
3 klas.dtree2.pruned <- predict(dtree2.pruned, newdata=abalon.tst,
4                               type="class")
5
6 cm.dtree2.pruned.tst <- confusionMatrix(klas.dtree2.pruned,
7                                         reference=abalon.tst$age_cat)
8
9
10
11
12
13
14
15
16
17
18
19
20
```

УЛАЗ

Стабло одлучивања

пример Б – редукција („орезивање“)

```
> cm.dtree2.pruned.tst
Confusion Matrix and Statistics

          Reference
Prediction young old
  young   182  34
  old     94 525

      Accuracy : 0.8467
      95% CI   : (0.8205, 0.8705)
  No Information Rate : 0.6695
  P-Value [Acc > NIR] : < 2.2e-16

      Kappa   : 0.6335

McNemar's Test P-Value : 1.839e-07

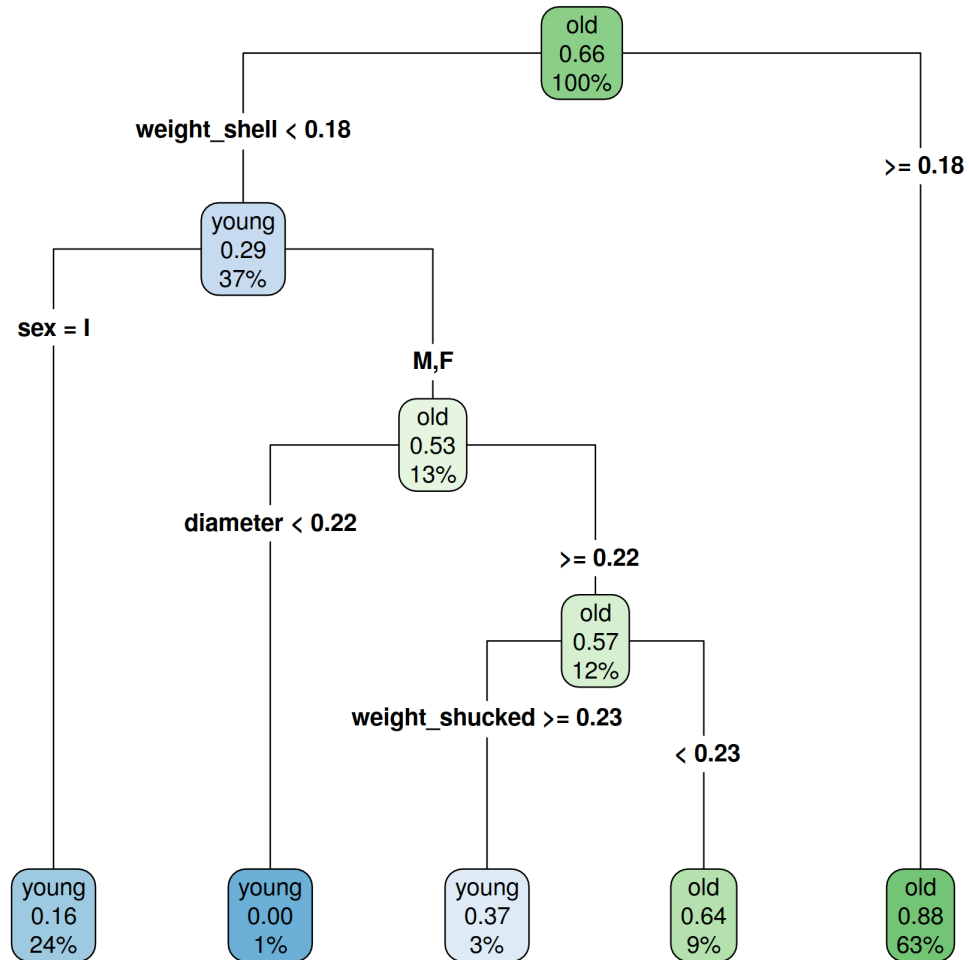
      Sensitivity : 0.6594
      Specificity : 0.9392
```

КОНЗОЛА

Стабла одлучивања

Стабло одлучивања

пример Б – редукција („орезивање“)



Стабло одлучивања

ПОГОДНОСТИ

- могућност визуалног приказа

 - погодност за тумачење

- објашњивост

- употреба категоријских предикторских обележја не захтева посебне додатне интервенције

- претпоставка да стабла одлучивања боље одговарају томе како људи доносе одлуке него неке друге методе

МАЊКАВОСТИ

- перформансе обично нису на нивоу перформанси неких других метода

- проблем робустности

 - мање промене у подацима могу осетније утицати на изглед стабла одлучивања

Метод *Random Forest*

формира се више стабала одлучивања

користи се више узорака с понављањем из скупа за обучавање
за сваки узорак формира се стабло одлучивања

не изводи се редукција стабла одлучивања

приликом поделе разматрају се обележја из случајног узорка предикторских обележја

процена грешке

одређивање класе појаве применом одређених модела

примена оних модела за чије формирање појава није коришћена

класификација за дату појаву

већинско гласање

Метод *Random Forest*

пример

