Откриване на информация

Класификация и прогнозиране 1

Класификация и прогнозиране

- Задача на класификацията
  - определяне на класове и принадлежността на обект към някой от класовете
  - номинални данни
- Задача на прогнозирането
  - определяне на числова стоност
  - непрекъснати данни
- Принципи
  - всеки обект има атрибути
  - един от атрибутите е ключов
  - ключовият атрибут определя класа на принадлежност или очакваната стойност
- Модел
  - търси се модел, чрез който се намира стойността на ключовия атрибут, като функция на неключовите

Процес

- Обучение
  - търсене на класификатор
  - построяване на модел
- Валидиране
  - проверка на модела и потвърждаване на действието на класификатора
- Имплементиране
  - прилагане на модела за анализ на нови данни

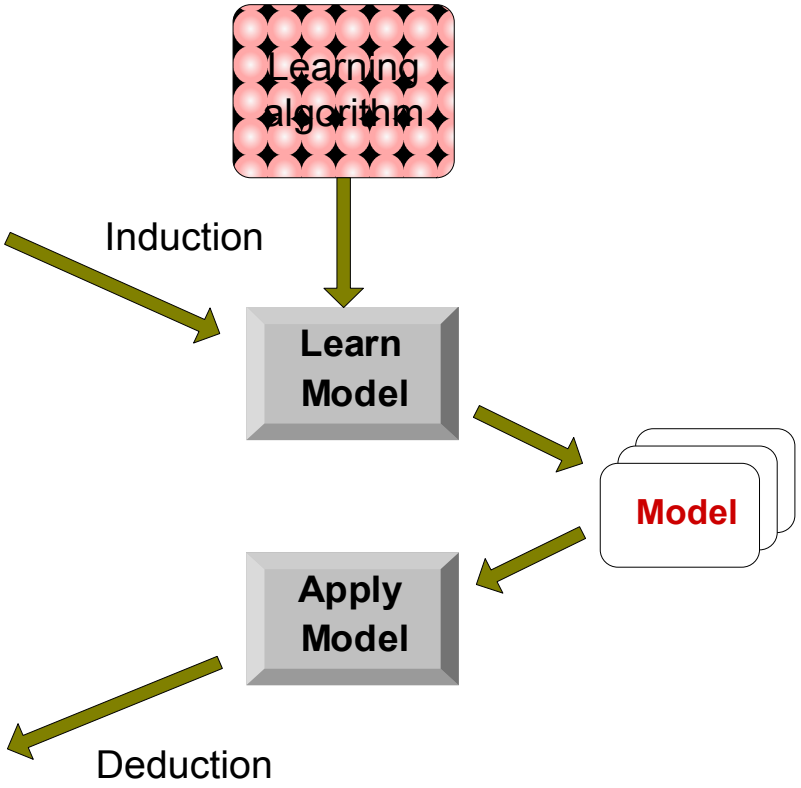
Процес

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Обучение

- Цел
  - откриване на фуннкцията на класифициране
- Две групи методи на обучение
  - supervised learning
    - ключовият атрибут е известен
    - предварително се знае кой обект на кой клас принадлежи
  - clustering
    - ключовият атрибут и броят на класовете са неизвестни
- Точност на класификатора
  - % от случаите, в които класификаторът е показал очкваните резултато

Данни за анализ

- Обучаваща извадка (*training set*)
  - използва се за построяване на модела
- Тестова извадка (*test set*)
  - използва се за валидиране на модела
- Реални данни и прогнози
  - прилагане на модела

Приложения

- В търговията и услугите
  - оценка на потенциални клиенти
- В медицината
  - предвиждане на заболявания
- В спорта
  - предвиждане и подготовка на постижения

и др.

Подходи за класификация

- Класовете са известни и обектите се разпределят в тях
- Класовете се формират по разделящи признаци на обектите

Методи за класификация

- Основани на дърво на решенията
- Вероятностни методи
- Основани на правила
- Невронни мрежи
- и др.

Дърво на решенията

Изграждане на йерархия от разделящи признаци

Дърво на решенията

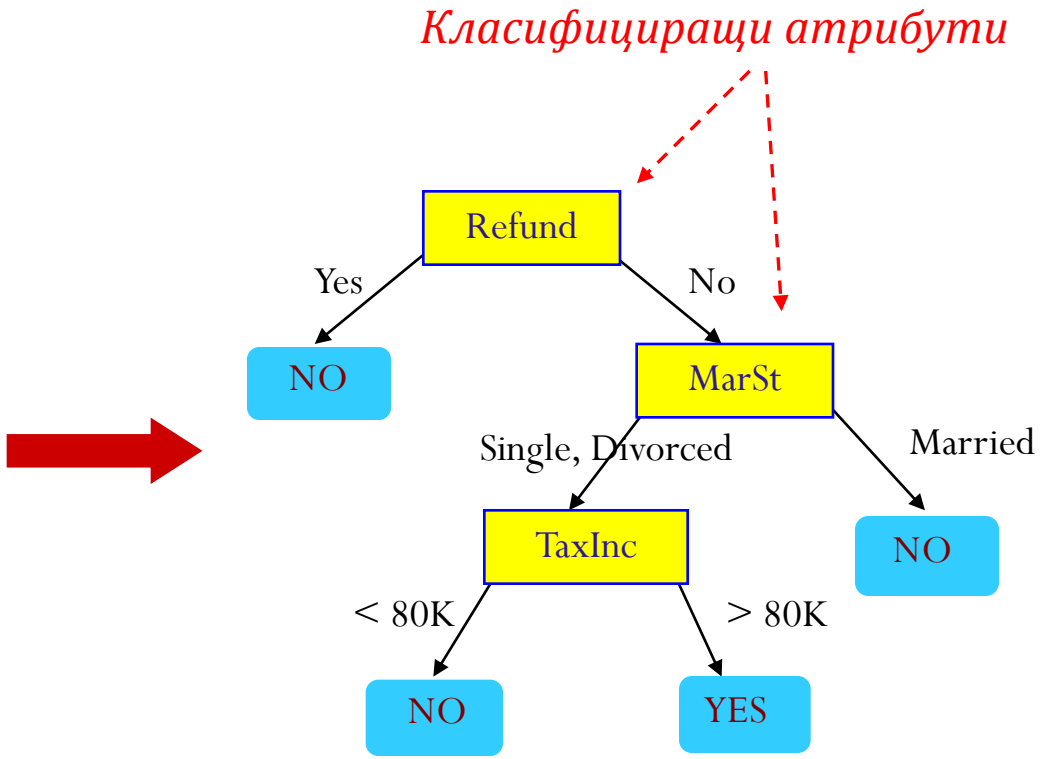
- Decision tree
- Дървовидна структура
  - всеки възел на дървото съдържа проверка на условие за атрибут
  - листата са стойности на клас (етикети на клас)
- Двоично дърво
- Предимства на метода
  - интуитивни
  - прости
  - точни алгоритми
  - използват се като основа на други алгоритми

Дърво на решенията

категории категории числови клас

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

Обучаваща извадка

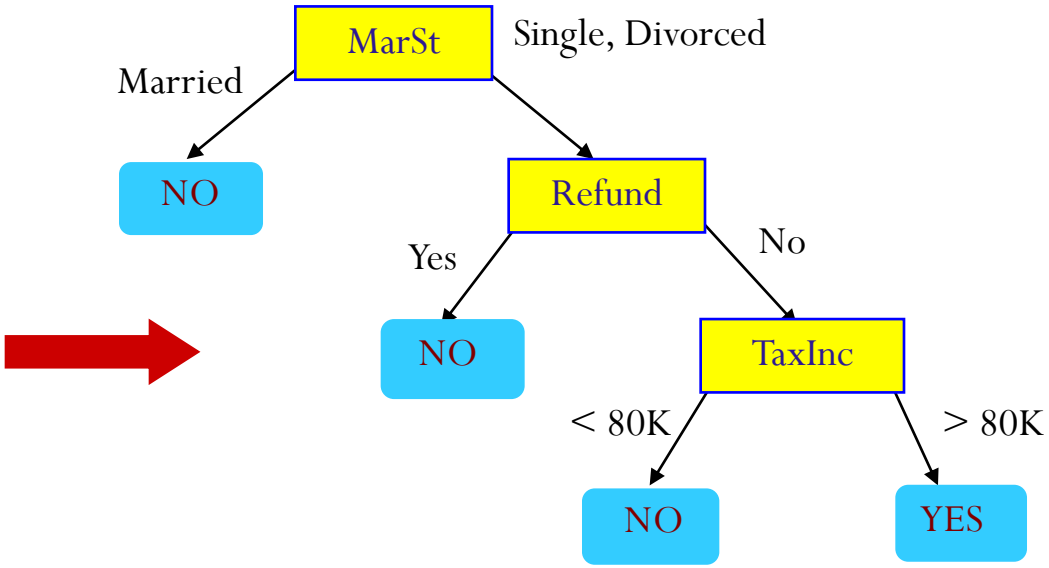


Модел

Дърво на решенията

категории категории числови клас

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



Обучаваща извадка

Алтернативен модел

Алгоритми

- Вход
  - обучаваща извадка от вече класифицирани записи
  - списък на атрибути – кандидати за класификационен атрибут
  - метод за избор на разделящи атрибути
- Изход
  - дърво на решенията
- Видове алгоритми
  - Hunt's Algorithm
  - CART – Classification and Regression Trees
  - ID3 - Iterative Dichotomizer
  - C4.5
  - SLIQ
  - SPRINT

