Classification Tree CART (1)

Kuliah 3 - STA1382 Teknik Pembelajaran Mesin

Septian Rahardiantoro

Outline

- Overview Decision Tree
- Regression Tree

Overview Decision Tree

- Untuk membuat prediksi untuk pengamatan tertentu, biasanya digunakan rata-rata pengamatan data training di wilayah yang menjadi miliknya.

- Karena seperangkat aturan pemisahan yang digunakan untuk mensegmentasi ruang prediktor dapat dirangkum dalam sebuah pohon, jenis pendekatan ini dikenal sebagai metode pohon keputusan.

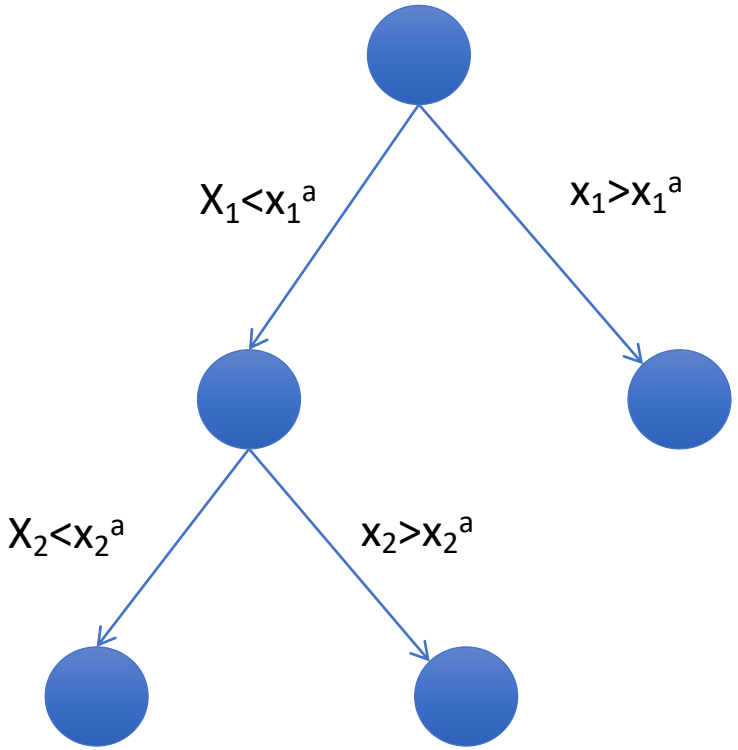
- Keunggulan metode berbasis pohon adalah bersifat sederhana dan berguna untuk interpretasi.

Ide Dasar

Segmentasikan ruang prediktor menjadi sub-wilayah

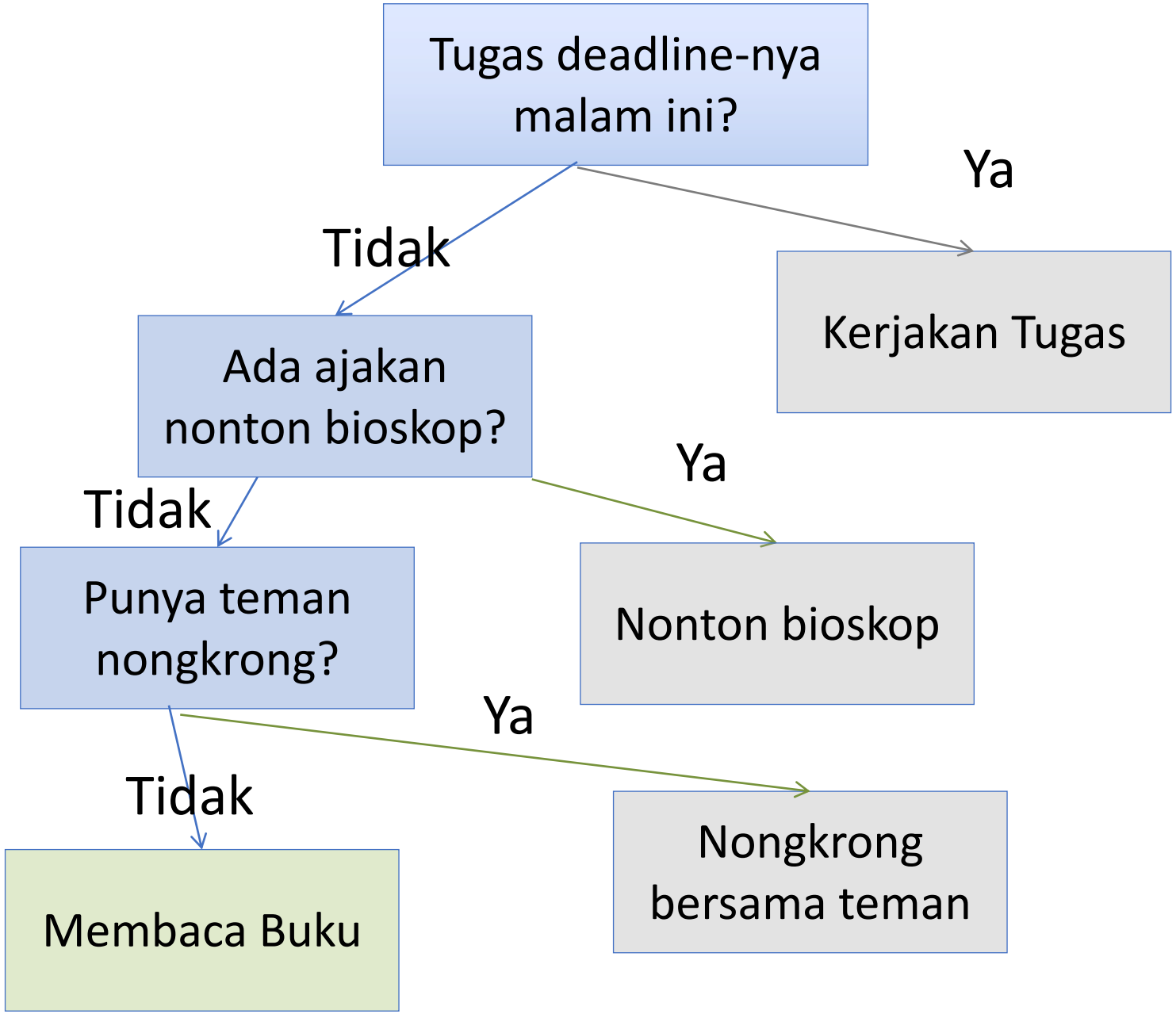
Berdasarkan data trainingnya, dapat ditentukan nilai yang akan diprediksi sebagai mean atau mode atau median dari peubah respons dari contoh pelatihan yang ada di segmen tersebut.

Kenapa Tree?



