Rozhodovací stromy

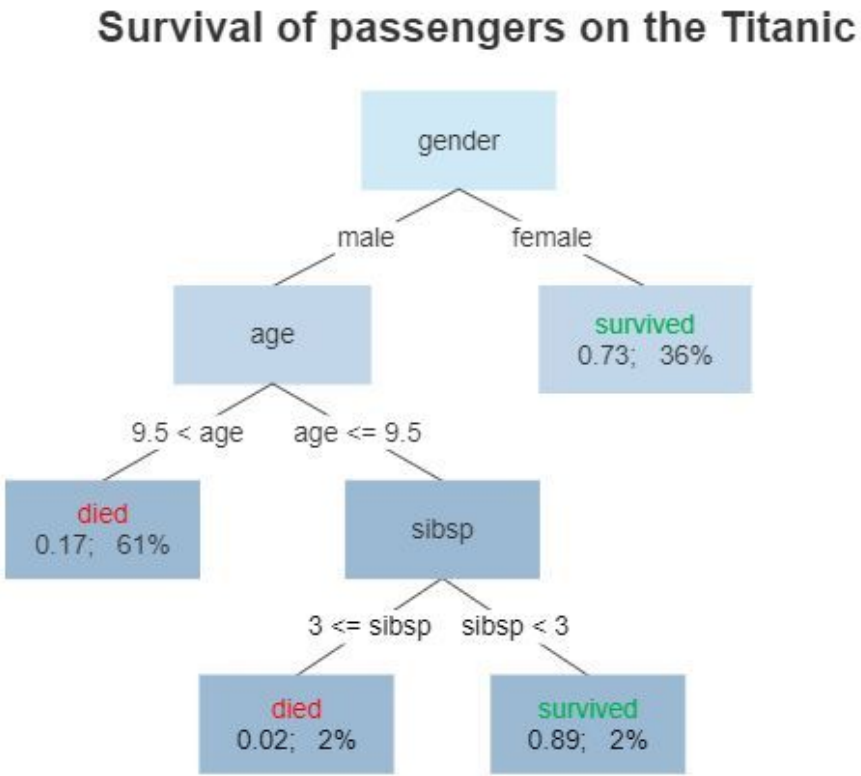
Rozhodovací strom

- Prediktivní model odhadující hodnotu z diskrétní množiny **Y** na základě příznaků **X**

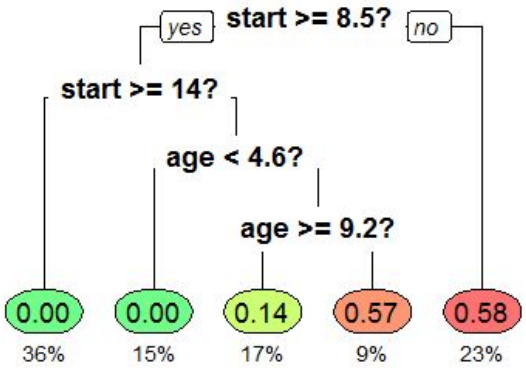
Rozhodovací strom

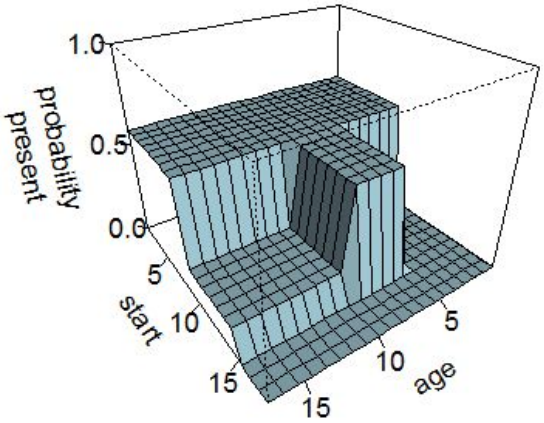
- Prediktivní model odhadující hodnotu z diskrétní množiny **Y** na základě příznaků **X**
- K rozhodnutí dochází testováním posloupnosti podmínek - uzel reprezentuje podmínku, list predikci