```
1 library(randomForest)
2
3 rf <- randomForest(age_cat ~ sex + length + diameter + height +
4                   weight_shucked + weight_shell,
5                   data=abalone.trn, ntree=300, mtry=2)
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

УЛАЗ

Метод *Random Forest*

пример

```
> rf

Call:
randomForest(formula = age_cat ~ sex + length + diameter + height +
weight_shucked + weight_shell, data = abalon.trn, ntree = 300, mtry =
2)

      Type of random forest: classification
      Number of trees: 300
No. of variables tried at each split: 2

      OOB estimate of error rate: 15.32%
Confusion matrix:
      young  old class.error
young   819  312  0.27586207
old     200 2011  0.09045681
>
```

КОНЗОЛА

Стабла одлучивања

Метод *Random Forest*

пример

```
> head(getTree(rf, 5))
left daughter right daughter split var split point status prediction
1           2           3           4      0.12250           1           0
2           4           5           4      0.10250           1           0
3           6           7           3      0.44250           1           0
4           8           9           6      0.09925           1           0
5          10          11          6      0.16950           1           0
6          12          13          6      0.22450           1           0
>
```

КОНЗОЛА

Метод *Random Forest*

пример

```
1 klas.rf <- predict(rf, newdata=abalon.tst, type="response")
2
3 cm.rf.tst <- confusionMatrix(klas.rf,
4                             reference=abalon.tst$age_cat)
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

УЛАЗ

Метод *Random Forest*

пример

```
> cm.rf.tst
Confusion Matrix and Statistics

          Reference
Prediction young old
  young   186  41
  old     90 518

      Accuracy : 0.8431
      95% CI   : (0.8166, 0.8671)
  No Information Rate : 0.6695
  P-Value [Acc > NIR] : < 2.2e-16

      Карра   : 0.6288

McNemar's Test P-Value : 2.743e-05

      Sensitivity : 0.6739
      Specificity : 0.9267
```

КОНЗОЛА

Метод *Random Forest*

пример

```
> predict(rf, newdata=abalon.tst, type="prob")[1:5, ]  
      young      old  
1 0.4300000 0.5700000  
2 0.0000000 1.0000000  
3 0.0000000 1.0000000  
4 0.1300000 0.8700000  
5 0.7366667 0.2633333  
>
```

КОНЗОЛА

Стабла одлучивања

Метод *Random Forest*

пример

```
> predict(rf, newdata=abalon.tst, type="vote", norm.votes=F)[1:5, ]
  young old
1    129 171
2     0 300
3     0 300
4     39 261
5    221  79
> abalon.tst[1:5, c(1:5, 7, 9, 11)]
# A tibble: 5 x 8
   id sex  length diameter height weight_shucked weight_shell age
  <int> <fct> <dbl>    <dbl> <dbl>    <dbl>    <dbl> <dbl>
1   837 M     0.47    0.375  0.12    0.266    0.169  9.5
2  2862 F     0.72    0.565  0.17    0.723    0.494 13.5
3  3828 M     0.68    0.54   0.195    0.556    0.428 12.5
4  1188 M     0.685   0.52   0.165    0.699    0.4   11.5
5   437 I     0.36    0.275  0.095    0.084    0.09  8.5
>
```

КОНЗОЛА

Поступак *XGBoost*

пример

