ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

Класификација

Дрва + евалуација

Д-р Ана Мадевска Богданова

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ
И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

ДРВА НА ОДЛУЧУВАЊЕ

Дрва на одлучување - вовед

- Учењето со дрвата на одлучување е една од најупотребуваните техники за класификација.
  - Резултатите се споредливи со останатите методи
  - Многу се ефикасни.
- Класификацискиот модел е дрво, наречено дрво на одлучување.

Класификација базирана на дрва на одлучување

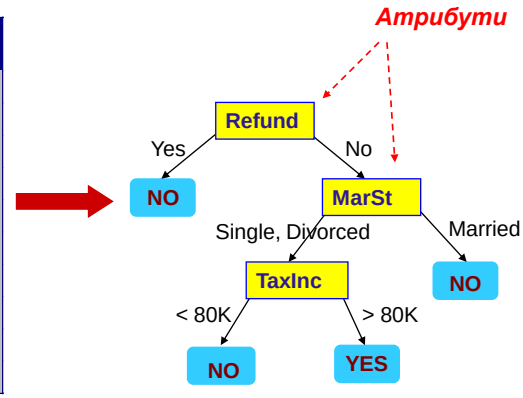
- Предности:
  - Конструирањето не е скапо
  - Многу брзи во класификување непознати облици (вектори, записи)
  - Мали дрва се лесни **за интерпретација**
  - За едноставни множества податоци, точноста на класификувањето е споредлива со останатите техники на класификување

Дрва на одлучување, пример

1. Фаза на градење модел

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

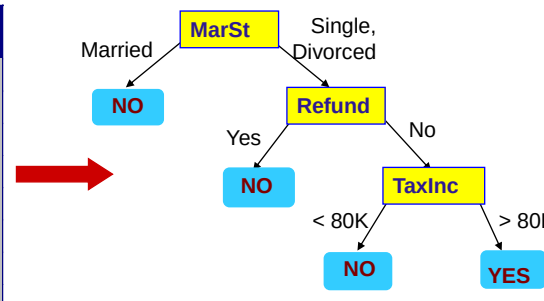
Обучувачки податоци



Модел: Дрво на одлучување

Дрва на одлучување, пример

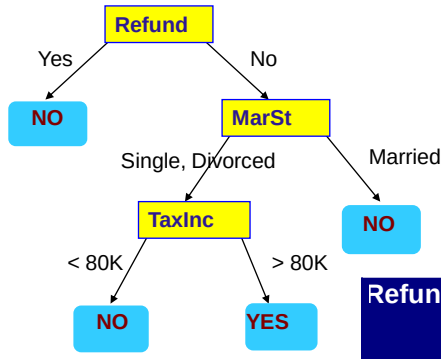
	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Може да има повеќе дрва што одговараат на исто множество податоци

2. Фаза на класификација

Старт од коренот на дрвото



Тест податок

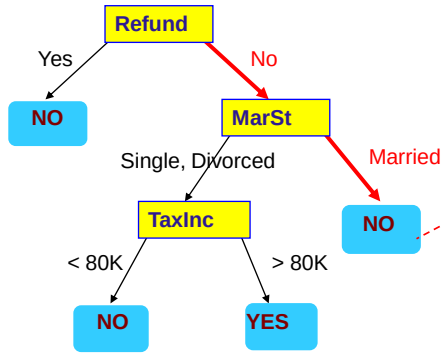
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ
И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

2. Фаза на класификација

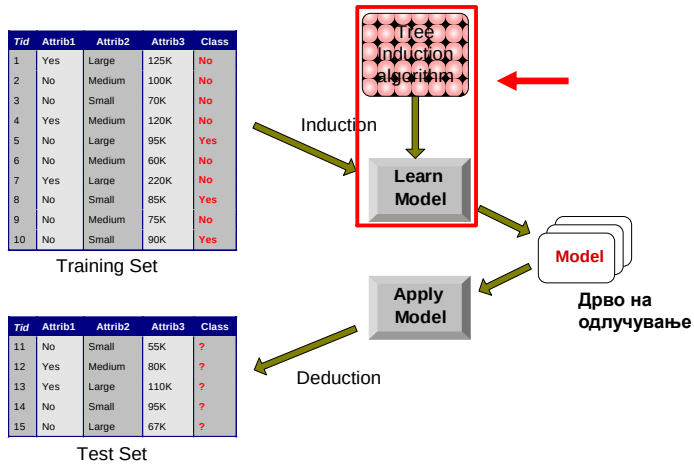
Тест податок

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Означи го Cheat на “No”

Класификација со дрва на одлучување



Индукција на дрво

- Многу алгоритми:
  - Hunt-ов алгоритам (еден од најраните)
  - CART
  - ID3, C4.5
  - SLIQ,SPRINT

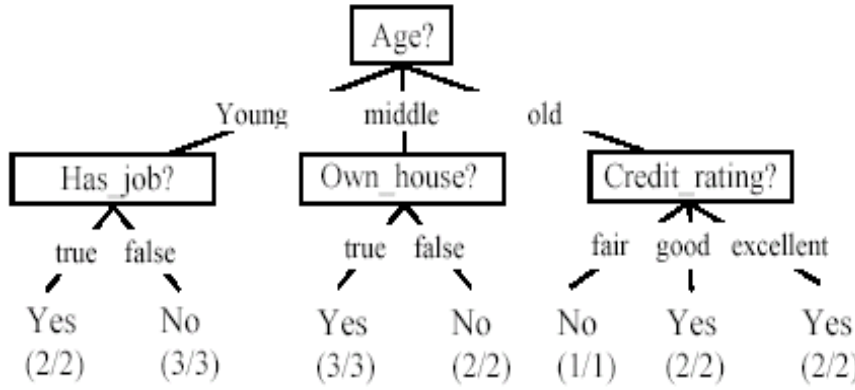
- C4.5 од Ross Quinlan е можеби најдобриот систем