Apa yang akan kamu lakukan malam ini?
Putuskan di antara hal-hal berikut:

- Menyelesaikan tugas
- Pergi nonton bioskop
- Membaca buku
- Nongkrong dengan teman-teman



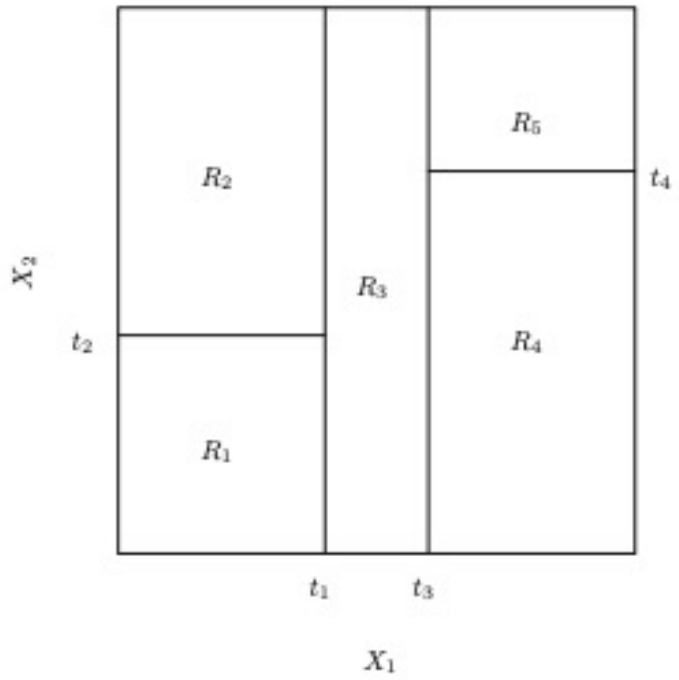
Kenapa Tree?

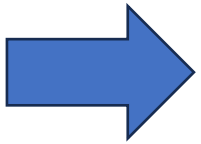
Membagi ruang prediktor sebagai cabang pohon untuk proses dua kemungkinan dan oleh karena itu metode ini disebut metode pohon keputusan

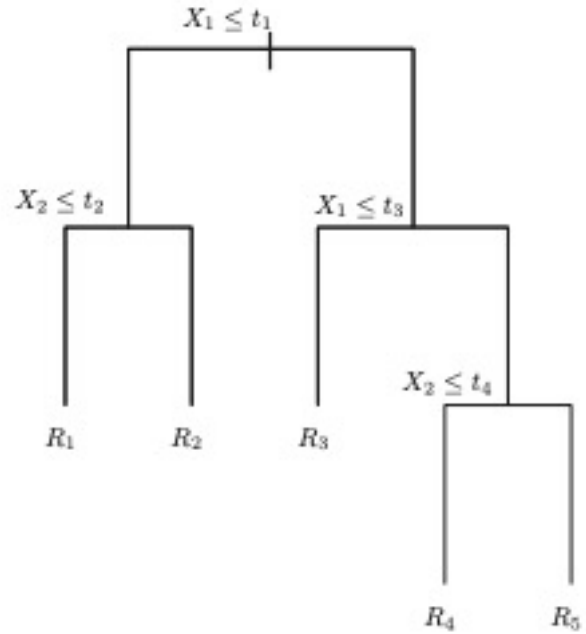
Pohon keputusan (decision tree) dapat diterapkan pada masalah regresi dan klasifikasi

Regression Tree

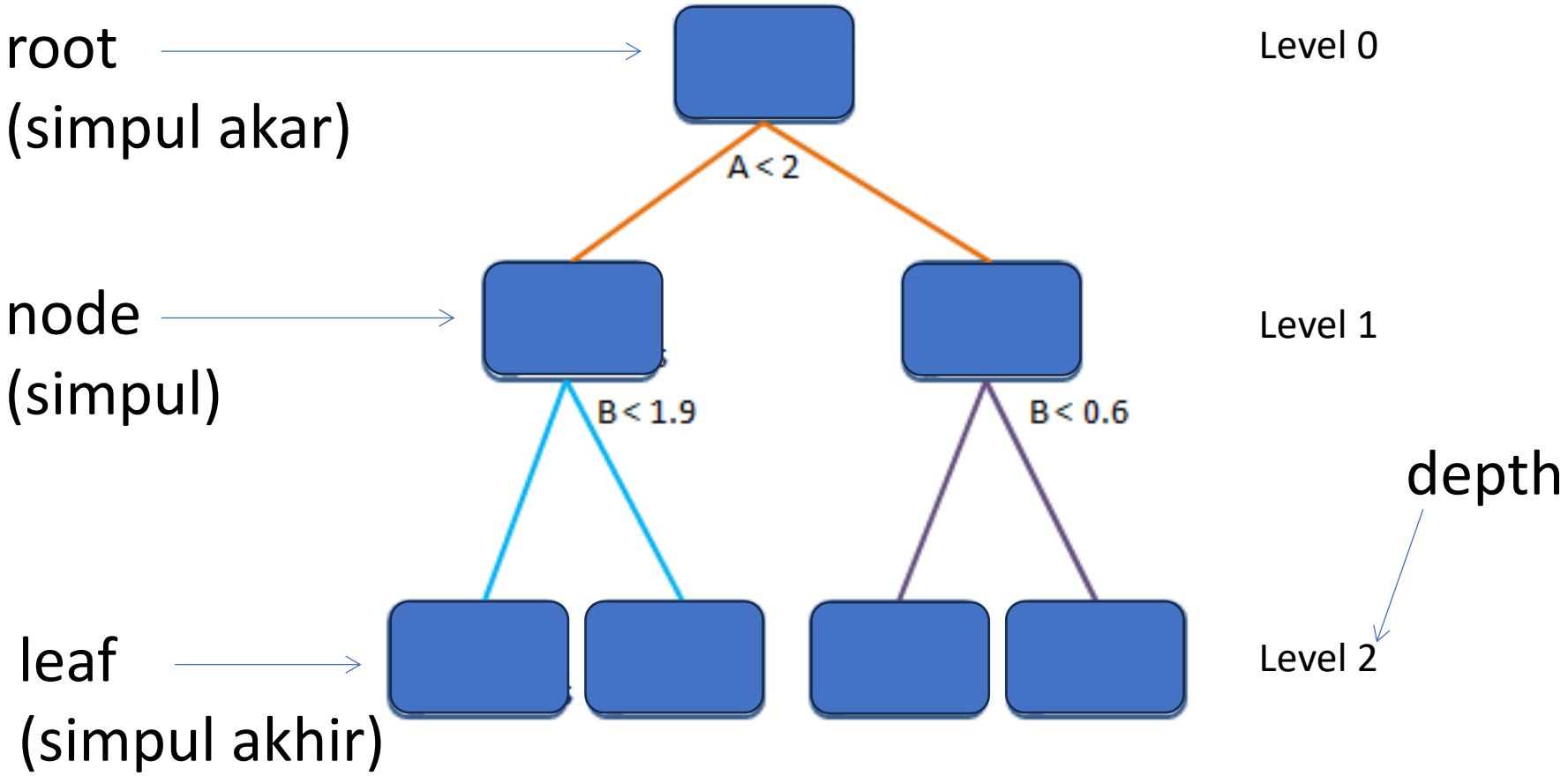
- Merupakan pohon keputusan yang diaplikasikan pada kasus regresi dengan peubah Y berskala numerik







