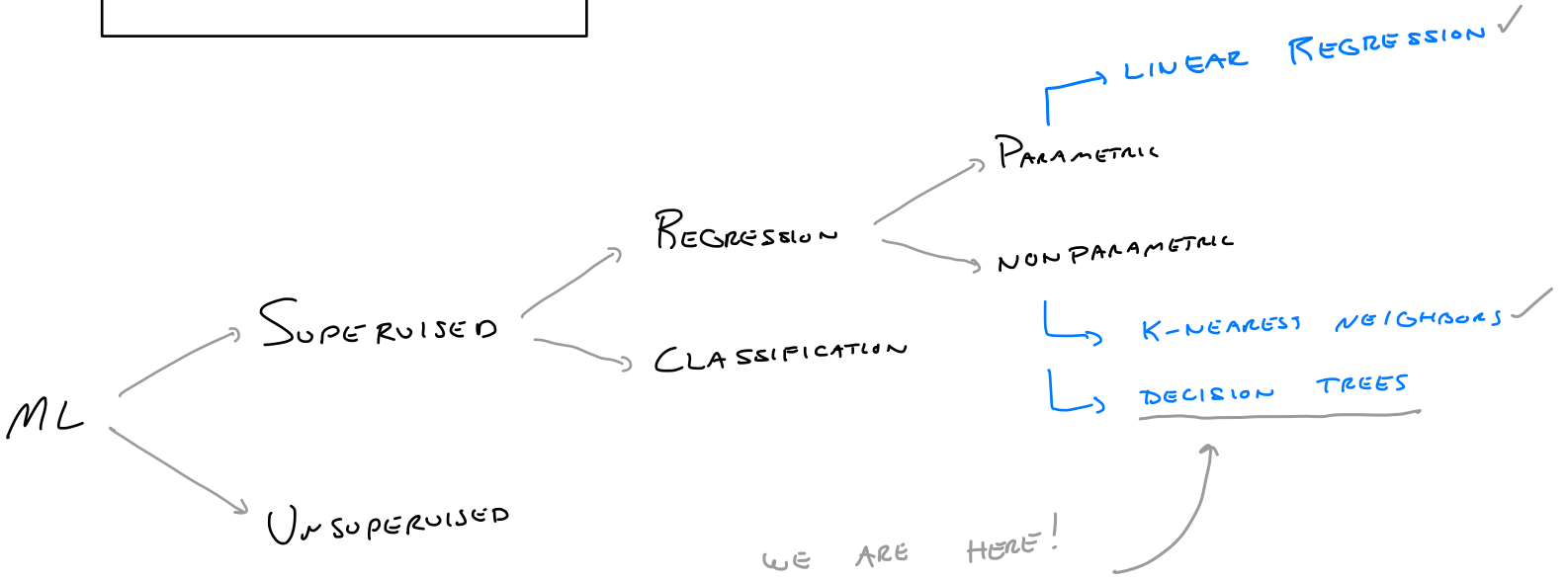
CS 307

Fall 2023

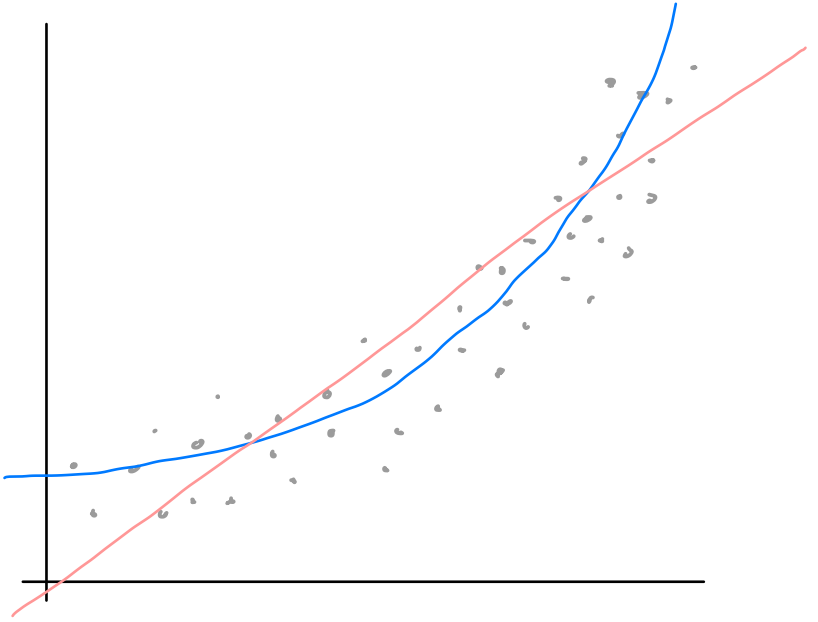
Dalpiaz

# Week 03

# Nonparametric Regression



| x | y |
| --- | --- |
| ⋮ | ⋮ |



WANT

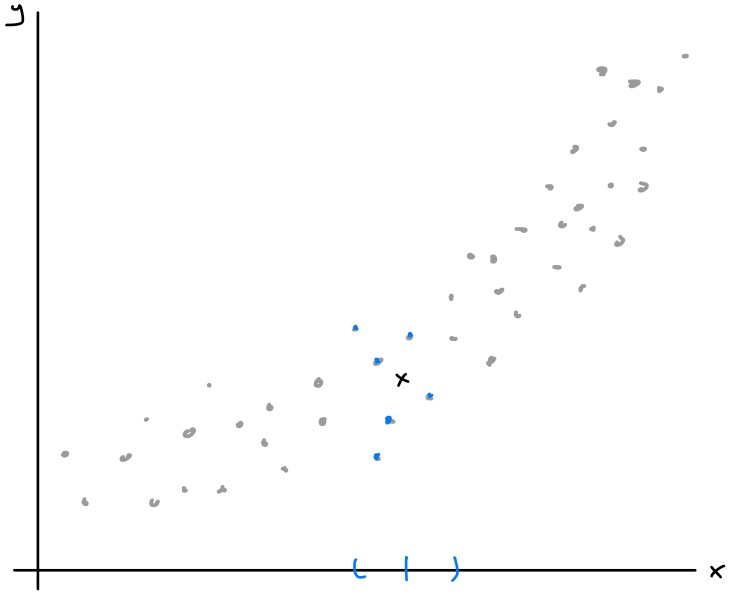
$\mathbb{E}[Y \mid X = x]$

ASSUME

$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$

ASSUME