Уште еден пример

Дозволен кредит YES / NO

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	**No**
2	young	false	false	good	**No**
3	young	true	false	good	**Yes**
4	young	true	true	fair	**Yes**
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	middle	true	true	good	**Yes**
9	middle	false	true	excellent	**Yes**
10	middle	false	true	excellent	**Yes**
11	old	false	true	excellent	**Yes**
12	old	false	true	good	**Yes**
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**

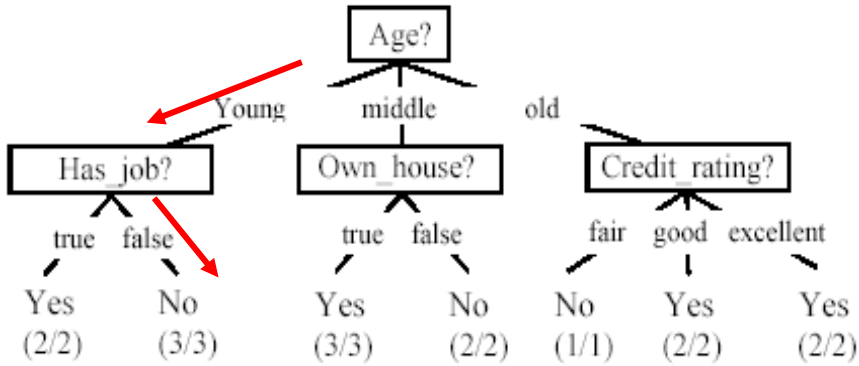
Дрво на одлучување за табелата



Со користење на дрво на одлучување

Age	Has_Job	Own_house	Credit-Rating	Class
young	false	false	good	?

No



Може да има повеќе дрва

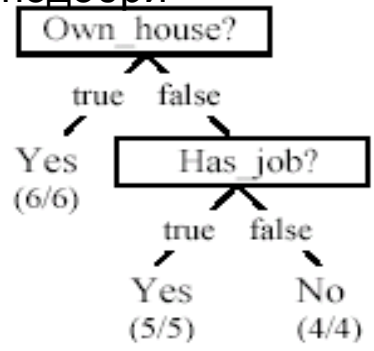
Еве едно поедноставно дрво.

Потребно е помало дрво и да биде точно дрво

Лесно за разбирање и со подобри перформанси.

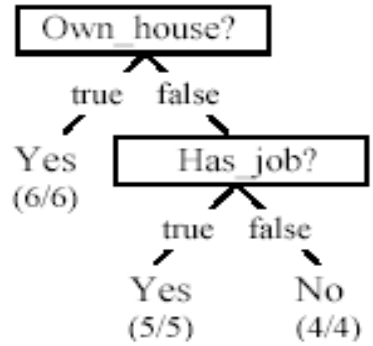
Наоѓање на најдоброто дрво е NP-проблем.

Сите постоечки алгоритми за градење дрва се ХЕВРИСТИЧКИ алгоритми



Од дрво до множество правила

Одлучувачко дрво може да се конвертира во множество правила



Секој пат од коренот до некој лист е ПРАВИЛО.

Own_house = true → Class =Yes [sup=6/15, conf=6/6]
Own_house = false, Has_job = true → Class = Yes [sup=5/15, conf=5/5]
Own_house = false, Has_job = false → Class = No [sup=4/15, conf=4/4]

Алгоритам за дрва за одлучување

- Основен алгоритам (**divide-and-conquer** алгоритам)
  - Атрибутите нека се категорични (и реалните вредности може да се преуредат)
  - Дрвото се конструира со top-down со помош на рекурзија.
  - На почетокот, сите обучувачки примери се кај коренот.
  - Примерите се разделуваат рекурзивно врз основа на селектирани атрибути.
  - Атрибутите се избираат на база на соодветна функција (информациска добивка - information gain)
- Услови за запирање на разделувањето :
  - Сите примери за дадено теме припаѓаат на истата класа
  - Нема повеќе атрибути за натамошно разделување
  - Нема повеќе примери во табелата

Decision tree learning algorithm

. **Algorithm** decisionTree(D, A, T)

```
1   if D contains only training examples of the same class c_j ∈ C then
2       make T a leaf node labeled with class c_j;
3   elseif A = ∅ then
4       make T a leaf node labeled with c_j, which is the most frequent class in D
5   else  // D contains examples belonging to a mixture of classes. We select a single
6         // attribute to partition D into subsets so that each subset is purer
7       p_0 = impurityEval-1(D);
8       for each attribute A_i ∈ {A_1, A_2, ..., A_k} do
9           p_i = impurityEval-2(A_i, D)
10      end
11      Select A_g ∈ {A_1, A_2, ..., A_k} that gives the biggest impurity reduction,
            computed using p_0 − p_i;
12      if p_0 − p_g < threshold then    // A_g does not significantly reduce impurity p_0
13        make T a leaf node labeled with c_j, the most frequent class in D.
14      else                             // A_g is able to reduce impurity p_0
15          Make T a decision node on A_g;
16          Let the possible values of A_g be v_1, v_2, ..., v_m. Partition D into m
                disjoint subsets D_1, D_2, ..., D_m based on the m values of A_g.
17          for each D_j in {D_1, D_2, ..., D_m} do
18              if D_j ≠ ∅ then
19                create a branch (edge) node T_j for v_j as a child node of T;
20                decisionTree(D_j, A-{A_g}, T_j)// A_g is removed
21              end
22          end
23      end
24  end
```


Како да се избере атрибут за разделување на податоците

- Главната работа при градење на дрво за одлучување е да се избере вистинскиот атрибут при разгранувањето
- Целта е да се намали impurity или несигурноста кај податоците во најголема можна мерка.
  - Подмножество податоци е чисто (pure) ако сите инстанци припаѓаат на иста класа.
- Хевристиката во C4.5 е да се избере атрибут со макимум Информациска добивка (Information Gain) или однос на добивка (Gain Ratio) базирно на Теорија на информации.