Beberapa istilah



Membangun Regression Tree

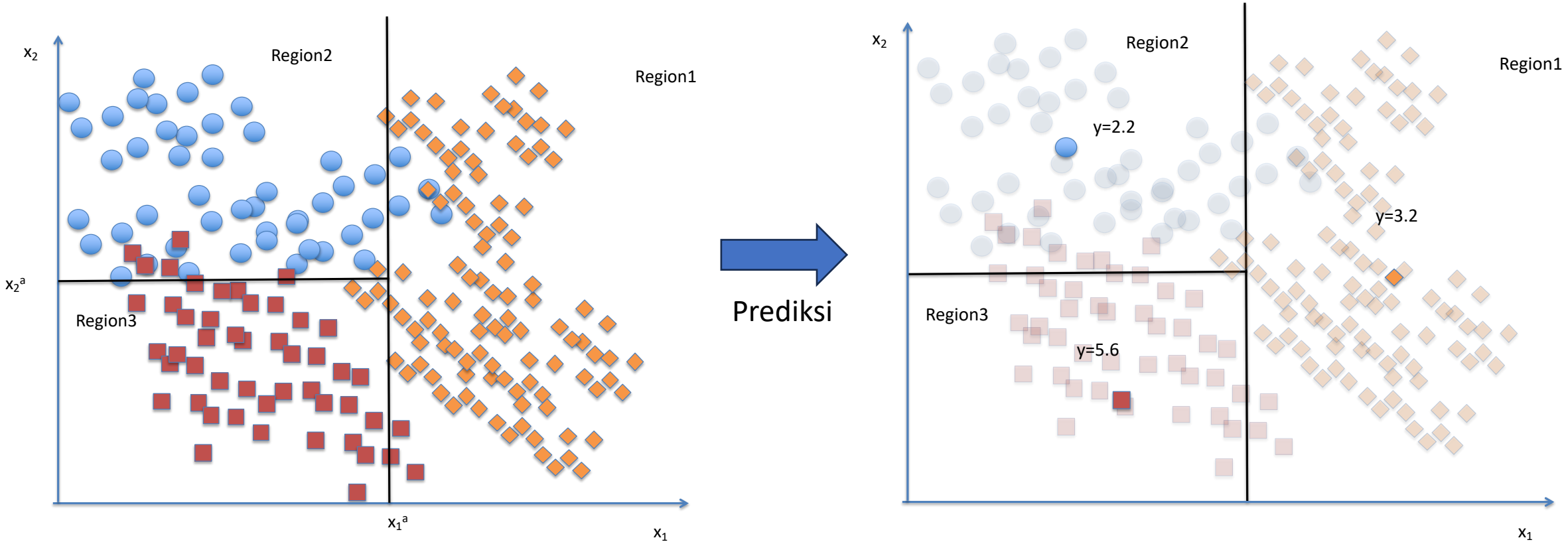
- Bangun pohon regresi:
  - Bagilah ruang prediktor menjadi J daerah berbeda yang tidak tumpang tindih $R_1, R_2, \ldots, R_J$
  - Nilai prediksi yang sama ditentukan untuk semua observasi di wilayah yang sama; gunakan rata-rata peubah respons untuk semua observasi data training yang ada di wilayah tersebut

Ilustrasi

Misalkan diketahui peubah respon Y dengan dua peubah prediktor X_1 dan X_2.

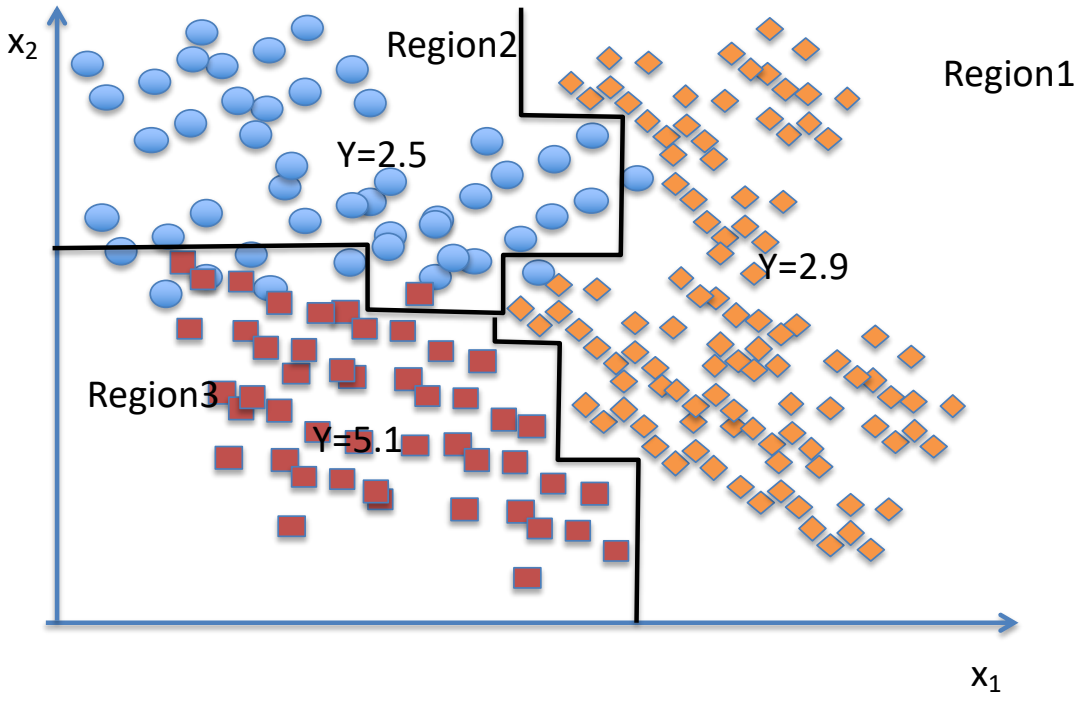
Prediktor X_1 disekat pada nilai $X_1 = x_1^a$, sehingga terbentuklah Region1 dan Region2*

Lalu pada Region 2*, prediktor X_2 disekat pada nilai $X_2 = x_2^a$ menjadi Region2 dan Region3