$Y = \beta_0 + \beta_1 x + \varepsilon$




WANT

$\mathbb{E}[Y \mid X = x]$

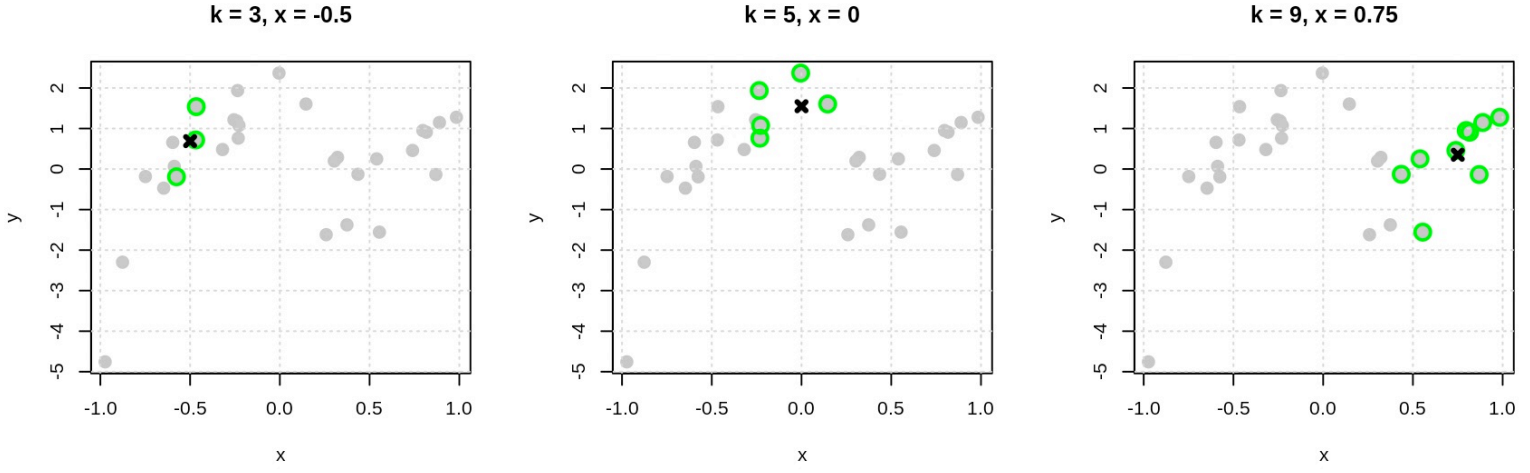
- $\hat{\mathbb{E}}[Y \mid X = x] = \text{AVE}\left(\{y_i \text{ WHERE } x_i = x\}\right)$ ← WON'T WORK
- $\hat{\mathbb{E}}[Y \mid X = x] = \text{AVE}\left(\{y_i \text{ WHERE } x_i \text{ "CLOSE" TO } x\}\right)$