Податоци за кредит

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	**No**
2	young	false	false	good	**No**
3	young	true	false	good	**Yes**
4	young	true	true	fair	**Yes**
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	middle	true	true	good	**Yes**
9	middle	false	true	excellent	**Yes**
10	middle	false	true	excellent	**Yes**
11	old	false	true	excellent	**Yes**
12	old	false	true	good	**Yes**
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**

Два можни корена, кое е подобро?

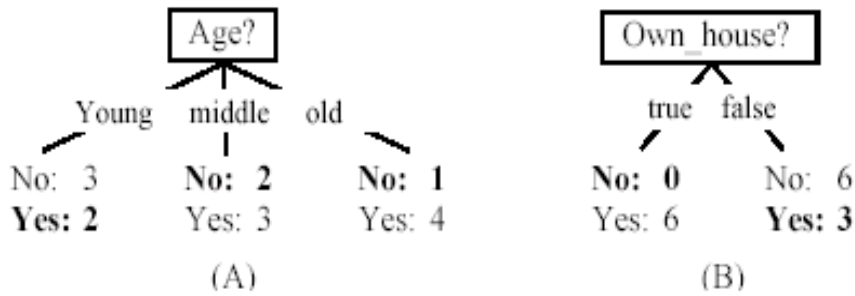


Fig. (B) Личи дека е подобро.

Теорија на информации – мерка на Ентропија

- Формулата за ентропија,

$$entropy(D) = -\sum_{j=1}^{|C|} \Pr(c_j) \log_2 \Pr(c_j)$$

$$\sum_{j=1}^{|C|} \Pr(c_j) = 1,$$

- $\Pr(c_j)$ е веројатноста дека класата c_j е во податочното множество D
- Тука ентропијата се користи како мерка за нечистотија или неред (impurity или disorder) на податочното множество D. (Или мерка на информација во дрво)

Мерка на Ентропија – пример

1. The data set D has 50% positive examples ($\Pr(positive) = 0.5$) and 50% negative examples ($\Pr(negative) = 0.5$).

$$entropy(D) = -0.5 \times \log_2 0.5 - 0.5 \times \log_2 0.5 = 1$$

2. The data set D has 20% positive examples ($\Pr(positive) = 0.2$) and 80% negative examples ($\Pr(negative) = 0.8$).

$$entropy(D) = -0.2 \times \log_2 0.2 - 0.8 \times \log_2 0.8 = 0.722$$

3. The data set D has 100% positive examples ($\Pr(positive) = 1$) and no negative examples, ($\Pr(negative) = 0$).

$$entropy(D) = -1 \times \log_2 1 - 0 \times \log_2 0 = 0$$

Како што податоците стануваат почисти и почисти (purer and purer), вредностите на Ентропијата опаѓаат. Ова е корисно за нас!

Информациска добивка

- За дадено множество *D*, прво се пресметува неговата Ентропија:

$$entropy(D) = -\sum_{j=1}^{|C|} \Pr(c_j) \log_2 \Pr(c_j)$$

- Ако имаме атрибут A_i, со v вредности, кај коренот на тековното дрво, множеството *D* ќе го раздели на v подмножества D_1, D_2 …, D_v . Очекуваната ентропија ако се користи A_i како тековен корен е :

$$entropy_{A_i}(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times entropy(D_j)$$

Информациска добивка (…)

- Добиената информација со избирање на атрибутот A_i за разгранување (или разделување на податоците)

$$gain(D, A_i) = entropy(D) - entropy_{A_i}(D)$$

- Го бираме атрибутот со најголемата добивка при градењето на секоја нова гранка на дрвото.

Пример

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	excellent	No
3	young	true	false	good	Yes
4	young	true	true	good	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

$$entropy(D) = -\frac{6}{15}\times\log_2\frac{6}{15} - \frac{9}{15}\times\log_2\frac{9}{15} = 0.971$$

$$entropy_{Own\_house}(D) = -\frac{6}{15}\times entropy(D_1) - \frac{9}{15}\times entropy(D_2)$$
$$= \frac{6}{15}\times 0 + \frac{9}{15}\times 0.918$$
$$= 0.551$$

$$entropy_{Age}(D) = -\frac{5}{15}\times entropy(D_1) - \frac{5}{15}\times entropy(D_2) - \frac{5}{15}\times entropy(D_3)$$
$$= \frac{5}{15}\times 0.971 + \frac{5}{15}\times 0.971 + \frac{5}{15}\times 0.722$$
$$= 0.888$$

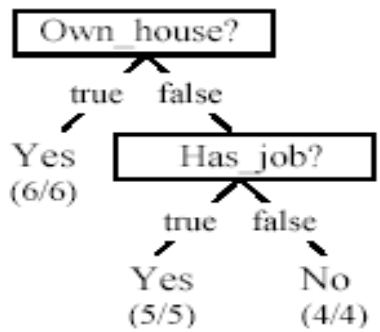
Age	Yes	No	entropy(Di)
young	2	3	0.971
middle	3	2	0.971
old	4	1	0.722

Own_house е најдобар избор за корен.