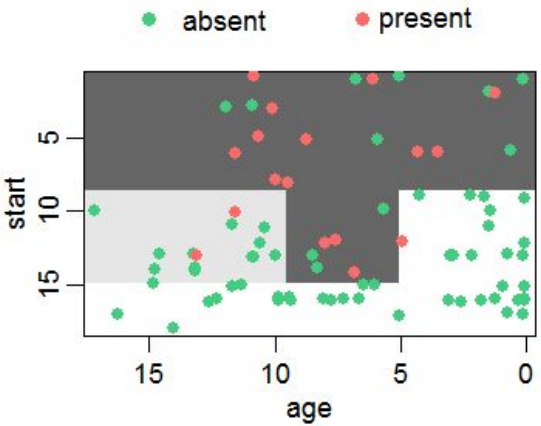
Rozhodovací strom - příklad



Rozhodovací strom - příklad







Rozhodovací strom

- Použití pro regresi i klasifikaci - výstupem může být i pravděpodobnostní rozdělení

Rozhodovací strom

- Použití pro regresi i klasifikaci - výstupem může být i pravděpodobnostní rozdělení
- Classification and regression tree analysis - **CART analysis**)

Rozhodovací strom

- Použití pro regresi i klasifikaci - výstupem může být i pravděpodobnostní rozdělení
- Classification and regression tree analysis - **CART analysis**)
- Neparametrická metoda učení s učitelem

Rozhodovací strom

- Použití pro regresi i klasifikaci - výstupem může být i pravděpodobnostní rozdělení
- Classification and regression tree analysis - **CART analysis**)
- Neparametrická metoda učení s učitelem
- Interpretovatelný “white box” algoritmus

Rozhodovací strom

- Použití pro regresi i klasifikaci - výstupem může být i pravděpodobnostní rozdělení
- Classification and regression tree analysis - **CART analysis**)
- Neparametrická metoda učení s učitelem
- Interpretovatelný “white box” algoritmus
- Budeme se zabývat binárním rozhodovacím stromem pro klasifikaci

Rozhodovací strom

- Jakákoli funkce ve výrokové logice se dá vyjádřit jako rozhodovací strom

Rozhodovací strom

- Jakákoli funkce ve výrokové logice se dá vyjádřit jako rozhodovací strom
- Pro n příznaků existuje 2^{2^n} funkcí, stromů pak ještě víc - **Jak vybrat ten správný?**

Jak sestrojit rozhodovací strom?

Jak sestrojit rozhodovací strom

- Jak určit podmínky, které nejlépe rozdělí vstupní množinu?

Jak sestrojit rozhodovací strom

- Jak určit podmínky, které nejlépe rozdělí vstupní množinu?
- Existuje více algoritmů v závislosti na vybrané metrice pro “nejlepší rozdělení”

Jak sestrojit rozhodovací strom

- Jak určit podmínky, které nejlépe rozdělí vstupní množinu?
- Existuje více algoritmů v závislosti na vybrané metrice pro “nejlepší rozdělení”
- Obvykle využívají strategii rozděl a panuj (divide and conquer) a hladový algoritmus (greedy search)

Jak sestrojit rozhodovací strom - algoritmus ID3

- Staví na konceptu entropie a informačního zisku

$$H_{Shannon}(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

Jak sestrojit rozhodovací strom - algoritmus ID3

- Staví na konceptu entropie a informačního zisku

$$H_{Shannon}(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

- Nechť ***X*** je množina všech tréninkových dat ve formě $(x, y) = (x_1, \ldots, x_k, y)$, kde $x_a \in V(a)$ je hodnota ***a***-tého příznaku ***x***, ***y*** je label a ***V(a)*** je množina hodnot ***a***-tého příznaku

Jak sestrojit rozhodovací strom - algoritmus ID3

- Nechť $S_a(v) = \{x \in X \mid x_a = v\}$ je množina tréninkových dat z ***X***, pro které je hodnota ***a***-tého příznaku rovna ***v***

Jak sestrojit rozhodovací strom - algoritmus ID3

- Nechť $S_a(v) = \{x \in X \mid x_a = v\}$ je množina tréninkových dat z ***X***, pro které je hodnota ***a***-tého příznaku rovna ***v***
- Pomocí vzájemně disjunktních $S_a(v)$ můžeme pokrýt celou množinu ***X***