```
1 library(xgboost)
2 library(forcats)
3
4 abalon.trn.num <- abalon.trn %>%
5   select(sex:height, weight_shucked, weight_shell) %>%
6   mutate(sex=as.numeric(sex)) %>%
7   as.matrix()
8 abalon.trn.num.klas <- as.numeric(abalon.trn$age_cat) - 1
9
10 abalon.tst.num <- abalon.tst %>%
11   select(sex:height, weight_shucked, weight_shell) %>%
12   mutate(sex=as.numeric(sex)) %>%
13   as.matrix()
14 abalon.tst.num.klas <- as.numeric(abalon.tst$age_cat) - 1
15
16
17
18
19
20
```

УЛАЗ

Поступак *XGBoost*

пример

```
1 xgbo <- xgboost(data=abalon.trn.num,  
2 label=abalon.trn.num.klas,  
3 eta=0.1, max_depth=2, nround=100,  
4 objective="binary:logistic",  
5 verbose=1, print_every_n=10)  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20
```

УЛАЗ

Поступак *XGBoost*

пример

```
1 klas.xgbo <- predict(xgbo, newdata=abalon.tst.num)
2
3 cm.xgbo.tst <- confusionMatrix(as_factor(round(klas.xgbo)),
4                               reference=as_factor(abalon.tst.num.klas))
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

УЛАЗ

Поступак *XGBoost*

пример

```
> cm.xgbo.tst
Confusion Matrix and Statistics

          Reference
Prediction  0    1
          0 183  33
          1  93 526

          Accuracy : 0.8491
          95% CI : (0.823, 0.8727)
    No Information Rate : 0.6695
    P-Value [Acc > NIR] : < 2.2e-16

          Kappa : 0.6392

McNemar's Test P-Value : 1.471e-07

          Sensitivity : 0.6630
          Specificity : 0.9410
```

КОНЗОЛА

Паралелизација за стабла одлучивања

примери скалабилних алгоритама

алгоритам *SLIQ*

алгоритам *SPRINT*

Паралелизација за стабла одлучивања

алгоритам *SPRINT*

енгл. *Scalable Parallelizable Induction of Decision Trees*

рад Шафера и сарадника (1996)

фазе изградње класификационог модела

- фаза формирања стабла одлучивања

- фаза редукције стабла одлучивања

две варијанте алгоритма

- серијска варијанта

- паралелна варијанта

могућност паралелизације

- фаза формирања може бити паралелизована

- фаза редукције изводи се серијски

 - фаза редукције релативно незахтевна у рачунском погледу у односу на фазу формирања

Паралелизација за стабла одлучивања

алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – општи ток

заснива се на рекурзивној подели

бинарна подела се користи у основи

могуће уопштење на поделе вишег реда

на почетку поступак поделе примењује се над целокупним скупом за обучавање (целокупни скуп за обучавање узима се као текући скуп података)

поступак поделе над текућим скупом података

1. провера да ли све појаве у текућем скупу података припадају истој класи
ако припадају, онда је поступак поделе над текућим скупом података завршен
2. за свако од обележја оцењују се могућа разврставања по том обележју
3. према разврставању које је најбоље оцењено изводи се подела текућег скупа података на први и други подскуп
4. поступак поделе примењује се над првим подскупом
5. поступак поделе примењује се над другим подскупом

Паралелизација за стабла одлучивања

алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – помоћне структуре података

два врсте помоћних структура података

листа вредности обележја

хистограм

Паралелизација за стабла одлучивања

алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – помоћне структуре података

листа вредности обележја

односи се на једно обележје и садржи сваку вредност тог обележја која се појављује у посматраном скупу података

структура елемента листе

вредност обележја на које се односи листа

одговарајућа вредност циљног обележја (ознака класе)

ознака одговарајуће појаве из посматраног скупа података

ако је обележје нумеричке природе, онда је листа сортирана по вредности обележја на које се односи листа

Паралелизација за стабла одлучивања

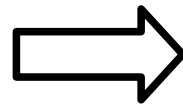
алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – помоћне структуре података