$gain(D, \text{Age}) = 0.971 - 0.888 = 0.083$
$gain(D, \text{Own\_house}) = 0.971 - 0.551 = 0.420$
$gain(D, \text{Has\_Job}) = 0.971 - 0.647 = 0.324$
$gain(D, \text{Credit\_Rating}) = 0.971 - 0.608 = 0.363$

25

Го градиме конечното дрво

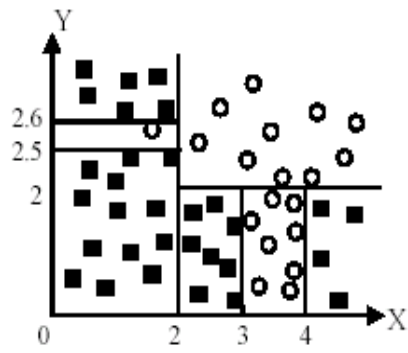


Информациската добивка може да се искористи за пресметување на impurity исто така

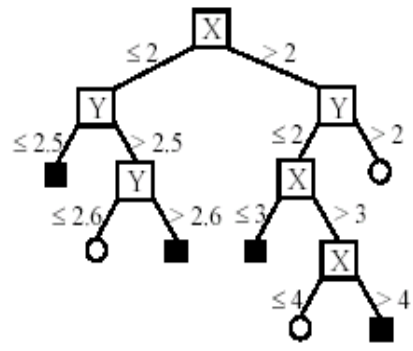
Работа со непрекинати атрибути

- Непрекинатите вредности на атрибутите се делат на два интервала (може и на повеќе) кај секое теме.
- Како да се определи најдобрата граница за разделување ?
  - Повторно да се искористи информацискатa добивка или gain ratio
  - Сортирај ги сите вредности на непрекинатите атрибути во растечки редослед $\{v_1, v_2, \ldots, v_r\}$,
  - Една можна поделба е помеѓу две соседни вредности v_i и v_{j+1}. Треба да се пробаат сите соседни поделби и да се пронајде онаа која ја максимизира добивката (или gain ratio).

An example in a continuous space



(A) A partition of the data space



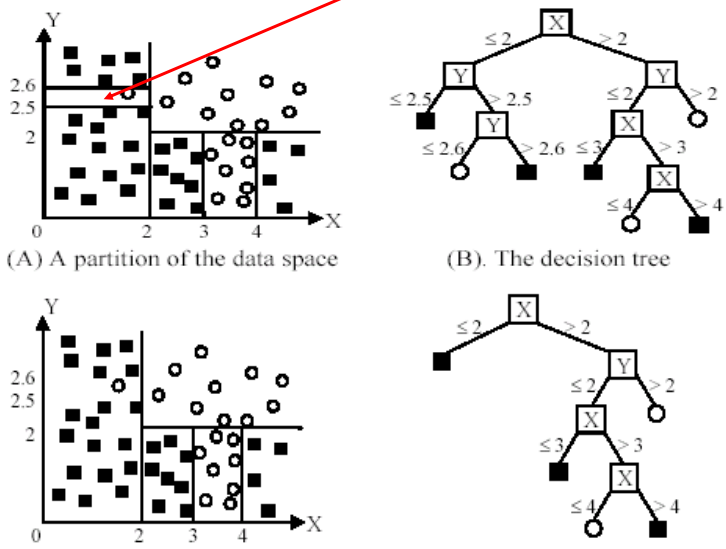
(B). The decision tree

Избегнување на overfitting во класификацијата

- Overfitting: Со дрвата на одлучување може да се дојде до пренаучување на обучувачките податоци
  - Добра точност на обучувачките податоци, но слаба на тест-податоците.
  - Симптоми: дрво со преголема длабочина и многу гранки
    - Некои може да се последица на аномалии поради шум и аутлаери.
- Две решенија за избегнување overfitting
  - Pre-pruning: да се запре со градење на дрвото доволно рано
    - Тешко е да се одлучи бидејќи не знаеме што ќе се случи ако продолжиме со растење на дрвото.
  - Post-pruning: Отстранување на гранки и поддрва од целосното изградено дрво.
    - Вообичаено се користи.
    - C4.5 користи статистички методи за проценка на грешките при отсекување на секое теме.

Пример

Може да се случи пренаучување (overfit) на податоците



(A) A partition of the data space

(B). The decision tree

Резултати

C4.5rules:

			PREDICTED CLASS			
		Amphibians	Fishes	Reptiles	Birds	Mammals
ACTUAL CLASS	Amphibians	2	0	0	0	0
	Fishes	0	2	0	0	1
	Reptiles	1	0	3	0	0
	Birds	1	0	0	3	0
	Mammals	0	0	1	0	6

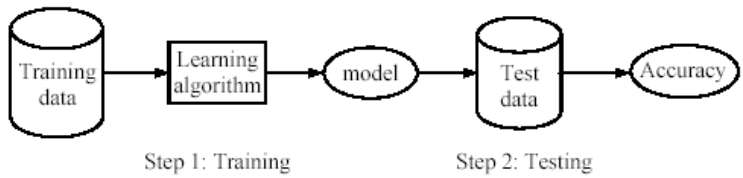
RIPPER:

			PREDICTED CLASS			
		Amphibians	Fishes	Reptiles	Birds	Mammals
ACTUAL CLASS	Amphibians	0	0	0	0	2
	Fishes	0	3	0	0	0
	Reptiles	0	0	3	0	1
	Birds	0	0	1	2	1
	Mammals	0	2	1	0	4

- Евалуација на системите за класификација

Надгледувано учење: два чекора

33



$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}},$$

Основните претпоставки на учењето

Претпоставка: Распределбата на обучувачките податоци е идентична на тест примерите (вклучувајќи ги невидените примероци).

- Во праксата, оваа претпоставка често не е исполнета (до некој степен)
- Ако распределбата многу се разликува, се добиваат лоша точност при класификација
- За да се постигне добар резултат на тест податоците
  - ...обучувачките примери мора да се добар репрезент на тест-податоците.