Jak sestrojit rozhodovací strom - algoritmus ID3

- Nechť $S_a(v) = \{x \in X \mid x_a = v\}$ je množina tréninkových dat z ***X***, pro které je hodnota ***a***-tého příznaku rovna ***v***
- Pomocí vzájemně disjunktních $S_a(v)$ můžeme pokrýt celou množinu ***X***
- $H(X \mid a)$ je entropie ***X*** podmíněná hodnotou atributu ***a***

$$H(X \mid a) = \sum_{v \in V(a)} \frac{|S_a(v)|}{|X|} H(S_a(v))$$

Jak sestrojit rozhodovací strom - algoritmus ID3

- Informační zisk $IG(X, a) = H(X) - H(X \mid a),$

Jak sestrojit rozhodovací strom - algoritmus ID3

- Informační zisk $IG(X, a) = H(X) - H(X \mid a)$,
- IG je tedy rozdíl entropií před a po rozdělení množiny podmínkou

Jak sestrojit rozhodovací strom - algoritmus ID3

- Informační zisk $IG(X,a) = H(X) - H(X\,|\,a)$,
- IG je tedy rozdíl entropií před a po rozdělení množiny podmínkou
- Střední hodnota ***IG(X,a) = I(X,A),*** kde ***A*** je množina všech příznaků a ***I(X,A)*** je vzájemná informace ***X*** a ***A***

Jak sestrojit rozhodovací strom - algoritmus ID3

- Informační zisk $IG(X,a) = H(X) - H(X\,|\,a)$,
- IG je tedy rozdíl entropií před a po rozdělení množiny podmínkou
- Střední hodnota ***IG(X,a) = I(X,A),*** kde ***A*** je množina všech příznaků a ***I(X,A)*** je vzájemná informace ***X*** a ***A***
- Průměrně se tedy ***H(X)*** v každém uzlu snižuje o ***I(X,A)***

Jak sestrojit rozhodovací strom - algoritmus ID3

Algoritmus:

1. Spočítáme entropii všech příznaků v datasetu ***X***

Jak sestrojit rozhodovací strom - algoritmus ID3

Algoritmus:

1. Spočítáme entropii všech příznaků v datasetu ***X***
2. Rozdělíme ***X*** podle atributu, který maximalizuje ***IG***

Jak sestrojit rozhodovací strom - algoritmus ID3

Algoritmus:

1. Spočítáme entropii všech příznaků v datasetu ***X***
2. Rozdělíme ***X*** podle atributu, který maximalizuje ***IG***
3. Vytvoříme rozhodovací uzel s daným atributem

Jak sestrojit rozhodovací strom - algoritmus ID3

Algoritmus:

1. Spočítáme entropii všech příznaků v datasetu ***X***
2. Rozdělíme ***X*** podle atributu, který maximalizuje ***IG***
3. Vytvoříme rozhodovací uzel s daným atributem
4. Rekurzivně zpracujeme zbytek atributů

Jak sestrojit rozhodovací strom - algoritmus ID3

Rekurze končí, pokud:

1. Všechny zbývající prvky ***x*** patří do stejné třídy -> uzel je listem

Jak sestrojit rozhodovací strom - algoritmus ID3

Rekurze končí, pokud:

1. Všechny zbývající prvky ***x*** patří do stejné třídy -> uzel je listem
2. Všechny atributy byly vyčerpány, ale zbývající prvky ***x*** patří do více tříd -> označíme list třídou s největším zastoupením

Jak sestrojit rozhodovací strom - algoritmus ID3

Rekurze končí, pokud:

1. Všechny zbývající prvky ***x*** patří do stejné třídy -> uzel je listem
2. Všechny atributy byly vyčerpány, ale zbývající prvky ***x*** patří do více tříd -> označíme list třídou s největším zastoupením
3. Množina zbývajících prvků ***x*** je prázdná -> uzel je listem a pro rozhodnutí použijeme předka

Rozhodovací strom

- **Výhody:** jednoduchý, interpretovatelný, rychlý, stačí mu málo dat, automaticky vybere příznaky, verzatilní, poradí si s nenormalizovanými, nelineárními i chybějícími daty

- **Nevýhody:** mají tendenci overfittovat, nestabilní, špatně zvládají klasifikaci v případě nevybalancovaných tříd, nespojité predikce

Rozhodovací strom

- Často se kombinují do tzv. ensemble metod:
  a. Boosted trees - následující strom se zaměřuje na ta data, která se v předchozím nedařilo modelovat, např. **Adaboost**
  b. Bootstrap aggregated (bagged) trees - sestrojuje stromy nad opakovaně vzorkovanými daty, např. **Random forest**
  c. **Rotation forest** - aplikuje se po použití PCA