листа вредности обележја – пример

посматрани скуп
података

	X_1	X_2	Y
0	a	5	1
1	b	2	0
2	c	3	0
3	a	7	1
4	b	4	1



листа вредности
обележја X_1

X_1	Y	Id
a	1	0
b	0	1
c	0	2
a	1	3
b	1	4

листа вредности
обележја X_2

X_2	Y	Id
2	0	1
3	0	2
4	1	4
5	1	0
7	1	3

Паралелизација за стабла одлучивања

алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – помоћне структуре података

хистограм

односи се на једно обележје и приказује расподелу ознака класа у односу на вредности тог обележја

случај нумеричког обележја

пар хистограма за нумеричко обележје – доњи и горњи хистограм

доњи хистограм приказује расподелу ознака класа за део листе вредности обележја који обрађен

горњи хистограм приказује расподелу ознака класа за део листе вредности обележја који није обрађен

случај категоријског обележја

хистограм за категоријско обележје – матрица учесталости

матрица учесталости приказује расподелу ознака класа по свим вредностима обележја

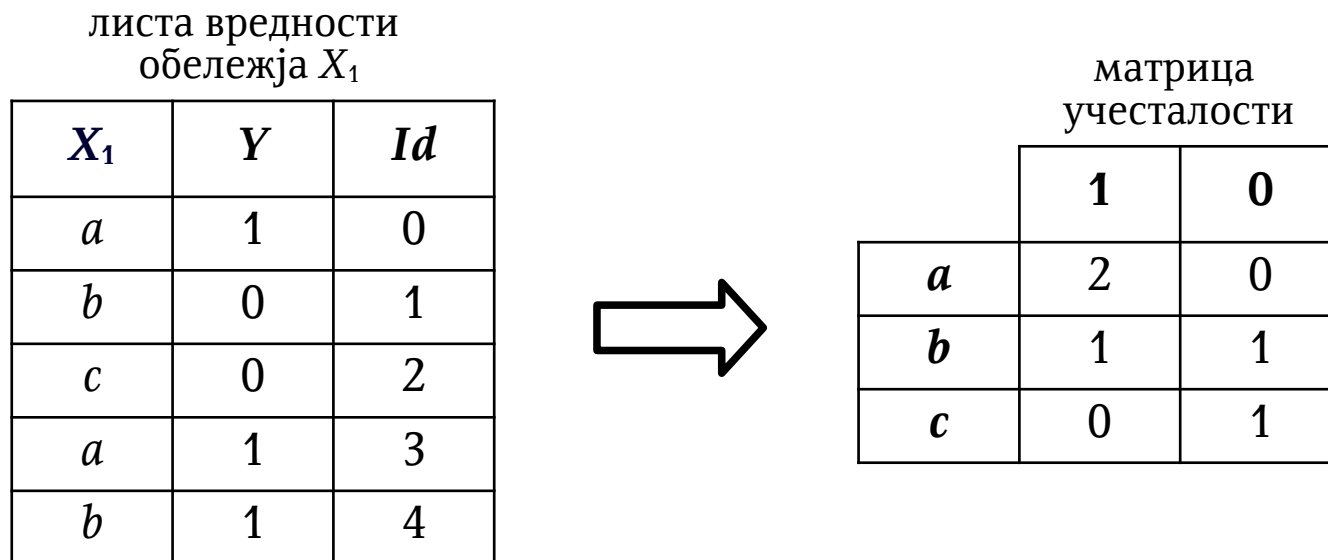
Стабла одлучивања

Паралелизација за стабла одлучивања

алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – помоћне структуре података