Евалуација на класификациските методи

- **Точност (Predictive accuracy)**

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}}$$

- Ефикасност (Efficiency)
  - Време да се конструира моделот time to construct the model
  - Време потребно да се користи моделот time to use the model
- Робустност (Robustness) : справување со шум и вредности што недостасуваат
- Скалабилност (Scalability): ефикасност во работа со бази податоци
- Интерпретабилност (Interpretability):
  - Да се разберат резултатите добиени со моделот
- Компактност на модел: големина на дрво за одлучување, број на правила

Евалуациски методи

- **1. Задржано множество (Holdout set)**: Множеството податоци се дели на две дисјунктни подмножества
  - Обучувачко D_{train} (for learning a model)
  - Тест-множество D_{test} (for testing the model)
- **ВАЖНО:** обучувачкото множество не може да се користи за тестирање, ниту тестирачкото множество за обучување
- Тест податоци што не биле видени за време на обучувањето, обезбедуваат независна оценка на точноста.
- Тест множеството уште се нарекува и задржано множество (holdout set). (the examples in the original data set *D* are all labeled with classes.)
- Оваа метода за евалуација се користи кога **податочното множество *D* е големо**.

Евалуациски методи_2

- **2. n-делна вкрстена валидација (n-fold cross-validation)**: Множеството на податоци се дели на *n* еднакви по големина дисјунктни подмножества.
- Да се искористи секое подмножество како тест множество, а останатите *n*-1 подмножества – како обучувачко множество за учење на класификаторот.
- Процедурата се одвива *n* пати, со што добиваме *n* пресметани точности (accuracy).
- Конечната точност е аритметичката средина на *n*-те пресметани точности.
- Вообичаено се користат10-делна и 5-делна вкрстена валидација.
- Оваа метода се користи кога **податочното множество *D* НЕ Е големо**.

Евалуациски методи_3

- **3. Leave-one-out вкрстена валидација (cross-validation)** : Оваа метода се користи кога податочното множесто **е многу мало.**
- Ова е специјален случај на вкрстена валидација
- Секој дел (fold) на вкрстената валидација има само еден тест пример а сите останат примери се искористени за обучување.
- Ако податочното множество содржи *m* примери, тогаш тоа е *m*-делна крос валидација (*m*-fold cross-validation)

Евалуациски методи_3

- **Валидациско множество (Validation set)**: достапните податоци се делат на 3 подмножества,
  - Обучувачко множество,
  - Валидацисако множество
  - Тест-множество.
- Валидациското множество често се користи за проценка на параметри кај алгоритмите за учење.
- Во овие случаи, вреднсотите на параметрите кои даваат најголема точност на валидациското множество, се користат како конечни параметри.
- Вкрсена валидација може да се користи и за добивање вредности и на параметрите.

Мерки за класификација – колку е добар системот?

- Точноста (acuracy) е само една мерка
  - (error = 1-accuracy).
- **Точноста не е применлива за некои системи за класификација**.
- На пр. во “рударење на текстови” (text mining), може да не интересира само одредена тема, а тоа да е мал дел од голема колекција текстови.
- Во класификациите каде што има небалансирани податоци (број на податоци по класа), може да не’ интересира само “малата класа”
- Висока точност не значи дека се пронајдени елементите од малата класа
  - Пр. 1% intrusion. Achieve 99% accuracy by doing nothing.
- Класата што не’ интересира се нарекува **позитивна класа**, а останатите класи се **негативни класи.**

2. **Precision** и **recall** мерки

- Користиме матрица на конфузија (confusion matrix) за нивно воведување.

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

where

TP: the number of correct classifications of the positive examples (**true positive**),
FN: the number of incorrect classifications of positive examples (**false negative**),
FP: the number of incorrect classifications of negative examples (**false positive**), and
TN: the number of correct classifications of negative examples (**true negative**).

Precision и **recall** мерки

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

$$p = \frac{TP}{TP + FP}. \qquad r = \frac{TP}{TP + FN}.$$

Precision *p* е бројот на точно класификувани позитивни примери поделени со вкупниот број на примери кои биле класификувани како позитивни.

Recall *r* е бројот на точно класификувани позитивни примери поделен со вкупниот број на позитивните примери од тест-множеството.

Пр. Програма која бара ѕвезди во некоја слика. Таа препознава 7 ѕвезди во сцена што има 9 ѕвезди и неколку точки. Ако 4 од идентификуваните ѕвезди се точно препознаени, а 3 се всушност точки (4 TP и 3 FP), precision е 4/7, а recall е 4/9.

Пример

	Classified Positive	Classified Negative
Actual Positive	1	99
Actual Negative	0	1000

- Оваа матрица на конфузија ги дава информациите:
  - precision p = 100% и
  - recall r = 1%

  бидејќи сме класификувале само еден позитивен пример, а негативните - сите добро класификувани.
- Забелешка: precision и recall ја мерат класификацијата САМО на позитивната класа
- Precision може да се гледа како мерка на прецизност или квалитет, а recall is a мерка на комплетност или квантитет.
- Уште поедноставно – висок **recall** значи дека системот точно ги класификувал најголемиот број релевантни резултати. (means that an algorithm returned most of the relevant results.) Висок **precision** значи дека системот точно класификувал повеќе релевантни, отколку нерелевантни податоци (means that an algorithm returned more relevant results than irrelevant.)

Пр. Проценка на алгоритам за обработка на слика

- Пример: Конструирана е нова техника за пронаоѓање рабови на слика
- Постигната е точност од 95% на некое тест множество
- Дали новата техника е успешна? Што ни кажува “95% точност?

