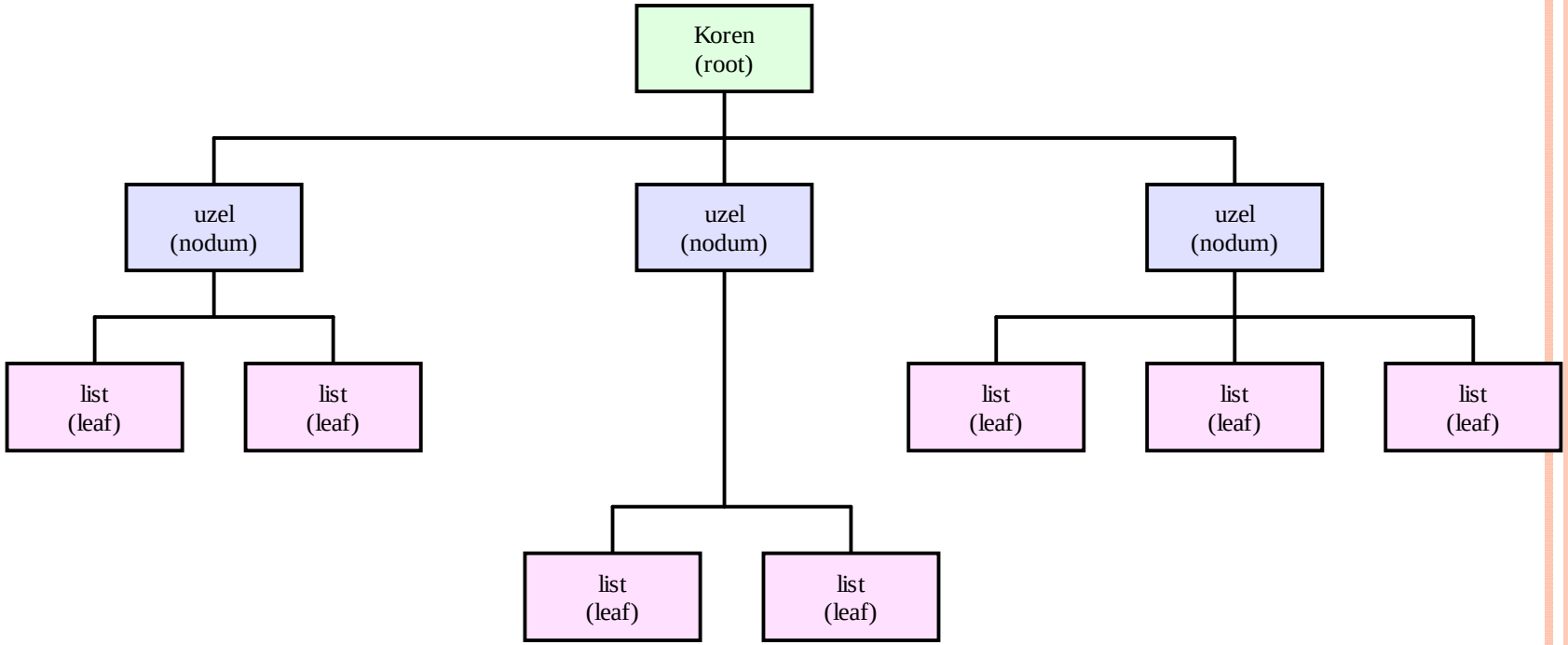
# Regresní a klasifikační stromy (rozhodovací stromy , Decision Trees)

# Regresní a klasifikační stromy (Regression , classification trees)

- Jsou nejméně formální a nejméně parametrickou skupinou **statistických modelů**
- Model – popisuje vzájemné vztahy mezi mezi pozorovanými veličinami
- Další příklady modelů: Lineární regresní model, Zobecněný regresní model(logistická regrese, poissonovská regrese), Neuronové sítě atd.