Greedy Approach

- Основен подход за построяване на дървото
  - Top-down recursive divide-and-conquer
  - В началото се анализират всички обекти в извадката, причислени към известни класове; извадката се разделя на подмножества, анализирани по същия начин
- Алгоритъм
  - В началото всички обучаващи записи/примери са в корена
  - Всички атрибути са категорийни (ако не са, трябва да се преобразуват)
  - Избират се атрибути за класификатори
  - Примерите се разделят рекурсивно на по-малки подмножества, според стойностите на избраните атрибути
  - Условия за край на делението
    - всички примери принадлежат на един и същи клас
    - няма повече атрибути, които да се използват за деление
    - няма повече примери за деление

Избор на атрибути за разделяне

- Splitting attribute
- Избират се на базата на евристични или статистични оценки (напр. информативност information gain)
- Критерий: Pure partition
  - по възможност, всички данни, разгледани по този критерий да попадат в една и съща група
  - търси се възможно най-чистото деление

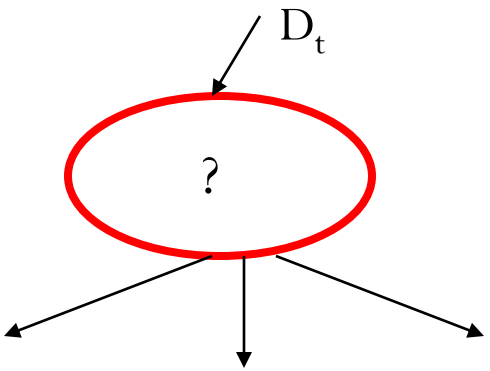
Splitting Attribute

- Всеки атрибут има множество стойности – класове, в които разделя обектите от извадката D
- Три възможности, според типа на атрибута A
  1. A има дискретни стойности
     - за всяка стойност на A се строи клон на дървото
     - всички елементи на D се разделят в подмножества
     - елементите в едно подмножество имат еднаква стойност на A
  2. A има непрекъснати числови стойности
     - множеството се дели на два класа: стойности по-големи или по-малки от разделящата (това може да е средната стойност)
     - необходимо е сортиране
  3. A има дискретни стойности
     - избира се подмножество на A - S_a
     - за всеки елемент се проверява дали принадлежи на S_a или не
     - дихотомно разделяне – в два класа

Hunt's Algorithm

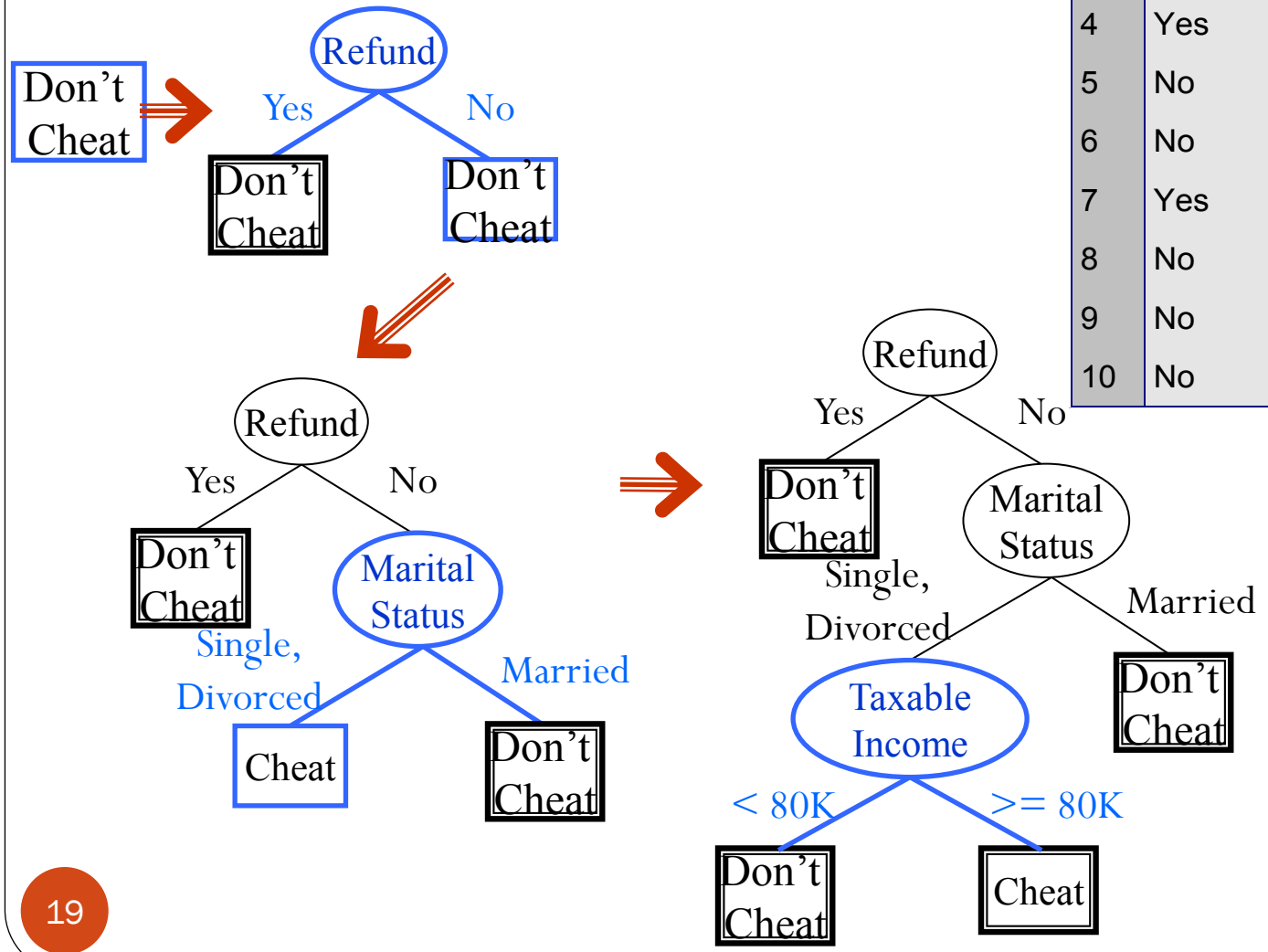
- D_t – множество от обучаващи записи от един клас t
- N – разделителен атрибут
- Процедура за разделяне
  - Ако всички записи на D_t принадлежат на един и същи клас y_t, по отношение на N, тогава е N класификационен атрибут
  - Ако D_t съдържа записи от повече класове, тогава се търсят атрибути за разделяне на множеството на подмножества и процедурата се прилага на подмножествата рекурсивно

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



Hunt's Algorithm

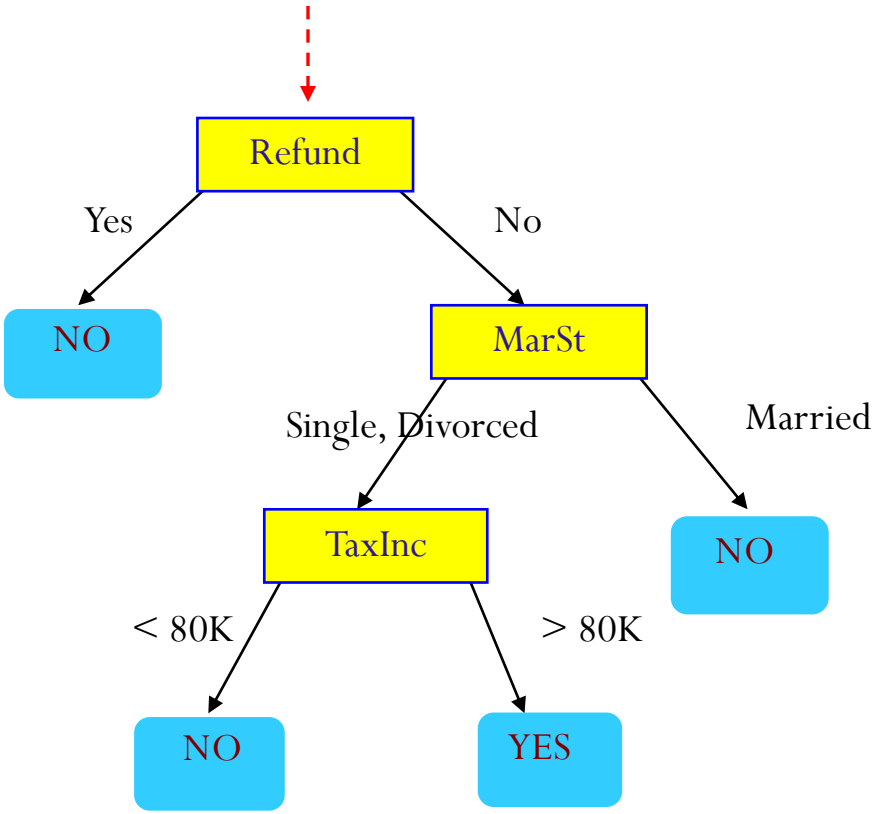
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Прилагане на модела с тестови данни