хистограм – пример



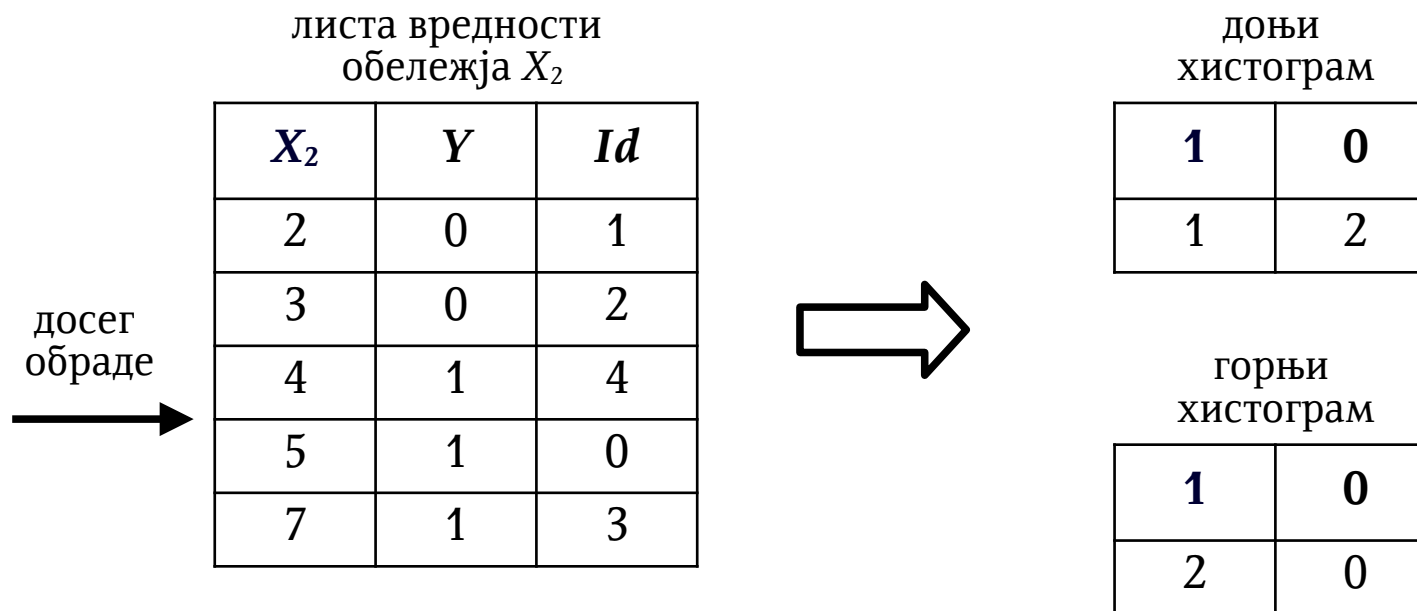
Стабла одлучивања

Паралелизација за стабла одлучивања

алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – помоћне структуре података

хистограм – пример



Паралелизација за стабла одлучивања

алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – анализа могућих разврставања

приликом формирања стабла одлучивања сваки текући скуп података одговара једном чвору стабла одлучивања

на почетку, посматра се целокупан скуп за обучавање, који одговара корену стабла одлучивања

поделом текућег скупа података добијају се подскупови података

подскупови података одговарају новим чворовима стабла одлучивања који су директно подређени чвору којем одговара текући скуп података

рекурзивно се изводе подела и формирање нових чворова стабла одлучивања докле год има потребе

листе вредности обележја настају за текући скуп података, односно чвор стабла одлучивања

по једна листа вредности обележја за свако од обележја

ако је потребно поделити текући скуп података изводи се трагање за најбољим разврставањем његових појава

на основу листа вредности обележја изводи се анализа могућих разврставања

Паралелизација за стабла одлучивања

алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – анализа могућих разврставања
повољност неког разврставања појава одређује се применом Џини индекса

$$G(A) = 1 - \sum_{c \in C} t_c^2$$

$$G_s(A) = \frac{n_1}{n} G(A_1) + \frac{n_2}{n} G(A_2)$$

где је A текући скуп података, C скуп класа, t_c удео појава из класе c међу свим појавама у скупу података A , s разврставање појава скупа података A којим настају скупови података A_1 и A_2 , а n , n_1 и n_2 редом одговарају бројевима појавама у скуповима података A , A_1 и A_2

Паралелизација за стабла одлучивања

алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – анализа могућих разврставања

подела се изводи применом најбољег разврставања

разврставање појава које има најмању вредност Џини индекса сматра се најбољим

могућа разврставања појава бивају пронађена на основу листа вредности обележја за текући скуп података

за сваку листу вредности обележја утврђују се могућа разврставања по вредностима тог обележја

разлике у поступку у зависности од тога да ли је обележје категоријско или нумеричко

Паралелизација за стабла одлучивања

алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – анализа могућих разврставања

подела се изводи применом најбољег разврставања – категоријско обележје на основу листе вредности обележја изводи се хистограм за категоријско обележје (матрица учесталости)