# Struktura stromu



# Struktura stromu

- Stromy se skládají z: **kořene, uzlů – neterminálních uzlů, listů - terminálních uzlů.**
- V každém neterminálním uzlu se strom větví

- **Binární stromy** – z jednoho uzlu vyrůstají právě dvě větve
- **Nebinární stromy** – z jednoho uzlu vyrůstají dvě a více větví

# Typy Stromů

- **Klasifikační (rozhodovací) strom** – modelujeme závislost *kategoriální* závisle proměnné na jedné či více nezávislých proměnných , prediktorech (kategoriálních, spojitých)

- **Regresní strom** - modelujeme závislost *spojité* závisle proměnné na jedné či více nezávislých proměnných, prediktorech (kategoriálních, spojitých)

# Úlohy - příklady

- **Klasifikační:**

1. Spamy – určení, který doručený e-mail je spam a který není spam.
2. Kosatce – třídění kostaců do jednotlivých druhů na základě velikosti jejich korunních a kališních plátků

- **Regresní:**

1. Ozón – modelování množství ozonu v závislosti na nadmořské výšce, teplotě a rychlosti větru

# Klasifikační stromy

# Co je to klasifikační strom?

- Patří mezi neparametrické metody (metody strojového učení, machine learning)
- Klasifikují vzorky do konečného (malého) předem daného počtu tříd
- Je to posloupnost rozhodnutí, jejímž výsledkem je zařazení objektu do jedné ze skupin na základě vlastnosti zkoumaného objektu
- V každém uzlu je určena **veličina**, podle které dělíme datový soubor a **hranice**, která určuje, kde se dělení má provést (je-li veličina spojitá)
- Kořen obsahuje celý datový soubor
- Z každého uzlu vyrůstají dvě (binární strom) nebo více větví
- Každý list představuje některou ze skupin *(úrovně kategoriální závisle proměnné). ....*
- Příklad: Botanický klíč

# Jinak řečeno….

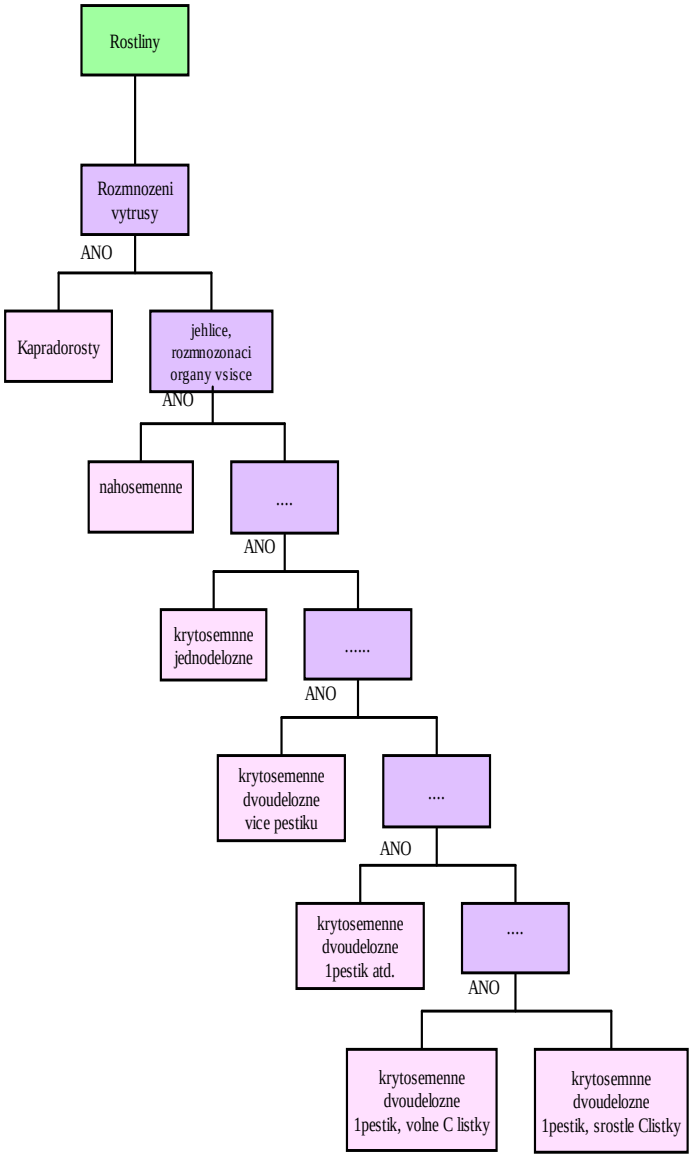
- Každé pozorování patří do jedné z tříd $C_1,\ldots,C_k$ ,
  $K \geq 2$