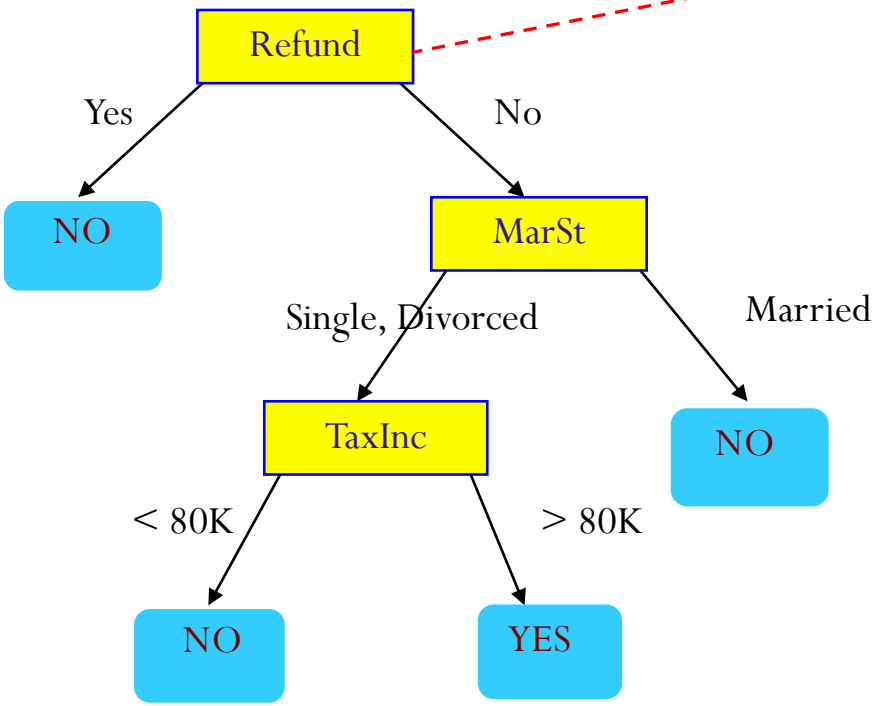
Тестови данни

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



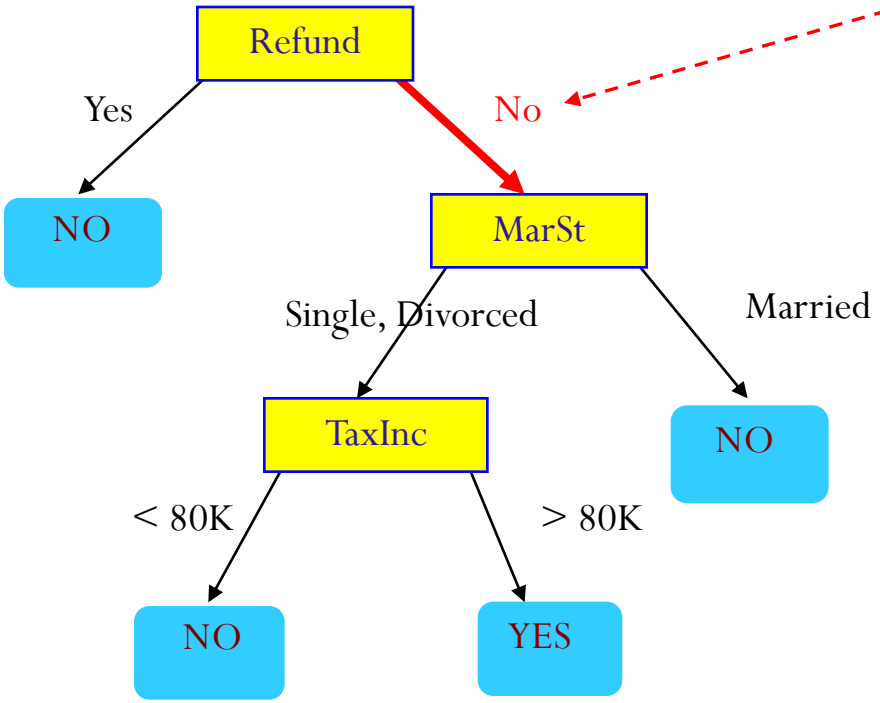
Прилагане на модела с тестови данни

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



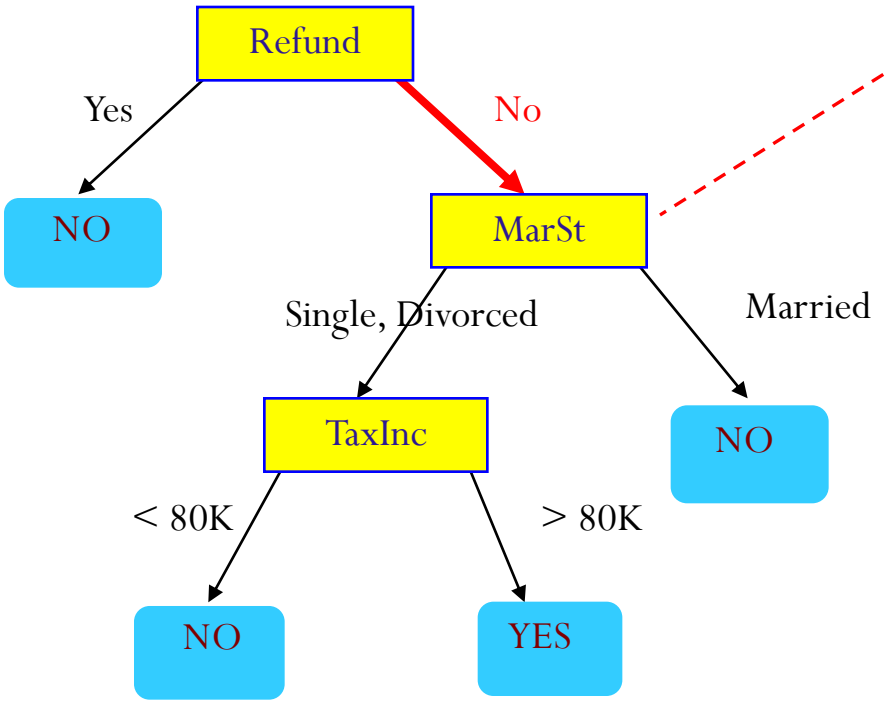
Прилагане на модела с тестови данни

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



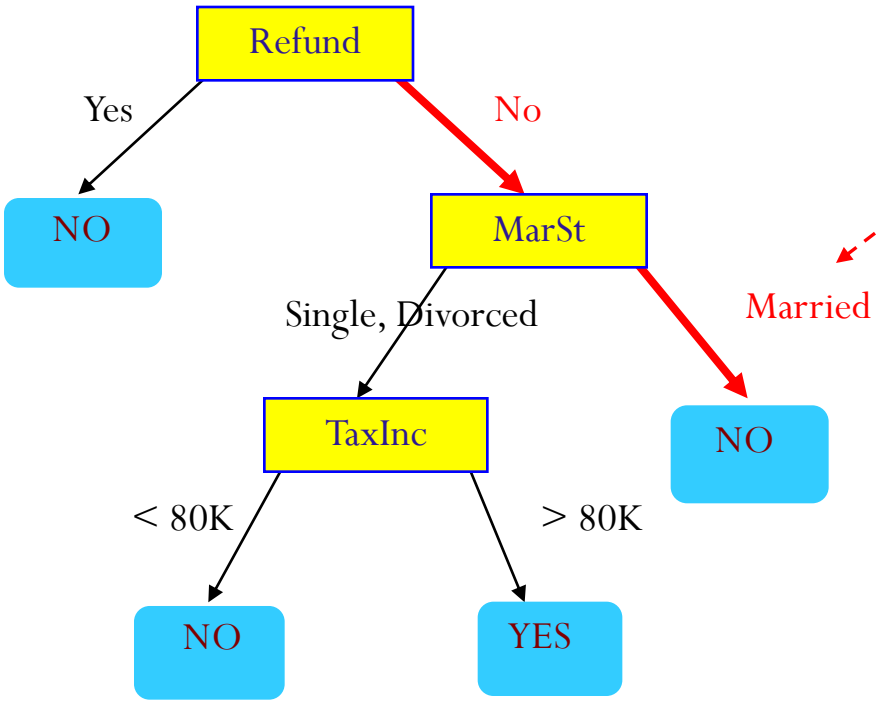
Прилагане на модела с тестови данни

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



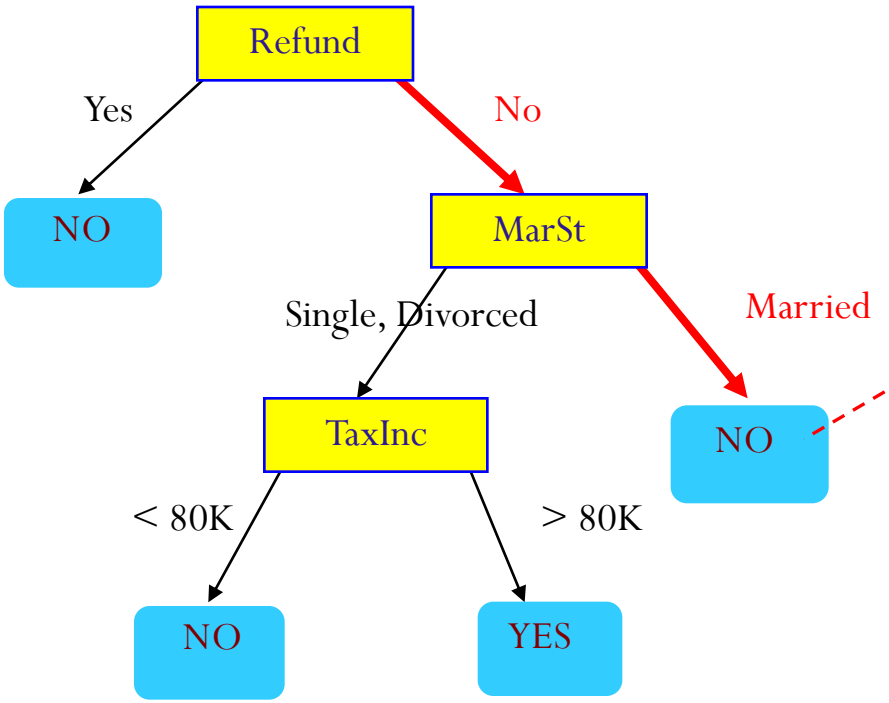
Прилагане на модела с тестови данни

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Прилагане на модела с тестови данни

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



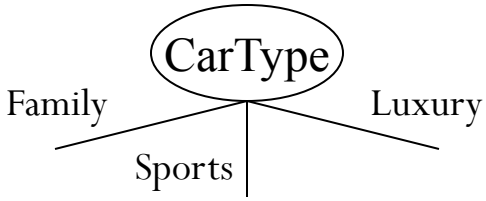
Присвояване на стойност “No”

Проблеми

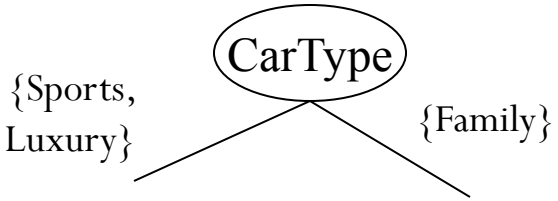
- Как да се раздели множеството?
  - коя е оптималната стратегия?
  - кои атрибути да се използват за деление?
    - според типа на атрибутите