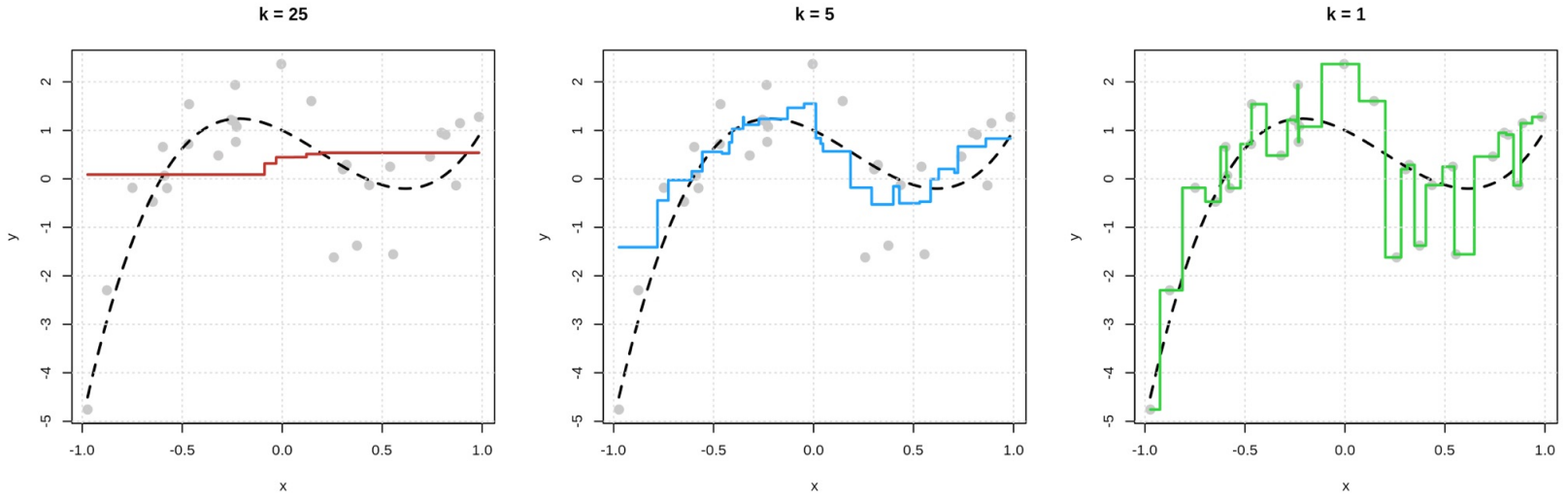
# k - NEAREST NEIGHBORS

TO ESTIMATE $\mu(x) = \mathbb{E}[Y \mid X = x]$

USE $\hat{\mu}_k(x) = \frac{1}{k} \sum_{\{i \,:\, x_i \in N_k(x, D)\}} y_i$

k OBSERVATIONS WITH $x_i$ NEAREST TO $x$


k = 3, x = -0.5
k = 5, x = 0
k = 9, x = 0.75
y
x
-5
-4
-3
-2
-1
0
1
2
-1.0
-0.5
0.0
0.5
1.0