- Pozorování podle hodnot prediktorů postupuje od kořenového uzlu přes větvení v neterminálních uzlech k některému terminálnímu uzlu (listu)

- Množina všech listů určuje disjunktní rozklad prostoru hodnot prediktorů

- Terminálnímu uzlu a zároveň pozorováním, která do něj patří je přiřazena některá z tříd $C_1,\ldots,C_k$ .

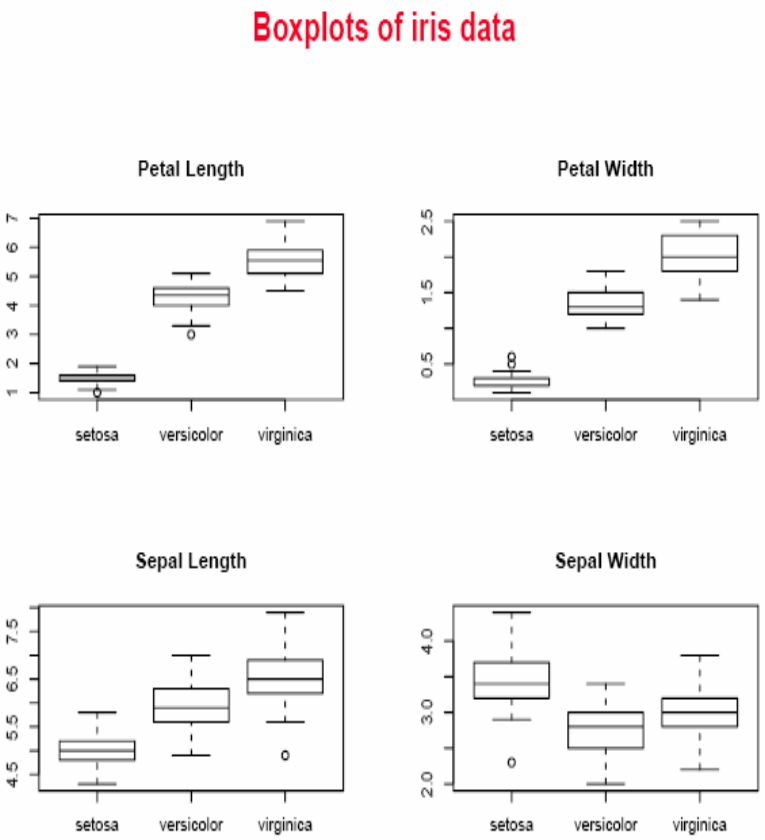
# Botanický klíč – určení skupin
# Klíč ke Květeně České republiky, str.48