- Двоично или многомерно деление?
- Кога да завърши делението?

Номинални атрибути

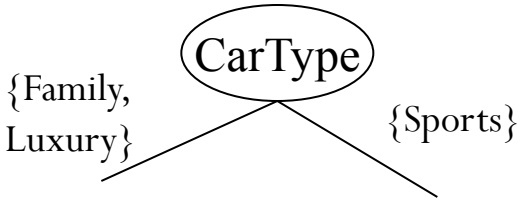
- Многомерно: Толкова подмножества, колкото номинални стойности на един атрибут



- Двоично: Последователно деление на две подмножества

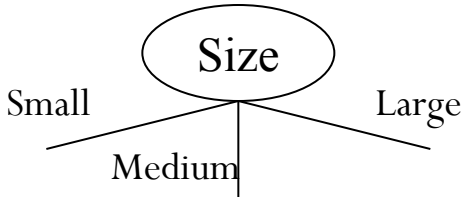


или

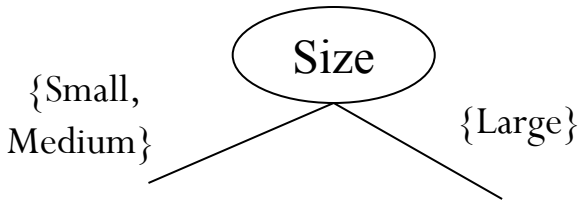


Ординални атрибути

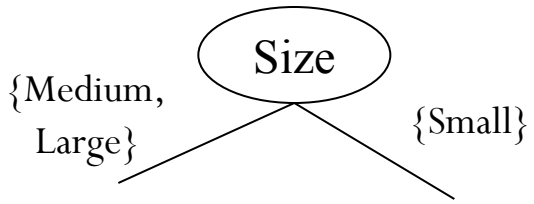
- Многомерно: Толкова подмножества, колкото различни стойности на един атрибут

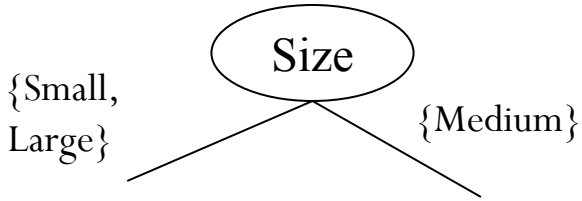


- Двоично: Последователно деление на две подмножества

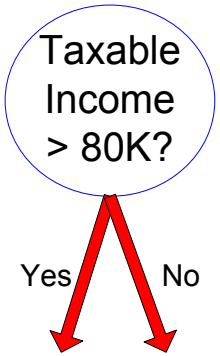


или

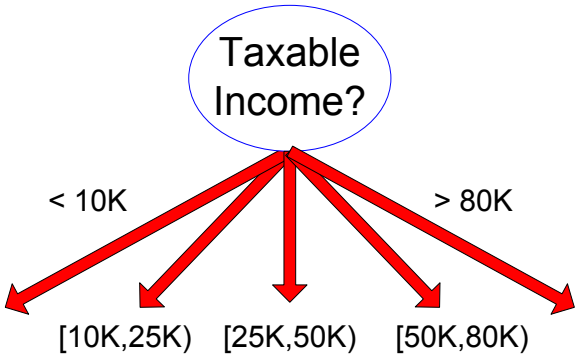




Непрекъснати атрибути



(i) Binary split

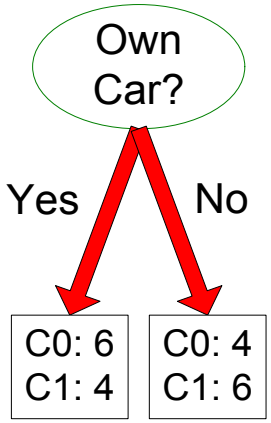


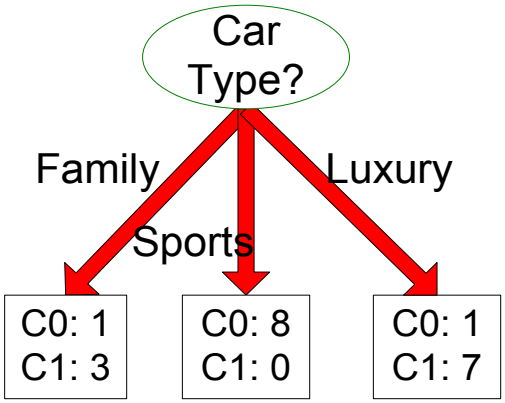
(ii) Multi-way split

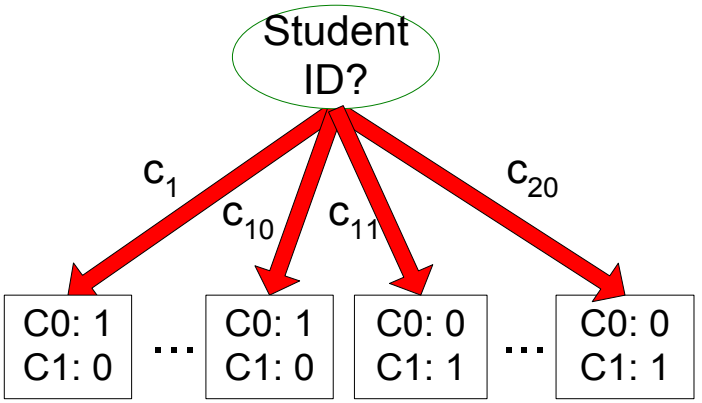
- Дискретизиране
  - на равни интервали, статично
  - на перцентили – динамично

- Двоично решение: $(A < v)$ или $(A \geq v)$

Коя е най-добрата стратегия?