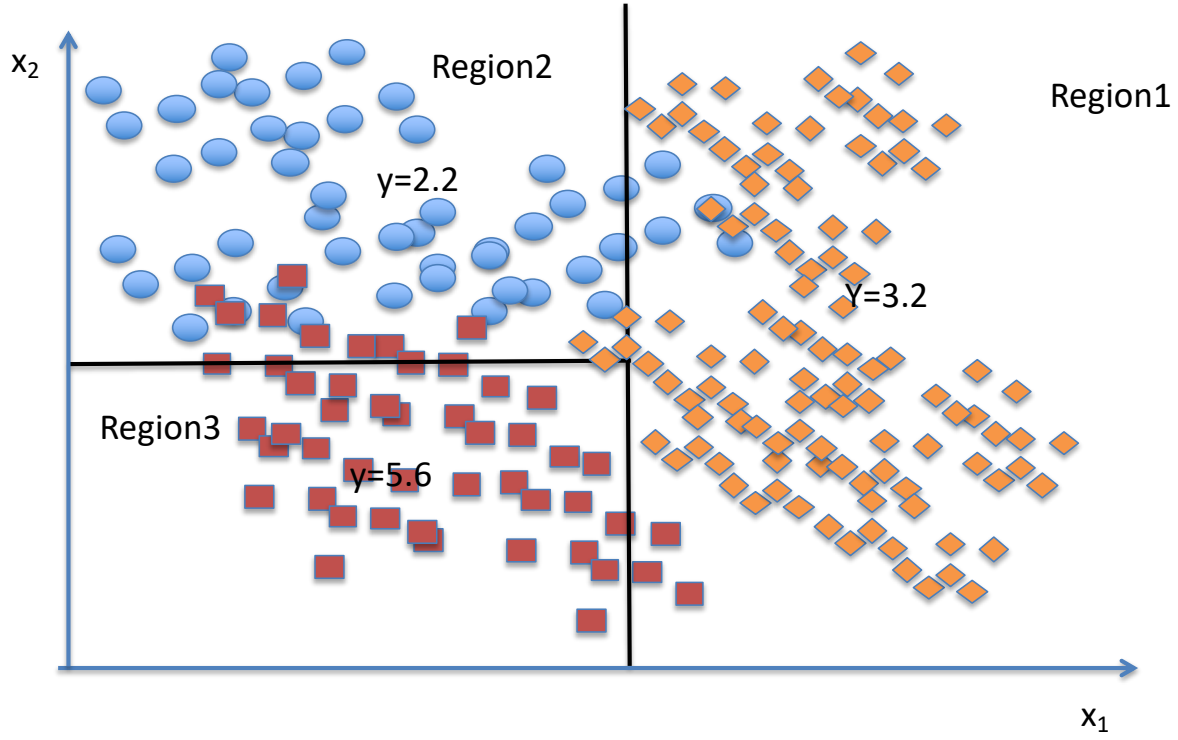


Bagaimana membagi wilayah prediktor?

Bentuk wilayahnya bisa apa saja.



Untuk memudahkan, dipilih persegi panjang saja



Caranya:

Temukan wilayah persegi panjang $R_1, R_2, \ldots, R_J$ yang meminimumkan JKG

$$JKG = \sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j} \right)^2$$

dengan $\hat{y}_{R_j}$ adalah nilai rataan peubah respons dari pengamatan data training di dalam wilayah R_j

→Namun proses komputasinya sangat mahal!

Solutsi: Top down approach, greedy approach

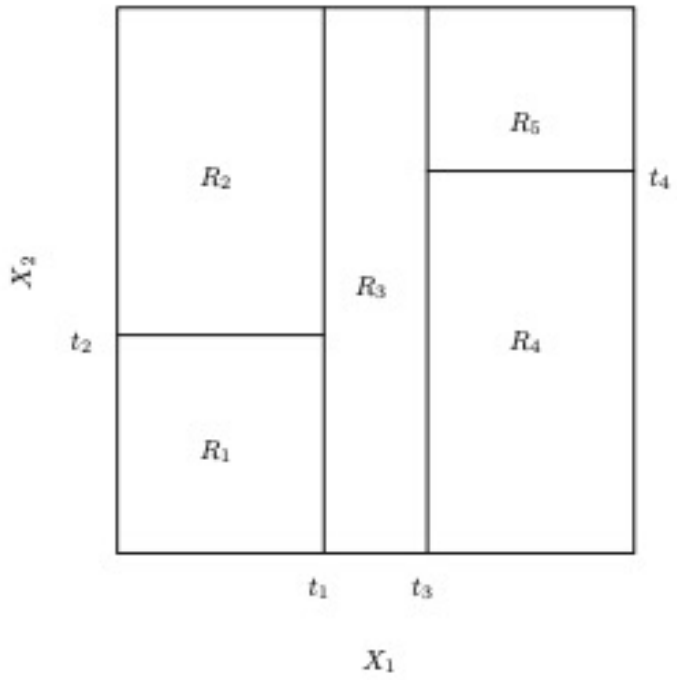
recursive binary splitting

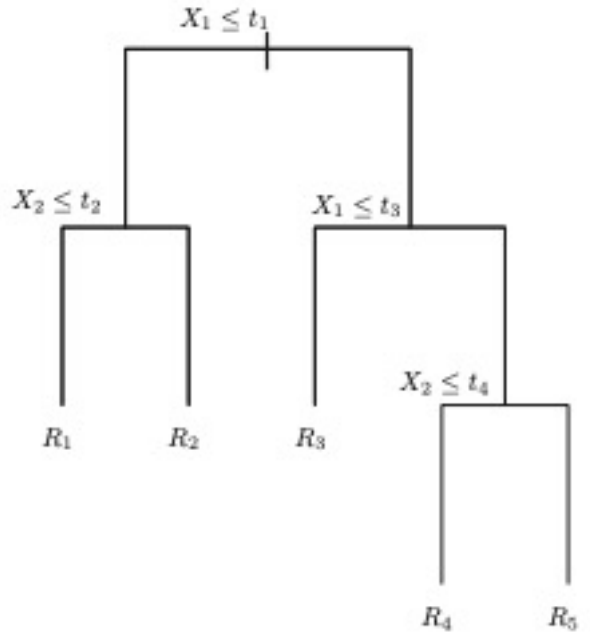
Recursive Binary Splitting

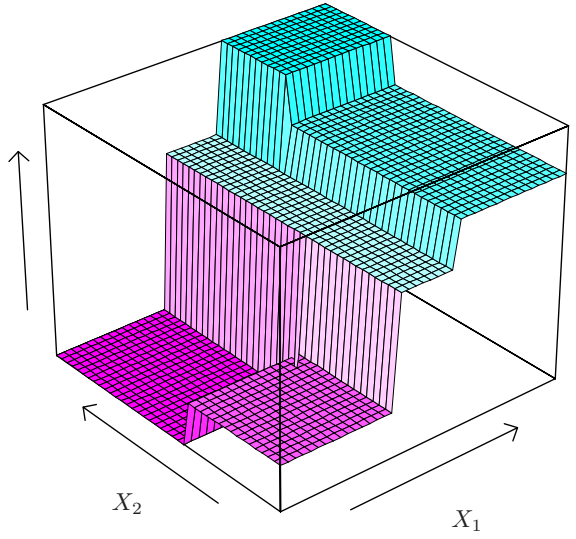
1. Consider all predictor X_p and all the all possible values of the cutpoints *s* for each of the predictors. Choose the predictor and cutpoint s.t. it minimizes the RSS (JKG)

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2$$

   This can be done quickly, assuming number of predictors is not very large
2. Repeat #1 but only consider the sub-regions
3. Stop: node contains only one class or node contains less than *n* data points or max depth is reached