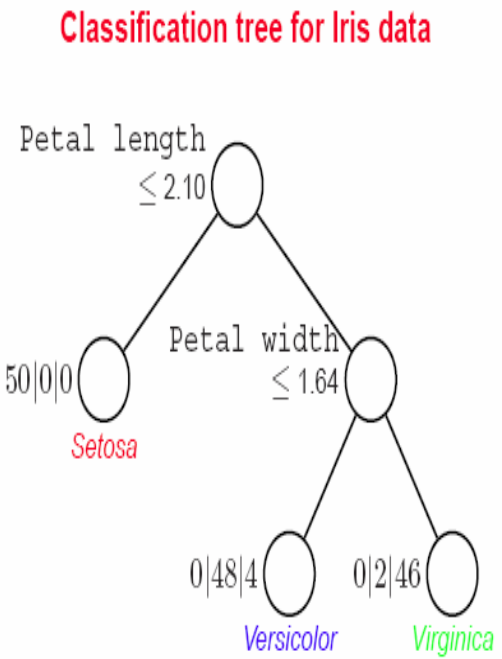
# IRIS data

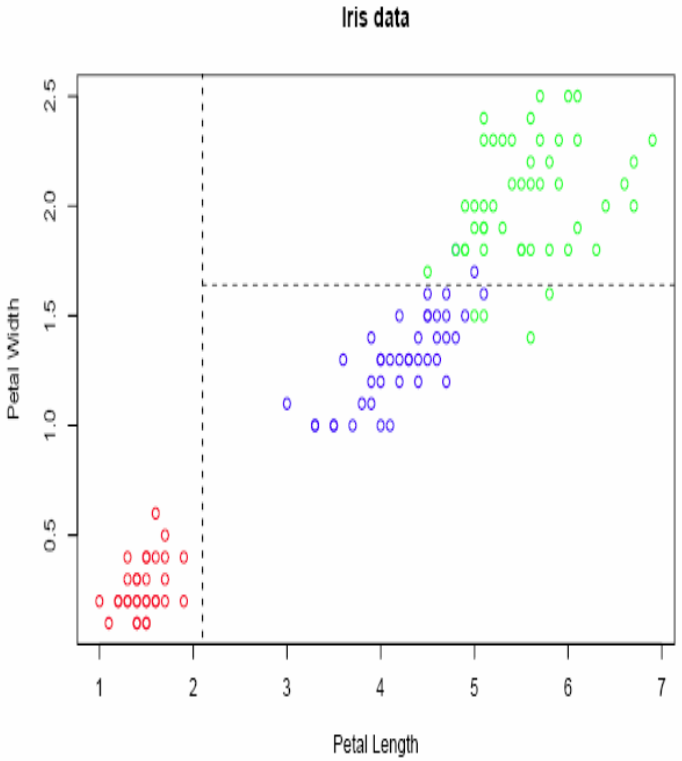
- 150 případů, vždy 50 případů ve skupině
- 3 skupiny – druhy kosatc Setosa, Versicolour, Virginica
- 4 prediktory: sepal length sepal width, petal length, petal width

- Zdroj příkladu: Yu-Shan Shih - Tree-structured methods



# IRIS data - stromy



# Klasifikační (rozhodovací) strom
# Výhody

- Snadné grafické znázornění
- Neklade žádné podmínky na typ rozdělení (lineární regresní model – požadavek normality reziduí)
- Algoritmy tvorby stromu jsou odolné vůči odlehlým hodnotám
- trénovací data mohou obsahovat chybějící hodnoty
- dokáží zachytit nelineární vztahy mezi proměnnými a vysvětlující proměnnou
- lze použít pro korelované proměnné

# Klasifikační (rozhodovací) strom
# Nevýhody

- **Nestabilita** - tvar stromu velmi závisí na datech, malá změna v datech způsobí změny v rozhodovacích pravidlech uvnitř uzlů

  + změna výsledných klasifikací

- Vzhledem k nestabilitě je nutná opatrnost při interpretaci.