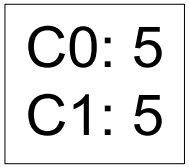






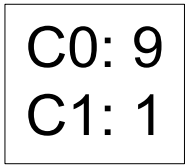
Коя е най-добрата стратегия?

- Принцип
  - хомогенно разпределение на подмножества



Non-homogeneous,

High degree of impurity

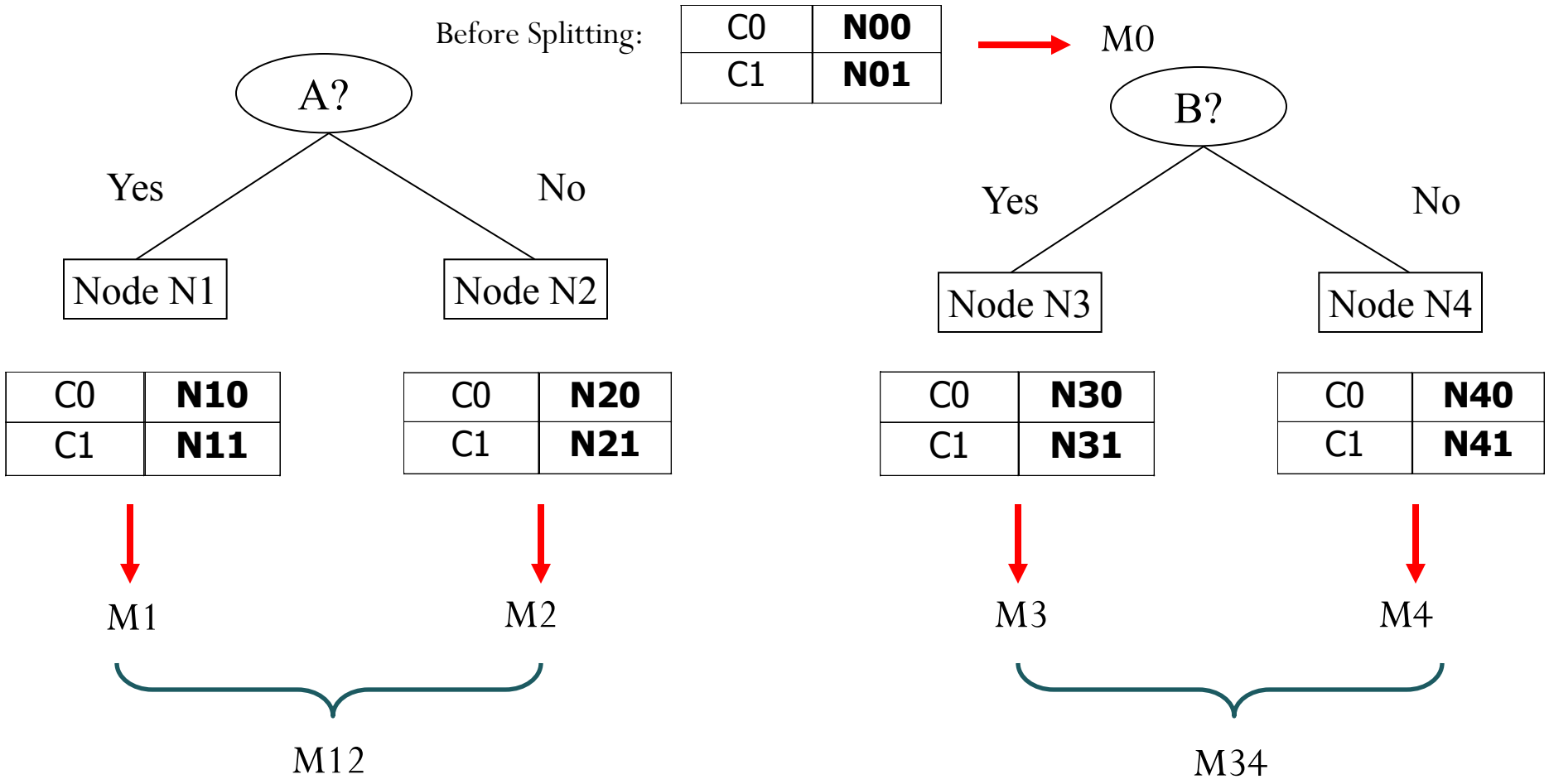


Homogeneous,

Low degree of impurity

- Мерки за чистота на избора на атрибут - оценка за информативност (information gain)
  - ентропия
  - индекс Gini
  - грешка от погрешна класификация

Коя е най-добрата стратегия?



Gain = M0 – M12 vs M0 – M34

Information Gain

- Избира се атрибутът с най-висока информативност - намалява се броят на последващите деления
- Ако p_i е вероятността подмножество от D да принадлежи на клас C_i, оценена като $|C_{i,D}|/|D|$, където $|C_{i,D}|$ е размера на $C_{i,D}$/, а |D| е размера на D, то
- очакваната информация (entropy), необходима за класифициране на D: $$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$
- информацията, необходима, все още и след разделянето на D на v части , спрямо стойностите на атрибут A: $$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$
- Придобитата информация от разделянето по A $$Gain(A) = Info(D) - Info_A(D)$$

Пример

- Class P: buys_computer = “yes”
- Class N: buys_computer = “no”

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2(\frac{9}{14}) - \frac{5}{14}\log_2(\frac{5}{14}) = 0.940$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
…40	medium	no	excellent	yes
…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

$\frac{5}{14}I(2,3)$ означава “age <=30” се среща 5 пъти от 14 примера: 2 yes и 3 no. Следователно

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Подобно,

$$Gain(income) = 0.029$$

$$Gain(student) = 0.151$$

$$Gain(credit\_rating) = 0.048$$

Други примери

C1	**0**
C2	**6**

$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$

$\text{Entropy} = -0 \log 0 - 1 \log 1 = -0 - 0 = 0$

C1	**1**
C2	**5**

$P(C1) = 1/6 \quad P(C2) = 5/6$

$\text{Entropy} = -(1/6) \log_2 (1/6) - (5/6) \log_2 (1/6) = 0.65$

C1	**2**
C2	**4**

$P(C1) = 2/6 \quad P(C2) = 4/6$

$\text{Entropy} = -(2/6) \log_2 (2/6) - (4/6) \log_2 (4/6) = 0.92$

Атрибути с непрекъснати стойности

- Оптимално дискретизиране - *best split point*
  - сортиране на A
  - средна точка – вероятна най-добра *split point*
    - $(a_i + a_{i+1})/2$
  - избира се точката с *minimum expected information requirement* за A
- Разделяне
  - D1: A ≤ split-point
  - D2: A > split-point

Gain Ratio (C4.5)

- Най-информативни атрибути са тези, които разделят на много подмножества
  - не винаги е разумно (напр. ЕГН)
- gain ratio – нормализация на информативността

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

  - GainRatio(A) = Gain(A)/SplitInfo(A)
- Ex.

$$SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$$

  - gain_ratio(income) = 0.029/1.557 = 0.019
- Избират се атрибутите с максимално gain_ratio

Gini Index

- Оценка на (не)чистотата на класифициране
- Ако D съдържа примери от n класа, gini index, $gini(D)$ се изчислява като

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

където p_j е относителната честота на срещане на клас j в D, вероятността един запис да попадне в клас Cj

- Ако D е разделена по A в две подмножества D_1 и D_2, $gini$ index $gini(D)$ е

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- Подобряване на информативността

$$\Delta gini(A) = gini(D) - gini_A(D)$$

- Избира се атрибутът с най-малко $gini_{split}(D)$
- (CART, IBM IntelligentMiner)

GINI Split

- Ако p се разделя на k части (children), качеството на разделяне се изчислява като

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

където, n_i = number of records at child i,

n = number of records at node p.