k = 25
k = 5
k = 1
y
x
-1.0
-0.5
0.0
0.5
1.0
2
1
0
-1
-2
-3
-4
-5

# Tuning Parameters

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon$$

Model parameters

→ Learned from data

k in KNN

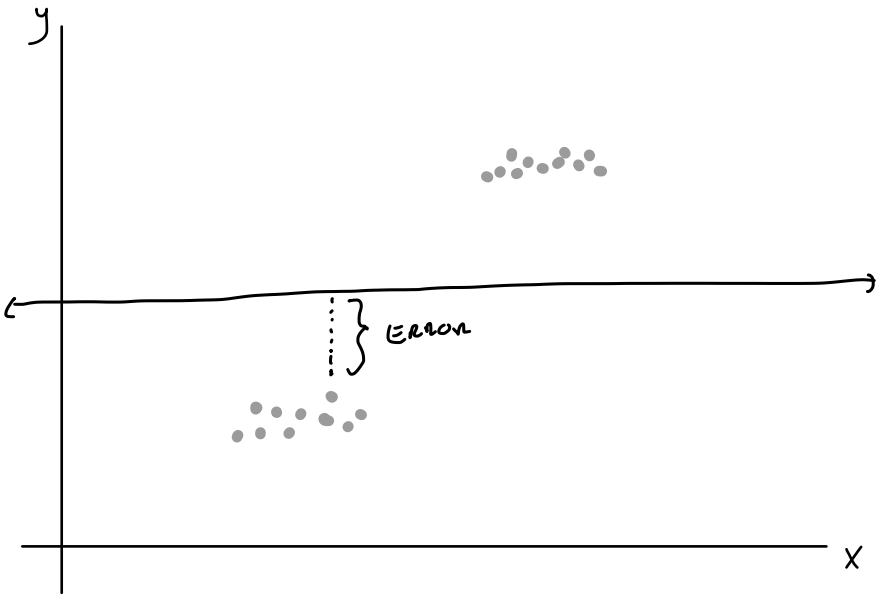
Tuning parameter

→ Defines how to learn from data

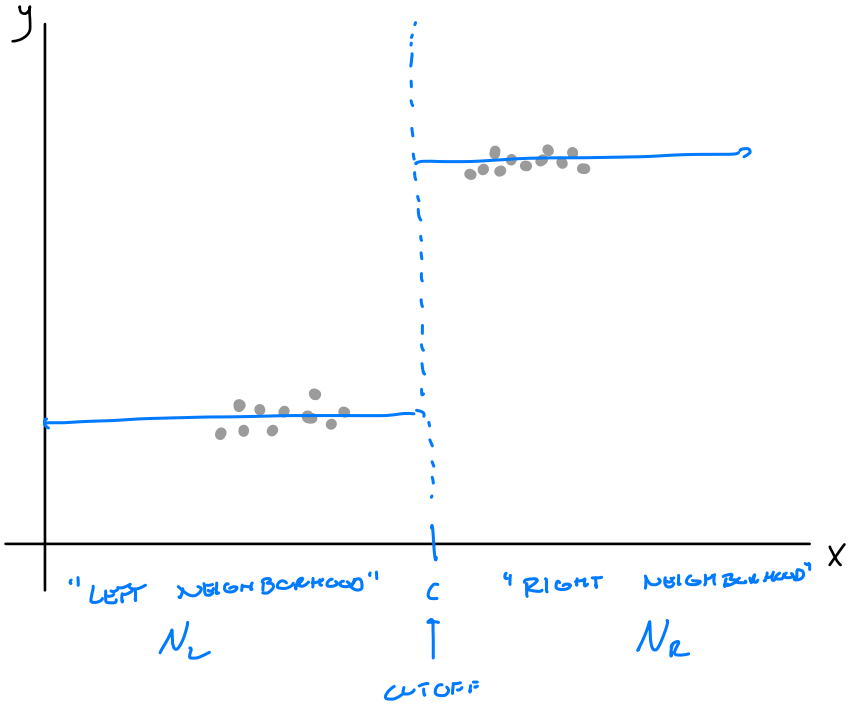
# OTHER KNN NOTES

- "FAST" TO TRAIN, "SLOW" TO PREDICT — LAZY!
- WHICH FEATURES SHOULD BE USED? — ???
- CATEGORICAL FEATURES? — DUMMY ENCODING
- HOW TO CALCULATE DISTANCE? — YOU PICK!
- FEATURE SCALING? — !!!

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$




$$\bar{y} = \frac{1}{n} \sum y_i$$




IDEA: FIND NEIGHBORHOODS, PREDICT AVERAGE OF $y_i$ IN NEIGHBORHOOD