- Řešení: např. **Bagging** – kombinace více stromů dohromady, aby se minimalizovala jejich variabilita (bude vysvětleno později viz. klasifikační lesy)

- accuracy (měření přesnosti stromu) je výrazně závislá na krosvalidačním mechanizmu, selekčních kritériích a výběru mechanizmu pro minimalizaci chyby stromu

- nevhodné pro malý počet vzorků

# Jak roste strom?

- Existuje mnoho **algoritmů**, jak vybírat proměnné a hranice podle kterých bude probíhat dělení datového souboru.
- hlavní princip: vyber takovou proměnnou, která rozdělí trénovací soubor na co nejhomogennější skupiny (algoritmy, jimiž se stromy vytváří, tj. algoritmy učení)
- Algoritmus speciální pro klasifikaci: **QUEST**
- Pro regresi i klasifikaci: **CART(C&RT), CHAID**
- Další algoritmy: **ID3, C4.5, C5.0**
- Obecně nelze říci, který z algoritmů je lepší
- Výsledkem je vždy strom, který se však liší obsahem uzlů i jejich počtem.

# systémy generující stromy

- ID3 (Quinlan 79)
- CART (Breiman et al. 84)
- Assistant (Cestnik et al. 87)
- C4.5 (Quinlan 93)
- C5 (Quinlan 97)
- Stromy ve Wece (Frank 2000)
- Stromy v Orange (Demšar, Zupan 2000)
- RETIS (Karalič 1992) – pro regresní stromy

# K čemu budeme klasifikační stromy využívat?

Máme data

- zajímá nás struktura těchto dat, postižení vzájemných vztahů - **DATA NYNÍ**
- Predikce, klasifikace dosud neznámých případů - **DATA V BUDOUCNU**

# Pěstujeme klasifikační strom

- Rozdělení dat do skupin:

  **trénovací data + testovací data**

- Tvorba stromu na základě **trénovacích** dat
- Posouzení schopnosti stromu správně klasifikovat neznáme případy pocházející z množiny **testovacích** dat. Posouzení predikční schopnosti stromu.
- Někdy také dělení dat do skupin:

**trénovací data+validační data + testovací data**

# Křížové ověřování
## Cross-validace

- Rozdělení datového souboru do **K** skupin

  (obvykle **k**=10)
- Jedna skupina vždy označena jako testovací. Zbytek skupin slouží k tvorbě stromu.
- Každá skupina je testovací právě jednou.
- Celkem vytvořeno **K** stromů. Na základě testovací skupiny ohodnotíme predikční schopnosti stromu.
- máme-li malý počet vzorků- možnost použití např. Jack-knife (vybere se vždy jen jeden vzorek…) -problémy při krosvalidaci není-li dostatečně velký soubor pro dělení na testovací a trénovací podsoubory – při rozdělení může dojít ke ztrátě informace u trénovacích dat a výsledný strom pak chybně klasifikuje

# Křížové ověřování (Cross-validace)

Rozdělení datového souboru do **K** skupin (zde **k**=6)

| | | | | | |
|---|---|---|---|---|---|
| TEST | TRAIN | TRAIN | TRAIN | TRAIN | TRAIN |
| TRAIN | TEST | TRAIN | TRAIN | TRAIN | TRAIN |
| TRAIN | TRAIN | TEST | TRAIN | TRAIN | TRAIN |
| TRAIN | TRAIN | TRAIN | TEST | TRAIN | TRAIN |
| TRAIN | TRAIN | TRAIN | TRAIN | TEST | TRAIN |
| TRAIN | TRAIN | TRAIN | TRAIN | TRAIN | TEST |

# Velikost stromu

## Příliš velký strom

- Může být „přeučený“, tj. může být příliš specializovaný na datový soubor, který se použil pro jeho konstrukci.
- Pokud ho použijeme pro klasifikaci „neznámých“ případu, nemusí být příliš úspěšný.
- Neplatí tedy, že čím je strom větší, tím je lepší.
- Dobře naučený strom nepopisuje každý konkrétní případ, spíše by měl popisovat obecnější závislosti, které se v datech vyskytují.
- Příliš malý strom

## Příliš malý strom

- Nemusí postihnout strukturu dat

# Snižování složitosti stromu

- **Zastavíme růst stromu,** když další dělení není statisticky významné

- Začínáme s „přerostlým“, příliš detailně větveným stromem. Tento strom následně redukujeme pomocí některé z metod
  - **Prořezávání (pruning)**
  - **Zmenšování, scvrkávání se (shrinking)** - ***metoda pro regresní strom!***