Пример

- D има 9 записа buys_computer = “yes” и 5 “no”

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- Ако income разделя D на 10 в D_1: {low, medium} и 4 в $D_{2:}$ (high)

$$gini_{income \in \{low,medium\}}(D) = \left(\frac{10}{14}\right) Gini(D_1) + \left(\frac{4}{14}\right) Gini(D_1)$$

$$= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right)$$

$$= 0.443$$

$$= Gini_{income \in \{high\}}(D).$$

$Gini_{\{low,high\}} = 0.458$

$Gini_{\{medium,high\}} = 0.450$

Следователно, разделянето на {low,medium} (и {high}) има най-малък Gini index и трябва да се избере

Други примери с GINI

C1	**0**
C2	**6**

$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$

$Gini = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$

C1	**1**
C2	**5**

$P(C1) = 1/6 \quad P(C2) = 5/6$

$Gini = 1 - (1/6)^2 - (5/6)^2 = 0.278$

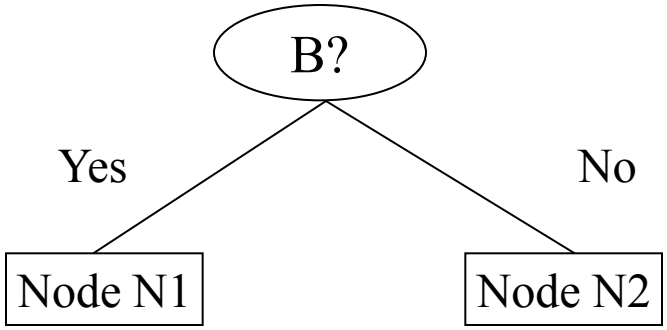
C1	**2**
C2	**4**

$P(C1) = 2/6 \quad P(C2) = 4/6$

$Gini = 1 - (2/6)^2 - (4/6)^2 = 0.444$

Бинарни атрибути

- Разделяне на две
- търся се по-големите и по-информативни части



	Parent
C1	6
C2	6
Gini = 0.500	

Gini(N1)
$= 1 - (5/6)^2 - (2/6)^2$
$= 0.194$

Gini(N2)
$= 1 - (1/6)^2 - (4/6)^2$
$= 0.528$

	N1	N2
C1	5	1
C2	2	4
Gini=0.333		

Gini(Children)
$= 7/12 * 0.194 +$
$5/12 * 0.528$
$= 0.333$

Категорийни атрибути

- За всяка стойност се изчислява индекс за всеки клас от множеството
- Изчислените стойности се записват в матрица за вземане на решение

Multi-way split

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

Two-way split
(find best partition of values)

	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

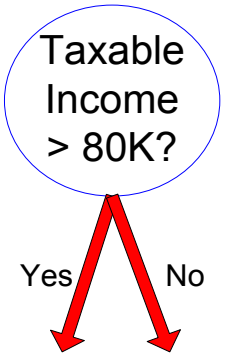
	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

Непрекъснати атрибути

- Прилагат се двоични дървета
- Възможности за избор на разделяне
  - колкото стойности има

- Всяка стойност има матрица с изчислени индекси
  - за всеки клас: $A < v$ или $A \geq v$
- Най-доброто v: с най-добър Gini index

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



Пример

- За всеки атрибут
  - сортиране по стойности
  - сканиране на стойностите и изчисляване на gini index
  - избор на позиция за разделяне: с най-малкия gini index

Cheat	No		No		No		Yes		Yes		Yes		No		No		No		No	
	Taxable Income																			
Sorted Values →	60		70		75		85		90		95		100		120		125		220	
Split Positions →	55		65		72		80		87		92		97		110		122		172		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		***0.300***		0.343		0.375		0.400		0.420	

Сравнение на мерките

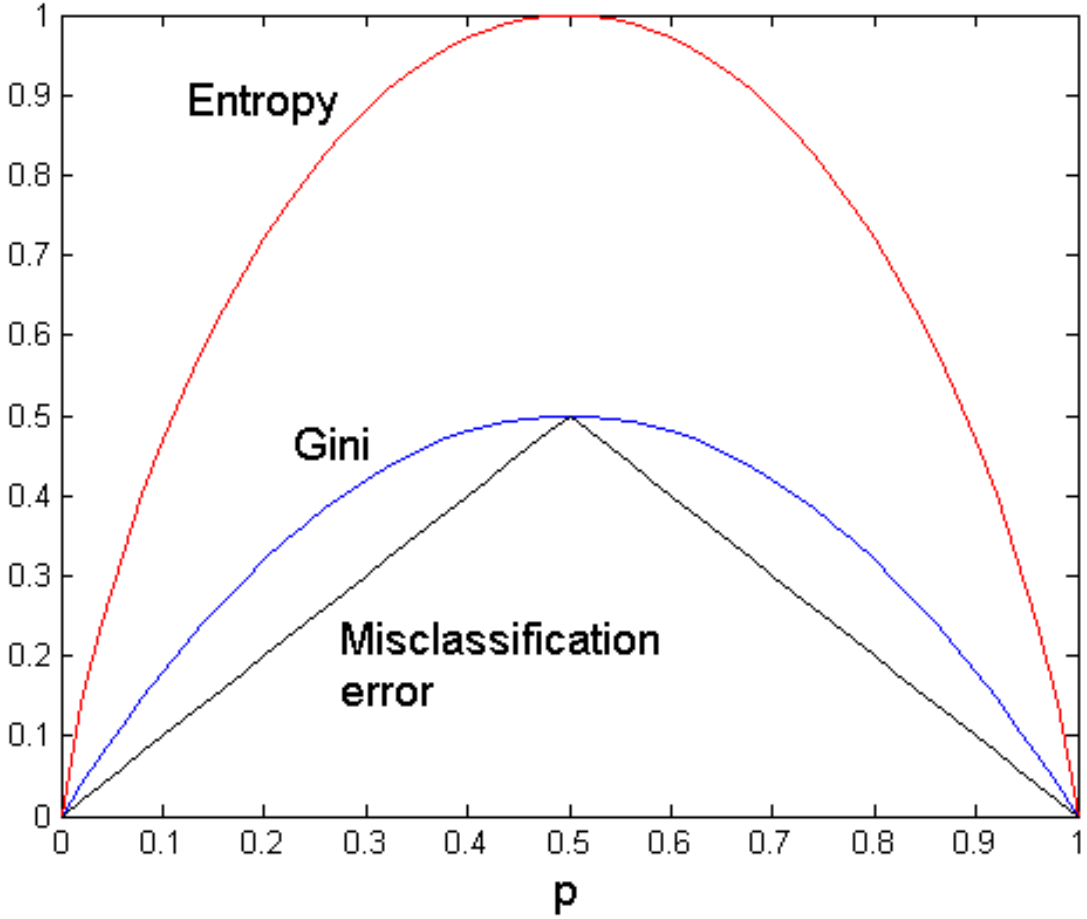
- **Information gain**:
  - за многовариантни атрибути
- **Gain ratio**:
  - за небалансирани разклонения
- **Gini index**:
  - проблеми при голям брой класове
  - за равномерно разпределени под-множества

Други мерки

- CHAID: a popular decision tree algorithm, measure based on χ^2 test for independence
- C-SEP: performs better than info. gain and gini index in certain cases
- G-statistic: has a close approximation to χ^2 distribution
- MDL (Minimal Description Length) principle (i.e., the simplest solution is preferred):
  - The best tree as the one that requires the fewest # of bits to both (1) encode the tree, and (2) encode the exceptions to the tree
- Multivariate splits (partition based on multiple variable combinations)
  - CART: finds multivariate splits based on a linear comb. of attrs.
- Which attribute selection measure is the best?
  - Most give good results, none is significantly superior than others