# Decision Trees

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

FEATURE + CUTOFF

FIND "SPLIT" THAT MINIMIZES

$C$ = "CUTOFF"

$$\sum_{i \in N_L} \left(y_i - \hat{\mu}_{N_L}\right)^2 + \sum_{i \in N_R} \left(y_i - \hat{\mu}_{N_R}\right)^2$$

AVG $y_i$ IN $N_L$

AVG $y_i$ IN $N_R$

$i \in N_L$ ↓ $x < c$

$i \in N_R$ ↓ $x > c$

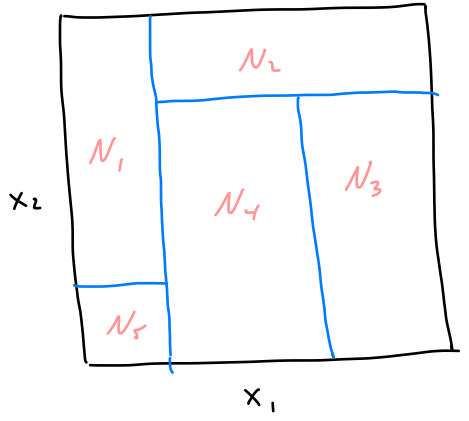
# Recursive Binary Partitioning

Do it again!

$$\sum_{i \in N_L} \left( y_i - \hat{\mu}_{N_L} \right)^2 + \sum_{i \in N_R} \left( y_i - \hat{\mu}_{N_R} \right)^2$$

$$\sum_{i \in N_{R_1}} \left( y_i - \hat{\mu}_{N_{R_1}} \right)^2 + \sum_{i \in N_{R_2}} \left( y_i - \hat{\mu}_{N_{R_2}} \right)^2$$