за подскупове вредности обележја као основе за могуће критеријуме поделе рачунају се вредности Џини индекса

у случају да је број могућих вредности обележја изнад одређене границе, примењује се посебан поступак заснован на похлепном принципу

критеријум поделе своди се на проверу да ли вредност обележја у некој појави припада правом подскупу скупа могућих вредности обележја

Паралелизација за стабла одлучивања

алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – анализа могућих разврставања
подела се изводи применом најбољег разврставања – категоријско обележје

листа вредности
обележја X_1

X_1	Y	Id
a	1	0
b	0	1
c	0	2
a	1	3
b	1	4



матрица
учесталости

	1	0
a	2	0
b	1	1
c	0	1

Паралелизација за стабла одлучивања

алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – анализа могућих разврставања

подела се изводи применом најбољег разврставања – нумеричко обележје на основу листе вредности обележја изводе се парови хистограма за нумеричко обележје (парови доњи хистограм – горњи хистограм)

редом се обрађује листа вредности обележја

за појединачне поделе листе вредности обележја формирају се парови хистограма за нумеричко обележје

за појединачне поделе листе вредности обележја као основе за могуће критеријуме поделе рачунају се вредности Цини индекса

критеријум поделе своди се на проверу да ли је вредност обележја у некој појави мања од неке вредности раздвајања

за могуће вредности раздвајања узимају се вредности које су подједнако удаљене од две узастопне вредности обележја

Стабла одлучивања

Паралелизација за стабла одлучивања

алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – анализа могућих разврставања
подела се изводи применом најбољег разврставања – нумеричко обележје



Паралелизација за стабла одлучивања

алгоритам *SPRINT* – серијска варијанта

фаза формирања стабла одлучивања – подела

подела текућег скупа података се изводи применом најбољег разврставања критеријум поделе је у сагласности с начином разврставања појава према најбољем разврставању

поделом настају први и други подскуп

поделом текућег скупа података долази и до поделе листа вредности обележја за свако обележје долази до поделе листе вредности обележја

у сагласности с критеријумом поделе

одговарајући делови листа вредности обележја везују се за одговарајуће подскупове текућег скупа података

свака нова листа вредности обележја у сагласности с одговарајућим подскупом

за обележје које је заступљено у критеријуму поделе релативно једноставно се изводи подела одговарајуће листе вредности обележја

непосредно на основу критеријума поделе

хеш табела где ознаке појава повезане с одговарајућим подскуповима

за обележја која нису заступљена у критеријуму поделе поделе листа вредности обележја нешто компликованије

на основу хеш табеле и ознаке појаве могуће елемент листе вредности обележја повезати с одговарајућим подскупом

Паралелизација за стабла одлучивања
алгоритам *SPRINT* – серијска варијанта
фаза редукције стабла одлучивања
према редукцији која се користи у алгоритму *SLIQ*

Паралелизација за стабла одлучивања

алгоритам *SPRINT* – паралелна варијанта

паралелизација формирања стабла одлучивања

нема централизованих структура података

примена више процесора без дељења

равномерно размештање појава на више процесора

равномерно размештање листа вредности обележја на више процесора

потребно водити рачуна о томе да листе вредности обележја која су нумеричка буду сортиране

примена алгоритма за паралелно сортирање

матрице учесталости за категоријска обележја потребно прикупити од свих процесора ради добијања општих учесталости

приликом поделе потребно прикупити ознаке појава од свих процесора

Додатак

формуле у језику R

знак \sim

однос између зависне и независне променљиве

$$y \sim x$$

знак $+$

линеарна веза између променљивих

$$y \sim x_1 + x_2$$

знак 0

уклањање слободног члана

$$y \sim 0 + x$$

знак 1

експлицитно навођење слободног члана

$$y \sim 1 + x$$

Додатак

формуле у језику R

функција $I()$

очување уобичајене аритметичке интерпретације задатог израза

$$y \sim x + I(x^2) + I(x^3)$$

знак $:$

интеракција између променљивих

$$y \sim x_1 : x_2$$

знак $*$

променљиве са интеракцијама између променљивих

$$y \sim x_1 * x_2$$

исто као $y \sim x_1 + x_2 + I(x_1 * x_2)$

знак $^$

променљиве са интеракцијама између променљивих, до одређеног степена

$$y \sim (x_1 + x_2 + x_3)^2$$

исто као $y \sim (x_1 + x_2 + x_3) * (x_1 + x_2 + x_3)$

Додатак

формуле у језику R

знак -

уклањање променљивих

$$y \sim a * b - a : b$$

знак .