Сравнение

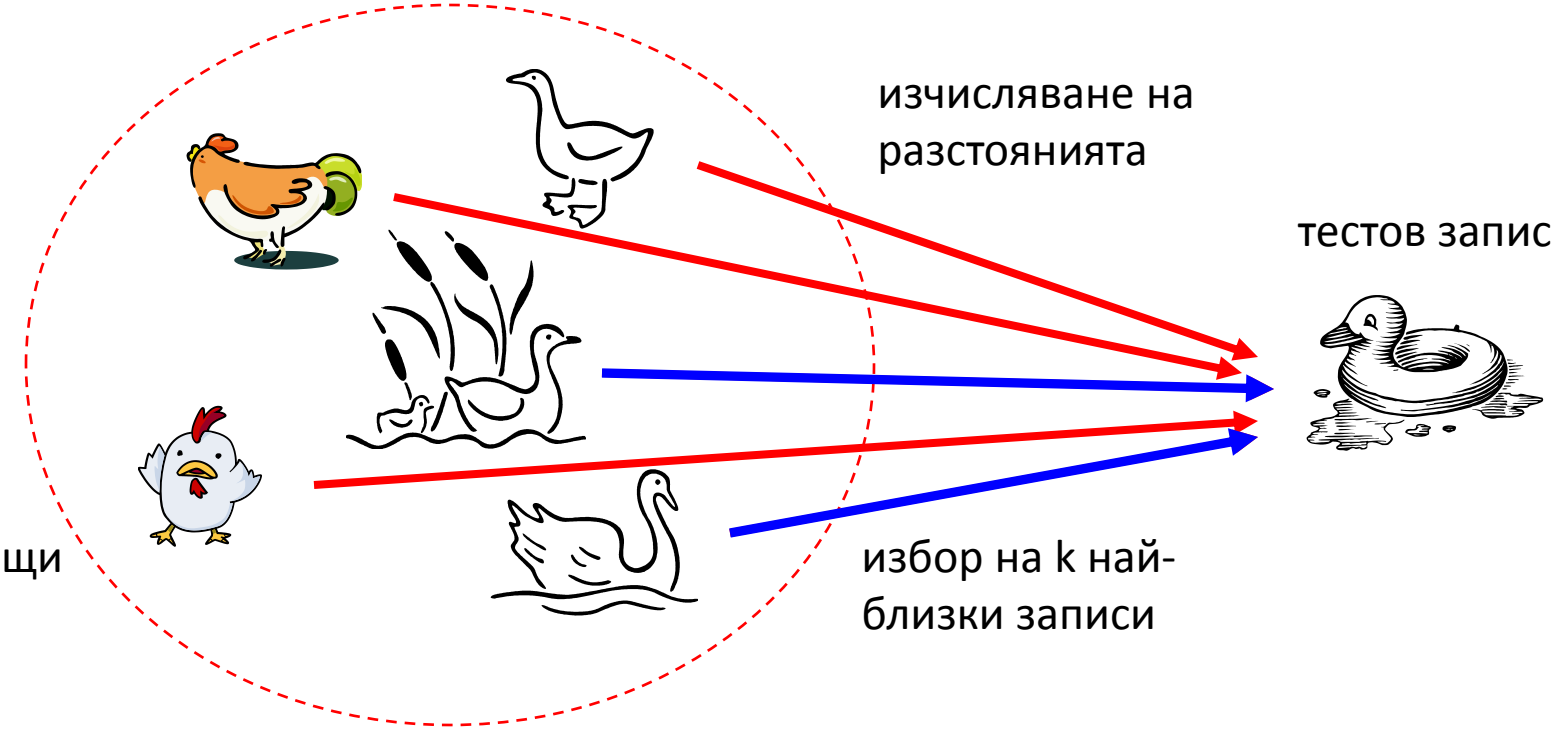
Пример: Разделяне в два класа



Най-близките съседи

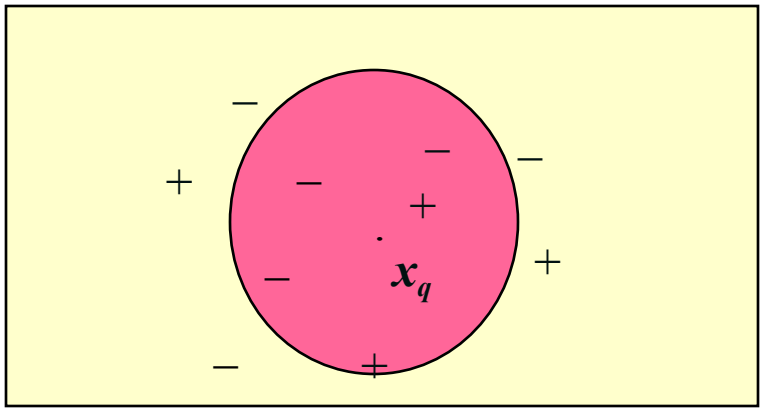
Най-близките съседи

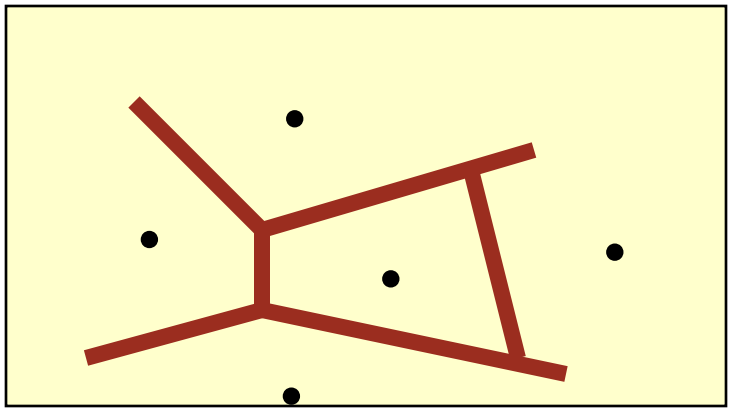
- Основна идея
  - Ако ходи като патица, плува като патица и квака като патица, най-вероятно е патица



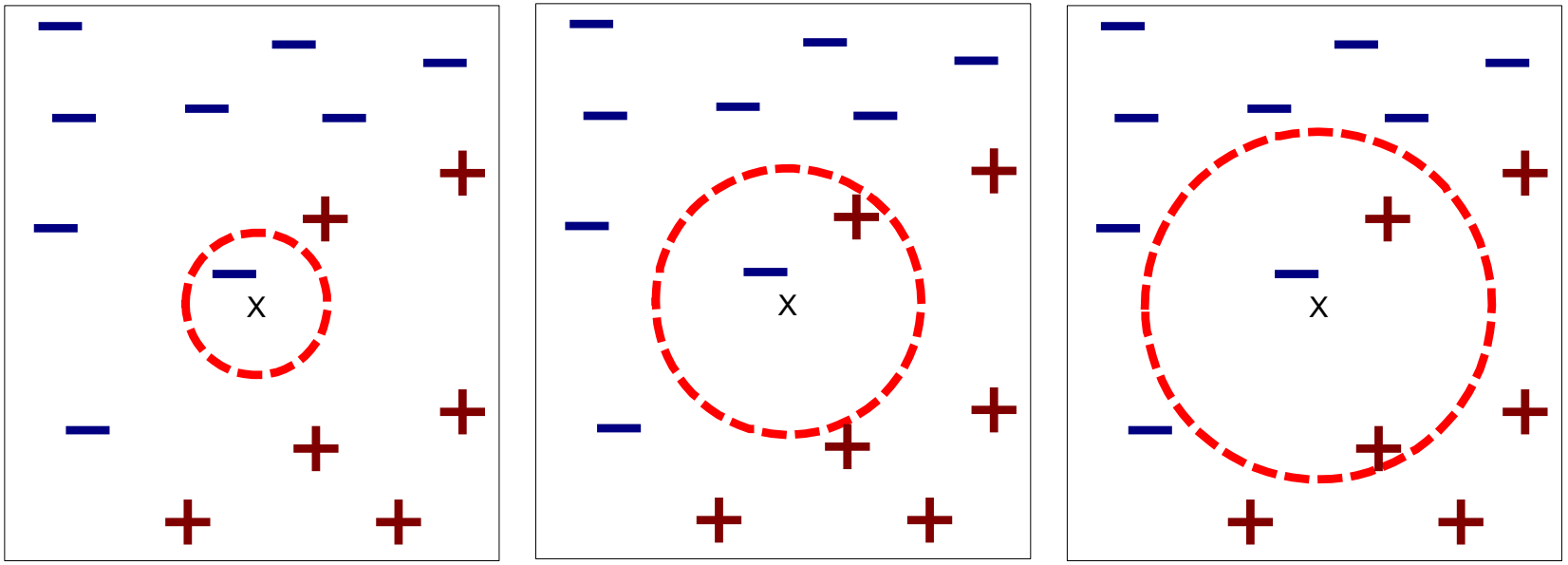
Алгоритъм

- Всички записи могат да бъдат представени като точки в n-D пространство
- Най-близкият съсед се дефинира в термините на Евклидовото разстояние между две точки $\mathrm{dist}(\mathbf{X}_1, \mathbf{X}_2)$
- Функцията за избор може да бъде дискретна или непрекъсната