# Recursive Partitioning




# NEIGHBORHOODS

$$SSE = \sum_{j=1}^{J} \sum_{i \in N_j} \left( y_i - \hat{\mu}_j \right)^2$$

AVE $y_i$ IN $N_j$

$$R^2 = 1 - \frac{SSE}{SST}$$

# How to stop?

sklearn.tree.DecisionTreeRegressor

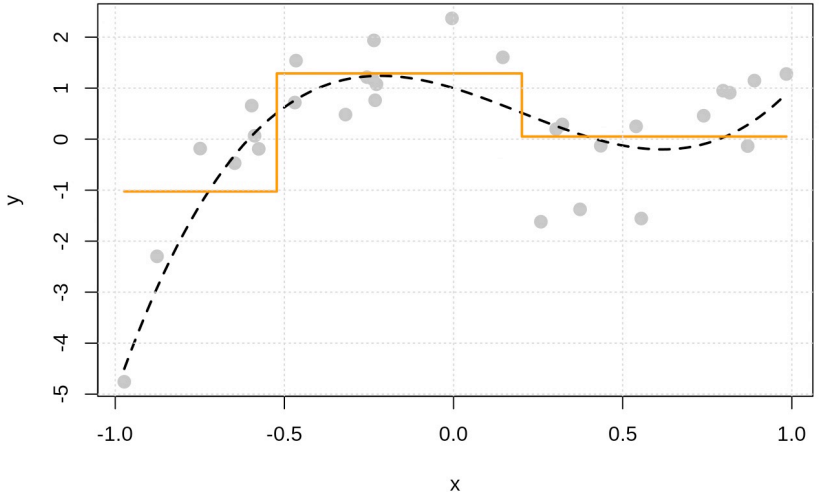
max_depth

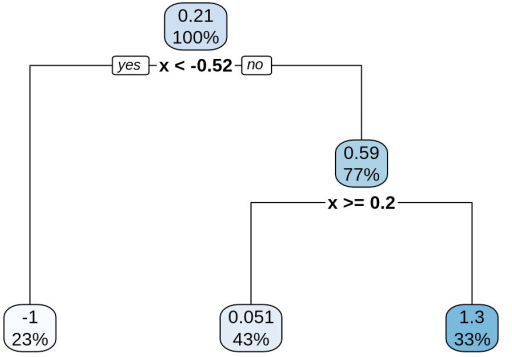
min_samples_split

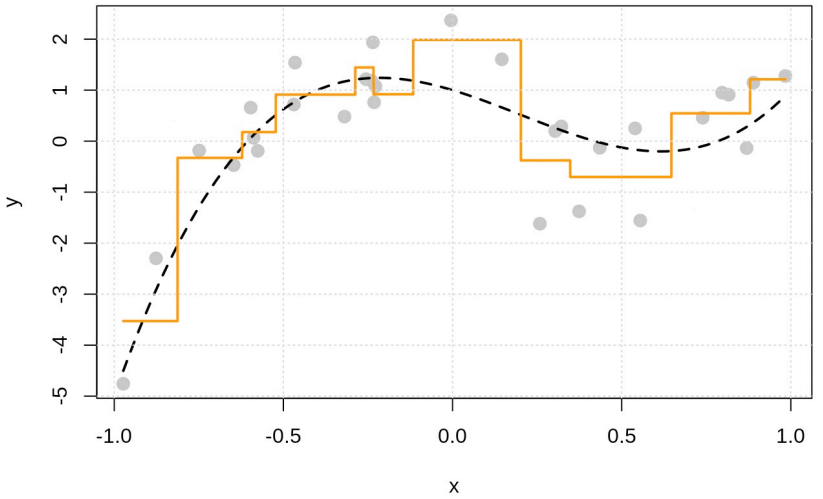
⋮

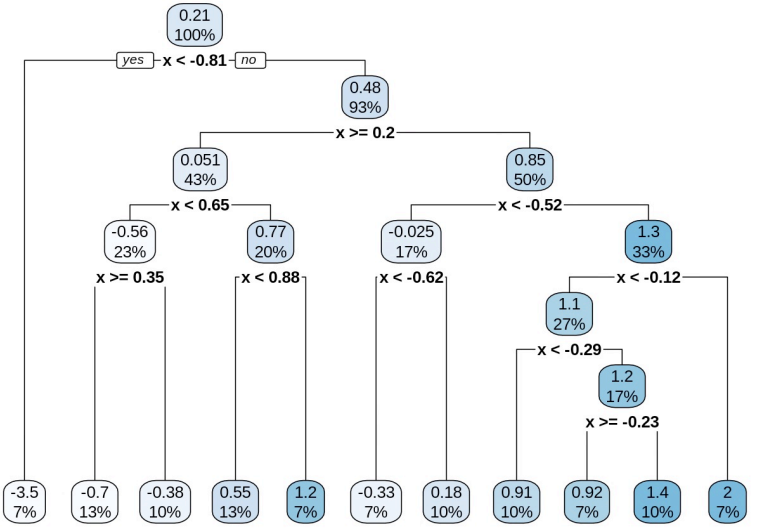
How "deep" (tall) can the tree grow?