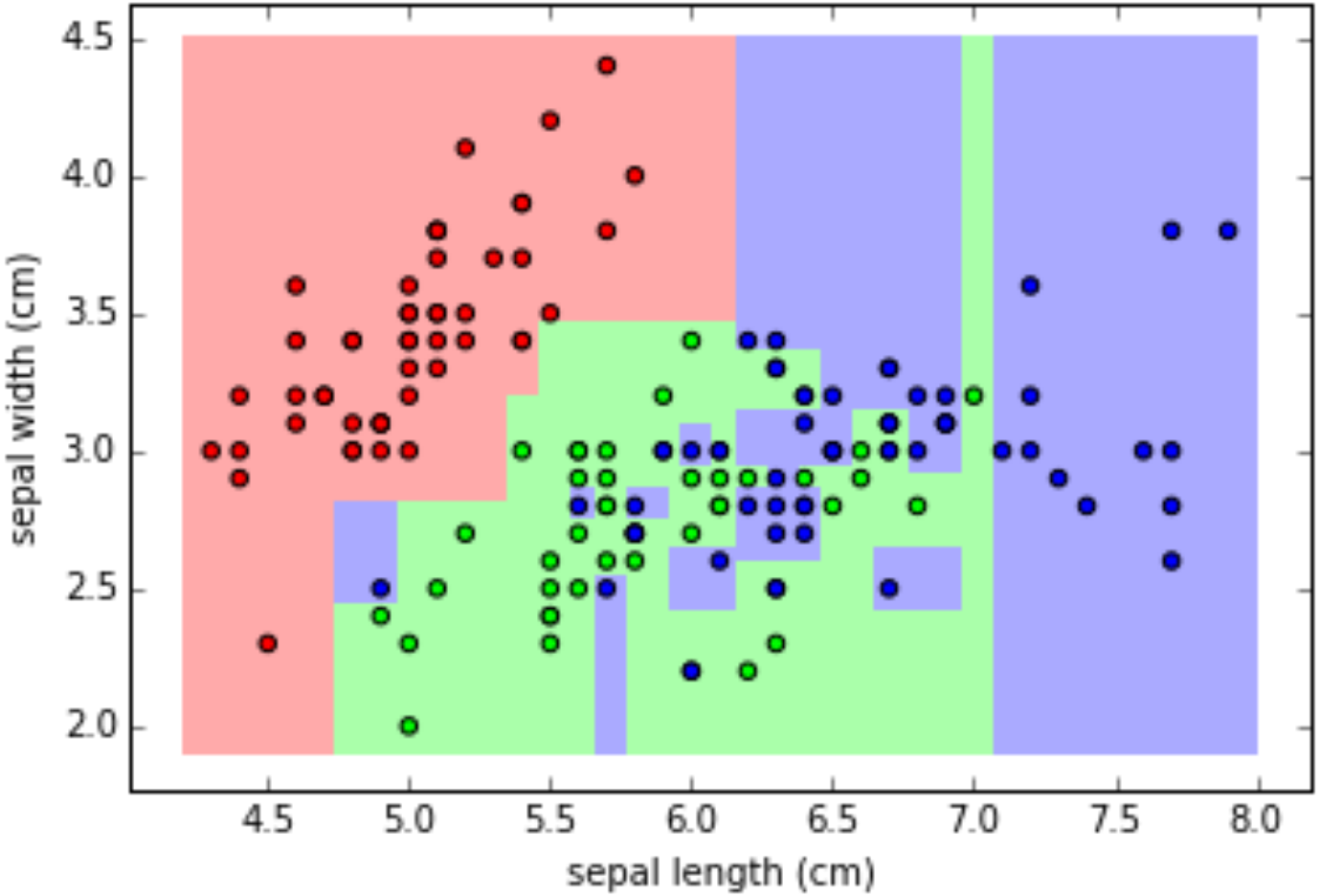

R_1
R_2
R_3
R_4
R_5
t_1
t_2
t_3
t_4
X_1
X_2


$X_1 \leq t_1$
$X_2 \leq t_2$
$X_1 \leq t_3$
$X_2 \leq t_4$
R_1
R_2
R_3
R_4
R_5


X_2
X_1

Overfitting

Jika terus dilakukan pemisahan, maka akan mengurangi JKG → terjadi overfitting



Perlu dilakukan proses **Pruning**

Pruning

Lebih sedikit pemisahan atau lebih sedikit wilayah dapat menurunkan ragam yang mana membuat interpretasi yang lebih baik, namun dengan mengorbankan lebih sedikit bias

Ide?

- Hentikan pemisahan ketika peningkatan JKG lebih rendah dari ambang batas (threshold)
  - Pohon lebih kecil tetapi tidak efektif (rabun dekat)
  - Pemisahan pada awal pohon mungkin akan diikuti dengan pemisahan yang sangat baik; pemisahan yang menyebabkan pengurangan JKG secara besar-besaran di proses selanjutnya

Pruning

Lebih baik menumbuhkan pohon besar dan kemudian mencari subpohon yang meminimalkan kesalahan pengujian

Bagaimana caranya?

Cross-validation of all possible subtrees?

This is too expensive

Cost complexity pruning—also known as weakest link pruning

Cost complexity pruning

Consider a tuning parameter α that for each value of α there is a subtree that minimizes

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

Where |T| is the number of terminal nodes. α controls the complexity of the tree similarly we saw with other regularizations (e.g. LASSO).

It turns out that as we increase α from zero in, branches get pruned from the tree in a nested and predictable fashion, so obtaining the whole sequence of subtrees as a function of α is easy.