Типови грешки

- Може да се случи следното:
  - 95% се правилно класификувани пиксели
  - Само 5% од пикселите се рабови
  - Ги промашува сите рабни пиксели!

- Како да ги издвоиме релевантните резултати, т.е. да имаме начин да ја процениме релевантноста на позитивните и на негативните класификации?

Повторно со матрица на конфузија

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

		Предикција	
		Раб	**Не е раб**
Вистинска ситуација	**раб**	True Positive	False Negative
	Не е раб	False Positive	True Negative

Sensitivity и Specificity

Sensitivity = $\frac{TP}{TP+FN}$

Може да се гледа како мерка за добивање на позитивен случај кога ќе се тестира со таков пример.

Или… пропорцијата на рабовите што ги наоѓаме

$$p = \frac{TP}{TP + FP}. \qquad r = \frac{TP}{TP + FN}.$$

Specificity = $\frac{TN}{TN+FP}$

Може да се гледа како мерка за добивање на негативен случај кога ќе се тестира со таков пример.

Или…пропорцијата на не-рабовите што ги наоѓаме

Sensitivity = $\frac{TP}{TP+FN}$ = ? Specificity = $\frac{TN}{TN+FP}$ = ?

		Предикција	
		1	**0**
Вистинска ситуација	**1**	60	30
	0	80	20

60+30 = 90 случаи од податоците се класа 1 (раб)

80+20 = 100 случаи од податоците се класа 0 (не-раб)

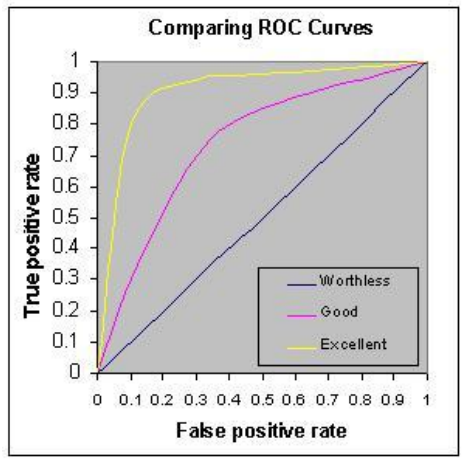
90+100 = 190 примери (pixels) во податочното множество

Сензитивност и специфичност (Sensitivity and specificity)

- **Сензитивност и специфичност се** статистички мерки на бинарен класификациски тест
- **Сензитивноста** (се совпаѓа со recall rate) ја мери пропорцијата на точно препознаени позитивни примери
  - (пр. Процент на болни луѓе кај кои точно е дијагностицирана болеста).
- **Специфичноста** ја мери пропорцијата на точно препознаени негативни примери
  - (пр. Процент на здрави луѓе кои се точно идентификувани дека не се болни).
- Овие мерки се блиску до концептот на грешки Тип1 и Тип2 (од статистика)
- Најдобар предвидувач е:
  - 100% сензитивност (сите болни луѓе се точно дијагностицирани како болни)

  И истовремено
  - 100% специфичност (сите здрави (не-болни) луѓе се препознаени како здрави

Receiver Operating Characteristic Методологија – ROC крива



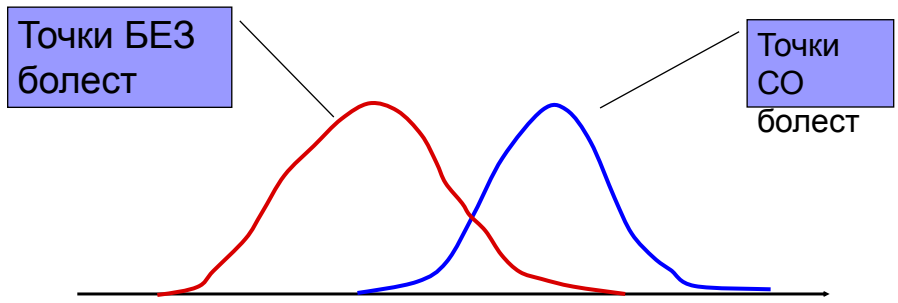
Што се тоа ROC криви?

- *ROC = Receiver Operating Characteristic*
- Започнале да се користат кај теоријата електронските сигнали (1940s - 1950s)
- Станаа многу популарни во биомедицински апликации, посебно во радиологијата и обработка на медицински слики
- Се користат и во МУ за оценка на класификатори

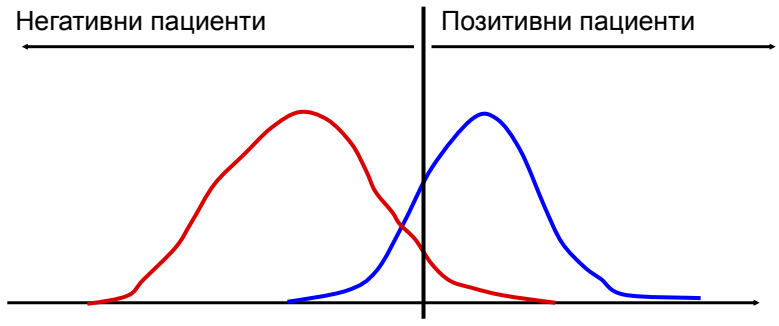
ROC криви: наједноставен пример

- Нека имаме дијагностички тест за откривање на некоја болест
- Тестот има 2 можни излези :
  - ‘позитивен’ = пациентот е болен (ја “има” болеста)
  - ‘негативен’ = пациентот е здрав
- Пациент се тестира дали е позитивен или негативен во врска со болеста