означава све оне променљиве које нису већ експлицитно наведене

$$y \sim .$$

1. Класификација
2. Регресија
3. Метод најближих суседа
4. Метод потпорних вектора
5. Стабла одлучивања
- 6. Извори и литература**

Основни извори и литература

- ◆ James G, Witten D, Hastie T, Tibshirani R. An introduction to statistical learning: With applications in R. Springer; 2013.
- ◆ Shalev-Shwartz S, Ben-David S. Understanding machine learning: From theory to algorithms. Cambridge University Press; 2014.
- ◆ Mehta M, Agrawal R, Rissanen J. SLIQ: A fast scalable classifier for data mining. Proceedings of the 5th International Conference on Extending Database Technology; 1996 March 25-29; Avignon, France. Berlin, Heidelberg (Germany): Springer; 1996. p. 18-32.
- ◆ Shafer J, Agrawal R, Mehta M. SPRINT: A scalable parallel classifier for data mining. Proceedings of the 22nd International Conference on Very Large Data Bases; 1996 September 3-6; Mumbai (Bombay), India. San Francisco (CA, USA): Morgan Kaufmann Publishers Inc.; 1996. p. 544-555.

Основни извори и литература

- ◆ R Project. R: A language and environment for statistical computing – Reference index – The R Core Team – Version 4.5.1 (2025-06-13). Internet: <https://cran.r-project.org/doc/manuals/r-release/fullrefman.pdf>
- ◆ R Project. CRAN: Package ggplot2. Internet: <https://cran.r-project.org/web/packages/ggplot2/index.html>
- ◆ R Project. CRAN: Package forcats. Internet: <https://cran.r-project.org/web/packages/forcats/index.html>
- ◆ R Project. CRAN: Package caret. Internet: <https://cran.r-project.org/web/packages/caret/index.html>
- ◆ R Project. CRAN: Package e1071. Internet: <https://cran.r-project.org/web/packages/e1071/index.html>
- ◆ R Project. CRAN: Package rpart. Internet: <https://cran.r-project.org/web/packages/rpart/index.html>
- ◆ R Project. CRAN: Package rpart.plot. Internet: <https://cran.r-project.org/web/packages/rpart.plot/index.html>

Основни извори и литература

- ◆ R Project. CRAN: Package randomForest. Internet: <https://cran.r-project.org/web/packages/randomForest/index.html>
- ◆ R Project. CRAN: Package xgboost. Internet: <https://cran.r-project.org/web/packages/xgboost/index.html>

Основни извори података

- ◆ скуп података **abaLon**
 - ◆ скуп података *Abalone*
 - ◆ аутори *Warwick Nash, Tracy Sellers, Simon Talbot, Andrew Cawthorn* и *Wes Ford*
 - ◆ електронска локација (доступни скуп података и пратеће информације)
 - ◆ <https://archive.ics.uci.edu/dataset/1/abalone>
 - ◆ <https://doi.org/10.24432/C55C7W>
 - ◆ репозиторијум *UC Irvine Machine Learning Repository* на електронској локацији <http://archive.ics.uci.edu/>
 - ◆ скуп података дониран за репозиторијум 30. 11. 1995.
 - ◆ датотека *abalone.zip* (преузето 27. 3. 2024)
 - ◆ електронска локација
 - ◆ <https://archive.ics.uci.edu/static/public/1/abalone.zip>
 - ◆ подаци у датотеци *abalone.data*
 - ◆ лиценца *Creative Commons Attribution 4.0 International (CC BY 4.0)*
 - ◆ електронска локација
 - ◆ <https://creativecommons.org/licenses/by/4.0/legalcode>
 - ◆ скуп података *Abalone* читан, обрађиван и анализиран језиком *R*

Мастер академске студије
Рачунарство и аутоматика

Рачунарство високих перформанси
у информационом инжењерингу

Основи класификације

(материјали за предавања)