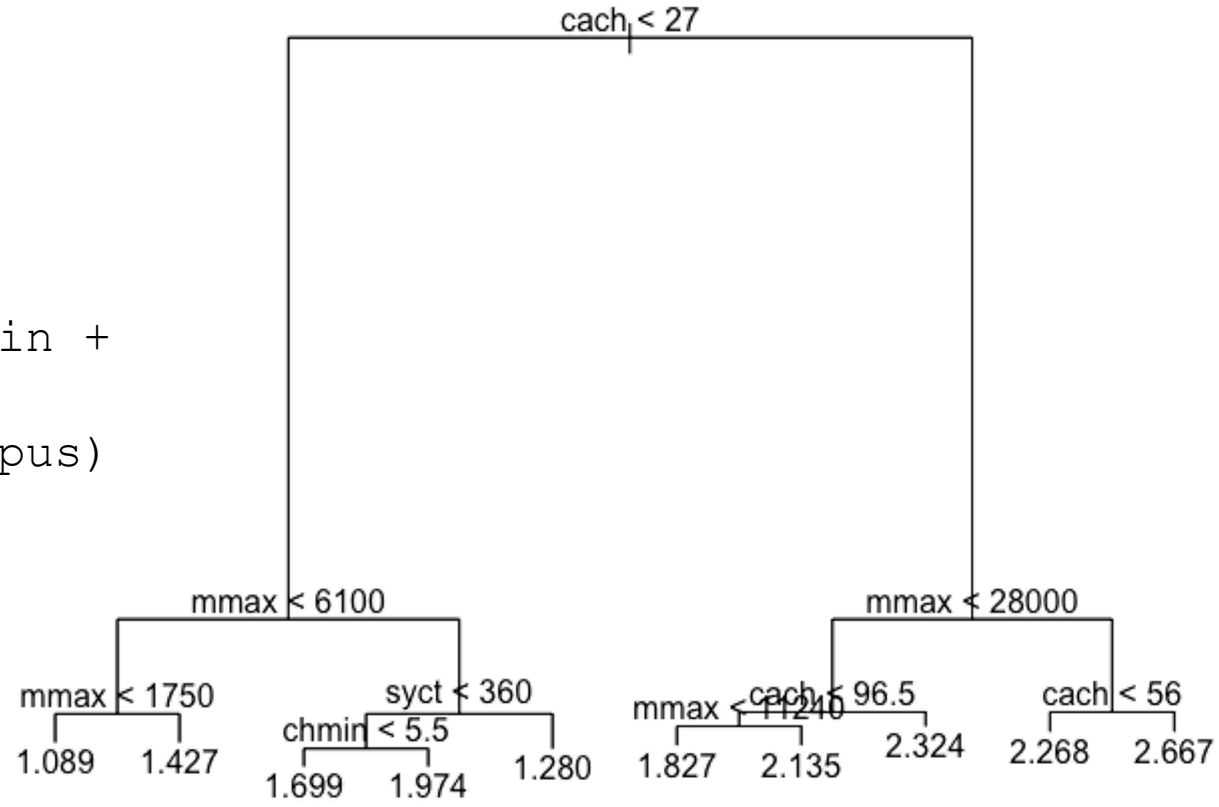
ALGORITHM FOR PRUNING

1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations
2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of α
3. Use K-fold cross-validation to choose α
   - Repeat #1 and #2 on the k-th fold
   - Estimate the MSE as a function of α
   - Average all and pick α
4. Return the subtree from Step 2 that corresponds to the chosen value of α

Ilustrasi di R

- Gunakan dataset “cpus” dari package “MASS”.
- Data ini berisi berbagai ukuran kinerja relatif dan karakteristik 209 CPU, yang akan dilakukan prediksi kinerja CPU-nya (perf).
  - Enam peubah prediktor:
    - syct : waktu siklus (ns)
    - mmin : minimum memori utama (KB)
    - mmax : maksimum memori utama (KB)
    - cach : ukuran cache (KB)
    - chmin : minimum banyaknya channel
    - chmax : maksimum banyaknya channel
  - Untuk membuat sebarannya lebih berbentuk simetri, akan dilakukan transformasi logaritma (log10) untuk peubah kinerja CPU (perf).

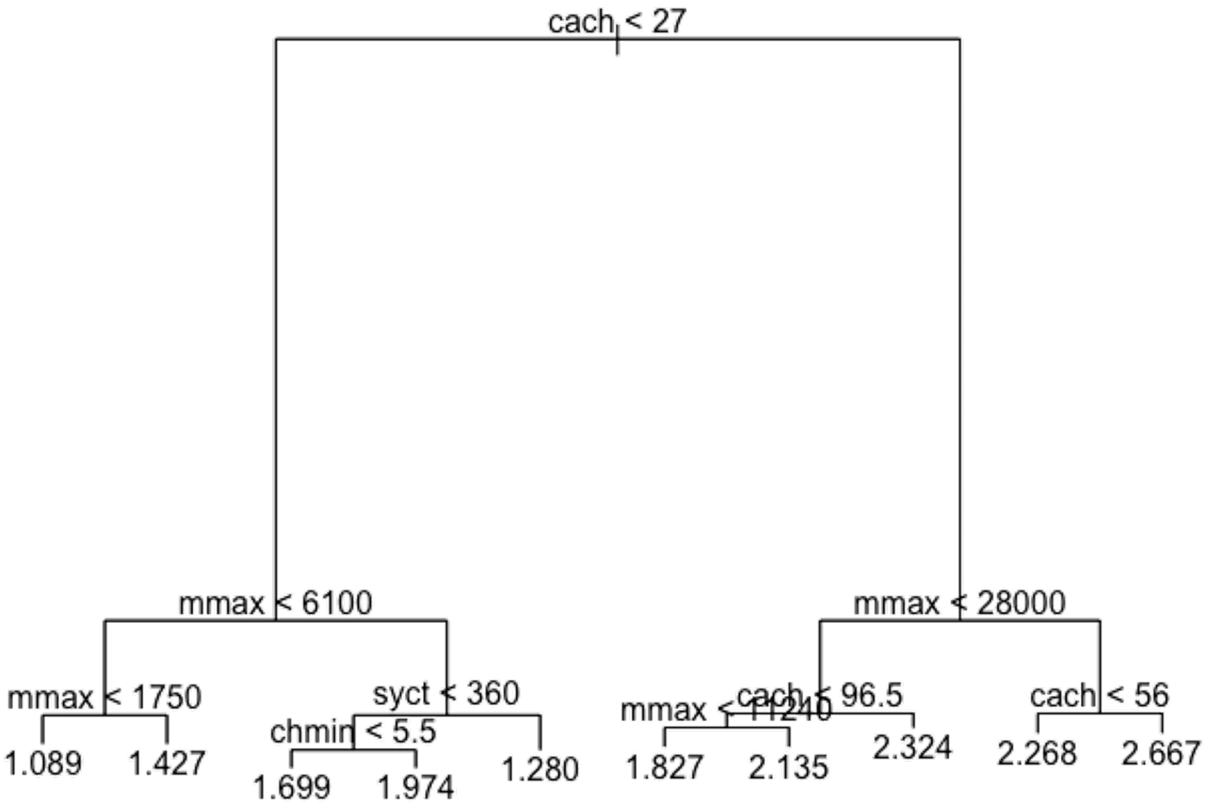
```
data(cpus, package="MASS")
summary(cpus)

#install.packages("tree")
library(tree)
cpus.ltr <- tree(log10(perf) ~ syct + mmin +
mmax + cach
                 + chmin + chmax, data=cpus)
plot(cpus.ltr)
text(cpus.ltr)
```



```
summary(cpus.ltr)

> summary(cpus.ltr)

Regression tree:
tree(formula = log10(perf) ~ syct + mmin + mmax + cach + chmin +
    chmax, data = cpus)
Variables actually used in tree construction:
[1] "cach"  "mmax"  "syct"  "chmin"
Number of terminal nodes:  10
Residual mean deviance:  0.03187 = 6.342 / 199
Distribution of residuals:
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-0.4945000 -0.1191000  0.0003571  0.0000000  0.1141000  0.4680000
```