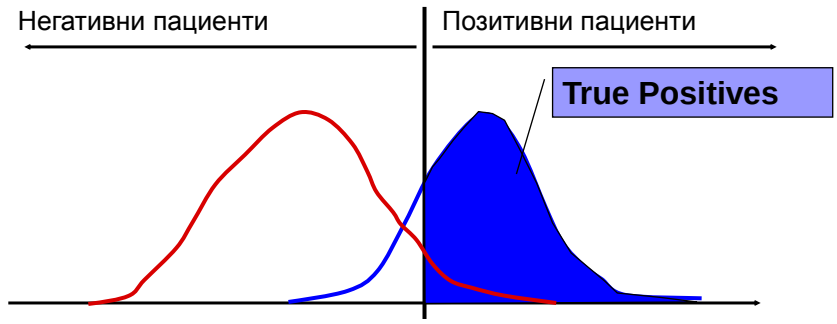
Пример - дијагноза



Threshold



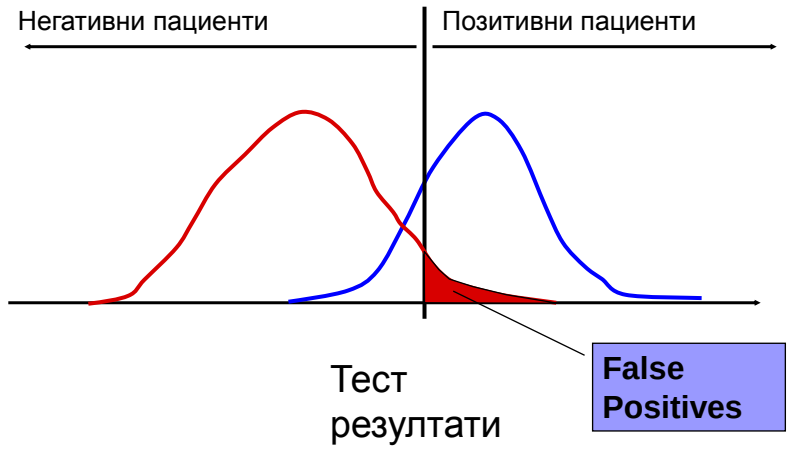
Неколку дефиниции...



Тест
резултати

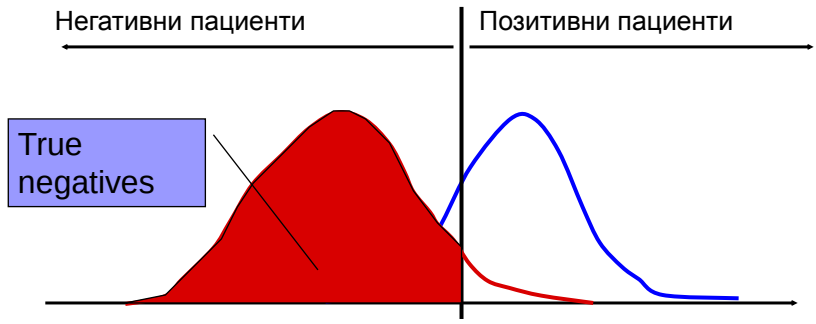
БЕЗ болест
Со болест

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ
И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО



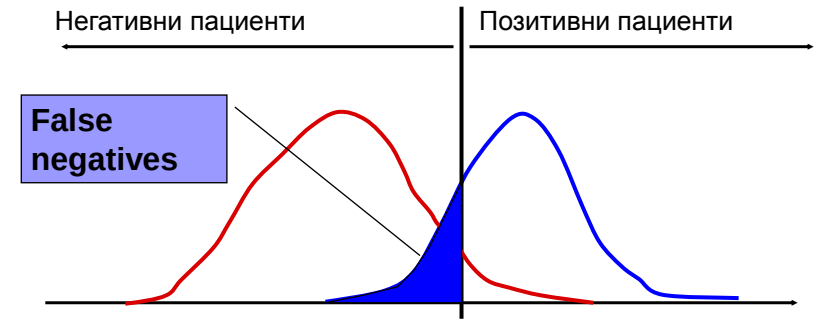
Тест
резултати

БЕЗ болест
Со болест



Тест резултати

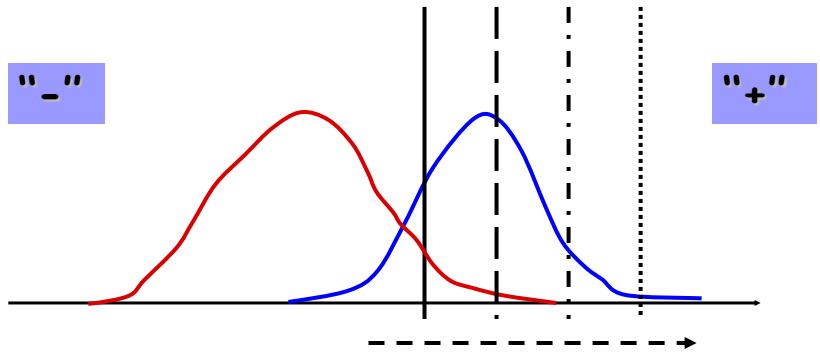
БЕЗ болест
Со болест



Тест резултати

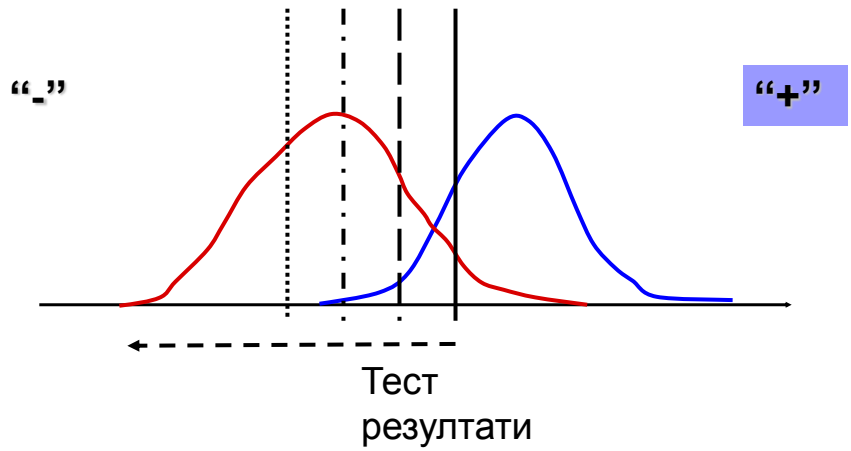
БЕЗ болест
Со болест

Moving the Threshold: right



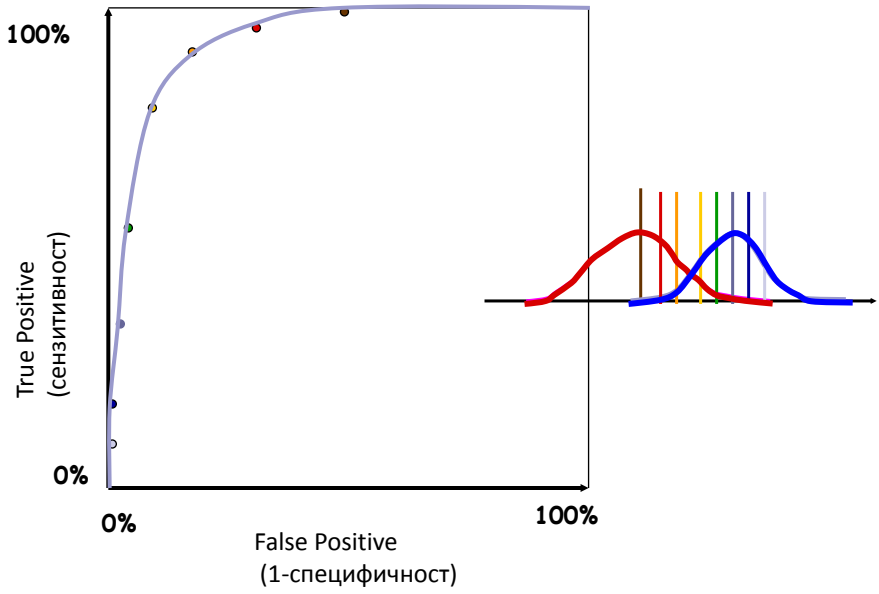
БЕЗ болест
Со болест

Движење на границата: во лево



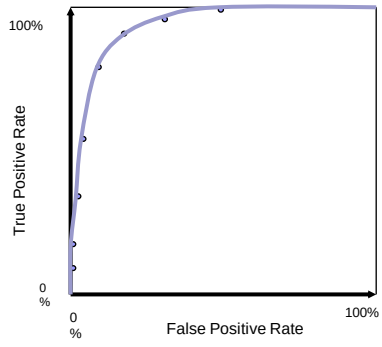
БЕЗ болест
Со болест

ROC крива

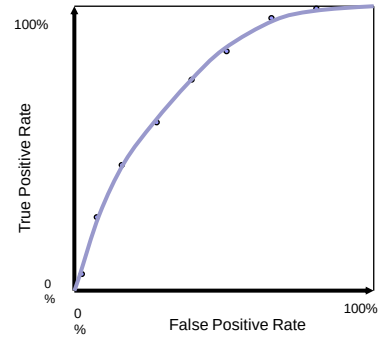


Споредба на ROC криви

Добар тест

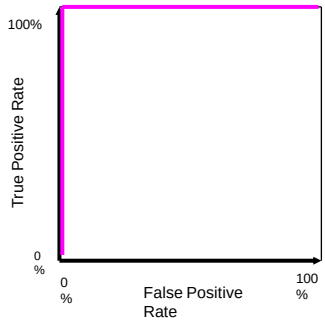


Слаб тест



Екстремни вредности на ROC крива

Најдобар тест:



Распределбите воопшто не се поклопуваат

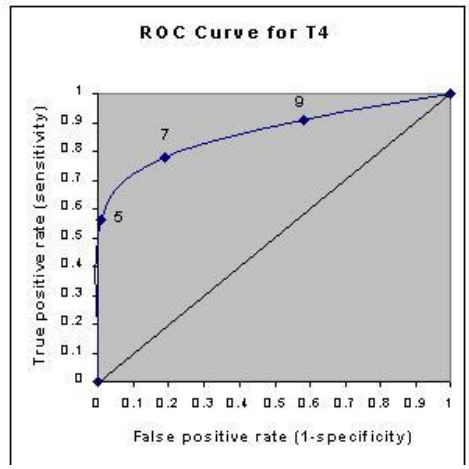
Најлош тест:

Распределбите комплетно се поклопуваат

Уште еден пример:

Гранични вредности	True Positives	False Positives
5	0.56	0.01
7	0.78	0.19
9	0.91	0.58

Гранични вредности	Сензитивност	Специфичност
5	0.56	0.99
7	0.78	0.81
9	0.91	0.42



Употреба на ROC крива со непрекинати податоци

- Многу биохемиски мерења се непрекинати, пр. Гликоза во крвта наспроти дијабетес.
- ROC анализата може да се употреби и кај непрекинати податоци

Примери за користење на ROC анализа

- Избирање на граница (threshold) при нагодување на веќе обучени класификатори (пр. Невронски мрежи, CBM)
- Дефинирање на граници кај DNA микрополиња
- Споредување на статистички тестови за идентификација на експресирани гени
- Проценка на перформанси на различни алгоритми за предвидување протеини

Разни ROC криви за различни методи