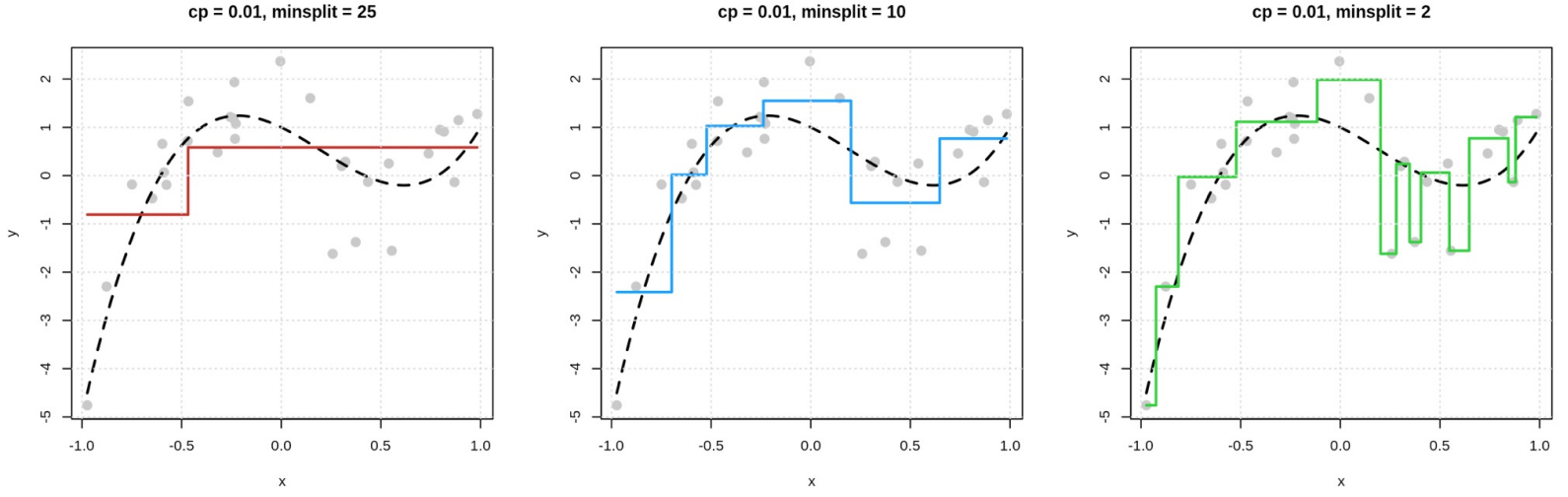
How big must a node be to consider a split?


y
2
1
0
-1
-2
-3
-4
-5
-1.0
-0.5
0.0
0.5
1.0
x


0.21
100%
yes
x < -0.52
no
0.59
77%
x >= 0.2
-1
23%
0.051
43%
1.3
33%


y
x
2
1
0
-1
-2
-3
-4
-5
-1.0
-0.5
0.0
0.5
1.0


0.21
100%
yes
x < -0.81
no
0.48
93%
x >= 0.2
0.051
43%
0.85
50%
x < 0.65
x < -0.52
-0.56
23%
0.77
20%
-0.025
17%
1.3
33%
x >= 0.35
x < 0.88
x < -0.62
x < -0.12
1.1
27%
x < -0.29
1.2
17%
x >= -0.23
-3.5
7%
-0.7
13%
-0.38
10%
0.55
13%
1.2
7%
-0.33
7%
0.18
10%
0.91
10%
0.92
7%
1.4
10%
2
7%


cp = 0.01, minsplit = 25
cp = 0.01, minsplit = 10
cp = 0.01, minsplit = 2
y
x
-5
-4
-3
-2
-1
0
1
2
-1.0
-0.5
0.0
0.5
1.0