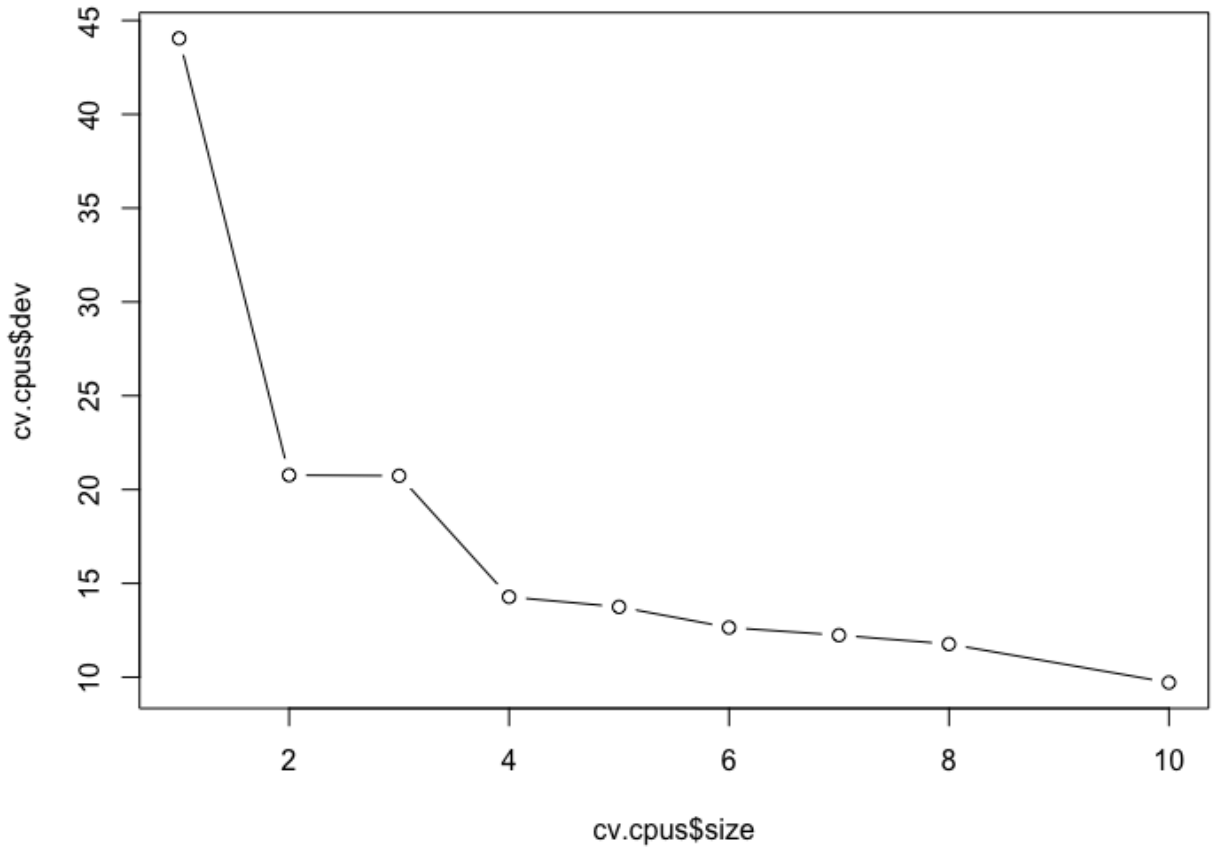
cpus.ltr

```
> cpus.ltr
node), split, n, deviance, yval
      * denotes terminal node

 1) root 209 43.12000 1.753
   2) cach < 27 143 11.79000 1.525
     4) mmax < 6100 78  3.89400 1.375
       8) mmax < 1750 12  0.78430 1.089 *
       9) mmax > 1750 66  1.94900 1.427 *
     5) mmax > 6100 65  4.04500 1.704
      10) syct < 360 58  2.50100 1.756
        20) chmin < 5.5 46  1.22600 1.699 *
        21) chmin > 5.5 12  0.55070 1.974 *
      11) syct > 360 7  0.12910 1.280 *
   3) cach > 27 66  7.64300 2.249
     6) mmax < 28000 41  2.34100 2.062
      12) cach < 96.5 34  1.59200 2.008
        24) mmax < 11240 14  0.42460 1.827 *
        25) mmax > 11240 20  0.38340 2.135 *
      13) cach > 96.5 7  0.17170 2.324 *
     7) mmax > 28000 25  1.52300 2.555
      14) cach < 56 7  0.06929 2.268 *
      15) cach > 56 18  0.65350 2.667 *
```



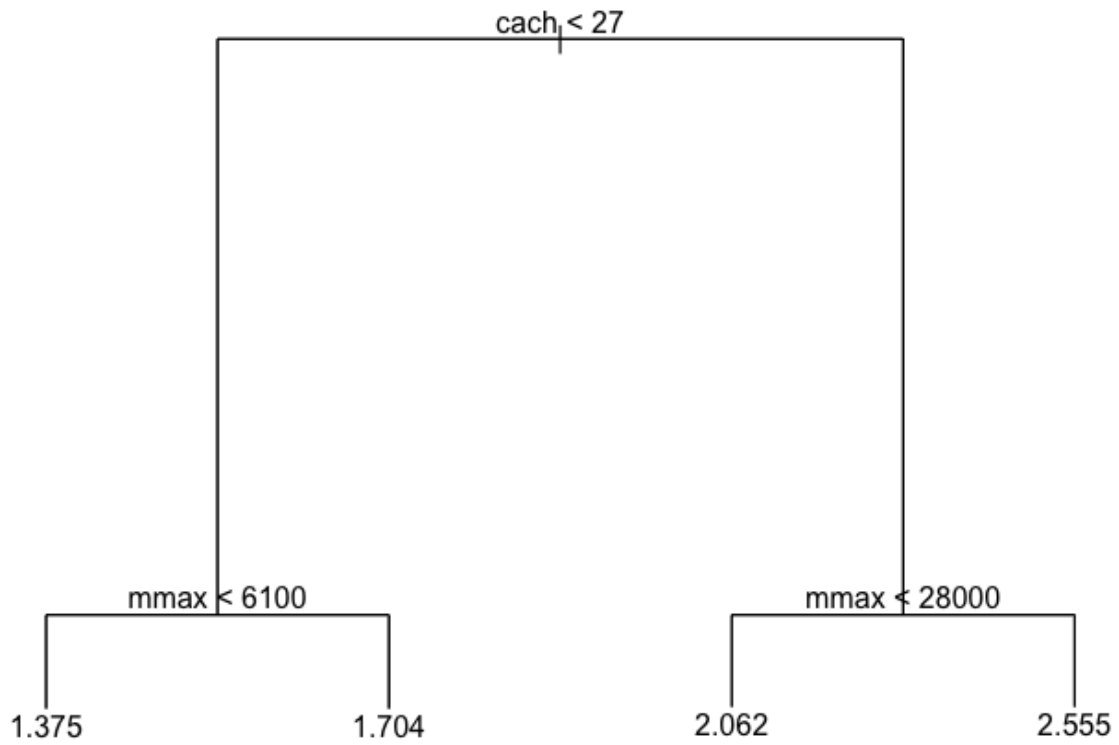
Pruning

```
cv.cpus <- cv.tree(cpus.ltr, , prune.tree)
plot(cv.cpus$size,cv.cpus$dev,type='b')
```