# Prořezávání (Cost - complexity Prunning)

1. "Přerostlý" strom, T
2. Odřezání koncových větví T - vytváření jednoduššího stromu, $T_1$.
3. Cena jednoduššího stromu (cost-complexity criterion):

$$D_{stromT_1} + \alpha * |T_1| = C_\alpha(T_1),$$

kde $|T_1|$ je počet listů daného stromu, $D_{stromT_1}$ je deviance stromu, $\alpha \geq 0$ vyvažuje vztah mezi velikostí stromu a kvalitou stromu (modelu)

4. Hledáme takový strom, který minimalizuje $C_\alpha(T_1)$.

# Regresní stromy

# Regresní stromy I.

- Modelujeme  závislost *spojité* závisle proměnné na jedné či více nezávislých proměnných (kategoriálních, spojitých)

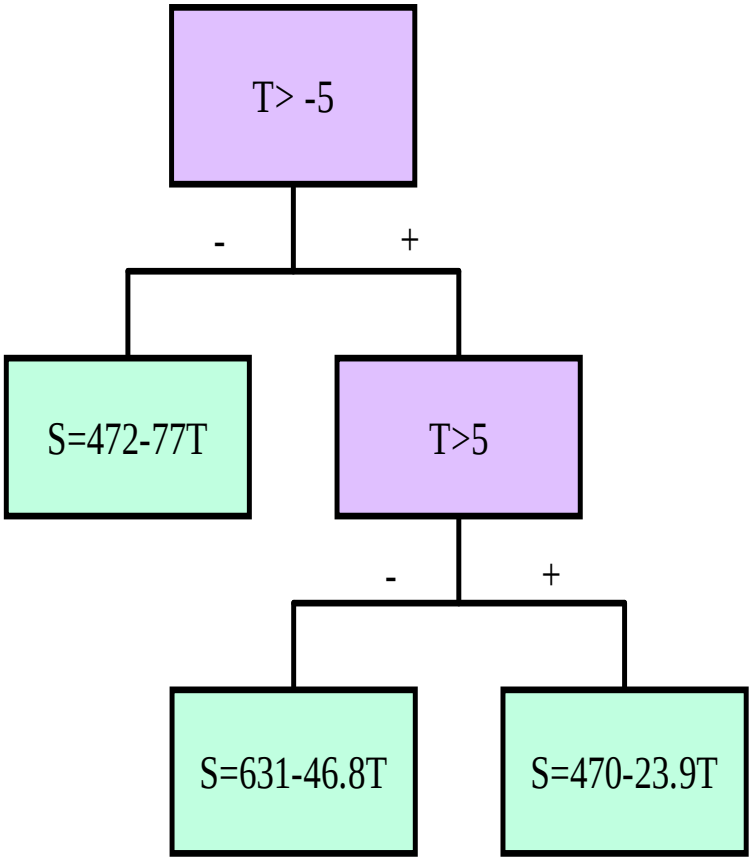
- Je to posloupnost rozhodnutí, jejímž výsledkem je zařazení objektu do jednoho  z listů na základě vlastností zkoumaného objektu