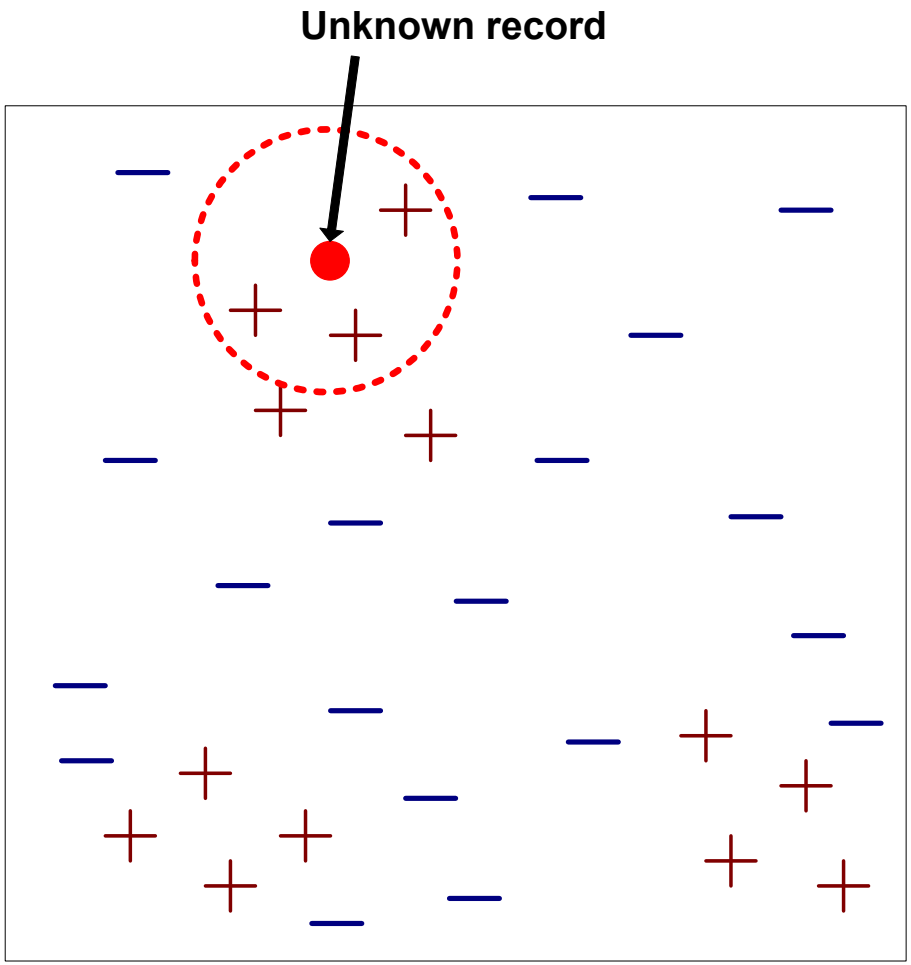
Дефиниране на най-близък съсед



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-най-близки съседи от запис x: обектите, които са разположени на първите k най-малки разстояния от x

Класификация



- Вход
  - обучаваща извадка
  - метрика за отстояние
  - брой на най-близките съседи, които да се изведат като кандидати - стойност *k*
- Процедура
  - изчисляване на отстоянията
  - избор на *k* най-близки съседи
  - определяне на най-вероятния клас

Класификация

- Изчисляване на растоянието между две точки
  - Евклидово разстояние

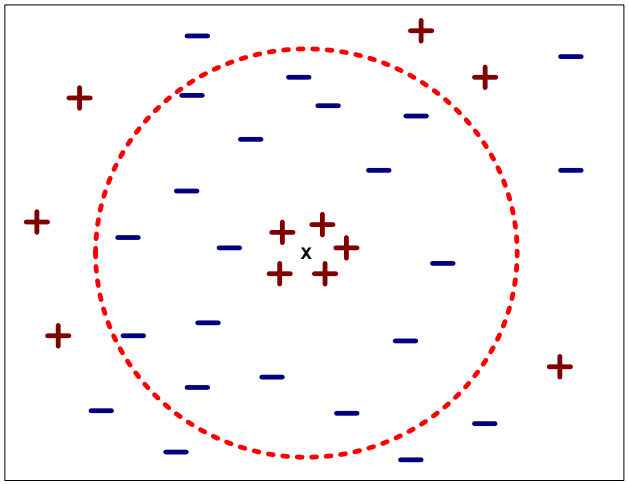
$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Избор на клас на принадлежност
  - вот / сравняване на k най-близки съседи
  - претегляне на вота
    - тегловен коефициент

$$w = 1 / d^2$$

Класификация

- Избор на стойност k:
  - Ако k е твърде малко, внася се висока чувствителност на шум
  - Ако k е твърде голямо, в обхвата може да попаднат обекти от други класове
  - Проверява се експериментално, започвайки от k=1



Класификация

- Мащабиране на атрибути
  - за приближаване на скалите на различните атрибути
  - Пример:
    - различни скали за мерките на един човек
      - височина – под 2 м
      - тегло – над 40 кг
      - доход – над 10000 лв
    - ако се обработват заедно, следва да са от един порядък

Допълнителни съображения

- Скалиране и нормализация на атрибутите
  - Ако има голяма разлика в метриките на атрибутите се налага нормализация
  - Примери
    - височината на човек варира от 1.5m до1.8m
    - теглото може да варира от 40 кг до 140 кг
    - доходът на човек може да варира от 250 лв. до 25000 лв
    - ако се обработват заедно, следва да са от един порядък
- Ако има липсващи стойности, заменят се с такива, които биха довели до най-голямо разстояние

Допълнителни съображения

- k-NN за прогнозиране на реални стойности - връща средните стойности от k съседи
- Усредняване на стойностите на k-съседи за намаляване на шума
- За избягване на доминирането на част от атрибутите на запис върху други, някои атрибути могат да бъдат изключени от изчисленията

Пример: PEBLS

Parallel Examplar-Based Learning System (Cost & Salzberg)

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

Разстояния между атрибути с номинални стойности:

d(Single,Married)

= | 2/4 – 0/4 | + | 2/4 – 4/4 | = 1

d(Single,Divorced)

= | 2/4 – 1/2 | + | 2/4 – 1/2 | = 0

d(Married,Divorced)

= | 0/4 – 1/2 | + | 4/4 – 1/2 | = 1

d(Refund=Yes,Refund=No)

= | 0/3 – 3/7 | + | 3/3 – 4/7 | = 6/7

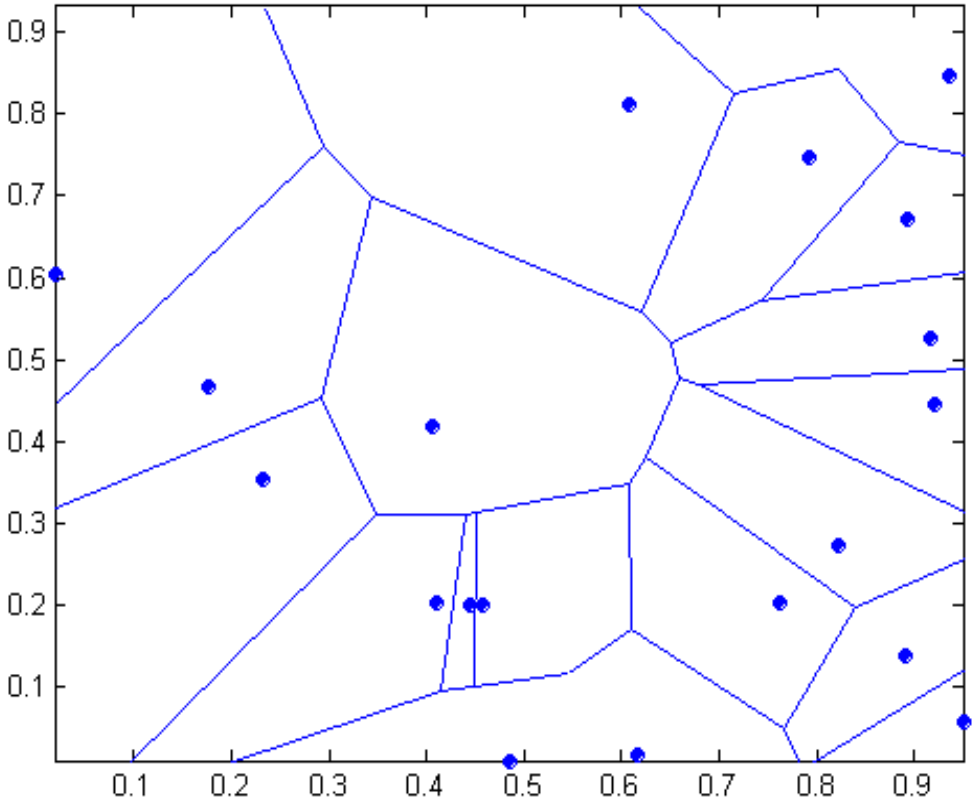
Class	Marital Status		
	Single	Married	Divorced
Yes	2	0	1
No	2	4	1

Class	Refund	
	Yes	No
Yes	0	3
No	3	4

$$d(V_1, V_2) = \sum_i \left| \frac{n_{1i}}{n_1} - \frac{n_{2i}}{n_2} \right|$$

K=1

Диаграма на Вороной



Недостатъци на метода

- Не се построява модел
- Относително скъпо класифициране на неизвестни записи
  - по отношение на изчислителните операции
- “Мързеливи класификатори”
  - в сравнение с дърво на решенията и система от правила