Dipilih ukuran 4

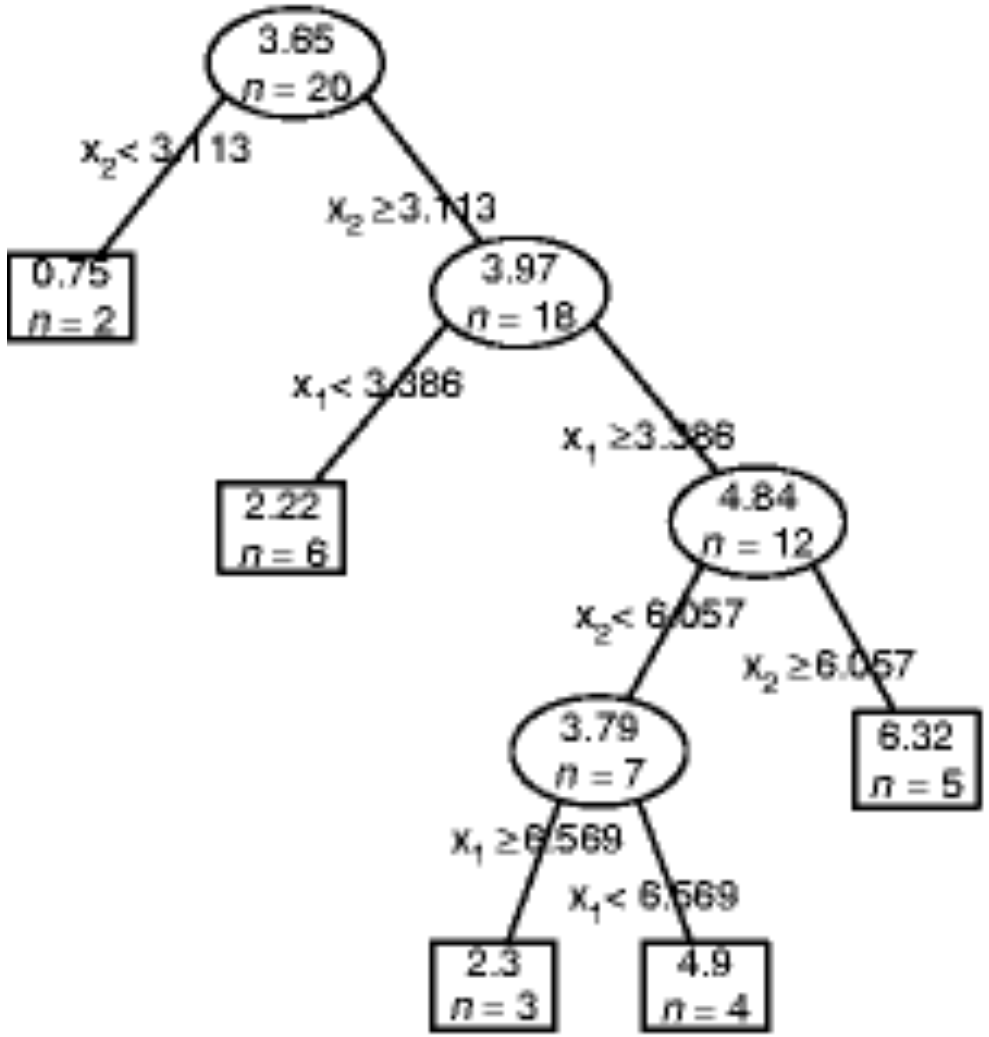
```
prune.cpus <- prune.tree(cpus.ltr,best=4)
plot(prune.cpus)
text(prune.cpus,pretty=0)
```



prune.cpus

```
> prune.cpus
node), split, n, deviance, yval
      * denotes terminal node

1) root 209 43.120 1.753
  2) cach < 27 143 11.790 1.525
    4) mmax < 6100 78  3.894 1.375 *
    5) mmax > 6100 65  4.045 1.704 *
  3) cach > 27 66  7.643 2.249
    6) mmax < 28000 41  2.341 2.062 *
    7) mmax > 28000 25  1.523 2.555 *
```

Latihan (isian singkat)



Diketahui regression tree di samping.

1. Berapa total pengamatan pada tree di samping?
2. Berapa kedalaman (depth) pohon di samping?

Tentukan nilai $\hat{y}$ dengan kondisi:

a. $x_1 = 2$ dan $x_2 = 4$
b. $x_1 = 6$ dan $x_2 = 4$
c. $x_1 = 3$ dan $x_2 = 4$
d. $x_1 = 6$ dan $x_2 = 6$

Tugas Kelompok – Sesi UTS

- Buatlah proposal penelitian mengenai Projek Kelompok-nya, yang di dalamnya berisi
  1. Judul berdasarkan topik projek yang ditentukan
  2. Latar belakang dan tujuan
  3. Data dan peubah-peubah yang digunakan
  4. Metodologi (rencana tahapan analisis data yang dilakukan)

- Selain proposal penelitian dalam format makalah, dikumpulkan juga file presentasi dalam powerpoint.

- File proposal penelitian dan file presentasi diupload pada class.ipb.ac.id

- Batas waktu pengiriman adalah **Hari Minggu, tanggal 3 Maret 2024 jam 23:59 WIB**

- Komponen Penilaian:
  - Kecepatan pengiriman
  - Kesesuaian isi proposal
  - Orisinalitas

- Selanjutnya, pada pertemuan 7, akan diadakan **Sesi Presentasi** sesuai dengan file presentasi yang dikirimkan, dengan aturan:
  - Kelompok dipilih secara acak maksimal berisi 6 mahasiswa
  - Penilaian sesi presentasi ini berdasarkan keaktifan dan kesesuaian pertanyaan/jawaban setiap mahasiswa pada forum diskusi yang ada

Topik Projek Kelompok

1. Topik 1: Supervised learning dengan peubah respon numerik
2. Topik 2: Supervised learning dengan peubah respon kategorik
3. Topik 3: Unsupervised learning dengan kasus penggerombolan
4. Topik 4: Unsupervised learning dengan kasus reduksi dimensi

ASSIGNMENT

Form Pengumpulan Tugas Kelompok Sesi UTS

Mark as done

Opens: Monday, February 5, 2024, 3:00 PM
Due: Sunday, March 3, 2024, 11:59 PM

•Buatlah **proposal penelitian** mengenai Projek Kelompok-nya, yang di dalamnya berisi
1.Judul berdasarkan topik projek yang ditentukan
2.Latar belakang dan tujuan
3.Data dan peubah-peubah yang digunakan
4.Metodologi (rencana tahapan analisis data yang dilakukan)

•Selain **proposal penelitian** dalam format makalah, dikumpulkan juga file **presentasi** dalam powerpoint.
•Batas waktu pengiriman adalah **Hari Minggu, tanggal 3 Maret 2024 jam 23:59 WIB**

Terima kasih ☺