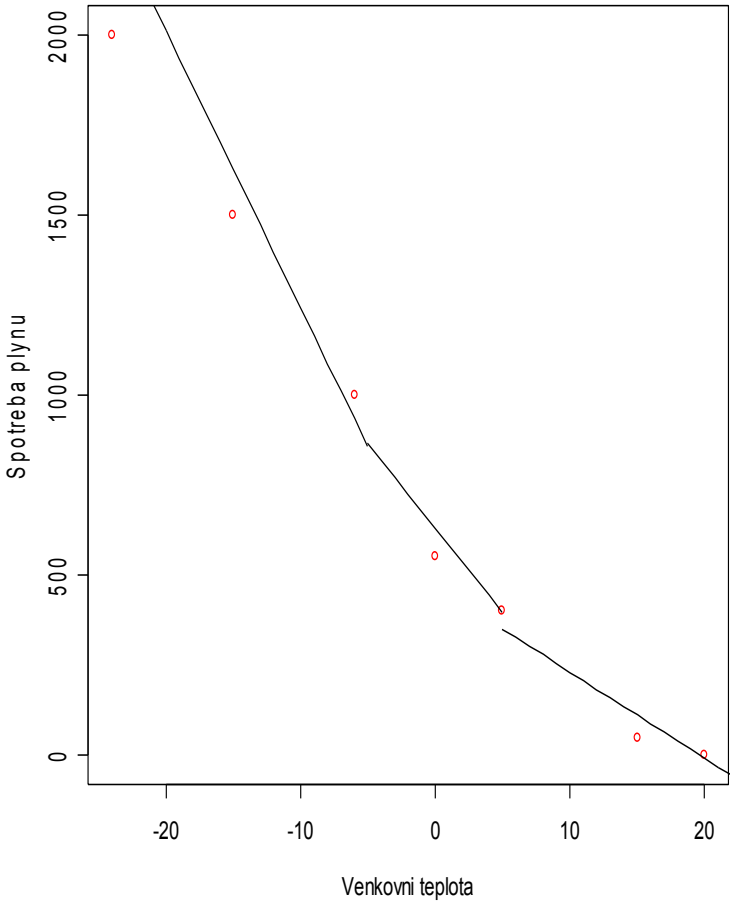
- V každém uzlu je určena **veličina**, podle které dělíme datový soubor a **hranice**, která určuje, kde dělení má provést (je-li veličina spojitá).

# Regresní stromy II.

- Kořen obsahuje celý datový soubor
- Z každého uzlu vyrůstají dvě (binární strom) nebo více větví
- Každému objektu z koncových listů je přiřazena hodnota, kterou vypočteme jako aritmetický průměr hodnot všech objektů v příslušném listu.
  - Další možností je vytvořit pro jednotlivé listy regresní modely (viz.následující příklad).

# Příklad: Ukázka regresního stromu
## Závislost spotřeby plynu na venkovní teplotě



# Výhody a nevýhody regresních stromů

*Výhody*

- Snadné grafické znázornění
- Neklade žádné podmínky na typ rozdělení (lineární regresní model – požadavek normality reziduí)
- Algoritmy tvorby stromu jsou odolné vůči odlehlým hodnotám
- Možno použít korelované proměnné

*Nevýhody*

- **Nestabilita - tvar stromu velmi závisí na datech**
- **Modelovaná regresní plocha je nespojitá (je-li výsledná hodnota listu rovna aritmetickému průměru hodnot všech objektů v příslušném listu).**

# Posouzení kvality regresního stromu

- Minimalizujeme Střední kvadratickou chybu (MSE – mean square error)

- S rostoucí velikostí stromu sice MSE na množině trénovacích dat klesá. Ale skutečná chyba zpravidla  klesá jen do určité velikosti stromu. Pak s narůstající velikostí stromu opět roste.

## Software - stromy

- ***Komerční:***
  1. Statistica
  2. Clementine, data miner k SPSS
  3. Answer Tree (SPSS)
  4. Matlab

- ***Volně šiřitelné:***
  1. R – knihovny tree, rpart
  2. C 4.5 http://*www.rulequest.com/Personal*

# Literatura k dalšímu studiu


- Hastie T., Tibshirani R., Friedman J.: The Elements of Statistical Learning, Data mining, Inference and Prediction, Springer 2003
- Lažanský et. Kol.: Umělá inteligence I.- IV.
- Jan Klaschka, Emil Kotrč: Klasifikační a regresní lesy, sborník konference ROBUST 2004
- Breiman, L. et al (1984) Classification and Regression Trees, Chapman and Hall
- Breiman L. (2001) Random forests. Machine Learning 45, pp. 5-32.